UForge AppCenter Admin Documentation

Release 3.8

FUJITSU

Jun 10, 2019

Contents

1 Architecture Considerations **3**
- 1.1 UForge Platform Overview . . . 3
- 1.2 Reliability . . . 8
- 1.3 Security . . . 8
- 1.4 Storage Considerations . . . 8
- 1.5 Scalability through Partitioning . . . 10
- 1.6 Image Generations . . . 10
- 1.7 Estimating Scan Size . . . 11
- 1.8 Deployment Examples . . . 11
- 1.9 Network Topology . . . 12
- 1.10 Minimum Software Topology . . . 13
- 1.11 Security Options . . . 14
- 1.12 High Availability . . . 14

2 Installing UForge **15**
- 2.1 UForge Installation Overview . . . 15
- 2.2 UForge Repository Setup . . . 16
- 2.3 UForge Repository on Shared Storage . . . 17
- 2.4 UForge Repository on Local Storage . . . 17
- 2.5 Installing from an ISO . . . 18
- 2.6 Post-Installation Cleanup . . . 19
- 2.7 Configuring UForge . . . 19
- 2.8 Cloud Platform Default Ports . . . 22
- 2.9 Testing the Deployment . . . 22

3 Further AppCenter Configuration **25**
- 3.1 Modifying the UForge Platform External URL Endpoints . . . 25
- 3.2 Adding a Compute Node . . . 26
- 3.3 Removing a Node . . . 28
- 3.4 Configuring Apache and Tomcat Web Services to use SSL Certificate . . . 28
- 3.5 Configuring SMTP Proxy . . . 30
- 3.6 Configuring UForge Behind Enterprise Proxy . . . 31
- 3.7 Configuring the Web Service . . . 31
- 3.8 Configuring the Database . . . 32
- 3.9 Configuring the Scheduler . . . 33
- 3.10 Managing the Watchdog Services . . . 36
- 3.11 Tuning the Services . . . 38

3.12 Using the Event Bus . . . 42
3.13 Configuring Email Notification Service . . . 43
3.14 Modifying the UForge IP . . . 45
3.15 Customizing UForge Authentication for SSO . . . 48
3.16 Allowing https Repositories with Self-Signed Certificate . . . 51
3.17 Hosting Proprietary Packages . . . 51
3.18 Populating ISO Skeleton . . . 54
3.19 Configuring Cloudsoft AMP . . . 54
3.20 Additional Configuration for Azure Stack . . . 56
3.21 Configuring Secret Manager . . . 56

4 Updating your UForge AppCenter **59**
4.1 Updating an Existing UForge Deployment . . . 59
4.2 Migrating UForge from 3.7 to 3.8 . . . 61
4.3 Going Back to a Previous Version of a UForge Deployment . . . 64
4.4 Retrieving Data from UForge . . . 64
4.5 Sending a Request to UForge . . . 64

5 Backup Overview Guidelines **67**
5.1 Backup Overview Guidelines . . . 67
5.2 Backup Recommendations . . . 68
5.3 Using Master-Slave Replication for Database Backup . . . 68
5.4 UForge Databases Basic Backup . . . 69
5.5 Basic Restore . . . 70
5.6 User Data Backup . . . 71

6 Managing Services **73**
6.1 Starting and Stopping the Application Server . . . 73
6.2 Starting and Stopping the Database . . . 73
6.3 Starting and Stopping the Secret Manager . . . 74

7 Managing Resources **75**
7.1 Managing the Project Catalog . . . 75
7.2 Allowing Access to Image Formats . . . 81
7.3 Managing the Organisation Artifact Accounts . . . 84
7.4 Adding OS Distributions . . . 85
7.5 Listing the Installed OSes . . . 86
7.6 Listing the Enabled OSes with CLI . . . 86
7.7 Adding an OS to an Organization . . . 87
7.8 Creating a Repository Using UI . . . 90
7.9 Updating a Repository Using CLI . . . 90
7.10 Removing OSes and Distributions . . . 90
7.11 Updating an OS Repository . . . 91
7.12 Deleting an OS Repository . . . 92
7.13 Creating Custom OS Profiles with UI . . . 92
7.14 Editing Custom OS Profiles Using UI . . . 93
7.15 Official UForge Tool Repositories . . . 94
7.16 Specific UForge Tool Repositories . . . 95
7.17 Adding RPM Type OSes Using CLI . . . 96
7.18 Adding DEB Type OSes Using CLI . . . 99
7.19 Microsoft Windows and UForge . . . 102
7.20 Creating a Golden Image from Scan . . . 104
7.21 Creating a Golden Image Manually . . . 106
7.22 Creating and Managing Categories . . . 114
7.23 Creating and Managing Milestones . . . 114

8 Managing Users **117**
8.1 Managing User Accounts . . . 117
8.2 RBAC Overview . . . 118
8.3 Granting a User Administrator Rights . . . 123
8.4 Setting Quotas . . . 123
8.5 Managing User Access to Operating Systems . . . 126
8.6 Managing User Access to Formats . . . 127
8.7 Granting a User API Access . . . 128
8.8 Granting a User Supervisor Rights . . . 129

9 Monitoring Overview **131**
9.1 Viewing the Web Service Logs . . . 131
9.2 Viewing Current Jobs in the Scheduler . . . 132
9.3 Viewing the Logs of a Job . . . 132
9.4 Configuring UForge to Send Logs to Elastic Stack . . . 133
9.5 Using Supervisor Mode . . . 134
9.6 Getting Support . . . 135

10 UForge Tooling **137**
10.1 Using the CLI Tool . . . 137

11 Rebranding Your UForge GUI **139**
11.1 Creating Dedicated Image Directory . . . 139
11.2 Modifying the Sign-In and Sign-Up Page . . . 140
11.3 Modifying the Signed In Header . . . 141
11.4 Modifying the Footer . . . 142
11.5 Restricting Change Password . . . 144
11.6 Restricting User Profile Usage . . . 144
11.7 Restricting Formats . . . 144
11.8 Restricting the Cloud Accounts . . . 144
11.9 Customizing the CSS . . . 145
11.10 Customizing the Platform . . . 145
11.11 Customizing the Portal Text . . . 146
11.12 Troubleshooting . . . 147

12 3rd Party Components **149**
12.1 Trademarks . . . 149
12.2 Copyright FUJITSU LIMITED 2019 . . . 149
12.3 3rd Party Components Used . . . 149

13 GDPR Compliance **155**
13.1 User Data in UForge . . . 155
13.2 Retrieving User Data . . . 155
13.3 Updating User Data . . . 159
13.4 Exporting User Data for Portability . . . 159
13.5 Deleting User Information . . . 160
13.6 Deleting User Information Using SQL . . . 162

14 Changelog **165**
14.1 3.8.fp13 . . . 165
14.2 3.8.fp12 . . . 166
14.3 3.8.fp11 . . . 167
14.4 3.8.fp10 . . . 168
14.5 3.8.fp9 . . . 169
14.6 3.8.fp8 . . . 170

14.7 3.8.fp7 . . . 172
14.8 3.8.fp6 . . . 173
14.9 3.8.fp5 . . . 174
14.10 3.8.fp4 . . . 175
14.11 3.8.fp3 . . . 176
14.12 3.8.fp2 . . . 177
14.13 3.8.fp1 . . . 179
14.14 3.8 . . . 181
14.15 3.7.fp8 . . . 182
14.16 3.7.fp7 . . . 184
14.17 3.7.fp6 . . . 185
14.18 3.7.fp5 . . . 186
14.19 3.7.fp4 . . . 187
14.20 3.7.fp3 . . . 188
14.21 3.7.fp2 . . . 190
14.22 3.7.fp1 . . . 192
14.23 3.7 . . . 193
14.24 3.6.fp2 . . . 194
14.25 3.6.fp1 . . . 196

15 Trademarks **199**

16 Copyright FUJITSU LIMITED 2019 **201**

17 High Risk Activity **203**

18 Export Restrictions **205**

This documentation is intended for Administrators of the UForge AppCenter. It covers an architecture presentation, installation instructions, as well as all the necessary steps to configure, manage and monitor your AppCenter once installed.

Note: There are multiple options for reading this documentation - click on the link at the lower left hand corner for these options.

Contents:

CHAPTER 1

Architecture Considerations

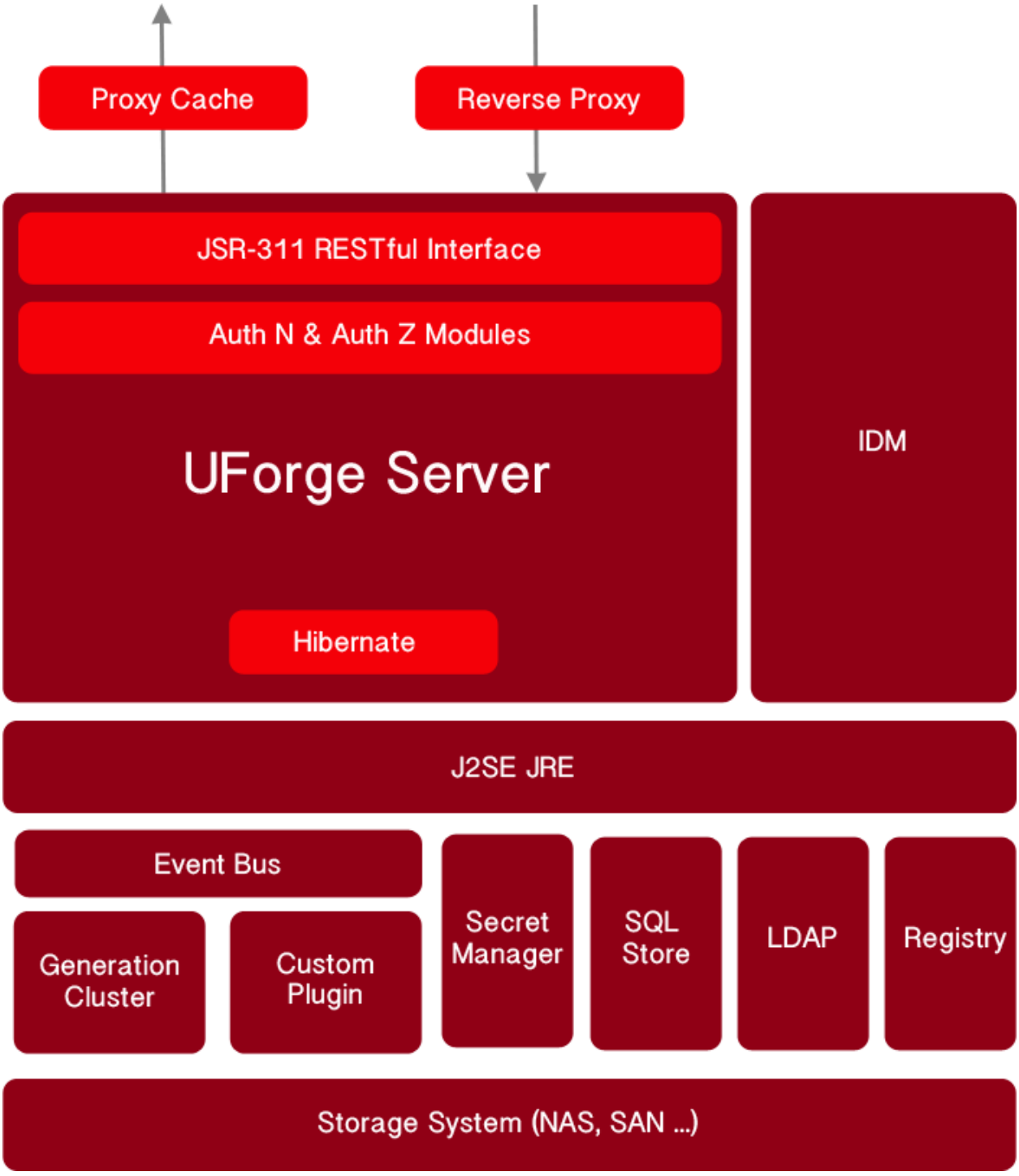
The following sections describe in detail things to consider when deploying a UForge platform.

1.1 UForge Platform Overview

UForge is a scalable multi-tenant platform. UForge can be split into the following distinct parts.

- UForge Server – This contains all the business logic of UForge, handling all incoming user requests.
- Meta-data SQL Store – A database holding all the configuration information and data of the platform.
- Secret Manager - A dedicated service to store user's confidential information, such as password, login ... in a secure way.
- LDAP Service – LDAP holding user authentication and access information
- Registry - Docker
- IDM Service – Authentication and authorization module
- Generation Cluster – A grid engine for scheduling and executing image generations
- Event-bus – Rabbit MQ
- Proxy cache - Squid

The UForge platform can be deployed on physical machines or in a virtualized or cloud environment.



UForge Server. The UForge Server is a RESTful (Representational State Transfer) web service built on top of Java and using the JSR-311 reference implementation (project Jersey). UForge Server is based on the design principles of REST and Resource Oriented Architecture (ROA). Resources are references with a unique global identifier (URI). UForge Server uses the semantics of the HTTP protocol to manipulate these resources. The HTTP response codes are used to determine whether a user's request was treated successfully or not.

Information is returned to the client in either XML or JSON, depending on the `Accept-Type` header attribute used by the client. If no `Accept-Type` header is provided, XML is returned by default. To ensure security, communication with the UForge Server is done via HTTPS. UForge Server provides authentication (AuthN) and authorization

(AuthZ) modules. These modules can be customized by the customer to provide or use an alternative mechanism for authentication and authorization. By default these modules communicate with the IDM service. This service determines whether a request has the correct authentication and the correct access.

The UForge Server interacts with the SQL Store using **Hibernate**. Hibernate is a high-performance Object/Relational persistence and query service. Hibernate maps from Java classes to database tables and from Java data types to SQL data types. It provides data query and retrieval facilities, providing a buffer between the two data representations and enables a more elegant use of object orientation on the Java side. The UForge Server is deployed inside a web/application server container. UForge, by default, uses Tomcat as the application server container.

Proxy Cache. UForge AppCenter includes a proxy cache. The proxy used is Squid. It is used for caching and centralizes all outgoing traffic. This improves the performance of UForge, specifically for the population of distribution repositories.

IDM Service. The UForge Identity management module is based on Apache Syncope (APL2 license). The module allows management of auditing (reports), policies, roles, users, tasks and entitlements. At present UForge uses this module for authentication and role-based access control (authorization). The persistence store for the UForge IDM is the SQL Store (described below) and there is a resource connector to the LDAP Service for storage of roles/users and entitlements in an ldap repository backend. The IDM service is deployed inside a web/application server container. UForge, by default, uses Tomcat as the application server container.

This IDM Service can be extended to provide a wider identity and access management (IAM) integration to an existing enterprise or corporate IAM system or to a brand new standards compliant IAM system(s) by using open source industry standard resource connectors.

LDAP. This service is an industry standard powerful AuthN ldap (v3) pure java server, based on the open source ForgeRock's OpenDJ offering. This can be run in single instance mode or multi-master replication mode for robustness.

Registry. UForge includes a Registry to store Docker images.

SQL Store. This is a relational database holding all the meta-data of UForge. Meta-data includes such items as:

- Appliance information including which operating system packages are included, install profile, middleware and configuration information.
- Images that have been generated from an appliance
- Images that have been published to a virtual or cloud platform
- Package information for an operating system
- Third party project software components

By default MariaDB is used as the SQL Store.

Secret Manager. This is a tool for securely accessing confidential data using Vault. A secret is anything that you want to tightly control access to, such as API keys, passwords, or certificates. A global token is used by the UForge Server to store secrets and generate tokens for external use such as publication, deployment ... These secrets are stored and encrypted on the disk at `/var/lib/vault` by using the Filesystem storage backend of Vault. This component is by default installed on the same node as the **SQL Store**.

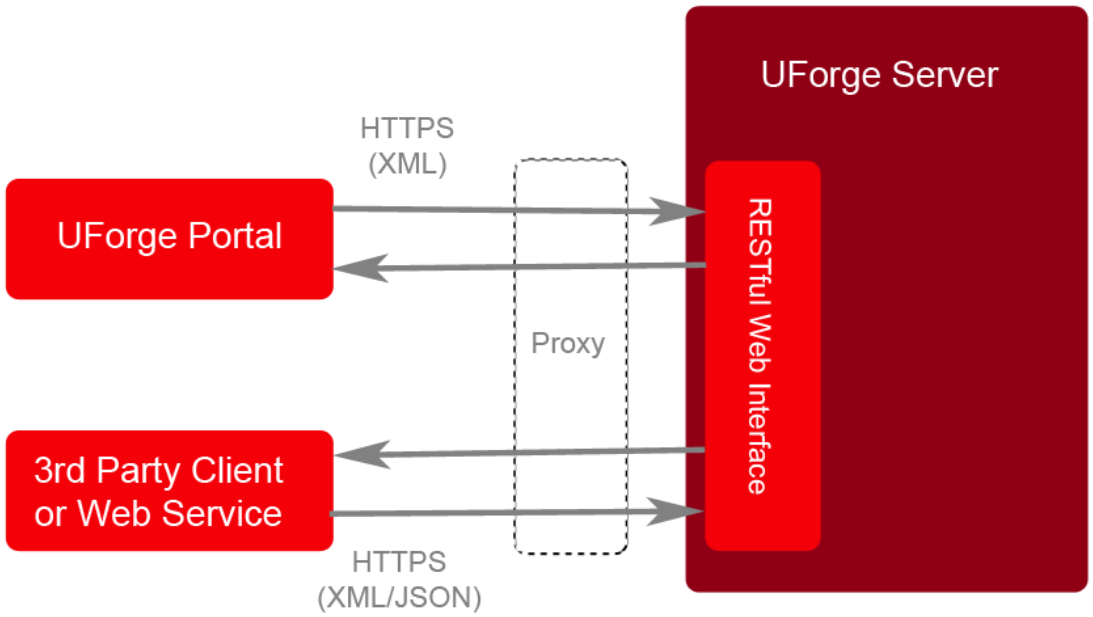
Generation Cluster. Image generation is I/O intensive and may take several minutes to complete. Consequently an HPC cluster is used to execute image generation jobs. There are two parts to this cluster:

- One or more compute nodes to execute a generation job
- A resource management system or batch scheduler that manages the reservation and access to the compute nodes

The resource management system holds its configuration information inside the meta-data SQL Store. The generation cluster is based on an open source project called OAR that is heavily used in production clusters of over 5000 nodes capable of handling over 5 million jobs.

Event-bus. UForge AppCenter uses RabbitMQ as an event bus. This event bus allows UForge administrators to track a number of user events on the platform. For more information on managing Rabbit MQ see: http://rabbitmq.com

UForge Clients. UForge provides a client, called UForge Portal that connects to UForge Server via HTTPS. This provides an interactive, visual experience when designing appliances or application stacks that can be deployed on physical, virtual and cloud environments. Consequently, UForge does not need to provide any presentation information to the client, greatly reducing the information (and therefore bandwidth) sent to the client.



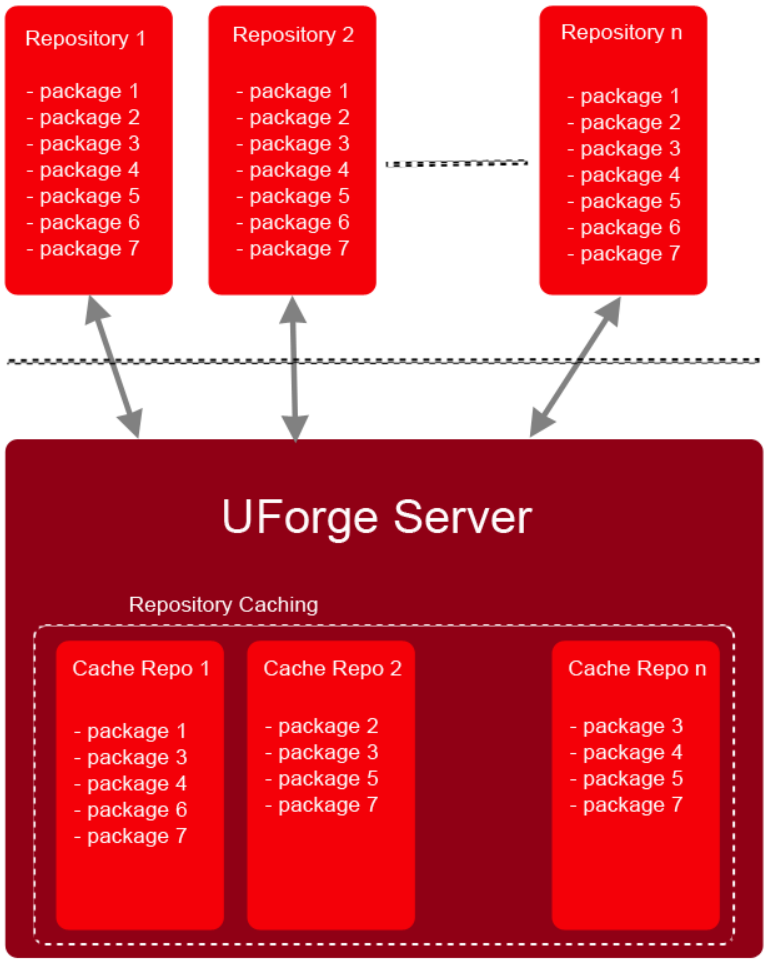
UForge allows users to implement their own clients to interact with UForge. UForge provides a RESTful API, allowing businesses to create a mashup and expose certain functionalities of UForge in their websites and portals. UForge provides two SDKs:

- Java
- Python

Users can use other languages to communicate with UForge.

1.1.1 Repository Caching

UForge AppCenter uses a repository cache in which it stores OS repositories as new images are generated. This means that the cache is empty when the AppCenter is installed and will be populated as images are generated by users. The cache ensures that all the versions a user needs are always available to generate images. As users generate images, the AppCenter connects to the official repositories to get the repositories and stores them in the cache.



1.1.2 Infrastructure Setup

UForge can be installed either on physical machines or in a virtual or cloud environment. The minimal installation requirements for UForge are:

- One physical or virtual machine where UForge will be installed
- A NAS or SAN for storage

1.1.3 UForge AppCenter Node Prerequisites

The UForge AppCenter components can be run on one physical or virtual machine, or can be distributed over several physical or virtual machines for scaling and reliability.

The UForge AppCenter requires the following hardware:

- CPU: 64-bit, 8 or more cores
- RAM: 16GB or more
- Local Hard Drive: 400GB
- NAS/SAN Storage: 200 GB (though this might be much more depending upon the usage)

This is the minimum for an "all in one" solution. For more information, refer to *Minimum Software Topology*.

1.2 Reliability

Fault tolerance is an important consideration for large-scale deployments. UForge platform has several types of data that must be replicated.

- The meta-data stored in the SQL store. The meta-data SQL store is replicated by using either master-slave or clustering. Both these configurations are supported by Percona Server. These configurations help scale-out the system and provide a level of fault tolerance if one of the database servers fails.
- The secret manager data stored on the filesystem
- The LDAP service data can be replicated over multiple LDAP instances via MMR (Multi-Master Replication)
- Binary data including operating system packages, project packages, uploaded software packages, license files, generated images and logo images. The binary data is stored on a storage system. This can be on a local filesystem of the database or on a NAS or SAN. This data is transparently replicated using RAID techniques.

In order to make the web service tier fault tolerant, multiple web servers can be deployed and load balanced. The administrator may also wish to have multiple load balancers in case the load balancer itself fails.

1.3 Security

UForge communicates with clients using HTTPS to ensure a secure connection. However, when deploying UForge, other security measures should be considered:

- Add a firewall in front of the web service tier, to only expose the HTTPS port.
- Provide a logical sub-network to protect the database, secret manager, LDAP, storage and generation cluster (DMZ).

1.4 Storage Considerations

The UForge AppCenter requires a variable amount of storage, depending, among other things, on the number of image generations and users. During the configuration it is important to size the local storage the various node instances making up the deployment and, if any, the size of the shared remote storage.

Local storage is used for installing the UForge service software and free disk space for log files.

> **Warning:** Logs for OpenDJ are stored under `/opt/OpenDJ/logs` and general usage logs for Tomcat and syncope are stored in subdirectories of `/var`:
>
> - `/opt/Tomcat/logs` is a symlink to `/var/log/tomcat`
> - `/opt/syncope/log` is a symlink to `/var/log/tomcat/syncope`
>
> Therefore, it is important to make sure there is enough space available under `/opt` and `/var`.

The SQL Store uses the local storage to store all the database meta-data. Finally a compute node used to generate an image requires local free disk space to generate one or more images from an appliance template.

A remote storage system (NAS or SAN) is used to store:

- Operating system cache repositories (this will be empty at the beginning and will grow as images are generated)
- Project catalog package binaries
- Binaries uploaded by users using UForge
- Generated images (copied from a compute node once the generation is complete)

For small deployments, the remote storage system can be on a local filesystem of the generation cluster.

Operating System Cache Repositories and Projects. The OS repositories are no longer stored in the UForge AppCenter. UForge AppCenter uses cache repositories in order to generate appliance templates. Therefore its size will grow and is completely variable. What consumes space are the images generated and the binaries uploaded in MySoftware. You can limit the space used by setting user quotas (refer to *Setting Quotas*).

The size of projects also vary, however they are usually in the 10 to 100 MB range. Consequently they take up a very small percentage of the entire disk space.

Image Generation. Compute nodes require local storage to create the disk images during the generation phase of an appliance template. To calculate the local storage required by a compute node, the generation of an image needs to be understood. There are six phases to the generation of an image:

- Phase 1: Check if the packages are in the cache. If they are not, download them and save to cache.
- Phase 2: Create a virtual disk for package installation.
- Phase 3: Install the packages, binaries and files to the disk (at this stage the local copy of the packages are deleted to save space).
- Phase 4: Convert the disk to the required target format.
- Phase 5: Compress the created disk.
- Phase 6: Copy the image to a more permanent storage (the remote storage so that the image can be downloaded or published by the user). At this stage the created disk is deleted.

At any given time during the construction of an image, the compute node requires three times as much space as the final image being generated. Since a compute node can generate more than one image simultaneously, the local disk storage for a compute node can be calculated as follows:

> average image size x simultaneous generations x 3

Image sizes vary from 300 MB to 12 GB. Note this is the disk size required to install all binary packages. A UForge compute node uses 'sparse' filesystem so that 'free' disk space required by a virtual machine does not need to be allocated during the generation of that virtual machine.

For a compute node configured to generate 10 images simultaneously and assuming that an average image size is 8 GB, the minimum local storage required by a compute node is:

> 8 x 10 x 3 = 240 GB

User Disk Space. Each user on UForge can generate images and also upload their own software. The user software is stored on the remote storage. Once an image has been generated it is also stored on remote storage allowing the user to download or publish the image directly to a virtual or cloud platform at a later stage. User uploaded software is usually a small percentage of the disk space compared to generated images. Note, that as the user has the appliance template, an image can be regenerated at any time, therefore, images can be deleted to save disk space. The total disk space required to store uploaded software and user images can be calculated as follows:

> user disk quota x total number of users

For example, a UForge platform providing all the operating systems UForge supports for 100 users, each having a quota of 18 GB (equivalent to having a maximum of 15 stored images of 1GB each and 3GB of uploaded software), the remote storage required is: 1880 GB (1.88 TB). When calculating, you should consider an addition 10% margin, plus 20 GB for UShareSoft bits (installer, ISO, etc).

Depending on the replication policy used for the remote storage, the total remote storage may have to be larger. Note that it may not be necessary to replicate all the information stored in the remote storage. Only important information must be replicated, including:

- the uploaded user software
- operating system repositories (this may be optional if there are backups of the operating systems elsewhere.)

Images generated are usually not replicated, as the image can be regenerated from the appliance template stored in the SQL Store.

1.5 Scalability through Partitioning

UForge was designed from the ground up to scale to meet the needs of businesses and service providers with 100,000s of users. The key to scaling is partitioning. Effective partitioning is based on leveraging "locality of reference" for both processing and data – if certain servers are specialized to solve a subset of the bigger problem, then the essential code and date are more likely to be in memory or close at hand. Partitioning techniques include vertical partitioning of functional tasks and horizontal partitioning of data and associated processing. Partitioning also helps to implement security to the platform.

Partitioning is increased by other distributed system techniques like automated replication, load balancing and failover.

Vertical Partitioning allows complex processing tasks to be divided into subtasks that can be independently optimized and managed. Vertical partitioning in UForge primarily consists of off-loading or splitting the I/O intensive generation tier, web service tier and database tier.

This allows the administrator to scale-up independently the number of CPUs, RAM and disk size for each of the tiers.

Horizontal Partitioning is crucial for large scale deployments. UForge is horizontally partitioned for larger deployments where thousands of users are required to interact simultaneously with the system.

One of the big bottlenecks, however, in such architectures is the database. When scaling out the web service tier, the number database reads and writes increases, and therefore the database becomes saturated.

Firstly UForge uses Db sharding. This basically consists of splitting up the data into buckets (shards) that can be stored on more than one database instance. The second is caching (memcached) which is an in-memory key-value store for small chunks of arbitrary data from results of database calls. This reduces the amount of potential reads directly into the database.

To help scale out further you can also set up a database cluster, providing multiple database instances to the web service tier (the default database service does not support clustering).

Such bottlenecks can also be reduced by scaling-up (vertical partitioning) where more RAM and CPU is provided to the machine.

The generation cluster is intrinsically scalable, allowing the administrator to easily add new compute nodes to scale-out the number of simultaneous generations. OAR also provides the ability to deploy multiple schedulers and to configure them in master-slave mode.

1.6 Image Generations

Image generations are very I/O intensive compared to CPU, consequently only 1 core per simultaneous generation is required. Therefore for 10 simultaneous generations, a total of 10 cores are required for the compute notes.

The generation capacity of UForge depends on the total number of subscribers using the service. From experience, the ratio between generation capacity per week and the number of subscribers can be calculated as:

number of subscribers x 0.022 = generation capacity/week

This is assuming that UForge is used 24 hours a day. If the service is only used in one geography, then users will typically use the platform within an 8 hour period. Therefore the generation capacity will have to be multiplied by 3 (as the other 2/3 of the day, no generations will take place). For an 8 hour day, the generation capacity per week required for a certain number of subscribers can be calculated as:

number of subscribers x 0.066 = number of generations/week

On average, a generation takes about 5 minutes (this depends upon the size of the image being created and whether it requires to be compressed). For each core used to generate images, the total generation capacity per week can be calculated as:

12 x number of hours x 7 = generation capacity per core

For an 8 hour period:

12 x 8 x 7 = 672/week (per core)

For an 24 hour period:

12 x 24 x 7 = 2016/week (per core)

1.7 Estimating Scan Size

It is difficult to estimate sizing of scans. However, the following guidelines may be useful to plan your first scan:

average scan size x number of scans

Scan sizes vary from 100 KB to 12 GB

You should note that the data that will be scanned does not include operating systems known by UForge. For example, Debian or other OS data will not be included in the scan size.

Also, if you have already run a scan, only the delta information will be included.

1.8 Deployment Examples

How to organize your UForge configuration depends on the specific customer needs. This may include:

- The number of users as well as the number of simultaneous connections to the platform
- The number of simultaneous generations
- The number of projects in the project catalog
- The SLA of UForge
- Whether the service needs to be reached over multiple sites
- Whether UForge is exposed to 3rd party customers and partners, or for internal use only

Note that UForge can be deployed on physical machines or on a virtual or cloud platform. The word 'node' describes either the specification of a physical machine or a virtual machine instance running in the virtual or cloud environment.

1.8.1 Mono Node Proof of Concept

If you want to set up a mono-node POC (Proof of Concept) you should have the following:

- 10 CPU
- 16 GB memory

- 150 GB local hard drive (standard or SSD)
- 300 GB local hard drive (for the generation processes). SSD is preferred for performance reasons.
- 1 additional NAS or disk for user data (size will depend on user data, it will store the generated images and scans - 1 TB is typically sufficient for a POC)

1.8.2 Multi-node AppCenter

The minimum software topology is described in *Minimum Software Topology*. The following is an overall recommendation, assuming you are grouping the nodes.

Node	RAM	CPU	Hard Drive	Comment
DB/LDAP/OAR Secret Manager	10 GB	4	100 GB	
Web Service	6 GB	4	15 GB	
UI (portal)	3 GB	2	15 GB	
Compute Node (2 identical)	6 GB	4	15 GB + 500 GB SSD	Alternative is 1 TB SSD NAS Shared between the 2 compute nodes. This storage could be upgraded if needed
NAS distributions			500 GB	
NAS user data			7 TB	

Note: The size of the user data NAS depends on the expected size of the user data

1.9 Network Topology

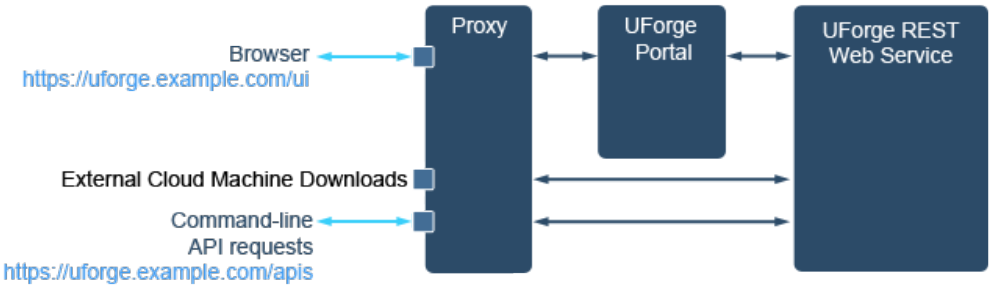
In order to optimize scalability, you should use a dedicated network subnet for UForge service.

If you want to deploy a public UForge service, you can use a public IP address for each component.

We recommend using a firewall with the NAT technology to map a public IP address to a private IP address. This allows you to use public IP addresses only for the Web Front-end and the Web Service. The other services stay hidden from the rest of the world.

For a private network, you can use a NAT translation, but it is better to use the firewall in gateway mode and filter the traffic. This way you can open ports on the different services.

By default UForge provides a proxy service, exposing the user interface (UI), REST web service tier and a download URL allowing external cloud platforms download generated machine images. This is configured automatically using the information you provide as part of the configuration phase (see *Configuring UForge*). These values can also be configured manually, for more information see *Modifying the UForge Platform External URL Endpoints*.



1.10 Minimum Software Topology

For the minimum software topology, use all services without security or fault tolerance.

For each of these components we recommend the following minimal (hardware or virtual) specification:

Component	RAM	CPU	Hard Drive	Comment
Web Front-End	3 GB	2	15 GB	
Web Service	3 GB	2	15 GB	
Generation Cluster (scheduler)	2 GB	1	15 GB	
Compute Node	4 GB	2	30 GB	With a good NAS/SAN you can reduce the disk space to 15 GB if you map the directory /space/REPOS with the NAS/SAN
SYNCOPE Webservice	2 GB	2	15 GB	
Database	4 GB	4	100 GB	With a good NAS/SAN you can reduce the disk space to 15 GB if you map the directory /space/REPOS with the NAS/SAN
LDAP	2 GB	2	15 GB	
NAS/SAN	2 GB	2		

It may be more logical to group the DB, LDAP and Generation Cluster (scheduler) on one node. Typically you can also group the Webservice and SYNCOPE Webservice. Refer to *Multi-node AppCenter*

Note: When a user creates an appliance, the packages are stored locally in the UForge cache repository, which is stored on the DB. Therefore, depending on the number of appliances created and OS used, you may need to adjust the DB size.

Note: If you are using high availability and you choose to split the Webservice and SYNCOPE on separate nodes, then you need a shared NAS/SAN (/tmp/userdata).

1.10.1 Compute Node Hard Drive

In order to ensure best performance when generating machine images, it is recommended to use a local SSD. The size of the disk depends on the number of parallel image generations and the size of images. For example, if 4 generations of 60 GB Windows images are expected to run in parallel, disk size should be at least 240 GB.

1.10.2 NAS/SAN

The NAS/SAN is a storage area which will typically contain UForge Repository data, such as:

- Operating system packages and updates
- Project catalog binaries
- User `My Software` binaries
- Generated images from users using the platform
- Data from user scans

1.11 Security Options

To increase the security of the UForge service, you can add replication in master/slave of the database component and the LDAP component.

The minimal requirement is 1 slave for the database and 1 slave for the LDAP service. Of course, you can add more slaves for each of these services.

See the specification of these options:

Component	RAM	CPU	Hard Drive	Comment
Database Slave	2 GB	1	70 GB	With a good NAS/SAN you can reduce the disk space to 15 GB if you map the directory /var/lib/mysql to the NAS/SAN
LDAP Slave	1 GB	1	15 GB	

1.12 High Availability

High Availability can be attained with the redundancy of all services. However, the database and the LDAP service have not yet been tested with the clustering mode.

For the moment we recommend using the "security options" for these services. This is a point of failure.

All other services can be clustered. For clustering a service you will need a pool of load balancers (physical or virtual).

Component	RAM	CPU	Hard Drive	Comment
Load Balancer	2 GB	2	15 GB	

All other services keep the same configuration, as seen above.

CHAPTER 2

Installing UForge

To install a fully functioning UForge platform, you must install and configure the UForge services as well as populate the UForge Repository with the operating systems you wish to build your server templates from.

This section covers:

2.1 UForge Installation Overview

To install a fully functioning UForge platform, we provide the following 2 ISOs:

- UForge Setup ISO – this includes all the necessary elements to have a functioning "UForge in a Box"
- ISO which contains the skeleton of the distribution installers in case you want to create ISO images with UForge. This only needs to be installed if you want to create ISO images.

2.1.1 Installation Checklist

Before you start deploying UForge, ensure that you have all the following:

- UForge Setup ISO
- Your activation credentials (ID and activation key) provided by your vendor
- Architecture of the deployment (number of nodes and networking topology. See *Network Topology*)
- The necessary system requirements (see *Storage Considerations*)
- Your SSL certificates, key and chaining certificates, and all files corresponding to the following entries in `/etc/httpd/conf.d/ssl.conf`:
  - SSLCertificateFile
  - SSLCertificateKeyFile
  - SSLCACertificateFile
  - SSLCertificateChainFile (might be empty)

Note: Ideally, UForge should be setup with your machine connected to internet, in order to be able to complete all the configuration steps, namely accessing:

- OS distribution repositories
- UForge Tools repository
- a YUM repository, in order to obtain all the UForge package when updating UForge.

However, UForge can be setup and operated without Internet. A proxy server can be setup in order to obtain updated UForge packages.

2.1.2 Installation Steps

UForge installation has three distinct phases:

Step 1: Install the UForge AppCenter for each node of the platform.

Step 2: Configure UForge AppCenter using the UForge Deployment Wizard.

Step 3: Additional configuration steps, as described in *Further AppCenter Configuration*.

2.2 UForge Repository Setup

The UForge Repository is a storage area containing:

- Operating system packages and updates
- Project catalog binaries
- User `My Software` binaries
- Generated images from users using the platform
- Data from user scans

The operating system and project binaries are separated from the `My Software` binaries and the images generated.

Warning: When new projects are populated by the administrator after the initial install, they are copied in the same location as `My Software` and generated images.

The UForge Setup Disk contains all the operating system information that needs to be copied to the UForge Repository. This repository can be shared storage or in the case where the entire UForge platform is being installed on one machine, can be on the local disk.

If you want to have the UForge Repository setup on a local disk, you must first install UForge prior to setting up UForge Repository. Otherwise, if this is shared storage, the UForge Repository can be setup independently.

The UForge Repository must be setup properly prior to completing the final configuration step. You must first populate the UForge database with the operating system package meta-data before you run the configuration phase described in *Configuring UForge*.

You can set up the UForge Repository either:

- on a shared storage. Refer to *UForge Repository on Shared Storage*. This option is mandatory if you are installing UForge as a multi-node environment.

- on a local storage. Refer to *UForge Repository on Local Storage*. This option can only be used if you are installing UForge as a single node environment.

2.3 UForge Repository on Shared Storage

NFS (Network File System) is used to share the information between the various UForge nodes. To setup the NAS or SAN for the UForge Repository you must create two shared directories, one for the operating system data and the other for all the user data (My Software and images generated).

Note: This option is mandatory if you are installing UForge as a multi-node environment.

To setup the shared storage:

1. Log in to the machine where the NFS server is running
2. Create the operating system directory, for example: /volume1/DISTROS

   The following NFS options are required:

   ```
   *(rw,async,no_wdelay,no_root_squash,insecure_locks,anonuid=0,anongid=0)
   ```

3. Create the user data directory, for example: /volume1/USER_DATA

   The following NFS options are required:

   ```
   *(rw,async,no_wdelay,no_root_squash,insecure_locks,anonuid=0,anongid=0)
   ```

Check the mount points on each UForge node:

1. Run the following commands:

   ```
   mount 192.20.777.205:/volume1/USER_DATA/ /mnt
   su - tomcat
   cd /mnt
   touch test
   ```

2. If it returns:

   ```
   touch: cannot touch 'test': Permission denied
   ```

   Then execute the following commands (as root)

   ```
   cd /mnt
   chown -R tomcat:tomcat .
   ```

3. Confirm you can create a file on /mnt as user tomcat. Then perform the following:

   ```
   cd ~
   umount /mnt
   ```

2.4 UForge Repository on Local Storage

..note:: You should only setup UForge repository on local storage if you are installing UForge as a single node environment.

When using local storage, UForge must already be installed, but not configured. Create two directories one for the operating system data and the other for all the user data (for example `My Software` and images generated).

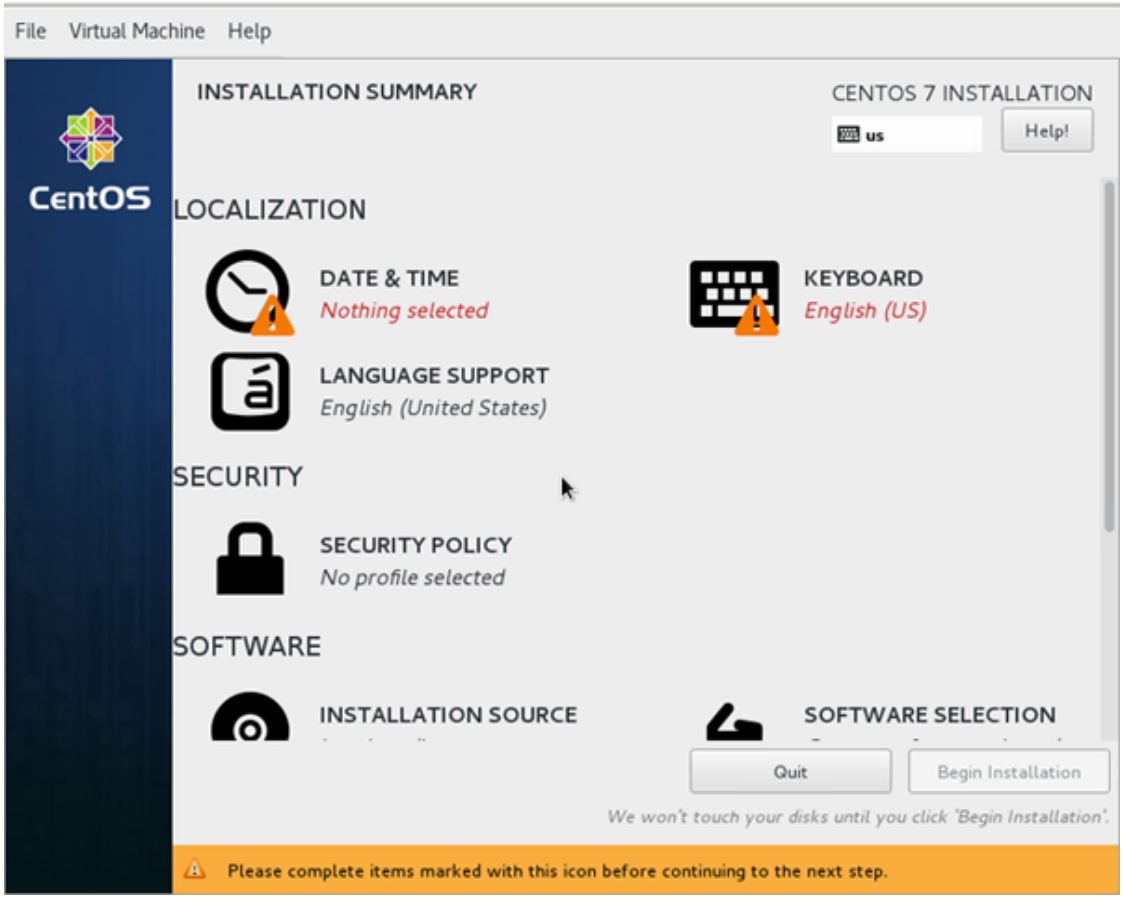
1. Create the operating system directory, for example: `/space/DISTROS`
2. Create the user data directory, for example: `/space/USER_DATA`

2.5 Installing from an ISO

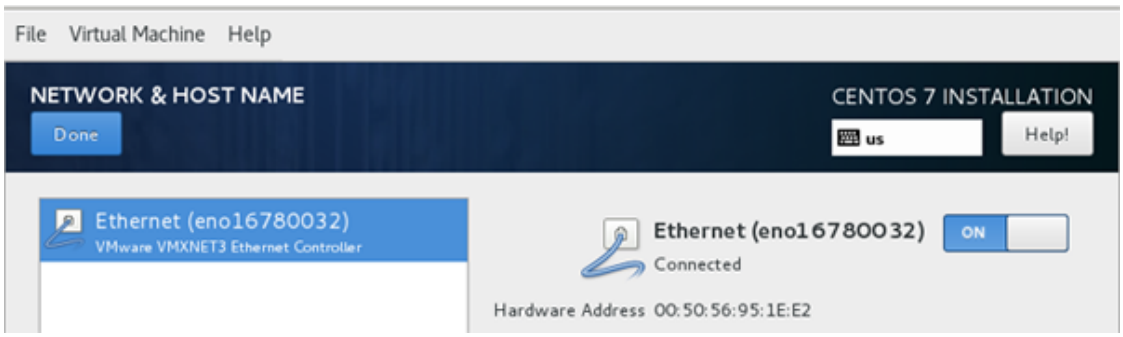
Note: The ISO install will be done on /dev/sda. The `kickstart` automates the installation of the operating system on the first disk, which must be managed by either a SCSI or SATA controller.

To install UForge from the ISO image:

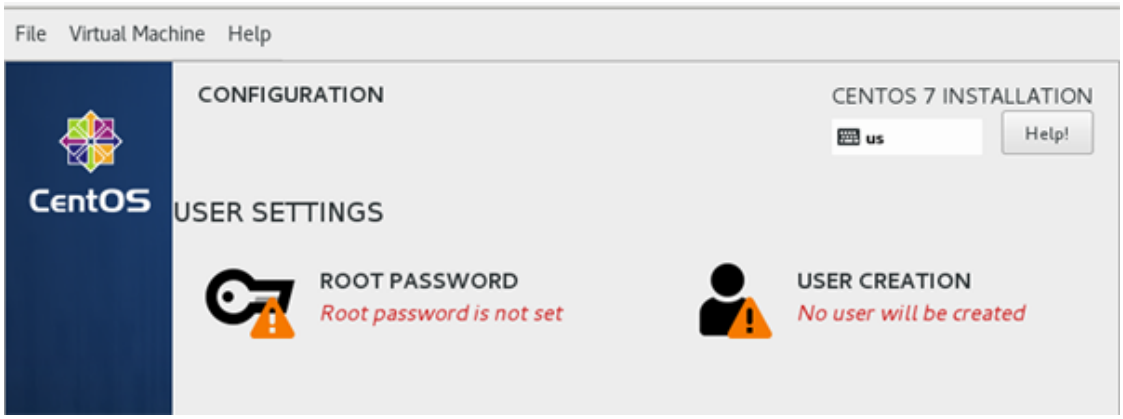
1. Attach the ISO to a VM or burn the ISO to a DVD (for installing to a physical machine, note the machine will require a DVD disk drive).
2. Boot the system from the ISO. The following main window will be displayed.



3. Click on `Date & Time`. Select a timezone and click `Done`.
4. Click on `Keyboard`. Choose the keyboard layout you want to use and click `Done`.
5. Click on `Network & Host name`. Using the top right toggle set it to `On`.



6. Click `Begin Installation` in the bottom right. The following window will be displayed.



7. Click `Root Password`. Set the root password and click `Done`.
8. Wait for the install to finish (this can take a while), then click `Reboot` to reboot the system
9. Once the system reboots, accept the EULA license agreement, select your network configuration (DHCP or manual), select the desired timezone. Then, click `Accept all and continue`.

For a multi-node installation, repeat these steps for every physical or VM instance you want to install. The installation phase is complete, you are now ready to configure the UForge AppCenter, see *Configuring UForge*.

2.6 Post-Installation Cleanup

To finalize the deployment phase, you should remove all the bits that helped to complete the deployment of UForge. To do so, run the following command on all the nodes composing the platform:

```
yum erase -y uforge-deploy uforge-studio-common-libs uforge-studio-runtime uforge-
↪studio-perl uforge-studio-java-jre uforge-studio-php uforge-studio-tcl uforge-
↪studio-lighttpd
```

2.7 Configuring UForge

Once the installation is complete on all the nodes you need for the UForge AppCenter, you are now ready to configure all the UForge AppCenter services. This is done via the UForge Deployment Wizard that guides you through the final steps of the installation process.

Note: The name of `LVM VolGroup Name` must be unique. The default when installing UForge will have a format similar to `vg_uss_150910-lv_uss_150910`. If your scanned instance has the same volume group name, or if you set up advanced partitioning with the same name, you will get an error when migrating.

2.7.1 Launching Deployment Wizard

To launch the UForge Deployment Wizard, use your browser and go to one of the nodes that have been installed:

http://<ip address of the node>:9998/deployments

Fill in the wizard, note that all the fields are mandatory.

1. Enter the Organization name. This is the name of the default Organization. The organization groups all the operating systems, formats and users for the platform.
2. Enter an email address. This is the email address of the root administrator of UForge. All administration email notifications are sent to this email address.
3. Set the root administrator's password. The password must be at least eight characters long, including at least one uppercase character, one lowercase character, and at least one digit. The following special characters are allowed: hyphen (-) and underscore (_). No other special characters are allowed, including spaces.
4. Provide a username, email and password for an initial user account to be created. This user account will have administration access rights for the default Organization.
5. Provide the UForge activation key and credential information to be able to receive UForge updates. If you do not have this information, please contact your vendor.
6. Select the operating systems and formats you would like to install. When you select an OS, a default mirror location is indicated. This is used for synchronization of packages and distributions. You can either accept the default values or modify the value to synchronize with another mirror.
7. Set the internet connection. By default, UForge expects to have a direct connection to the internet. If you de-select this option, you will need to enter the proxy hostname and port.
8. Set the SMTP Server to use for sending email notification messages created by UForge. If you want to use a SMTP relay, then also indicate the relay hostname and port number.
9. Click `next` to continue.
10. Enter the Web Server IP address, external hostname and the database IP address.

    The external hostname is used to construct two external URL endpoints. The first as the external URL endpoint of the user interface, and the other for REST API calls and command-line usage. The external hostname should normally be a fully qualified hostname. For example, if the external hostname is `uf.example.com`, then the following URL endpoints are created:

    - User interface URL endpoint: https://uf.example.com/uforge
    - Command-line URL endpoint: https://uf.example.com/apis

    These external URL endpoints can be changed after the initial configuration is complete, refer to *Modifying the UForge Platform External URL Endpoints* for more information.

    Note: In order to avoid potential errors due to length limits, it is recommended that the external host name of UForge be limited to 170 characters.

11. Select if OS and image storage should be local or remote. OS storage will be used for distributions, which image storage will include user data such as images, projects, mysoftware and other user data created with UForge.
    - If you choose to use a remote storage, indicate the NFS server with mount point and you should enter the full path for the OS directory e.g. `/DISTROS`.
    - If you select remote storage for the image store, you have to make sure that the path `USER_DATA` exists, with the correct permissions i.e. tomcat:tomcat. Refer to *UForge Repository on Shared Storage*, step 4, Check mount points, for more details.
12. Indicate the generation cluster compute node hostnames. You can add additional compute nodes by clicking the add button.
13. Once you have finished the configuration, click the `Deploy` button.

    Note: The deployment may take a few hours, depending on the number of operating systems you have chosen.

    Note: If the deployment starts but seems to hang (for example it remains in the `Starting` mode in the progress bar, or never goes beyond `Initializing`), there may be an error in the data entered into the wizard. In order to resume the deployment, log into the AppCenter server, and run the following command: `service oas-deploy restart`. You can then re-load the deployment URL in your browser, correct the data which was wrong and click `Deploy` again.

14. Once the deployment is complete, click `Finish`.

Warning: As part of the deployment phase, the wizard logs all the steps of the configuration. These logs include the administration passwords and other sensitive data you have used to configure this platform. It is important that these logs do not remain on the machine once the deployment is finished. To do this:

```
$ cd /var/log/UShareSoft
$ /bin/rm -rf oas-deploy
```

2.7.2 Configuring Ports

The following ports need to be configured for your UForge AppCenter.

For outgoing:

- 20 and 21
- 22 for SSH
- 443 and 80

For incoming:

- 80 and 443
- 22

In addition, communication ports between UForge and the cloud platform to which you will publish the images have to be open and depend entirely on your cloud platform configuration, see *Cloud Platform Default Ports* for more information. All other ports not in this list should be blocked by the firewall to protect internal services from outside access.

2.7.3 Configuring NTP

Some cloud platforms will reject uploading machine images, if the HTTP request date is in the future of the target cloud platform. To ensure proper function of UForge, please edit 'server' directives in `/etc/ntp.conf` if UForge servers cannot connect to NTP servers on the internet.

Note: If you want to contact NTP servers on the internet, then port 123 (UDP) should be opened on your firewall.

Note: If you have deployed UForge AppCenter and believe that ntp is not correctly setup, check if the log file `/var/log/UShareSoft/oas-deploy/logs/deployment/UForgeDeploy.pl/latestOASlog/<IP ADDRESS OF NODE>/NodeSetup.pl_4.log` contains the error: `[OAS ERRR] Unable to configure ntpd service`.

If this is the case, make sure that you configure ntp with an ntp server accessible from your environment (UForge uses pool.ntp.org by default). Assuming this ntp server has FQDN accessible.ntp.server, then run the command: `ntpdate accessible.ntp.server`

2.8 Cloud Platform Default Ports

To allow UForge to register machine images, you must ensure that there is network connectivity to the cloud platforms you want to push the machine images to. The table below provides the default port numbers of the cloud platforms supported, and indicates whether UForge uploads the machine image or requests the cloud platform to download.

Warning: For private cloud platforms, the port number may not be the default port number indicated.

Cloud Platform	Protocol	Default Port Number	Method of Registration
AWS	TCP	443	UPLOAD
Azure	TCP	443	UPLOAD
CloudStack	TCP	8080	DOWNLOAD
Google Compute Engine	TCP	443	UPLOAD
K5	TCP	443	UPLOAD
OpenStack	TCP	9292 and 5000	UPLOAD
Oracle Cloud	TCP	443	UPLOAD
vCloud Director	TCP	80 / 443	UPLOAD
vCenter	TCP	80 / 443	UPLOAD

2.9 Testing the Deployment

Once the configuration phase is complete, you should carry out some basic sanity tests to ensure that the UForge AppCenter is running normally:

Step 1: Check if the web service is operational

Use the values in the `uforge.conf` to contact the web service and expect a `200 OK` response.

Get the values from the `uforge.conf` and add them to some environment variables (you could also manually view the uforge.conf)

```
$ eval `grep '^UFORGE_WEBSVC_\|^UFORGE_GF_INTERNAL_IP\|^UFORGE_GF_HTTP_PORT\|^UFORGE_
↪GF_WEBSVC_ROOT_CONTEXT' /etc/UShareSoft/uforge/uforge.conf`
```

Run a simple http request (using basic authentication) using curl

```
$ curl http://$UFORGE_GF_INTERNAL_IP:$UFORGE_GF_HTTP_PORT/$UFORGE_GF_WEBSVC_ROOT_
↪CONTEXT/users/$UFORGE_WEBSVC_LOGIN -H "Authorization:Basic $UFORGE_WEBSVC_LOGIN:
↪$UFORGE_WEBSVC_PASSWORD"
--verbose

*   Trying 10.0.0.207...
* Connected to 10.0.0.207 (10.0.0.207) port 8080 (#0)
> GET /ufws/users/root HTTP/1.1
> Host: 10.0.0.207:8080
> User-Agent: curl/7.43.0
> Accept: */*
> Authorization:Basic root:welcome
>
< HTTP/1.1 200 OK
< Set-Cookie: JSESSIONID=77B407785AFC8F8FAE5120625F986F68; Path=/ufws/; HttpOnly
< Last-Modified: Wed, 30 Aug 2017 10:05:33 GMT
< ETag: "75cf28ed12f40a5c09b31bfe0dd2121d"
< Content-Language: en
< version: <UForge version>
< Content-Type: application/xml
< Transfer-Encoding: chunked
< Date: Wed, 30 Aug 2017 10:08:10 GMT
< Server: Apache
...<rest of the body removed>
```

Step 2: Check to see if the database is running

The database service should be running and available on the port `3306` and the database table is present. The Percona Server instance should have `usharedb` and `oar`

```
$ systemctl status mysql
MySQL running (22661)                                      [  OK  ]
```

Get the values from the `auth.conf` and add them to some environment variables (you could also manually view the auth.conf)

```
$ eval `grep '^UFORGE_DB' /etc/UShareSoft/auth.conf`
$ echo "show databases" | mysql -f -N -u $UFORGE_DB_ADMIN_LOGIN -p$UFORGE_DB_ADMIN_
↪PASSWORD -h db
information_schema
mysql
oar
performance_schema
usharedb
```

Step 3: Check the generation cluster resources

Check that all the cluster resources are available ('alive'). On each compute node

```
$ oarnodes | awk '/resource_id/ {n=$NF} /state : Suspected/ {printf "oarnodesetting -
↪s Alive -r %s\n",n}' | sh
```

This should return without any output.

CHAPTER 3

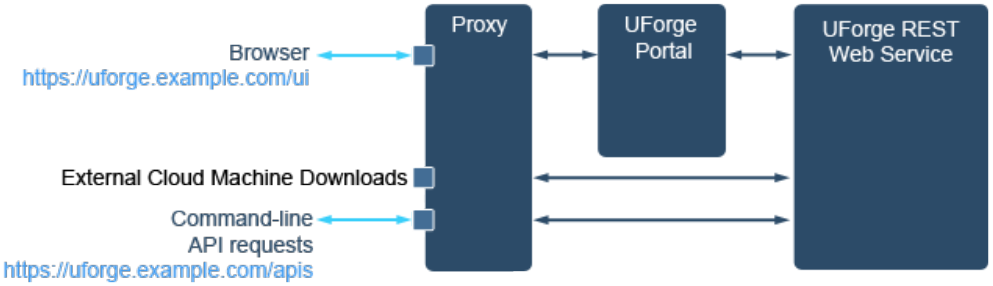
Further AppCenter Configuration

This section assumes that you have completed the installation of your UForge platform. Once it is installed and configured, you can:

3.1 Modifying the UForge Platform External URL Endpoints

There are three external URL endpoints for the UForge platform, namely:

- URL endpoint to access the UForge Portal (user interface)
- URL endpoint to access directly the REST web service for command-line tools and REST API calls
- URL endpoint for cloud platforms to download machine images from UForge. This URL endpoint is not used by end users, but only by cloud platforms that request to download machine images, rather than UForge uploading those machine images



These URL endpoints are automatically created based on the external hostname provided during the initial configuration of the UForge platform (see *Configuring UForge*). These URL endpoints can be changed by updating certain variables in the `/etc/UShareSoft/uforge/uforge.conf` file.

The UForge Portal URL endpoint is constructed using the following variables:

https://<UFORGE_PROXY_INFOS>/<UFORGE_UI_ROOT_CONTEXT>

The URL endpoint for direct REST web service access is constructed using the following variables:

https://<UFORGE_PROXY_INFOS>/<UFORGE_API_ROOT_CONTEXT>

The download URL endpoint is constructed using the `UFORGE_IAAS_DOWNLOAD_URL` variable.

If you want to use `http` rather than `https` (even though this is not recommended) then you must set the following variable in the `uforge.conf` file:

```
UFORGE_PROXY_USE_SSL = false
```

For example, if you set the following variables in `uforge.conf`:

```
UFORGE_PROXY_INFOS = hq.example.com:5666
UFORGE_UI_ROOT_CONTEXT = /ui
UFORGE_API_ROOT_CONTEXT = /apis
UFORGE_IAAS_DOWNLOAD_URL = http://hq.example.com:5777/downloads
UFORGE_PROXY_USE_SSL = true
```

The external URLs will appear as follows:

```
* UForge Portal: https://hq.example.com:5666/ui
* REST URL endpoint: https://hq.example.com:5666/apis
* Machine Image downloads (for external cloud platforms): http://hq.example.com:5777/
↪downloads
```

To update the external URLs:

1. Update the `/etc/UShareSoft/uforge/uforge.conf` file for each node with the updated variables you wish.
2. Launch the following two scripts (if multi-node the following order should be respected: compute notes, db nodes, web service, and UI nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

3.2 Adding a Compute Node

You can add a new OAR compute node which was instantiated from UForge but not configured as part of the initial deployment.

1. Make a snapshot of the UForge Server (to be able to come back to the state without the additional OAR compute node).
2. Initial setup: `oar-server` and `oarnode1` to `oarnodeN` already configured. Initialise two variables as follows:
   - OARNODEX=oarnodeX. Replace `X` in `oarnodeX` with the number of the new node to be added
   - IPoarnodeX=10.0.0.123. Replace `10.0.0.123` with the local IP address of the new node to be added

Note: The following commands are run on the first existing oarnode, for example oarnode1 until stated otherwise.

3. Copy over `/etc/hosts` from oarnode1 to oarnodeX with modification.
4. Get first local network interface (eth0).

```
ITF=`cat /proc/net/dev | grep : | cut -d ':' -f 1|grep -v lo | tr -d ' '|␣
↪head -1`
```

5. Get IP address associated with this interface.

```
IP=`/sbin/ip -o addr show $ITF 2>/dev/null| grep 'inet ' | awk '{print \$4}'␣
↪|
sed -e 's?/.*??'`
```

6. Modify and copy `/etc/hosts`

```
sed -e "s/\<oarnode1\>/$oarnodeX/g" /etc/hosts | awk -v ip=$IP -v␣
↪name=oarnode1 '{print}END{printf "%s %s\n",ip,name}' | ssh $IPoarnodeX dd␣
↪of=/etc/hosts
```

7. Copy over `uforge.conf` file and others.

```
rsync -a /etc/UShareSoft/uforge/uforge.conf $IPoarnodeX:/etc/UShareSoft/
↪uforge/
rsync /etc/oar/oar.conf $IPoarnodeX:/etc/oar/
rsync /etc/ssh/sshd_config $IPoarnodeX:/etc/ssh
rsync -a ~oar/.ssh $IPoarnodeX:~oar
```

8. Run the following commands on all oarnode1 .. oarnodeN and oar-server but **not** oarnodeX. This adds new node in all machines `/etc/hosts`.

```
awk -v newoar="$oarnodeX" -v newip="$IPoarnodeX" 'BEGIN{d=1}/\<{print␣
↪newoar\>/ {d=0}{print}END{if(d==1){print newip " " newoar}}' /etc/hosts > /
↪tmp/hosts.NEW
diff -q /tmp/hosts.NEW /etc/hosts >/dev/null
if [ $? -ne 0 ]; then
rsync -a /etc/hosts /etc/hosts.SVG-`date +"%Y-%m-%d"`
cp /tmp/hosts.NEW /etc/hosts
rm -f /tmp/hosts.NEW
fi
```

9. Edit `/etc/UShareSoft/uforge/uforge.conf` on all oarnode1 .. oarnodeN, oar-server and oarnodeX. Add the oarnodeX IP at the end of UFORGE_PROXY_IGNORED variable.

```
UFORGE_PROXY_IGNORED=X.X.X.X,Y.Y.Y.Y,...,<IPoarnodeX>
```

10. Run the following commands on oarnodeX

```
chmod 666 /etc/oar/oar.conf
/opt/UShareSoft/uforge/conf/oar_user_setup.sh
cp /opt/UShareSoft/uforge/tmpl/nfs.tmpl /etc/sysconfig/nfs
cp /opt/UShareSoft/uforge/tmpl/mountisos_init /etc/init.d/mountisos;␣
↪chkconfig
--add mountisos
```

(continues on next page)

(continued from previous page)

```
for s in ntpd tomcat mysql httpd oar-server openstack-glance-api
openstack-glance-registry oas oas-deploy; do service $s stop; chkconfig --
↪levels
0123456 $s off ; done >/dev/null 2>&1
ntpdate pool.ntp.org ; service ntpd start
for s in oar-node ntpd postfix mountisos; do chkconfig --levels 2345 $s on;␣
↪done
service ntpd stop ; ntpdate pool.ntp.org ; chkconfig ntpd on ; service ntpd
start
service mountisos start
service oar-node start
service sshd restart
/opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh -f >/dev/null 2>
↪&1
```

11. Run the following commands on oar-server to create new resources on oarnodeX from existing oarnode1 resources. In the following command, replace `oarnodeX` with the name of the new oar node.

```
/usr/bin/oarnodes | /bin/awk '/network_address=oarnode1/
{s=$0;gsub(".*nature=","",s);gsub(",.*","",s);printf "/usr/sbin/
↪oarnodesetting
-a -h oarnodeX -p cpuset=0,nature=%s\n",s}' | sh
```

You can also use a remote disk space of the compute node to generate multiple machine images in parallel by mounting the `/space` directory with a NAS or SAN.

3.3 Removing a Node

In order to remove a node, run the following command on the UForge server:

```
/usr/bin/oarnodes | /bin/awk "/resource_id/ {n=\$NF} /network_address=$
{REMOVENODE}/ {printf \"/usr/sbin/oarnodesetting -s Dead -r %s ; sleep 2;
/usr/sbin/oarremoveresource %s\n\",n,n}" | sh
```

3.4 Configuring Apache and Tomcat Web Services to use SSL Certificate

It is highly recommended that all communication with UForge is done via HTTPS. After the initial installation of UForge, neither the HTTP server (Apache) nor the application server (Tomcat) have yet been configured to use an SSL certificate and allow HTTPS.

To configure both servers to use an SSL certificate:

1. Log in as `root` to the machine running the UForge Apache and Tomcat Web Services.
2. Copy the SSL certificate files locally to the machine. Note that you should have three or four files, for example:
   - SSLCertificateFile: server.crt.pem
   - SSLCertificateKeyFile: server.key.pem
   - SSLCACertificateFile: CA.crt.pem

- SSLCertificateChainFile: intermediate.CA.crt.pem (this one is optional)

3. You need to build a self contained certificate as follows:

```
$ cat server.crt.pem CA.crt.pem > server_CA_chain.crt.pem
```

or:

```
$ cat server.crt.pem intermediate.CA.crt.pem CA.crt.pem > server_CA_
↪chain.crt.pem
```

Note that `.pem` files contain the following type of data for certificate files:

```
$ cat server.crt.pem
-----BEGIN CERTIFICATE-----
MIIHJTCCBg2gAwIBAgIDB25YMA0GCSqGSIb3DQEBBQUAMIGMMQswCQYDVQQGEwJJ
...
aW9uIEF1dGhvcml0eTADAgECGmRMaWFiaWxpdHkgYW5kIHdhcnJhbnRpZXMgYXJl
V4XfZvZtrRcZ
-----END CERTIFICATE-----
```

And the following data for the key file:

```
$ cat server.key.pem
-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDaRIAE7wrKbS9T
...
GdIr+qaNjk+eZLVsuPsAvwPlsWI/Cip7ZqygtvviteyeN0VZbLpRJgbbjXqh9GwP
G33VnWF89pfm5FNRu3WHIf8Ukw==
-----END PRIVATE KEY-----
```

If this is not the case, refer to the OpenSSL documentation on how to convert certificate and key files from one format to another.

4. Put the following entries in `/etc/httpd/conf.d/ssl.conf`:
   - SSLCertificateFile `/etc/pki/tls/certs/server.crt.pem`
   - SSLCertificateKeyFile `/etc/pki/tls/private/server.key.pem`
   - SSLCertificateChainFile `/etc/pki/tls/certs/intermediate.CA.crt.pem`
   - SSLCACertificateFile `/etc/pki/tls/certs/CA.crt.pem`
5. Verify the permissions and ownerships of these files:

```
$ ll -d /etc/pki/tls/certs/server.crt.pem /etc/pki/tls/private/
↪localhost.key /etc/pki/tls/private/ /etc/pki/tls/certs/
drwxr-xr-x. 2 root root 4096 Sep 25 12:05 /etc/pki/tls/certs/
-rw-------. 1 root root 1188 Sep 25 12:05 /etc/pki/tls/certs/
↪server.crt.pem
drwxr-xr-x. 2 root root 4096 Sep 25 12:05 /etc/pki/tls/private/
-rw-------. 1 root root  887 Sep 25 12:05 /etc/pki/tls/private/
↪server.key.pem
```

6. (Re)start the httpd server:

```
$ service httpd restart
```

If the server does not start, this may be because of a bad certificate, key or CA certificate file. In this case, check the appropriate logs in `/var/log/httpd`.

1. Verify the validity of the certificates:

```
$ openssl s_client -connect localhost:443
...
Verify return code: 0 (ok)
---
Ctrl-C or Ctrl-D to leave openssl client
```

If there is a problem with the certificate you might get outputs like:

```
$ openssl s_client -connect localhost:443
...
Verify return code: 18 (self signed certificate)
---
```

or

```
$ openssl s_client -connect localhost:443
...
Verify return code: 21 (unable to verify the first certificate)
---
```

2. Verify the certificate:

```
$ openssl s_client -showcerts -connect <ip-of-the-uforge-web-
↪service-machine>:<port>
```

Or you can use same openssl client command used for the Apache server in the previous step.

To verify that the new certificate is correct and if the Tomcat service is accessible from the outside, go to http://www.digicert.com/help/ and type the public name or IP address of your web service.

Note that there is no way to specify another port than HTTPS (443) on this page therefore you might need to add an iptables redirection rule like:

```
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j REDIRECT--to-
↪port 9191
```

3.5 Configuring SMTP Proxy

Once your UForge platform deployment is complete you can configure SMTP proxy. To configure SMTP:

1. In `/etc/UShareSoft/uforge/uforge.conf` you can modify:
   - UFORGE_RELAY_HOST=
   - UFORGE_RELAY_PORT=
   - UFORGE_RELAY_USER=
   - UFORGE_RELAY_PASSWORD=
   - If you want to use SSL set: UFORGE_RELAY_USE_SSL= 1 or 0

   These can be empty.
2. Run the following command on all the nodes (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

3.6 Configuring UForge Behind Enterprise Proxy

Once your UForge platform deployment is complete you can configure it behind your enterprise proxy if you have not already done so with the deployment wizard. You can configure UForge to use both an HTTP(S) and/or a SOCKS5 proxy.

Note: If you have a proxy and you want to register machine images to AWS, you must configure UForge to use a SOCKS5 server.

To configure UForge to use a proxy:

1. Open `/etc/UShareSoft/uforge/uforge.conf`, and then update the following attributes:

   For HTTP(S) modify:

   - EXTERNAL_PROXY_HOST =
   - EXTERNAL_PROXY_PORT =
   - EXTERNAL_PROXY_USER =
   - EXTERNAL_PROXY_PASSWORD =

   For SOCKS5 modify:

   - SOCKS5_HOST =
   - SOCKS5_PORT =
   - SOCKS5_LOGIN =
   - SOCKS5_PASSWORD =

   The user and password information is optional.

2. Run the following command on all the nodes (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

3.7 Configuring the Web Service

Each UForge Web Service instance runs inside a Tomcat application server. The web service has the following basic configuration information that is stored in a central configuration file: `uforge.conf`. The main configuration attributes for the web service are:

- External hostname used for incoming user connections
- External hostname for downloading images (this uses port `80` allowing all cloud platforms to be able to communicate with UForge for publishing images)
- Internal IP address used to connect with the other UForge services
- HTTP port (default: `8080`)

- HTTPS port (default: `8443`)
- Administration console port (default: `4848`)
- Administration console credential information (user and password)
- Root context of the web service (for example `/uforge`)

When installing UForge via the deployment wizard some of the configuration attributes can be decided by the administrator. The deployment wizard also creates the uforge.conf file with all the configuration information.

To view the uforge.conf file:

1. Log in to the web service node as root:

```
$ ssh root@<ip address of the node>
```

2. Open the uforge.conf file:

```
$ vi /etc/UShareSoft/uforge/uforge.conf
```

3. After making appropriate changes in these files, you should run the following command on all the nodes (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

For more information on Tomcat, see http://tomcat.apache.org

3.8 Configuring the Database

UForge uses the MariaDB database to store all the UForge meta-data and user information. The web service communicates with the database using hibernate. When installing UForge using the deployment wizard, one database instance is configured.

Note: By default no mechanism is configured to backup the contents of the UForge database. MariaDB can be configured as a cluster or in master-slave mode to provide reliability and to have a replicate of the data. Regular backups of the database should also be done.

The MariaDB database has the following basic configuration information that is stored in a central configuration file: `uforge.conf`. The main configuration attributes for the database are:

- Administration credential information (user and password)
- Address of the uforge database host.

When installing UForge via the deployment wizard some of the configuration attributes can be decided by the administrator. The deployment wizard also creates the `uforge.conf` file with all the configuration information.

If you decide to change database configuration information including the password, then you must also update the `auth.conf` and `uforge.conf` files with the correct information on all the nodes of the platform.

To view the `uforge.conf` or `auth.conf` files:

1. Log in to the web service node as root:

```
$ ssh root@<ip address of the node>
```

2. Open the uforge.conf file or auth.conf files:

```
$ vi /etc/UShareSoft/uforge/uforge.conf
$ vi /etc/UShareSoft/auth.conf
```

3. After making appropriate changes in these files, you should run the following command on all the nodes (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

For more information on MariaDB, see http://www.mariadb.com

3.9 Configuring the Scheduler

The scheduler can be configured to throttle the maximum number of:

- generations allowed in parallel per compute node.
- publishes allowed in parallel per compute node. A `publish` is when a user wishes to upload and register a generated image to a particular cloud environment.
- shares allowed in parallel per compute node. A `share` is when a user requests to share a template that they have in their private workspace.
- live system scans allowed in parallel per compute node. This is used for migrating live systems from one environment to another.

By default, after installing UForge using the deployment wizard, UForge will have X compute nodes. The scheduler is configured to allow 1 generation that has been chosen during the deployment on each compute node.

You can re-configure the scheduler to have different scheduling policies for different job types for each compute node. This is done be declaring a resource with a `nature` in OAR. Each `nature` corresponds to a specific job type. For example by declaring 4 resources with the nature ID `0` for compute node `node1` will configure the scheduler to allow up to 4 parallel generations. The following table shows the mapping between the different job types and the nature ID.

Job Type	Nature ID
Image Generation Job Type	0
Publish Image Job Type	1
Share Job Type	2
System Scan Job Type	3
Update cache repo (cron job)	4
Update_repos_pkgs.sh (Spider) (cron job)	5
Migration Job Type	6

By default UForge (during the deployment) populates oarnodesetting, and uses the "overflow" mode. If you want to modify this, refer to *Changing the OAR Configuration to Round-Robin*.

3.9.1 Viewing Current Resources

To view all the current resources in the scheduler, log in to the oar scheduler node as root and run the command `oarnodes`:

```
$ ssh root@<ip address of the node>
$ oarnodes
network_address : compute1.example.com
```

(continues on next page)

(continued from previous page)

```
resource_id : 1
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO, nature=0,␣
↪network_address=iso, type=default, cm_availability=0

network_address : compute1.example.com
resource_id : 4
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO, nature=1,␣
↪network_address=vm, type=default, cm_availability=0
```

This provides the basic information of each resource including:

- `network_address`: the compute node this resource is attached to
- `resource_id` is the ID of the node
- `state` is the current state of the resource
- `properties`: the main properties of the resource, including the nature ID that determines the job type this resource is providing

3.9.2 Adding or Updating a Resource

To add or update a resource, first log in to the oar scheduler node as `root`.

To add a resource to compute node `node1` and allow image generation (`nature=0`):

```
$ oarnodesetting -a -h node1 -p cpuset=0,nature=0;
$ oarnodes
network_address : node1
resource_id : 92
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO, nature=0,␣
↪network_address=vm, type=default, cm_availability=0
```

To change a current resource (resource_id: 92) to a different job type (for example publish images, which is nature ID 1):

```
$ oarnodesetting -h node1 -r 92 -p nature=1;
```

3.9.3 Removing a Resource

To remove a resource, first log in to the oar scheduler node as `root`.

To remove a resource from the compute node, run the following commands:

```
$ oarnodesetting -s Dead -r <resource_id>
$ oarremoveresource <resource_id>
```

3.9.4 Changing the OAR Configuration to Round-Robin

By default UForge (during the deployment) populates oarnodesetting using the "overflow" mode. This means that when a oar-node is full (in term of jobs), UForge overflows to the second node, and so on. This fits well with a typical topology where there is one powerful generation node and other smaller generation nodes (used in case of overflow).

This being said, it is not the role of UForge to replace the role of the scheduler and manage all the different topologies (Overflow mode, Alternative mode, custom mode, etc.). UForge integrates the oar scheduler which can be configured according to the user requirements.

So, by default, when a new OAR job is launched, it is processed on one OAR node until the number of simultaneous executions exceeds the set value, and the next OAR node is used when the number of simultaneous executions exceeds the set value.

If you want to have a round robin assignation of jobs, you can change the OAR configuration as follows.

1. In the oar-server node (and only on the node where oar-server runs, not on the other compute nodes), edit `/etc/oar/oar.conf`.
2. Comment the `SCHEDULER_RESOURCE_ORDER=[...]` line and add the new one with other parameters:

```
#SCHEDULER_RESOURCE_ORDER="scheduler_priority ASC, state_num ASC, available_
↪upto DESC, suspended_jobs ASC, network_address DESC, resource_id ASC"

SCHEDULER_RESOURCE_ORDER="resource_id ASC"
```

3. Restart the oar-server using the command: `service oar-server restart`. The following example orders the job scheduling by `resource_id` in ascending order. This will work only if your oarnodesetting are populated by nature and server.

```
network_address : oarnode1
resource_id : 1
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO,␣
↪available_upto=0, nature=0, network_address=oarnode1, last_available_
↪upto=0, type=default

network_address : oarnode2
resource_id : 2
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO,␣
↪available_upto=0, nature=0, network_address=oarnode2, last_available_
↪upto=0, type=default

network_address : oarnode1
resource_id : 3
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO,␣
↪available_upto=0, nature=0, network_address=oarnode1, last_available_
↪upto=0, type=default

network_address : oarnode2
resource_id : 4
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO,␣
↪available_upto=0, nature=0, network_address=oarnode2, last_available_
↪upto=0, type=default

network_address : oarnode1
resource_id : 234
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO,␣
↪available_upto=0, nature=1, network_address=oarnode1, last_available_
↪upto=0, type=default
```

(continues on next page)

(continued from previous page)

```
network_address : oarnode2
resource_id : 235
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO,␣
↪available_upto=0, nature=1, network_address=oarnode2, last_available_
↪upto=0, type=default
[...]
```

In this example, if you look the `nature=0` (in the properties). You have:

```
oarnode1 = id 1
oarnode2 = id 2
oarnode1 = id 3
oarnode2 = id 4
[...]
```

For `nature=1`, you have:

```
oarnode1 = id 234
oarnode2 = id 235
oarnode1 = id 236
[...]
```

In this case the first generation will be handled by the `oarnode1` and the second by the `oarnode2` (same for other types, publication is 1, scan is 3, etc.)

3.9.5 Deleting a Job

In case of a problem, you may want to delete a job which is stuck in a waiting state.

In this case, run:

```
$ oardel <job_id>
```

3.10 Managing the Watchdog Services

UForge has a set of watchdog services that carry out housekeeping tasks on a regular basis. These are:

- Cleanup Tickets: Task to remove generated images that are no longer attached to a parent image ticket in the database. This happens when the user "deletes" the generated image from their account. By default this is run once a day at 04:10AM.
- Update Distribution Packages: Task to regularly search the operating system repositories for any new updates and synchronize metadata into the UForge database. By default this is run every hour.
- Reset OAR Resources: Check the OAR scheduler resources and ensure that their state is "Active". By default this is run every 5 minutes.

Each of these housekeeping tasks are registered as a cron job in the first database node of the UForge platform to schedule the task to be run periodically. The frequency of these tasks can be changed.

To change the frequency of these housekeeping tasks, you need to update the crontab. Each line of a crontab file represents a job and is composed of a CRON expression, followed by a shell command to execute. The syntax is:

`dw month day hr min` followed by the command to be executed

Where:

- `dw` is the day of the week (0 - 6) (0 is Sunday, or use names)
- `month` is 1 - 12
- `day` is day of the month (1 - 31)
- `hr` is the hour (0 - 23)
- `min` is minutes (0 - 59)

To view these cron jobs, log in to the oar scheduler node as root and view the cron jobs:

```
$ crontab -l
*/5 * * * * /opt/UShareSoft/uforge/cron/reset_oar_resources.sh
10 2 * * * /opt/UShareSoft/uforge/cron/cleanup_userdata.sh
42 * * * * /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
05 * * * * /opt/UShareSoft/uforge/cron/update_repos_local_cache.sh
1 8 * * * /opt/UShareSoft/uforge/cron/drop_caches.sh
```

To update the crontab, log in to the oar scheduler node as root and edit the crontab

```
$ crontab -e
```

3.10.1 Cron Job Guidelines

There is no specific order to be respected when running cron jobs. These jobs are not inter-dependent.

`*/5 * * * * /opt/UShareSoft/uforge/cron/reset_oar_resources.sh`

This needs to be launched on a regular basis to avoid having issues with the OAR scheduler computation nodes. In fact, following network issue or other, nodes may move to state "Suspected". This job tries to fix that.

This job executes very quickly and does not take resources on the machine. It is set by default to 5 minutes but this can be changed.

`10 2 * * * /opt/UShareSoft/uforge/cron/cleanup_userdata.sh`

When a user deletes a machine image persisted on the NAS, only the metadata is removed from the database to avoid using a webservice thread to delete the file. This could take time and should be done asynchronously.

This job goes through the NAS and the database and checks which directories could be removed.

This job could potentially take a long time and be IO-intensive. It is highly recommended to execute it when there is not a lot of activity on the platform. We have set this at 2:10AM because there is not a lot of activity on our platform at that time of day.

This script could be launched several times in a day depending on the size of the infrastructure. For example, if the NAS is not big and if there are a lot of images created and deleted per day, it might be good to launch it several times a day.

`10 2 * * * /opt/UShareSoft/uforge/cron/cleanup_userdata.sh`

`42 * * * * /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh`

This mechanism launches UForge Spider to crawl the packages from the registered repositories.

It is important to keep these repositories up to date so that:

- when a user creates a new template, the latest package updates are listed

- when a user checks the appliance library, the number of updates available are listed
- the image generation is faster. If this is not done on a regular basis, when launching a generation, the repositories of the template distribution will be updated and the user will have to wait longer.

Also, if a repository the platform is connected to deletes packages (e.g. because of newer package version – this is not the case of UShareSoft official repositories), having the latest packages available is important.

```
05 * * * * /opt/UShareSoft/uforge/cron/update_repos_local_cache.sh
```

The UForge platform does not download all the packages from all the repositories listed. Instead, only the necessary packages are downloaded "on demand".

In the case of repositories that remove packages (because of newer packages), it is important to be able to reproduce a machine image with the same packages even though these packages no longer exist on the remote repository.

The local partial copy of the repository is registered in the platform as another repository. Each time new packages are downloaded for a repository, the local directory is marked "to be refreshed". When `update_repos_local_cache.sh` is executed, it checks all the local repo directories tagged as "to be refreshed" and executes the tool to update the native distribution repository (for Red Hat Enterprise Linux: createrepo with options).

It is important to execute regularly if the repositories use removed packages.

On extremely large platforms, it could take time and be IO-intensive.

If this command fails, usually it will only have an impact several days later (depending on the removing-package-repo policy with package removal). For example, if you generate a machine image with a pinned package version 1.2.3 on NTP and NTP is on a repository that removes packages. You generate a machine image and NTP gets downloaded. You generate the machine image again. No issue. Three days later, 1.2.4 version is released and 1.2.3 is removed from the remote repository. In that case, you will no longer be able to generate as the package in not in the remote repository, nor in the cache.

```
1 8 * * * /opt/UShareSoft/uforge/cron/drop_caches.sh
```

This calls native Linux commands to free up some memory on the platform.

If this commands fails it means the platform (not UForge but the machine itself) is in bad shape. It has no direct consequence on the UForge platform (only side effect is an issue with memory).

3.11 Tuning the Services

You can set the priority of tasks on the UForge platform in the file `services_conf.json`.

You can run COMMAND with an adjusted value for the nice value, which affects process scheduling. This allows you to fine tune the order in which services are treated by the platform and allows you to improve performances.

Note: If the event controller profile in `services_conf.json` is modified, you must restart the eventcontroller service in order for the modification to be taken into account.

3.11.1 Modifying a Configuration Profile

UForge includes a configuration file written in json that includes a set of profiles. These profiles contain a set of parameters to deal with resource priority (CPU, disk) and set different properties.

The configuration file is located at `/etc/UShareSoft/uforge/services_conf.json`

For each profile, you can modify the following parameters:

- `ionice`: set io priority of a process
- `nice`: set cpu priority of a process
- `sudo`: give sudo access or not to a process
- `jvm`: set java system properties and options

The following is a basic example of a configuration profile, named `profile1`. The parameters defined in `profile1` will be applied to the process specified by exec.

```
"profile1": {
  "exec": "/opt/UShareSoft/uforge/tool/demo.sh",
  "ionice": {
  "class": 2,
  "level": 7
},
"jvm": {
  "options": [
    "-Xmx2048m",
    "-Xms768m",
    "-XX:MetaspaceSize=256m",
    "-XX:MaxMetaspaceSize=1024m"
  ],
  "sys-properties": {
    "jna.library.path": "/opt/UShareSoft/uforge/lib"
  }
},
  "nice": {
    "niceness": 15
  }
}
```

3.11.2 Launching a Configuration Profile

To launch the process in a specific profile, run the script `runjob.py` located in /opt/UShareSoft/uforge/bin/

This sets all the properties you have defined in the profile and forwards all the arguments.

Command usage:

```
$ /opt/UShareSoft/uforge/bin/runjob.py profile_name arg1 arg2
```

where:

- `profile_name` is the name of the profile
- `arg1...argn` is the list of arguments to forward to the process this profile is to be applied to (defined in "exec" element)

3.11.3 Including an Override

You can include a profile that will override a process in a profile.

The following is an example of a profile that includes an override profile called `low_prio`

```
"my_process": {
  "exec": "/opt/UShareSoft/uforge/tools/demo.bin",
  "include": "low_prio",
  "sudo": true
  },
```

3.11.4 Tuning for Migration Projects

When migrating Linux-based systems, UForge scans the filesystem. In certain cases where the filesystem contains more than 300K files, you may need to increase the Java heap memory of some UForge services. The example below illustrates increasing the Java heap memory to 3GB (or more).

Note: On multi-node UForge platforms, this modification must be done on the compute nodes.

Note: If you modify the amount of memory dedicated to the Java Virtual Machine in the `webserver_uforge_services` section of the `services_conf.json` file, you must restart Tomcat webserver in order for the modification to be effective.

```
"create_vm": {
    "exec": "/opt/UShareSoft/uforge/tools/createimage/createvm.sh",
    "sudo": false,
    "ionice": {
        "class": 2,
        "level": 7
    },
    "jvm": {
        "options": [
            "-XX:MaxMetaspaceSize=512m",
            "-Xmx3096m",
            "-Xms512m"
        ],
        "sys-properties": {
            "jna.library.path": "/opt/UShareSoft/uforge/lib",
            "log.filename": "BuildImageEnv.log"
        },
        "proxy": true
    },
    "nice": {
        "niceness": 15
    }
},
"create_iso": {
    "exec": "/opt/UShareSoft/uforge/tools/createimage/createiso.sh",
    "sudo": false,
    "ionice": {
        "class": 2,
        "level": 7
    },
    "jvm": {
        "options": [
            "-XX:MaxMetaspaceSize=512m",
            "-Xmx3096m",
```

(continues on next page)

(continued from previous page)

```
            "-Xms512m"
        ],
        "sys-properties": {
            "jna.library.path": "/opt/UShareSoft/uforge/lib",
            "log.filename": "BuildImageEnv.log"
        },
        "proxy": true
    },
    "nice": {
        "niceness": 15
    }
},
"scan_vm_sudo": {
    "exec": "/opt/UShareSoft/uforge/tools/migratevm/migratevm.sh",
    "sudo": true,
    "ionice": {
        "class": 2,
        "level": 7
    },
    "jvm": {
        "options": [
            "-XX:MaxMetaspaceSize=512m",
            "-Xmx3096m",
            "-Xms512m"
        ],
        "sys-properties": {
            "jna.library.path": "/opt/UShareSoft/uforge/lib",
            "log.filename": "ScansEnv.log"
        },
        "proxy": true
    },
    "nice": {
        "niceness": 15
    }
},
"webserver_uforge_services": {
    "jvm": {
        "options": [
            "-Djava.awt.headless=true",
            "-Dfile.encoding=UTF-8",
            "-server",
            "-Xms500m",
            "-Xmx3096m",
            "-XX:NewSize=256m",
            "-XX:MaxNewSize=1024m",
            "-XX:MetaspaceSize=64m",
            "-XX:MaxMetaspaceSize=4096m"
        ],
        "sys-properties": {
            "jna.library.path": "/opt/UShareSoft/uforge/lib"
        },
        "proxy": true
    }
},
```

3.12 Using the Event Bus

UForge can be easily extended to interact and integrate with other products and services via an event bus service. Whenever a `POST`, `PUT` or `DELETE` request to UForge, a corresponding event is sent to the event bus (RabbitMQ). Custom plugins can be built to listen for these events and trigger custom business logic or call out to other 3rd party systems.

The event bus service is based on RabbitMQ. Custom plugins are known as "consumers".

Custom plugins (RabbitMQ consumers) can be written in many different languages including:

- Java
- Ruby
- C
- Python
- Erlang
- PHP
- Node.js etc

RabbitMQ is a message broker. In essence, it accepts messages from "producers" and delivers them to "consumers". In-between, it can route, buffer and persist messages, known as the "queue".

When a `POST`, `PUT` or `DELETE` request is sent to the web service, then UForge creates a message (the producer) and posts this to a routing service, called an exchange. This exchange is configured to publish messages to the UForge queue. When writing a custom plugin (consumer), it registers itself to this queue.

Administrators can also reconfigure the event bus routing service to adapt to more complex workflows. This is done in the event bus administration UI and use the root account of the UForge service, at:

```
http://youripaddress:15672
Login: uss-admin
Password: administration password
```

Refer to the RabbitMQ documentation for more information: https://www.rabbitmq.com/documentation.html

3.12.1 Writing a Custom Plugin (Consumer)

Prior to writing a custom plugin, it is important to understand the producer messages created by the UForge AppCenter. When a request is sent to the web service (`POST`, `PUT` or `DELETE`) the web service creates a producer message with the contents of the DTO (data transfer object) that is sent as a response to the request. The attributes for each DTO is described in the *APIs xsd file <apis:apis-index>*.

When creating a plugin, you will have access to all the attributes in a message. The plugin will contain the custom business logic required.

By default, this mechanism is used to send notification emails. For examples, refer to the RabbitMQ tutorials: https://www.rabbitmq.com/getstarted.html

3.12.2 Registering a Custom Plugin to Event Bus

To register a custom plugin, you must provide:

- The event bus URI, can be found in the UForge configuration file: `/etc/UShareSoft/uforge/uforge.conf`

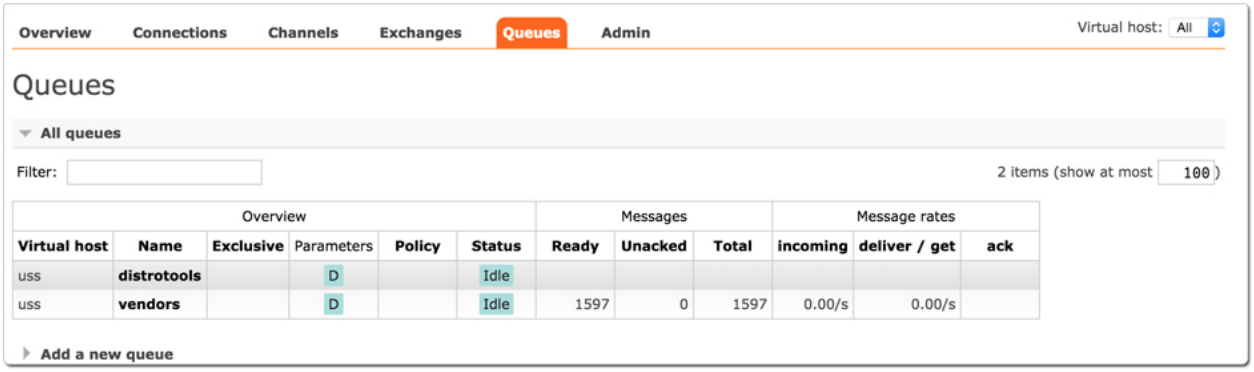
- The name of the queue

This information can be found in the RabbitMQ administration console.

1. Login to the administration console:

```
http://youripaddress:15672
Login: uss-admin
Password: administration password
```

2. Go to the Queues Tab, you will see the available queues. The default queue is `distrotools`.



3.13 Configuring Email Notification Service

The UForge AppCenter provides an email notification service using RabbitMQ to notify the super-user administrator and users via email for certain types of information.

Email notifications are triggered for the administrator when:

- There is a server template generation failure.
- A new user has been created on the platform (or in the case where this is manual, a request for a new user account to be created).
- A user has reported an abusive comment that requires moderation.

Email notifications are triggered to the end users when:

- A new user account has been created using their email address. This includes information on their credential information to access UForge.
- For all the templates the user is "following", emails are sent when the template is updated or a new comment has been added to the template.

3.13.1 Changing Email Address for Notifications

During the initial install of the UForge AppCenter, you must set the email address for the administrator. If you need to change the administrator's email address then you must update the uforge.conf file for each node that comprises the UForge AppCenter. UForge allows you to have a different email address for:

- new registrations (user account creations) to UForge
- all other administrator emails (generation failures etc)

You can also provide a no-reply email address when an email is sent.

To change the email address:

1. Log in to the node as root and edit the `uforge.conf` file:

```
$ vi /etc/UShareSoft/uforge/uforge.conf
```

2. Update the following variables:
   - `UFORGE_REGISTRATIONS_EMAIL`: to receive notifications on new user accounts being created
   - `UFORGE_POSTMASTER_EMAIL`: to receive all other email notifications (errors etc)
3. Run the script to force UForge to use the new `uforge.conf` file, this will restart certain UForge services (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

4. Update the `MAILFROM` and/or `MAILTO` variables in the crontab.

3.13.2 Customizing the Email Templates

You can modify or internationalize the information sent during an email notification. UForge provides a set of default templates for each email type sent by the system. The templates are stored in: `/opt/UShareSoft/uforge/tmpl`

Warning: When UForge is upgraded all the templates in the default directory will be overwritten. To ensure that any custom templates are restored during an update, a copy must be made of the custom template.

User account management emails:

- `ForgotPasswordEmail.tmpl` – email sent to the user when resetting their account password
- `SubscriptionAdminEmail.tmpl` – email sent to the administrator when a new user account has been created
- `SubscriptionUserEmail.tmpl` – email sent to the user requesting a new UForge account

Error emails:

- `ImageGenerationErrorMessage.tmpl` – email sent to administrator when an image generation failure occurs
- `WebServiceErrorsMessage.tmpl` (not used today)

The following keywords are reserved for substituting information into the emails:

- `#USER#` - The user name of the person carrying out the request
- `#EMAIL#` - The email of the user carrying out the request
- `#PASSWD#` - The password of the new user account created or during a "forgotten password" request
- `#WORKSPACE#` - The name of the workspace in the Collaboration Tab
- `#REFERRER_NAME#` - The name of the user who invited a guest to join a workspace
- `#MESSAGE#` - The content of the message the referrer sent to the guest

UForge uses the directory `/var/opt/UShareSoft` for custom files. If this directory exists, as part of the upgrade process and custom files will be restored.

Warning: When UForge is upgraded all the templates in the default directory will be overwritten. To ensure that any custom templates are restored during an update, a copy must be made of the custom template.

Therefore to change an email template:

1. Log in to the node as root then go to the template directory:

```
$ cd /opt/UShareSoft/uforge/tmpl
$ vi AppStoreNotificationNewComment.tmpl
```

2. Change the contents of the template and rename using the extension for the new language, if appropriate.
3. Copy the new template to all the other nodes of the UForge AppCenter.
4. Save a copy of the new template to protect against an upgrade overwriting the custom template:

```
$ mkdir -p /var/opt/UShareSoft/uforge/tmpl
$ cp /opt/UShareSoft/uforge/tmpl/AppStoreNotificationNewComment.tmpl /var/
↪opt/UShareSoft/uforge/tmpl
```

5. Instantiate the changes by running the following command:

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

3.14 Modifying the UForge IP

Note: This procedure is only valid for mono node UForge platforms. For an example of modify the IP in a multi-node environment, refer to *Modifying the UForge IP in a Multi-Node Environment*.

To modify the UForge IP you will need to:

1. Stop the following services:

```
service tomcat stop
service oar-server stop
service squid stop
service rabbitmq-server stop
service eventcontroller stop
```

2. Clean the squid cache as follows:

```
[root@db ~]# grep cache_dir /etc/squid/squid.conf
cache_dir ufs /data/proxy 10000 16 256
[root@db ~]# rm -rf /data/proxy/*
```

3. Modify all occurences of the old IP to the new IP in the following files:
   - `/etc/UShareSoft/uforge/uforge.conf`
   - `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/forge-config.xml`
   - `/etc/httpd/conf.d/uforge-ui.conf`

- /etc/squid/squid.conf

For example:

```
sed -i.bak "s/192\.168\.2\.177/192.168.2.200/g" /etc/UShareSoft/uforge/
↪uforge.conf /var/opt/UShareSoft/uforge-client/gwt/uforge/templates/forge-
↪config.xml /etc/httpd/conf.d/uforge-ui.conf /etc/squid/squid.conf
```

4. Modify all occurences of the old IP to the new IP in the following files:
   - /etc/sysconfig/network-scripts/ifcfg-ens160
   - /etc/hosts

For exemple:

```
sed -i "s/192\.168\.2\.177/192.168.2.200/g" /etc/sysconfig/network-scripts/
↪ifcfg-ens160 /etc/hosts
```

5. Restart the network service:

```
service network restart
```

Note: You will need to open a new terminal and connect to the machine using the new IP.

6. Restart the services as follows:

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
$ service oar-server start
```

Note: While oar-server is down, root user may receive emails with an error message about UForge cron execution.

3.14.1 Modifying the UForge IP in a Multi-Node Environment

The following is an example of modifying the UForge IP in a multi-node environment. This example should be modified to correspond to your platform configuration.

The environment in this example has four nodes:

- Node 1 for the UI
- Node 2 for the Web Service
- Node 3 for the DB with the OAR service Manager, RabbitMQ and UForge Event Controller
- Node 4 for the Compute Node

Old DB IP: 10.1.2.180

New DB IP: 10.1.2.25

1. Stop the following services:

   On the WS node and UI node run:

```
$ service tomcat stop
```

On the DB node run:

```
$ service oar-server stop
$ service squid stop
$ service rabbitmq-server stop
$ service eventcontroller stop
```

2. Clean the Squid cache as follows:

   Search and delete the cache directory in the DB node

```
$ grep cache_dir /etc/squid/squid.conf
 cache_dir ufs /data/proxy 10000 16 256
$ rm -rf /data/proxy/*
```

3. Modify all occurences of the old IP to the new IP in the following files:

   Modify `uforge.conf` and `squid.conf` modification in all nodes:

```
$ sed -i.bak "s/10.1.2.180/10.1.2.25/g" /etc/UShareSoft/uforge/uforge.conf
$ sed -i.bak "s/10.1.2.180/10.1.2.25/g" /etc/squid/squid.conf
```

   If the IP of the UI node or WS node was changed, the following files have to be modified on UI node:

   - /var/opt/UShareSoft/uforge-client/gwt/uforge/templates/forge-config.xml
   - /etc/httpd/conf.d/uforge-ui.conf

4. Modify all occurences of the old IP to the new IP in the following files:

   Modify `ifcfg-ens160` only on the node which IP will be changed

```
$ sed -i  "s/10.1.2.180/10.1.2.25/g" /etc/sysconfig/network-
↪scripts/ifcfg-ens192
```

   Modify `/etc/hosts` on all nodes

```
$ sed -i  "s/10.1.2.180/10.1.2.25/g" /etc/hosts for each node
```

5. Restart the network service only for the node where the IP has been changed:

```
$ service network restart
```

6. Restart the services as follows:

   (a) Execute `uforge_update.sh` script on all nodes in the next order:

   - Compute node
   - DB node
   - WS node
   - UI node

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

   (b) Start oar-server in the DB node:

```
$ service oar-server start
```

(c) If the IP of the UI node or WS node was changed, the next script has to be executed in the UI node:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

3.15 Customizing UForge Authentication for SSO

You need to build a custom AuthN/AuthZ module in order to perform Single Sign On to the UForge Platform.

The goal is to gain authenticated and authorized access to the RESTful UForge Webservice underlying methods. These methods are protected by a SecurityFilter and a SecurityContext using what is called a filter chain. The UForge Platform uses JSR 311(JAX-RS: The JavaTM API for RESTful Web Services), JSR 250 (Common Annotations for the JavaTM Platform) and Jersey (RESTful Web Services in Java).

To gain access through this filter chain to the UForge Webservice methods, you first need to be "Authenticated" (to prove who you are) and then be "Authorized", that is, be granted entitlements to perform or access certain methods of the UForge Webservice.

So, in order to connect to the UForge Webservice you first have to pass through the "authentication filter chain" of the web service. These filters are defined by default in web.xml

By default there are two filters:

- BasicAuthenticationFilter: the Basic method from RFC2617
- APIKeyAuthenticationFilter: the UForge internal APIKey authentication.

You should create an implementation of the interface IAuthenticationFilter returning an object implementing IUserAuthentication if authentication is ok, an exception if the authentication is invalid and null if your filter can not handle the request authentication.

The following is a basic example of an implementation of the RFC.

```
package com.usharesoft.authentication;
import com.sun.jersey.api.core.HttpRequestContext;
import com.sun.jersey.core.util.Base64;
import com.sun.jersey.spi.container.ContainerRequest;
import com.usharesoft.common.messages.User;
import com.usharesoft.db.DbAccess;
import com.usharesoft.db.DBException;
import com.usharesoft.services.IDMService;
import com.usharesoft.services.ServicesContext;
import com.usharesoft.services.exceptions.UForgeException;
import org.apache.log4j.Logger;
import org.hibernate.Criteria;
import org.hibernate.criterion.Restrictions;
import java.net.URI;
import java.util.List;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class BasicAuthenticationFilter implements IAuthenticationFilter {
    private static final Logger logger = Logger.getLogger(BasicAuthenticationFilter.
↪class);
    private static final Pattern CREDENTIALS = Pattern.compile("^basic\\s+(?.*)$",␣
↪Pattern.CASE_INSENSITIVE);
    private static final Pattern USER_PASS = Pattern.compile("^(?[^:]*):(?.*)$",␣
↪Pattern.CASE_INSENSITIVE);
```

(continues on next page)

(continued from previous page)

```
    @Override
    public IUserAuthentication filter(ContainerRequest containerRequest) throws␣
↪UForgeException {
        logger.trace("Getting Auth from Http Header");
        return getUserAuthentication(containerRequest);
    }
    protected IUserAuthentication getUserAuthentication(HttpRequestContext␣
↪containerRequest) throws UForgeException {
        /*
        * Get the authentication information from the HTTP header Return an
        * empty string if no authorization information is found
        */
        List authorizationList = containerRequest.getRequestHeaders().get(
↪"Authorization");
        if (authorizationList == null || authorizationList.isEmpty()) {
            return null;
        }
        /*
        * Get the first value as there should only be one value here as we will
        * only support Basic authentication for the moment Remove the Basic Tag
        */
        String credentials = authorizationList.get(0);
        logger.trace("Credentials: " + credentials);
        /*
        * Decode credentials
        */
        Matcher matcher = CREDENTIALS.matcher(credentials);
        if (!matcher.matches()) {
            logger.debug("Not matched Authorization header: " + credentials);
            return null;
        }
        /*
        * Consider that we match
        */
        /*
        * Decode base64 credentials
        */
        String base64basicCredentials = matcher.group("credential");
        if (!Base64.isBase64(base64basicCredentials)) {
            logger.warn("Invalid Base64 basic-credentials: " +␣
↪base64basicCredentials);
            throw new UForgeException(UForgeException.UNAUTHORIZED, "ERROR.
↪AUTHENTICATION.INVALID");
        }
        String basicCredentials = Base64.base64Decode(base64basicCredentials);
        /*
        * Decode user-pass
        */
        matcher = USER_PASS.matcher(basicCredentials);
        if (!matcher.matches()) {
            logger.warn("Invalid basic-credentials: " + basicCredentials);
            throw new UForgeException(UForgeException.UNAUTHORIZED, "ERROR.
↪AUTHENTICATION.INVALID");
        }
        /*
        * Compute fields
        */
```

(continues on next page)

(continued from previous page)

```
        String userId = matcher.group("userId");
        String userName;
        String targetUserName = null;
        if (UserAuthentication.isCompositeUserTargetUser(userId)) {
            userName = UserAuthentication.getCompositeUser(userId);
            targetUserName = UserAuthentication.getCompositeTargetUser(userId);
        } else {
            userName = userId;
        }
        String password = matcher.group("password");
        /*
        * Sanity checks
        */
        if (password == null) {
            logger.warn("Invalid password");
            throw new UForgeException(UForgeException.UNAUTHORIZED, "ERROR.
↪AUTHENTICATION.INVALID");
        }
        if (userName == null) {
            logger.warn("Invalid user");
            throw new UForgeException(UForgeException.UNAUTHORIZED, "ERROR.
↪AUTHENTICATION.INVALID");
        }
        return getUserAuthentication(containerRequest.getRequestUri(), userName,␣
↪password, targetUserName);
    }
    protected IUserAuthentication getUserAuthentication(URI requestUri, String␣
↪userName, String password, String targetUserName) throws UForgeException {
        /*
        * Check with IDM
        */
        ServicesContext.get().getService(IDMService.class).checkUserAuth(userName,␣
↪password);
        /*
        * Grab users
        */
        User user;
        User targetUser = null;
        DbAccess db = ServicesContext.get().getService(DbAccess.class);
        try {
            Criteria userCriteria = db.getDbManager().newCriteria(User.class);
            userCriteria.add(Restrictions.eq("loginName", userName));
            user = UserAuthentication.getUser(db, userCriteria);
            if (targetUserName != null) {
                Criteria targetUserCriteria = db.getDbManager().newCriteria(User.
↪class);
                targetUserCriteria.add(Restrictions.eq("loginName", targetUserName));
                targetUser = UserAuthentication.getUser(db, targetUserCriteria);
            }
        } catch (DBException e) {
            throw new UForgeException(UForgeException.DB_ERROR, e);
        }
        logger.debug("Basic Authentication is OK");
        return new UserAuthentication(user, targetUser);
    }
}
```

There are two ways to provide your authentication filter:

- modifying the webservice `web.xml`. You will need to modify the web.xml template by replacing the `com.usharesoft.authentication.AuthenticationFilters` value by your filter classname if you want only your authentication
- using the `@Provider` annotated class (the order can be important if the two authentication methods are used in the same request)

You can add a filter, but you cannot remove the default authentication filters, nor choose the order. To add your filter, use the following Jersey annotation:

```
package my.company.authentication

import javax.ws.rs.ext.Provider;
implements com.usharesoft.authentication.IAuthenticationFilter;

@Provider
class SuperAuthenticationFilter implements IAuthenticationFilter {
...
}
```

3.16 Allowing https Repositories with Self-Signed Certificate

Warning: The use of self-signed certificate can create security risks.

The following command allows you to add self-signed certificates (exposed by your repositories) as trusted into the UForge server.

1. Download the SSL self-signed certificate from the server (repository) you want to populate using the following command:

```
$ echo -n | openssl s_client -connect localhost:443 | sed -ne '/-BEGIN␣
↪CERTIFICATE-/,/-END CERTIFICATE-/p' > /tmp/cert.crt
```

2. Recreate the CA file on your UForge platform (containing all trusted CA):

```
$ cp /tmp/cert.crt /etc/pki/ca-trust/source/anchors/
$ update-ca-trust
$ update-ca-trust enable
```

3. Create a link to use the system java keystore instead of the jre default keystore:

```
$ ln -s /etc/pki/java/cacerts /usr/java/latest/jre/lib/security/cacerts
```

The default password is `changeit`.

3.17 Hosting Proprietary Packages

Proprietary packages, such as Red Hat Enterprise Linux or SUSE Linux are not delivered as part of the UForge repository. There are several options to use these packages with your UForge. You can:

- Create an enterprise repo

- Create a local mirror, as described in *Creating a Local Mirror for Red Hat Without Using a Satellite Server* and *Creating a Local Mirror for SUSE*.
- Install the OSes from the original ISO, as described in *Populating Red Hat Enterprise Linux Using ISO*.

3.17.1 Creating a Local Mirror for Red Hat Without Using a Satellite Server

You can create a local mirror of the latest updates for Red Hat Enterprise Linux without using a satellite server. How this is set up will depend on your version of Red Hat Enterprise Linux.

You can also refer to the Red Hat Knowledgebase solutions:

- Migrating from RHN to RHSM in Red Hat Enterprise Linux.
- How to register and subscribe a system to the Red Hat Customer Portal using Red Hat Subscription-Manager.

Red Hat 5 and 6

1. Install Red Hat Enterprise Linux 5.0 or 6.0.
2. Register the RH system running and register the system:

```
rhn_register command
```

3. Migrate your subscription tool, RHN to RHSM, execute the following two commands:

```
yum install subscription-manager-migration subscription-manager-migration-data
rhn-migrate-classic-to-rhsm
```

4. Install your web server:

```
yum install httpd
```

5. Create `/var/www/html/RHEL/6/X86_64/`.
6. Open port `80`.
7. Create a local mirror following the procedure described on the Red Hat Knowledgebase.
8. Add Red Hat Enterprise Linux to your AppCenter. For detailed instructions, refer to *Example for Adding Red Hat Enterprise Linux*.

Red Hat 7

1. Install Red Hat Enterprise Linux 7.0 as any other installation.
2. Register and subscribe your system to the Red Hat Customer Portal using Red Hat Subscription-Manager. Run the subscription manager with the RH Custom credential:

```
subscription-manager register –username –password –auto-attach
```

3. Install your web server:

```
yum install httpd
```

4. Create `/var/www/html/RHEL/6/X86_64/`.

5. Open port `80`.
6. Create a local mirror following the procedure described on the Red Hat Knowledgebase.
7. Add Red Hat Enterprise Linux to your AppCenter. For detailed instructions, refer to *Example for Adding Red Hat Enterprise Linux*.

3.17.2 Creating a Local Mirror for SUSE

To create a local mirror for SUSE, refer to official SUSE documentation. You must then declare the URL in UForge using `org repo create`. The method is similar to the one used for Red Hat Enterprise Linux *Example for Adding Red Hat Enterprise Linux*.

3.17.3 Populating Red Hat Enterprise Linux Using ISO

You must have the original ISO images of the operating system in question and follow the steps below.

Warning: If you populate UForge AppCenter with Red Hat Enterprise Linux using an ISO, only the package versions in the ISO image will be known to UForge AppCenter. If you later scan a Red Hat Enterprise Linux machine with package versions more recent than those of the ISO, the scan will succeed, but will be extremely inefficient since all packages will have to be rebuilt. Therefore, if you have access to a Red Hat Satellite, then adding the Red Hat Enterprise Linux repository exposed by the Red Hat Satellite is a better option.

To add a Red Hat repository using your ISO:

1. Mount the iso into `/mnt` (on the works node)
2. Create the appropriate directory layout under `/tmp/USER_DATA/repos/` for example: `/tmp/USER_DATA/repos/RHEL/6.5/x86_64/`
3. Copy all the content of the DVD into `/tmp/USER_DATA/repos/RHEL/6.5/x86_64/` using the `rsync -a` command.
4. If the repository does not already contain a repodata folder, you must create it inside the package directory:

```
$ cd /tmp/USER_DATA/repos/RHEL/6.5/x86_64/
$ createrepo .
```

5. Create a file in `/etc/httpd/conf.d` called `repos.conf`. The file should contain the following:

```
Alias /repos /tmp/USER_DATA/repos

<Directory /tmp/USER_DATA/repos>
    Options +Indexes
</Directory>
```

6. Run the following from the command line:

```
service httpd restart
```

7. You must now populate the Red Hat Enterprise Linux repository, as described in *Example for Adding Red Hat Enterprise Linux*.

3.18 Populating ISO Skeleton

If you want to create ISO images with UForge then you must copy the ISO skeleton files and mount them to your system, as follows.

Note: The procedure is the same for shared storage or local storage. The file location for shared storage is `/volume1/DISTROS` or the volume name as created when setting up the shared storage. For local storage it is `/space/DISTROS` or the directory name as created when setting up local storage.

1. Download the ISO that corresponds to the open source distributions you want. These are divided into two seperate ISOs:
   - CentOS, Debian, Ubuntu
   - Scientific Linux, Open SUSE, Oracle Linux

```
$ cd /space/DISTROS ; wget https://repository.usharesoft.com/downloads/
↪uforge-centos-debian-ubuntu-iso-installers-latest.iso
```

   or

```
$ cd /space/DISTROS ; wget https://repository.usharesoft.com/downloads/
↪uforge-opensuse-scientific-iso-installers-latest.iso
```

2. Mount the distribution by running:

```
$ mount -o loop /space/DISTROS/uforge-centos-debian-ubuntu-iso-installers-
↪latest.iso /mnt
```

   Note: The example above will mount Centos, Debian and Ubuntu, but the same could be done for other open source distributions by modifying the path.

3. Copy the content of the ISO:

```
$ rsync -a --progress /mnt/USS /space/DISTROS/
```

4. Umount the volume:

```
$ umount /mnt
```

3.19 Configuring Cloudsoft AMP

Once your UForge platform deployment is complete you can configure Cloudsoft AMP in order to allow users to deploy published machine images directly to Amazon or OpenStack, without having to connect to their cloud account. Before you start the configuration, ensure that you have:

- a running AMP server (version AMP-5.3 or Brooklyn 1.0.0-M1), see Cloudsoft AMP documentation for installation
- a configured user in AMP server that will be used by UForge to connect

3.19.1 Configuring AMP Server with Required Options

On the AMP server instance, run the following command as root:

```
$ grep -q "extraSshPublicKeyData=true" /etc/amp/system.properties ||␣
↪(echo "brooklyn.jclouds.authorizePublicKey.
↪extraSshPublicKeyData=true" >> /etc/amp/system.properties)
$ systemctl restart amp
```

Note: On a Brooklyn server, replace `amp` by `brooklyn` in the two commands.

3.19.2 Configuring AMP Access in UForge

1. On each UForge webservice node and the UI node, in `/etc/UShareSoft/uforge/uforge.conf` uncomment and complete the following lines:
   - AMP_API_ENDPOINT=<The URL to access AMP server eg: http://amp-instance.com:8081/v1>
   - AMP_API_USER=<The AMP user password that uforge should connect with>
   - AMP_API_PASSWORD=<The AMP user password>

Note: If you want to connect to your AMP server with HTTPS instead of HTTP, you can follow Cloudosft AMP documentation to configure it. In this case your AMP_API_ENDPOINT should use the secure URL eg: https://amp-instance.com:8443/v1.

2. Run the following command on the webservice node:

```
$ service tomcat restart
```

3. Run the following command on the UI node:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

3.19.3 Installing UForge Brooklyn Plugin into AMP Catalog

1. Run the following command on the UForge UI node:

```
$ /opt/UShareSoft/uforge/tools/update_scripts/deploy_brooklyn_plugin/
↪deploy_brooklyn_plugin.sh
```

2. If the script output contains `please restart the Brooklyn/AMP server`, then on the AMP server instance, run the following command as root:

```
$ systemctl restart amp
```

Note: On a Brooklyn server, replace `amp` by `brooklyn` in the command.

3.19.4 Configuring UForge Users Access Rights

At this point the deployment feature should be available for the root user. In order to allow other users to use this feature, give them the entitlement `deployments_access`. For more information on entitlements refer to *Creating and Updating User Roles*.

```
role create --name launcher --entitlement deployments_access
user role add --roles launcher --account someUser
```

3.20 Additional Configuration for Azure Stack

If Azure Stack is configured to use self-signed certificates, it is necessary to import Azure Stack's certificate into UForge server so it can be allowed access to Azure Stack.

To do this, follow the steps below:

1. On Azure Stack's host, open the Certificate Manager (certmgr.msc)
2. Go to `Trusted Root Certification`, select `AzureStackSelfSignedRootCert`, export the file as `DER encoded binary X.509 (.CER)` and copy it to the UForge server.
3. To import the certificate to the UForge server, execute the following commands:

```
$ keytool -import -alias AzureStackSelfSignedRootCert -keystore /
↪usr/java/jdk1.8.0_131/jre/lib/security/cacerts -file
↪<certificate_path>
Enter keystore password: changeit
Trust this certificate? [no]: yes
```

Note: Currently, UForge only supports the Azure Active Directory (Azure AD) authentication method.

3.21 Configuring Secret Manager

By default the secret manager is already configured and installed on the database node. UForge accesses Vault by using a token stored in a file and given in **uforge.conf**. This token should be able to read, write and create tokens on Vault.

Settings regarding the secret manager are stored in `/etc/UShareSoft/uforge/uforge.conf`:

- VAULT_PROTO=<The protocol to access Vault server eg: http or https>
- VAULT_HOST=<The URL to access Vault server eg: http://vault-instance.com>
- VAULT_PORT=<The port number used to communicate with Vault eg: 8200>
- VAULT_TOKEN_FILE_PATH=<The file containing the token that the UForge Server should use eg: /etc/UShareSoft/vault/root_token>
- VAULT_PREFIX=<The prefix used for UForge Server secrets. This will allow multiple UForge servers to use the same Vault server. Should be alpha-numeric without special characters eg: uforge1521559085>

Note: During installation the keys to unseal Vault and the root token are stored in separate files in `/etc/UShareSoft/vault/`. The root token is used to facilitate the token management; therefore you should not use this Vault instance for managing other secrets than UForge.

The Vault server must be unsealed and accessible in order to be used by UForge Server.

Vault is by default installed with UForge and the platform supports only this one.

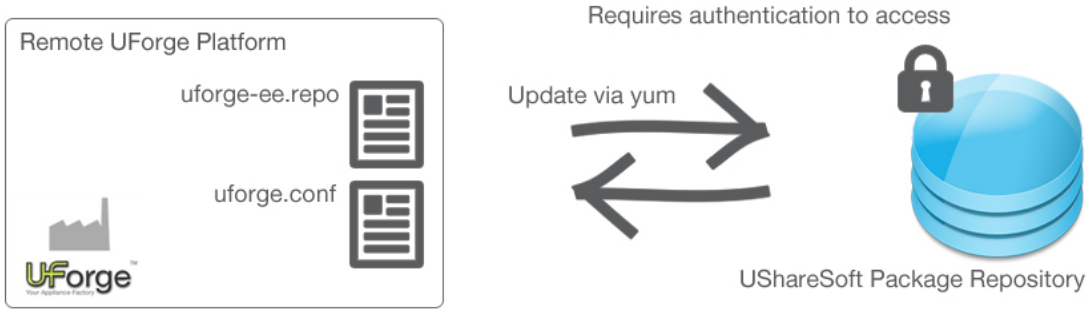
CHAPTER 4

Updating your UForge AppCenter

Once you have an installed and fully functional UForge AppCenter, you may wish to update it, or migrate it to a newer version. The following sections help you with the following:

4.1 Updating an Existing UForge Deployment

All the UForge components are delivered as native RPM packages. We maintain a custom repository of the UForge platform. All updates are added to this repository. The update mechanism of a remote UForge platform uses the standard "yum" command-line package management utility.



In order to interact with the official package repository, you must already have an active UForge license (stored in Fujitsu's database) and corresponding authentication credentials, set during the initial install of the platform. The authentication credentials are stored in the `uforge.conf` file and used in the `uforge-ee.repo` file. To view these files:

```
$ vi /etc/UShareSoft/uforge/uforge.conf
UFORGE_PRODUCT_ACCESS_USER=username
UFORGE_PRODUCT_ACCESS_PASSWORD=password
```

Warning: You should never update the following file: `/etc/yum.repos.d/uforge-ee-repo`

The request to update uses these credentials via HTTPS to Fujitsu who then determines whether you have the access rights to update the platform.

You must also have set up the yum repo file to authorize UForge updates using configure_yum_repos.sh as follows:

```
$ /opt/UShareSoft/uforge/conf/configure_yum_repos.sh -u <uss account user> -p <uss␣
↪account password> -t <uforge install type>
```

The (optional) parameter <uforge install type> can take the values `uforge-ee` or `uforge-dev`. By default `uforge-ee` is used.

Note: Before upgrading UForge, make sure that no cron jobs are running. For more information on cron jobs, refer to *Managing the Watchdog Services*.

Note: It is recommended to take a snapshot of all impacted VMs before starting the update procedure, in case you need to rollback.

To update the platform, use the "yum" command-line tool as follows:

1. Run `yum update uforge-common`.
2. Run `yum update uforge uforge-cli uforge-client uforge-gen`.

The RPM packages will be replaced and the services will be reconfigured to correctly update the platform. If you have a multi-node UForge platform, then this command must be run on all the nodes. The updates should be run in the following order:

1. database node
2. compute node(s)
3. web service and Portal nodes

3. Run the following CLI command in order to know if Squid is running:

```
$ systemctl status squid
```

If squid is stopped, run the following command-line

```
$ systemctl start squid
```

Note: If you are in a multi-node environment, there will be extra steps to do manually, depending on the version you are upgrading from.

- For UForge 3.8.FP3 and earlier versions, copy the file `/etc/UShareSoft/vault/root_token` from database node to web service node and launch `service tomcat restart` on the web service node. This file will be used to configure access to the secret manager. Then you will also need to follow the instructions for upgrading from 3.8.FP5.
- For UForge 3.8.FP5 and earlier versions run the following commands on each node using the shared storage to update NFS mount options:

```
$ sed -i "s@vers=3@vers=3,lookupcache=positive@g" /etc/init.d/mountisos
$ service mountisos restart
$ systemctl daemon-reload
```

4.2 Migrating UForge from 3.7 to 3.8

Before you start upgrading from UForge AppCenter version 3.7 to version 3.8, please note that the UForge root user password is now subject to restrictions. The UForge root user password must be 8 characters or more, contain at least 1 upper case letter, 1 lower case letter and 1 number. Characters - and _ are also allowed.

If your current UForge root user password matches those requirements, you do not need to make any changes. If your current UForge root user password does not match these requirements, you will have to change it in your UForge 3.7 AppCenter before starting migration.

A separate set of scripts and associated README.txt file is available on the UShareSoft repository.

1. Deploy UForge 3.8 with the same topology as the original UForge 3.7 one to migrate, using the CentOS 7 UForge in a box ISO
   - Deploy without any OS and format
   - Use the same organisation name (UFORGE_DEFAULT_ORG_NAME info) as the one defined in `/etc/UShareSoft/auth.conf on the <source_db>`
   - Use the same admin password (UFORGE_DB_ADMIN_PASSWORD info) as the one defined in `/etc/UShareSoft/auth.conf on the <source_db>`
2. Upgrade your UForge to the latest 3.8 patch, for each node of the platform (if multi-node the following order should be respected: compute notes, db nodes, web service and UI nodes).

   ```
   yum update -y uforge-common
   yum update -y uforge-gen uforge uforge-cli uforge-client
   ```

3. To upgrade from UForge 3.8 to a 3.8.fpx, you have to do the following on each node composing the topology. The following order should be respected: compute notes, db nodes, web service and UI nodes:
   (a) Edit `/etc/yum.repos.d/uforge-ee-centos.repo` and update `baseurl` info replacing `stable/3.8/official/centos/releases/$releasever/$basearch/` by `stable/latest/official/centos/releases/$releasever/$basearch/`.
   (b) Run the following command on each node:

   ```
   yum update -y uforge-common
   yum update -y uforge-gen uforge uforge-cli uforge-client
   ```

4. The source UForge DB server has to be upgraded to the last 3.7 version. On the <source_db> run the following commands:

   ```
   yum update -y uforge-common
   yum update -y uforge-gen uforge uforge-cli uforge-client
   ```

5. The source UForge Web Service server has to be upgraded to the last 3.7 version. On the <source_ws> run the following commands:

```
yum update -y uforge-common
yum update -y uforge-gen uforge uforge-cli uforge-client
service tomcat stop
```

6. Once both platforms are ready, on <source_db> run the following commands:

```
service oar-server stop
service OpenDJ  stop
cd /tmp/USER_DATA
source /etc/UShareSoft/auth.conf
mysqldump -u${UFORGE_DB_ADMIN_LOGIN} -p${UFORGE_DB_ADMIN_PASSWORD} --
↪routines --triggers --databases oar syncope uauthdb usharedb > db_
↪backup_3.7.sql
/opt/OpenDJ/bin/export-ldif -l uforge-export.ldif -n userRoot
rsync  -avl -H -F --progress /tmp/USER_DATA/ root@<target_db>:/tmp/USER_
↪DATA/
```

7. If you have Windows Golden images created manually in 3.7, run the following command:

```
rsync  -avl -H -F --progress /tmp/DISTROS/Windows/ root@<target_db>:/tmp/
↪DISTROS/Windows/
```

8. Once the previous rsync command has finished oar-server and opendj services have to be stopped. On <target_db > run the following commands:

```
systemctl stop oar-server
service opendj stop
```

9. Once oar-server and opendj are stoppen you need to stop Tomcat. On <target_ws> run the following command:

```
service tomcat stop
```

10. The DB on <target_db> server has to be reset. Tun the following commands:

```
source /etc/UShareSoft/auth.conf
echo "DROP DATABASE usharedb;" | mysql -u${UFORGE_DB_ADMIN_LOGIN} -p$
↪{UFORGE_DB_ADMIN_PASSWORD}
echo "DROP DATABASE oar;" | mysql -u${UFORGE_DB_ADMIN_LOGIN} -p${UFORGE_
↪DB_ADMIN_PASSWORD}
echo "DROP DATABASE syncope;" | mysql -u${UFORGE_DB_ADMIN_LOGIN} -p$
↪{UFORGE_DB_ADMIN_PASSWORD}
echo "DROP DATABASE uauthdb;" | mysql -u${UFORGE_DB_ADMIN_LOGIN} -p$
↪{UFORGE_DB_ADMIN_PASSWORD}
cd /tmp/USER_DATA
mysql -u${UFORGE_DB_ADMIN_LOGIN} -p${UFORGE_DB_ADMIN_PASSWORD} < db_
↪backup_3.7.sql
service mysql restart
oar-database --setup (answer y to the question)
/opt/opendj/bin/import-ldif -l uforge-export.ldif -n userRoot
systemctl start oar-server
service opendj start
if ( <target_db> != <target_ws> )
        /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh 2>&1␣
↪| tee -a /tmp/USER_DATA/uforge_update_db_3.8.log
fi
```

11. If the database node is not the same server as the webservice node, run the following command on the database node:

```
/opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh 2>&1 | tee -
↪a /tmp/USER_DATA/uforge_update_db_3.8.log
```

12. Once the DB is reset, Tomcat has to be started. On <target_ws> run the following commands:

```
service tomcat start
/opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh 2>&1 | tee -
↪a /tmp/USER_DATA/uforge_update_ws_3.8.log
rm /tmp/USER_DATA/db_backup_3.7.sql
rm /tmp/USER_DATA/uforge-export.ldif
```

13. On <target_ui> if you have customized the UI you will need to follow these additional steps before migration to 3.8. This step applies to all the customization files in `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates` (as decribed in), as well as `config.xml`, `forge-config.xml`, and css directory (in case of css customisation). Do the following on <target_ui>:

    (a) rsync the customised files from the UI node of the source UForge AppCenter to the UI node of the target AppCenter.

    (b) Edit file `forge-config.xml`, line

    ```
    <c:uForgeUrl>http://10.2.1.11:8080/ufws/</c:uForgeUrl>
    ```

    (c) Replace the old web service node IP address with the IP address of the new web service node.

    (d) With Apache2.2, you will need to add the aliases to the `/etc/httpd/conf.d/uforge.conf`, for example:

    ```
    Alias /publishimages/ "/tmp/USER_DATA/FactoryContainer/cloud-
    ↪downloads/"
    Alias /uforgestats/imgs/ "/tmp/USER_DATA/FactoryContainer/
    ↪images/"
    Alias /resources/uforge-scan.bin "/opt/UShareSoft/uforge/tools/
    ↪uforge-scan/uforge-scan"
    Alias /resources/uforge-scan.exe "/opt/UShareSoft/uforge/tools/
    ↪uforge-scan/uforge-scan.exe"
    Alias /resources/uforge-scan.zip "/opt/UShareSoft/uforge/tools/
    ↪uforge-scan/uforge-scan.zip"
    Alias /resources/uforge-sync.bin "/opt/UShareSoft/uforge/tools/
    ↪uforge-sync/uforge-sync"
    Alias /resources/rsync "/opt/UShareSoft/uforge/tools/uforge-
    ↪sync/rsync"
    Alias /resources/ipxe-init "/opt/UShareSoft/uforge/tools/pxe/
    ↪ipxe-init"
    ```

    With Apache 2.4, the above aliases also apply but you will have to add another Directory to the `/etc/httpd/conf/httpd.conf` to allow the users accessing the resources, for example:

    ```
    <Directory /opt/UShareSoft/uforge/tools/uforge-scan/ >
        Options Indexes FollowSymLinks MultiViews
        Require all granted
    </Directory>

    <Directory /opt/UShareSoft/uforge/tools/uforge-sync/ >
        Options Indexes FollowSymLinks MultiViews
        Require all granted
    </Directory>
    ```

(continues on next page)

(continued from previous page)

```
<Directory /opt/UShareSoft/uforge/tools/pxe/ >
    Options Indexes FollowSymLinks MultiViews
    Require all granted
</Directory>
```

(e) Launch the command

```
/opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

14. If you detect an issue, please contact support@usharesoft.com with the following files:
    - The standard output of the command
    - The result of running the `ifconfig` command
    - The `/etc/hosts` file
    - The name of the current node

4.3 Going Back to a Previous Version of a UForge Deployment

Note: Using `yum downgrade` to return to a previous version of UForge is not supported.

Even though the `yum downgrade` command works from a packaging perspective, it will not roll back possible changes done to the database (especially the database schema).

Moreover, there are additional factors that may lead to fatal errors, including configuration or properties files which are not considered by `yum downgrade`, or possible changes of users and their permissions on the file system.

If you need to downgrade a UForge deployment to a previous version, you should create a snapshot of your machine prior to the upgrade.

4.4 Retrieving Data from UForge

Before retrieving data from resellers.usharesoft.com using the lftp command from a UForge instance do the following:

1. Verify if the UForge instance is running in a virtualized infrastructure with security rules by default (AWS, OpenStack, ...). Ports 20, 21 (as well as 22 for SSH) must be allowed for outgoing traffic.
2. Due to the new proxy mechanism you must run:

```
export ftp_proxy=""
```

This ensures that squid does not interfere with ftp transfer.

4.5 Sending a Request to UForge

As the UForge Web Services are RESTful, clients communicate via the standard HTTP(S) protocol. That means you can easily construct request URLs that will work on the command line and in your code.

All UForge requests (with some exceptions) require authentication information as part of the request. There are several ways to communicate with UForge:

- Using API keys – A public and secret key is used to construct the URL. This URL will contain public key and a signature that authenticates the request.
- Basic Authentication – Where the login name and password are provided in the requesting HTTP(S) headers.
- Custom – UForge provides AuthN and AuthZ modules that can be customized to provide other authentication mechanisms (refer to Customizing UForge Authentication for SSO).

All request URLs start with the hostname of where UForge is running, the port where UForge is listening for incoming requests, the service name and version number. This is known as the BASE URL.

Even though UForge accepts HTTP requests, it is highly recommended for security reasons that HTTPS requests be used. HTTP requests should only be used for debugging purposes. Sensitive information will be exposed using HTTP.

UForge expects certain headers containing authentication information to be present as part of the URL request. UForge also accepts other header information, for example, to specify response content type and caching.

The following is an example of a request sent to a UForge AppCenter with hostname 10.0.0.20 using cURL to get the user `myUser`. Note that basic authentication is used for clarity.

```
$ curl "http://10.0.0.20:9090/ufws/users/myUser" -H "Authorization: Basic␣
↪myUser:password" -H "Accept: application/xml" -v | tidy -xml -indent -quiet

        * About to connect() to 10.0.0.20 port 8080 (#0)
        * Trying 10.0.0.20... connected
        * Connected to 10.0.0.20 (10.0.0.20) port 8080 (#0)
        > GET /ufws/users/myUser HTTP/1.1
        > User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/
↪0.9.8r zlib/1.2.3
        > Host: 10.0.0.20:8080
        > Accept: application/xml
        >

        < HTTP/1.1 200 OK
        < X-Powered-By: Servlet/2.5
        < Server: Sun GlassFish Enterprise Server v3.1.2
        < Last-Modified: Thu, 08 Aug 2013 19:52:13 GMT
        < ETag: "80f76a81b033572861260548dd748bb3"
        < Content-Type: application/xml
        < Transfer-Encoding: chunked
        < Date: Thu, 21 Jul 2011 17:02:10 GMT
        <

        * Closing connection #0
        <?xml version="1.0" encoding="utf-8" standalone="yes"?>
        <user>
        ...omitted for clarity
        </user>
```

CHAPTER 5

Backup Overview Guidelines

Warning: Fujitsu is not responsible for any customer data loss. The database backup techniques highlighted in this document are standard best practices used by the industry.

The goal of this section is to highlight some best practices on how to correctly backup the UForge data. Before you begin you might want to read:

5.1 Backup Overview Guidelines

Warning: Fujitsu is not responsible for any customer data loss. The database backup techniques highlighted in this document are standard best practices used by the industry.

There are two zones of UForge data that should be backed up:

- UForge database service.
- UForge user data (user uploaded software etc)

Database replication and backup are sometimes confused as achieving the same goal. Many people believe that by having replication, the data within the database is being stored elsewhere and therefore in the event of a serious problem, the data can be recovered. However, with replication, any accidental changes or deletions are also replicated to the other systems.

Database replication is meant for high-availability of your service, in other words building a redundant database service. If a problem occurs with one database instance, then having a second database with the data replicated, this instance can continue providing the database service (even if the service is degraded).

Database backup is used to take a snapshot of the database data (either a complete or incremental snapshot) and storing this information in an archive.

When thinking about your database strategy, you should consider the following goals:

- The backup should be an internally consistent snapshot (this may seem obvious, but if you do not lock the database or turn the database off before doing the snapshot then the data may be inconsistent).
- Keep the service running during the backup (in production systems this will be mandatory, however for less-critical platforms this may not be a necessary goal)
- Make the backup automated (via cron or using some 3rd party tools)

5.2 Backup Recommendations

> **Warning:** Fujitsu is not responsible for any customer data loss. The database backup techniques highlighted in this document are standard best practices used by the industry.

Here are some general recommendations and guidelines when doing database backup.

Never Back Up Databases to Local Disk

If the database server crashes, especially due to a hardware problem or a severe OS problem, the local drives may not be available. In the event of a hardware disaster, if the archives and current copy is on the shared filesystem allowing someone to quickly build a new server and start doing restores while resuscitating the server.

Back up Databases to a Fileshare, then Back the Share to Tape

Tape drives these days are fast enough that the vendors like to say DBAs should go straight to tape, and they are technically right: tape backup & restore speed is not a bottleneck. However, there is a limited number of drives available. This, however, gives you an extra layer of backup protection in case something goes wrong with the shared filesystem.

Back Up to a Different NAS/SAN

This is not possible for everyone, however, in extremely rare occasions where the production NAS/SAN does go down then at least the backup array was not affected as it is on a different NAS/SAN than the production system.

Consider Raid 10 SATA

Depending on the backup windows, multiple database servers may be writing backups to that same share simultaneously. Raid 10 gives better write performance than raid 5, (using fiber channel backup drives is even better but cost-prohibitive). Raid 10 on SATA gets a good balance of cost versus backup throughput.

5.3 Using Master-Slave Replication for Database Backup

To ensure consistent backups of the data and to limit the impact of the actual backup process on the service performance, a common approach is to use Master-Slave replication. This is an easy way to provide a complete running copy of the data on another database instance (called the slave). Consequently you can then stop the slave instance to do the backups while leaving the other instance (the master) to continue providing the database service. Master-slave replication is also the first implementation of high-availability.

The configuration of master-slave replication is out of scope of this document, but we recommend the following reading:

- How to Set Up Replication, MySQL documentation
- High Availability MySQL Cookbook
- MySQL High Availability by O'Reilly

You can now safely run the same commands provided in section *UForge Databases Basic Backup* on the slave database instance.

To automate the entire process, a script can be written and executed by a cron job at the frequency you wish to back up the database (nightly backups are recommended).

If you want to add high-availability, then we recommend using MySQL Cluster. This is a separate MySQL product that is not distributed with UForge.

The following zones of UForge data should be backed up:

5.4 UForge Databases Basic Backup

UForge has four databases to store all its data, namely:

- LDAP (OpenDJ) for storing user credential information
- IDM (Syncope using MariaDB) for storing user entitlements and roles
- Secret Manager (Vault) for storing user's secrets.
- UForge (MariaDB) for storing everything else for UForge features.

MariaDB provides many different types of backup methods you can choose from, including:

- Doing a hot backup (for example with Percona XtraBackup)
- Copying Table files
- Delimited-Text backups
- Using the mysqldump or mysqlhotcopy commands
- Creating incremental backups by enabling the binary log
- Creating a backup from a slave
- Making backups using a file system snapshot

For more details on all these back up strategies, refer to: http://dev.mysql.com/doc/refman/5.5/en/backup-methods.html

MariaDB holds both UForge and IDM data. All this information is located under:

```
/var/lib/mysql
```

By copying this information, you are taking a snapshot of the entire database.

The secret manager also saves secrets on the filesystem at `/var/lib/vault` and this folder must also be backed up on a regular basis. This should be performed during a time frame where there is no user activity to ensure Vault data integrity.

Note: Vault and mysql must be backup synchronously to keep data integrity.

LDAP (OpenDJ) has its own tool for doing backup, called `backup`. To ensure data consistency across the entire platform, we recommend you backup all the databases at the same time. Here is an example of how to back up the MariaDB (containing IDM and UForge information) and LDAP databases. Note, we are also stopping the web service instances; this ensures we do not generate connection error messages in the logs:

```
$ systemctl stop tomcat
$ systemctl stop mysql
$ systemctl stop vault
$ f=$(date +%y%m%d)
$ l="$f-ldap-backup.tgz" #what to name ldap backups
$ m="$f-mysql-backup.tgz" #what to name mysql backups
$ v="$f-vault-backup.tgz" #what to name vault backups
$ tar -czf /tmp/$m /var/lib/mysql/*
$ tar -czf /tmp/$v /var/lib/vault/*
$ /opt/opendj/bin/backup -a -c -d /tmp/ldap_backup
$ tar -czf /tmp/$l /tmp/ldap_backup
$ systemctl start vault
$ systemctl start mysql
$ systemctl start tomcat
```

If using SAN mountpoint, add:

```
$ rsync /tmp/$m /<SAN-MOUNTPOINT-BACKUP-DESTINATION>
$ rsync /tmp/$l /<SAN-MOUNTPOINT-BACKUP-DESTINATION>
$ rsync /tmp/$v /<SAN-MOUNTPOINT-BACKUP-DESTINATION>
```

If using remote backup, add:

```
$ rsync /tmp/$m root@<BACKUP-DESTINATION>
$ rsync /tmp/$l root@<BACKUP-DESTINATION>
$ rsync /tmp/$v root@<BACKUP-DESTINATION>
```

Warning: This only works if the database instance is stopped to ensure a consistent dump of the database. Otherwise you get a corrupt, inconsistent backup.

The issue with this method is if you have only one database instance, then you are effectively stopping your service to do the backup. To overcome this you need to use replication, for example master-slave.

Note: For LDAP, the service does not need to be stopped.

5.5 Basic Restore

Restoring the database is a simple copy using rsync back to the database directory. Note the use of slashes (/) is important. You should stop the web service to avoid creating connection error messages in the logs.

```
$ systemctl stop tomcat
$ systemctl stop opendj
$ systemctl stop mysql
$ systemctl stop vault
$ cd /tmp
$ tar -xvf /tmp/<ldap tarball>
$ /opt/opendj/bin/restore -d /tmp/<ldap dir>/userRoot
$ /opt/opendj/bin/restore -d /tmp/<ldap dir>/tasks
$ /opt/opendj/bin/restore -d /tmp/<ldap dir>/config
$ /opt/opendj/bin/restore -d /tmp/<ldap dir>/schema
$ tar -xvf /tmp/<mysql tarball>
```

(continues on next page)

(continued from previous page)

```
$ tar -xvf /tmp/<vault tarball>
$ rsync -a --delete-before /tmp/<mysql dir>/* /var/lib/mysql/
$ rsync -a --delete-before /tmp/<vault dir>/* /var/lib/vault/
$ systemctl start opendj
$ systemctl start vault
$ systemctl start mysql
$ systemctl start tomcat
```

To restore the SAN mountpoint:

```
$ rsync /<SAN-MOUNTPOINT-BACKUP-DESTINATION>/<ldap tarball> /tmp
$ rsync /<SAN-MOUNTPOINT-BACKUP-DESTINATION>/<mysql tarball> /tmp
$ rsync /<SAN-MOUNTPOINT-BACKUP-DESTINATION>/<vault tarball> /tmp
```

To restore remote backup:

```
$ rsync root@<BACKUP-DESTINATION>/<ldap tarball> /tmp
$ rsync root@<BACKUP-DESTINATION>/<mysql tarball> /tmp
$ rsync root@<BACKUP-DESTINATION>/<vault tarball> /tmp
```

5.6 User Data Backup

UForge also stores information that is uploaded by the users of the platform. These are:

- packages, binaries and files that are uploaded in `My Software` or `Projects`
- images (photos, appliance logos etc)
- boot scripts
- licenses (attached to projects or My Software)

All this information is stored in the following directory: `/tmp/USER_DATA`

Like the database, this is important information that must be backed up on a regular basis. The same mechanism can be used for back up as the database, namely using rsync as highlighted in *UForge Databases Basic Backup*.

You do not need to stop the service, but the danger is that if during that time there is damaged data, when rsync is run then this corrupted data is copied and there is no possible way to recuperate the old data.

The time it will take to back up the data will depend on the size of your data and the connection between the two servers.

CHAPTER 6

Managing Services

The following sections cover information regarding managing the UForge services.

6.1 Starting and Stopping the Application Server

Tomcat is registered as an operating system service. Each time a web service node is rebooted, the application server is restarted automatically as part of the node boot sequence. To manually stop or start the web service you must be the unix root user of the node.

6.1.1 Stopping the web service

```
$ service tomcat stop
```

6.1.2 Starting the web service

```
$ service tomcat start
```

6.1.3 Restarting the web service

```
$ service tomcat restart
```

6.2 Starting and Stopping the Database

The database is registered as an operating system service. Each time a database node is rebooted, MariaDB is restarted automatically as part of the node boot sequence. To manually stop or start the database you must be the unix root user of the node. Log in to the database node as root:

```
$ ssh root@<ip address of the node>
```

6.2.1 Stopping the Database

```
$ systemctl stop mysql
```

6.2.2 Starting the Database

```
$ systemctl start mysql
```

6.2.3 Restarting the Database

```
$ systemctl restart mysql
```

6.3 Starting and Stopping the Secret Manager

The Secret Manager is registered as an operating system service. Each time the secret manager node is rebooted, the service is restarted automatically as part of the node boot sequence. To manually stop or start the secret manager you must be the unix root user of the node. Log in to the secret manager node as root:

```
$ ssh root@<ip address of the node>
```

6.3.1 Stopping the Secret Manager

```
$ service vault stop
```

6.3.2 Starting the Secret Manager

```
$ service vault start
```

6.3.3 Restarting the Secret Manager

```
$ service vault restart
```

CHAPTER 7

Managing Resources

The following sections cover information regarding managing the UForge resources. All of these elements can be managed by the UForge Administrator from the page `Administration`.

7.1 Managing the Project Catalog

A Project Catalog is a collection of software components that are available for UForge users to use when building their server templates. A project is one software component. A project includes files, binaries or native packages and may have an EULA (End User License Agreement).

Each organization within the UForge platform has a project catalog, and each member of the organization has access to the project catalog, allowing them to add projects to their appliances. The contents of the project catalog can be updated and managed by administrators, in order to be adapted to the users of the organization. As the administrator, you can:

- add new projects to the catalog
- update existing projects to the catalog
- mark a project obsolete
- delete a project

You can also create custom OS profiles to include specific packages to an existing (standard) OS profile. See *Creating Custom OS Profiles with UI*.

7.1.1 Adding a Project

To add a new project:

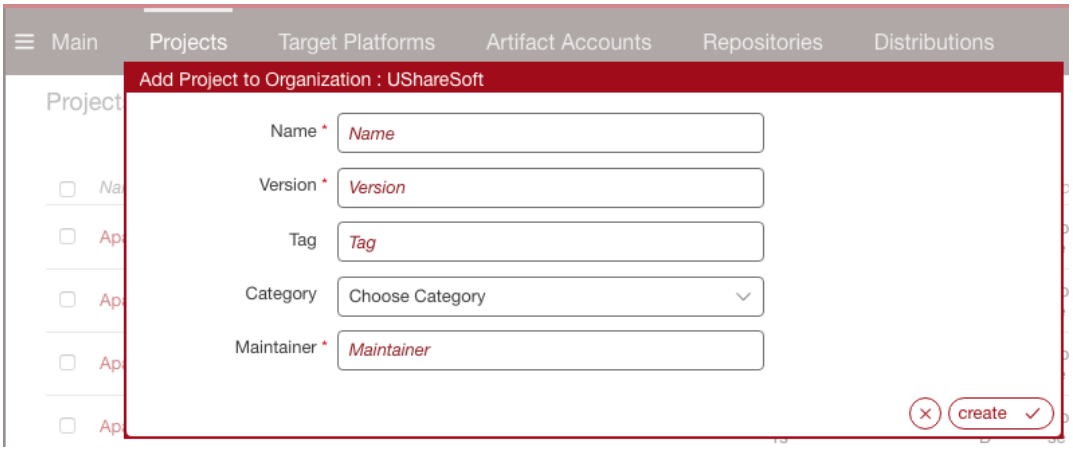
1. Under the `Administration` tab, click `Projects`.
2. Click on `add project` in the top right hand.

Note: Projects are associated with a specific version of operating system. If a project supports more than one version of operating system, then you must re-create a new project for each operating system version. To automate the way projects are added and maintained, use the UForge APIs (for example to add the same project to multiple operating systems).

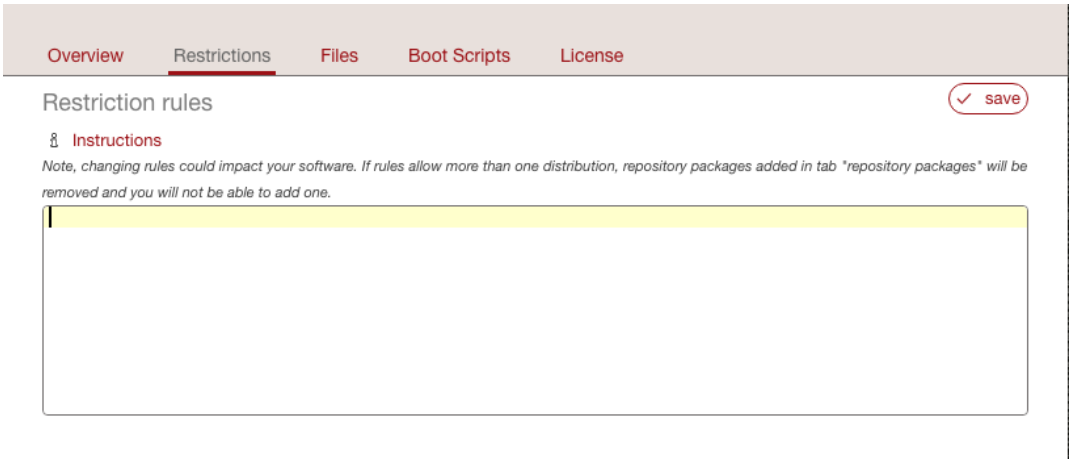
3. Fill in the mandatory information including:
    - Name of the project

Note: If you are creating a project with specific VirtIO drivers which will allow users to publish Windows images to KVM on Red Hat Linux, then the name must begin with `UForgeWinDrivers`.

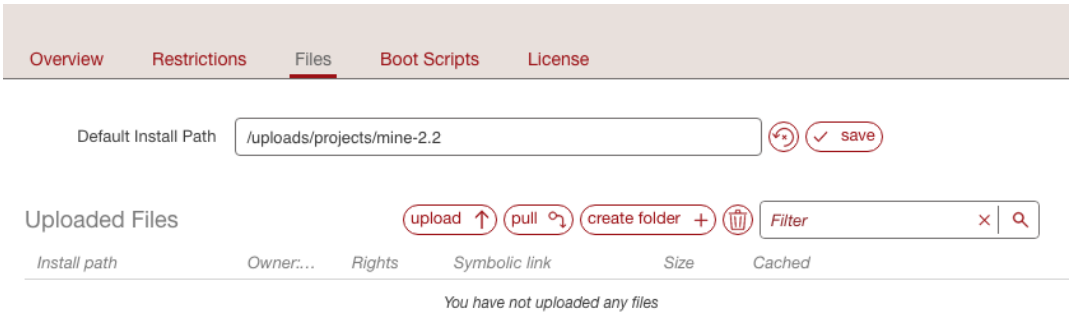
    - Version
    - Project tag
    - Category
    - Maintainer



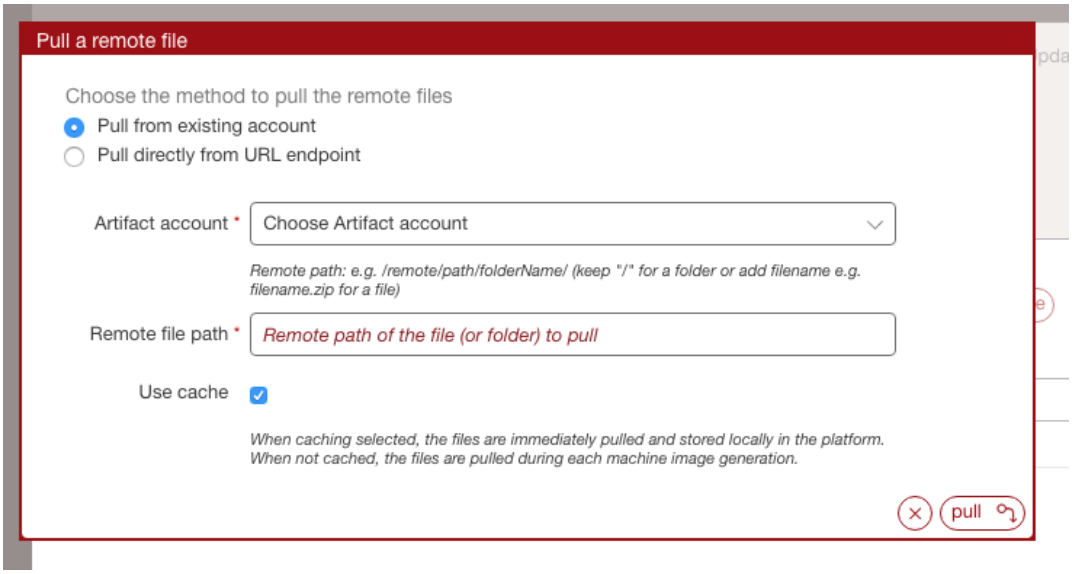
4. Click `create`.
5. Optionally, you can add the following meta-data on the `Overview` tab:
    - a logo for the project
    - website information of where the project came from
    - description
6. Optionally, you can restrict the distribution or target format that the project applies to, from the `Restrictions` page. The restriction rule is set as a regular expression. For more information, refer to *Restricting Projects for OSes and Formats*.



7. You can add files by clicking `upload`. If you want to add many files, you can create a folder and add files within the folder.



If you want to include files from a remote location click on `pull`. Select how the file should be retrieved and if it should be cached.



8. You can add a license (optional). To do so,
   (a) Go to the `License` tab
   (b) Select the license type from the drop-down menu.

(c) Click `upload`.

(d) Select the file and click `open`.

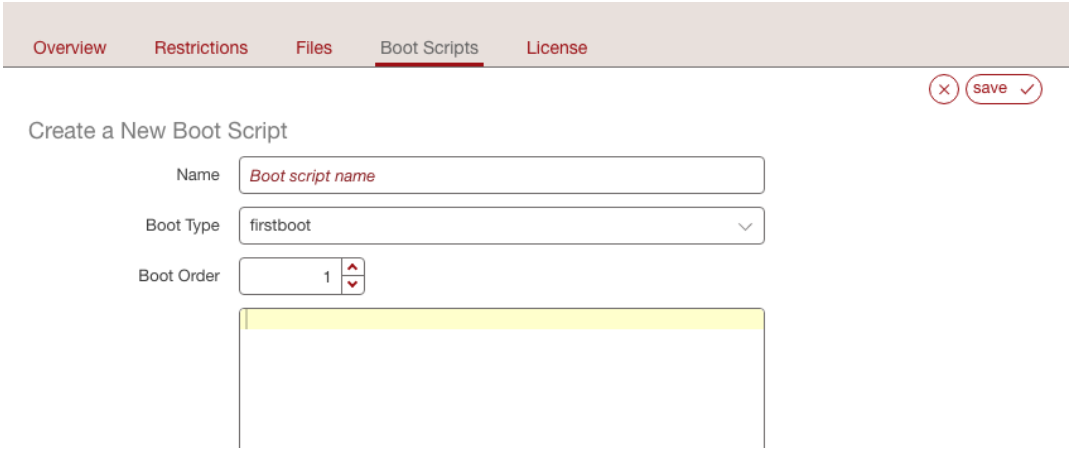
9. You can add a bootscript (optional) on the `Bootscript` tab.

   (a) Enter the name.

   (b) Select the type. If you select first boot, then the boot script will be launched only once, the first time the instance is launched. If you select every boot, then the boot script will be launched every time the instance is rebooted.

   (c) Set the boot order.

   (d) Enter the boot script.



10. Click `Save`.

7.1.2 Restricting Projects for OSes and Formats

Under `Projects` you can restrict the usage of a project based on a distribution name, family, architecture or for a specific machine image format.

- To set restriction rules for a project:

  1. Under the `Administration` tab, click `Projects`.
  2. Select the project you want to modify.
  3. **Go to the `Restrictions` tab. Enter the restriction rule. The restriction rule is represented by a logical expression**

     - `object` is either `Distribution` or `TargetFormat`
     - for `Distribution` field is `family`, `pkgType`, `name`, `version` or `arch`. The `version` must be a major version.
     - for `TargetFormat` field is `name` or `type`
     - `value` is the value you want to match with the fields. For example, `CentOS` for Distribution name, `linux` for Distribution family, `x86_64` for Distribution arch, `VirtualBox` for TargetFormat name, `cloud` for TargetFormat type.
     - logical operator is `||` for OR and `&&` for AND
     - carriage return is not authorized

For example, if the project is designed only for distributions CentOS 7 x86_64 or Debian 8 x86_64, or for TargetFormat with type virtual, then you would note the Restriction rule as follows:

```
(Distribution#arch=x86_64 && ((Distribution#name=CentOS && Distribution#version=7) ||␣
↪(Distribution#name=Debian && Distribution#version=8))) || TargetFormat#type=virtual
```

Note: If your project is limited to a certain target format and you generate an image in another format, your appliance will be generated but the project will not be part of the final image.

- Restriction rule values:

By using the command-line tool hammr, you are able to retrieve available distributions:

```
$ hammr os list --url https://uforge.example.com/api -u username -p password
```

Where `Distribution` fields possible values are:

- `name`: the value listed in the *Name* column
- `version`: the value listed in the *Version* column
- `arch`: the value listed in the *Architecture* column
- `family`: can be one of the following `linux`, `unix` or `windows`
- `pkgType`: can be either `application/x-rpm` or `application/x-debian-package`

By using the command-line tool hammr, you are able to retrieve available machine image formats:

```
$ hammr format list --url https://uforge.example.com/api -u username -p password
```

Where `TargetFormat` fields possible values are:

- `name`: the value listed in the *Builder Type* column
- `type`: can be one of the following `cloud`, `container`, `physical` or `virtual`

7.1.3 Updating a Project

UForge provides the default projects for the OSes provided.

To modify the projects:

1. Under the `Administration` tab, click `Projects`.
2. You will see a list of all the projects that are part of the org.
3. Click on the project you wish to edit. At this stage you can change any of the meta-data, restrictions and files uploaded.

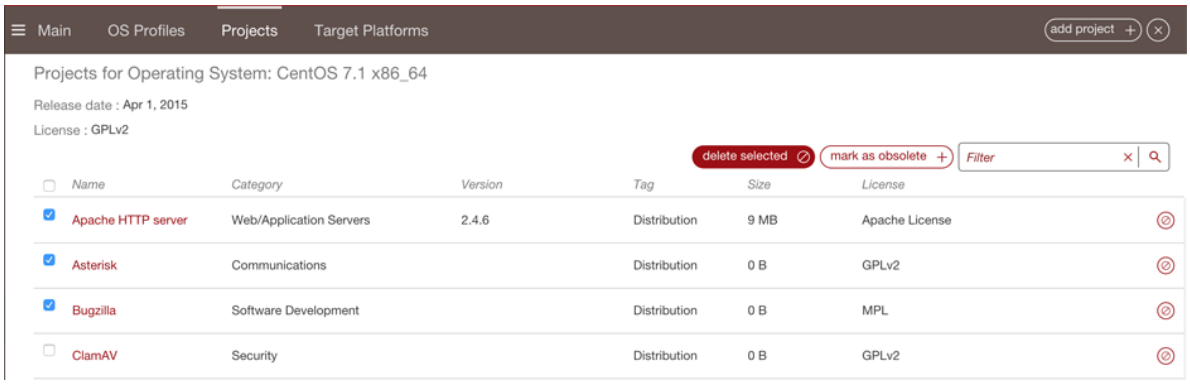
7.1.4 Deleting a Project

When deleting projects from the catalog, it is important to understand that you should check that no appliance templates are using the project you wish to delete (otherwise this may break generating a machine image for the appliance template).

Note: If an appliance template is still using a project, you can remove the project from the catalog by marking it as `obsolete`. This does not delete the project contents, however users cannot use the project for new appliance templates. See *Marking a Project as Obsolete* for more information.

To delete the project:

1. Under the `Administration` tab, click `Projects`.
2. You will see a list of all the projects that are part of the org.
3. To delete the project you can either:
   - Click on the `delete` icon at the extreme right-hand side in the table for the project item.



   - Click on the project item to edit it, then click on the `delete` icon at the top right-hand side of the project edit page.

7.1.5 Marking a Project as Obsolete

Another way to remove a project from the project catalog is to mark the project as `obsolete`. This ensures that any existing templates that use the project can still generate images correctly, however the project is no longer accessible in the catalog for new appliance templates.

To mark a project as `obsolete`:

1. Under the `Administration` tab, click `Projects`.
2. You will see a list of all the projects that are part of the org.
3. Click on the project to edit it.
4. Click on the `obsolete` icon at the top right-hand side of the project edit page.

7.2 Allowing Access to Image Formats

Image formats can be managed at the level of the organization or of the user.

An administrator who wants to add format rights to several users within an organization should create the formats at the level of the organization and then add them to a subscription profile. In this case, all new users created with the given subscription profile will have access to the formats.

In order to add a format you must create a target format and target platform.

Note: Without a target platform, the image cannot be published.

The administrator can add or remove image formats for a specific user account using the command-line interface, as described in *Adding Formats to a Specific User*.

7.2.1 Listing Formats

In UForge AppCenter, in order for a user to use a given format, a Target Format must be created and enabled for the given user and organization. Moreover, if the user wants to use a given Target Format for publishing an image, then a Target Platform must also be created, enabled and associated to the Target Format.

Note: As Target Platforms and Target Formats are created at the org level but then must be enabled for a given user, the list of available formats can differ for users of the same organization.

In order to list the formats available for a given *org*, use the command `org targetformat list`.

In order to list the formats available for a given *user*, use the command `user targetformat list`.

7.2.2 Adding Formats to an Organization Using the CLI

To add access to a format to an organization, you must:

1. Create a target format category using `uforge org category create`
2. Create target format using `uforge org targetformat create`
3. If the target format type is Cloud, create a target platform using `uforge org targetplatform create`. This target platform is necessary to publish image generated with the target format. The target platform must have a cloud account associated.
4. If the target format type is Cloud, add the target format to the target platform using `uforge org targetplatform addTargetFormat`.

5. (Optional) Add the format to a subscription profile using `uforge subscription targetFormat add`. To add a target platform use either:
   - `uforge subscription targetformat add`
   - `uforge subscription targetplatform add`

Note: In order to apply the changes to all users (even those already created), use the option `--allusers`. For example:

```
$ uforge subscription targetformat add --targetformat ovf qcow2 vbox --
↪allusers --name sub --url https://uforge.usharesoft.com:443 -u $ADMIN -p
↪$PASS
```

6. Enable the target format for the organization using `uforge org targetformat enable`.
7. Enable the target format using `user targetformat enable`.

Once the format is created in an organisation, you can either:

- add it to a group of users by adding it to the subscription profile (refer to *Adding Formats Using Subscription Profile*)
- add it to an individual user (refer to *Adding Formats to a Specific User*)

7.2.3 Adding Image Formats from the GUI

You can create your own image formats from the UForge GUI. To do so:

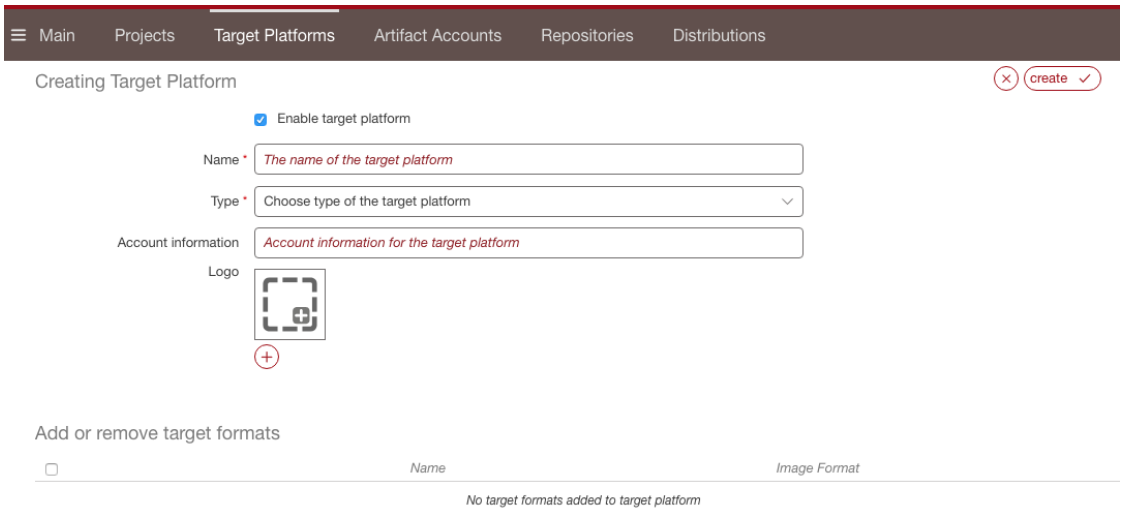
1. Create a target platform.
2. Create a target format.

Create a Target Platform

To create a target platform:

1. From the `Administration` tab, click on `Target Platforms`.
2. In the top right-hand, click on `new target platform`.

The following window will appear:



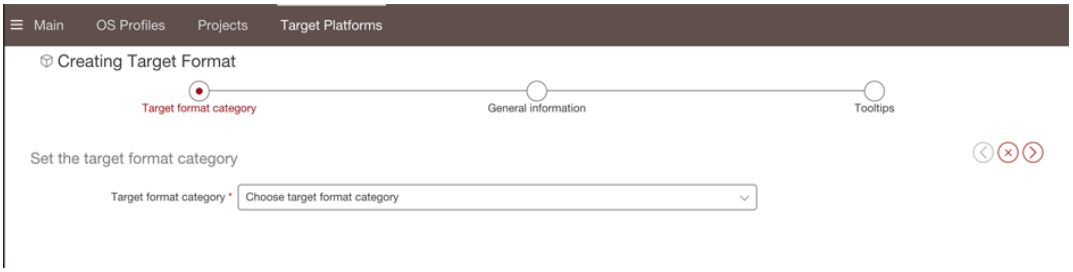
3. Enter the name of the target platform.
4. Select the type from the drop-down menu.
5. Optionally you can click on the plus (+) to add a logo.
6. If you do not want the target platform to be visible immediately, click on the check box next to `Enable` to deselect.
7. You can add available target formats by selecting and clicking on the up arrow. The top list shows the formats that have been added to the target platform. To create new target formats, refer to the following section.
8. Click `create` in the top right to complete the creation.

Create a Target Format

To create a target format:

1. From the `Administration` tab, click on `Target Platforms`.
2. In the top right-hand, click on `new target format`.

3. From the drop-down menu, choose the target format category and click the next arrow button.
4. Enter the name of the target format.
5. Select the type and the image format from the drop-down menus.
6. Optionally you can click on the plus (+) to add a logo.
7. If you do not want the target format to be visible immediately, click on the check box next to `Enable` to deselect.
8. On the `Tooltips` page enter the Credentials, image and publish information.
9. On the `Target Platforms` page you can attach your target format to a target platform. To add the target format to a target platform, select the target platform from the bottom table and click the up arrow. Your target format will be attached to all the target platforms listed in the top table.
10. Click next to complete.

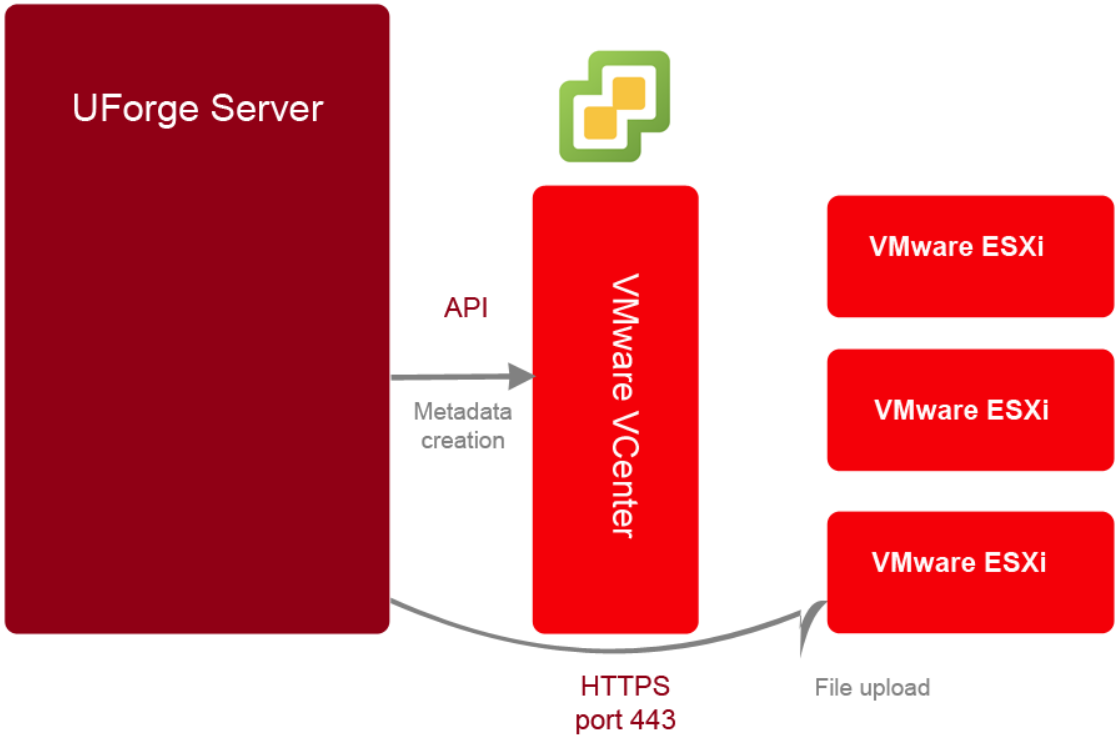


7.2.4 Opening Ports for Publication to Specific Formats

Depending on the target format your users will have access to, you may have to open specific ports.

VCenter

Publishing to VMWare VCenter has the following workflow. If your users will publish to VCenter, you should open port 443 in HTTPS.

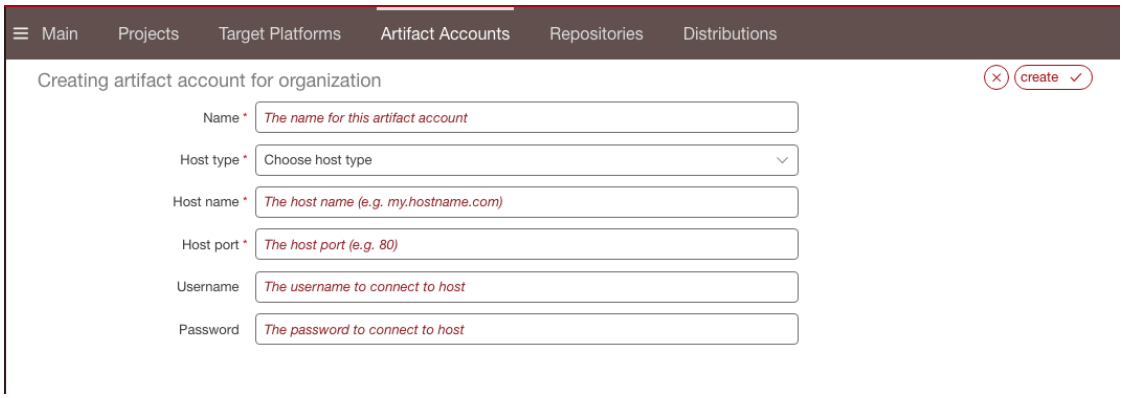


7.3 Managing the Organisation Artifact Accounts

You can create artifact accounts in UForge, which can be used by all the users of the organisation.

1. Go to the `Credentials` tab in the left-hand sidebar.
2. Go to `Artifact Account`.

3. Click on `create`.



4. Enter a name for the account.
5. Select the type from the drop-down menu.
6. Enter the host name and port.
7. Click `create` to complete.

7.4 Adding OS Distributions

Open source operating system versions are taken from the official repository mirror or the UForge repository cache. Proprietary operating systems such as Red Hat Enterprise Linux are not; therefore it is the responsibility of the end customer (or reseller if they have correct agreements in place to re-distribute an operating system) to have the original ISO images of the operating system in questions. Refer to *Hosting Proprietary Packages*.

To enable UForge to generate images based on the operating system it needs all the meta-data of the packages comprising the operating system. This meta-data includes the location in the storage of the package as well as dependency information that is used during generation. Furthermore, certain specific UForge packages must be populated for this operating system.

Note: Custom repositories are supported in UForge, only if `Directory listings` configuration is enabled. They are treated like other OS packages.

Warning: When using UForge, you have to comply with the license agreement of OSes and software which UForge handles, in particular:

- **Publishing OS image of RHEL (Red Hat Enterprise Linux) subscription to public cloud** Cloud provider has to be CCSP (Certified Cloud & Service Provider) and you must register to Red Hat Cloud Access. For more details, please confirm with cloud provider.
- **Scanning server** You have to check whether the licenses of OS and software which the source machine contains allow you to use them on the destination server which you are migrating to. If the source machine contains rpm packages which Red Hat provides, you must register repository with these rpm packages to UForge. Unless you register repository, UForge automatically regenerates rpm packages which the source machine contains, and regenerated packages are NOT supported by Red Hat.
- **Handling Microsoft Windows** Refer to *Microsoft Windows and UForge*.

Note: When installing a major version, all minor versions will be included. If you want to restrict to only a few minor versions, you will have to follow this procedure for each minor version you want to install. You should note however, that a scan will take longer if not all minor versions of a distribution are install in your UForge AppCenter. For example, if you scan a CentOS 6.8 machine, but your AppCenter has only been populated with packages up to CentOS 6.7, then the AppCenter will use the machine's yum repo to download the missing packages. As a result, the scan will take longer before completing.

Warning: If you are going to use the migration feature for RHEL or CentOS, you must add the major OS version to UForge AppCenter and attach to this major OS version all the repositories of the OS minor versions.

The steps for adding a distribution are described further on in *Adding an OS to an Organization*. For examples on adding a specific OS version using the CLI, refer to *Adding RPM Type OSes Using CLI*.

7.5 Listing the Installed OSes

All the open source operating system versions are delivered as part of the UForge repository. Proprietary operating systems are not.

From the UI you can see the installed OSes by going to the `Administration` page and going to the `Distributions` tab.

All the open source operating system versions are delivered as part of the UForge repository. Proprietary operating systems such as Red Hat Enterprise Linux or older operating system versions (that have been EOL'd) are not; therefore it is the responsibility of the end customer (or reseller if they have correct agreements in place to re-distribute an operating system) to have the original ISO images of the operating system in questions.

Note: You can only add an operating system version that is officially supported by the UForge AppCenter and has been certified by Fujitsu.

7.6 Listing the Enabled OSes with CLI

To get a list of the operating systems that are currently enabled on your UForge platform, run the CLI run the command `uforge org os list`.

Login to one of the UForge instances:

```
$ uforge org os list -u $ADMIN -p $PASS
Getting operating systems ...
Success: Found the following operating systems
+----------------+-----------------+--------------+--------+---------+---------+------
↪------------+
| Distribution   | Version         | Architecture | Access | Visible | Default |␣
↪Release Date     |
+----------------+-----------------+--------------+--------+---------+---------+------
↪------------+
| CentOS         | 6.3             | i386         |   X    |         |         | 2012-
↪07-01       |
| CentOS         | 6.3             | x86_64       |   X    |         |         | 2012-
↪06-26       |
```

(continues on next page)

(continued from previous page)

```
| CentOS          | 6.4              | i386          |   X    |          |          | 2013-
↪03-01       |
| CentOS          | 6.4              | x86_64        |   X    |    X     |          | 2013-
↪03-01       |
.... rest omitted for clarity
```

7.7 Adding an OS to an Organization

You can add a supported OS using the UI or using the CLI.

Older operating system versions (that for example have been EOL'd) or proprietary operating systems such as Red Hat Enterprise Linux are not automatically populated at the installation phase. Population of such operating system versions must be done manually after the initial installation of UForge is complete.

Note: You can only add an operating system version that is officially supported by the UForge platform and has been certified by Fujitsu.

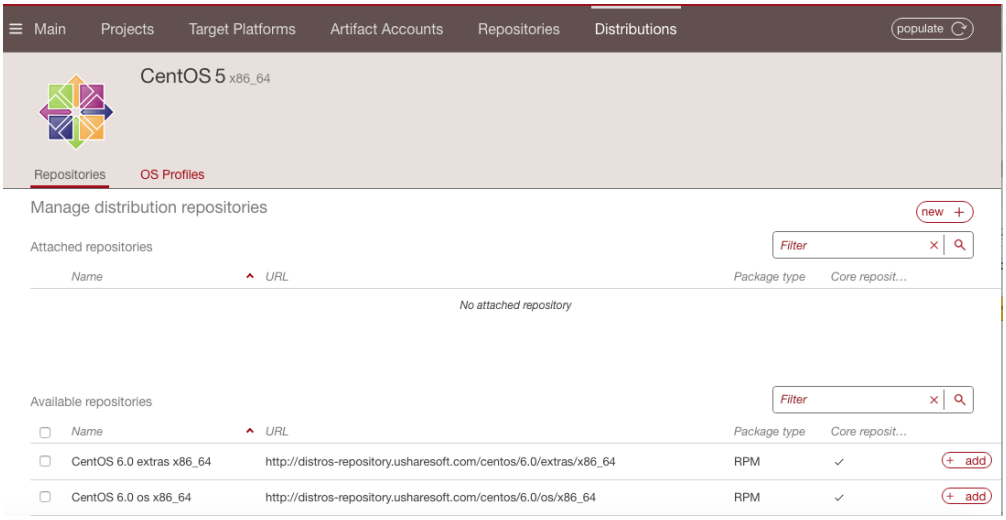
Before adding an OS to the organization you must retrieve data from the Fujitsu repository. For more information refer to *Adding OS Distributions*.

7.7.1 Adding an OS Using the UI

In order to add an OS to an organisation you manage:

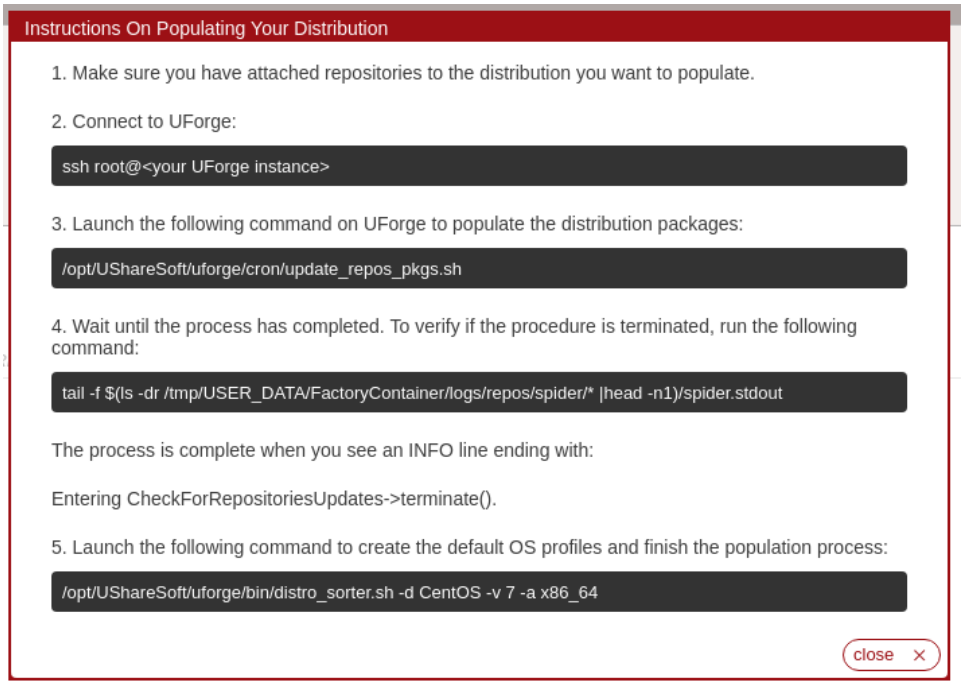
1. From the `Administration` page, go to the `Distributions` tab.
2. Click on `add` in the top left.
3. Select the distribution, version and architecture from the list.
4. Click `add`.
5. You must add repositories to this distribution from the list of `Available repositories` on the `Repositories` tab. The repositories that you can add are listed in the bottom list. The top list is the list of repositories that have been added.

To add a repository, check it from the bottom table and click `add selected`. It will now appear in the top table.



Note: You can also create new repositories by clicking on `new`. You will be redirected to the page to create new repositories. For more information, refer to *Creating a Repository Using UI*.

1. You must populate the repositories. To do so, click on `populate` at the top right and follow the instructions in the pop-up window.



Note: The populate process will provide the base OS profiles.

2. To add an OS profile (opional), go to the `OS Profiles` tab. You can create one by clicking on `add os profile` in the top right. Refer to *Creating Custom OS Profiles with UI*.

When adding CentOS, Debian and Red Hat, the major versions are automatically marked as Milestones when the

distribution is added to the platform. For more information on Milestones, refer to *Creating and Managing Milestones*.

7.7.2 Adding an OS Using the CLI

In order to add an OS to an organization you must do the following:

1. Connect to one of your UForge platform instances.
2. Create the distribution in the organization.
3. Create the repositories. This includes the official repository (see *Official UForge Tool Repositories*) as well as the specific UForge tool repository (see *Specific UForge Tool Repositories*). This is covered in steps 6 and 7 in the section *Example for Adding CentOS*.
4. Attach the repository to the distribution.
5. Launch Spider to fill the repositories with the packages.

This must be done for each version of an OS. For example CentOS 6.5 i386. It is not possible to do this for all CentOS versions at once.

For example, to add CentOS 6.5 i386:

1. Add the distribution to the organization, using the official name and version.

```
$ uforge org os add --name CentOS --version 6.5 --arch i386 -u $ADMIN -p
↪$PASS
Getting default organization ...
Success: Add operating system OK
```

2. Create the repositories.

```
$ uforge org repo create --name CentOSplus --repoUrl http://vault.centos.org/
↪6.5/centosplus/i386 --type RPM -u $ADMIN -p $PASS
Getting default organization ...
Success: Created repository with url [http://vault.centos.org/6.5/centosplus/
↪i386] to default organization
+------+----------------+------------+---------------------------------------
↪------+-------+
|   Id | Off. Supported | Name       | Url                                  ␣
↪      | type  |
+------+----------------+------------+---------------------------------------
↪------+-------+
|  355 |                | CentOSplus | http://vault.centos.org/6.5/
↪centosplus/i386 | RPM   |
```

3. Attach the repository to the distribution.

```
$ uforge org repo os attach --name CentOS --version 6.5 --arch i386 --repoId␣
↪355 -u $ADMIN -p $PASS
Getting default organization ...
Success: Attach distribution to repository [355]
```

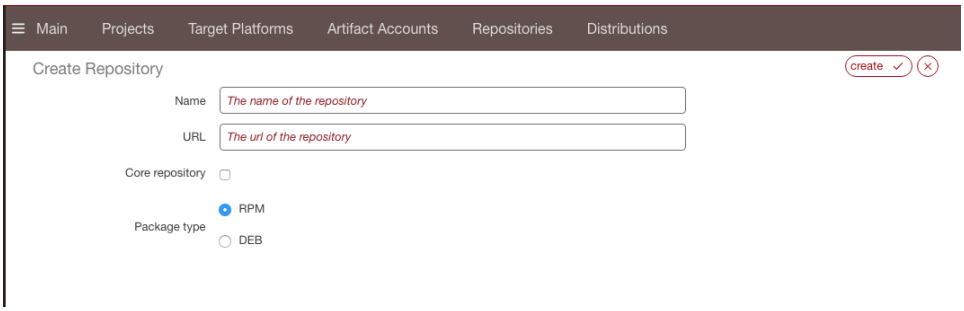
4. Launch Spider and other population steps. Refer to *Adding OS Distributions*.

When adding CentOS, Debian and Red Hat, the major versions are automatically marked as Milestones when the distribution is added to the platform. For more information on Milestones, refer to *Creating and Managing Milestones*.

7.8 Creating a Repository Using UI

You can manage existing repositories and create new ones from the UForge UI, from the `Administration` page.

1. Go to the `Repositories` tab. The existing repositories will be listed.
2. To create a new repo, click on `add repository` in the top right.
3. Enter the name and URL.



Note: Check `core repositories` for all the default repositories of officially supported OSes (for a list of supported OSes, refer to the section `Supported Operating Systems` in the User Guide). Do not check this box for repositories that are not part of the core distribution, such as epel or VMware tools. When generating a machine image, packages tagged as `core` are installed first, before other packages.

7.9 Updating a Repository Using CLI

You can manually launch an update of a specific repository using `uforge org repo refresh`. To do this, you must have the `repoId`, which you can find using `uforge org repo list`. For example, to launch the update:

```
$ uforge org repo refresh trigger --repoId 355 -u $ADMIN -p $PASS
```

In order to view the status of a repository update, launch:

```
$ uforge org repo refresh status --repoId 355 --refreshId 887 -u $ADMIN -p $PASS
```

To view a list of all the updates launched, use: `uforge org repo refresh list`.

7.10 Removing OSes and Distributions

You cannot remove an OS from an organization once added. You can only disable it, in which case it can no longer be used.

To disable a distribution, for example CentOS, for all users of an organization specify only the OS name, in which case all the versions will be disabled:

```
$ uforge org os disable --name CentOS -u $ADMIN -p $PASS
```

If you only want to remove a specific version of a distribution (for example CentOS 5), run:

```
$ uforge org os disable --name CentOS --version 5 -u $ADMIN -p $PASS
```

7.11 Updating an OS Repository

This procedure assumes that you have already populated an OS repository as described in *Adding OS Distributions*.

The following section gives an example for updating CentOS. They can be adjusted for your particular version, and are applicable to OpenSUSE, RedHat Enterprise Linux and Scientific Linux.

1. Create the distribution repository.

   For example, for the CentOS 6.5 repository:

   ```
   $ uforge org repo create --name "CentOS 6.5 os" --repoUrl http://
   ↪vault.centos.org/6.5/os/x86_64/ --type RPM --coreRepository -u
   ↪$ADMIN -p $PASS

   Success: Created repository with url [http://vault.centos.org/6.5/
   ↪os/x86_64/] to default organization
   ```

   The `--name` specified here is the "tagname" that will be shown in the UI when creating an appliance. The `--repoUrl` can be either `http://` or `file://`.

   Warning: You must use the `--coreRepository` flag for all the default repositories of officially supported OSes. For a list of supported OSes, refer to the User Guide. Do not use `--coreRepository` for repositories that are not part of the core distribution, such as epel or VMware tools. When generating a machine image, packages tagged as `--coreRepository` are installed first, before other packages.

2. Attach repository to the distribution as follows for each repository (your own repository and the UShareSoft tool repository):

   ```
   $ uforge org repo os attach --name CentOS --arch x86_64 --version 6.5 --repoIds␣
   ↪354 -u $ADMIN -p $PASS
   ```

   The `--repoIds` specified here are the space-separated "id" of previously created repositories, shown by command `uforge org repo list -u $ADMIN -p $PASS`.

3. Populate repository packages:

   ```
   $ /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
   ```

   Note: This procedure may take a long time.

4. To verify if the procedure is terminated, run the following command:

   ```
   $ tail -f /tmp/USER_DATA/FactoryContainer/logs/repos/spider/<directory␣
   ↪name with date>/spider.stdout
   ```

The procedure is complete when you see the line `INFO` ends with `Entering CheckForRepositoriesUpdates->terminate()`

7.12 Deleting an OS Repository

If you want to remove an OS repository from your UForge AppCenter, you will need to run `org repo delete` as follows:

1. You will need to identify the ID number associated with the repo you want to delete. Run:

```
org repo list
```

2. Delete the repository by running the `org repo delete` with the argument `--id` indicating the repo to be deleted. For example:

```
org repo delete --ids 127
```

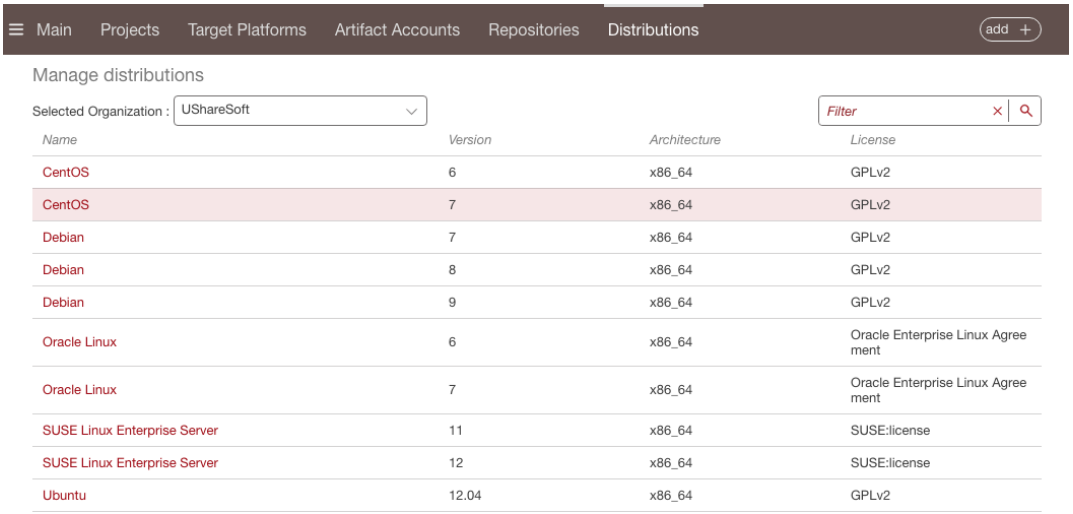
7.13 Creating Custom OS Profiles with UI

UForge provides a set of OS profiles. If you want to create another OS profile, you can create one using UForge's graphical user interface.

> **Warning:** You can delete packages but we do not guarantee that your OS profile will be functional.

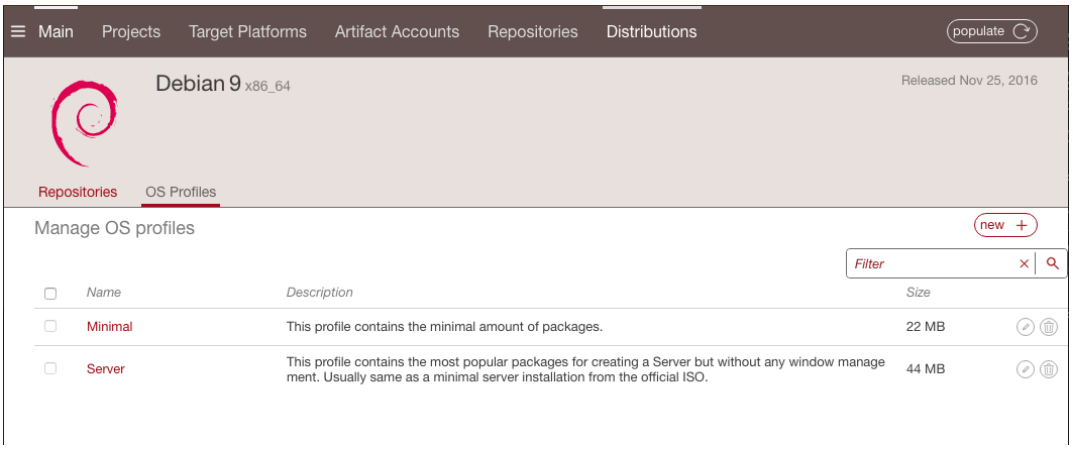
To create a new OS profile:

1. Under the `Administration` tab go to the `Distributions` tab.
2. Click on the operating system. This will provide the current list of profiles this operating system has.

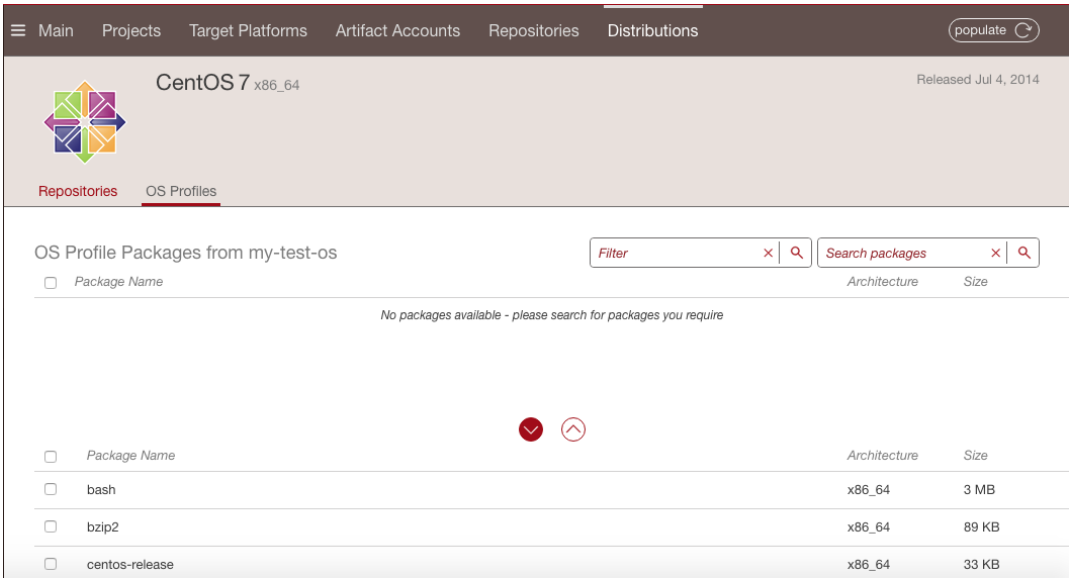


Name	Version	Architecture	License
CentOS	6	x86_64	GPLv2
CentOS	7	x86_64	GPLv2
Debian	7	x86_64	GPLv2
Debian	8	x86_64	GPLv2
Debian	9	x86_64	GPLv2
Oracle Linux	6	x86_64	Oracle Enterprise Linux Agreement
Oracle Linux	7	x86_64	Oracle Enterprise Linux Agreement
SUSE Linux Enterprise Server	11	x86_64	SUSE:license
SUSE Linux Enterprise Server	12	x86_64	SUSE:license
Ubuntu	12.04	x86_64	GPLv2

3. Go to the `OS Profiles` tab.
4. Click on `new` in the middle right.



5. Select the profile which will serve as the base for your custom OS profile.
6. Enter your profile name and click `create`.
7. The packages that are listed in the bottom table are packages that are currently part of the profile. You can search for the packages you want to add. Select the package(s) you want to add to your OS profile and click on the down arrow. Once added they will appear in the bottom table.



7.14 Editing Custom OS Profiles Using UI

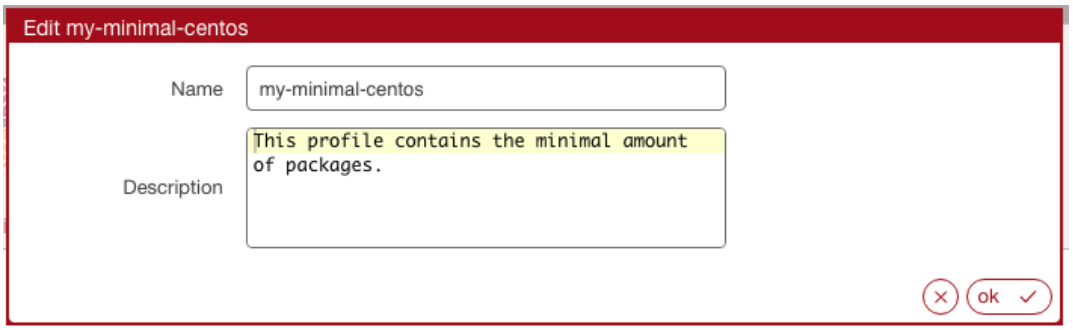
If you have created a custom OS profile, you can edit it at any time.

Note: You cannot modify a profile provided by UForge.

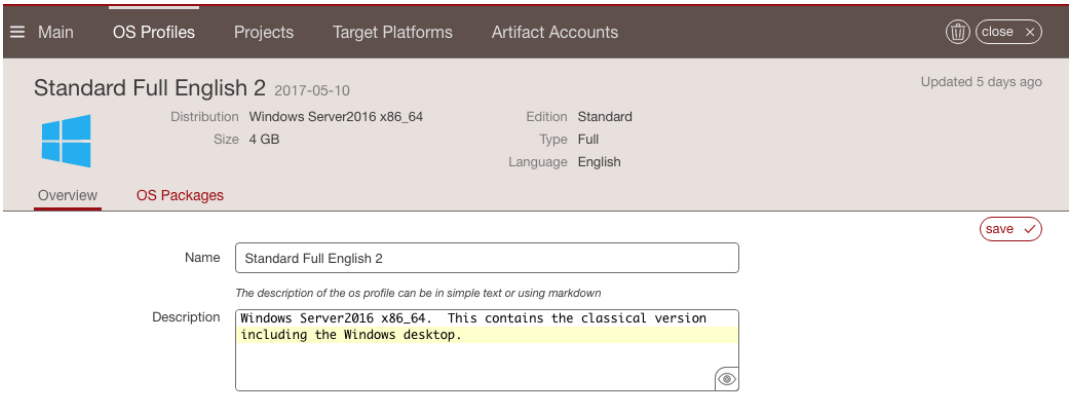
Warning: You can delete packages but we do not guarantee that your OS profile will be functional.

To edit your OS profile:

1. Under the `Administration` tab, go to `Distributions`.
2. Click on the operating system the OS profile is associated with.
3. Go to the `OS Profiles` tab you will see a list of the current profiles this operating system has.
4. To modify the name or description, click on the pencil icon on the right hand side. A pop-up with appear. Modify and click `ok`.



5. To modify the packages, click on the OS profile in the list.



   On this view, for Windows, you can see the distribution your OS profile is based on, its size, the date you created the OS profile in UForge (next to the OS profile name) as well as the last time it was edited.

6. Search for the packages you want to add. Select the package(s) you want to add to your OS profile. The packages that you can add are listed in the top list. The bottom list is the list of packages that are currently part of the profile.

7.15 Official UForge Tool Repositories

The following is a list for all the distributions that can be used to create an official repository. These will be used in step 6 of the examples below.

`Ubuntu (example 10.04)`

- http://distros-repository.usharesoft.com/ubuntu/lucid/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid multiverse restricted universe main
- http://distros-repository.usharesoft.com/ubuntu/lucid-security/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid-security multiverse restricted universe main

- http://distros-repository.usharesoft.com/ubuntu/lucid-backports/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid-backports multiverse restricted universe main
- http://distros-repository.usharesoft.com/ubuntu/lucid-updates/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid-updates multiverse restricted universe main

`Debian (example version 6)`

- http://distros-repository.usharesoft.com/debian/squeeze/mirror/ftp.fr.debian.org/debian/ squeeze contrib non-free main
- http://distros-repository.usharesoft.com/debian/squeeze-updates/mirror/ftp.fr.debian.org/debian/ squeeze-updates contrib non-free main
- http://distros-repository.usharesoft.com/debian/security/squeeze/updates/mirror/security.debian.org/ squeeze/updates main contrib non-free

`CentOS (example CentOS 6.7)`

- http://distros-repository.usharesoft.com/centos/6.7/updates/x86_64
- http://distros-repository.usharesoft.com/centos/6.7/extras/x86_64
- http://distros-repository.usharesoft.com/centos/6.7/os/x86_64

`OpenSUSE (example version 12.2)`

- http://distros-repository.usharesoft.com/opensuse/distribution/12.2/repo/oss/
- http://distros-repository.usharesoft.com/opensuse/distribution/12.2/repo/non-oss/
- http://distros-repository.usharesoft.com/opensuse/update/12.2/

`Scientific Linux (example version 6.6)`

- http://distros-repository.usharesoft.com/scientificlinux/6.6/x86_64/os/
- http://distros-repository.usharesoft.com/scientificlinux/6.6/x86_64/updates/fastbugs/
- http://distros-repository.usharesoft.com/scientificlinux/6.6/x86_64/updates/security/

`RedHat Enterprise Linux`

- You need to use your own repository.

7.16 Specific UForge Tool Repositories

The following is a list of specific UForge tool repositories that can be added. These will be used in step 7 of the examples below.

`CentOS` (example version 6, arch x86_64):

- http://distros-repository.usharesoft.com/usharesoft/centos/6/x86_64/

`Red Hat Enterprise Linux` (example version 6.2, arch x86_64):

- http://distros-repository.usharesoft.com/usharesoft/rhel/6.2/x86_64/

`Oracle Linux` (example version 6, arch x86_64)

- http://distros-repository.usharesoft.com/usharesoft/oraclelinux/6/x86_64/

`OpenSUSE` (example version 12.1, arch x86_64):

- http://distros-repository.usharesoft.com/usharesoft/opensuse/12.1/x86_64/

`Scientific Linux` (example version 6, arch x86_64):

- http://distros-repository.usharesoft.com/usharesoft/scientificlinux/6/x86_64/

`Debian` (example version 8, arch x86_64) [arch=amd64]:

- http://distros-repository.usharesoft.com/usharesoft/debian/ jessie main

`Ubuntu` (example version 14.04, arch x86_64) [arch=amd64]:

- http://distros-repository.usharesoft.com/usharesoft/ubuntu/ trusty main

7.17 Adding RPM Type OSes Using CLI

The following sections give examples for adding CentOS and RedHat Enterprise Linux using the CLI. They can be adjusted for your particular version, and are applicable to OpenSUSE and Scientific Linux.

7.17.1 Example for Adding CentOS

The following is a concrete CLI example to begin the population of CentOS 6.5 64bit:

1. Connect to UForge:

```
$ ssh root@<your UForge instance>
```

2. In order for the following commands to be generic you can set some variables in your environment.

```
$ . /etc/UShareSoft/uforge/uforge.conf
ADMIN=$UFORGE_WEBSVC_LOGIN ; PASS=$UFORGE_WEBSVC_PASSWORD
```

3. Run the following CLI command in order to create the distribution:

```
$ uforge org os add --name CentOS --arch x86_64 --version 6.5 -u $ADMIN -
↪p $PASS
```

4. Enable the new operating system for the organization. The following command enables CentOS 6.5 in the default organization:

```
$ uforge org os enable --name CentOS --version 6.5 --arch x86_64 -u
↪$ADMIN -p $PASS
```

5. Enable the user to use the operating system. The user must be a member of the organization. This step can be done later.:

```
$ uforge user os enable --account root --name CentOS --version 6.5 --
↪arch x86_64 -u $ADMIN -p $PASS
```

6. Create the distribution repository. The following example shows the creation of an official CentOS repository. However, you can also create a repository based on the UForge official repository as shown later.

   For example, for the CentOS 6.5 repository:

```
$ uforge org repo create --name "CentOS 6.5 os" --repoUrl␣
↪http://vault.centos.org/6.5/os/x86_64/ --type RPM --
↪coreRepository -u $ADMIN -p $PASS

Success: Created repository with url [http://vault.centos.
↪org/6.5/os/x86_64/] to default organization
```

The `--name` specified does not need to be an official name. It is the repository name that will be shown in the UI when pinning an appliance. The `--repoUrl` can be either `http://` or `file://`.

Warning: You must use the `--coreRepository` flag for all the default repositories of officially supported OSes (for a list of supported OSes, refer to :ref: *uforge-supported-os-formats*). Do not use `--coreRepository` for repositories that are not part of the core distribution, such as epel or VMware tools. When generating a machine image, packages tagged as `--coreRepository` are installed first, before other packages.

http://distros-repository.usharesoft.com/ is an official public repository that users can use to populate the distributions. Official repositories such as Ubuntu and Debian periodically delete some package versions. In the http://distros-repository.usharesoft.com/ repository, package versions are never deleted. This can facilitate investigations on older systems.

7. You must then add the specific UForge tool repository. The repository to attach for `CentOS` (example version 6, arch x86_64) is the following:

   - http://distros-repository.usharesoft.com/usharesoft/centos/6/x86_64/

   For example:

```
$ uforge org repo create --name "CentOS 6.5 os" --repoUrl http://
↪distros-repository.usharesoft.com/usharesoft/centos/6/x86_64/ --
↪type RPM -u $ADMIN -p $PASS
```

Note: For a complete list of the different repositories that can be attached, refer to *Specific UForge Tool Repositories*.

8. Attach repository to the distribution as follows for each repository (your own repository and the UShareSoft tool repository):

```
$ uforge org repo os attach --name CentOS --arch x86_64 --version 6.5 --
↪repoIds 354 -u $ADMIN -p $PASS
```

   The `--repoIds` specified here are the space-separated "id" of previously created repositories, shown by command `uforge org repo list -u $ADMIN -p $PASS`.

9. Populate repository packages:

```
$ /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
```

   Note: This procedure may take a long time.

10. To verify if the procedure is terminated, run the following command:

```
$ tail -f /tmp/USER_DATA/FactoryContainer/logs/repos/spider/
↪<directory name with date>/spider.stdout
```

The procedure is complete when you see the line `INFO` ends with `Entering CheckForRepositoriesUpdates->terminate()`

11. Create OS profile based on packages (minimal, server, etc.):

```
$ /opt/UShareSoft/uforge/bin/distro_sorter.sh -d CentOS -v 6.5 -a x86_64
```

7.17.2 Example for Adding Red Hat Enterprise Linux

Note: Before populating Red Hat Enterprise Linux, you should complete the steps in *Hosting Proprietary Packages*, unless you have a Red Hat Satellite, in which case you should contact your Red Hat Satellite administrator for the Satellite repo URL.

The following is a concrete example to begin the population of Red Hat Enterprise Linux version 7, 64bit:

1. Connect to UForge:

```
$ ssh root@<your UForge instance>
```

2. In order for the following commands to be generic you can set some variables in your environment.

```
$ . /etc/UShareSoft/uforge/uforge.conf
ADMIN=$UFORGE_WEBSVC_LOGIN ; PASS=$UFORGE_WEBSVC_PASSWORD
```

3. Run the following CLI command in order to create the distribution:

```
$ uforge org os add --name "RedHat Enterprise Linux" --arch x86_64 --
↪version 7 -u $ADMIN -p $PASS
```

4. Enable the new operating system for the organization. The following command enables Red Hat Enterprise Linux version 7 in the default organization:

```
$ uforge org os enable --name "RedHat Enterprise Linux" --arch x86_64 --
↪version 7 -u $ADMIN -p $PASS
```

5. Enable the user to use the operating system. The user must be a member of the organization. This step can be done later.:

```
$ uforge user os enable --account root --name "RedHat Enterprise Linux" -
↪-arch x86_64 --version 7 -u $ADMIN -p $PASS
```

6. Create the distribution repository. The following example shows the creation of an official RedHat Enterprise Linux repository.

```
$ uforge org repo create --name "RedHat 7" --repoUrl http://<your-
↪repo> --type RPM --coreRepository -u $ADMIN -p $PASS
```

The `--name` specified here is the "tagname" that will be shown in the UI when creating an appliance. The `--repoUrl` can be either `http://` or `file://`.

> **Warning:** You must use the `--coreRepository` flag for all officially supported OSes. If you do not include this argument the packages will not appear in the install profile of appliances built with the corresponding operating system. Do not use `--coreRepository` for distributions that are part of the core distribution. For example, epel or vmwatools are not officially part of the distribution, therefore you should not use `--coreRepository` when adding such repositories.

7. You must then add the specific UForge tool repository. The repository to attach for RedHat Enterprise Linux version 7 arch x86_64 is the following:

   - http://distros-repository.usharesoft.com/usharesoft/rhel/7/x86_64/

   For example:

```
$ uforge org repo create --name "UShareSoft RedHat 7" --repoUrl␣
↪http://distros-repository.usharesoft.com/usharesoft/rhel/7/x86_
↪64/ --type RPM -u $ADMIN -p $PASS
```

Note: For a complete list of the different repositories that can be attached, refer to *Specific UForge Tool Repositories*.

8. Attach repository to the distribution as follows for each repository (your own repository and the UShareSoft tool repository):

```
$ uforge org repo os attach --name "RedHat Enterprise Linux" --arch x86_
↪64 --version 7 --repoIds 432 -u $ADMIN -p $PASS
```

   The `--repoIds` specified here are the space-separated "id" of previously created repositories, shown by command `uforge org repo list -u $ADMIN -p $PASS`.

9. Populate repository packages:

```
$ /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
```

   Note: This procedure may take a long time.

10. To verify if the procedure is terminated, run the following command:

```
$ tail -f /tmp/USER_DATA/FactoryContainer/logs/repos/spider/
↪<directory name with date>/spider.stdout
```

    The procedure is complete when you see the line `INFO` ends with `Entering CheckForRepositoriesUpdates->terminate()`

11. Create OS profile based on packages (minimal, server, etc.):

```
$ /opt/UShareSoft/uforge/bin/distro_sorter.sh -d RedHat -v 7 -a x86_64
```

7.18 Adding DEB Type OSes Using CLI

The following section give a CLI example for adding Ubuntu. It is also applicable for Debian.

7.18.1 Example for Adding Ubuntu

The following is a concrete example to begin the population of Ubuntu 10.04 64bit:

1. Connect to UForge:

```
$ ssh root@<your UForge instance>
```

2. In order for the following commands to be generic you can set some variables in your environment.

```
$ . /etc/UShareSoft/uforge/uforge.conf
ADMIN=$UFORGE_WEBSVC_LOGIN ; PASS=$UFORGE_WEBSVC_PASSWORD
```

3. Run the following CLI command in order to create the distribution:

```
$ uforge org os add --name Ubuntu --arch x86_64 --version 10.04 -u
↪$ADMIN -p $PASS
```

4. Enable the new operating system for the organization. The following command enables Ubuntu 10.04 in the default organization:

```
$ uforge org os enable --name Unbuntu --version 10.04 --arch x86_64 -u
↪$ADMIN -p $PASS
```

5. Enable the user to use the operating system. The user must be a member of the organization. This step can be done later.:

```
$ uforge user os enable --account root --name Unbuntu --version 10.04 --
↪arch x86_64 -u $ADMIN -p $PASS
```

6. Create the distribution repository. The following example shows the creation of an official Ubuntu repository.

```
$ uforge org repo create --name "Ubuntu x86_64 lucid-main" --
↪repoUrl "[arch=amd64] http://distros-repository.usharesoft.com/
↪ubuntu/lucid/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/␣
↪lucid multiverse restricted universe main" --type DEB --
↪coreRepository -u $ADMIN -p $PASS

$ uforge org repo create --name "Ubuntu x86_64 lucid-security" --
↪repoUrl "[arch=amd64] http://distros-repository.usharesoft.com/
↪ubuntu/lucid-security/mirror/bouyguestelecom.ubuntu.lafibre.
↪info/ubuntu/ lucid-security multiverse restricted universe main
↪" --type DEB --coreRepository -u $ADMIN -p $PASS

$ uforge org repo create --name "Ubuntu x86_64 lucid-backports" --
↪repoUrl "[arch=amd64] http://distros-repository.usharesoft.com/
↪ubuntu/lucid-backports/mirror/bouyguestelecom.ubuntu.lafibre.
↪info/ubuntu/ lucid-backports multiverse restricted universe main
↪" --type DEB --coreRepository -u $ADMIN -p $PASS

$ uforge org repo create --name "Ubuntu x86_64 lucid-updates" --
↪repoUrl "[arch=amd64] http://distros-repository.usharesoft.com/
↪ubuntu/lucid-updates/mirror/bouyguestelecom.ubuntu.lafibre.info/
↪ubuntu/ lucid-updates multiverse restricted universe main" --
↪type DEB --coreRepository -u $ADMIN -p $PASS
```

The `--name` specified here is the "tagname" that will be shown in the UI when creating an appliance. The `--repoUrl` can be either `http://` or `file://`.

> **Warning:** You must use the `--coreRepository` flag for all officially supported OSes. If you do not include this argument the packages will not appear in the install profile of appliances built with the corresponding operating system. Do not use `--coreRepository` for distributions that are part of the core distribution. For example, epel or vmwatools are not officially part of the distribution, therefore you should not use `--coreRepository` when adding such repositories.

The syntax of the repoURL for Debian based OSes follows that of the sources.list file.

See https://wiki.debian.org/SourcesList and https://wiki.debian.org/Multiarch/HOWTO (section Setting up apt sources)

Typically, a correct value for the repoURL parameter is either

- http://httpredir.debian.org/debian jessie main
- http://ftp.riken.go.jp/Linux/ubuntu/ precise-security multiverse restricted universe main

Users may also want to restrict per architecture. For example:

```
[arch=amd64] http://distros-repository.usharesoft.com/ubuntu/ ...
```

http://distros-repository.usharesoft.com/ is an official public repository that users can use to populate the distributions. Official repositories such as Ubuntu and Debian periodically delete some package versions. In the http://distros-repository.usharesoft.com/ repository, package versions are never deleted. This can facilitate investigations on older systems.

7. You must then add the specific UForge tool repository. The repository to attach for `Ubuntu` (example version 6, arch x86_64) is the following:

   - http://distros-repository.usharesoft.com/usharesoft/ubuntu/

   For example:

```
$ uforge org repo create --name "UShareSoft Ubuntu x86_64 lucid" -
↪-repoUrl "[arch=amd64] http://distros-repository.usharesoft.com/
↪usharesoft/ubuntu/ lucid main" --type DEB -u $ADMIN -p $PASS
```

Note: For a complete list of the different repositories that can be attached, refer to *Specific UForge Tool Repositories*.

---

8. Attach repository to the distribution as follows for each repository (your own repository and the UShareSoft tool repository):

```
$ uforge org repo os attach --name Ubuntu --arch x86_64 --version 10.04 -
↪-repoIds 354 -u $ADMIN -p $PASS
```

   The `--repoIds` specified here are the space-separated "id" of previously created repositories, shown by command `uforge org repo list -u $ADMIN -p $PASS`.

9. Populate repository packages:

```
$ /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
```

Note: This procedure may take a long time.

10. To verify if the procedure is finished, run the following command:

```
$ tail -f /tmp/USER_DATA/FactoryContainer/logs/repos/spider/
↪<directory name with date>/spider.stdout
```

The procedure is done when you see the line `INFO` ends with `Entering CheckForRepositoriesUpdates->terminate()`

11. Create OS profile based on packages (minimal, server, etc.):

```
$ /opt/UShareSoft/uforge/bin/distro_sorter.sh -d Ubuntu -v 10.04 -a x86_
↪64
```

7.19 Microsoft Windows and UForge

Warning: UForge users must acquire Windows licenses in order to handle Windows OSes in UForge. When publishing Windows OS images or scanning a Windows server, you have to confirm usage conditions of cloud provider and virtualization software which you publish to or scan.

Within UForge, Microsoft Windows is treated differently from Linux/UNIX operating systems because Windows is not bundled with packages. Consequently, it is not possible to create standard (package based) OS Profile as for all the other supported distributions.

Instead, UForge uses a Golden Image as a profile. A Golden Image is an image that contains the basic installation of the Windows version and some extra files. You can generate Golden Images at any time.

A Golden Image can be between 5 to 10 Gb, depending on the selected version.

You will need Golden Images to create Windows appliance templates. If you want to incorporate a Windows update, then you need to create a new set of Golden Images. You can create Golden Images using two different methods:

- scanning a Windows machine and import it to UForge as a Golden Image (*Creating a Golden Image from Scan*) (strongly recommended)
- creating the Golden Image manually and install it on UForge (*Creating a Golden Image Manually*)

Within UForge, the Golden Image used when you create appliances will be the last Golden Image created. In future releases, the different Golden Images will appear as Milestones.

7.19.1 Supported Microsoft Windows Versions

UForge supports Microsoft Windows Server 2008 R2 (this distribution is only 64 bits), 2012, 2012R2, and 2016.

Microsoft delivers 4 versions:

- DataCenter
- Standard
- WebServer
- Enterprise

For each version, there is a Full release and a Core release (without a Desktop).

These versions are available in French, English and Japanese. Unfortunately, one version of Microsoft Windows can NOT support multiple languages.

7.19.2 Restrictions

The following UForge features are not supported with appliances based on Microsoft Windows:

- Package selection at the OS level (however, users can add software via MySoftware or Projects)
- Windows ISO format not supported

7.19.3 Using Microsoft Windows Key Mechanism

UForge generates images without a licence key. Microsoft Windows Server operating systems may be used without an activated licence for a certain period of time, with no functional limitation, but it is the responsibility of users to enter a licence key and to activate it if they want to keep using the server beyond this period.

7.19.4 First Installation of Microsoft Windows

To install Microsoft Windows, follow the instructions below. This example is for Microsoft Windows Server 2008R2. If you want to install for Microsoft Windows Server 2012, 2012R2 or 2016, replace the string `Server2008R2` with the version you want to install in all commands below.

1. First population of Windows:

```
$ uforge org os add --name Windows --arch x86_64 --version Server2008R2
```

2. Activate Windows Server2008R2 inside the UForge organization:

```
$ uforge org os enable --name Windows --arch x86_64 --version Server2008R2
```

3. Activate Windows Server2008R2 for root user (and potentially other preexisting users):

```
$ uforge user os enable --name Windows --arch x86_64 --version Server2008R2 --
↪account root
$ uforge user os enable --name Windows --arch x86_64 --version Server2008R2 --
↪account <OTHER_USER_NAME>
```

Note: UForge does not manage Windows updates as it does for Linux. Each time you want to have an update for Windows, you will need to create and install a new set of Golden Images.

7.19.5 Windows Licences

When UForge AppCenter is installed, default license files for Windows Server are installed under `/tmp/DISTROS/USS/licenses/win_license.html`. They are default licenses which are not the official legally binding licenses and should be replaced with the appropriate licenses:

- for cloud service provider, their SPLA End User License Terms should be used;
- for in-house enterprise installations of UForge AppCenter, the enterprise license agreement document with Microsoft should be used.

In order replace the licence agreement with the correct one, you will need to do the following for each of the Windows versions (Server2008R2, Server2012, Server2012R2 and Server2016):

1. Create an html file containing the appropriate license agreement.
2. Overwrite the file on the UForge server under the directory `/tmp/DISTROS/USS/licenses`.

The following table lists the Windows Server version and the corresponding license file name:

Version	Location
Server2008R2	file:/tmp/DISTROS/USS/licenses/win_license.html
Server2012	file:/tmp/DISTROS/USS/licenses/win_license2012.html
Server2012R2	file:/tmp/DISTROS/USS/licenses/win_license2012R2.html
Server2016	file:/tmp/DISTROS/USS/licenses/win_license2016.html

7.19.6 Listing Existing Golden Images

In order to view a list of existing golden images installed on your UForge run:

```
$ uforge org golden list --arch x86_64 --version Server2008R2
```

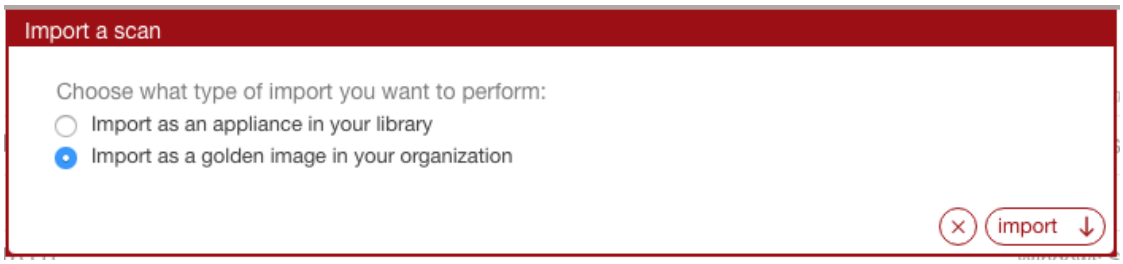
7.20 Creating a Golden Image from Scan

The easiest method to install Golden Images on your UForge AppCenter is to create a golden image from a scanned Windows instance.

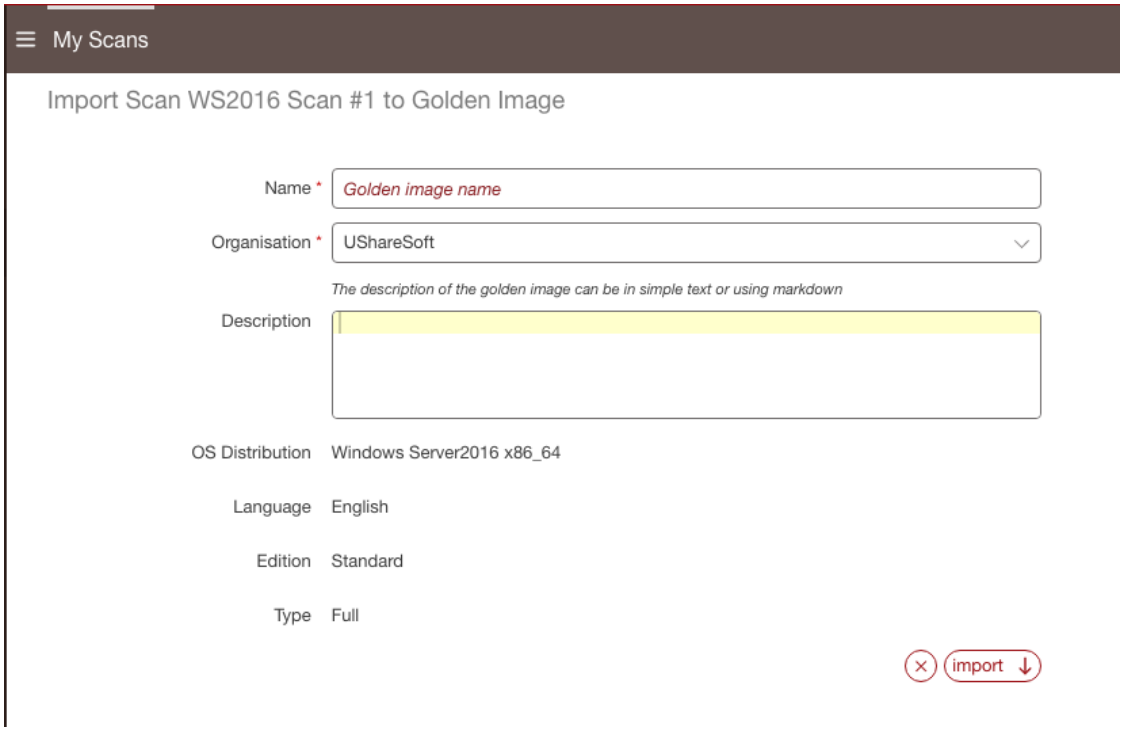
Before you create a Golden Image from scan, you must have access to a machine which has the version of Windows installed which corresponds to the version of the golden image you want to create. For example, if you want to create a golden image for Windows 2012R2, then you must scan a machine with Windows 2012R2 installed. For more information on supported versions refer to *Supported Microsoft Windows Versions*.

In order to create a Golden Image from scan:

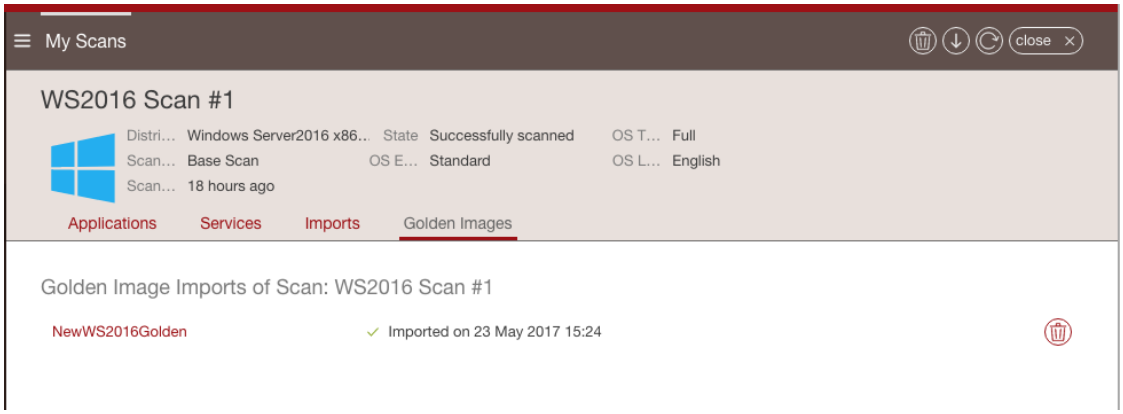
1. Install and configure Windows. For more information refer to *First Installation of Microsoft Windows*.
2. Install and configure applications.
3. Scan a Windows machine.
4. On the `Scans` page, select the scan you want to use as the base for your golden image.
5. Click on the `import` button.
6. Select `import as golden image in your organization` from the pop-up window and click `import`.



7. Enter the name and version you want to give to the golden image. You can also add a description (optional). Click `import`.



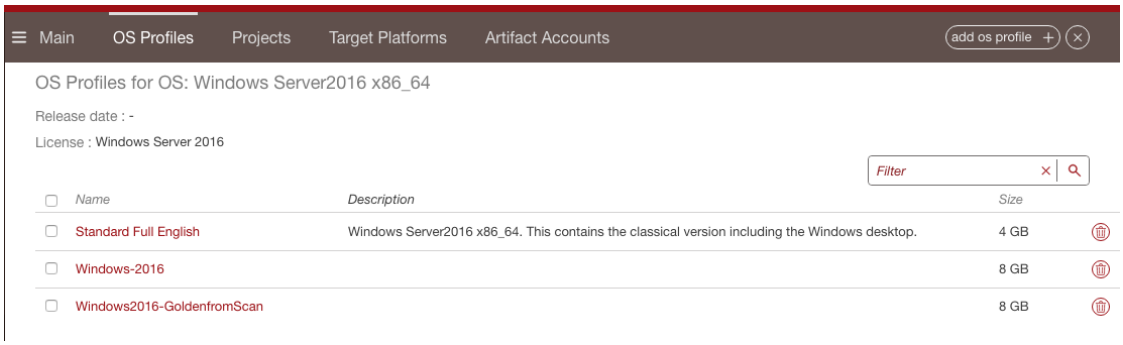
8. The progress will be shown. Once complete, the golden image will be visible under the `Golden Images` tab of the scan.



7.20.1 Deleting a Golden Image Created from Scan

To delete a golden image you created from scan:

1. Go to the `OS Profile` tab on the `Administration` page.
2. Select the operating system for which the golden image was created.
3. Click on the delete icon next to the golden image you want to remove.



Note: From the `Golden Images` tab on the `Scans` page you also have an option to delete. However, this does not delete the golden image from UForge, only the metadata. Therefore, when you want to create an appliance, the golden image will still appear under the OS Profile list.

7.21 Creating a Golden Image Manually

Note: A good knowledge of Microsoft Windows is required to create your own Golden Images manually. This method is more complex than creating a Golden Image by scanning a Windows machine and importing the scan as Golden Image. We strongly recommend you create your golden from scan. Please refer to *Creating a Golden Image from Scan*.

Warning: It is the responsibility of the UForge administrator to ensure that golden images are created by following the steps below. Not following the procedure could result in dysfunctional machine images.

Generating all the profiles available (in one language) takes roughly 4 to 7 hours depending on the machine/network performance. The Golden Image includes all updates at the moment when generated and therefore you will not need to run package updates later.

Once the Golden Image is created, you will need to

1. Save it to the proper location (as described in *Storing Golden Images on the NAS*).
2. Store the golden images (all profiles in one language) as described in *Storing Golden Images on the NAS*. You will need about 40Gb of disk space on the UForge NAS.
3. Add the Golden Image to your UForge AppCenter, as described in *Adding a Golden Image to UForge AppCenter*.

To create a new Golden Image, you will need to:

1. Ensure the following two partitions exist. These partitions are created by default during a standard Windows installation. There must be no other partitions.
   - System partition. This one is hidden, created automatically during installation of Windows Server.
   - Drive `C:`
2. We recommend that you run Windows Update to ensure that the latest updates are pre-installed in the Golden Image.
3. Optionally, you can also add the following customizations:

- Modify the registry
- Extra software installation
- User creation

4. Optionally, you can free several gigabytes of space by cleaning up Windows updates installers.

> **Warning:** After this optimization you may not be able to uninstall some of the Windows updates.

```
$ dism /online /Cleanup-Image /StartComponentCleanup /ResetBase
```

5. If you have Service Packs installed, you can free up some space by executing the following command, which will merge the Service Pack installer to the operating system.

> **Warning:** After this optimization, you will not be able to uninstall the Service Pack.

```
$ dism /online /Cleanup-Image /SPSuperseded
```

6. You can optionally perform optimizations in size for the compressed raw virtual disk image. To do so, you must:

   (a) Before the sysprep step, use the Microsoft Sysinternals tool called sdelete.exe (or sdelete64.exe) with option `-z` in a command line for all partitions, example:

   ```
   $ sdelete -z C:
   ```

   (b) After finishing the golden image (after sysprep at the last step), but before compressing the .raw with gzip or lrzip, perform the following command on the .raw virtual disk image:

   ```
   $ cp --sparse=always image.raw newimage.raw
   ```

   This will copy the image file but skip the zeros, so the .raw image will be as sparse as possible, also helping the compression program.

   ```
   $ mv -f newimage.raw image.raw
   ```

7. For Windows 2008R2, you can optionally change the password of the admin user at the first boot by creating a file as follows. Note that the admin user name may be different depending on the environment. Please replace `Administrator` in the script with the appropriate one.

```
mkdir C:\Windows\Setup\Scripts
notepad C:\Windows\Setup\Scripts\SetupComplete.cmd
---
net user Administrator /logonpasswordchg:yes
---
```

8. For Windows 2012, 2012R2 and 2016, you can optionally change the password of the admin user at the first boot by creating a file as follows. Note that the admin user name may be different depending on the environment. Please replace `Administrator` in the script with the appropriate one.

```
mkdir C:\Windows\Setup\Scripts
notepad C:\Windows\Setup\Scripts\SetupComplete.cmd
---
@echo off
```

(continues on next page)

(continued from previous page)

```
if not exist C:\etc\UShareSoft\no_console (
    net user Administrator /logonpasswordchg:yes
)
---
```

`changepasswd.bat` is specified in `Unattend.xml`. The script is launched only when the image has no console, just after `uforge-install-config` before displaying desktop.

```
notepad C:\uforge\changepasswd.bat
---
@if exist C:\etc\UShareSoft\no_console (
    @title Changing Administrator password
    echo Please provide new Administrator password.
    net user Administrator *
)
---
```

9. Open a command prompt window as an administrator and go to the `%WINDIR%\system32\sysprep` directory. Then run:

```
$ sysprep.exe /generalize /oobe /shutdown /unattend:c:\path-to-
↪sysprep\Unattend.xml
```

> **Warning:** This will shutdown the machine. Do not boot the machine again!

> **Note:** If the `unattend.xml` is not properly configured, the setup initiated by sysprep may suspend or stop during processing. In this case, you need to connect to the system through a console, not by Remote Desktop Service, in order to read setup instructions.

> **Note:** The command argument `/unattend:c:\path-to-sysprep\Unattend.xml` should not be specified when creating a Golden Image that will be used to generate Windows machine images to be published to Azure. This is because the unattend file created by Azure will be used instead.

10. You can now compress the golden images by running:

```
$ gzip image.raw
```

You can now save your golden image to the location you wish. This path will need to be specified when you add the golden images to your UForge.

7.21.1 Storing Golden Images on the NAS

Each time you create a new Golden Image manually, you need to store them in an appropriate NAS location.

> **Note:** To store the golden images (all profiles in one language) you will need about 40Gb of disk space on the UForge NAS.

You can store your golden images in the NAS location of your choice, but will need to specify the full path when adding the golden to your UForge. We recommend you store the golden images in:

```
Base dir = Windows/releases/Server2012/x86_64/
```

The path is:

```
{Language}/{Edition}/{Type}/{generation date}(YYYY-MM-DD)/
↪goldenImagePathCompressedInGz
```

Where

- {Language} is one of `English`, `French` or `Japanese`
- {Edition} corresponds to an official edition name such as `Datacenter`, `Standard`, `Enterprise` or `Webserver`
- {Type} is `Full` or `Core`

So for example:

```
Windows/releases/Server2012/x86_64/English/Standard/Core/2015-10-19/
Windows_2012_Standard_Core_2015-10-19.raw.gz
```

Note: If you plan to deploy generated Windows instances onto K5 Fujitsu Public Cloud, only "Standard" and "Enterprise" editions are supported.

For more detailed information, please refer to official Fujitsu K5 IaaS Documentation.

7.21.2 Adding a Golden Image to UForge AppCenter

Once you have created a Golden Image manually, you need to add it to your UForge AppCenter in order to be able to use the Windows version to create appliance templates. Your golden image must be in one of the following formats:

- raw.gz
- raw.zip
- raw.bz2
- raw.lrz
- vdi
- vhd
- vmdk

To add your Golden Image to UForge:

1. Copy the image to:

   ```
   $ /tmp/DISTROS/Windows/releases/<windows os version>/x86_64/<language>/
   ↪<Edition>/<Core|Full>/<YYYY-MM-DD>/golden.xxx
   ```

   For example: /tmp/DISTROS/Windows/releases/Server2012/x86_64/English/Standard/Core/2015-04-28/Windows_Server2012_English_Datacenter_Core_2015-04-28.raw.gz

   Note:

   - File and directory ownership should be `tomcat:tomcat`.

- Permissions should be readable for all users
- Disk name must be unique in the `/tmp/DISTROS/Windows` file tree

2. You must ensure that the Windows distribution exists on your UForge AppCenter. If it does not, run:

```
$ uforge org os add --name Windows --arch x86_64 --version Server2012
```

3. In order to add the new golden image to the distribution, run:

```
$ uforge org golden create --arch ARCH --edition EDITION --goldenDate␣
↪GOLDENDATE --goldenPath GOLDENPATH --language LANGUAGE --type TYPE --
↪name NAME --version VERSION --profileName PROFILENAME
```

Where the following apply :

- `--edition` Should be an official Microsoft Edition (Datacenter, Enterprise, Standard, Webserver)
- `--goldenDate` The date of the golden image (YYYY-MM-DD). If the option is not present, will be set to the date the command is run
- `--goldenPath` The full path where the golden image is stored.
- `--name` The distribution name (Windows)
- `--version` The OS version
- `--arch` The architecture
- `--profileName` The name of the profile, which will be visible in the user interface when creating a new Windows appliance. The name should be unique. If this option is not present, the name is generated automatically with the following info `EDITION TYPE LANGUAGE`. So for example: `Standard Full English`.

For example:

```
$ uforge org golden create --name Windows --arch x86_64 --version␣
↪Server2012 --edition Standard --goldenDate 2016-01-28 --language␣
↪English --type Full --goldenPath /tmp/DISTROS/Windows/releases/
↪Server2012/x86_64/WS2012.raw.gz --profileName StandardK5
```

Warning: When running `uforge org golden create` you can use the `--force` flag. This force flag will allow you to overwrite an existing golden with the same name. The `--force` flag should be used with caution as the new changes will be applied for all appliances already using this golden image.

7.21.3 Deleting a Golden Image Using the CLI

In order to delete a golden image from your UForge, run the command `org golden delete` with the following arguments:

- `--arch` : The operating system architecture (i386, x86_64).
- `--profileName` : The name of the profile to delete
- `--name` : Operating system name
- `--version` : Operating system version

For example :

```
$ uforge org golden delete --name Windows --arch x86_64 --version Server2012␣
↪--profileName Standard Full Edition
```

7.21.4 Example of Unattend File for Windows 2008R2

The following is an example of an unattend file to be used when creating a golden image for Windows 2008R2.

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
    <settings pass="oobeSystem">
        <component name="Microsoft-Windows-Shell-Setup"␣
↪processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language=
↪"neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/
↪WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
↪">
            <OOBE>
                <HideEULAPage>true</HideEULAPage>
                <NetworkLocation>Work</NetworkLocation>
                <ProtectYourPC>3</ProtectYourPC>
                <SkipUserOOBE>true</SkipUserOOBE>
            </OOBE>
            <UserAccounts>
                <AdministratorPassword>
                    <Value>Welcome@UShareSoft</Value>
                    <PlainText>true</PlainText>
                </AdministratorPassword>
            </UserAccounts>
        </component>
        <component name="Microsoft-Windows-International-Core"␣
↪processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language=
↪"neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/
↪WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
↪">
            <InputLocale>0409:00000409</InputLocale>
            <SystemLocale>en-US</SystemLocale>
            <UILanguage>en-US</UILanguage>
            <UILanguageFallback>en-US</UILanguageFallback>
            <UserLocale>en-US</UserLocale>
        </component>
    </settings>
    <settings pass="specialize">
        <component name="Microsoft-Windows-Shell-Setup"␣
↪processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language=
↪"neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/
↪WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
↪">
            <ProductKey>XXXXX-XXXXX-XXXXX-XXXXX-XXXXX</ProductKey>
            <ComputerName />
        </component>
        <component name="Microsoft-Windows-DNS-Client" processorArchitecture=
↪"amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope=
↪"nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"␣
↪xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <DNSDomain />
```

(continues on next page)

(continued from previous page)

```
            <UseDomainNameDevolution>true</UseDomainNameDevolution>
        </component>
    </settings>
    <settings pass="generalize">
        <component name="Microsoft-Windows-PnpSysprep" processorArchitecture=
↪"amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope=
↪"nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"␣
↪xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <PersistAllDeviceInstalls>false</PersistAllDeviceInstalls>
            <DoNotCleanUpNonPresentDevices>false</
↪DoNotCleanUpNonPresentDevices>
        </component>
    </settings>
</unattend>
```

Note: <ProductKey> element in the unattend file may not be mandatory. Whether the element is necessary or not depends on the type of installation media you used for the system. For example, the Volume License media does not require any <ProductKey> element in the unattend file. Please refer to Microsoft documentation for details.

Note: Elements for the locale and the language in the unattend file should have appropriate values in accordance with the language of the target OS. The following example shows the elements and their values for Japanese Windows.

```
<InputLocale>0411:00000411</InputLocale>
<SystemLocale>ja-JP</SystemLocale>
<UILanguage>ja-JP</UILanguage>
<UILanguageFallback>ja-JP</UILanguageFallback>
<UserLocale>ja-JP</UserLocale>
```

7.21.5 Example of Unattend File for Windows 2012, 2012R2, or 2016

The following is an example of an unattend file to be used when creating a golden image for Windows 2012, 2012R2 or 2016.

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
    <settings pass="oobeSystem">
        <component name="Microsoft-Windows-Shell-Setup"␣
↪processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language=
↪"neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/
↪WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
↪">
            <OOBE>
                <HideEULAPage>true</HideEULAPage>
                <NetworkLocation>Work</NetworkLocation>
                <ProtectYourPC>3</ProtectYourPC>
                <SkipUserOOBE>true</SkipUserOOBE>
            </OOBE>
            <UserAccounts>
```

(continues on next page)

(continued from previous page)

```
                <AdministratorPassword>
                    <Value>Welcome@UShareSoft</Value>
                    <PlainText>true</PlainText>
                </AdministratorPassword>
            </UserAccounts>
            <FirstLogonCommands>
                <SynchronousCommand wcm:action="add">
                    <CommandLine>c:\uforge\changepasswd.bat</CommandLine>
                    <Description>ChangeDefaultPassword</Description>
                    <Order>1</Order>
                </SynchronousCommand>
            </FirstLogonCommands>
        </component>
        <component name="Microsoft-Windows-International-Core"␣
↪processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language=
↪"neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/
↪WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
↪">
            <InputLocale>0409:00000409</InputLocale>
            <SystemLocale>en-US</SystemLocale>
            <UILanguage>en-US</UILanguage>
            <UILanguageFallback>en-US</UILanguageFallback>
            <UserLocale>en-US</UserLocale>
        </component>
    </settings>
    <settings pass="specialize">
        <component name="Microsoft-Windows-Shell-Setup"␣
↪processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language=
↪"neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/
↪WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
↪">
            <ProductKey>XXXXX-XXXXX-XXXXX-XXXXX-XXXXX</ProductKey>
            <ComputerName />
        </component>
        <component name="Microsoft-Windows-DNS-Client" processorArchitecture=
↪"amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope=
↪"nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"␣
↪xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <DNSDomain />
            <UseDomainNameDevolution>true</UseDomainNameDevolution>
        </component>
    </settings>
    <settings pass="generalize">
        <component name="Microsoft-Windows-PnpSysprep" processorArchitecture=
↪"amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope=
↪"nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"␣
↪xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <PersistAllDeviceInstalls>false</PersistAllDeviceInstalls>
            <DoNotCleanUpNonPresentDevices>false</
↪DoNotCleanUpNonPresentDevices>
        </component>
    </settings>
</unattend>
```

Note: `<ProductKey>` element in the unattend file may not be mandatory. Whether the element is necessary or not depends on the type of the installation media you used for the system. For example, the

Volume License media does not require any <ProductKey> element in the unattend file. Please refer to Microsoft documentation for details.

Note: Elements for the locale and the language in the unattend file should have appropriate values in accordance with the language of the target OS. The following example shows the elements and their values for Japanese Windows.

```
<InputLocale>0411:00000411</InputLocale>
<SystemLocale>ja-JP</SystemLocale>
<UILanguage>ja-JP</UILanguage>
<UILanguageFallback>ja-JP</UILanguageFallback>
<UserLocale>ja-JP</UserLocale>
```

7.22 Creating and Managing Categories

Categories are used to organize projects. Users can use categories in order to filter projects in the appliance library.

7.22.1 Managing Categories with CLI

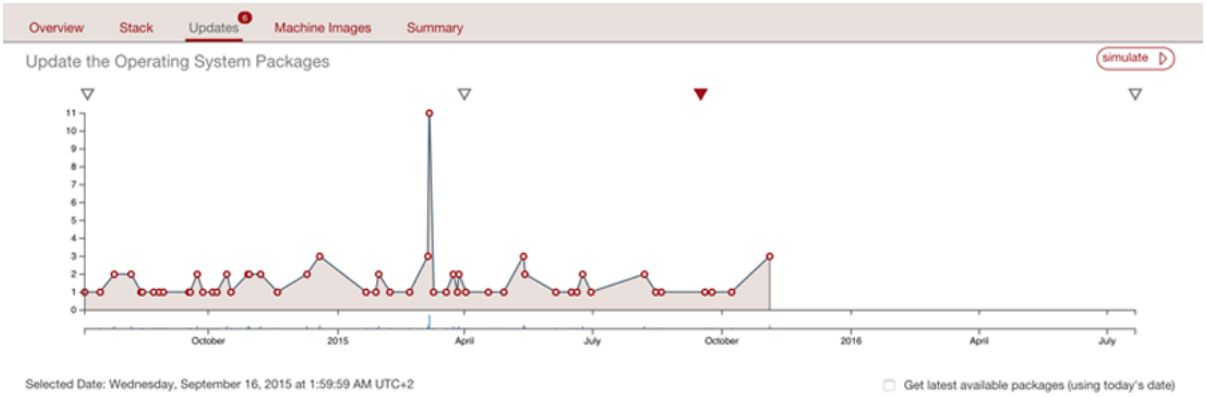
To create or manage categories with the CLI, use the command `uforge org`.

The available commands are:

- `category create`
- `category delete`
- `category list`

7.23 Creating and Managing Milestones

Milestones are used as a marker for a specific event, such as beta or GA for example. They appear on the GUI under OS package updates.



In the figure above, the triangle shape indicates a Milestone. For CentOS, this is the versions (6.1, 6.2 etc). For CentOS, Debian and Red Hat, the major versions are automatically marked as Milestones when the distribution is added to the platform.

Milestones can be created by the UForge administrator using the command line interface, in order to mark a specific date or version, for example for a beta version or for a security fix.

To create or manage milestones using the CLI, use the command `uforge os`.

The available commands are:

- milestone add
- milestone list
- milestone modify
- milestone remove

In order to create a milestone called "beta" for Debian, run:

```
# uforge os milestone add --dname Debian --darch i386 --dversion 6 --date 2014-01-01␣
↪12:00:00 --name beta -u $ADMIN -p $PASS
```

CHAPTER 8

Managing Users

The following sections cover information regarding managing the UForge users. Managing users is intrinsically linked to role based access control (RBAC). Please make sure to read the information on role-based access control in *RBAC Overview*.

8.1 Managing User Accounts

When managing user accounts, the administrator must have administration privileges for the organization the user is part of or where the member will be created.

8.1.1 Creating User Accounts with CLI

When a user account is created, the account is added to the specified organization. If no organization is specified, the UForge default organization is used. When creating a user, the administrator must specify a subscription code. This subscription code defines the roles, OSes, formats the user has access to, in addition to any quotas.

Note: Since a subscription code is required to create a user, the UForge administrator must first create these subscription profiles, and add the organization administrator (or another user) to the admin list.

To create a user account use `uforge user create` with the arguments as follows:

```
# uforge user create --account kermit --code <subscriptionProfileCode> --email␣
↪kermit@usharesoft.com --url https://uforge.usharesoft.com:443 -u $ADMIN -p $PASS
```

As no default password is specified, UForge creates a temporary random password for the new user account. An email is automatically sent to the email address assigned to the user account when the account has been successfully created. The user account is automatically active, allowing the user to sign in and begin using UForge.

The `create` command provides a set of optional arguments allowing the administrator to control certain aspects of the account being created.

8.1.2 Disabling a User Account

Once disabled, the user no longer has access to the UForge platform.

To disable a user account use `uforge user disable` and specify the user account:

```
# uforge user disable --account kermit --url https://uforge.usharesoft.com:443 -u
↪$ADMIN -p $PASS
Getting user [kermit] ...
Success: User kermit has been de-activated
```

8.2 RBAC Overview

Access to the various services of the UForge platform is enforced by roles. Each role contains one or more entitlements that define the access to one or more services or permissions for various actions.

8.2.1 Viewing Default Roles

A set of default roles are provided by UForge. To view these default roles, use the command `uforge role list`:

```
$ uforge role list -u $ADMIN -p $PASS
Getting all the roles for the default organization ...
+--------------------------+-------------------------------------------------------------
↪-------------------------+
| Name                      | Description                                                 ␣
↪                         |
+--------------------------+-------------------------------------------------------------
↪-------------------------+
| admin                     | Provides organization administration rights.  Includes␣
↪managing the organizations project catalog; os profiles and |
|                           | user accounts.                                              ␣
↪                         |
+ Entitlements ------------------+-------------------------------------------------------
↪-------------------------+
| org_administrate          | Access to manage the entitlements list available for an␣
↪organization.            |
| org_formats_administrate  | Access to manage the generation formats available for an␣
↪organization.            |
| org_members_administrate  | Access to manage user accounts for an organization.         ␣
↪                         |
| org_os_administrate       | Access to manage the operating systems available for an␣
↪organization.            |
| org_os_profiles_administrate| Access to manage the os profiles available for an␣
↪organization.              |
| org_projects_administrate| Access to manage the project catalog for an organization.
↪                         |
| user_create               | Access to create user accounts in UForge.                   ␣
↪                         |
+--------------------------+-------------------------------------------------------------
↪-------------------------+
Found 15 role(s)
| apiuser                   | Allows the user to communicate with UForge via the APIs.␣
↪                         |
```

(continues on next page)

(continued from previous page)

```
+ Entitlements ------------+-------------------------------------------------------------
↪------------------------+
| api_key_access            | Access to UForge APIs.                                      ␣
↪                          |.... rest omitted for clarity
```

Organization administrators can create and manage roles as well as assign these roles to the members of the organization. For a list of all the commands possible to manage RBAC, use `uforge role --help`.

To view the entitlements associated to a role run the command `uforge role info --name <roleName>`.

Note: Adding a specific role to a user (such as Administrator) is often only the first step in granting specific rights. You must then specify the entity (for an Administrator this would be the organization) for which the user has the newly assigned rights. See the examples later in this section.

Listing Entitlements

An entitlement describes the right to perform an action on the UForge platform. These are pre-defined by UForge. Entitlements are added to roles to describe the access rights to the various UForge features.

To view all the entitlements provided by UForge, use the command:

```
$ uforge entitlement list -u $ADMIN -p $PASS
```

Listing Roles for an Organization

To view the available roles for an organization, run the command:

```
$ uforge role list --org usharesoft -u $ADMIN -p $PASS
```

If no organization is provided, then the default organization is used.

8.2.2 Creating and Updating User Roles

If the pre-defined roles delivered with UForge do not match with the permissions you want to allow users, you can either create a new role or modify an existing one.

To create a new role in the default organization:

```
$ uforge role create --name newrole --description "description of new role" -u $ADMIN␣
↪-p $PASS
```

By default, this role will be empty (containing no entitlements). You can then add entitlements to a role by using the command:

```
$ uforge role entitlement add --name newrole --entitlements entitlement-name -u
↪$ADMIN -p $PASS
```

You can also remove an entitlement from a role by using the command:

```
$ uforge role entitlement remove --name newrole --entitlements entitlement-name -u
↪$ADMIN -p $PASS
```

You can add entitlements as part of the creation by using the `--entitlements` option in the create command:

```
$ uforge role create --name newrole --description "description of new role" --
↪entitlements entitlement-name studio_access -u $ADMIN -p $PASS
```

Note: You cannot modify the "root" role.

8.2.3 Listing Roles Assigned to a User

To view the roles already assigned to a specific user, run the command:

```
$ uforge user role list --account <username> -u $ADMIN -p $PASS
```

8.2.4 Adding Roles to a Subscription Profile

If you want a group of users to have the same role, then you can add it to the subscription profile (refer to *Managing Subscription Profiles*). This means that all the users that are created with this subscription profile will have this role. However, this will apply only to the new users created, unless you use the option `--allusers`.

```
$ uforge subscription role add --name sub --roles newrole --allusers -u $ADMIN -p
↪$PASS
```

In the example above `sub` is the name of the subscription profile.

8.2.5 Adding a Role to a User

Note: Adding a specific role to a user (such as Administrator) is often only the first step in granting specific rights. You must then specify the entity (for an Administrator this would be the organization) for which the user has the newly assigned rights. See the examples later in this section.

To add a role to a user:

```
$ uforge user role add --account <username> --roles newrole -u $ADMIN -p $PASS
```

To add several roles to the same user:

```
$ uforge user role add --account <username> --roles role1 role2 -u $ADMIN -p $PASS
```

8.2.6 Deleting a Role from a User

To delete a role from a user:

```
$ uforge role delete --account <username> --roles roleA -u $ADMIN -p $PASS
```

Note: If this is the only role assigned to a user, once deleted, the user will no longer have any roles. With no roles the user will only be able to view the UForge account and dashboard.

8.2.7 Managing Subscription Profiles

A subscription profile defines the roles, OSes, formats the user has access to, in addition to any quotas. Since a subscription code is required to create a user, the UForge administrator must:

1. Create subscription profiles
2. Add the organization administrator to the admin list.

To create a subscription profile you must be the UForge administrator.

The subscription command allows you to create, manage and delete user accounts in UForge. Each user command can have mandatory and optional arguments to complete the request.

The subscription target has the following commands:

```
================================================================================================================
Subscription_Cmd help
================================================================================================================
admin                          | Manage subscription profile admins
create                         | Create a new subscription profile within an␣
↪organization.
delete                         | Delete a subscription profile from an organization.
disable                        | Activates or enables a subscription profile within an␣
↪organization.
enable                         | Activates or enables a subscription profile within an␣
↪organization.
help                           | List available commands with "help" or detailed help␣
↪with "help cmd".
info                           | Get detailed information on a subscription profile␣
↪within an organization.
list                           | List all the subscription profiles for a given␣
↪organization. If no
os                             | Manage subscription profile formats
quota                          | Manage subscription profile quotas
role                           | Manage subscription profile roles
targetformat                   | Manage subscription profile target formats
targetplatform                 | Manage subscription profile target platforms
update                         | Updates an existing subscription profile.
```

Creating and Enabling Subscription Profiles

To create a subscription profile you must be a subscription profile administrator. In order to become a subscription profile administrator see *Allowing Administrators to Create Users*.

There are two steps to adding subscription profiles:

1. creating the profile
2. enabling the profile

By default, the subscription profiles you create are not active. You can activate the subscription profile when you create it by using the argument `--active`. Otherwise, you will need to activate it using the command `subscription enable`.

You can add the list of administrators that will be allowed to create users by using the argument `--admin`. If you do not want to define the administrators that will create the users at the same time as you create the subscription profile, you can do so later. See *Allowing Administrators to Create Users*.

When creating a subscription profile, you can also specify the following:

- formats (using targetformat and targetplatform)
- OS
- quotas
- roles

If you specify any of the above when you create a subscription profile, then all users created using this subscription profile will have the formats, OSes, quotas and roles defined in the subscription profile.

Note: Once you create a user with a specific subscription profile, even if you modify the subscription profile, the rights of the users already created will not be modified. For example, if profileA used to create UserA has quota set to unlimited. Once UserA is created, you modify the profileA to set quota to 3 generations. UserA will still have quota set to unlimited, but UserB created with the updated profileA will have quota set to 3 generations.

In order to force the changes to apply to all users (even those already created), use the option `--allusers`. This option can be used for roles, OS and format management in subscription profile. It cannot be used for quota.

1. To create a subscription profile for an organization, run the command:

```
$ uforge subscription create --code <string> --name <string> --org␣
↪usharesoft --active -u $ADMIN -p $PASS
```

   The code can be any alphanumeric string, excluding spaces and special characters.

2. To enable a subscription profile for an organization, run the command:

```
$ uforge subscription enable --name <string> --org usharesoft -u $ADMIN -p $PASS
```

Allowing Administrators to Create Users

In order to create a user, the following are required:

- The organization Administrator needs to be part of the list of subscription profile administrator. Only users that are part of this list can create user accounts.
- Subscription profile code. This code must be part of the request to create a user. Only the UForge administrator can create these codes.

When creating subscription profiles, the UForge administrator can add subscription profile administrators. However, they can also be added after the fact, as follows:

```
$ uforge subscription admin add --admin kermit --name profileA --org usharesoft -u
↪$ADMIN -p $PASS
```

The argument `--admin` is the login of the user you want to add as an administrator. This administrator will be able to create users with the subscription profile specified by the argument `--name`.

Disabling a Subscription Profile

If you no longer want a subscription profile to be used when creating new users, you can either delete or disable the subscription profile. However, we recommend that you simply disable the subscription profile, in order to keep a history of the profile. Regardless of whether you delete or disable the subscription profile, the users created with the associated subscription code will not be deleted or deactivated.

To disable a subscription profile:

```
$ uforge subscription disable --name profileA -u $ADMIN -p $PASS
```

If no org is specified, the default organization is used.

8.3 Granting a User Administrator Rights

Note: The user you set as Administrator will have access to the Admin tab on UForge Portal and have rights to manage the organization. However, an Admin user does not automatically have the right to create users. In order to create new users the user must be part of the Admin list. See *Allowing Administrators to Create Users*.

There are two steps in granting a user Administrator privileges; you must:

1. Give the user the administrator role. This only indicates that the user CAN be an administrator, but does not specify for which organizations:

```
$ uforge user role add --account <username> --roles admin -u $ADMIN -p $PASS
```

2. Assign the user as administrator to a specific organization. If no organization is provided, then the default organization is used:

```
$ uforge user admin promote --account <username> --org <org name> -u $ADMIN -p
↪$PASS
```

8.4 Setting Quotas

You can set a number of quotas for a user account using the command: `uforge user quota modify`.

Setting quotas allows you to limit the user's access to UForge based on one (or several) of the following criteria:

- number of appliances (includes imports and scans)
- number of images generated (includes all image generations)
- diskusage in bytes (includes storage of mysoftware uploads, bootscripts, image generations, scans)
- number of scans for migration (includes initial scan and incremental scans)

Note: You can set the size of the Scan Overlay. This is done not through the CLI but using the uforge.conf file. This is described in *Setting the Overlay Limit*.

You can set the quotas to refresh once a month using the argument `--frequency`. You can set the frequency to:

- `monthly`: the quota counter will be reset every month. The day of the reset is based on the date of the user creation (and not the date when the limit is set).
- `none`: once the quota is reached it will not be reset automatically (it can however be increased manually).

8.4.1 Viewing the Quotas for a User

You can see the quotas set for a given user as follows:

```
$ uforge user quota list --account <user> -u $ADMIN -p $PASS
```

In the example above, the argument `--user` is the account of the administrator. The argument `--account` is the user name of the account you want to view the quotas for.

Typically, when no limits are set, the results should be:

```
$ uforge user quota list --account kermit -u $ADMIN -p $PASS
Getting quotas for [kermit] ...
List of quotas available for [kermit] :
Scan (0)         --------------------UNLIMITED---------------------
Template (0)     --------------------UNLIMITED---------------------
Generation (0)   --------------------UNLIMITED---------------------
Disk usage (0B) --------------------UNLIMITED---------------------
```

8.4.2 Setting a Quota for Appliance

You can set a limit to the number of appliances a given user can have. This limit can be reset monthly. Note that if the user deletes an appliance, the count will go down. For example, if the user has reached the set limit of 10 appliances, the user can delete an appliance in order to create a new one and stay within the quota limit.

- The option `--type` must be set to `appliance`.
- The option `--limit` determines the quota.

For example to set the quota of appliances to 10 per month:

```
$ uforge user quota modify --user $ADMIN --password $PASS --account kermit --type␣
↪appliance --limit 10 --frequency monthly
```

In the example above, the argument `--user` is the account of the administrator. The argument `--account` is the user name of the account you want to view the quotas for.

8.4.3 Setting a Quota for Image Generations

You can set a limit to the number of images a given user can generate. This limit can be reset monthly.

- The option `--type` must be set to `generation`.
- The option `--limit` determines the quota.

For example to set the quota of images a user can generate to 10 per month:

```
$ uforge user quota modify --user $ADMIN --password $PASS --account kermit --type␣
↪generation --limit 10 --frequency monthly
```

8.4.4 Setting a Quota for Migration

You can set a limit to the number of scans a given user can run. This quota includes both scan generation and scan appliance generation. This limit can be reset monthly.

- The option `--type` must be set to `scan`.

- The option `--limit` determines the quota.

For example to set the number of scans the user can run to 5 per month:

```
$ uforge user quota modify --user $ADMIN --password $PASS --account kermit --type␣
↪scan --limit 5 --frequency monthly
```

8.4.5 Setting the Overlay Limit

You can set the maximum size of the scan overlay. The overlay includes all the files in MySoftware and all other non-native files. This limit is set in the `uforge.conf` file. You must add the parameter:

```
UFORGE_SCAN_OVERLAY_MAX_SIZE = maximum size in octets
```

For example, to set the limit to 10GB (10 x 1024 x 1024 x 1024):

```
UFORGE_SCAN_OVERLAY_MAX_SIZE=10737418240
```

8.4.6 Setting a Quota for Disk Usage

You can set a limit to the disk space a user can use. Disk space usage includes: MySoftware uploads, bootscripts, images generations, scans etc.

- The option `--type` must be set to `diskusage`
- The option `--limit` determines the quota in bytes. For disk usage, the quota is expressed in bytes.

For example to set the disk space quota a user can use to 10Gb per month:

```
$ uforge user quota modify --user $ADMIN --account user --type diskusage --limit␣
↪10737418240 --password $PASS
```

The results should be:

```
$ uforge user quota list --user $ADMIN --account <username> --password $PASS
Getting user [user] ...

+------------------+--------------+------------+------------+--------------------------
↪------+
| Type             |     Consumed |      Limit | Frequency  | Renewal date             ␣
↪      |
+------------------+--------------+------------+------------+--------------------------
↪------+
| appliance        |            1 |  unlimited |            | -                        ␣
↪      |
| diskusage        |          0.0 |      10 GB |            | -                        ␣
↪      |
| generation       |            0 |  unlimited |            | -                        ␣
↪      |
| scan             |            0 |  unlimited |            | -                        ␣
↪      |
+------------------+--------------+------------+------------+--------------------------
↪------+
Found 4 formats
```

8.4.7 Resetting Quotas

If you want to remove a quota set on a user, you can do this using the `--unlimited` flag. For example, to remove a quota limit you might have set on the number of scans for a user, run:

```
$ uforge user quota modify --user $ADMIN --account user --type scan --unlimited --
↪password $PASS
```

8.5 Managing User Access to Operating Systems

Each user account can be configured to have access to certain operating systems and image formats that are enabled in the organization. When a user account is created, the organization's default operating systems and image formats are automatically added, thanks to the subscription profile (refer to *Managing Subscription Profiles*). The administrator can then add or remove operating systems and image formats for a specific user account using the command-line interface.

8.5.1 Listing Available OSes

To list all the operating systems the user has access to:

```
$ uforge user os list --account <username> -u $ADMIN -p $PASS
```

8.5.2 Allowing Access to OSes Using Subscription Profile

An administrator can modify access to OSes for a group of users using the subscription profile option `--allusers`. For example if a group of users with a specific subscription profile is created but you decide you want to modify the operating systems available, then you can modify the subscription profile. In order to force the changes to apply to all users (even those already created), use the option `--allusers`.

```
$ uforge subscription os add --name sub --os CentOS --version 6 --arch x86_64 --
↪allusers -u $ADMIN -p $PASS
```

In the example above `sub` is the name of the subscription profile.

8.5.3 Adding or Removing Access to OSes for a Specific User

An administrator can add or remove access to operating systems for a specific user.

- Adding/removing all CentOS versions:

```
$ uforge user os enable --account <username> --name CentOS
$ uforge user os disable --account <username> --name CentOS
```

- Adding/removing CentOS version 5.8 all architectures:

```
$ uforge user os enable --account <username> --name CentOS --version 5.8
$ uforge user os disable --account <username> --name CentOS --version 5.8
```

- Adding/removing CentOS version 5.8 i386:

```
$ uforge user os enable --account <username> --name CentOS --version 5.8 --arch␣
↪i386
$ uforge user os disable --account <username> --name CentOS --version 5.8 --arch␣
↪i386
```

Note: For Scientific Linux and Red Hat Enterprise Linux, use the following syntax: `Scientific.*` and `RedHat.*` for the distribution name.

8.6 Managing User Access to Formats

Image formats can be managed at the level of the organization or of the user.

An administrator who wants to add format rights to several users within an organization should create the formats at the level of the organization (refer to *Adding Formats to an Organization Using the CLI*) and then add them to a subscription profile (refer to *Adding Formats Using Subscription Profile*). In this case, all new users created with the given subscription profile will have access to the formats.

8.6.1 Adding Formats Using Subscription Profile

To add access to a format to a group of users, you can add it as part of a subscription profile. This means that all the users that are created with this subscription profile will have access to the format. You cannot add access to a format that is not included in the organization. In order to force the changes to apply to all users (even those already created), use the option `--allusers`. For a list of formats that are part of the organization, use the command:

- `uforge org targetformat list`
- `uforge org targetplatform list`

Therefore, in order to add formats using the subscription profile, run the following command. For example:

```
$ uforge subscription targetformat add --targetformat ovf qcow2 vbox --allusers --
↪name sub --url https://uforge.usharesoft.com:443 -u $ADMIN -p $PASS
```

In the example above, the argument `--name` is the name of the subscription profile.

8.6.2 Adding Formats to a Specific User

To add access to a format for a specific user, you must follow these steps:

1. Ensure the target format exists for the organization. For a list of formats that are part of the organization, use the command `uforge org targetformat list`.
2. Enable the format for the user using `uforge user targetFormat enable`.

8.6.3 Enabling Access to a Format

Note: When enabling or disabling a format, you must use the "name" of the format and not the "format".

For example, if you are trying to enable the following format:

```
+----+---------------+--------------+----------+-------+-----------------+--------+
| Id |     Name      |    Format    | Category | Type  | CredAccountType | Access |
+====+===============+==============+==========+=======+=================+========+
| 34 | OpenStack VHD | openstackvhd | Cloud    | cloud | openstack       |        |
+----+---------------+--------------+----------+-------+-----------------+--------+
```

You would use the following command:

```
$ uforge user targetformat enable --targetFormats "OpenStack VHD" --account kermit --
↪url https://uforge.usharesoft.com:443 -u $ADMIN -p $PASS
```

8.6.4 Disabling User Access to Formats

To disable access to one or more formats for a specific user (in this example "kermit"), you must specify the format name with option `--targetFormats`, as follows:

```
$ uforge user targetformat disable --targetFormats "OVF or OVA" QCOW2 VirtualBox --
↪account kermit --url https://uforge.usharesoft.com:443 -u $ADMIN -p $PASS
```

8.7 Granting a User API Access

Each user account can be configured to have access to the UForge APIs. This allows the user to automate the creation of appliances.

To use UForge APIs, an API key pair (public and secret keys) must be used as part of the communication. When a user account is enabled, an API key pair is created automatically. The administrator may also grant the user to have more than one set of API key pairs. This allows the user to create and manage multiple API key pairs.

To grant API access to a user account you must create a role to which you will assign the entitlement `api_key_access`.

1. Create the new role:

```
$ uforge role create --name newrole --entitlement api_key_access -u
↪$ADMIN -p $PASS
```

2. Assign this new role to the user:

```
$ uforge user role add --roles newrole --account kermit -u $ADMIN -p
↪$PASS
```

3. Optionally you can set the number of API keys:

```
$ uforge user api quota --apimax 5 --account kermit -u $ADMIN -p $PASS
Getting user [kermit] ...
Success: User kermit now has an API Key Quota of [5]
```

To disable API access simply remove the entitlement `api_key_access`:

```
$ uforge user role remove --name newrole --account kermit -u $ADMIN -p $PASS
```

8.8 Granting a User Supervisor Rights

If you want to allow a user to login to UForge as another user, you must grant the user supervisor rights. To do so, use the entitlement `supervisor_access`. You can either:

- add the entitlement to an existing role (note this will give supervisor rights to ALL the users with that role)
- create a new role with the supervisor access.

Warning: Users with Supervisor Access will be able to log in as ANY of the users in the organization without entering a password. This right should be limited to support or managed services. Users with Supervisor Role need to respect the privacy of the user data, according to current legislation.

It is probably safer to create a new "supervisor" role and add the supervisor rights, to limit the number of users that can access all the user accounts in the organization.

1. To create a new role, refer to *Creating and Updating User Roles*.
2. Add this new supervisor role to the user who will be acting as "supervisor"; see *Adding a Role to a User*.
3. If the user is not already an administrator, you will need to promote the user as an administrator of the organization for which supervisor rights are assigned:

```
$ uforge user admin promote --account kermit --org MyOrg -u $ADMIN -p
↪$PASS
```

If no organization is provided, then the default organization is used. If you prefer to add supervisor access to an existing role (in this case "admin") run the following command:

```
$ uforge role entitlement add --name admin --entitlements supervisor_access -
↪u $ADMIN -p $PASS
```

CHAPTER 9

Monitoring Overview

The following sections cover some elements to monitor and troubleshoot your UForge Platform.

9.1 Viewing the Web Service Logs

All the web service logs can be found in the domain directory of the Tomcat application server. The web service uses the log4j logger. You can change the log level of the web service resources for debugging purposes.

To view the logs:

Log in to the web service node as root:

```
$ ssh root@<ip address of the node>
$ cd /var/log/tomcat/
$ tail -f catalina-daemon.out
```

To change the log level of the web service:

1. Log in to the web service node as root

   ```
   $ ssh root@<ip address of the node>
   $ cd /opt/Tomcat/webapps/ufws/WEB-INF/classes
   $ vi log4j.properties
   ```

2. Update the logging level for each resource you wish by using the following keywords: `info|warn|debug`

3. Force Tomcat to reload

   ```
   $ touch /opt/Tomcat/webapps/ufws/.reload
   ```

4. Restart the Web Service

   ```
   $ service tomcat restart
   ```

9.2 Viewing Current Jobs in the Scheduler

To view the scheduler's current queue, log in to the oar scheduler node as root and run the command oarstat:

```
$ oarstat
Job id     Name           User           Submission Date     S Queue
---------- -------------- -------------- ------------------- - ----------
5725       4825           glassfish      2012-05-01 18:53:32 R default
```

This provides information on the jobs currently being executed as well as the jobs that are scheduled to be executed once a resource is free.

To view more details of a specific job, log in to the oar scheduler node as root and run the command oarstat and specify the `JOB_ID`:

```
$ oarstat --job 5725 --full
Job_Id: 5725
    job_array_id = 5725
    job_array_index = 1
    name = 4825
    project = [% 62, Installing distribution]
    owner = glassfish
    state = Running
    wanted_resources = -l "{type = 'default'}/resource_id=1,walltime=2:0:0"
    types =
    dependencies =
    assigned_resources = 51
    assigned_hostnames = vm
    queue = default
    command = /tmp/USER_DATA/FactoryContainer/images/4825/oar/ISO_4825.sh
    launchingDirectory = /tmp/USER_DATA/FactoryContainer/images/4825/oar
    jobType = PASSIVE
    properties = (nature=6) AND desktop_computing = 'NO'
    reservation = None
    walltime = 2:0:0
    submissionTime = 2012-05-01 18:53:32
    startTime = 2012-05-01 18:53:33
    cpuset_name = glassfish_5725
    initial_request = oarsub -d /tmp/USER_DATA/FactoryContainer/images/4825/
↪oar -E oar_image_job4825.stderr -O oar_image_job4825.stdout -n 4825 --
↪project null --checkpoint=1 --signal=15 -p nature=6 /tmp/USER_DATA/
↪FactoryContainer/images/4825/oar/ISO_4825.sh
    message = FIFO scheduling OK
    scheduledStart = 2012-05-01 18:53:33
    resubmit_job_id = 0
    events =
```

9.3 Viewing the Logs of a Job

The main logs of OAR are stored in: `/var/log/oar.log` Each job launched on the OAR cluster, whether it be an image generation or publish to a cloud, logs are stored for the job. These include all the traces and error information during the execution of the job. Each job has a unique ID provided to it, which can be recuperated using the `oarstat` command as shown in *Viewing Current Jobs in the Scheduler*.

For jobs that generate an image, the log files are stored under: `cd <user data mount point>/FactoryContainer/images/<job_name>`

For example:

```
$ oarstat
Job id     Name           User           Submission Date     S Queue
---------- -------------- -------------- ------------------- - ----------
5725       4825           glassfish      2012-05-01 18:53:32 R default
```

The directory will be:

```
$ cd <user data mount point>/FactoryContainer/images/4825
```

Logs for jobs that publish an image to a specific cloud are stored in a sub-directory of the generated image directory. So for example if a user generates an Amazon image, then publishes the machine image to Amazon, the directory structure created is:

```
/<user data mount point>/FactoryContainer/images
                 |
                 |
                 |-- generated image logs dir --> 4825
                                                  |
                                                  |
                                                  |-- published image logs dir -->␣
↪publish_<job_name>
```

To view the logs of a job, log in to the oar scheduler node as root:

```
$ cd /<user data mount point>/FactoryContainer/images/<job_name>/oar
```

so for example

```
$ cd /tmp/USER_DATA/FactoryContainer/images/4825/oar
$ ls -al
total 376
drwxr-xr-x 2 glassfish glassfish   4096 Apr 30 18:21 .
drwxr-xr-x 6 glassfish glassfish   4096 Apr 30 18:22 ..
-rwxr-xr-x 1 glassfish glassfish    980 Apr 30 18:15 ISO_4825.sh
-rwxr-xr-x 1 glassfish glassfish   1088 Apr 30 18:15 cmd_4825.sh
-rwxrwxrwx 1 glassfish glassfish    300 Apr 30 18:18 oar_image_job4825.stderr
-rwxrwxrwx 1 glassfish glassfish 360500 Apr 30 18:21 oar_image_job4825.stdout
```

To check for suspicious jobs you can run:

```
$ oarnodes | grep -i suspected
```

9.4 Configuring UForge to Send Logs to Elastic Stack

It is possible to visualize UForge logs in a dedicated tool: an Elastic Stack server (ElasticSearch, Logstash, Kibana). Elastic Stack centralizes all the logs and offers the possibility to navigate and search through the logs. It allows you to analyze and visualize the logs. If you have an Elastic Stack server installed, you can configure UForge so that UForge logs are sent to your Elastic Stack server.

UForge logs retrieval can be configured as follows:

> **Note:** The following steps need to be done on each node of your platform.

1. Edit the file `/etc/UShareSoft/uforge/uforge.conf`. Change the value of `FILEBEAT_SERVICE_ENABLED` to `true`.
2. Edit the file `/etc/filebeat/filebeat.yml` and put the IP and port of your Elastic Stack server in the section:

```
output.logstash:
  hosts: ["#LOGSTASH_HOST_MARKER:#LOGSTASH_PORT_MARKER"]
```

   By default, the Logstash port is 5044 (which may be different in your Logstash installation).

3. Launch the following script on every node:

```
/opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

Your UForge will now automatically send logs to your Elastic Stack server.

To update the IP or port of the Elastic Stack server, edit the file `/etc/filebeat/filebeat.yml`. Change the value, and restart filebeat service on every node using the command `systemctl restart filebeat`.

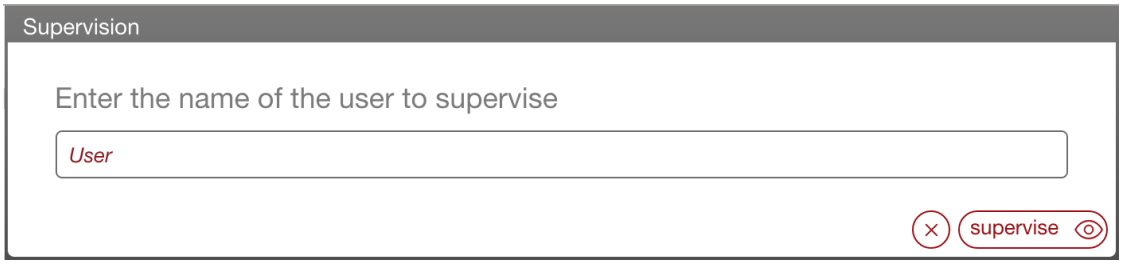
9.5 Using Supervisor Mode

UForge allows you to login as another user if you have supervisor access. Supervisor access rights are assigned by the UForge administrator. Once you have supervisor rights, you will see the eye icon in the top right of the UForge GUI banner.

Warning: Users with Supervisor Access will be able to log in as ANY of the users in the organization without entering a password. This right should be limited to support or managed services. Users with Supervisor Role needs to respect the privacy of the user data, according to current legislation.

To log in as another user with supervisor mode:

1. Click on the eye icon (`supervision`) in the top right of the UForge banner.
2. In the login screen, enter the name of the user you want to log in as. You will note that you do not need to enter a password.



9.5.1 UForge REST API in Supervisor Mode

The syntax for using basic authentication in supervisor mode is:

```
Authorization:Basic supervisorusername\supervisedusername:supervisorpassword
```

9.6 Getting Support

If you require support for your UForge platform, please contact:

support [@] usharesoft [dot] com

In order to help us understand your issue. Please include any logs pertinent to the issue you are having.

To help gather information of the UForge platform, we have included a script you can run on the UForge platform instances.

To run the script as superuser:

```
$ /opt/UShareSoft/uforge/tools/support/machine_infos.sh
```

The script recuperates the logs from the main services that are running on the instance where the script is launched and creates an archive in `/tmp`, with the name starting `uforge-support-`. This archive can be sent to our support team as part of your support request.

CHAPTER 10

UForge Tooling

To manage the UForge AppCenter, there are a number of tools available:

- The graphical user interface
- The command line interface
- Hammr (refer to Hammr documentation)
- The APIs (refer to API documentation)

This section covers:

10.1 Using the CLI Tool

The UForge platform has a command-line tool called `uforge` to administer the platform. The CLI has the following usage:

```
uforge -u <name> -p <password> -U <URL> <target> <command> [options][args]
```

where the flags:

- `-u`: provides the username of the administrator
- `-p`: provides the password of the administrator
- `-U`: provides the UForge URL endpoint to communicate with the platform

If the `target`, `command` and arguments are ommitted, then you enter into an interactive mode e.g.:

```
$ uforge -u <user> -p <password> -U https://uforge.usharesoft.com/api
```

The following targets are available for the `uforge` command-line tool:

- `entitlement` : Administration of all the entitlements in UForge for RBAC
- `images` : Administer generated images for a user

- `org` : Organization administration (list/info/create/update/delete etc)
- `os` : Administer operating systems/distributions (list/info etc)
- `pimages` : Administer published images for a user
- `role` : Manage platform roles
- `subscription` : Manage subscription profiles : list profile, create profiles, update profiles
- `template` : Administer templates for a user (list/info/create/delete/generate/share etc)
- `user` : User's administration (list/info/create/update/delete etc)

To list the command for each target, use the `-h` or `--help` option. For example:

```
$ uforge -u root -p mypassword -U https://uforge.usharesoft.com/api user --help
```

The arguments are:

- `-U URL`, `--url URL` : the server URL endpoint to use
- `-u USER`, `--user USER` : the user name used to authenticate to the server
- `-p PASSWORD`, `--password PASSWORD` : the password used to authenticate to the server
- `-v` : displays the current version of the uforge-cli tool
- `-h`, `--help` : show this help message and exit
- `-k PUBLICKEY`, `--publickey PUBLICKEY` : public API key to use for this request. Default: no default
- `-s SECRETKEY`, `--secretkey SECRETKEY` : secret API key to use for this request. Default: no default
- `-c`, `--no-check-certificate` : Don't check the server certificate against the available certificate authorities

CHAPTER 11

Rebranding Your UForge GUI

The following sections cover information regarding rebranding elements of the UForge GUI. Most customizations are done using the `config.xml` and `forge-config.xml` files located in: `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates`. This is a relative path.

Once you have completed your changes, you will need to run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

Note: Other than the sections indicated in this document, we recommend that you do not modify other elements of the configuration files, otherwise this may cause issues when using UForge Portal.

11.1 Creating Dedicated Image Directory

If you plan to include your own logos and images when customizing Portal, you should create a sub-directory under: `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/images`

For example, place all your logos and images under: `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/images/myCompany`

In all of the following sections, if you update logos and images, use your new path. For example:

```
<c:signInLogoUrl>images/signInLogo.png</c:signInLogoUrl>
```

Must be changed to:

```
<c:signInLogoUrl>images/myCompany/signInLogo.png</c:signInLogoUrl>
```

11.2 Modifying the Sign-In and Sign-Up Page

You can modify a few elements on the sign-in and sign-up pages.

The following figure shows the default Sign-in page.

11.2.1 Modifying the Logo

You can modify the logo that appears on the sign-in and sign-up page in section `<c:client>`:

```
<c:signInLogoUrl>images/common/fujitsu.svg</c:signInLogoUrl>
```

You can also modify the text that appears when you scroll over the logo, as well as the link the user is directed to when clicking on the link in section `<c:client>`:

```
<c:signInLogoTitle>Go to Fujitsu.com</c:signInLogoTitle>
<c:signInLogoLink>http://fujitsu.com</c:signInLogoLink>
```

11.2.2 Modifying the Title

You can modify the title that appears in the Sign In and Sign Up pages by updating the following attribute in section `<c:client>`:

```
<c:signInProductName>FUJITSU Software&lt;br/&gt;UForge</c:signInProductName>
```

11.2.3 Modifying the Sign Up Landing Page

You can modify the page you are directed to when you click on “Need an account? Sign up”, in section `<c:client>`:

```
<c:externalSignUpUrl>http://www.usharesoft.com/products/signup</c:externalSignUpUrl>
```

This allows you to create a dedicated external sign up page outside this user interface so you can customize the way users are created on the platform.

11.2.4 Removing the Sign Up Page

If you want to hide the text `Need an account? Sign up` then change the following attribute to `false`:

```
<c:showSignUp>false</c:showSignUp>
```

11.2.5 Removing User and Password Caching

By default, the Sign In page allows the user to click a check box to remember previous sign in credentials. This is the check box labelled `Remember Me`. If you want to remove this checkbox from the Sign In page, then set the following attribute to `false`:

```
<c:showRememberMe>false</c:showRememberMe>
```

11.2.6 Hiding Option to Reset Password

You can choose to hide the option that allows the user to reset the password from the sign in page. To do so set the option to "false" in section `<c:client>`:

```
<c:showChangePassword>false</c:showChangePassword>
```

This is useful if you have your own authentication mechanism and do not want UForge users to be able to change the password in UForge, since their credentials are managed somewhere else.

Note: When setting this value to `false`, this also hides password management for the user on the `User Profile` section of the user interface.

11.2.7 Modifying or Hiding the Opt In Message for Sign Up

As part of GDPR, you require to inform your users what information you store and how that information is used (stated in your privacy policy and terms of use). The sign up page provides an opt-in checkbox and message. You can change the default message by updating the `signUpOptInText` in section `<c:client>`:

```
<c:signUpOptInText>I accept the Terms of Use and agree to receive emails␣
↪regarding upgrades and maintenance.</c:signUpOptInText>
```

You may also hide this message by updating the `showOptIn` attribute in the same section to `false`.

```
<c:showOptIn>true</c:showOptIn>
```

11.3 Modifying the Signed In Header

You can modify several elements of the header that is displayed when the user is signed in. This includes:

- The logo displayed on the right-hand side
- An external URL link and tooltip (title) when the logo is clicked
- The title displayed on the left-hand side

The following figure displays the default signed in header.

For colours and layout modifications, please refer to *Customizing the CSS*.

11.3.1 Modifying the Title

You can modify the title displayed in the top left-hand corner of the header by updating the following attribute under the `<c:client>` section:

```
<c:headerProductName>UForge AppCenter</c:headerProductName>
```

11.3.2 Modifying the Logo

You can modify the displayed signed in logo by updating the following attribute under the `<c:client>` section:

```
<c:headerRightLogoUrl>images/common/fujitsu.svg</c:headerRightLogoUrl>
```

You can also modify the text that appears when you scroll over the logo, as well as the link the user is directed to when clicking on the link by updating the following attribute respectively:

```
    <c:headerRightLogoTitle>Go to Fujitsu.com</c:headerRightLogoTitle>
<c:headerRightLogoLink>http://fujitsu.com</c:headerRightLogoLink>
```

11.3.3 Adding Links to the Header

You can add links that will be displayed in the header to external resources.

The following example shows how to add a link to a blog under the `<c:header>` section:

```
    <c:header>
    <c:linkItem>
            <c:title>Blog</c:title>
            <c:link>https://blog.usharesoft.com/</c:link>
            <c:target>blank</c:target>
    </c:linkItem>
</c:header>
```

If `<c:target>blank</c:target>` or `<c:target>_blank</c:target>` are used with `<c:linkItem>` the page will open in a new tab.

11.4 Modifying the Footer

UForge Portal allows you to customize the information that appears in the footer. The config.xml file provides two sections to modify this info:

- `<c:client>`
- `<c:footer>`

The version number is automatically provided as part of the initial configuration and any update of the platform.

For colours and layout modifications, please refer to *Customizing the CSS*.

11.4.1 Modifying the Copyright

To modify the copyright under <c:client>:

```
<c:copyright>Copyright © 2007-2014</c:copyright>
```

11.4.2 Adding Links to the Footer

You can add links to external resources under the <c:footer> section.

The following example shows how to add a link to twitter:

```
<c:footer>
        <c:linkItem>
           <c:title>twitter</c:title>
           <c:icon>images/common/icons/twitter.svg</c:icon>
           <c:link>https:twitter.com/usharesoft</c:link>
           <c:target>blank</c:target>
        </c:linkItem>
</c:footer>
```

You can add as many links as you like by adding a linkItem for each link.

If <c:target>blank</c:target> or <c:target>_blank</c:target> are used with <c:linkItem> the page will open in a new tab.

11.4.3 Adding Terms of Use or Privacy Policy

Note: If you want to add your own "Terms of Use" or "Privacy Policy", you MUST include the unchanged UShareSoft Terms of Use and Privacy Policy as part of it.

To modify the Terms of Use or Privacy Policy, go to the sections under <c:footer> and enter the path to the new terms of use and/or privacy policy. You can also modify the link text.

```
<c:footer>
        <c:linkItem>
           <c:title>terms of use</c:title>
           <c:link>https//www.usharesoft.com/about/terms-of-use.html</c:link>
           <c:target>blank</c:target>
        </c:linkItem>
        <c:linkItem>
           <c:title>privacy policy</c:title>
           <c:link>https//www.usharesoft.com/about/privacy-policy.html</c:link>
           <c:target>blank</c:target>
        </c:linkItem>
</c:footer>
```

If <c:target>blank</c:target> or <c:target>_blank</c:target> are used with <c:linkItem> the page will open in a new tab.

11.5 Restricting Change Password

If you want to remove the ability for a user to change their password in the `User Profile` section of the user interface, then set the following attribute in the `<c:client>` section to `false`:

```
<c:showChangePassword>false</c:showChangePassword>
```

11.6 Restricting User Profile Usage

If you want to manage user information (email, etc) outside the user interface, then you can restrict the user profile to `read only`.

```
<c:editUserProfile>false</c:editUserProfile>
```

To hide completely the `User Profile` section of the user interface, set the following attribute to `false`:

```
<c:showUserProfile>false</c:showUserProfile>
```

11.7 Restricting Formats

UForge Portal allows users to generate the templates provided to all the formats the user has access to. In certain circumstances, you may want to restrict the formats shown to the user. To restrict the available machine image formats in the UI you must update the `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/forge-config.xml` file.

To disable a format in the UI configuration, find the `<t:target>` section of the machine image format you want to disable, then either add or change the value of the `<t:enabled>` tag. For example to deactivate OpenStack, the following changes should be made to the configuration file:

```
<t:target>
    <t:id>openstack</t:id>
    <t:name>OpenStack</t:name>
    <t:enabled>false</t:enabled>
    <t:visible>true</t:visible>
</t:target>
```

To hide a format completely, then update the `<t:visible>` tag to `false`.

Note: Restricting formats for specific users can also been done through RBAC in the platform.

11.8 Restricting the Cloud Accounts

When using the UForge Portal, all the cloud account types to which the user has access are displayed. You can restrict the cloud accounts that are visible by updating the `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/forge-config.xml` file of the UI. To hide a specific cloud format, find the `<t:account` section of the account type you want to hide, then set the `<t:visible>` tag to `false`. For example, to remove the CloudStack account type, the following changes should be made to the configuration file:

```
<t:account>
    <t:id>cloudstack</t:id>
    <t:type>cloudstack</t:type>
    <t:name>Cloudstack</t:name>
    <t:visible>false</t:visible>
</t:account>
```

11.9 Customizing the CSS

You can customize all the CSS elements of Portal. You can modify a number of elements like text color, background color, font type and size using the custo.css file located in: `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/css`.

11.10 Customizing the Platform

You can customize some elements of your UForge Portal using the config.xml file located in: `/etc/UShareSoft/uforge/uforge.conf`. This is a relative path.

11.10.1 Configuring UForge Sign Up Information

To modify the UForge sign up information, you can modify the following variables in the uforge.conf file.

- `UFORGE_REGISTRATION_USER`
- `UFORGE_REGISTRATION_CODE`
- `UFORGE_REGISTRATION_PASSWORD`

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

11.10.2 Configuring Email Notification Sender

To modify the sender of automatic emails generated from UForge, you can modify:

- `UFORGE_REGISTRATIONS_EMAIL`

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

11.10.3 Modifying UForge WebService URL

To modify the UForge webservice URL, you can modify the following variable in the uforge.conf file:

`UFORGE_URL`

Warning: The address used for `UFORGE_URL` must be listed in `UFORGE_PROXY_IGNORED`. For example:

```
UFORGE_URL=http://mydomain.com:8080/ufws/
UFORGE_PROXY_IGNORED=mydomain.com,127.0.0.1

#or

UFORGE_URL=http://<ip>:8080/ufws/
UFORGE_PROXY_IGNORED=<ip>,127.0.0.1
```

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

11.10.4 Modifying UForge Portal Application Root Context

To modify the UForge root context of the UForge Portal, you can modify the following variable in the uforge.conf file:

`UFORGE_UI_ROOT_CONTEXT`

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

11.10.5 Modifying UForge API Root Context

By default the root context for the UForge API is set to `/api`. To change the UForge API root context, modify the following variable in the uforge.conf file:

`UFORGE_API_ROOT_CONTEXT`

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

11.11 Customizing the Portal Text

You can customize the text that appears on the left hand menu of the Portal.

Note: At this time, the text on the left hand menu is the only text that can be customized.

You can modify the text using the `dictionary.js` file located in: `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/js/directory`.

You should note the following syntax rules:

- escape ” and character

- use {#} syntax for parameters, follow original constant order, starting at 0
- For plural syntax suround the parameter by [and] at the end of the key, default form (without parameter) is mandatory:
- none: the count is 0
- one: the count is 1
- two: the count is 2
- few: the last two digits are from 03-10
- many the last two digits are from 11-99

The default form is used for everything else (for example 101, 202, etc.)

Note: Javascripts are not supported within the `dictionary.js` file.

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

11.12 Troubleshooting

If you have errors with your Portal once you have applied the changes, please do the following:

1. Check the syntax of the customization files.
2. Check `/var/log/tomcat/catalina-daemon.out` for error reports.

CHAPTER 12

3rd Party Components

12.1 Trademarks

UForge is a registered trademark of UShareSoft, a Fujitsu company.

LINUX is a registered trademark of Linus Torvalds.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

Apache Ant, Ant, and Apache are trademarks of The Apache Software Foundation.

Red Hat Enterprise Linux is a trademark of Red Hat.

UNIX is a registered trademark of the Open Group in the United States and in other countries.

Other company names and product names are trademarks or registered trademarks of their respective owners.

12.2 Copyright FUJITSU LIMITED 2019

12.3 3rd Party Components Used

- Ace: http://ace.c9.io/
- Ace-gwt: https://github.com/daveho/AceGWT
- Angular: https://angular.io/

- Angular JS: https://angularjs.org/
- Angular Resize Event: https://github.com/vdolek/angular-resize-event
- Apache Brooklyn: https://brooklyn.apache.org/
- Apache Commons: http://commons.apache.org/
- Apache HttpComponents: http://hc.apache.org/
- Apache Syncope: https://syncope.apache.org/
- Apache Tomcat: http://tomcat.apache.org/
- Apt libraries: https://packages.debian.org/wheezy/apt
- argparse: https://pypi.python.org/pypi/argparse/
- AWS SDK: http://aws.amazon.com/sdk-for-java/
- Azure SDK: https://github.com/Azure/azure-sdk-for-java
- bzip2-libs: http://www.bzip.org
- CentOS: https://www.centos.org/
- cmd2: https://pypi.python.org/pypi/cmd2/
- cyrus-sasl-lib: http://asg.web.cmu.edu/sasl/sasl-library.html
- device-mapper-event-libs: http://sources.redhat.com/lvm2
- device-mapper-libs: http://sources.redhat.com/lvm2
- dnf: http://dnf.baseurl.org/
- docker-distribution: https://github.com/docker/distribution
- Dozer: http://dozer.sourceforge.net/
- D3: http://d3js.org/
- EclipseLink: http://www.eclipse.org/eclipselink/
- expat: http://www.libexpat.org/
- Filebeat: https://www.elastic.co/products/beats/filebeat
- fontconfig: http://fontconfig.org
- glibc: http://www.gnu.org/software/libc/
- GWT: http://www.gwtproject.org/
- GWTd3: https://github.com/gwtd3/gwt-d3
- Gwt-log: http://code.google.com/p/gwt-log/
- Gwt-reflector: https://github.com/sykesd/gwt-reflector
- Gwt-Storage: https://github.com/seanchenxi/gwt-storage
- Gwt-visualization: https://code.google.com/p/gwt-google-apis/wiki/VisualizationGettingStarted
- Hibernate: http://hibernate.org/
- Hibernate c3p0: http://www.mchange.com/projects/c3p0/
- hivex: http://libguestfs.org/
- HtmlCompressor: http://code.google.com/p/htmlcompressor/

- httplib2: https://pypi.python.org/pypi/httplib2/
- hurry.filesize: https://pypi.python.org/pypi/hurry.filesize/
- ImageMagic: http://www.imagemagick.org/
- Jackson: https://github.com/FasterXML/jackson
- Jackson-dataformat-xml: https://github.com/FasterXML/jackson-dataformat-xml
- Java MariaDb Client Library (JDBC): https://mariadb.com/products/connectors-plugins
- Jawr: https://jawr.java.net/
- jclouds: https://jclouds.apache.org/
- Jersey: https://jersey.github.io/
- Jersey server: https://jersey.java.net/
- Jiruka RSQL: https://github.com/jirutka/rsql-parser
- junit-xml: https://pypi.python.org/pypi/junit-xml/
- krb5-libs: http://web.mit.edu/kerberos/www/
- lcms2: http://www.littlecms.com/
- lesscss: http://lesscss.org/
- libblkid: ftp://ftp.kernel.org/pub/linux/utils/util-linux-ng
- libcom_err: http://e2fsprogs.sourceforge.net/
- libcurl: http://curl.haxx.se/
- libgcc: http://gcc.gnu.org/
- libgcrypt
- libgpg-error: ftp://ftp.gnupg.org/gcrypt/libgpg-error/
- libICE: http://www.x.org
- libidn: http://www.gnu.org/software/libidn/
- liblvm2cmd.so.2.02
- liblvm2app.so.2.2
- libmount lib: http://en.wikipedia.org/wiki/Util-linux
- libpcre: http://www.pcre.org/
- libpng: http://www.libpng.org/pub/png/
- libselinux: http://oss.tresys.com/git/selinux.git
- libsepol: http://www.selinuxproject.org
- libSM: http://www.x.org
- libssh2: http://www.libssh2.org/
- libtar: http://www.feep.net/libtar/
- libtool-ltdl: http://www.gnu.org/software/libtool/
- libxcb: http://xcb.freedesktop.org/
- libXau: http://www.x.org

- libuuid: ftp://ftp.kernel.org/pub/linux/utils/
- libXext: http://www.x.org
- libXt: http://www.x.org
- libX11: http://www.x.org
- lib magic 1.0.0
- lvm2-devel
- log4j: http://logging.apache.org/log4j/1.2/
- MariaDB: https://mariadb.org/
- MariaDB J-Connector: https://mariadb.org/
- Material Design Icon: https://material.io/tools/icons
- ncurses-libs: http://invisible-island.net/ncurses/ncurses.html
- nspr: http://www.mozilla.org/projects/nspr/
- nss: http://www.mozilla.org/projects/security/pki/nss/
- nss-util: http://www.mozilla.org/projects/security/pki/nss/
- nss-softokn-freebl: http://www.mozilla.org/projects.security/pki/nss
- OAR: https://oar.imag.fr/
- OpenDJ: https://forgerock.github.io/opendj-community-edition/
- openldap: http://www.openldap.org/
- openssl-libs: http://www.openssl.org
- paramiko: https://pypi.python.org/pypi/paramiko/
- progressbar: https://pypi.python.org/pypi/progressbar/
- pyparsing: https://pypi.python.org/pypi/pyparsing/
- pyreadline: https://pypi.python.org/pypi/pyreadline/
- RabbitMQ: https://www.rabbitmq.com/
- RabbitMQ client: https://www.rabbitmq.com/clients.html
- Reflections: http://code.google.com/p/reflections/
- Rhino: https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino
- Rpm libraries: http://rpm.org/releases/
- Saxon: http://saxon.sourceforge.net/
- slf4j: http://www.slf4j.org/
- Squid: http://www.squid-cache.org/
- systemd-libs: http://www.freedesktop.org/wiki/Software/systemd
- termcolor: https://pypi.python.org/pypi/termcolor/
- texttable: https://pypi.python.org/pypi/texttable/
- Tika: http://tika.apache.org/
- tinyxml: http://www.grinninglizard.com/tinyxml

- uforge_python_sdk: https://pypi.python.org/pypi/uforge_python_sdk
- URL Rewrite filter: http://tuckey.org/urlrewrite/
- util-linux-ng
- Vault: https://www.vaultproject.io/
- VMware vijava: http://vijava.sourceforge.net/
- VMware SDK: https://developercenter.vmware.com/web/sdk/5.5.0/vcloud-java
- Weld: http://weld.cdi-spec.org/
- xz-libs: http://tukaani.org/xz
- Xz libraries: http://tukaani.org/xz/
- zlib: http://www.gzip.org/zlib
- zypper: https://en.opensuse.org/Portal:Libzypp

CHAPTER 13

GDPR Compliance

General Data Protection Regulation (GDPR) is a regulation that requires businesses to protect the personal data and privacy of EU citizens. In the context of GDPR, the following sections cover some basic information to help UForge administrators list user data held by UForge as well as delete the data associated to a specific user, as needed.

For more information on GDPR refer to the Guide to the General Data Protection Regulation (GDPR)

13.1 User Data in UForge

UForge AppCenter is a multi-tenant platform which can serve multiple users. In addition to the user profile which (which includes user name, email, phone number, address, etc) each user on the platform has (or can have):

- an Appliance Library containing all the appliance templates created by the user
- a Software Library (My Software) containing all the custom software uploaded by the user, which can be used in one ore more appliance templates
- a list of one or more Cloud Accounts to allow the user to publish and register generated machine images to various cloud and virtualization platforms. Each user can also create and join workspaces.
- a workspace. This is an area where members collaborate and share appliances.

13.2 Retrieving User Data

All the information about a specific User can be accessed by specifying the user name in `user_get` API call.

Note: The User, Address, Company and Photo must be deleted using an SQL request described in *Deleting User Information Using SQL*.

The following is an example of a `user_get API` curl call:

```
$ curl "http://10.1.2.41/api/users/john" -X GET -u root:uforgedemo -H "Accept:␣
↪application/json" | json_pp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5250    0  5250    0     0  47806      0 --:--:-- --:--:-- --:--:-- 48165
{
   "user" : {
      "baseWorkspacesUri" : "users/john/workspaces",
      "quotasUri" : "users/john/quotas",
      "admin" : false,
      "usageUri" : "users/john/usage",
      "apiKeysQuota" : 2,
      "digest" : "db79b9a897f9182f71e6c4efa4569f6c",
      "loginName" : "john",
      "name" : "john",
      "jobTitle" : "inge",
      "surname" : "Yorche",
      "officePhone" : "330000000",
      "loginName" : "john",
      "homePhone" : "000000000",
      "homePhone" : "000000000",
      "website" : "mysite.com",
      "firstName" : "smith",
      "mobilePhone" : "044000000",
      "email" : "john.smith@usharesoft.com",
      "address" : {
         "country" : "Mexico",
         "town" : "mexico",
         "digest" : "ebceef9e30482733426c75990160c7aa",
         "lastModified" : "2018-04-17T07:45:32.000Z",
         "dbId" : 101,
         "created" : "2018-04-16T11:43:59.000Z",
         "region_state" : "df"
      },
      "company" : {
         "name" : "myCompany",
         "website" : "gugle.com",
         "dbId" : 101,
         "uri" : "users/john/company/101",
         "created" : "2018-04-16T11:43:59.000Z",
         "lastModified" : "2018-04-16T11:23:29.000Z",
         "digest" : "90978c0fa8ed8029e2db864d19d1db42",
         "address" : {
            "created" : "2018-04-16T11:43:59.000Z",
            "digest" : "8244b7c44a9856e1a8e0151e7ae13029",
            "lastModified" : "2018-04-16T09:43:59.000Z",
            "dbId" : 102
         },
         "parentUri" : "users/john",
      },
      "photo" : {
         "uploaded" : false,
         "lastModified" : "2018-04-16T09:43:59.000Z",
         "dbId" : 101,
         "uri" : "users/john/photo/101",
         "digest" : "db59d1d9a96cbf267dd94947c0211ba3",
         "parentUri" : "users/john",
```

(continues on next page)

(continued from previous page)


```
         "created" : "2018-04-16T11:43:59.000Z"
      },
      "scannedInstancesUri" : "users/john/scannedinstances",
      "baseDistributionsUri" : "users/john/distros",
      "scansUri" : "users/john/scans",
      "baseAppliancesUri" : "users/john/appliances",
      "baseProjectsUri" : "users/john/projects",
      "instancesUri" : "users/john/instances",
      "mySoftwareImportsUri" : "users/john/bundleimports",
      "active" : true,
      "orgUris" : {
         "uri" : []
      },

      "applianceUris" : {
         "uri" : []
      },
      "baseOrgsUri" : "users/john/orgs",
      "baseImagesUri" : "users/john/images",
      "creationCode" : "default",
      "baseTagsUri" : "users/john/tags",
      "baseFormatsUri" : "users/john/formats",
      "baseExportsUri" : "users/john/exports",
      "adminOrgsUri" : "users/john/adminOrgs",
      "dbId" : 101,
      "credAccountUris" : {
         "uri" : []
      },
      "statsUri" : "users/john/stats",
      "baseMySoftwareLibraryUri" : "users/john/mysoftware",
      "baseTargetFormatsUri" : "users/john/targetformats",
      "baseRankingsUri" : "users/john/rankings",
      "baseImportsUri" : "users/john/imports",
      "sshKeys" : {
         "sshKey" : []
      },
      "roles" : {
         "role" : [
            {
               "description" : "Allows a user to manage his profile.",
               "entitlements" : {
                  "entitlement" : [
                     {
                        "description" : "Ability to change the user's own password",
                        "name" : "user_change_password"
                     },
                     {
                        "name" : "user_profile_edit",
                        "description" : "Ability to edit the user's own profile␣
↪information"
                     },
                     {
                        "name" : "user_profile_access",
                        "description" : "Access to manage the user's own profile␣
↪information"
                     }
                  ]
```


(continues on next page)

(continued from previous page)


```
                },
                "name" : "user"
            },
            {
                "entitlements" : {
                   "entitlement" : [
                      {"description" : "Access to edit and manage one or more cloud␣
↪accounts (your credentials to one or more cloud environments).",
                         "name" : "cloud_account_edit"
                      },
                      {
                         "description" : "Access to generate images.",
                         "name" : "image_generate"
                      },
                      {
                         "description" : "Access to upload (or publish) generated␣
↪images to cloud environments.",
                         "name" : "image_publish"
                      },
                      {
                         "name" : "software_upload",
                         "description" : "Access to upload and manage your own custom␣
↪software library.  These software components can be added to your own appliance␣
↪templates."
                      },
                      {
                         "name" : "cloud_account_delete",
                         "description" : "Access to delete one or more cloud accounts␣
↪associated with your UForge account."
                      },
                      {"description" : "Access to your own private appliance template␣
↪library where you can create and manage your own appliance templates.",
                         "name" : "appliance_create"
                      },
                      {"description" : "Access to create one or more cloud accounts␣
↪(your credentials to one or more cloud environments).",
                         "name" : "cloud_account_create"
                      },
                      {  "name" : "cloud_account_access",
                         "description" : "Access to just listing cloud accounts."
                      }
                   ]
                },
                "name" : "designer",
                "description" : "Provide access to a private appliance template␣
↪library.  This allows the user to create and manage appliance templates; generate␣
↪images and publish the images to a cloud environment."
            }
         ]
      },
      "baseGalleriesUri" : "users/john/galleries",
      "vAppAccess" : {
         "digest" : "339b229bf737176bb6be0be176296ef9",
         "dbId" : 101,
         "lastModified" : "2018-04-16T09:43:59.000Z",
         "allowed" : false,
         "cryptAlgo" : 0,
```


(continues on next page)

(continued from previous page)


```
            "created" : "2018-04-16T11:43:59.000Z"
         },
         "adminOrgUris" : {
            "uri" : []
         },
         "productAccess" : {
            "cryptAlgo" : 0,
            "created" : "2018-04-16T11:43:59.000Z",
            "allowed" : false,
            "dbId" : 101,
            "lastModified" : "2018-04-16T09:43:59.000Z",
            "digest" : "b0bca8cf30c3109611297a60f90d85d2"
         },
         "artifactAccountsUri" : "users/john/artifactaccounts",
         "baseTargetPlatformsUri" : "users/john/targetplatforms",
         "baseApiKeysUri" : "users/john/apikeys",
         "baseRolesUri" : "users/john/roles",
         "projectUris" : {
            "uri" : []
         },
         "sshKeysUri" : "users/john/sshkeys",
         "publishGalleries" : {
            "gallery" : []
         },
         "rankings" : {
            "ranking" : []
         },
         "mySoftwareLibraryUris" : {
            "uri" : []
         },
         "baseGalleryOrgsUri" : "users/john/gorgs",
         "lastModified" : "2018-04-17T07:45:32.000Z",
         "baseCredAccountsUri" : "users/john/accounts",
         "created" : "2018-04-16T11:43:59.000Z",
         "tags" : {
            "tag" : []
         },
         "uri" : "users/john",
         "basePublishImagesUri" : "users/john/pimages"
      }
}
```


13.3 Updating User Data

The personal information of a specific User can be updated by specifying the User DTO in the following API call: `user_update`.

13.4 Exporting User Data for Portability

For data portability, users can request a copy of their personal data for their own purposes across different services. The following sections describe all the necessary API calls to collect all information linked to a specific User, such as Appliances, MySoftware and User Credentials. Once all data is gathered, this has to be packed in a zip or tar file.

13.4.1 Exporting Appliances

Use `appliance_export` to export the specified appliance as an archive file. This creates an export ticket that contains an archive file that can be downloaded. The archive file contains the meta-data of the appliance and any 3rd party software or configuration scripts associated with the appliance.

The appliance export is an asynchronous job which may take some time. The response body is an export appliance ticket. To poll the status of this export use `applianceExportStatus_get`. Once complete, the archive file can be downloaded using `applianceExport_download`.

13.4.2 Dowloading an Appliance Export Archive

Use `applianceExport_download` to download the archive of an exported appliance. The archive file is a .tar.gz file. The archive contains the meta-data file of the appliance and any 3rd party software or configuration scripts associated with the appliance.

Note: You can only download an archive file once the creation of the archive file is complete. Use `applianceExportStatus_get` to check the status of an export prior to sending a download request.

13.4.3 Exporting My Software

Use `mySoftware_export` to export the specified software bundle as a compressed archive file. This creates an export ticket that contains an archive that can be downloaded. The archive file contains the meta-data of the mySoftware.

The response body is an export software bundle ticket. To check the status of this export, use `mySoftwareExportStatus_get`. Once complete, the archive can be downloaded using `mySoftwareExport_download`.

13.4.4 Downloading a My Software Archive

Use `mySoftwareExport_download` to download the archive file of an exported software bundle. The archive contains the meta-data file of the mySoftware.

Note: You can only download an archive file once the creation of the archive file is complete. Use `mySoftwareExportStatus_get` to check the status of an export prior to a download request.

13.4.5 Exporting User Cloud Credentials

Use `cloudAccount_getAll` to retriever all the cloud accounts for a user. A list of credAccount objects are returned.

13.5 Deleting User Information

The GDPR introduces a right for individuals to have personal data erased. Individuals can request their data be deleted verbally or in writing.

Note: All data related to the user contained in UForge, such as Appliances, scans, credential accounts or software components, must be deleted before deleting the User. This means you need to identify all the elements related to a given user. Refer to *Retrieving User Data*.

13.5.1 Delete Appliances

Use `appliance_delete` to delete the specified appliance, the following API call should be used, only {uid} should be replaced by the login name of the user to erase.

In order to get the appliance ID {aid} for each appliance created by the user to be erased, use `appliance_getAll` API call has to be used:

Example of how to get the list of IDs of all the appliances linked to a specific user named 'johndoe' :

```
[user@localhost ~]$ curl "http://uforge.example.com/api/users/{uid}/appliances" -X␣
↪GET -u username:password -H "Accept: application/json" | json_pp  | grep␣
↪applianceUri
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 30284    0 30284    0     0   166k      0 --:--:-- --:--:-- --:--:--  167k
                  "applianceUri" : "users/johndoe/appliances/201",
                  "applianceUri" : "users/johndoe/appliances/201",
                  "applianceUri" : "users/johndoe/appliances/203",
                  "applianceUri" : "users/johndoe/appliances/203",
                  "applianceUri" : "users/johndoe/appliances/204",
                  "applianceUri" : "users/johndoe/appliances/204"
                  "applianceUri" : "users/johndoe/appliances/301"
                  "applianceUri" : "users/johndoe/appliances/301",
```

Note the IDs of all the appliances in order to use them as the *{aid}* parameter in `appliance_delete`.

Note: This also deletes all associated machine images (and possible associated image publications) that have been generated from this appliance.

```
curl "https://uforge.example.com/api/users/{uid}/appliances/{aid}" -X DELETE \
-u username:password -H "Accept: application/xml"
```

13.5.2 Deleting Imported Appliances

Similarly imported appliances should be deleted using `applianceImport_delete`

13.5.3 Deleting My Software Objects

Use `mySoftware_delete` to remove a software component from a user's private Software Library

Use `mySoftwareArtifact_delete` to remove a software artifact binary from a software component.

13.5.4 Deleting Scans

Use `scanInstance_deleteAll` to remove all the scanned instances for a user. This call will also delete all the children scan reports and overlay files associated with the scan.

13.5.5 Deleting Machine Images

Use `machineImage_deleteAll` to delete all the machine images generated from an appliance. This will not automatically delete published images.

13.5.6 Deleting Published Images

Use the `scanPublishedImage_delete` to delete the images published based on a machine image. This will not automatically delete the machine image.

13.5.7 Deleting Migration Object

Use `migration_deleteAll` to delete all finished migrations created by a particular user.

13.5.8 Deleting Credentials Information

UForge sorts certain credential information. The following may or may not exist for a given user.

- API key. Use `apiKey_delete` to delete an API key from the specified user API keys.
- Cloud Account. Use `cloudAccount_delete` to delete a cloud account from the specified user cloud accounts.
- SSH key. Use `sshKey_delete` to delete an ssh key from the specified user ssh keys.

13.5.9 Deleting a User from a Workspace

A user may be part of user workspaces. In order to remove a user from a workspace, use `workspaceMember_delete`

Note: You will not be able to delete a user from a workspace if this user is the administrator. In this case, you must promote another user as administrator before deleting the user.

13.6 Deleting User Information Using SQL

UForge API does not provide an API method for deleting a User. The SQL requests have to be executed.

Warning: All data related to the user contained in UForge, such as Appliances, scans, cred accounts or software components, must be deleted before deleting the User.

Note: The table where User information is stored is actually called `Customers`, so the ID of Customer has identified to be able to execute these SQL requests. The Customer ID can be found using the `get_user` API call and looking for field `dbId` inside the User Json objet.

The following is an example of what needs to be deleted (some of these may be empty as not all these fields are mandatory)

```
DELETE FROM addr USING usharedb.AddressesAuth  AS addr INNER JOIN usharedb.Customers␣
↪as cu WHERE cu.db_id = addr.db_id and cu.db_id = '102';
DELETE FROM comp USING usharedb.CompaniesAuth  AS comp INNER JOIN usharedb.Customers␣
↪as cu WHERE cu.db_id = comp.db_id and cu.db_id = '102';
DELETE FROM lo   USING usharedb.LogosAuth      AS lo   INNER JOIN usharedb.Customers␣
↪as cu WHERE cu.db_id = lo.db_id   and cu.db_id = '102';
DELETE FROM cu   USING usharedb.Customers      AS cu   WHERE cu.db_id = '102';
DELETE FROM uo   USING uauthdb.lt_user_orgs    AS uo   INNER JOIN uauthdb.Customers ␣
↪as cu   WHERE cu.db_id = uo.user_id   and cu.db_id = '102';
DELETE FROM uapi USING uauthdb.lt_user_apikeys AS uapi INNER JOIN uauthdb.Customers ␣
↪as cu   WHERE cu.db_id = uapi.user_id and cu.db_id = '102';
DELETE FROM addr USING uauthdb.Addresses       AS addr INNER JOIN uauthdb.Customers ␣
↪as cu   WHERE cu.db_id = addr.db_id   and cu.db_id = '102';
DELETE FROM comp USING uauthdb.Companies       AS comp INNER JOIN uauthdb.Customers ␣
↪as cu   WHERE cu.db_id = comp.db_id   and cu.db_id = '102';
DELETE FROM lo   USING uauthdb.Logos           AS lo   INNER JOIN uauthdb.Customers ␣
↪as cu   WHERE cu.db_id = lo.db_id     and cu.db_id = '102';
DELETE FROM cu   USING uauthdb.Customers       AS cu   INNER JOIN usharedb.Customers␣
↪as uscu WHERE cu.db_id = uscu.db_id   and cu.db_id = '102';
```

CHAPTER 14

Changelog

14.1 3.8.fp13

Release Date: 2019-06-10

14.1.1 New Features

- Docker images now include entrypoint. Previously generated Docker based images have to be regenerated to include it.
- **Lightweight container images. Docker images built by UForge AppCenter from Software Bundles now include a reduced**
  - RedHat Enterprise Linux 7
  - Debian 9
  - Ubuntu 18.04
  - SUSE Enterprise Linux Server 12
  - openSUSE 42
- **Blueprint deployment has been improved. Users can now deploy blueprint on Microsoft Azure Plublic Cloud.**
  - Please note the following restriction: the browsers that support the Blueprint Runtime are the latest public releases of Firefox and Chrome.

14.1.2 Enhancements

- root ssh access is allowed for UForge Azure images

- When creating a blueprint, if an appliance is interactive, the "Deploy" button cannot be selected. A "warning" icon is positioned on the appliance. When the warning icon is selected, a popup opens with a button "Configure Install Profile". Selecting the button saves the blueprint and redirects the user to the appliance profile allowing him to modify it.
- New deployment view: the new UI allows you to manage and monitor the evolution of your blueprint deployment. It provides clearly status for each instances deployed, indicating any errors as well as configuration and runtime information. Please note UForge RPM installation will update the Brooklyn plugin. If the AMP/Brooklyn endpoint is configured, the Brooklyn server must be restarted after installing UForge RPM.

14.1.3 Bug Fixes

- 1343 Repository refresh fails for repositories where file listing is disabled
- 12200 We can't download some binaries if the root context is "/"
- 13203 Supervisor can not change supervised user
- 13411 Need to login with root/sudo user to VM before being able to use it even if no interaction required at installation
- 13538 Scan fails with API key authentication if user name includes '@'
- 13583 Remove call to another unexistant supervision icon
- 13654 Error when adding a new SSH key that ends with a blank line
- 13867 No clear error given when deploying an image that is deleted from Azure/AWS
- 13904 500 call failed in blueprint YAML editor

14.2 3.8.fp12

Release Date: 2019-04-29

14.2.1 New Features

- Users can publish Windows machine images to Azure Stack.
- Lightweight container images. Docker images built by UForge AppCenter for Software Bundles used to have unnecessary packages such as kernel. They now include a reduced set of packages, included in a base image, that allow users to build images on top of it. Only CentOS 7 is supported for now.

> **Warning:** *Container* is now an OS Profile provided by UForge for all Linux distributions. Existing OS Profiles with this name must be renamed before the update.

14.2.2 Enhancements

- Ability to view the list of packages for images generated from a software component.
- Ability to generate a Docker image from a Linux appliance without the kernel.

14.2.3 Bug Fixes

- 13489 Fix incompatible version error when generating Oracle 7 image
- 13551 Make all Azure agent packages sticky
- 13350 UForge system update: `yum update docker` breaks docker command
- 10912 Empty folders in MySoftware are not present in generated machine
- 12518 Fix ssh connection for openstack
- 13291 Slash in MySoftware files not well handled
- 13302 Some appliances with an interactive profile do not have the warning icon when composing a blueprint
- 11945 Publishing to AWS stops at 39% when image size is large
- 12893 Only the bottom part of the appliance logo can be clicked
- 12472 Icon overlap in "Advanced Partitioning" and "Welcome Message" views
- 12877 Icons overlap after leaving edit network card without using ok or cancel
- 12239 Status of some 'services' like ufw is not detected by scan agent
- 11940 One volume remains after AWS publish cancelled

14.3 3.8.fp11

Release Date: 2019-03-18

14.3.1 New Features

- Develop and test your software applications faster. Thanks to the new local docker registry installed with UForge, you can docker run your containers directly.

  Note: Previously generated Docker based images will no longer be downloadable. You must regenerate them to use this feature.

14.3.2 Enhancements

- Blueprint deployment has been improved. The user can now generate and publish all images of the blueprint appliances by clicking on the new `prepare` button.
- Blueprint Composer now raises a warning if one of the appliance component configuration requires user interaction at first boot
- UForge provides a PowerShell command to download uforge-scan.exe from a Windows Server

14.3.3 Bug Fixes

- 13028 Imported appliances from workspace break the blueprint appliance list
- 10553 Windows generation for some formats fails due to some registry keys

- 10854 'uforge template list' shows created time in local time while modified time in UTC
- 13077 Creating an Oracle-based target format using the CLI raises an error
- 13080 Can't connect to a CentOS 6.7 AWS instance with SSH from a "whitebox" migration
- 12854 'uforge pimages list' fails with AttributeError
- 9637 Creating a migration with unauthorized characters in name succeed through the API
- 6495 Highlighted element does not handle overflow correctly
- 12759 ImageDTO returned from API has always CentOS 7 as distributionName when image generated from software
- 12284 Windows on Azure generated from UForge-published image for Blackbox migration does not start correctly
- 12909 Mandatory fields are not highlighted in account summary of user
- 12116 Command to launch distribution population is wrong in administration part
- 11942 Sourced network interfaces are not detected by the scan agent
- 6316 Windows scan fails if second disk is full
- 12856 uforge-sync.bin overwrites PID file and makes a service behave strangely
- 12605 Files & Folders scan tab displays wrong information
- 10096 There are inconsistencies in the French translation
- 12806 Downloading an OpenShift image directly retrieves the archive without explanation
- 12588 Cannot delete multiple users in Workspace Members page
- 12734 Publishing to Azure ARM, K5 and VCenter terminate with error when it takes more than 8 hours
- 10068 Error generating Windows Server 2012r2 Standard Full English for Amazon AWS (VirtIO drivers issue)
- 9404 Error while generating an Azure image from a Windows Server 2012-R2 appliance
- 9247 Unable to generate Windows Server 2012R2

14.4 3.8.fp10

Release Date: 2019-02-04

14.4.1 New Features

- Added a new `Delta` tab in Migrations section to visually compare the difference two scan results (from one or more live instances). This allows users to visualize which files have been added, modified or deleted. Users can browse the filesystem, search and filter the results. This feature does not provide differences in the packages added, modified or deleted between the two scans at this time.
- **Introduce the new user interface Blueprint Composer which allows users to create and deploy multi-node applications. Pl**
  - This release currently supports Linux and Amazon AWS only.
  - For the deployment of the blueprint, UForge must be associated with the new AMP version 5.3

  - You need to install the Brooklyn Plugin Uforge after the installation of AMP server by following the procedure described in Admin Guide->Further AppCenter Configuration->Configuring Cloudsoft AMP
  - The Browsers that support the Blueprint Composer are the latest public releases of Firefox and Chrome.
  - Only the English version of the Blueprint Composer view is available.
  - Blueprints created with a version of UForge older than 3.8.FP10 will be deleted
  - Note that this new release brings an API break for Blueprint DTO model.

14.4.2 Known Issues

- Package file deletions are not detected when comparing scans

14.4.3 Enhancements

- Disable Software Bundle generation page if no base OS available
- Inject firewall configuration package when "ask during installation" is selected on Centos 7
- Disable unsupported "ask during installation" firewall option on Debian and Ubuntu

14.4.4 Bug Fixes

- 8626 After scanning Windows 2016, one service, CDPUserSvc_XXXXX, is missing on the list of scan results
- 8871 Changing partitioning from Advanced to Basic in a template imported from a Windows scan leads to generation error
- 10519 ISO image created by UForge does not respect UTC
- 11777 Password is displayed in clear text on Summary for an application
- 12265 When publishing an appliance from blueprint UI, the row is moved to the bottom of the table
- 12280 CentOS 5 scan fails without explicit message when duplicated GPG Pubkey packages are installed
- 12293 Updates counter on appliance view is placed too far to the right
- 12337 Copyright customized through config.xml is not shown
- 12433 User password displayed in clear when deploying Windows image on AWS
- 12474 Reset the password after an attempt of sign-in with a wrong password fails
- 12480 Root password is displayed in clear inside the generated machine image
- 12483 A previously assigned mountpoint can't be reused in the UI after a file system type change
- 12619 Password is displayed in clear text on Summary for a workspace

14.5 3.8.fp9

Release Date: 2018-12-21

14.5.1 New Features

- Ability to generate a software component to a Docker image. The user can select any Linux distribution as a "Base OS".
- Support of hotkey Escape (ESC) to cancel/close a popup window.
- UForge logs can now be pushed to an ELK instance.

14.5.2 Enhancements

- UForge Microsoft Azure images now use the Azure agent version 2.2.14-1 for Debian 8 (Jessie)
- UForge Microsoft Azure images now use the Azure agent version 2.2.18-3 for Debian 9 (Stretch)
- Improved Outscale publish connector robustness
- Blueprint deploy view has been improved. The user can now see which blueprint appliances are not ready to be deployed.
- Blueprint deploy view has been improved. Required actions for deploying the blueprint can now be triggered from this view.

14.5.3 Bug Fixes

- 6596 Unknown server error when editing incremental scan after deleting the base scan
- 9171 Publication to Fujitsu K5 timeout with slow network
- 9419 The forgotten password email contains both the username and the password
- 11191 Firewall is not configured on an image generated from an appliance with firewall activated
- 11812 Google certificate is in clear in log when uploaded
- 11933 uforge-scan.bin fails to execute when /tmp has noexec mode
- 11970 Deleting pinned package looks ok in UI but package is not deleted
- 12193 Cannot publish to Outscale us-west-1
- 12349 uforge-scan.bin cannot be downloaded from the UForge UI
- 12352 SELinux configuration is not supported for Oracle Linux

14.6 3.8.fp8

Release Date: 2018-11-12

14.6.1 New Features

- Ubuntu 18.04 supported, except for synchronization feature in migration workflow.

14.6.2 Enhancements

- Using uforge-sync binary, users can now synchronize the target environment with scans (without overlay) of openSUSE 42 systems.
- Improve usability of the add and delete actions on Projects page.
- Blueprint deploy view has been improved. Appliances used in the blueprint, with their status, are now displayed.

14.6.3 API changes

- Google Cloud Engine authentication method has been updated to support the new format used to authenticate to the platform. As a result, credentials accounts have changed. The certificate is no longer a `.p12` file but a `.json` file.

> **Note:** Old Google Cloud Engine credential accounts will no longer be usable. You must replace them by new ones in the correct format.

14.6.4 Bug Fixes

- 11941 Get requests to vault fail in proxy environment
- 11863 Cannot login to migrated CentOS6 image on AWS with SSH key
- 11799 Outscale images built by UForge do not boot on Outscale
- 11637 Empty directories are not synchronized to the target machine with uforge-sync.bin
- 11608 Credentials secret keys are visible in clear for Outscale, Amazon and CloudStack
- 11548 "500 call failed" is shown on non-english summary tab when a new Windows template is created
- 11532 OpenShift installation is incomplete when upgrading to 3.8.fp6 with many users
- 11499 Vault in proxy environment does not work
- 11390 User email address should not be exposed to the other user
- 11354 Partitioning Table Volumes subtitle misses a white space
- 10870 Publishing to Google Compute creates unnecessary disks and images
- 10697 Cannot publish to GCE when AppCenter is behind a Proxy Server
- 10503 Mislabelled UForge on Update tab when creating a Windows appliance
- 10444 When user quota limit of appliance is set, "Quota used" increases by 2 when importing from scan
- 10443 Cannot create Azure VM from published VHD from AWS Ubuntu template - No NIC detected
- 8989 Some French translations are not accurate or missing
- 8897 Spelling mistakes in English i18n constants
- 5224 Typo: "Unformated" in Install Profile -> Partitioning
- 1351 "Internal server error." displayed when publishing a Google Compute Engine with wrong credentials

14.7 3.8.fp7

Release Date: 2018-10-01

14.7.1 New Features

- UForge administrator can now register and manage software repositories and operating systems from the user interface
- Users can deploy Windows instances from published machine images to Azure

14.7.2 Enhancements

- UForge Microsoft Azure images now use the Azure agent version 2.2.21 for Ubuntu 14.04 and 16.04
- UForge Microsoft Azure images now use the Azure agent version 2.2.18 for Red Hat Enterprise Linux 6 and 7
- Using `uforge-sync` binary, users can now synchronize the target environment with scans (without overlay) of Debian 6 systems
- New icon in the UI for blueprints

14.7.3 API changes

- Update the Repository DTO model: rename field `officiallySupported` to `coreRepository`

14.7.4 Deprecated Features

- UForge command line tool option `--officialySupported` for command `org repo create` is deprecated. Please Use `--coreRepository` instead

14.7.5 Bug Fixes

- 8341 Fixed issue allowing japanese characters to be used in the description field for an OS Profile
- 8934 Fixed issue to allow a user to correctly delete a pinned package in the OS profile of a template
- 8936 Fixed validation tooltips where backslashes are actually not supported
- 8940 Fixed unclear error message when editing a software component bootscript used in an ongoing generation
- 10708 Fixed issue when generating an ISO image from a scan of a live system with CentOS installed
- 10822 Fixed issue generating Ubuntu 10.04 with the latest `debootstrap` package
- 11096 Fixed the unpinning of a package from failing in the UI
- 11201 Fixed arrow buttons from expanding in Distribution > OS Profile view in the UI
- 11226 Cannot log in to a CentOS AWS instance with SSH key pair set by AWS
- 11292 Replaced `vssadmin.exe` with `diskshadow.exe` to properly flush Windows registry during migration process
- 11349 Fix to display tenant name in the details of a published image for OpenStack

- 11375 Fixed refresh issue for the top navigation menu to display its children when the window is resized horizontally
- 11376 Fixed refresh issue for the main navigation menu to display its items when the window is resized vertically
- 11492 Fixed misaligment in the Updates notification information in the UI
- 11504 Fixed issue with the blueprint view filter being cut during loading

14.8 3.8.fp6

Release Date: 2018-08-20

14.8.1 New Features

- New option to migrate (Lift & Shift) an instance without transferring any overlay information (only keep the operating system information, remove software application and users data)
- Using uforge-sync binary, users can now synchronize the target environment with scans (without overlay) of Ubuntu 16.04, 14.04, 12.04 systems.
- Support generation and publication of Linux machine images for OpenShift. However, it is no longer possible to publish from a Docker image to OpenShift.

14.8.2 Enhancements

- UForge Microsoft Azure images now use the Azure agent version 2.2.18 for CentOS
- Improve display of error details for failed migrations
- Enhance Clone Appliance view to display the version and revision of the current appliance to be cloned
- Improve usability of the add and delete actions on MySoftware page
- Clicking on “UForge AppCenter” (top-left corner) now redirects the user to the dashboard
- Clicking on an Appliance now redirects to the Stack tab
- Ability to publish Windows 2016 appliances on Fujitsu K5 cloud

14.8.3 API changes

- Update the Image DTO model: rename field applianceUri to parentUri.

14.8.4 Bug Fixes

- 5175 UI returns 500 call failed when portal has changed and requires clearing cache and reloading
- 7195 /etc/sysconfig/system-config-firewall file created after migration though it is not supported in RHEL 5.2
- 8050 CentOS 5 scan fails with duplicate GPG Pubkey package installed without explicit message
- 8439 UForge version displayed in the portal is incorrect
- 8724 The file name of a cloned software is incorrect

- 9475 /etc/UShareSoft/uforge-install-config-CheckRootLogin.sh not found after CentOS blackbox migration to AWS
- 10246 CLI timeout following *subscription os add* with many users
- 10653 CLI command *template info –all* always displays 0
- 10811 Deployment of Linux images to Microsoft Azure does not take ssh key into account
- 10478 Deleting a publication raises errors
- 11045 Deleting two publications raises errors
- 11059 Migration does not launch generation in a multi-node UForge environment
- 11170 Impossible to delete a PublishImage that comes from a migration
- 11171 Deleting an Image from a Scan does not work
- 11318 Some dependencies of platform tools are not injected when generating from a scan
- 11343 uforge-install-config does not execute correctly for Ubuntu 14.04

14.9 3.8.fp5

Release Date: 2018-07-09

14.9.1 New Features

- Allow users to cancel running migrations
- API users can now publish to OpenShift from a Docker image (compatible with Hammr CLI), tested on OpenShift Online and OpenShift Origin
- BTRFS filesystem support
- New customizable opt-in message in Sign Up page.
- Support Entrypoint in Docker images
- Users can deploy Windows instances from published machine images to AWS.

14.9.2 Deprecated Features

- Remove support of following formats: Abiquo, Eucalyptus, Flexiant, Nimbula

14.9.3 Enhancements

- The uforge-migrate binary displays now the progress of each phase.
- User can specify the network bandwidth allowed for data transfer when scanning a Windows system.

14.9.4 Bug Fixes

- 10251 VirtualBox image of UFIAB fails to boot with initrd root filesystem dependency failure
- 10331 Generation stucked at 55% and nothing work anymore after it
- 10335 Failed to import OVA image into vCenter
- 10430 Publish to all regions of Outscale does not work in UForge (unsupported regions, ami id out of date, wrong user for connection)
- 10555 On AppCenter with many users, appliance GET for one user has performance issues
- 7617 In the publish views, some select lists are randomly sorted
- 8638 License cannot be changed in the clone Software
- 9847 [Documentation] File size of /boot/grub2/i386-pc/core.img is changed during blackbox/whitebox migration
- 10217 Floating point exception occurs on uforge-scan.bin
- 10325 OAR jobs logs show WELD "Exception in thread"
- 1415 Portal - Language selection menu truncated in firefox
- 9672 SLES 11 scan sticks during creating report
- 9676 When adding a certificate for the creation of a google compute engine, the spinner never stops
- 9782 Windows black box migration failed at publication with classCastException
- 9836 [Documentation] configuration to connect to AMP is incorrect for multinodes environnement
- 9956 Overlapping text when generate an K5 image from a Linux scan
- 9961 Migration is stuck in progress
- 10103 Import a bundle first as first action will block subsequent template imports
- 10105 Scrollbar is not well displayed on deployments and blueprints views
- 10588 Windows on Azure generated from UForge-published image for Whitebox migration does not start correctly.
- 10657 You cannot save a Windows template imported from a scan with an error.
- 10795 Generation of a Debian 8 Server OS profile image fails with server install profile

14.10 3.8.fp4

Release Date: 2018-05-01

14.10.1 New Features

- The automated migration process is now available for Windows
- Cloud credentials are now stored in a new secret manager (Vault) in order to improve security
- New option to exclude some files and directories when migrating a live machine
- Introduce the new Blueprint module which allows users to create and deploy multi-node applications. This release currently supports Linux and Amazon AWS only.

14.10.2 Enhancements

- Improve the migration details page in the user interface

14.10.3 Bug Fixes

- 3695 No space left on virtual disk is not caught as an error and generation returns "internal server error"
- 9013 Docker image format generated by UForge is incompatible with latest Docker executables
- 9044 Publication to AWS or Outscale: message when cannot connect to proxy is misleading
- 9406 Azure publishing feature transfers VM images via HTTP instead of HTTPS
- 9555 Subscription quota update command without a limit sets the limit to 0
- 9679 Google Compute Engine Regions are outdated
- 9733 Use public IP address instead of private one in the security group rule when publishing to AWS
- 9918 UI freeze in Google Chrome when entering the 'Stack' tab
- 9940 Cannot publish to azure with existing account through REST API
- 9941 Image generation for K5 failed while checking the dependencies
- 10099 Boot scripts cannot be found in the cloned MySoftware

14.11 3.8.fp3

Release Date: 2018-04-16

14.11.1 New Features

- Windows system with an extended partition is now supported for templating, image generation, and scanning.
- The automated migration process is now available for all supported platforms

14.11.2 Enhancements

- Visualise all the software (libraries, drivers or packages) that is automatically injected by UForge during the generation of a machine image for a specific target cloud environment.
- "Migrations" tab now contains both automated migration and scan features, available under "Lift & Shift" and "Re-platform" sub menus
- Allow users to delete multiple migrations
- 9057 Remove End-of-life Microsoft Azure Classic

14.11.3 Restrictions

- Currently the image generation of Linux system for K5 migration fails. This is due to a known issue during dependency checking. 9941 Image generation for K5 failed while checking the dependencies

14.11.4 Bug Fixes

- 9937 Segmentation fault when generating a machine image
- 9762 Default chunk size for publishing to K5 is too low and leads to K5 error
- 9411 When scanning a machine, UForge portal UI displays a big OS Logo. Fix the logo size in the header and allow to debug CSS from remote computers
- 9771 Docker image format should not be proposed for Windows migrations
- 9781 When selecting Azure platform, AWS or K5 for generation, an unknown server error is displayed
- 9424 When a migration is deleted, the error message of the migration tool is not relevant
- 9641 Delete account with a certificate, after a publish, fails
- 9639 Wrong URI for Scan installProfile
- 9657 Unable to publish Docker images to Azure Containers
- 9615 Generation dashboard is in error after a Migration Generation Stage has completed
- 8917 After Blackbox Migration, RHEL5.3 is updated to RHEL5.5.
- 9598 Provisioning on Azure fails due to No DVD device
- 9582 SLES generation for Azure format does not work (no platform tools injected)
- 9258 Unable to do a Debian 8 blackbox migration to Microsoft Azure
- 9074 Improve error handling in getCredAccountResources service
- 9188 Partitions in LVM logical groups/volumes appear in wrong order if a group's name is changed
- 9532 When importing an appliance, Firewall is set to Ask during installation
- 9502 NIC configuration disappears on install profile in an imported appliance from a CentOS5.7 scan
- 8682 If the root context is not modified in the deployment wizard, the page redirected to at the end is / which is forbidden
- 9081 "should contains" to be replaced with "should contain" in bootscript name in UI
- 6200 Fix scan installProfile URI
- 8973 Ubuntu 16.04 image generation requires debootstrap to be of version at least 1.0.85 and does not support "proposed" packages
- 9094 If a user's home directory is in a multiple levels folder hierarchy, the user creation fails
- 8951 Remove GoldenPath field from uforge.xsd
- 9176 Please remove obsolete Squid directives from /etc/squid/squid.conf
- 7937 Debian dependency checker returns too much detailed message when failed.
- 8371 Add SYSPREP setting in windows yml template
- 9317 uforge_update.sh fails with SQL error in db_modifs_180130-01.sql

14.12 3.8.fp2

Release Date: 2018-03-05

14.12.1 New Features

- The migration process (scan, generate and publish) can now be automated by using a simple and intuitive workflow in the user interface. The user simply creates a new migration through the interface, launches it from the server to migrate and follows the overall status.

  Note:

  - This new feature is available under the "Migrations" tab. This tab already existed. This previous tab has been renamed "Scans".
  - This feature is currently available for the following platforms : Fujitsu K5, Microsoft Azure, VMware vCenter, Openstack and SUSE Cloud.

14.12.2 Enhancements

- 9004 As a user, I would like to have my appliance revision incremented when I modify an attached Software-Bundle
- 5994 Provide a way to offuscate and reveal passwords in the UI
- 9054 RHEL 7 and 6 should be supported in K5 format

14.12.3 Bug Fixes

- 9091 iptables rules have been changed - whitebox migration
- 8721 Scanning CentOS 6 generated with UForge results in an error
- 8648 The Deployment Wizard should block non-numeric inputs on the Proxy Port
- 8646 Debian 9 missing in the deployment wizard
- 8745 Disk usage increases when uploading twice the same file for limited quota user
- 9178 Overlapping text in the summary view of an appliance template
- 8664 systemd-tmpfiles-clean.timer clears /tmp thus removes the symlinks and breaks AppCenter
- 8613 [RHEL7/CentOS7] /boot/grub2/device.map is cleared during Blackbox/Whitebox Migration
- 8222 Software bundles are not extracted in the correct directory
- 8847 ComboList in portal appears empty after selection with firefox
- 8921 Add scan import to golden cancel webservice
- 9024 Outscale image generation limited to 10 GB disk size
- 8701 UForge deployment fails behind proxy because of unaccessible ntp server
- 8422 openssh bits in uforge template do not need to be sticky anymore
- 9198 Service mysql restart display FAILED message
- 8865 Despite deleting the golden images, the files of the golden image are not deleted in the file system
- 9053 user login and password are sent to the user in the same e-mail message
- 8842 Docker publishing cannot be canceled
- 9010 After a file upload error, the value of consumed diskusage increases when the quota is changed to unlimit

- 8412 Ubuntu 14.04 generation fails with stack overflow error
- 7493 A letter '&' in comment field of /etc/passwd file changed to '&' after migration
- 8873 CPU usage of Dozer thread sticks to over 99% and never ends
- 8995 script machine_infos.sh fails
- 9045 Shell injection, the user can execute command as tomcat user when calling publish api
- 9026 template imported from scan fails to generate
- 8899 openssh package version has been changed after a white box migration
- 9185 If user role is only Migrator, an error occurs in cloud account selection of publish image
- 9165 No need to call reset_eventcontroller.sh in the crontab anymore
- 9125 After Black box migration, Firewall setting changes to enable in Cent OS 6.
- 9257 Error occurs when migrating to Hyper-V of Windows Server 2012 R2
- 9108 [Server-side]Add the Timezone param to uforge-install-config.conf
- 9089 [Server-side]don't write a firewall param if the template is Windows
- 8320 "user enable" uforge-cli command always resets password
- 8987 "Request timeout" is be shown during a generation and requires refreshing the screen
- 8444 Missing /opt/Tomcat/.bashrc file
- 9144 a logical group disappears after removing an LVM disk
- 9175 removed partitions from a volume group still remain in the group
- 9210 Scanning a CentOS 7 server with a CD in the drive causes an additional hard disk to be added

14.13 3.8.fp1

Release Date: 2018-02-01

14.13.1 New Features

- Users can deploy Linux instances from published machine images to Apache CloudStack.
- Fujitsu Cloud Service K5 jp-east-2 region is supported.

14.13.2 Enhancements

- The uforge-sync binary now requires the API endpoint, to improve usability.
- Enhance UI headers for Apps and Migration tab.
- Add French internationalization.

14.13.3 Bug Fixes

- 835 Refresh the generation page causes a 500 call failed
- 1060 Files permissions changes after blackbox migration
- 1064 Timezone is always reset to Europe/London after a blackbox migration.
- 1416 Portal - MySoftware - Files - package file path not restored
- 6769 The presence of a malformed filename in the source filesystem causes the scan to hang badly (segmentation fault)
- 7019 Filename vCneter.log spelled wrong. Should be vCenter.log.
- 7021 Backslash not properly escaped in credentials causes VMware vCenter publish to fail
- 7087 UTC and ARC settings in /etc/sysconfig/clock has been changed during blackbox migration
- 7112 In a blackbox migration /etc/USharesoft/ files are not deleted
- 7134 'org repo delete' fails frequently and the error message is confusing
- 7187 Parameters in /etc/fstab has been changed during blackbox generation
- 7193 /etc/gshadow has been changed during the blackbox migration
- 7196 /etc/shadow lock and 'no password' options not taken into account during migration
- 7214 Popup have an unexpected scroll bar
- 7275 Error message does not include any information when publish to AWS failed.
- 7284 Directories/files changed during blackbox migration
- 7416 Packages of custom repo still visible even after repo detach
- 7423 The UI view for searching and adding an OS package to a template shows too many versions
- 7659 Blackbox migration of CentOS7.2 on fresh forge failed "Detaching loop"
- 7680 /etc/sysconfig/clock file is added in Blackbox and Whitebox migration
- 7684 /etc/sysconfig/kernel is modified after Blackbox and Whitebox migration
- 7687 Hammr deploy OpenStack retrieval timed out
- 7712 Viewing bootscript of a cloned template raises a 404 error
- 7730 Uploading several files to a Software bundle randomly leads to 500 error
- 7738 Bad concatenation in kernel parameters after two blackbox of a debian appliance
- 7747 The field for disk size at generation for AWS should be in GiB
- 7748 Unsupported AWS region are displayed in the publish view (cn-north-1, us-gov-west-1)
- 7758 UForge cli takes minutes to manipulate repositories
- 7819 uforge-sync does not resolve fully qualified names for AppCenter endpoint
- 7869 Enabled Firewall becomes disabled after Scan and Import
- 7935 uforge-scan output is not proper english
- 7940 Version of uforge-scan is not consistent with version of UForge platform it has been downloaded from
- 8054 "UForge critical error" e-mail is sent after a successful scan import
- 8055 "C:fakepath" is displayed when selecting an appliance archive to import

- 8062 Publishing a compressed image failed on OpenStack
- 8063 UForge update logs show WELD "Exception in thread"
- 8064 Software bundles are not extracted in the correct directory
- 8076 "500 call failed error" when uploading a boot script to a project catalog
- 8097 VMware vCenter publish fails in multiple vlan/vnic environment
- 8102 Display explicit error message when template has no partition
- 8146 Typo in Artifact account in the creation page
- 8180 Folders where VMware vCenter templates will be published are changed randomly
- 8192 OpenStack generation from scan fails with message Installed packages more than expected (240 > 237)
- 8214 When moving from Name to Version with the tab key, the Version box becomes red (error)
- 8306 Export, Import and Scan features does not work when UForge user login contains '@'
- 8322 Publishing a compressed image failed on VMware vCenter
- 8326 The order of NICs is changed by exporting/importing a template appliance
- 8476 uforge-cli command template info throws AttributeError: NoneType for Windows Appliances
- 8649 Only one architecture of an OS package is kept when there were multiple in the imported template
- 8889 uforge-cli command template info throws AttributeError for Windows Appliances
- 8898 Publish on OVH Openstack does not work

14.13.4 Known issues

- In some situation, deployment in CloudStack could fail if the CloudStack image is duplicated in different zones.

14.13.5 Compatibility issues

- The import / export of appliance templates from UForge 3.7 to UForge 3.8 may not work if the template contains software bundles. Please refer to the section Importing and Exporting Templates (Updating a 3.7 Appliance) to make your template compatible.

14.14 3.8

Release Date: 2018-02-01

14.14.1 New Features

- SLES 11 and 12 operating system supported for all features (templating and migration)
- OpenSUSE 42.x operating system supported for all features (templating and migration)
- Debian 9 (Stretch) operating system supported for all features (templating and migration)

For other features, please refer to 3.7.fp8 release notes

14.14.2 Migrating to 3.8

For specificities relating to migrating a 3.7 or 3.7.fpx to 3.8 please refer to the section Migrating UForge from 3.7 to 3.8 in the Admin guide.

14.14.3 Bug Fixes

- 8656 Estimated size of Windows templates is 0 B
- 8653 Generation fails for an imported Windows template built on a "Scan To Golden" profile
- 8578 Generation does not finish if there are volume groups though it was cancelled or got an error
- 8577 Image generation of a CentOS 6.7 scan from ISO fails in grub installation
- 8507 Update error message about RHEL not supported for K5 in UForge
- 8505 Publishing a Docker image fails in slow network environment
- 8501 Export, import and scan features do not work when UForge user login contains `@`
- 8486 Image generated from a CentOS 7.1 scan from ISO fails to boot showing the grub shell
- 8499 UNIX group ID is not taken into account when import a template
- 8437 Name and downloadId missing in the download URL for appliances generated from a template
- 8417 Windows image generation from a legacy golden fails without displaying the details if required disk is too small
- 8309 Windows automatically shuts down after being instantiated on AWS
- 8270 Scan of CentOS 7 fails with message `Unable to rebuild package dialog 1.2 x86_64` on 3.8
- 8094 Whitebox image generation failure with `non encrypted password` error
- 8078 Add arch selection, in order to allow install of package with multiple architectures
- 7831 Scan on CentOS 7.4 with LVM fails silently and causes generation error

14.15 3.7.fp8

Release Date: 2017-10-16

14.15.1 New Features

- Using uforge-sync binary, users can now synchronize the target environment with scans (without overlay) of CentOS 6, 7, Red Hat Enterprise Linux 6, 7 and Debian 7, 8 systems.
- Microsoft Azure connector has been updated. Previously with UForge the machine image was publish as a "vhd" blob file in the Azure cloud Account. Now an image will be accessible in the cloud console from this blob file. In order to support this additional information must be entered in `Credentials` for Microsoft Azure ARM connector.
- Support generation and registration of machine images for Oracle Cloud with the metered service subscriptions.
- Users can deploy Linux instances from published machine images to Microsoft Azure ARM.

Note: If you have an existing Microsoft Azure ARM account already setup in UForge, then you must update the credential information.

14.15.2 Enhancements

- Improved deploy button tooltip in Dashboard view
- Replace spinner by ProcessStepWidget for OpenStack
- UForge users can inject specific VirtIO drivers for Windows appliances
- Amazon AWS connector can now publish Windows images with multiple disks

14.15.3 Bug Fixes

- 1311 Error "WELD-ENV-002002: Weld SE container was already shut down" can be displayed in the portal when generation failed
- 6196 Image generation from a scan fails when the repository is updated by the spider simultaneously
- 6359 Scan comparison shows two packages with different versions instead of package's target scan
- 6669 Installing UForge AppCenter in a root context other than /uforge breaks some features
- 6848 Disk order and partition number are not kept after migration
- 6862 All fields in deployment tables should be displayed entirely
- 6957 When scaning a RHEL machine, UForge portal UI displays a big RHEL Logo in IE
- 7004 /etc/sudoers is reinitialized after migration
- 7016 CentOS images from blackbox migration fail to start on Microsoft Azure: no WALinuxAgent installed
- 7076 Generation error when extracting overlay if size is bigger than / (root) partition
- 7109 Tooltip of source used on a deployments is wrong if come from a scan
- 7114 Protect Deploy Activity from incomplete publish image
- 7149 When scanning Windows 2012 R2 and blackboxing it to VirtualBox, Windows requires to change admin password at first boot
- 7150 Error when specifying a directory of more than depth 1 in mount points in install profile
- 7164 Blackbox migration of debian 7, 8 and ubuntu 14 does not boot on major clouds due to DHCP ipv6 activation
- 7184 NetworkManager package is present in "server" profile and the generation does not work with Azure
- 7194 CentOS 5.11 scan fails at phase 4/7 by segmentation fault
- 7253 Scan fails with SQL Error: 1205, SQLState: HY000 when running two scans concurrently
- 7408 CentOS whiteBox migration to Microsoft Azure: wrong version of WALinuxAgent selected
- 7510 CentOS 7.4 and Oracle Linux 7.4 fail to boot
- 7673 Generation of a migrated debian 8 fails randomly
- 7686 Whitebox migration : multinic method of second interface is disabled instead of static or manual
- 7697 File System type not set properly for logical partitions

- 7711 Outscale cloud: cannot see and publish in new regions

14.16 3.7.fp7

Release Date: 2017-09-04

14.16.1 New Features

- Fujitsu Cloud Service K5 US, Finland and Spain regions are supported.
- Amazon AWS Ohio, Mumbai, London and Canada regions are supported.
- Introduce a REST API for users to list files to synchronize to the target environment after a CentOS scan without overlay.
- Users can deploy instances from published machine images to OpenStack.
- Images can be created in PXE format for CentOS.

14.16.2 Enhancements

- Improve UI text and tooltip message for K5 Project ID

14.16.3 Bug Fixes

- 944 Scanning failed at Phase 6 (heap memory in eventcontroller)
- 960 Failed to delete together two or more artifact accounts with error
- 985 Error occurs in the UForge CLI images list if user has generations only from scan
- 1323 Using AWS zone ap-south-1 (Mumbai) with the API raises a publication error
- 1370 CLI command "uforge image list" results in SimpleFacetValueError when the keyboard is "jp"
- 1379 Artifact accounts display bug under IE
- 1419 Package kernel-PAE not recognized as a kernel when doing a black box migration
- 5740 Incomplete Japanese translation of the "Pull a remote file" dialog box
- 6103 Modifying a used software component restriction rule raises an internal server error
- 6108 Support /dev/cciss/cXdXpX disks that exist on HP server
- 6133 AWS connector logs are set to DEBUG and should be set to NORMAL
- 6157 Debian Scan: All the files are uploaded to UForge server when scanning with overlay
- 6162 Tooltip when generating from scan (blackbox) mentions install profile changes
- 6165 Comparing two scans, there are no strike-through on the delete files
- 6235 Uploading boot scripts or my software using IE causes an error if the local directory path is included
- 6261 UForge UI for Windows scan using the command line misses the -p parameter
- 6268 Viewing My Software from Imported Scan raises an unknown server error
- 6385 When a scan source CentOS 7 machine has "/boot/grub/grub.conf", the generated image does not boot

- 6386 Messaging bus consumer breaks down if cloud-init is installed on UForge server
- 6444 Error in NIC API examples, request URI is wrong
- 6501 Deleted package files are recovered after migration
- 6507 Debian migration cannot detect missing info changes
- 6573 "Uploading Archive 0 %" and "Transfer in progress -1 %" are displayed alternately by template import
- 6611 Error message when trying to delete a UForge OS profile milestone not self-explanatory
- 6614 UForge web service response (401 unauthorized) is not RFC compliant
- 6683 Heap memory error when scanning VM with huge files number
- 6753 AWS publish failed in some circumstances
- 6760 uforge-install-config TUI does not appear when using K5 Console
- 6768 Unable to build a package using rpmgen with a file located in /
- 6853 Unable to rebuild RPM, using rpmgen, with hook scripts containing a commented spec file section tag like #%install
- 6903 Outscale Publish final status never reached
- 6906 yum is injected during blackbox migration
- 6917 When cloning an appliance, the parentApplianceUri of the clone is null

14.17 3.7.fp6

Release Date: 2017-07-24

14.17.1 New Features

- Introduce a new deployment feature which allows users to deploy published machine images directly to Amazon, without having to connect to their Amazon account.
- The scan of Windows is optimized by extracting only "used space" from target disks on the source system. "Free space" on target disks will not be copied by the scan.
- Fujitsu Cloud Service K5 Germany region is supported.

14.17.2 Enhancements

- Improved the information displayed in the banner when administering OS Profiles. Now the date the OS profile was created is displayed (for Windows only), as well as the size and the associated distribution (for both Windows and Linux)
- Support Linux multiple disks publication to AWS

14.17.3 Bug Fixes

- 851 After K5 Black box migration, some packages were updated
- 884 uforge-scan.bin ignores option `-e "/"`

- 953 Cannot add a license in a MySoftware
- 972 Internal error happens when clicking directory name `{|}` in Files & Folders in Migration
- 982 debootstrap.log should be preserved
- 1001 Publish Outscale changes proxy instance ID configuration
- 1003 Image generation fails for a CentOS 6.1 imported from scan and upgraded to the 6.5 milestone
- 1061 /etc/ssh/sshd_config file is changed after the migration.
- 1304 All POST and PUT API examples in the documentation miss Content-Type parameter
- 1305 API doc cloudAccount_create needs correcting
- 1306 API doc for creating API key pair needs updating
- 1317 Dashboard quotas are used more than 100%
- 1353 Correct https_proxy that breaks perl cloud (openstack) connectors
- 1393 Remove uforge-anytermd and remove its pid file after uninstalling uforge-install-config
- 1402 Migrator Role does not allow to generate image from a Scan
- 5447 Scanning a UForge server raises an internal server error and a Mapping error
- 6084 Images list Cli command returns wrong OS name
- 6127 The latest `perl-Compress-Raw-Zlib` package is not used
- 6147 Generation fails when selinux packages are manually specified in a MySoftware
- 6148 /etc/ssh/sshd_config modifications are ignored after a blackbox migration
- 7431 Cannot generate debian 8 when /tmp partition is small and has large extra files
- 7806 Scan comparison raises a 500 call failed error when clicking on a package modification of the comparator
- 7842 Cloud account password is saved as plain text in text file
- 7915 cleanup_tickets.sh and cleanup_scans.sh do not delete Generated Images from Scans
- 7986 The UI in Stack u003e OS profile displays the latest version of the OS packages instead of the one used in the appliance template

14.18 3.7.fp5

Release Date: 2017-06-12

14.18.1 New Features

- Add a mechanism to restrict the usage of a project (for Administrator) or software bundle (in MySoftware for users) based on a Distribution name, family, architecture or for an output format
- Administrators can now create a golden image from the UI. Once a Windows scan is complete, the Administrator can import the scan as a golden image. This golden image will be available to users to create new Windows appliances.
- Publication to VMware vCenter improved. UForge now publishes templates to VMware vCenter, rather than instances. The datacenter information (ESXi hypervisor, datastore, network name, etc) is automatically retrieved by UForge and prefilled for publication to VMware VCenter.

14.18.2 Bug Fixes

- 7560 Oracle Linux is treated as RHEL at scan
- 7622 OpenSUSE generation Failed with default OS Profile due to package conflict.
- 7423 CentOS 7.3 VBox image never ends up booting if '/' partition is a logical volume
- 7429 Error message is always logged in oar error log "unary operator expected" when generating Linux image.
- 7361 Windows generation error when disk too small does not raise an understandable error message
- 7620 db_modifs and postupdate modifs applied to several versions of UForge are not handled properly by update_mechanism
- 7758 On the generation UI pages, there is a CSS issue between headers and content
- 7853 License of WS2008R2 is displayed on WS2012R2
- 7771 Hover message on items in scan list is not internationalized
- 7871 Updating the certificate of a google cred account generates a null pointer exception and "this should never happen, please updateTemplateInfo" in the UI
- 7682 Published image tag summary displayed wrong tooltip
- 7635 The type of the password input field of artifact accounts is inconsistent in the UI
- 7584 Applications and Services for Windows are not parsed correctly
- 7767 Missing timezone data on branch master
- 7897 500 error occurs in image generation using a template without a partition table
- 7669 The EventController service does not consume ScanAction event correctly
- 6285 AWS publication is not working behind an external http proxy
- 7630 Outscale publish connector is not working anymore
- 6789 When booting a migrated instance, haldaemon starts although autostart setting is off in the source machine
- 7298 Import/Export Software bundle fails with "Permission denied error"

14.19 3.7.fp4

Release Date: 2017-05-02

14.19.1 New Features

- New user dashboard providing usage statistics and quota information
- New option to scan a live machine without transfering any overlay information (allowing a light-weight audit of the instance)
- Support to create appliance templates based on Windows 2016 operating system
- Ability to scan and migrate Windows 2016 instances
- Application and services information now captured and displayed when scanning a Windows-based instance

14.19.2 Enhancements

- Ability to export an appliance template in either YAML or JSON format (default now YAML)
- Enhanced the information displayed after registering machine images to a cloud environment. machine ID and cloud location (region, zone etc depending upon the cloud target) now displayed in the UI

14.19.3 Bug Fixes

- 7553 A workload based on Scientific Linux cannot be scanned
- 7546 Scanning failed at Phase 6 (heap memory in eventcontroller to the even bus - message too large)
- 7534 Wrong values in /etc/fstab if the appliance has both partition '/' and partition '/boot'
- 7521 `hammr template import` fails for certain types of advanced partitioning tables
- 7500 K5 publication fails with message "Error creating publish command for K5"
- 7436 500 call fail when displaying the detailed information of a scan when i18n is Japanese
- 7403 Windows scan command displayed in the UI is wrong
- 7369 Error badly handled during appliance import if message contains ""
- 7360 Oracle Linux 7 and Scientific Linux 7 machine images do not boot if the appliance templates has logical volumes
- 7340 Scanning a server with a file larger than 40 GB fails
- 7314 Cannot generate a machine image for Fujitsu K5 format from a scan
- 7229 Registering a machine image to AWS fails with Java `PublishCommunicator` error
- 7157 The scan binary ignores option `-e "/"`
- 7153 Scan cannot treat files whose name includes >
- 7147 Docker publish does not work anymore
- 7092 When launching Service Management Tool from `run` -> `services.msc`, an error occurs
- 7071 The check box `Ignore dependency checking warnings` is displayed in the UI when a Windows image is created
- 7063 Inconsistent update of template revision
- 6960 Simultaneous scans of two `CentOS 7.3` machines fails
- 6932 When cloning an imported appliance and exporting, the wrong page is displayed
- 6748 Unable to download a generated machine image via the UI twice
- 5977 When resetting password, the information message to indicate that an email has been sent is badly positioned
- 5907 When inviting a collaborator to a workspace, email textbox is case insensitive
- 5074 Bad vertical aligned text in expandable button

14.20 3.7.fp3

Release Date: 2017-03-21

14.20.1 New Features

- Users can now import a Windows based scan, creating an appliance template. This allows users to update the appliance template prior to migration.
- Users can specify to run `sysprep` as part of a machine image generation for Windows-based appliances that have been imported from a Scan. This allows users to provide a new administrator password as part of the install profile.
- Ability to trigger Repository updates manually via an API call.

14.20.2 Enhancements

- The UI updated to display the language, type and edition of Windows OS profile
- The UI can be customized to allow hyperlinks in the footer or header to either open in a new tab (default) or in the same tab (replacing the UI).
- API Keys now have optional name and description meta-data to help the user identify what API keys are used for.

14.20.3 Bug Fixes

- 7146 Scan cannot treat files whose name includes >
- 6995 The scan status is not updated to `error` when the error occurs during uploading
- 6993 A file or directory name whose include a line feed (LF) is not present in the scan result
- 7069 Upload a logo which is not `png` or `jpg` raise an error but erase the existing logo
- 7065 Incorrect warning message when appliance has multiple disks during generation of some formats
- 7061 Issue when adding PDF files as custom license to project (should not be allowed)
- 7035 `rpmgen -e` (exclude dir list) option is not working correctly
- 7074 MySoftware files are not copied on the filestystem when generating CentOS7 ISO images
- 7024 Windows scan of a machine with 2 disks, when user excludes 1 disk, UForge still creates 2 disks in the scan meta data
- 7067 `uforge org category delete` fails with two arguments
- 7029 Cannot create unformatted logical volumes
- 6939 My profile picture is not displayed on Activity Stream Workspace
- 7048 Search for packages does not take into account hour of the day
- 6873 Amazon publication - S3 bucket is not necessary anymore
- 7009 UForge root password can not be changed wrong message
- 7002 Spider do not cleanup all temporary dirs in /tmp
- 6948 Projects non-native files are ignored if my software has the same name.
- 7003 Windows generation is failing during OS check
- 6998 When exporting a linux appliance without OS Profile an internal servor error is raised
- 6986 After delete a custom license in MySofware or Project , the icon `done.svg` is still visible

- 6971 After K5 Black box migration, Firewall setting changes to enable in Cent OS 6.
- 6970 CentOS 6 scan and generation leads to an error
- 6884 Generation of AWS image for Windows Server 2012R2 fails with illegal seek exception
- 6834 After the migration from 3.5.1 to 3.6, created API keys no longer displayed in the UI
- 6964 Canceling the K5 publication finishes with ERROR message.
- 6961 Incoherence in template and mysoftware revisions when sharing to workspace
- 6963 Internal generation tools must generate the correct guestOS inside vmx when windows+vmware
- 6747 An image can be downloaded more than once by using the URL with same Download ID
- 6855 Cannot retrieve directory from remote site with http basic authentication in software library.
- 6794 Documentation mentions copyright in customisation but copyright is not displayed
- 6870 A generated CentOS 6.8 image kernel panics if it has a logical volume in the partitioning table
- 6815 Cannot pull files from FTP in MySoftware.
- 6875 When uploading a file for the second time the confirm popup has two handlers and so the action is carried out twice
- 6872 Success message for `org os add` is not correct
- 6800 Cannot download non-cached software artifact correctly if the remote file size has been changed.
- 6819 While scanning Windows OS, Scan progress is continued to copy on the clipboard.

14.21 3.7.fp2

Release Date: 2017-02-13

14.21.1 New Features

- Support registration of machine images for Azure ARM and Azure Enterprise Accounts
- Support for Ubuntu 16.04
- Ability to register docker images built in UForge to DockerHub. This includes managing credential information to authenticate against DockerHub.
- In `Projects` or `My Software` can now provide restrictions to determine if they are compatible with a particular OS family, type or version.

14.21.2 Enhancements

- Renamed `VM Builder` Tab in the UI to `Apps`.
- Better internal logging information when publishing/registering machine images to a target cloud environment.
- Better validation in the web service for information used in publishing/registering machine images.
- Better UX experience when managing and choosing `pinned` (or `sticky`) packages.
- UI now displaying the size of the generated machine images.

- Can now delete an invitation of a user to a Workspace if a user has invited someone to join a collaboration workspace, and the person is not responding, there is no way to cancel the invitation.
- Added an `Id` column for all UForge CLI commands that lists information (for better referencing in other commands).
- Added the ability to reset a user's password via the UForge CLI (`--resetPassword` option).

14.21.3 Compatibility Issues

Migrating to UForge 3.7-2 will have the following compatibility issues:

- any Windows golden image that use a non-standard Edition (for example `Windows K5` instead of the official `Standard`, `Enterprise`, `Webserver` or `Database`) will be changed to `Standard` edition. A warning will be added to the log files. If you would like to change the Edition of the golden image, you should re-register the golden image with `org golden create` command.

Warning: Fujitsu is not legally responsible for any damage or loss caused by the possible inconsistency between the assumed and the actual Editions.

The following API interface and calls have been modified:

- The object `DistribProfile` is now an abstract object and is implemented by either linuxProfile-object or windowsProfile-object (which are new object types).
- The deprecated object `DistribProfileTemplate` has now been deleted. The object distribProfile-object is now used. The attribute `standardProfileUri` is now deprecated and been set to `null`.

Due to the above object changes, the following API calls have been modified:

- orgOSWindows-add
- orgOSWindows-delete
- osTemplate-getAll
- osTemplate-get
- orgOSWindows-getAll

The following API calls have been added to enhance scanned Windows-based workloads:

- workspaceTemplateOSApplications-get
- workspaceTemplateOSServices-get
- workspaceTemplateOSPartitionTable-get

14.21.4 Bug Fixes

- 6853 While scanning Windows OS, Scan progress is continued to copy on the clipboard.
- 6821 Blob name must finish with `.vhd` and add some information in the publish popup.
- 6820 Issues in properties i18n file.
- 6809 OpenStack account turned into another type of cloud account after the migration from 3.5.1 to 3.6.
- 6706 Fix backward compatibility for golden edition with custom names.
- 5607 Even if the scan ends the UI continues to ask for information of the scan.

- 6737 Sub menu scrollable inside the Dashboard.
- 6734 Cannot delete template with software component from workspace.
- 6732 Unexpected scroll bar in My Software view.
- 6716 Cannot download rpms from yum repos whilst scanning a centos system.
- 6713 Error message containing typo for windows disk size.
- 6711 Golden location is retrieved from Pkgs table, it should be retrieved from WindowsProfile table.
- 6672 Scan does not read KEYBOARD in metadata.
- 6646 File conflicts against packages built with when installing centos distribution packages.
- 6639 Primary disk size is changed to the other disk size on UI when having multiple disks.
- 6627 Cannot export a template if the software component has rpm file in Repository Packages tab.
- 6614 Creating folder failed but displayed on UI.
- 6599 i18n properties breaking master build.
- 6596 Imported appliances from archive are not counted statistics in Dashboard.
- 6529 Image generation fails when a template includes rpm file with no cached.
- 6497 Can't display `Projects` as guest user.
- 6495 The `org golden xxx` command fails if edition name in db is not allowed.
- 6492 Badly formed error label for Credentials Microsoft Azure.
- 6480 Spelling mistake retrieving remote path and error message shown.
- 6478 Sharing a template in collaboration, including software that does not use the cache of the fetch, raises an Internal Server Error.
- 6460 Imported appliances are counted as created on statistics in Dashboard.

14.22 3.7.fp1

Release Date: 2017-01-09

14.22.1 New Features

- Multi-NIC support for Linux based appliance templates.
- Driver injection improvements (internal mechanism) for Windows-based appliance templates.

14.22.2 Bug Fixes

- 6326 Impossible to publish an `OpenStack VDI` image
- 6323 Cloud account name appears twice in the public informations in UI for all Cloud formats
- 6234 Sticky package of imported template is not shown in the UI
- 6141 User gets a 500 call failed if a custom target platform has been added but not enabled specifically for the user
- 6042 OS packages are not sortable in the `Repository` column

- 6237 Spelling mistakes in the API docs
- 6222 Format enabling/disabling not working when updating the UI config
- 6453 Impossible to generate image when install profile contains users
- 6199 Migration fails because the user ID taken from a scan and user ID that the package makes overlap.
- 6409 OE-lite can't fetch QT source file
- 6206 Filter inactive pkgs on `DistributionPackages.getAll()` method
- 6200 Scanning a disabled OS is possible
- 6190 Scanning an azure vm with advance partitioning : install profile partioning not correct
- 6180 Errors outputted into `/oar/oar_scan_job*.stderr` when scanning CentOS 6
- 6154 Launching windows scan binary from command line with API key does not launch the scan
- 6134 Pkg overlay archive are built differently if a black box migration is done first or if it's a scan import to appliance
- 6309 Several concurrent generations could fail if there are uncached software bundles files in it
- 6211 Creating a two bootscripts with same name does not show an error message
- 6194 Japanese Characters are OK to use but encoded incorrectly for `Tag` and `Maintainer` fields of a software component
- 6193 Same rpm file can be uploaded without overwritten to a software component
- 6178 Errors outputted into `/oar/job_finalize.log` when generating CentOS image
- 6169 Total Disk Usage doesn't count the size of files uploaded to software components
- 6027 Exported template has lost some information on MySoftware
- 6346 WARP should skip to inject uforge agent in the specific condition
- 6327 Scripts are not imported when sharing a template in a `Workspace`
- 6057 Yum update error `uforge_update.sh: line 660: [: too many arguments`
- 6055 The volume shadow copy is not deleted after scan of Windows.
- 6007 Code in `distrotools/lib/str.[c|h]` in function `repl_str()` cannot compile for windows using `mingw c++`
- 6440 Can't display `Projects` as guest user
- 6453 Impossible to generate image when install profile contains users

14.23 3.7

Release Date: 2016-12-27

14.23.1 New Features

None (released based 3.6-fp2)

14.23.2 Bug Fixes

- 6537 Removed AMI format for AWS S3
- 6521 Launching windows scan binary from command line with API key does not launch the scan
- 6517 Impossible to know which publish image on UForge corresponds to which Image in K5 portal
- 6515 CentOS 6 images can be accessed with SSH on K5
- 6513 Validation for K5 publish view is not properly handled
- 6511 Launching uforge-scan.exe from command prompt still fails if the file path includes Japanese characters
- 6507 The `uforge-install-config` binary for windows does not start because `uforge-install-profile-1-1.noarch.zip` does not contain the correct directory structure.
- 6505 The `no_console` file is not created for Windows.
- 6504 Problem with OpenDJ port 4444 usage in several UForge config scripts
- 6503 The `uforge.conf.ORIG` contains plain passwords with very weak permission
- 6502 AWS connector uses a fixed size 3.4 GB disk and publication fails for larger images
- 6422 Uploading an avatar image twice, the first image is still used
- 6410 Loading page empty during 5 seconds for the first time in `Sofware Library` view
- 5897 If a space is used in cloud accounts in openstack in the URL, then an internal error is observed
- 5849 Displaying the logo in view package details of a target format is not displayed
- 6488 Impossible to generate image when install profile contains users
- 6362 AWS resource connector no longer work due to credential changes
- 6064 The CLI command `org repo update` returns exception if `--type` param value is invalid.
- 5900 Generation sometimes fails if the second disk of the appliance is too small

14.24 3.6.fp2

Release Date: 2016-12-05

14.24.1 New Features

- Fujitsu K5 support. Can now register machine images generated on the platform to Fujitsu K5.

  Note: The following operating systems are supported for the moment (others will be supported soon):

  - CentOS 7.0
  - Ubuntu 14.04

- SELinux support when creating appliance templates and during migration
- Docker machine image generation support. This allows users to build docker base images.
- When scanning Windows machines, the scan report now includes the services detected.

Note: The platform does not support the comparison of windows-based scans at this time.

14.24.2 RFEs

- Better progress status when scanning Windows machines
- Less restrictive validation of website information in the MySoftware/Project Overview
- New icons for 'pull' and 'upload' for software/project files management
- Added directory icon when displaying all the files for software/project files view
- When deleting a folder, the confirm message should be more explicit (that all sub folders and files will also be deleted)
- Better explanation of the "cached" option for software/project files in the UI
- Managing licenses for software/project components; there is now an explicit delet button to remove an uploaded license file

14.24.3 Bug Fixes

- 6123 Publishing a generation from a scan results in 500 error in UI
- 6089 Member's role on workspace couldn't be changed if language is set as French or Japanese
- 6017 Canceling from Appliance Create no longer returns to previous page
- 5946 Publishing to CloudStack fails with the next error: vhd.gz: No such file or directory
- 5942 RHEL is added despite launching *org os add* for Oracle Linux or Scientific Linux with cli
- 5909 User ID and group ID of the install profile can be set 0
- 5906 UserResourcesAccessRights database mapping not proxied
- 5896 Deployment fails due to NIC settings
- 5892 Deployment fails when using eth1
- 5843 "org category delete" raises an error
- 5777 Launching uforge-scan.exe from command prompt fails with an error if the file path to the binary includes Japanese characters.
- 5762 Cannot register the third disk with a VirtualBox image
- 5756 New users, the defaukt country is: Abkhasia
- 5754 opening the Dashboard > Generations page first shows progress bar for all publications
- 5752 Number of MySoftware components not properly refreshed in the UI
- 5750 Number of Appliance not properly refreshed in the UI
- 5748 The diskusage of "uforge user quota list" is displayed by byte
- 5684 Invite the same user in the collaboration members list does not show error message
- 5676 Duplicated variable in /etc/default/grub if distribution provides default values.
- 5647 Keyboard and kernel parameters are not taken into the scan report on CentOS 7 scan.

- 5635 Broken incremental scan for windows 2012R2
- 5627 Cancelling scan via ctrl+c is not correctly displayed in the UI
- 5625 uforge-scan does not respect bandwidth limit
- 5623 When the image of CentOS7 is generated, RPM-GPG-KEY-CentOS import read fails
- 5621 rpmgen fails to build package if file path in %file includes space.
- 5570 Impossible to delete an incremental scan
- 5562 UForge CLI accesses to interactive mode even if the user or password are wrong
- 5560 The input value of the activation key is not saved in a Windows appliance
- 5342 Scan incremental with Ubuntu does not appear in UI
- 5265 No dialog box displayed after running an instance on Azure

14.25 3.6.fp1

Release Date: 2016-10-31

14.25.1 New Features

- Import/Export of appliance templates in the user interface
- **Software (MySoftware) and Project bundles now consolidated. New features added including:**
  - pulling files from remote locations (HTTP, FTP endpoints) so the user no longer requires to upload software components to the platform
  - pulling files can be cached for future generations or pulled on each generation
  - file permissions added for files and directories
  - can create directory structures in a software bundle
  - can add tagging information to a software bundle
  - can add native packages from OS respositories to a software bundle
  - can add boot scripts to a software bundle
  - identify the software bundle only being supported on a subset of operating systems
- API keys can be used for authentication when running a scan for migration.
- Scan messages and error messages cleaned up and more understandable
- Japanese language localization for the UI

14.25.2 Bug Fixes

- 5293 Image generation error: Windows image must have at least 512 MB of memory
- 5729 Issues with migration from 3.5.1. to 3.6
- 5465 Build fails due to unreachable rpm-4.11.2.tar.bz2
- 5740 Fix DB schema checks

- 5331 AWS publish no longer works
- 5637 Windows generation from scan fails at boot
- 5427 Unable to generate a virtual machine with LVM inside a MSDOS disk
- 5291 All combo boxes are empty when a value has been selected
- 5876 Logo broken on Dashboard
- 5444 Unable to populate Fedora/RHEL distributions
- 5420 When a template is remobed from a workspace, a DELETE error appears in the log file
- 5527 Subscription info does not list the frequency of quotas
- 5494 Scan fails because of files of type c (character device file)
- 5483 The service command not found in a machine generated by UForge
- 5442 The file deletion of Project fails
- 5429 Root can disable root account in UForge CLI
- 5746 Timeout of 10 seconds for the UForge CLI is not usable
- 5563 Internal error in Migration tab
- 5558 500 Call Fail Error when generating an image from scan
- 5556 The targetformat of Amazon is not displayed when generating an image
- 5553 If a scan is deleted, the image generated from the same scan is not deleted
- 5551 Spelling mistake in UI when publishing to Flexiant
- 5549 The error of Keystone version is displayed in Keystone Server URL
- 5403 Scan fails when trying to rebuild a non repo package

CHAPTER 15

Trademarks

UForge is a registered trademark of UShareSoft, a Fujitsu company.

LINUX is a registered trademark of Linus Torvalds.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

Apache Ant, Ant, and Apache are trademarks of The Apache Software Foundation.

Red Hat Enterprise Linux is a trademark of Red Hat.

UNIX is a registered trademark of the Open Group in the United States and in other countries.

Other company names and product names are trademarks or registered trademarks of their respective owners.

CHAPTER 16

CHAPTER 17

High Risk Activity

The Customer acknowledges and agrees that the Product is designed, developed and manufactured as contemplated for general use, including without limitation, general office use, personal use, household use, and ordinary industrial use, but is not designed, developed and manufactured as contemplated for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could lead directly to death, personal injury, severe physical damage or other loss (hereinafter "High Safety Required Use"), including without limitation, nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system. The Customer shall not use the Product without securing the sufficient safety required for the High Safety Required Use. In addition, FUJITSU (or other affiliate's name) shall not be liable against the Customer and/or any third party for any claims or damages arising in connection with the High Safety Required Use of the Product.

CHAPTER 18

Export Restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.