UForge AppCenter Admin Documentation

Release 3.5.1

FUJITSU

February 01, 2017

Contents

This documentation is intented 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

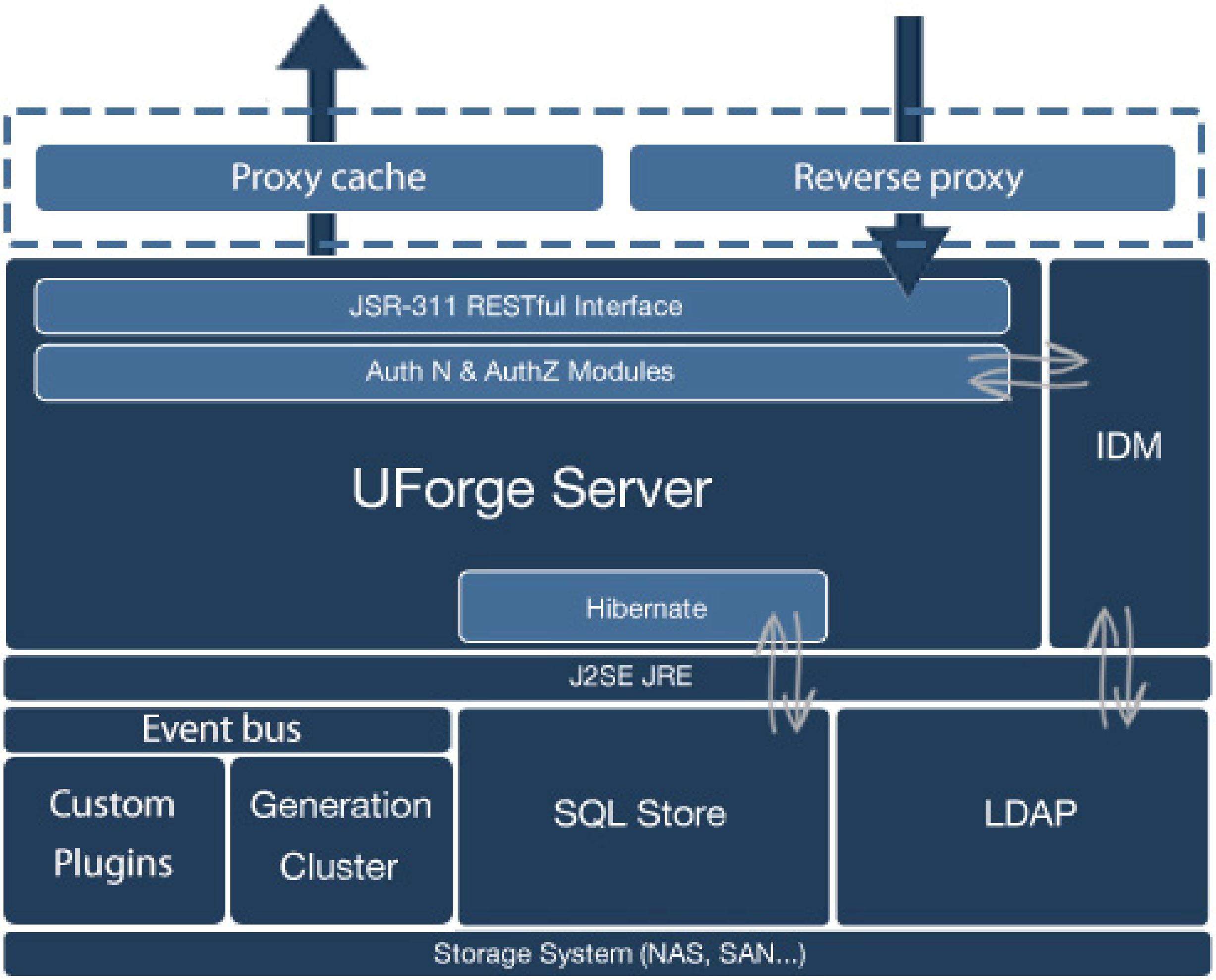
The following sections describe in detail things to consider when deploying an UForge platform.

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.
- LDAP Service – LDAP holding user authentication and access information
- IDM Service – Authentication and authorization module
- Generation Cluster – A grid engine for scheduling and executing image generations
- 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 Orientated 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 Glassfish (v3) 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 licence). 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 Glassfish (v3) 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.

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

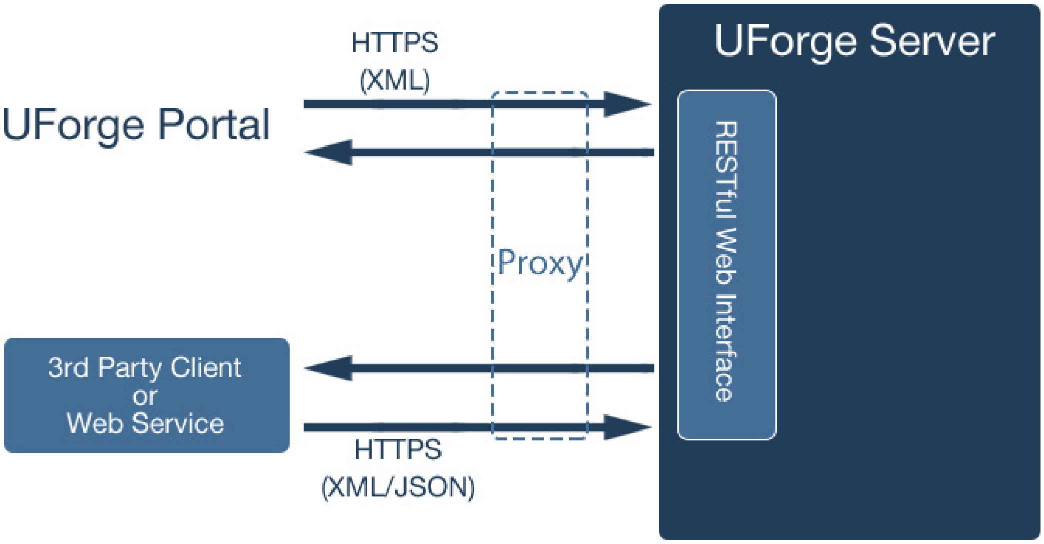
By default MySQL is used as the SQL Store.

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.

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.



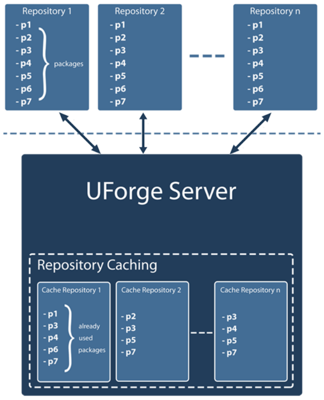
UForge allows users to implement their own clients to interact with UForge. UForge provides a RESTful API, allowing businesses to create a mashup and expose certain functionalities of UForge in their websites and portals. UForge provides two SDKs:

- 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 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
- Storage: 2 TB. Note this depends on the number of OS you want to populate and the number of images generated and stored.

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 fail.
- 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, LDAP, storage and generation cluster (DMZ).

1.4 Storage Considerations

The UForge platform requires a significant amount of storage. During the configuration it is important to size the local storage, the various node instances of the deployment and, if any, the size of the shared remote storage. You will need both a local and a remote file storage.

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

> **Warning:** Logs for UShareSoft, Glassfish, OpenDJ, Tomcat, and syncope are stored under /opt and NOT under /var. Under /var/log you will find a symlink to the /var location where the logs are stored. Therefore you should allow enough space under /var for these logs.

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 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.

The size of projects also vary, however 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).

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 Example

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 operating systems offered by UForge
- 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.

1.9 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:

UShareSoft recommends 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.

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	2 GB	2	15 GB	
Compute Node	4 GB	2	100 GB	If you have a good NAS/SAN you reduce the disk space to 15 GB if you map the directory /space/REPOS with the NAS/SAN
SYNCOPE Web Service	2 GB	2	15 GB	
Database	4 GB	4	100 GB	If you have a good NAS/SAN you 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		

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	25 Go	If you have a good NAS/SAN you 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 Go	

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 UShareSoft recommends 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

UForge Installation

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 AppCenter, 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. To make the UForge installation process as easy as possible, UShareSoft provides an external hard drive `UForge Setup Disk` that contains:

- UForge Operating System Repository – all the packages and updates for each open source operating system distribution supported by UForge. For other operating systems including RedHat Enterprise Linux, you must have access rights to the operating system repositories
- Default Projects Catalog – all the 3rd party software that UShareSoft has packaged for its own Online platform Project Catalog
- UForge AppCenter Installation Images – ISO image or VM images to install the UForge services

2.1.1 Installation Steps

UForge can be split into three distinct phases:

Step 1: Copy the operating system licences, logos, and ISO installers (that you wish to be supported in the UForge AppCenter) into the storage space that will be used as the UForge Repository.

Step 2: Install the UForge AppCenter (either from the ISO or pre-created VMs) for each node comprising the platform.

Step 3: Configure UForge AppCenter using the UForge Deployment Wizard.

UForge is delivered in one of three forms:

1. OVF (ESXi hypervisor) VM. This contains all the UForge AppCenter binaries to allow you to provision UForge in VMware vCloud Director or equivalent.
2. VHD VM (Xen hypervisor). This contains all the UForge AppCenter binaries to allow you to provision UForge in Citrix CloudStack (XenServer-based deployments)
3. ISO image. This contains all the UForge AppCenter binaries to allow you to carry out a bare-metal installation or to install into a virtualized or cloud VM instance.

These images can be found in the UForge Setup Disk. Choose the image type most appropriate to the hardware, virtualized or cloud environment you will be deploying UForge to.

Note: If you want to use partitioning on the deployment nodes, then use the ISO image. This gives greater flexibility in sizing correctly the disks on each node depending on their role (for example, an application server node from a database node).

2.1.2 Installation Checklist

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

- UForge Setup Disk
- Your activation credentials (ID and activation key) provided by UShareSoft
- 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)

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

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 and default project binaries 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
- on a local storage

2.3 UForge Repository on Shared Storage

When using shared storage NFS 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).

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)
```

4. Check the mount points:

```
mount 192.20.777.205:/appstore_qualif_data/ /mnt
su - glassfish
cd /mnt
```

5. Execute the command:

```
touch test
```

6. If it returns:

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

Then execute the following commands (as root)

.. code-block:: shell

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

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

```
cd ~
umount /mnt
```

8. Once these directories have been exported, use the UForge Setup Disk to copy all the operating system information to the DISTROS sub-directory. To mount the shared directory on the machine where the UForge Setup Disk is plugged to:

```
        mount <nfs server ip address>:/volume1/DISTROS /mnt
```

Then copy the DISTROS information from the Setup Disk to the shared directory:

```
    rsync -a --progress <my disk>/DISTROS/ /mnt/DISTROS/
```

Note: It may take up to an hour to copy all of the data.

2.4 UForge Repository on Local Storage

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
3. Now copy all the operating system information from the UForge Setup Disk to the DISTROS sub-directory:

```
    //# rsync -a --progress /<disk directory>/DISTROS/ /space/DISTROS/
```

Note: It may take up to an hour to copy all of the data.

2.5 Installing from an ISO

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 to have a DVD disk drive).
2. Boot the system from the ISO.
3. Choose the default menu proposed by the installer.
4. Choose the keyboard layout you want to use.
5. Set the timezone.
6. Set the root linux password.
7. Wait for the install to finish, then reboot the system and detach the ISO from the VM or remove the DVD from the disk drive.
8. Once the system reboots, accepts the EULA license agreement.

For a multi-node installation, repeat these steps for every physical or VM instance you wish to install.

The installation phase is complete, you are now ready to configure the UForge platform, see *Configuring UForge*.

2.6 Installing from a VM Template

A number of virtual machines have already been prepared by UShareSoft to make it easier to install quickly UForge on a virtualized or cloud environment. If your environment does not support the proposed VM formats, then install UForge using the ISO image, as described in *Installing from an ISO*.

To install UForge from a VM template.

1. Import the VM to the cloud datastore or public template library.
2. Use the cloud or virtualization management platform to instantiate one or more instances of the VM template.
3. Recuperate the console of the newly created VM instance.
4. Accept the licensing agreement.
5. Choose the keyboard layout you want to use.
6. Choose the UNIX root user and guest user passwords.
7. Set the time zone of the VM.
8. Choose `Accept All and Continue`.

The installation phase is complete, you are now ready to configure the UForge platform, see *Configuring UForge*.

2.7 Configuring UForge

Once the installation is complete on all the nodes you wish to use for the UForge platform, you are now ready to configure all the UForge platform services. This is done via the UForge Deployment Wizard that helps guide you through the final steps of the installation process.

Note: The LVM VolGroup Name name should 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.

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 five characters long, including alphanumeric characters and the following special characters !#$%&'()*+,-/:;<=>?@[]/^_{|}~`. Spaces are not allowed at the beginning or end.
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 UShareSoft.

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. Decide whether you would like to populate UForge with UShareSoft's default project list. This list must already be copied to the UForge repository in the storage used for this deployment.
8. 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.
9. 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.
10. Click `next` to continue.
11. Enter the Web Server IP address, external hostname and the database IP address.
12. 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 Portal.

    > 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. glassfish:glassfish (see :ref: *repository-shared-storage*, step 4, Check mount points, for more detail).

13. Indicate the generation cluster compute node hostnames. You can add additional compute nodes by clicking the add button.
14. 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.

15. 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 though 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 over. To do this:

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

2.8 Testing the Deployment

Once the configuration phase is complete, you may wish to carry out some basic sanity tests to ensure that the UForge platform 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 form the uforge.conf and add them to some environment variables (you could also just manually view the uforge.conf)

```
eval `grep '^UFORGE_WEBSVC_\|^UFORGE_GF_INTERNAL_IP\|^UFORGE_GF_HTTP_PORT\|^UFORGE_GF_WEBSVC_ROOT_CON
```

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

```
# curl http://$UFORGE_GF_INTERNAL_IP:$UFORGE_GF_HTTP_PORT/$UFORGE_GF_WEBSVC_ROOT_CONTEXT/users/$UFORG
--verbose

* About to connect() to 10.0.0.240 port 9090 (#0)
*   Trying 10.0.0.240... connected
* Connected to 10.0.0.240 (10.0.0.240) port 9090 (#0)
> GET /ufws-3.0/users/root HTTP/1.1
> User-Agent: curl/7.20.1 (x86_64-redhat-linux-gnu) libcurl/7.20.1 NSS/3.12.8.0 zlib/1.2.3 libidn/1.1
> Host: 10.0.0.240:9090
> Accept: */*
> Authorization:Basic root:welcome
>
< HTTP/1.1 200 OK
< X-Powered-By: Servlet/3.0 JSP/2.2 (GlassFish Server Open Source Edition 3.1.1 Java/Sun Microsystems
< Server: GlassFish Server Open Source Edition 3.1.1
< Last-Modified: Thu, 03 May 2012 08:32:18 GMT
< ETag: "ef286bf07b8d18928287e12cb122ccf2"
< Content-Type: application/xml
< Content-Length: 5477
< Date: Thu, 03 May 2012 08:33:11 GMT
...<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 that the database table is present. The Percona Server instance should have "usharedb" and "oar"

```
# service mysql status
MySQL running (22661)                                          [  OK  ]
```

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

```
# eval `grep '^UFORGE_DB' /etc/UShareSoft/uforge/uforge.conf`
# echo "show databases" | mysql -f -N -u $UFORGE_DB_LOGIN -p$UFORGE_DB_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
```

This should return without any output.

CHAPTER 3

Further Configuration Overview

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

3.1 Adding a Compute Node

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

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

   Name oarnodeX and IP-oarnodeX respectively the name and IP address of the new node to be added to the UForge "cluster".

3. Copy over /etc/hosts from oarnode1 to oarnodeX with mods
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 IP-oarnodeX dd of=/etc/hosts
```

7. Copy over uforge.conf file and others

```
rsync -a /etc/UShareSoft/uforge/uforge.conf IP-oarnodeX:/etc/UShareSoft/uforge/
rsync /etc/oar/oar.conf IP-oarnodeX:/etc/oar/
rsync /etc/ssh/sshd_config IP-oarnodeX:/etc/ssh
rsync -a ~oar/.ssh IP-oarnodeX:~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 'BEGIN{d=1}/\<oarnodeX\>/ {d=0}{print}END{if(d==1){print "IP-oarnodeX
oarnodeX"}}' /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. 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
for s in ntpd glassfish 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
```

10. Run the following commands on oar-server to create new resources on oarnodeX from existing oarnode1 resources

```
/usr/bin/oarnodes | /bin/awk '/network_address=oarnode1/
{s=$0;gsub(".*nature=","",s);gsub(",.*","",s);printf "/usr/sbin/oarnodesetting
-a -h oarnode3 -p cpuset=0,nature=%s\n",s}' | sh
```

3.2 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
```

3.3 Modifying the UForge IP Address

To modify the UForge IP address or hostname, do the following.

1. If UForge is installed on a single node, launch a terminal on the UForge machine.
2. Use system-config-network to modify hostname <MY-NEW-HOSTNAME>. It should also allow you to change the host name (that is the fully qualified domain name like ufiab.mycompany.com). This will apply the changes automatically to the directories in /etc/sysconfig
3. Ensure that the changes are applied in the following locations, otherwise make the appropriate changes:

- /etc/sysconfig subdirectory files like ifcfg-XXX where XXX is your main interface name (like eth0)
- /etc/sysconfig/network for HOSTNAME and GATEWAY
- /etc/resolv.conf for your DNS configuration
- /etc/hosts for all host definitions
- /etc/fstab for remote NFS mounts
- **/etc/UShareSoft/uforge/uforge.conf for the following entries:**
    - UFORGE_PROXY_INFOS
    - UFORGE_EXTERNAL_HOSTNAME
    - UFORGE_IAAS_DOWNLOAD_URL
    - UFORGE_NOREPLY_EMAIL
    - UFORGE_URL
- /etc/udev/rules.d/70-persistent-net.rules

4. Modify the header, footer, IP information in /var/opt/UShareSoft/uforge-client/gwt/uforge/templates/config.xml (except for <c:uForgeUrl>) with the EXTERNAL_URL variable. For more information on modifying the portal information, refer to *Rebranding Your UForge GUI*.

5. Launch the following two scripts:

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

If you have a load balancer where rules have been entered for accessing the UForge web service make sure the URIs match the following parameter in uforge.conf:

```
UFORGE_GF_WEBSVC_ROOT_CONTEXT=ufws-3.3
```

For the UForge CLI you should have:

```
UFORGE_GF_ADMIN_WEBSVC_ROOT_CONTEXT=ufadmws-3.3
```

3.4 Configure Apache and GlassFish 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 (GlassFish) have yet been configured to use a SSL certificate and allowing HTTPS.

To configure both servers to use an SSL certificate:

1. Log in as root to the machine running the UForge Apache and GlassFish Web Services
2. Copy the SSL certificate files locally to the machine. Note that you should have three to 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)

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.

3. 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

4. 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/
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
```

5. (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.

6. 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)
    ---
```

7. Go to the glassfish configuration directory:

```
    cd /usr/glassfish-3.1/glassfish/domains/*/config/
```

8. Save the original keystore with the correct permissions and groups:

```
    rsync -a keystore.jks keystore.jks.ORIG
```

9. Delete the current certificate:

```
    /usr/java/latest/bin/keytool -delete -alias s1as -keystore keystore.jks -storepass <admin pa
```

Note that the default admin password for a standard GlassFish installation is changeit. So the default command to run is:

```
/usr/java/latest/bin/keytool -delete -alias s1as -keystore keystore.jks -storepass changeit
```

10. Convert your certificate pem files to one pkcs#12 file:

```
    openssl pkcs12 -export -in <SSLCertificateFile CA chain (pem) path> -inkey <SSLCertificateKe
```

So in our case:

```
openssl pkcs12 -export -in server_CA_chain.crt.pem -inkey server.key.pem -out keystore.pkcs12 -n
```

11. Import the newly generated pkcs12 file:

```
    /usr/java/latest/bin/keytool -importkeystore -srckeystore keystore.pkcs12 -srcstoretype pkcs
```

Giving:

```
/usr/java/latest/bin/keytool -importkeystore -srckeystore keystore.pkcs12 -srcstoretype pkcs12 -
Entry for alias s1as successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled
```

12. Restart the application service:

```
    service glassfish restart
```

13. Verify the certificate:

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

Or you can also use same openssl client command than for the Apache server used in step 6.

To verify that the new certificate is correct and if the GlassFish 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 a 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 has been done you can configure SMTP proxy as follows. In /etc/UShareSoft/uforge/uforge.conf you can modify:

- UFORGE_RELAY_HOST=
- UFORGE_RELAY_PORT=
- UFORGE_RELAY_USER=
- UFORGE_RELAY_PASSWORD=

These can be empty

Then run:

```
/opt/UShareSoft/uforge/conf/postfix_configure.sh
```

3.6 Configuring the Web Service

Each UForge Web Service instance runs inside a Glassfish 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: 9090)
- HTTPS port (default: 443)
- Administration console port (default: 4848)
- Administration console credential information (user and password)
- Root context of the web service (for example ufws-3.3)

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
```

For more information on Glassfish, see http://glassfish.java.net

3.7 Configuring the Database

UForge uses the PerconaDB 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.

The PerconaDB 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)
- Port number (default: 3306)

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
```

For more information on PerconaDB, see http://www.percona.com

..note:: By default no mechanism is configured to backup the contents of the UForge database. PerconaDB 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.

3.8 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

3.8.1 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
resource_id : 1
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO, nature=0, network_address=isc

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,
```

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: 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

3.8.2 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` allowing to generate an image:

```
# 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,
```

To change a current resource (resource_id: 92) to a different job type (for example publish images):

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

3.8.3 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>
```

3.9 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_tickets.sh
10 3 * * * /opt/UShareSoft/uforge/cron/cleanup_scans.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
```

3.9.1 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, it happens, following network issue or other, that nodes 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 could be changed.

```
10 2 * * * /opt/UShareSoft/uforge/cron/cleanup_tickets.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 recommend to execute it when there is not a lot of activity on the platform. UShareSoft has 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 so big and if there are a lot of images created and deleted per day, it could be a good idea to launch it several times a day.

```
10 3 * * * /opt/UShareSoft/uforge/cron/cleanup_scans.sh
```

Same as for /opt/UShareSoft/uforge/cron/cleanup_tickets.sh

```
15 3 * * * /opt/UShareSoft/uforge/cron/update_uss_pkgs.sh
```

This updates the UShareSoft packages that are introduced in images like uforge-install-config. The schedule can be changed.

Updating once a day or on demand is good enough updates are not provided frequently.

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

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

In all the cases, it is important to keep these repositories up to date so that:

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

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

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

Since UForge 3.5.1, 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 rhel : 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 stick package (version 1.2.3) on NTP. Let’s consider ntp is on a repository that remove packages. You generate a machine image, ntp gets downloaded. update_repos_local_cache.sh. 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, neither 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, and as it is a basic Linux command, 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 effects: usually, issue with memory)

3.10 Tuning the Services

You can set the priority of tasks on the UForge platform in the file uforge_services.conf

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.

3.10.1 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 deals 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:PermSize=256m",
    "-XX:MaxPermSize=1024m"
  ],
  "sys-properties": {
    "jna.library.path": "/opt/UShareSoft/uforge/lib"
  }
},
  "nice": {
    "niceness": 15
  }
}
```

3.10.2 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)

3.10.3 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 Email Notification Service

The UForge platform provides an email notification service 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 in the Marketplace (App Store) 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 "watching" in the Marketplace (App Store), emails are sent when the template is updated or a new comment has been added to the template.

3.11.1 Changing Email Address for Notifications

During the initial install of the UForge platform, 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 platform. 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. Run the script to force UForge to use the new uforge.conf file, this will restart certain UForge services:

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

3.11.2 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)

Marketplace notification emails:

- `AppStoreNotificationNewComment.tmpl` – email sent to the user watching the template where a new comment is added
- `AppStoreNotificationTemplateNew.tmpl` – email sent to the user watching the template where a new version of the current template is available
- `AppStoreNotificationTemplateUpdate.tmpl` – email sent to the user watching the template where the template is updated by the publisher
- `AppStoreTemplateReportAbuse.tmpl` – email sent to the administrator when a user reports an abuse for a comment in a template
- `AppStoreTemplateReportAbuseConfirm.tmpl` – confirmation email sent to the user reporting an abuse

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

Specific keywords for Marketplace (App Store) email notifications:

- #COMMENT# - The comment added to a template in the Marketplace
- #REPORTER# - The user name of the person carrying out the request (used for adding a comment or reporting an abuse)
- #REPORTEREMAIL# - The user name of the person carrying out the request (used for adding a comment or reporting an abuse)
- #NBREPORTEDABUSE# - The total number of times this comment has been flagged as abusive
- #CMDACCEPTABUSE# - The command to use to remove the comment from the Template in the Marketplace
- #TEMPLATENAME# - The name of the template in the Marketplace
- #PUBLISHDATE# - The date the template was published in the Marketplace
- #TEMPLATEVERSION# - The version of the template published in the Marketplace
- #ORGNAME# - The organization name where the Marketplace resides
- #PUBLISHEREMAIL# - The email address of the publisher (to the Marketplace)

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 platform.
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/
```

5. Instantiate the following changes by running the following command:

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

3.12 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+(?.*)$", Pattern.CASE_INSEN
    private static final Pattern USER_PASS = Pattern.compile("^(?[^:]*):(?.*)$", Pattern.CASE_INSENSI
    @Override
    public IUserAuthentication filter(ContainerRequest containerRequest) throws UForgeException {
        logger.trace("Getting Auth from Http Header");
        return getUserAuthentication(containerRequest);
    }
    protected IUserAuthentication getUserAuthentication(HttpRequestContext containerRequest) throws U
        /*
        * 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
        */
        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, targetUse
    }
    protected IUserAuthentication getUserAuthentication(URI requestUri, String userName, String pass
        /*
        * 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 {
...
}
```

3.13 Allowing https Repositories with Self-Signed Certificate

Note: The use of self-signed certificate can comport 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 CERTIFI
```

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`.

3.14 Populating Database with OS Packages

Open source operating system versions are taken from the official repository mirror or the UForge repository cache. Proprietary operating systems such as RedHat 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 UShareSoft packages must be populated for this operating system.

Note: Custom repositories are supported in UForge. They are treated like other OS packages.

For information specific about Windows, refer to First Installation of Windows.

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.

In order to add an operation system in your UForge platform you must:

1. Create the OS in the organization.
2. Create the repository. This includes the official repository as well as the specific UShareSoft tool repository.
3. Link the distribution to the repository.
4. Launch spider to fill the repository with the packages.

The following is a concrete example to begin the population of CentOS 6.5 32bit:

1. 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
```

2. Run the following CLI command in order to create the distribution:

```
uforge org os add --name CentOS --arch i386 --version 6.5 -u $ADMIN -p $PASS
```

3. Enable the new operating system for the organization. The following command enables CentOS 6.5 32bit in the default organization:

```
uforge org os enable --name CentOS --version 6.5 --arch i386 -u $ADMIN -p $PASS
```

4. 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 i386 -u $ADMIN -p $PASS
```

5. 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 UShareSoft official repository as shown later.

   Centos 6.5 repository:

   ```
   uforge org repo create --name "CentOS 6.5 os" --repoUrl http://vault.centos.org/6.5/os/i386/

   Success: Created repository with url [http://vault.centos.org/6.5/os/i386/] to default orga
   ```

   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://`

   Only use `--officiallySupported` 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 the variable `--officiallySupported`.

   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.

   To populate the distribution using the official UShareSoft repository for Centos:

   ```
   uforge org repo create --name "CentOS 6.5 os" --repoUrl http://distros-repository.usharesoft
   ```

   The following is a list for all the other distributions:

   `Ubuntu (exemple 10.04)`

   - http://distros-repository.usharesoft.com/ubuntu/lucid-security/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-proposed/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid-proposed 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 (exemple 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 (exemple 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 (exemple version 12.2)
```

http://distros-repository.usharesoft.com/usharesoft/opensuse/12.2/x86_64

```
Scientific (exemple version 6.6)
```

http://distros-repository.usharesoft.com/usharesoft/scientificlinux/6.6/x86_64

6. You must then add the specific UShareSoft tool repository. The repository to attach is one of the following:

   - Centos (example version 6, arch x86_64): http://distros-repository.usharesoft.com/usharesoft/centos/6/x86_64
   - Redhat: (example version 6.2, arch x86_64): http://distros-repository.usharesoft.com/usharesoft/rhel/6.2/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

   For example:

```
uforge org repo create --name "CentOS 6.5 os" --repoUrl http://distros-repository.usharesoft
```

7. Attach repository to the distribution as follows:

```
uforge org repo os attach --name CentOS --arch i386 --version 6.5 --repoIds 354 -u $ADMIN -p $PA
```

8. Populate repository packages:

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

Note: This procedure may take a long time.
```

9. 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.stdo

Note: The procedure is terminated when you see the line: INFO  CheckForRepositoriesUpdates:275 -
```

10. Create OS profile based on packages (minimal, server, etc.):

```
/opt/UShareSoft/uforge/bin/runjob.py sorter_low_prio -d CentOS -v 6.5 -a i386
```

11. Populate the UShareSoft packages for the new operating system:

```
ARCHS=i386 DEBUG=y COS_VERS=6.5 /opt/UShareSoft/uforge/bin/exec_uploads.sh -w UssPkgs -p <uforge
```

3.15 Hosting Proprietary Packages

Note: The use of self-signed certificate can comport security risks.

Proprietary packages, such as RedHat are not delivered as part of the UForge repository. You must have the original ISO images of the operating system in questions and follow the steps below.

For example, to add a RedHat repository:

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 contents of the DVD into /tmp/USER_DATA/repos/RHEL/6.5/x86_64/
4. If the repositories 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 the repository using the UForge CLI as follows:

```
uforge org repo create --name "RHEL 6.5 os" --repoUrl "http://127.0.0.1/repos/RHEL/6.5/x86_6

The --name specified here is the “tagname” that will be shown in the UI when creating an app
The --repoUrl can be either http://
```

6. Attach repository to the distribution as follows:

```
uforge org repo os attach --name RHEL --repoIds 954 -u $ADMIN -p $PASS
```

7. Populate repository packages:

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

This procedure may take a long time.
```

8. 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.st

The procedure is terminated when you see the line: // INFO  CheckForRepositoriesUpdates:275 - En
```

9. Create OS profile based on packages (minimal, server, etc.):

```
/opt/UShareSoft/uforge/bin/launch_distro_sorter.sh -a x86_64 -d RHEL -v 6.5
```

10. Populate the packages for the new repository:

```
ARCHS=x86_64 DEBUG=y COS_VERS=6.5 /opt/UShareSoft/uforge/bin/exec_uploads.sh -w UssPkgs -p <
```

3.16 Populate Project Catalog (Optional)

UForge provides a project catalog of commonly used software that is used when constructing server templates. When a new operating system is added to UForge, there are no software components registered to the operating system in the Project Catalog. UForge bundles in a set of open source software components. If you want to associate these software components to the operating system for the project catalog, then run the following command:

```
# ARCHS=i386 DEBUG=y COS_VERS=5.5 /opt/UShareSoft/uforge/bin/exec_uploads.sh -w Projects -p <uforge p
```

3.17 Provide User Access to Operating System

By default, when adding a new operating system, it is not available for all users. You must explicitly provide access to the users who can access this resource. Login to one of the UForge instances and run:

```
uforge user os enable --name CentOS --version 5.5 --arch i386 --account <username> --url <uforge url>
```

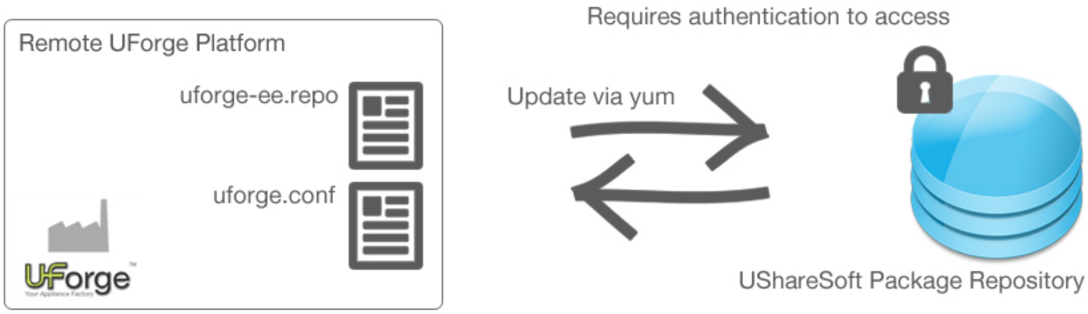
3.18 Windows Key Mechanism

Windows Operating System requires a key validation. UForge generates images without a key in it. Users have 30 days to enter their key once the Appliance is booted.

CHAPTER 4

Updating an Existing UForge Deployment

All the UForge components are delivered as native RPM packages. UShareSoft maintains 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 UShareSoft package repository, you must already have an active UForge license (stored in UShareSoft'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
```

Note: You should never update the following file: *# cat /etc/yum.repo.d/uforge-ee-repo*

The request to update uses these credentials via HTTPS to UShareSoft 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
```

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

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

Note: Running `yum update` may also update OS packages from CentOS official repository. You should accept all the updates because UForge is qualified based on the lasest packages.

```
# yum update

Loaded plugins: presto
UForge-ee-uforge                                               |  951 B     00:00
UForge-ee-uforge/primary                                       | 1.9 kB     00:00
UForge-ee-uforge
                                                                                   4/4
UForge-ee-uforge-client                                        |  951 B     00:00
UForge-ee-uforge-client/primary                                | 1.0 kB     00:00
UForge-ee-uforge-client
                                                                                   4/4
fedora/metalink                                                     | 2.6 kB     00:00
fedora                                                              | 4.3 kB     00:00
fedora/primary_db                                                   |  13 MB     00:36
updates/metalink                                                    | 1.9 kB     00:00
updates                                                             | 4.7 kB     00:00
updates/primary_db                                                  | 6.4 MB     00:21
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package MySQL-client.x86_64 0:5.5.23-1.linux2.6 set to be installed
---> Package MySQL-server.x86_64 0:5.5.23-1.linux2.6 set to be installed
---> Package MySQL-shared.x86_64 0:5.5.23-1.linux2.6 set to be installed
---> Package glassfish.noarch 0:3.1-2 set to be updated
---> Package perl-Compress-Raw-Zlib.x86_64 0:2.030-1.fc13 set to be updated
---> Package perl-Test-Simple.noarch 0:0.94-1.fc13 set to be updated
---> Package perl-parent.noarch 1:0.223-3.fc13 set to be updated
---> Package perl-threads.x86_64 0:1.81-1.fc13 set to be updated
---> Package uforge.noarch 0:3.2.5-0 set to be updated
---> Package uforge-client.noarch 0:3.2.5-0 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
=================================================================================================
 Package                        Arch           Version              Repository             Size
=================================================================================================
Installing:
 MySQL-client                   x86_64         5.5.23-1.linux2.6    UForge-ee-mysql        14 M
     replacing  MySQL-client.x86_64 5.5.17-1.linux2.6
 MySQL-server                   x86_64         5.5.23-1.linux2.6    UForge-ee-mysql        40 M
     replacing  MySQL-server.x86_64 5.5.17-1.linux2.6
 MySQL-shared                   x86_64         5.5.23-1.linux2.6    UForge-ee-mysql        1.7 M
     replacing  MySQL-shared.x86_64 5.5.17-1.linux2.6
Updating:
 glassfish                      noarch         3.1-2                UForge-ee-glassfish    85 M
 perl-Compress-Raw-Zlib         x86_64         2.030-1.fc13         updates                57 k
 perl-Test-Simple               noarch         0.94-1.fc13          updates                116 k
 perl-parent                    noarch         1:0.223-3.fc13       fedora                 13 k
 perl-threads                   x86_64         1.81-1.fc13          updates                47 k
 uforge                    noarch         3.2.5-0              UForge-ee-uforge  38 M

Transaction Summary
=================================================================================================
Install       3 Package(s)
Upgrade       7 Package(s)
```

```
Total download size: 204 M
Is this ok [y/N]: y

<traces removed for readability>

....

Complete!
```

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

```
    # service squid status
    squid is stopped
    // if squid is stopped, run the following command-line
    # service squid start
```

CHAPTER 5

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.

CHAPTER 6

Retrieving Data from UShareSoft

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

1. 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. Before launching the lftp command and due to the new proxy mechanism you must run:

```
export ftp_proxy=""
```

This ensures that squid does not interfere with ftp transfer.

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 an 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 "Ac

            * 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 zli
            > 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>
            ...ommitted for clarity
            </user>
```

CHAPTER 8

Updating the Java Version

The following procedure should be run on the web service and generation nodes of a single-node machine in order to update the Java version.

1. Check that nothing is returned on web service node and gen nodes:

```
oarstat
```

2. Check that nothing is returned on gen nodes only:

```
ps -ef | grep oar
```

3. On the web service node, check the log to ensure that everything is out:

```
tailf /var/log/glassfish/domain_uforge/uforge-web-service.log
```

4. Stop the web service:

```
service glassfish stop ;
```

5. Now that nothing can happen on the web service side, you can shut down other services.

   - On the UI node(s):

   ```
   service tomcat stop
   ```

   - On the LDAP node:

   ```
   service OpenDJ stop
   ```

6. Check if the processes are stopped. Usually this is run on the DB node for the cron update_repos_pkgs.sh:

```
ps -ef | egrep -i 'java|spider'
```

7. Launch update.:

```
yum update jdk
```

8. Restart the services.

```
    # service OpenDJ start
    # service glassfish start
    # service tomcat start
```

CHAPTER 9

Backup Overview Guidelines

Warning: UShareSoft 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. The following zones of UForge data should be backed up:

9.1 UForge Database Basic Backup

UForge uses the PerconaDB database to store all its data. PerconaDB 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

UShareSoft uses two methods for copying the database information. All the database information is located under: /var/lib/mysql

By copying this information, you are taking a snapshot of the entire database. The first uses the rsync command to copy the current latest information of the database to another location.

```
# service mysql stop
# rsync -a --delete-before /var/lib/mysql root@<BACKUP-DESTINATION>
# service mysql start
```

As the backup destination does not change, you are creating a replication of the entire database. The rsync command only copies the incremental differences each time.

The danger here 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. To overcome this, UShareSoft also creates archive backups of the entire database.

```
# service mysql stop
# f=`date +%y%m%d`
# f=`echo "$f-mysql-backup.tgz"` #what to name backups
# tar -czf /tmp/$f /var/lib/mysql/*
# rsync -a /tmp/$f root@<BACKUP-DESTINATION>
# service mysql start
```

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.

9.2 Basic Restore

Restoring the database is a simple copy using rsync back to the database directory. Note the use of slashes (/) is important.

```
# service mysql stop
# rsync -a --delete-before root@<BACKUP-DESTINATION>/mysql/ /var/lib/mysql
# service mysql start
```

9.3 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 instanceto 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 "Doing a 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 backup the database (nightly backups are recommended).

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

9.4 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 and `My Software`)
- UForge Studio packages

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 Database Basic Backup*, or disk mirroring (RAID), or both.

9.5 LDAP Backup and Restore

You can backup your data without stopping the server using:

```
/opt/OpenDJ/bin/backup -a -c -d /tmp/dir_to_backup
```

Note that the example above uses /tmp for the backup; however, we recommend you use a mounted disk, SAN or NAS to store your backup.

To restore the latest backup, you must first stop the LDAP service:

```
service OpenDJ stop
```

Then run the restore:

```
/opt/OpenDJ/bin/restore -d /tmp/dir_to_backup/userRoot
/opt/OpenDJ/bin/restore -d /tmp/dir_to_backup/tasks
/opt/OpenDJ/bin/restore -d /tmp/dir_to_backup/config
/opt/OpenDJ/bin/restore -d /tmp/dir_to_backup/schema
```

To restart the LDAP service:

```
service OpenDJ start
```

9.6 IDM Service Backup and Restore

UForge uses Syncope for identity management. Note that the following example uses /tmp for the backup; however, we recommend you use a mounted disk, SAN or NAS to store your backup.

9.6.1 To Backup Syncope

1. Set the environment variables:

```
    source /etc/UShareSoft/uforge/uforge.conf
```

2. Backup the syncope database as follows:

```
        curl --noproxy idm-server -o /tmp/content.xml -u $UFORGE_IDM_ADMIN:$UFORGE_IDM_ADMIN_PWD -H
```

3. You can verify the backup was created as follows:

```
    ls -lrt /tmp/content.xml
    -rw-r--r-- 1 root root 89875 Dec  9 16:09 /tmp/content.xml
```

9.6.2 To Restore the IDM Data

1. Make sure you “drop” the syncope database (ensure that it is scrubbed clean)

```
    export MYSQLCMD="mysql -s -u$UFORGE_DB_LOGIN -p$UFORGE_DB_PASSWORD"
    echo "DROP DATABASE IF EXISTS syncope;" | $MYSQLCMD
```

2. Recreate an empty syncope database:

```
echo "CREATE DATABASE syncope CHARACTER SET utf8 COLLATE utf8_bin;" | $MYSQLCMD
```

3. Stop the webservice.

```
    service glassfish stop
    Running stop glassfish domain syncope:                     [  OK  ]
    Running stop glassfish domain uforge:                      [  OK  ]
```

4. Copy the backup to the restore path:

```
scp /tmp/content.xml root@idm-server:/opt/GlassFish/glassfish/domains/syncope/applications/synco
```

5. Start the Webservices (IDM & UForge)

```
    service glassfish start
    Running start glassfish domain syncope:                    [  OK  ]
    Running start glassfish domain uforge:                     [  OK  ]
```

6. Send reconciliation task to synchronize the new Syncope database with the UForge LDAP database:

```
    source /etc/UShareSoft/uforge/uforge.conf
    curl --noproxy idm-server -u $UFORGE_IDM_ADMIN:$UFORGE_IDM_ADMIN_PWD -H "Content-Type: appli
```

7. When the reconciliation task has completed the service should be restored and functional. You can check the reconciliation task has completed successfully:

```
curl --noproxy idm-server -u $UFORGE_IDM_ADMIN:$UFORGE_IDM_ADMIN_PWD -H "Content-Type: applicati
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5312  100  5312    0 <latestExecStatus>SUCCESS</latestExecStatus> 0
  0   205k      0 --:--:-- --:--:-- --:--:--  235k
```

8. Then run:

```
    curl --noproxy idm-server -u $UFORGE_IDM_ADMIN:$UFORGE_IDM_ADMIN_PWD -H "Content-Type: appli
```

9. Then run:

```
    curl --noproxy idm-server -u $UFORGE_IDM_ADMIN:$UFORGE_IDM_ADMIN_PWD -H "Content-Type: appli
```

Before you begin you might want to read:

9.7 Backup Overview Guidelines

> **Warning:** UShareSoft 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 backups 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)

9.8 Backup Recommendations

> **Warning:** UShareSoft 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.

9.9 Getting Professional Services for Database Backup

UShareSoft are partners with SkySQL who give professional advice on database high-availability and backup solutions. Contact UShareSoft if you wish to be introduced to SkySQL.

CHAPTER 10

Manage Services

The following sections cover information regarding managing the UForge services/.

10.1 Starting and Stopping the Application Server

Glassfish 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.

10.1.1 Stopping the web service

```
# service glassfish stop
```

10.1.2 Starting the web service

```
# service glassfish start
```

10.1.3 Restarting the web service

```
# service glassfish restart
```

10.2 Starting and Stopping the Database

The database is registered as an operating system service. Each time a database node is rebooted, PerconaDB 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>
```

10.2.1 Stopping the Database

```
# service mysql stop
```

10.2.2 Starting the Database

```
# service mysql start
```

10.2.3 Restarting the Database

```
# service mysql restart
```

CHAPTER 11

Manage Resources

The following sections cover information regarding managing the UForge resources.

11.1 Viewing The Supported OSes

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

To get the complete list of the currently supported operating systems on the UForge platform use the command `uforge os list`.

```
Login to one of the UForge instances
# uforge os list -u $ADMIN -p $PASS
Getting operating systems ...
Success: Found the following operating systems
+----------------+-----------------+--------------+--------+---------+---------+------------------+
| Distribution   | Version         | Architecture | Access | Visible | Default | Release Date     |
+----------------+-----------------+--------------+--------+---------+---------+------------------+
| CentOS         | 5               | i386         |   X    |    X    |         | NK               |
| CentOS         | 5.3             | i386         |   X    |    X    |         | 2009-03-17       |
| CentOS         | 5.3             | x86_64       |   X    |    X    |         | 2009-03-18       |
| CentOS         | 5.4             | i386         |   X    |    X    |         | 2009-10-01       |
| CentOS         | 5.4             | x86_64       |   X    |    X    |         | 2009-10-01       |
| CentOS         | 5.5             | i386         |   X    |    X    |         | 2010-04-27       |
| CentOS         | 5.5             | x86_64       |   X    |    X    |         | 2010-04-27       |
| CentOS         | 5.6             | i386         |   X    |    X    |         | 2011-03-07       |
| CentOS         | 5.6             | x86_64       |   X    |    X    |         | 2011-03-22       |
| CentOS         | 5.7             | i386         |   X    |    X    |    X    | 2011-08-29       |
.... rest omitted for clarity
```

All the open source operating system versions are delivered as part of the UForge repository. Proprietary operating systems 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.

11.2 Viewing the Enabled OSes

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

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

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

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.

11.3 Adding an OS to an Organization

Before adding an OS to the organization you must retrieve data from the UShareSoft repository. For more information refer to *Populating Database with OS Packages*.

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

1. Add the distribution to the organization.
2. Create the repositories.
3. Attach the repository to the distribution.
4. You will then need to launch Spider to fill the repositories.

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 to the default org:

In order to add the OS to a specific distribution, you will need to specify `--org <value>` for each of the commands in the steps below.

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
    Getting default organization ...
    Success: Created repository with url [http://vault.centos.org/6.5/centosplus/i386] to default or
    +-------+----------------+------------+---------------------------------------------+-------+
    |   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 $P
    Getting default organization ...
    Success: Attach distribution to repository [355]
```

4. Launch Spider and other population steps. Refer to *Populating Database with OS Packages*.

11.4 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 you can specify only the OS name, in which case all the versions will be removed:

```
# 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
```

11.5 Creating Custom OS Profiles

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

Note: You can delete packages but UShareSoft does not guarantee that your OS profile will be functional.

To create a new OS profile:

1. Under the `Administration` tab, click `OS Profiles`.
2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.
3. Double-click on the operating system you want to administer. This will provide the current list of profiles this operating system has.
4. Select the profile which will serve as the base for your custom OS profile and click create.

5. Enter your profile name and click create.
6. Enter a description (mandatory).
7. Select the package(s) you want to add to your OS profile and click `save`. You can search for the package you want to add. 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.
8. Click `Save`.

11.6 Editing Custom OS Profiles

If you have created a custom OS profile, you can edit it at any time. You cannot modify a profile provided by UShareSoft.

To edit your OS profile:

1. Under the `Administration` tab, click `OS Profiles`.
2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.
3. Click on the operating system you want to administer. This will provide the current list of profiles this operating system has.
4. Click on the profile you want to edit.
5. You can edit the name and description from the Overview page.
6. To modify the packages included in your profile, click OS Packages. 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. Select the package(s) you want to add or remove to your OS profile and click save. You can search for the package you want to add.
7. Click `Save`.

11.7 Windows and UForge

Within UForge Windows is treated differently from other Linux/UNIX operating systems. In fact, 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 has been generated by UShareSoft or by the customer (see *Creating a Golden 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.

This chapter describes how to create your own Windows Golden Images. Windows Golden Images need to be created in order to create Windows templates. If you want to incorporate a Windows update, then you need to create and install a new set of Golden Images. UShareSoft has a tool called the Golden Creator and can provide Golden Images. This chapter assumes you want to create the Golden Image yourself.

Note: A good knowledge of Windows is required.

11.7.1 Supported Windows Versions

UForge supports Windows Server 2008 R2 (this distribution is only 64 bits), 2012 and 2012R2. 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 Windows can NOT support multiple languages.

11.7.2 Restrictions

The following UForge features are not supported with Windows Appliances:

- Package selection at the OS level (however, users can add software via MySoftware or Projects. See the UForge User Guide.)
- UForge Studio
- Windows migration possible as blackbox only
- Windows ISO format not supported

11.7.3 Workflow for Windows Golden Image

The following graphic illustrates the workflow to obtain and install Windows Golden Images.

11.7.4 First Installation of Windows

To install Windows, follow the instructions below. This example is for Windows Server 2008R2. If you wish to install for Windows Server 2012 or Windows Server 2012 R2, replace the string `Server2008R2` with the version you want to install in all commands below.

1. First population of Windows:

```
org os add --name Windows --arch x86_64 --version Server2008R2
```

2. Activate Windows Server2008R2 inside the UForge organization:

```
org os enable --name Windows --arch x86_64 --version Server2008R2
```

3. Activate Windows Server2008R2 for root user (and potentially other preexisting users) user os enable –name Windows –arch x86_64 –version Server2008R2 –account root user os enable –name Windows –arch x86_64 –version Server2008R2 –account <OTHER_USER_NAME>

4. Populate the packages for WindowsServer2008R2 (variable WIN_VERS must be changed accordingly for other versions of Windows).

```
. /etc/UShareSoft/uforge/uforge.conf
ARCHS=x86_64 DEBUG=y WIN_VERS=Server2008R2 /opt/UShareSoft/uforge/bin/exec_uploads.sh -w UssPkgs
```

Note: UForge does not manage Windows updates as it does for Linux (crawling updates every night and populating the database). Instead, each time you want to have the updates for Windows, you will need to a new set of Golden Images.

11.7.5 Listing Existing Golden Images

In order to view a list of existing golden images installed on your UForge run:

```
org golden list --name Windows --arch x86_64 --version Server2008R2
```

11.7.6 Creating a Golden Image

To create a Golden Image UShareSoft has a tool called the Golden Image Creator. This tool can run only on top of a physical machine or as a KVM virtual machine.

Generating all the profiles available (in one language) takes roughly 4 to 7 hours depending on the machine/network performance. You can regenerate Golden Images as often as you like, based on your individual needs. However, it is recommended that you regenerate only for specific updates—these updates will be in the Golden Image and you will not need to run package updates. When you generate a Golden Image the updates are the ones at the moment at which the Golden Image is generated.

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.

Once the Golden Image is created, you will need to save it to the proper location and follow the steps described Storing Golden Images on the NAS. To store the golden images (all profiles in one language) you will need about 40Gb of disk space on the UForge NAS.

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. The following Windows features must be installed as Administrator:
   - ServerCore-WOW64
   - NetFx2-ServerCore
   - NetFx2-ServerCore-WOW64
   - NetFx3ServerFeatures
   - NetFx3

   To install these features, you can either go to the `Control Panel` under `Programs and Features` or open a command prompt window as Administrator and run the following Windows commands:

```
start /w dism /online /enable-feature /all /featurename:ServerCore-WOW64
start /w dism /online /enable-feature /all /featurename:NetFx2-ServerCore
start /w dism /online /enable-feature /all /featurename:NetFx2-ServerCore-WOW64
start /w dism /online /enable-feature /all /featurename:NetFx3ServerFeatures
start /w dism /online /enable-feature /all /featurename:NetFx3
```

3. Install gtk-sharp-2.12.10.win32.msi.
   - You can download it from http://download.mono-project.com/gtk-sharp/gtk-sharp-2.12.10.win32.msi.old
   - Rename gtk-sharp-2.12.10.win32.msi.old to gtk-sharp-2.12.10.win32.msi
   - Run the following command: gtk-sharp-2.12.10.win32.msi

4. We recommend that you run Windows Update to ensure that the latest updates are pre-installed in the Golden Image.

5. Optionally, you can also add the following customizations:

   - Modify the registry
   - Extra software installation
   - User creation

6. 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
```

7. 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
```

8. 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:
```

   (a) After finishing the golden image (after sysprep at the last step), but before compressing the .raw with gzip or lrzip, perform the following command to 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
```

9. Open a command prompt window as an administrator and go to the %WINDIR%\system32sysprep directory. Then run:

```
    sysprep.exe /generalize /oobe /shutdown
```

Note: This will shutdown the machine. Do not boot the machine again!

10. You can now compress the golden images by running:

```
gzip image.raw
```

11.7.7 Installing Updated Golden Images

Each time you have a new Golden Image, you need to put them in the right NAS location as described above. Note that the parameters set when running `org golden create` should correspond to the path on the NAS, that is: {Language}/{Edition}/{Type}/{generation date}(YYYY-MM-DD)/goldenImagePathCompressedInGz

For example to install the golden image saved to the following path: Windows/releases/Server2008R2/x86_64/English/Standard/Full/2012-10-19/Windows_2008R2_Standard_Full_2012-10-19.raw.gz, you need to run:

```
org golden create --name Windows --arch x86_64 --version Server2008R2 --language English --edition St
```

11.7.8 Storing Golden Images on the NAS

To store the golden images (all profiles in one language) you will need about 40Gb of disk space on the UForge NAS.

The Golden Images should be stored in:

```
Base dir = Windows/releases/Server2008R2/x86_64/
```

The path is:

```
{Language}/{Edition}/{Type}/{generation date}(YYYY-MM-DD)/goldenImagePathCompressedInGz
```

For instance, below is a sample hierarchy:

Windows/ Windows/releases Windows/releases/Server2008R2 Windows/releases/Server2008R2/x86_64 Windows/releases/Server2008R2/x86_64/English Windows/releases/Server2008R2/x86_64/English/Standard Windows/releases/Server2008R2/x86_64/English/Standard/Core Windows/releases/Server2008R2/x86_64/English/Standard/Core/2012-
10-19 Windows/releases/Server2008R2/x86_64/English/Standard/Core/2012-10-19/Windows_2008R2_Standard_Core_2012-
10-19.raw.gz Windows/releases/Server2008R2/x86_64/English/Standard/Full Windows/releases/Server2008R2/x86_64/English/Standar
10-19 Windows/releases/Server2008R2/x86_64/English/Standard/Full/2012-10-19/Windows_2008R2_Standard_Full_2012-
10-19.raw.gz Windows/releases/Server2008R2/x86_64/English/WebServer Windows/releases/Server2008R2/x86_64/English/WebServe
Windows/releases/Server2008R2/x86_64/English/WebServer/Core/2012-10-19 Windows/releases/Server2008R2/x86_64/English/WebS
10-19/Windows_2008R2_WebServer_Core_2012-10-19.raw.gz Windows/releases/Server2008R2/x86_64/English/WebServer/Full
Windows/releases/Server2008R2/x86_64/English/WebServer/Full/2012-10-19 Windows/releases/Server2008R2/x86_64/English/WebSe
10-19/Windows_2008R2_WebServer_Full_2012-10-19.raw.gz Windows/releases/Server2008R2/x86_64/English/Enterprise
Windows/releases/Server2008R2/x86_64/English/Enterprise/Core Windows/releases/Server2008R2/x86_64/English/Enterprise/Core/20
10-19 Windows/releases/Server2008R2/x86_64/English/Enterprise/Core/2012-10-19/Windows_2008R2_Enterprise_Core_2012-
10-19.raw.gz Windows/releases/Server2008R2/x86_64/English/Enterprise/Full Windows/releases/Server2008R2/x86_64/English/Enterp
10-19 Windows/releases/Server2008R2/x86_64/English/Enterprise/Full/2012-10-19/Windows_2008R2_Enterprise_Full_2012-
10-19.raw.gz Windows/releases/Server2008R2/x86_64/English/Datacenter Windows/releases/Server2008R2/x86_64/English/Datacente
Windows/releases/Server2008R2/x86_64/English/Datacenter/Core/2012-10-19 Windows/releases/Server2008R2/x86_64/English/Datac
10-19/Windows_2008R2_Datacenter_Core_2012-10-19.raw.gz Windows/releases/Server2008R2/x86_64/English/Datacenter/Full
Windows/releases/Server2008R2/x86_64/English/Datacenter/Full/2012-10-19 Windows/releases/Server2008R2/x86_64/English/Datace
10-19/Windows_2008R2_Datacenter_Full_2012-10-19.raw.gz

11.7.9 Adding a Golden Image to UForge

Once you have your Golden Image, you need to add it to your Uforge platform 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>/<my custom profile name
```

   For example: /tmp/DISTROS/Windows/releases/Server2008R2/x86_64/English/MyProfile/Core/2014-04-28/Windows_2008R2_English_Datacenter_Core_2014-04-28.raw.gz

   Note:

   - File and directory ownership should be glassfish:glassfish.
   - Permissions should be readable for all users
   - Disk name must be unique in the /tmp/DISTORS/Windows file tree

2. You must ensure that the Windows distribution exists on your UForge platform. If it does not, run:

```
uforge org os add --name Windows --arch x86_64 --version Server2008R2
```

3. In order to add the new golden image to the distribution, run:

```
uforge org golden create --name Windows --arch x86_64 --version Server2008R2 --edition Stand
```

Note: The parameters set when running `org golden create` should correspond to the path on the NAS, that is: {Language}/{Edition}/{Type}/{generation date}(YYYY-MM-DD)/goldenImagePathCompressedInGz

11.8 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 any of the projects to their server templates. The contents of the project catalog, however, can only be updated and managed by the administrators of the organization where the project catalog belongs. An organization administrator can:

- add new projects to the catalog
- update existing projects to the catalog
- delete a project. The project will no longer appear in the project catalog, however it is not permanently deleted. This ensures that any existing templates that use the project can still generate images correctly.

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

11.8.1 Adding a Project

To add a new project:

1. Under the Administration tab, click `Projects`.
2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.
3. Projects are associated with a specific version of operating system. Click on the operating system you want to create a project for. This will provide the current list of projects this operating system has.

> **Note:** 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).

4. Click the `create` button in the `Projects` section of the page.

Fill in the mandatory information including:

- Name of the project
- Version of the project
- Project tag
- Category
- License type. Note if the software has a proprietary license, then choose `Custom`. This allows you to upload a custom license
- Company information. All previous custom companies registered for projects are provided in the list. If the company of the project does not yet exist, choose any existing company and edit it afterwards.

11.8.2 Updating a Project

UShareSoft provides the default projects for the OSes provided.

To modify the projects:

1. Under the `Administration` tab, click `Projects`.
2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.
3. Projects are associated with a specific version of operating system. Click on the operating system you want to modify. You will see a list of all the projects that are part of the OS.
4. You can select and delete projects. Note that this will not delete the project but simply mark it as obsolete.

11.9 Creating and Managing Categories

Categories are used to organize projects. Users can use categories in order to filter projects in the appliance library.

11.9.1 Creating Custom Categories

To create custom categories using GUI:

1. Under the `Administration` tab, click `Categories`.
2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.
3. Enter the name of the category you want to add.
4. Click `add`.

11.9.2 Deleting Categories

If you want to limit the list of categories users can see and use, you can delete unwanted categories. To delete categories:

1. Under the `Administration` tab, click `Categories`.
2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.
3. Select the categories you no longer want.
4. Click `delete`.
5. Click confirm in the pop-up window.

Note: If you want to delete all category customizations, click on reset. This will delete all the custom categories you created; however, this will also add any other default categories you might have deleted.

11.9.3 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
- category reset

Warning: If you reset the category list, all custom categories you created will be deleted.

11.10 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 M indicates a Milestone. For CentOS, this is the versions (6.1, 6.2 etc). Milestones can be customized by the UForge administrator using the command line interface.

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
```

CHAPTER 12

Manage Users

The following sections cover information regarding managing the UForge users. Managing users is intrinsically linked to role based access control. Please make sure to read the information on role-based access control in *RBAC Overview*.

12.1 Managing User Accounts

When managing user accounts, the administrator must have administration privileges for the organization where the user account is part of or where the member will be created.

12.1.1 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:

```
# uforge user create --account kermit --code <subscriptionProfileCode> --email kermit@usharesoft.com
```

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.

12.1.2 Disabling a User Account

Once disabled, the user no longer has access to the UForge platform.

To disable a 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
```

12.2 Creating User Groups

As an administrator you can create user groups to group users that will have the same publishing rights for Marketplace. For example rather than assign publishing rights to 10 members of a group individually, you can create a user group ("MyGroup"), and assign the publishing rights to the group. The users can then publish to the Marketplace as "MyGroup".

To create a user group, use the command `uforge usergrp create`.

For example, run:

```
uforge usergrp create --user $ADMIN --password $PASS --email kermit@usharesoft.com --name MyGroup
```

In the example above, the argument `--user` is the account of the administrator. The argument `--name` is the name of the user group you are creating.

12.2.1 Adding Users to a Group

To add users to a group, use the command `uforge usergrp user add`. Note the users you want to add must already exist on UForge. To create a user, see *Creating User Accounts with CLI*.

For example, run:

```
uforge usergrp user add --user $ADMIN --password $PASS --name MyGroup --accounts userA userB userC
```

In the example above, the argument `--user` is the account of the administrator. The argument `--name` is the name of the user group you want to add users to. The argument `--accounts` lists the user names for the accounts you want to add to the group.

12.2.2 Listing User Groups

To list the groups which a given user is part of, use the command `uforge usergrp list`:

```
uforge usergrp list --user $ADMIN --password $PASS
```

In the example above, the argument `--user` is the account of the user.

If no organization is provided, then the default organization is used.

12.3 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.

12.3.1 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 orga
|                           | user accounts.
+ Entitlements ------------------+-------------------------------------------------------------------
| marketplace_access        | used to determine whether to allow a user to interact with the marketpla
|                           | templates + template info, but voting, adding a comment, import, follow
| 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.
+ Entitlements ------------+-------------------------------------------------------------------------
| api_key_access            | Access to UForge APIs.
```

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`.

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.

12.3.2 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 appliance_create studio_access -u $ADMIN -p
```

You can also remove an entitlement from a role by using the command:

```
uforge role entitlement remove --name newrole --entitlements studio_access -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 appliance_cr
```

Note: You cannot modify the “root” role.

12.3.3 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
```

12.3.4 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
```

12.3.5 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.

12.3.6 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:

```
Usage: uforge subscription <command> [options] [args]

Commands:
        admin add                      : Add a user as a subscription profile administrator.
        admin remove                   : Remove a user as a subscription profile administrator.
        create                         : Create a new subscription profile within an organization.
        delete                         : Delete a subscription profile from an organization.
        disable                        : Disables a subscription profile within an organization    (c
        enable                         : Activates or enables a subscription profile within an organi
        format add                     : Add an image format to a subscription profile.
        format remove                  : Remove an image format to a subscription profile.
        info                           : Get detailed information on a subscription profile within a
        list                           : List all the subscription profiles for a given organization
        os add                         : Add an operating system to a subscription profile.
        os remove                      : Remove an operating system from a subscription profile.
        quota update                   : Updates a subscription profile quota
        role add                       : Add a role to a subscription profile
        role remove                    : Remove a role to a subscription profile
        update                         : Updates an existing subscription profile.

...removed options for clarity
```

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

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
- 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 quotas 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.

1. To create a subscription profile for an organization, run the command:

```
        uforge subscription create --code <string> --name <string> --org usharesoft -u $ADMIN -p $PA
```

The code can be any alpha-numeric 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 if 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.

12.4 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
```

12.5 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 created (includes import from marketplace, collaboration workspace and scans)
- number of images generated (includes image generation from a scan or from the marketplace)
- 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).

12.5.1 Viewing the Quotas for a User

You can see the quotas set for a given user as follows:

```
uforge user quota list --user $ADMIN --account <username> --password $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 --user $ADMIN --account <username> --password $PASS
Getting user [user] ...

+------------------+--------------+------------+------------+-------------------------------+
| Type             |     Consumed |      Limit | Frequency  | Renewal date                  |
+------------------+--------------+------------+------------+-------------------------------+
| appliance        |            1 |  unlimited |            | -                             |
| diskusage        |          0.0 |  unlimited |            | -                             |
| generation       |            0 |  unlimited |            | -                             |
| scan             |            0 |  unlimited |            | -                             |
+------------------+--------------+------------+------------+-------------------------------+
Found 4 formats
```

12.5.2 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
```

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.

12.5.3 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
```

12.5.4 Setting a Quota for Migration

You can set a limit to 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 --freq
```

12.5.5 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 de 10G (10 x 1024 x 1024 x 1024):

```
UFORGE_SCAN_OVERLAY_MAX_SIZE=10737418240
```

12.5.6 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 --passwor
```

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
```

12.5.7 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
```

12.6 Managing Access to Formats and OSes

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. The administrator can then add or remove operating systems and image formats for a specific user account using the command-line interface.

12.6.1 Listing Formats

To list all the formats the user (in this example "kermit") has access to:

```
$ uforge user format list --account kermit --url https://uforge.usharesoft.com:443 -u $ADMIN -p $PAS
Getting user [kermit] ...
Success: User [kermit] can have access to the following image formats in the default organization
+------------------+--------+
| Format           | Access |
+------------------+--------+
| abiquo           |   X    |
| ami              |   X    |
| cloudburstvmware |        |
| cloudcomova      |   X    |
| cloudcomqcow2    |   X    |
| cloudcomvhd       |   X    |
| cloudsigmaraw     |        |
| emi-kvm           |   X    |
| emi-xen           |   X    |
| esx               |   X    |
| hyper-v           |   X    |
| ISO               |   X    |
| kvm               |   X    |
| nimbulaesx        |   X    |
| nimbulakvm        |   X    |
| openstack         |   X    |
| ovf               |   X    |
| qcow2             |   X    |
| raw               |   X    |
| vbox              |   X    |
| vhd               |   X    |
| vmware            |   X    |
| xen               |   X    |
+-------------------+--------+
```

12.6.2 Adding or Removing Formats

To add access to one or more formats for a specific user, use the command uforge user format enable. You cannot add access to a format that is not included in the organization. For a list of formats that are part of the organization, use the command uforge org format list.

To disable access to one or more formats for a specific user (in this example “kermit”):

```
$ uforge user format disable --format ovf qcow2 vbox --account kermit --url https://uforge.usharesoft
Getting user [kermit] ...
Success: Disabled 3 image formats for user [kermit]
Success: Disabled image formats:
+-------------------+--------+
| Format            | Access |
+-------------------+--------+
| ovf               |        |
| qcow2             |        |
| vbox              |        |
+-------------------+--------+
```

12.6.3 Listing Available OSes

To list all the operating systems the user has access to:

```
$ uforge user os list --account <username> -u $ADMIN -p $PASS
```

12.6.4 Adding or Removing User Access to OSes

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 "RedHat Enterprise Linux", use the following syntax: Scientific.* RedHat.* for the distribution name.

12.7 Allowing a User to Share to the Marketplace

You can add individual users or groups of users as publishers to the Marketplace. Only users/groups that you have added will be able to publish to the Marketplace.

Note: The users that have publish rights and are also part of a user group that has publish rights will be allowed to publish as the group or as themselves.

There are two steps in order to allow a user to share appliances to the Marketplace. You must:

1. Ensure the user has the entitlement `marketplace_share` as part of one of the assigned roles. See details below.
2. Add the user to the organization publisher list.

12.7.1 Ensuring User Has Entitlement to Publish to Marketplace

To ensure that the user has the entitlement `marketplace_share` you can:

- assign a new role which has the entitlement "marketplace_share" (by default the role `partner` has this entitlement).
- add the entitlement to a role which the user has.

Note: Any changes you make to an existing role will also be applied to other users with this role.

To view the roles that are assigned to a user, use the command:

```
uforge user role list --account <username> -u $ADMIN -p $PASS
```

To view the entitlements that are part of a role, use the command:

```
uforge user role info --name <rolename> -u $ADMIN -p $PASS
```

12.7.2 Adding a User as Publisher to the Marketplace

You can add an individual user or group of users as publishers to the Marketplace. Only users/groups that you have added will be able to publish to the Marketplace.

Note: The users and groups that you want to allow to publish to Marketplace must have the `marketplace_share` entitlement.

To add a user or group as a publishers to the Marketplace, use the command uforge publish add:

```
uforge publish add --user $ADMIN --password $PASS --accounts MyGroup userA userB
```

In the example above, the argument `--user` is the account of the administrator. The argument `--accounts` lists the user names for the accounts you want to add to the group.

If no organization is provided, then the default organization is used.

12.8 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 --name new role --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
```

12.9 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 needs 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 13

Monitoring Overview

The following sections cover some elements to monitor and troubleshoot your UForge Platform.

13.1 Viewing the Web Service Logs

All the web service logs can be found in the domain directory of the Glassfish 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/glassfish/
tail -f server.log
```

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 /var/log/glassfish/
    vi log4j.properties
```

2. Update the logging level for each resource you wish by using the following keywords: `info|warn|debug`
3. Force glassfish to reload

```
    touch /opt/GlassFish/glassfish/domains/uforge/applications/uForgeWebService .reload
```

4. Restart the Web Service

```
    service glassfish restart
```

13.2 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_job
        message = FIFO scheduling OK
        scheduledStart = 2012-05-01 18:53:33
        resubmit_job_id = 0
        events =
```

13.3 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 AMI image, then publishes the AMI 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
```

13.4 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 option supervisor access in the top right of the UForge GUI banner.

To log in as another user with supervisor mode:

1. Click on `supervisor mode` 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.

CHAPTER 14

UForge Tooling

To manage the UForge AppCenter, there are a number of tools available:

- The graphical user interface
- The command line interface
- Hammr
- The APIs

This section covers:

14.1 Using the CLI Tool

There are two methods to use the command-line tool:

- interactive mode. In this case, you launch the UForge CLI as follows: uforge -u <user> -p <password> -U https://uforge.usharesoft.com/api
- by individual commands, assuming you know the command you want to launch. For example to list user information for root: uforge -u root -p <password> -U https://uforge.usharesoft.com/api user info –account root

Command-line Tool Presentation

The UForge AppCenter command-line tool has the form:

Usage: uforge <target> <command> [options][args]

Where <target> is one of the following:

```
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)
usergrp               user group administration (list/info/create/delete
                      etc)

optional arguments:
  -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 15

Rebranding Your UForge GUI

The following sections cover information regarding rebranding elements of the UForge GUI. Most customizations are done using the config.xml file 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.

15.1 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>
```

15.2 Customizing 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 following figure illustrates a sample footer.

15.2.1 Setting the Version

To set the version number in the footer, under <c:client>:

```
<c:version>3.4.2</c:version>
```

15.2.2 Hiding the UForge URL

To show or hide the UForge URL, under <c:client> enter “false” if you do not want to show the URL:

```
<c:showUForgeUrl>false</c:showUForgeUrl>
```

15.2.3 Modifying the Copyright

To modify the copyright under <c:client>:

```
<c:copyright>Copyright © 2007-2014</c:copyright>
```

15.2.4 Modifying the Links

To modify the links in the footer, under <c:footer>.

The following example shows the twitter link. If you want to remove the twitter link, delete the complete section shown below. You can also modify simply the title or link.

You can also include any other links you like in this section:

```
<c:footerItem>
   <c:title>twitter</c:title>
   <c:link>https:twitter.com/usharesoft</c:link>
</c:footerItem>
```

15.2.5 Modifying the Terms of Use or Privacy Policy

If you want to modify the Terms of Use or Privacy Policy, you MUST include the UShareSoft Terms of Use and Privacy Policy as part of your own.

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 if you wish:

```
<c:footerItem>
   <c:title>terms of use</c:title>
   <c:link>https//www.usharesoft.com/about/terms-of-use.html</c:link>
</c:footerItem>
<c:footerItem>
   <c:title>privacy policy</c:title>
   <c:link>https//www.usharesoft.com/about/privacy-policy.html</c:link>
</c:footerItem>
```

15.3 Modifying the Sign-In and Sign-Up Page

You can modify a few elements on the sign-in and sign-up page.

The following figure shows a sample Sign-in page, illustrating the elements that can be modified.

You can modify the logo that appears on the sign-in and sign-up page in section <c:client>:

```
<c:signInLogoUrl>images/signInLogo.png</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 UShareSoft.com</c:signInLogoTitle>
<c:signInLogoLink>http://usharesoft.com</c:signInLogoLink>
```

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>
```

15.4 Modifying the Title of the UForge GUI

You can modify the title that appears in the browser when you open the portal in section <c:client>:

```
<c:title>UForgePortal</c:title>
```

15.5 Hiding Option to Modify Password

You can choose to hide the option to modify the password on the My Account 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.

15.6 Modifying the Banner

You can modify several elements of the banner.

The following figure shows a sample banner, illustrating the elements that can be modified.

You can modify the logo on the left in section <c:client>:

```
<c:headerLogoUrl>images/headerLogo.png</c:headerLogoUrl>
```

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:headerLogoTitle>Go to UShareSoft.com</c:headerLogoTitle>
<c:headerLogoLink>http://usharesoft.com</c:headerLogoLink>
```

You can modify the image that appears in the middle of the banner, in section <c:client>:

```
<c:bannerUrl>images/banner.png</c:bannerUrl>
```

You can also modify the text that appears when you scroll over the banner image, as well as the link the user is directed to when clicking on the link in section <c:client>:

```
<c:bannerTitle>Go to sign up form</c:bannerTitle>
<c:bannerLink>http://usharesoft.com/products/signup/</c:bannerLink>
```

15.7 Modifying the Logo when Signed In

You can modify the logo that appears at the top of the page once a user is signed in. In section <c:client>:

```
<c:headerSignedInLogoUrl>images/headerSignedInLogo.png</c:headerSignedInLogoUrl>
```

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:headerSignedInLogoTitle>Go to UShareSoft.com</c:headerSignedInLogoTitle>
<c:headerSignedInLogoLink>http://usharesoft.com</c:headerSignedInLogoLink>
```

15.8 Restricting Formats

UForge Portal allows users to generate the templates provided to all the formats the user has access to. If you are using the UForge Online SaaS platform, users will have access to all the formats. In certain circumstances, you may wish to restrict the formats shown to the user. You can either grey out a format or remove it altogether.

To grey out a format, find the <t:format> section of the format you want to be greyed out. This format will remain visible, but will not be available to the users. Set enabled to false:

```
<t:publish>
    <t:enabled>false<t:enabled>
<t:publish>
```

If you want to remove a format completely, then you can either comment it out (which will make it easier to integrate at a later time if needed), or you can simply remove the complete section, from <t:target> to </t:target>

15.9 Restricting the Cloud Accounts

When using the UForge Portal, all the cloud account types are displayed. You can restrict the cloud accounts visible using the config.xml file. You can either comment out a cloud account type or remove it altogether. In both cases, the cloud account format will no longer be visible.

To remove the complete section, delete from <t:account> to </t:account>

15.10 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.

15.11 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.

15.11.1 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_CODE
- UFORGE_REGISTRATION_USER
- 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
```

15.11.2 Modifying UForge WebService URL

To modify the UForge webservice URL, you can modify the following variable in the uforge.conf file:

`UFORGE_URL`

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
```

15.11.3 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
```

15.11.4 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-client/bin/uforge_ui_update.sh
```

15.11.5 Modifying the UForge IP address

To modify the UForge IP address or hostname, do the following:

1. If UForge is installed on a single node, launch a terminal on the UForge machine.
2. Use system-config-network to modify hostname <MY-NEW-HOSTNAME>. It should also allow you to change the host name (that is the fully qualified domain name like ufiab.mycompany.com). This will apply the changes automatically to the directories in /etc/sysconfig
3. Ensure that the changes are applied in the following locations, otherwise make the appropriate changes:

   > /etc/sysconfig subdirectory files like ifcfg-XXX where XXX is your main interface name (like eth0) /etc/sysconfig/network for HOSTNAME and GATEWAY /etc/resolv.conf for your DNS configuration /etc/hosts for all host definitions /etc/fstab for remote NFS mounts /etc/UShareSoft/uforge/uforge.conf for the following entries: UFORGE_GF_INTERNAL_IP UFORGE_EXTERNAL_HOSTNAME UFORGE_NFS_* UFORGE_IAAS_DOWNLOAD_URL /etc/udev/rules.d/70-persistent-net.rules

4. Launch the following script:

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

5. If you have a load balancer where rules have been entered for accessing the UForge web service make sure the URIs match the following parameter in uforge.conf:

```
UFORGE_GF_WEBSVC_ROOT_CONTEXT=ufws-3.3
```

For the UForge CLI you should have:

```
UFORGE_GF_ADMIN_WEBSVC_ROOT_CONTEXT=ufadmws-3.3
```

Note: Some virtualization solutions (like VirtualBox) also add this info into ifcfg-XXX files as HWADDR=MAC

15.12 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 16

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, GlassFish, 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.

MySQL and the MySQL logo are the servicemarks, trademarks, or registered trademarks owned by Oracle Corporation Inc.

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 17

Copyright FUJITSU LIMITED 2017

CHAPTER 18

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 19

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.