UForge AppCenter User Documentation

Release 3.6

FUJITSU

February 24, 2017

Contents

1 Getting Started **3**
1.1 Signing In to UForge Portal . . . 3
1.2 Basic Concepts . . . 4
1.3 Supported Browsers for UI Access . . . 6

2 Managing Appliance Templates **9**
2.1 Supported Operating Systems . . . 9
2.2 Creating an Appliance Template . . . 10
2.3 Searching for an Appliance Template . . . 12
2.4 Listing Appliance Templates . . . 13
2.5 Modifying an Appliance Template . . . 13
2.6 Adding a Logo . . . 14
2.7 Managing the OS Profile . . . 14
2.8 Generating a Machine Image . . . 15
2.9 Publishing a Machine Image . . . 16
2.10 Tracking OS Package Updates . . . 17
2.11 Adding Custom Software Components . . . 19
2.12 Updating the Install Profile . . . 24
2.13 Configuring Advanced Partitioning . . . 26
2.14 Managing Configuration . . . 29
2.15 Cloning an Appliance Template . . . 31
2.16 Importing and Exporting Templates . . . 31

3 Migrating Live Workloads **35**
3.1 Migration Process Overview . . . 35
3.2 Blackbox Migration Process . . . 36
3.3 Whitebox Migration Process . . . 37
3.4 Migration Process In Detail . . . 38
3.5 Scanning the Source System . . . 41
3.6 Viewing a Scan . . . 44

4 Using Workspaces **51**
4.1 Creating a Workspace . . . 51
4.2 The Activity Stream . . . 52
4.3 Managing Workspace Members . . . 52
4.4 Sharing an Appliance Template in a Workspace . . . 53
4.5 Adding a Comment to a Shared Appliance Template . . . 53

5 Managing Your Accounts **55**
5.1 Modifying Your User Profile . . . 55
5.2 Viewing Your Statistics . . . 56
5.3 Managing Cloud Accounts . . . 56
5.4 Managing Your Artifact Accounts . . . 57
5.5 Managing API Keys . . . 57
5.6 Managing SSH Keys . . . 57
5.7 Changing Your Password . . . 58

6 Using the REST API **59**
6.1 Response & Error Codes . . . 59
6.2 Sending a Request . . . 61
6.3 Using the API Keys . . . 62
6.4 Query Parameters . . . 63

7 Using the Java SDK **65**
7.1 Download and Installing the SDK . . . 65
7.2 Communicating with UForge . . . 65
7.3 Creating an Appliance Template . . . 66
7.4 Adding an OS Profile . . . 66
7.5 Generating a Machine Image . . . 68
7.6 Publishing an Image . . . 68
7.7 Adding a Project from the Project Catalog . . . 69
7.8 Uploading a Software Component . . . 70
7.9 Adding a Boot Script . . . 70

8 Using the Python SDK **71**
8.1 Download and Installing the SDK . . . 71
8.2 Communicating with UForge . . . 72
8.3 Creating an Appliance Template . . . 73
8.4 Adding an OS Profile . . . 73
8.5 Generating a Machine Image . . . 74

9 Hammr Command Line Tool **75**
9.1 Getting the Source Code . . . 75
9.2 Further Reading . . . 75

10 Using Python CLI **77**
10.1 Installing Python CLI . . . 77
10.2 Launching Python CLI . . . 77

11 Changelog **79**
11.1 3.6-fp1 . . . 79

12 Trademarks **81**

13 Copyright FUJITSU LIMITED 2017 **83**

14 High Risk Activity **85**

15 Export Restrictions **87**

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

Getting Started

Signing In to UForge Portal

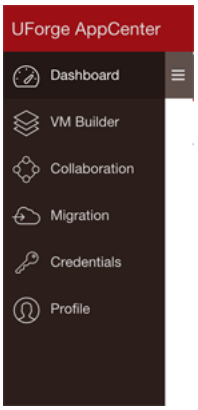
To sign in, go to the UForge Portal sign in page:

https://your-uforge-server-hostname/uforge/

The UForge Portal has the following pages, accessible from the left-hand sidebar:

- `Dashboard`: It shows statistics on your UForge usage.
- `VM Builder`: This is where your appliances are created and listed. You also go to this page to add custom software, update packages in appliances, and create images, among other things.
- `Collaboration`: This is a private area where you can share appliances with other users who are part of your workspace. These users must be invited and join your workspace. They can be part of your organization or part or another organization.
- `Migration`: This is where you can launch a scan of a live system, view the results, or compare scans.
- `Credentials`: This is where you manage your cloud account information, SSH keys and API keys.
- `Profile`: This is your UForge account information.
- `Administration`: (only for administrators of platform). Provides administration tasks including operating system and formats management.

Warning: Depending upon your access rights one or more of these tabs may not be visible.



Basic Concepts

Organization

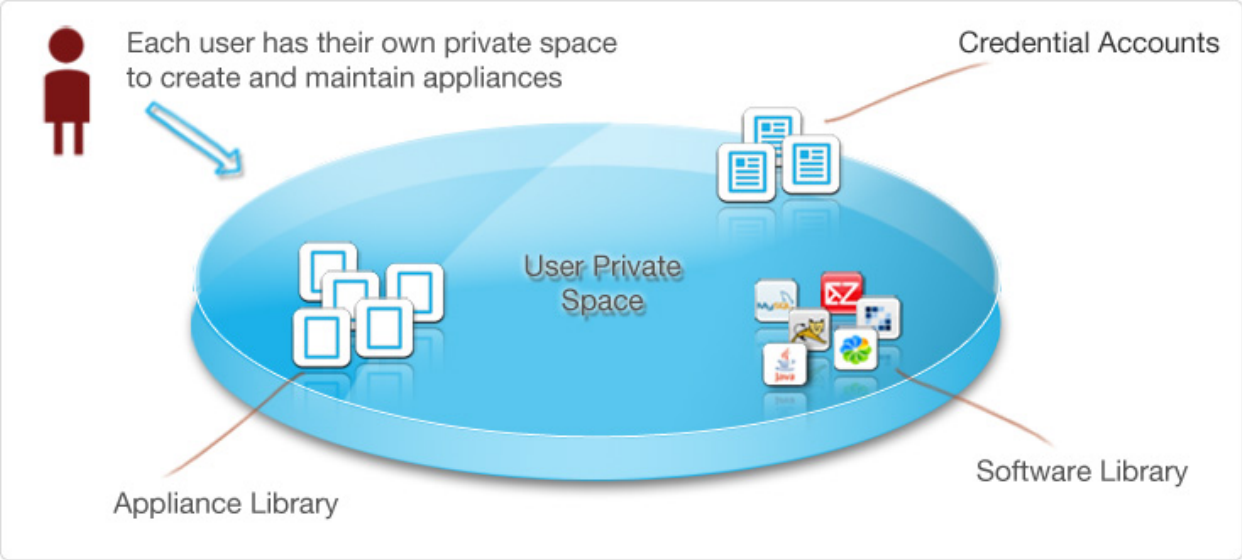
UForge AppCenter is a multi-tenant platform which can serve multiple users. All the resources of the platform are held within an Organization. The organization contains:

- One or more users
- One or more operating systems
- A project catalog containing software components that can be used by its users
- One or more formats available to generate images

User

Each user on the platform has:

- an **Appliance Library** containing all the appliance templates created by the user
- a **Software Library** (My Software) containing all the custom software uploaded by the user, which can be used in one ore more appliance templates
- a list of one or more **Cloud Accounts** to allow the user to publish and register generated machine images to various cloud and virtualization platforms



Appliance Templates

An Appliance Template is meta-data describing a software stack. It consists of five layers, namely:

- an `Install Profile` (mandatory) - specific information for the first time the image boots
- an `OS Profile` (mandatory) - a list of operating system packages. Each operating system within the organization provides one or more standard OS profiles to choose from when creating the OS Profile of the appliance template. It is also possible to create custom OS profiles.
- `Projects` (optional) - a list of Project software components chosen from the Organization´s Project Catalog
- `My Software` (optional) - a list of software components chosen from the User´s private "Software Library"
- `Configuration` (optional) - configuration information including boot scripts and/or other software components to manage the image after provisioning

Depending on the user´s roles and privileges, the user may only have access to a restricted number of operating systems, projects and image formats the organization has to offer.

Using an appliance template, the user can generate machine images in different formats. For some formats, the user can publish and register machine images to a target cloud or virtualization platform. Each appliance template stores meta-data regarding each machine image generated and published.

Workspaces

Each user can also create and join **workspaces**. A workspace is an area for members to collaborate and share appliances. A workspace is created and maintained by users. The user can invite members to be part of a workspace. When the user invites a member that is not part of the UForge database, an email is sent to the new member to invite them to register on UForge.

The workspaces are listed under the `Collaboration` tab. Each workspace has:

- an activity stream, which lists the members' comments, the activities, such as invites and appliances shared
- a templates page, which lists all the templates shared with the people that are part of the workspace
- a members page where the user who created the workspace (the workspace administrator) can invite new members, delete members or change the role of a workspace member.

Members of a workspace are either:

- `Administrator`. This is generally the user who has created the workspace. There can be several administrators in a workspace. The administrator can invite or delete members and is able to delete a workspace. The administrator has all the same basic rights as the collaborator.
- `Collaborator`. The collaborator has the same basic rights as the Guest, but can also share templates.
- `Guest`. They can read and post to the activity stream, and import templates into their private appliance library.

Supported Browsers for UI Access

The following browser versions are officially supported when using the user interface:

- Firefox v35 or later
- Chrome v29 or later
- Safari v9 or later
- Internet Explorer 11 or later
- Opera v15 or later

Note: Javascript is required when using the UI. The following error will appear if you have disabled Javascript (check your browser configuration, plugins or security settings).

FUJITSU
FUJITSU Software
UForge AppCenter
JavaScript is required
Please enable JavaScript in your browser
and try again.

CHAPTER 2

Managing Appliance Templates

An appliance template is a meta-data description of an entire software stack. The following sub-sections go into detail on how to create and manage your appliance templates:

Supported Operating Systems

The following is a list of supported OSes that users can use as the guest operating system when creating their appliance templates.

OS	Factory	Migration
CentOS	5.2+, 6 (32bit and 64bit), 7 (64bit)	5.2+, 6 (32bit and 64bit), 7 (64bit)
Debian	6 (Squeeze), 7 (Wheezy), 8 (Jessie) (32bit and 64bit)	6 (Squeeze), 7 (Wheezy), 8 (Jessie) (32bit and 64bit)
Fedora	8 to 22	8 to 22
Open SUSE	11.3, 11.4, 12.1, 12.2 (32bit and 64bit)	11.3, 11.4, 12.1, 12.2 (32bit and 64bit)
Pidora	18 to 21	Not Supported
Raspbian	7	Not Supported
Red Hat Enterprise Linux*	5.2+, 6 (32bit and 64bit), 7 (64bit)	5.2+, 6 (32bit and 64bit), 7 (64bit)
Scientific Linux	5.2+, 6 (32bit and 64bit), 7 (64bit)	5.2+, 6 (32bit and 64bit), 7 (64bit)
Ubuntu LTS	10.04 (Lucid), 12.04 (Precise), 14.04 (Trusty) (32bit and 64bit)	10.04 (Lucid), 12.04 (Precise), 14.04 (Trusty) (32bit and 64bit)
Microsoft Windows Server	2008R2, 2012, 2012R2	2008R2, 2012, 2012R2

Note: For Red Hat Enterprise Linux you must provide the ISO images or access to a repository.

Warning: If you intend to generate machine images for cloud environments, ensure that the operating system you are using in the machine image is correctly supported by the cloud environment. For example, Microsoft Azure supports the following operating systems.

Supported Machine Image Types

The following machine image formats are supported:

- Physical: ISO

- Virtual: Hyper-V, KVM, LXC, OVF, QCOW2, Raw, tar.gz, Vagrant Base Box, VHD, VirtualBox, VMWare vCenter, VMware Server, Vagrant, Xen, Citrix XenServer
- Cloud: Abiquo, Amazon AWS, CloudStack, Cloudwatt, Eucalyptus, Flexiant, Google Compute Engine, Microsoft Azure, Nimbula, OpenStack, SUSE Cloud, VMware vCloud Director

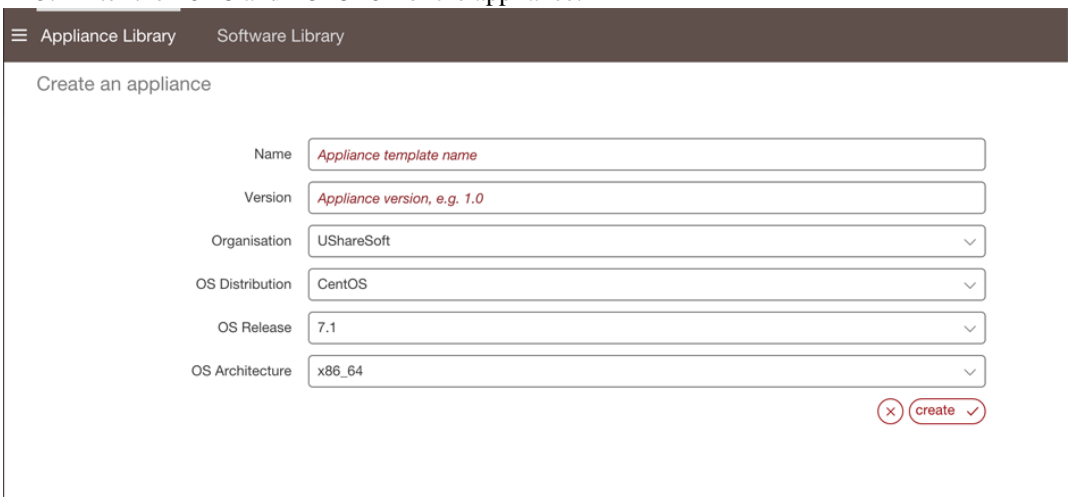
Creating an Appliance Template

You can create either Linux-based or Windows-based appliance templates. The steps differ slightly. Please refer to the appropriate section below.

Creating a Linux-based Appliance

To create a new appliance in your private workspace:

1. Select `VM Builder` icon on the left.
2. On the `Appliance Library` page, click on `create` in the top right.
3. Enter the `Name` and `Version` of the appliance.



4. From the drop-down menus, select the operating system (distribution, release and architecture).
5. Click the `create` button. This creates a skeleton of an appliance template in the platform which you can now customize with operating system packages, middleware and application software.
6. You should now see the appliance overview page. You can add a description to your appliance (optional) and a logo (optional). The logo format must be .jpg, .jpeg or .png.
7. An OS profile is mandatory. See *Adding a New OS Profile*. However, you can leave the appliance at this point and edit it later.
8. If you have made any modifications, click the checkmark to save.

Note: When you create an appliance, the packages are stored locally in the UForge cache repository. This ensures that the packages will always be available.

Creating a Windows-based Appliance

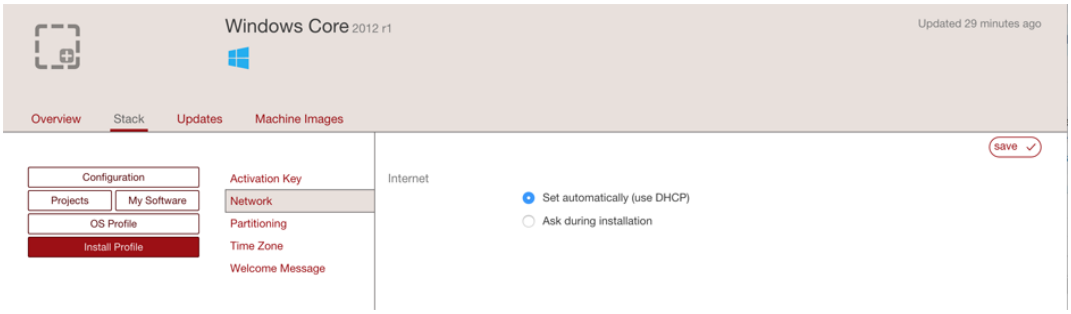
To create a Windows Appliance:

1. From the VM Builder tab, select `create`.
2. Enter the appliance name.
3. Choose `Windows` from the OS Distribution drop down menu.
4. Click `create`.
5. From the `Stack` page, select the OS Profile. Each version has a Core or Full release. Click `save`.

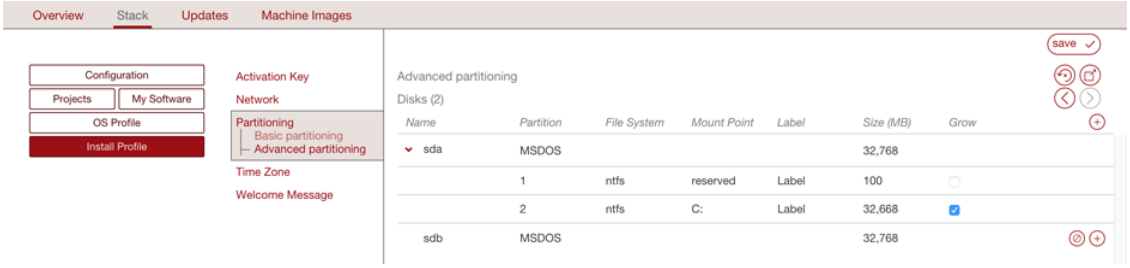
> **Note:** Once you have chosen the OS Profile, you cannot add any packages or run updates. The OS Profile is static. Once created, if you select OS Profile, you will only be able to view the details of the profile you selected.

6. Set the Install profile and click `Save`.

> **Note:** Unlike Linux, the following cannot be set for Windows appliances: Keyboard, Root user, User & Groups, Kernel Parameters and Firewall.



7. Optionally, you can set the Activation Key as part of the Install Profile. If it is not entered in the Install Profile, the default key will be used for a 30-day trial period once the appliance is booted.
8. **Optionally you can add partitions.**
   (a) Click on `Partitioning` and select `Advanced Partitioning`.
   (b) Click on the green + sign at the top.
   (c) You can modify the name and partitions type
   (d) Select the filesystem to ntfs and mount point to D: (or such).
   (e) Enter the size. The install disk should be 12 Gb for core versions and not less than 32Gb for the full version
   (f) Check the box in the Grow column if you want the partition to be growable.
   (g) Click `save`.



9. Add software bundles from the Project or MySoftware pages.

> **Warning:** Software bundles included in MySoftware and Project will be put on the image disk but the UForge generation tool WILL NOT install them even if these are executable/installers files (.exe, .msi, etc.). It is up to the end user to manually complete the installation of the software bundles.

For Windows, .exe or .msi files can be given extra parameters. The parameters depend on the .exe or .msi file, and can be used for example for silent installation, providing extra configuration values, etc.

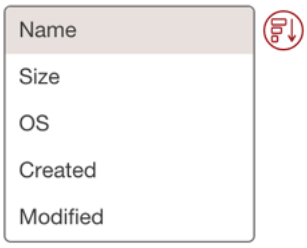
Note: A binary called `UShareInstallConfig` is embedded at generation time, which helps the final user of the Appliance do the last-mile configuration.

Searching for an Appliance Template

To find a particular appliance template you can:

1. Go to the `VM Builder` tab on the left sidebar.
2. Use the search engine. The search runs on the appliance name or the OS name. To use the search engine, enter the text in the filter field.

3. If you are in the grid view you can sort the appliances. From the drop-down menu select how you want to sort the appliances in your library.

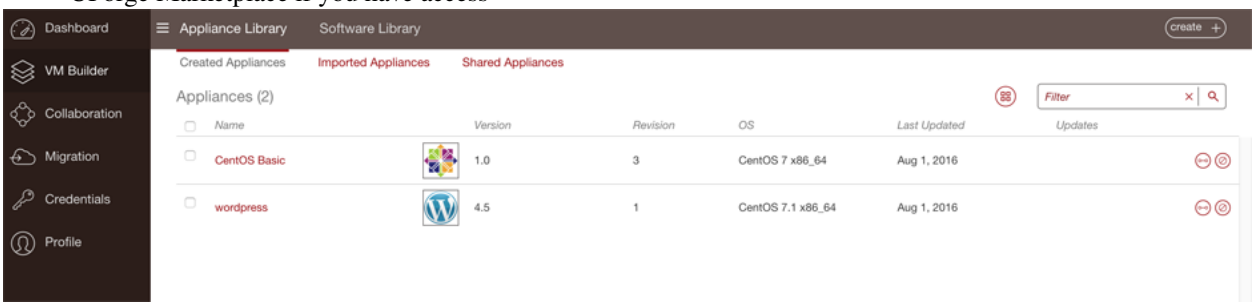


Listing Appliance Templates

To view the appliance templates you have access, go to the `VM Builder` tab in the user interface. By default the appliance templates are listed by latest first.

Under the `Appliance Library`, the appliances are organized as follows:

- `Created Appliances` are appliance templates you created
- `Imported Appliances` have been imported from a Collaboration workspace or from the UForge Marketplace if you have access to one
- `Shared Appliances` are appliance templates that you have shared to a Collaboration workspace or to the UForge Marketplace if you have access



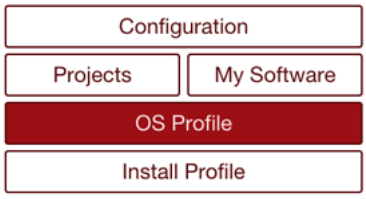
You can view the details of an appliance template, generate images from them and comment on them.

Modifying an Appliance Template

You can modify and update appliances that are in your library, including ones that have been imported.

To modify an appliance template:

1. Click on the appliance template to modify.
2. From the `Overview` page you can add or change the logo, modify the name, version or description.
3. On the `Stack` page, you will notice the appliance toolbox on the left-hand side.



The toolbox allows you to define the five key elements of an appliance, namely:

- `Install Profile` – to customize the questions asked when the image is booted for the very first time (or during installation for an ISO image). It also allows you to customize the disk size and partitioning. For more information see *Updating the Install Profile*.
- `OS Profile` – (mandatory) to choose the operating system packages that are to be used for the appliance. For more information, see *Managing the OS Profile*. Note that the OS Profile cannot be modified for Windows-based appliances. Refer to *Modifying a Windows-based Appliance*.

- `Projects` – to access the UForge Project Catalog. This catalog provides a set of commonly used 3rd party software components when building appliances. The Project Catalog is maintained by the UForge administrator. To add software from the Project Catalog to an application, see *Adding Software from the Project Catalog*
- `My Software` – to add to the appliance any software components that you have uploaded. This is also where you can use the Overlay features to manage where the files are installed during generation, if UForge should unzip archives as part of the generation, and set if UForge should on the contrary not install native OS packages. For more information, see *Adding Software from Your Software Library*
- `Configuration` – to add boot scripts to configure the appliance after provisioning. For more information, see *Managing Configuration*

Adding a Logo

You can add or modify the appliance logo as follows:

1. Click on the appliance to modify.
2. From the `Overview` page click on the square and plus (+) image on the left hand-side.
3. Select the image you want to use as the logo. The format must be .jpg, .jpeg or .png.
4. Click `ok` and save.

Managing the OS Profile

Adding a New OS Profile

Every appliance must have an OS profile, which contains all the operating system packages for the appliance. UForge allows you to easily create an OS profile from a set of standard profiles. You can then add specific operating system packages.

The goal is to only include the operating system packages you require to run your application. This process is known as JeOS (Just Enough Operating System). By only using the operating system packages you need, not only do you reduce the footprint of the resulting machine image, you also make the machine image easier to maintain (as there are less updates) and (hypothetically) more secure as there will be less unwanted services started.

To add an OS profile to your appliance:

1. Double click on the appliance you want to edit.
2. Go to the `Stack` page.
3. Click on `OS Profile` in the toolbox.
4. From the drop-down menu, select the `OS profile` template you wish to use. The operating system packages are added automatically and your appliance revision number is increased.

Adding Packages to the OS Profile

You may want to add packages that are provided as part of the operating system distribution. To get a list of all the packages that correspond to your search criteria:

1. Select the appliance to modify and go to the `Stack` page.
2. Click on `OS Profile` in the toolbox.
3. Click on the `search` button in the top right hand.
4. Enter your search string. For example, add `php` as a search string to get a list of all the PHP packages provided by CentOS.
5. Click the search icon.

6. Optionally you can filter the results by selecting `Show only 32-bit packages` (this displays only 32bit packages available) or by entering text in the `filter` box.
7. Select the packages you want to add.
8. Click the `save` button to add the packages to the OS profile.

Note: When you create an appliance, the packages are stored locally in the UForge cache repository. This ensures that the packages will always be available. However, UForge tracks all available updates.

For more information on package updates, see *Tracking OS Package Updates*. If you want to make sure you always have a specific version of a package, read *Making Packages "Sticky"*.

Generating a Machine Image

Once an appliance template has been created, you can generate a machine image that packages the stack to run on a particular virtual, cloud or physical environment.

To generate a machine image:

1. Go to the `VM Builder` tab.

2. Select the appliance from the `Appliance Library`.
3. Click on the `generate` button at the top right to display all possible image formats which can be generated. The formats are organized by type: Cloud, Container, Virtual, Physical.
4. Choose the image format you would like to generate.
5. You will see a recap of the image you are about to generate.

> **Note:** If you want to ignore dependency checking during image generation (for example because you have knowingly removed a package dependency that is not required in your environment), then you check the option `ignore dependency checking warnings`. Note that this is an advanced option and should not be used systematically.

6. You can set the disk size, then click the `generate` button to launch a generation in UForge for this appliance template.

> **Note:** Depending on the packages you install and the size of your software, make sure that the disk size is large enough to accommodate the software to be installed. For windows based operating systems, it is advised to have a disk size of at least `14GB` for core versions, and at least `20GB` for full versions.

The generation will take a few minutes to complete (depending on the number of packages in the appliance template and the disk size chosen). The generation progress is shown. Once the generation is complete, you can download the image locally, or for certain cloud formats register the machine image directly to the target environment using your cloud credentials.

You will note that a package `uforge-install-config` is injected in the generated image. This file is responsible for:

1. launching the dialog for the install profile configurations which are not automatic (keyboard, root password, licenses, time zone, static IP)
2. executing the installation bootscript of the template

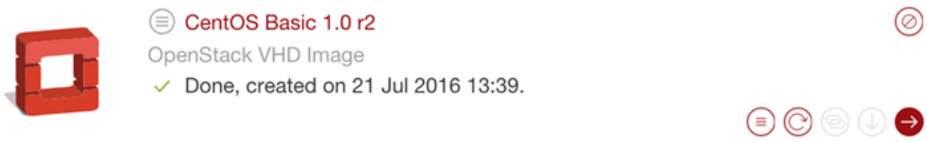
If the template is configured to be fully automatic in the `Install Profile` and has no bootscript that is supposed to run at every boot, the uforge-install-config package and associated `/etc/UShareSoft` directory can be removed safely.

However, it is preferable to leave this file.

Publishing a Machine Image

In order to publish a machine image to a cloud environment, you must already have credentials to access that environment.

1. If not already done, create a new cloud account for the target environment. For more information, see *Managing Cloud Accounts*.
2. Go to the appliance and click the `Machine Images` page. If you have not generated a machine image, you will need to do so as described in *Generating a Machine Image*.
3. Click on the arrow to publish your image.



4. Following the instructions, choose the cloud account to use and fill in any additional information required.
5. Click `publish`.

Note: Publishing an image to Amazon will launch an instance and therefore will be billed to the user account. Azure trial accounts are not supported for publishing images from UForge. Only full Azure accounts can be used.

6. The publication will take a few minutes to complete (depending on the size of the image and the network connectivity between UForge and the target cloud environment). The publication progress is shown. At the end of the publication, the machine image has been published by UForge to your target cloud environment. The published image can be found in the target cloud environment.

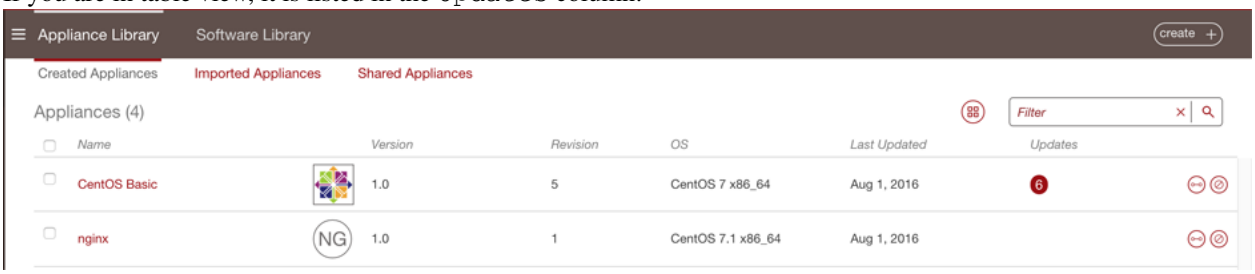
Note: UForge does not launch instances in the target cloud environment. If you wish to launch an instance from this machine image, you should go to your target cloud environment console for further actions.

Tracking OS Package Updates

All the OS packages added to the OS Profile section of the appliance templates are tracked for any updates by UForge AppCenter. Based on a timestamp stored in the appliance template, UForge AppCenter can detect any OS package updates that are available. Updates are displayed in the user interface for each appliance template.

If you are in grid view:

If you are in table view, it is listed in the `Updates` column:



You can then easily update the packages using the UForge GUI. Using this tool, you can also roll-back to previous versions of OS packages.

Note: This feature is only available for appliance templates using Linux operating systems.

To update the OS packages:

1. Select your appliance from your appliance library under the `VM Builder` tab.
2. Go to the `Updates` page.
3. You can see from the graphic when updates were made and how many are available.

4. You can select current updates (if any) or select to return to a previous version of the OS by moving the cursor on the graph.
5. Click `simulate` to see the changes that will be applied. The changes will be listed at the bottom of the screen. Scroll to view the results.
6. If you want to apply the changes listed, then click `update`.

Note: The triangle indicates a Milestone. For Centos, this is the versions (6.1, 6.2 etc). Milestones are customized by the UForge administrator.

Modifying a Windows-based Appliance

For Windows-based appliances UForge will indicate the number of updates available, however you cannot use this procedure to update the packages for an existing Windows appliance.

In order to benefit from a newer version of Windows, you will have to:

1. Create or retrieve a new Golden Image. See your administrator.
2. Create a new appliance.
3. You can re-use the MySoftware components contained in the current appliance.
4. You can download from the current template the boot scripts and save them on your local hard drive. You can then upload them to the new appliance.
5. You must re-produce the configuration (Install Profile, Configuration).

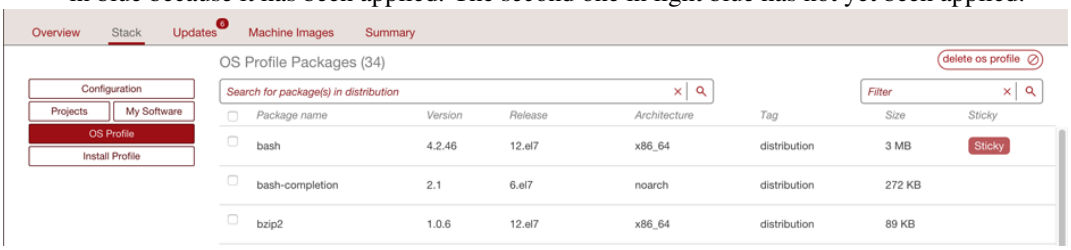
Making Packages "Sticky"

UForge allows you to select certain packages as "sticky". This means that during image generation, this package version is chosen regardless of the current appliance template timestamp for calculating package versions. All the package dependencies of this package are also calculated.

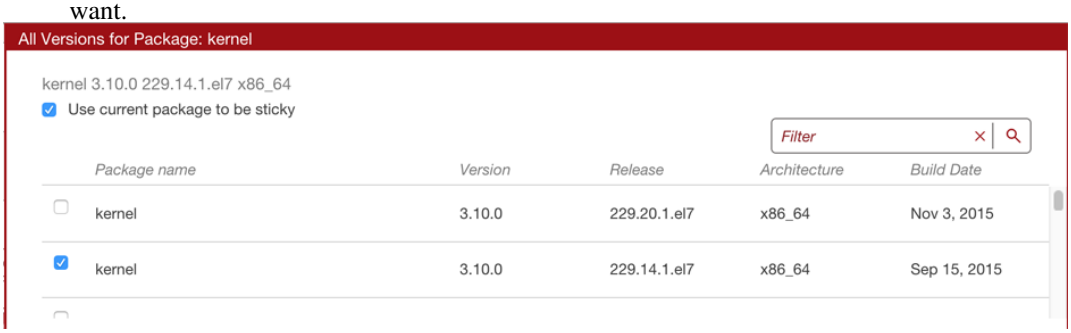
To make a specific package "sticky":

1. Select the appliance you want to modify.
2. Go to the `Stack` page.

3. From the `OS profile`, click on `sticky` in the right hand side of the package info. The `sticky` button will only be visible when you scroll over this part the page. In the following image, the first package has `sticky` in blue because it has been applied. The second one in light blue has not yet been applied.



4. A pop-up window will list all of the versions of the package available, allowing you to select the version you want.



5. Choose the version of the package you would like, then click `save`.

Adding Custom Software Components

There are two ways to add 3rd party software components or your own software to an appliance template.

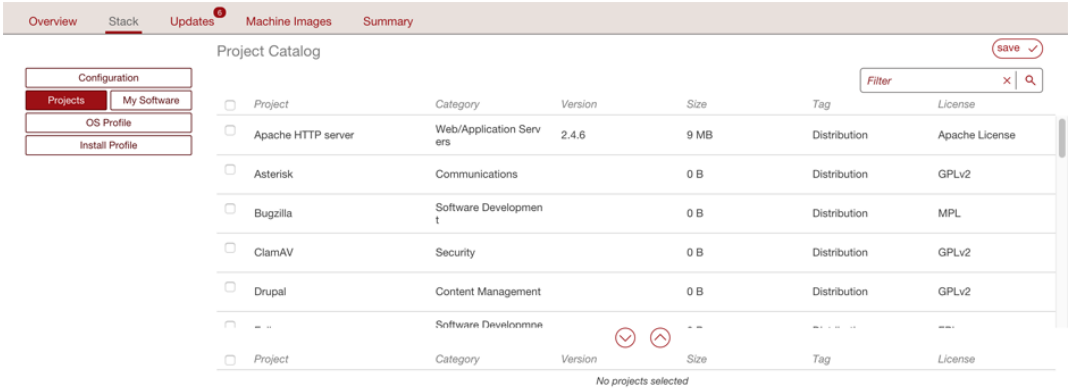
The first way is through the `Project Catalog`. This catalog is public to all the users on UForge and is maintained by the privileged users and administrators.

If the catalog does not contain the software component you are looking for, then you can upload the software into the `My Software` library. This is your own private software library, allowing you to upload any software into UForge and be able to add it to any of your appliance templates.

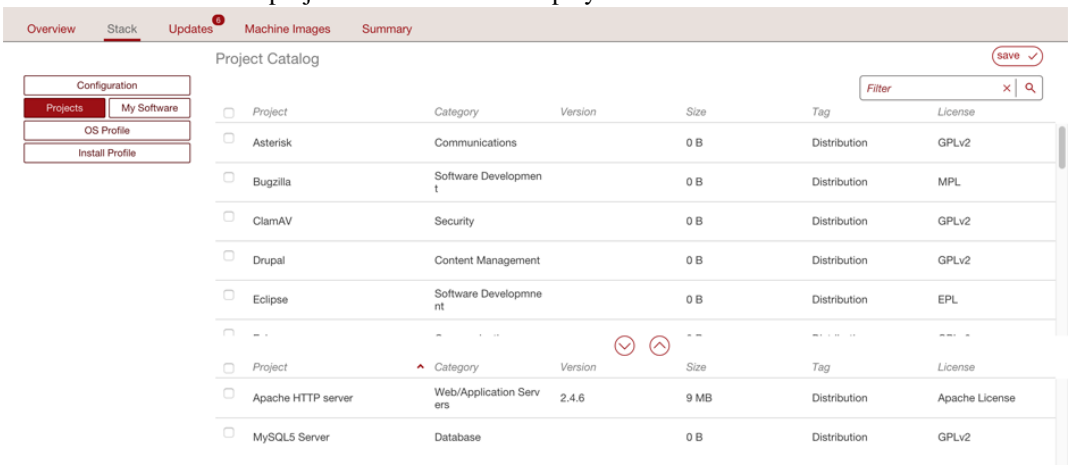
Adding Software from the Project Catalog

The following describes how to add MySQL and Apache to an appliance.

1. Select the appliance to modify and go to the `Stack` page.
2. Click on the `Projects` in the toolbox. This displays the Project Catalog. The bottom table lists which projects have already been added to the appliance.



3. Select the projects, for example Apache and MySQL and click the down arrow button. You can scroll through the available projects or enter a search string to filter the list.
4. Click Save. These projects should now be displayed in the second table.



Adding Software from Your Software Library

You can add your own custom software either using the `Software Library` or by including boot scripts. My Software overlay files (for example `/etc/profile.d/xxx.sh`) will be run before bootscripts when the machine is booted.

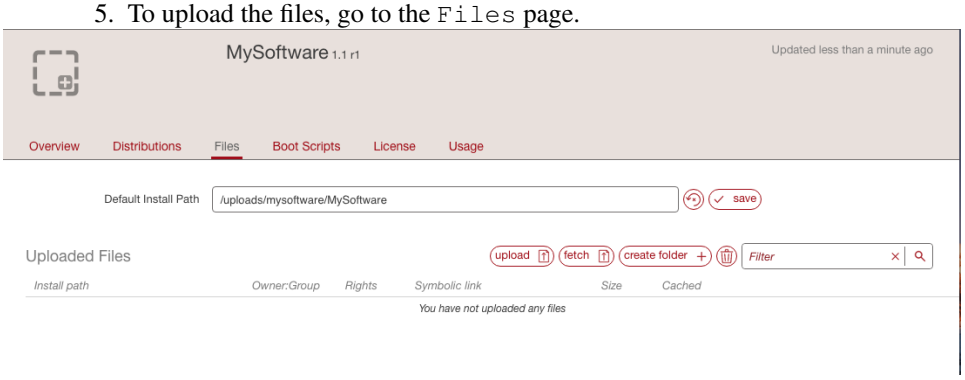
The following is a list of supported file formats:

- Linux only, note by default .rpm and .deb files will be installed at generation. This can be modified when you upload the files (see procedure below):
  - .rpm ("rpm i" will be executed)
  - .deb ("dpkg -i" will be executed)
- **Windows only:**
  - .msi
  - .exe

- **Linux and Windows compatible:**
  - .tar.bz2
  - .bz2
  - .tar.gz
  - .tgz.bz2
  - .tgz
  - .gz
  - .tar
  - .zip
  - .tar.zip

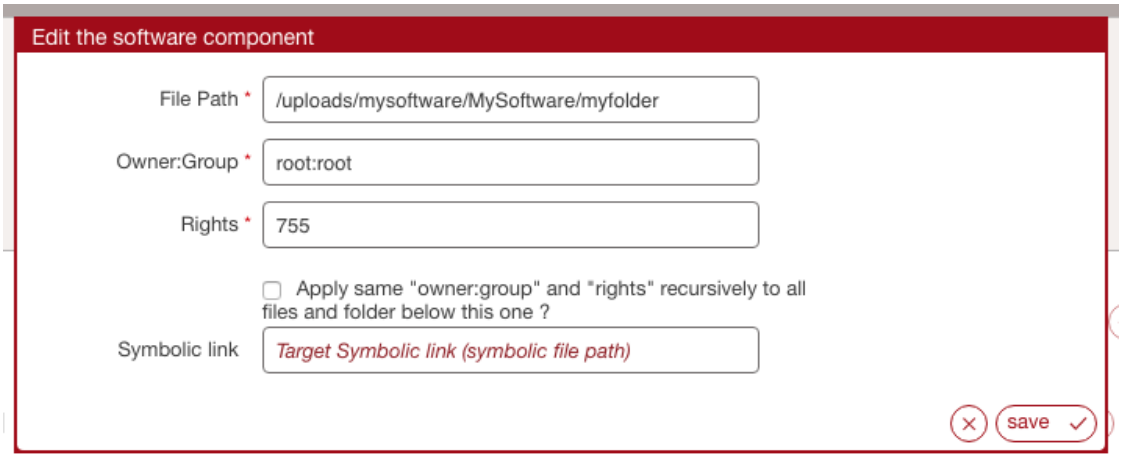
To add custom software components to an appliance:

1. From the `VM Builder` tab, click on `Software Library` in the top left hand side of the UI. This opens your private Software Library.
2. Click on the `add software` in the top right hand side.
3. You are now prompted for the name, version and maintainer of the software component you would like to add. You can also set a tag and category. Click `create` to complete.
4. The software `Overview` page will now open. You can modify the name and version, and add a description.
5. To upload the files, go to the `Files` page.



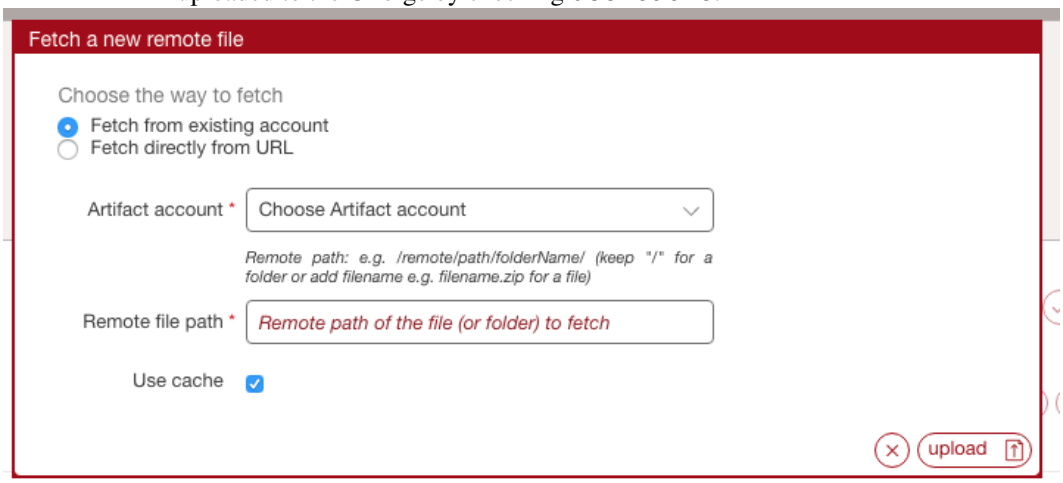
Note: If you want to group a set of files, you can create a folder by clicking `create folder`. Then enter a folder name and click `create folder`. Now if you want to put files in this folder, click on the upload icon next to the folder name.

If you create a folder, you can then click on the pencil icon to edit the file properties. Here you can select to apply the same owner and rights to all the files added to the folder.



You can now upload the files as follows:

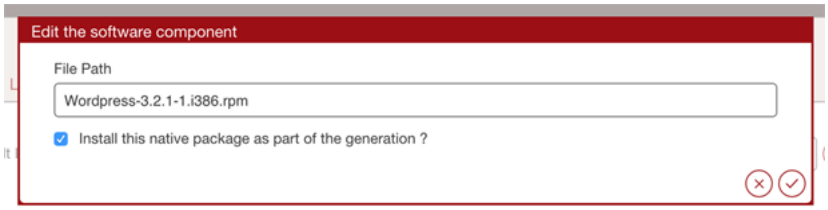
1. Click `upload` to select the files you want to add and click `open`.
2. Click `fetch` to set an archive location where the files should be retrieved. When using the `fetch` option you can indicate a remote URL or artifact account (for information on artifact accounts, refer to *Managing Your Artifact Accounts*). You can also select to have the files uploaded to the UForge by checking `use cache`.



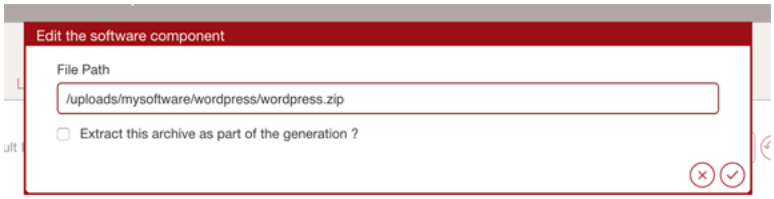
6. Optionally, you can modify the `default install path` that will be used. If you have modified the name of your appliance, it may not be reflected in the install path.
7. **By default `.deb` and `.rpm` files will be installed at generation time. Optionally you can edit the settings for those file**

   - Click on the pencil on the right hand side of your uploaded file.
   - Un-select `Install this native package as part of the generation`.
   - Click the check mark to save your changes.

   In this case, the `.deb` or `.rpm` archive file will be in the directory but will not be installed at generation time.



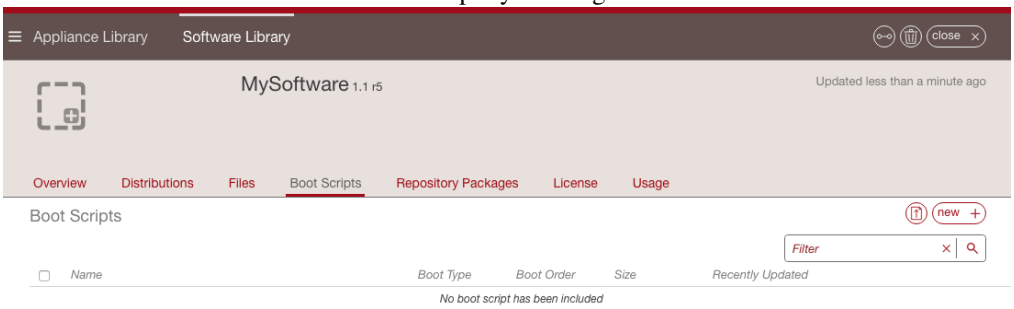
8. **Optionally, you can select to unzip archives as part of the generation. To do so:**
   - Click on the pencil on the right hand side of your uploaded file.
   - You can then edit the install path and select if it should be extracted.
   - Click the check mark to save your changes.



9. From the `Distribution` page, you can set the distributions with which your software is compatible. By default, all the distributions listed in the bottom table are compatible. If you want to remove a distribution from the list, select and click the up arrow and `save`.

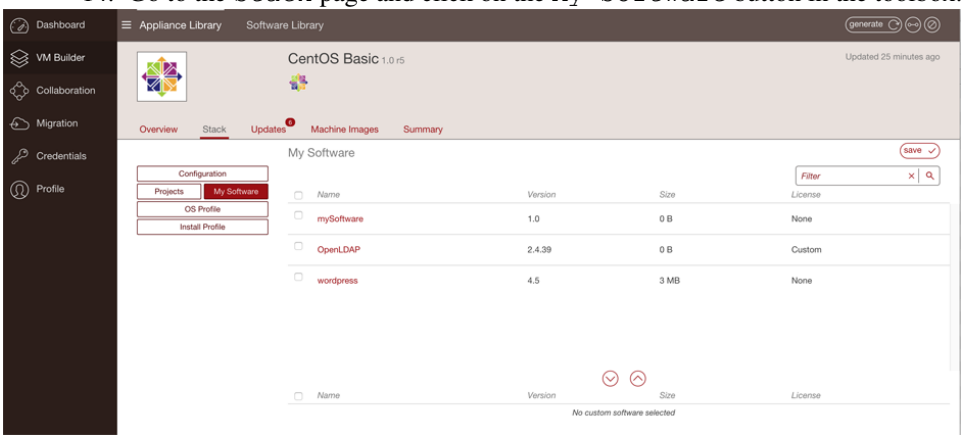
> **Note:** If you remove all the distribution compatibility, the software will not be visible for any appliances.

10. From the `Repository Packages` page, you can set the packages with which your software is compatible. This page will only be visible if you have selected only one distribution. You can search for packages. Select and click the down arrow to add them. Click `save`.
11. **From the `Boot Scripts` page, upload any boot scripts you want to add to this software. You can either:**
    - Upload an existing boot script file by using the `upload` icon.
    - Create a new boot script by clicking `new`.



In both cases, you must select the type. If you select `first boot`, then the boot script will be launched once the first time the instance is launched. If you select `every boot`, then the boot script will be launched every time the instance is rebooted. You must also set the boot order.

12. From the `Licenses` page, upload any licenses you want to add to this software. Click `upload`, select your license and click `save`.
13. Add the uploaded software component to the appliance. Click on the `Appliance Library` to view your Appliance Library. Double-click on the appliance template you want to add the software to.
14. Go to the `Stack` page and click on the `My Software` button in the toolbox.



15. Select the software components you want to add and click the down arrow button.
16. Click `save` to add this software component to your appliance template.

Updating the Install Profile

The `Install Profile` on the `Stack` page allows you to customize the questions asked when the image is booted for the very first time (or during installation for an ISO image). The install profile is mandatory.

You can define the following as part of the install profile:

- `Root User`: The root user password by default is prompted during the first boot of the machine image i.e. `ask during installation`. However, you can pre-set a root password. You can enter an SSH key to allow users to login as root. If you select `Disable root password login via SSH`, root will still be able to login from the console.
- `Users and Groups` (optional): you can add operating system users and groups. See *Adding Users and Groups* for more information
- `Network`: You can set the internet settings. The default is `set automatically`.
- `Partitioning`: You can modify the disk and swap size for the automatic set up, select `ask during install`, or set up `Advanced Partitioning` (for several disks). For more information see *Configuring Advanced Partitioning*.
- `Kernel`. You can add kernel parameters by clicking plus, entering data and click `save`.
- `Keyboard`: default is `ask during installation`. You can choose `set automatically` and select the keyboard from the drop down menu.
- `Licenses`: default is `accept licenses during installation`.
- `Time Zone`: default is `set automatically to London`.
- `Welcome Message`: You can enter a welcome message.

- `Services`: You to activate or deactivate the firewall present in the filesystem when launching the appliance (regardless of whether the firewall is iptables or other). Firewall is set to `Off` by default.

Note: For basic partitioning disk size, you must ensure that the disk is large enough to store all the binaries and files for the appliance template. For windows based operating systems, it is advised to have a disk size of at least `14GB` for core versions, and at least `20GB` for full versions.

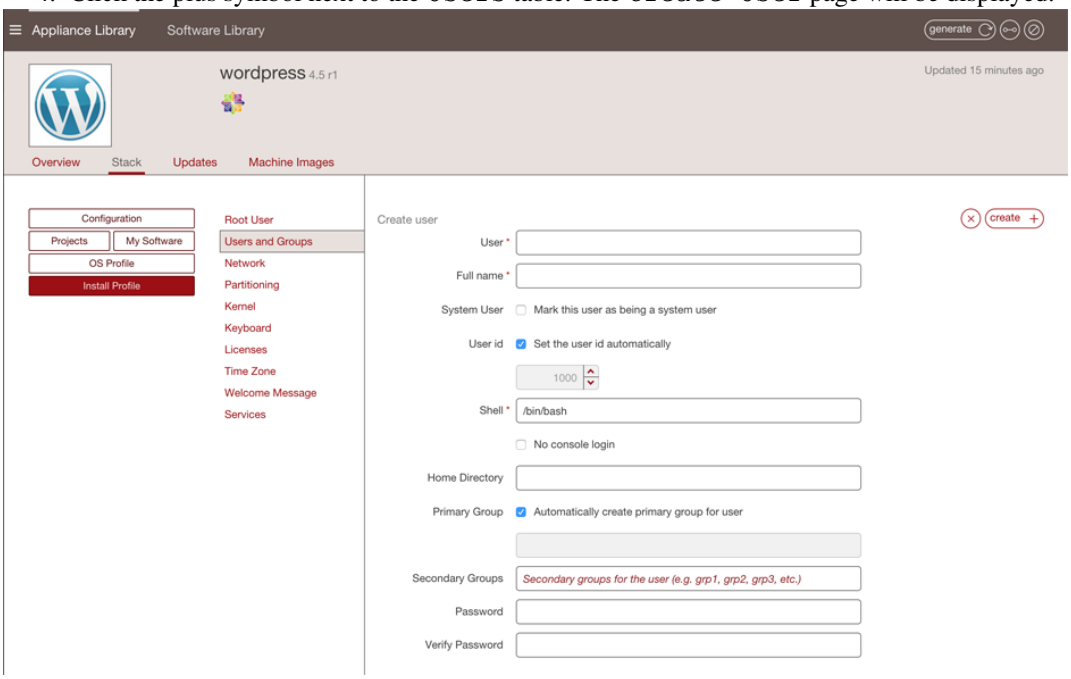
Adding Users and Groups

You can add operating system users and groups to the appliance Install Profile. These will be integrated to the appliance template.

Warning: The users and groups created in UForge are not linked. If you create a user and list it as part of a specific group, you must then also create the group; otherwise the image generation will fail.

To add a user to an appliance:

1. Select the appliance template you want to modify.
2. From the `Stack` page, click on `Install Profile` in the toolbox.
3. Select `Users and Groups`.
4. Click the plus symbol next to the `Users` table. The `Create User` page will be displayed.



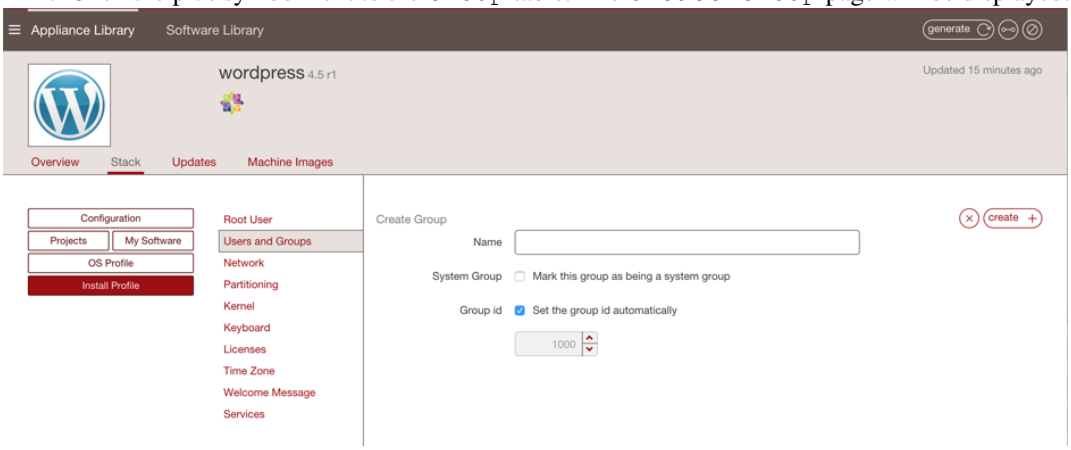
5. Enter a user name.
6. If you want to manually enter the user ID, deselect `set the user id automatically` and enter the ID number.
7. If you want to manually create the primary group the user is part of, deselect `automatically create primary group for user` and enter the group name.

8. Click `create`

> **Warning:** If you create a user and list it as part of a specific group, you must then also create the group; otherwise the image generation will fail.

To add a group to an appliance install profile:

1. Select the appliance template you want to modify.
2. From the `Stack` page, click on `Install Profile` in the toolbox.
3. Select `Users and Groups`.
4. Click the plus symbol next to the `Group` table. The `Create Group` page will be displayed.



5. Enter a group name.
6. Check if you want this group to be a system group.
7. If you want to manually enter the group ID, deselect `Set the group id automatically` and enter the group ID number.
8. Click `create`.

Configuring Advanced Partitioning

You can configure advanced partitioning as part of your appliance template in the Install Profile. The elements you can configure will depend if your template is Linux or Windows based.

Advanced Partitioning for Linux

The following example assumes that you want to build the following partitions, with a virtual hard drive of 20 GB.:

```
part /boot -fstype=ext4 -size=500 -ondisk=sda
part pv.1 -grow -size=1 -ondisk=sda
volgroup ROOTVG -pesize=4096 pv.1
logvol / -fstype=ext4 -name=LogVolROOT -vgname=ROOTVG -size=3072
logvol swap -name=LogVolSWAP -vgname=ROOTVG -size=1024
logvol /usr -fstype=ext4 -name=LogVolUSR -vgname=ROOTVG -size=5120
logvol /var -fstype=ext4 -name=LogVolVAR -vgname=ROOTVG -size=1024
logvol /home -fstype=ext4 -name=LogVolHOME -vgname=ROOTVG -size=5120
```

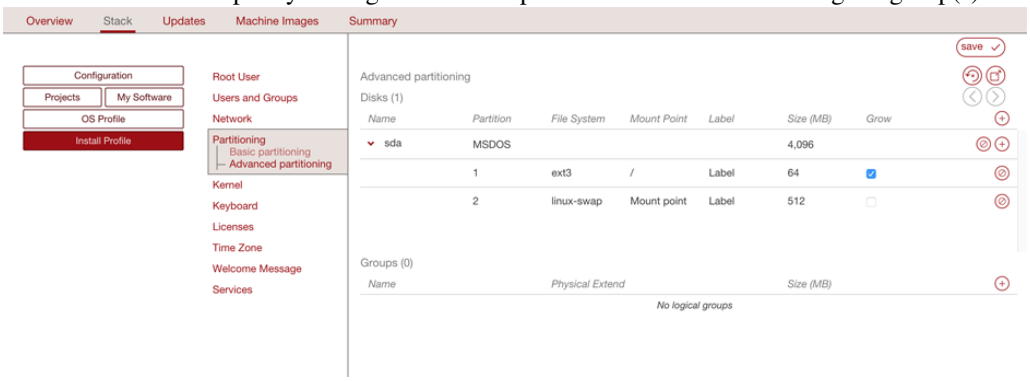
```
logvol /tmp -fstype=ext4 -name=LogVolTMP -vgname=ROOTVG -size=1024
logvol /opt -fstype=ext4 -name=LogVolOPT -vgname=ROOTVG -size=1024
```

To set advanced partitioning for an appliance template:

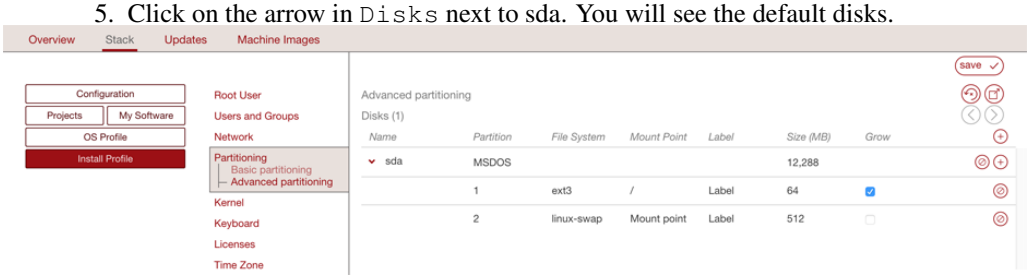
1. Select the appliance you want to modify.
2. From the `Stack` page, click on `Install Profile` in the toolbox.
3. Select `Partitioning`, then `Advanced Partitioning`.

> **Note:** In order to view the information more easily, you can click on the enlarge button in the top right. This opens a separate window where you will see all the advanced partitioning.

4. Advanced partitioning works sequentially, from top to bottom. The three sections offered by the UI, which are `Disks`, `Logical Groups`, and `Logical Volumes` should be filled in order, sequentially:
   - disks with partitions including the total virtual disk size required
   - logical group(s), assigning the associated physical extent (partition)
   - specify the logical volume specification for the created logical group(s)



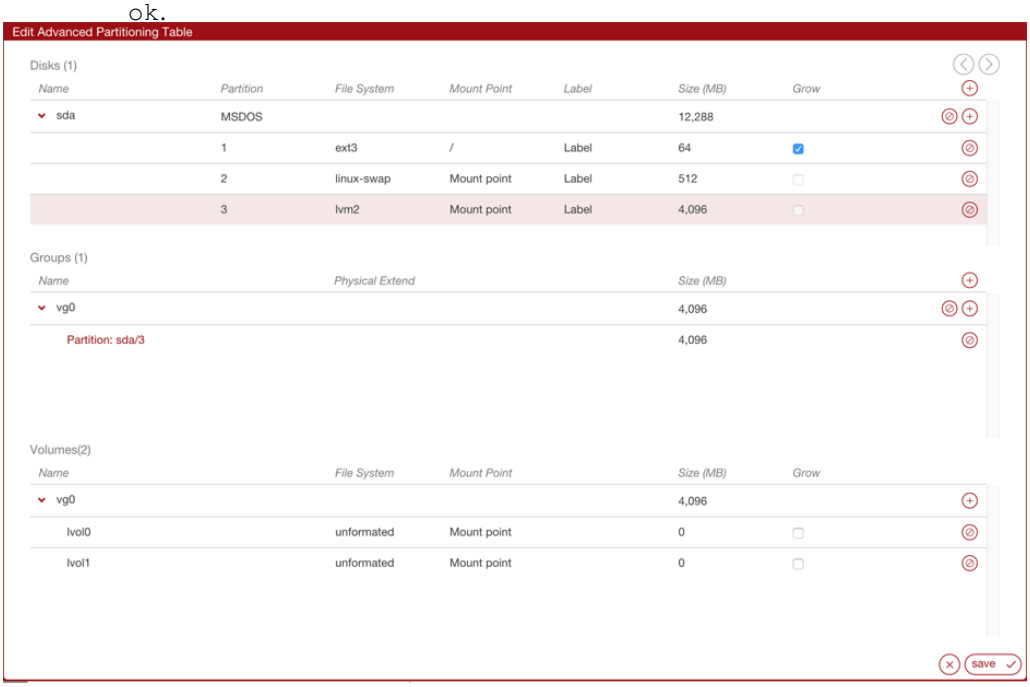
5. Click on the arrow in `Disks` next to sda. You will see the default disks.



6. Delete the default `linux swap` partition by clicking the x at the end of the line with `linux-swap`. You must not set the swap size to 0.
7. Click on the size of the `MSDOS` partition to set it to `20000`.
8. Click on the partition 1 information to modify the file system to `ext4` and the mount point to `/boot`.
9. Click on the + sign to create a new partition with type `lvm2` and size set to `18000 MB`.
10. Unselect `Grow` and set the size of the `/boot` disk to `500`.
11. In the `Logical Groups` section, click on the + sign and set the name of the logical group. For this example: `ROOTVG`.

Note: Image generation will fail when migrating if the volume group name set in the Partitioning Table is the same as the name of LVM volume group in UForge server.

12. Next to the newly created volume group, click on the + sign to create a new volume extent. A pop-up window will appear proposing a `sda/2` physical extent with size automatically set to `18000 MB`. Click `ok`.



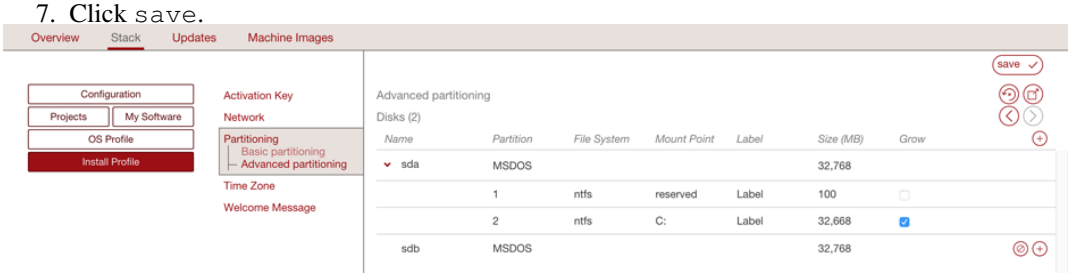
13. Create the logical volumes one by one, or create them all at once and then edit the respective specifications. For each logical volume to create, click on the + sign in the `Logical Volumes` section. For our example, you will need seven logical volumes.
    - LogVolROOT ext4 / 3072
    - LogVolSWAP linux-swap 1024
    - LogVolUSR ext4 /usr 5120
    - LogVolVAR ext4 /var 1024
    - LogVolHOME ext4 /home 5120
    - LogVolTMP ext4 /tmp 1024
    - LogVolOPT ext4 /opt 1024
14. Click `Save`.

Advanced Partitioning for Windows

You can set an advanced partitioning table for a Windows-based appliance template. To set advanced partitioning:

1. Click on `Partitioning` and select `Advanced Partitioning`
2. Click on the green + sign at the top.

3. You can modify the name and partitions type
4. Select the filesystem to `ntfs` and mount point, for example: `D:`.
5. Enter the size. The install disk should be at least `14 Gb` for core versions and `20Gb` for full versions
6. Check the box in the `Grow` column if you want the partition to be growable.
7. Click `save`.



Managing Configuration

When you create your appliance, you can include boot scripts to configure the appliance after provisioning or to configure the appliance to communicate with a DevOps platform for further communication steps.

Adding a Boot Script

You can add boot scripts that will be run either the first time the appliance is booted or every time the machine or virtual machine is started. The boot scripts will be run once all the software and appliance packages are installed, prior to launching the machine. The scripts are run in numeric then alphabetical order. The boot scripts will be executed as root.

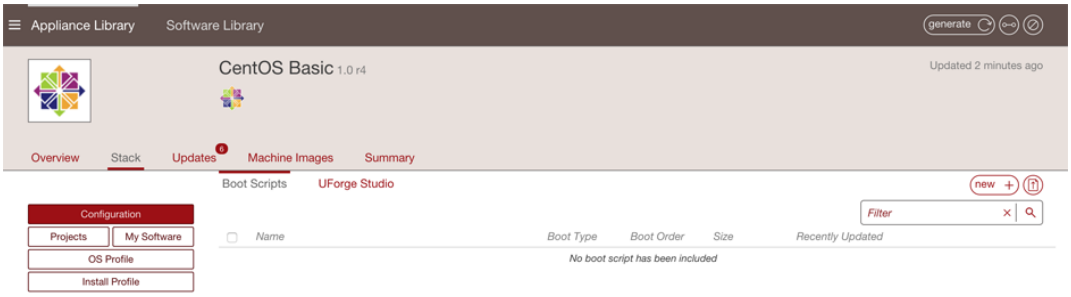
Note: The firstboots are run before everyboot scripts. Bootscripts named 1_script will be run before a_script, which will be run before script_1, which is run before script_a.

Warning: Only .bat files will be executed for Windows. If you want to upload a Powershell script then you should upload it to `My Software` and call the execution of the Powershell script from a .bat configuration bootscript.

If you want to install software or packages as part of the installation, you can use My Software to upload overlay files (e.g. /etc/profile.d/xxx.sh). For more information, refer to *Adding Software from Your Software Library*.

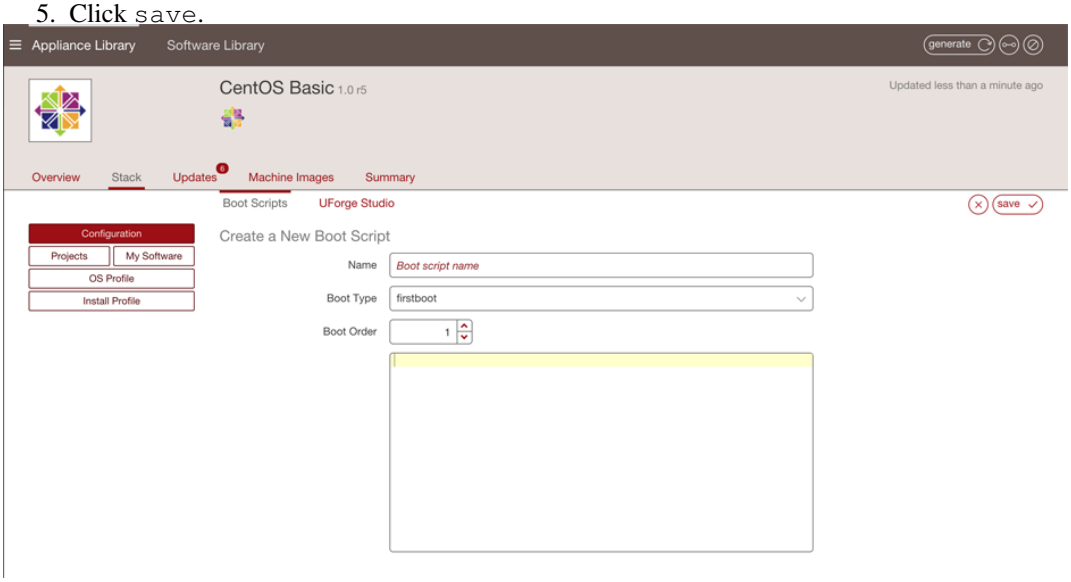
To add a boot script to your appliance:

1. Select the appliance you want to modify.
2. From the `Stack` page, click on `Configuration` in the toolbox.
3. Select `Boot Scripts`.
4. Click on `new` or `upload` button on the right hand side.



If you are creating a new boot script:

1. Enter the name.
2. Select the type: If you select `first boot`, then the boot script will be launched once the first time the instance is launched. If you select `every boot`, then the boot script will be launched every time the instance is rebooted.
3. Select the boot order.
4. Enter the contents of the boot script.
5. Click `save`.



If you are uploading an existing boot script:

1. Select the type: If you select `first boot`, then the boot script will be launched once on the first time the instance is launched. If you select `every boot`, then the boot script will be launched every time the instance is rebooted.
2. Select the boot order.
3. Click `choose` to locate your file and click `open`.
4. Click `save`.

Cloning an Appliance Template

Once you have created an appliance, you can clone it. This clones all the meta-data of the original appliance template.

Note: Your clone will not include any machine image information that was generated or published for the original.

You can clone an appliance template as follows:

1. Select the appliance you want to clone.
2. Click `clone` in the top right hand toolbar.

3. Enter the appliance `name` and `version` that you wish to have for the clone.
4. Click `clone`.

Importing and Exporting Templates

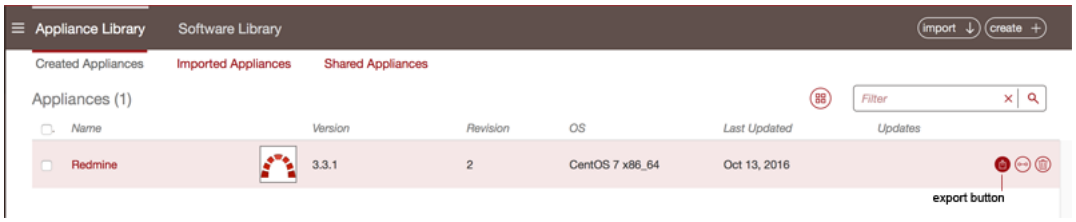
You can import and export appliance templates. When exporting, an archive is created of the appliance template. This archive includes a meta-data file describing the appliance template (based on the hammr specification) as well as any bundled software that was initially uploaded as part of the template creation.

Likewise, an archive can be imported to the UForge platform, creating a new appliance template in your `Appliance Library` under the `Imported Appliances` section.

Exporting

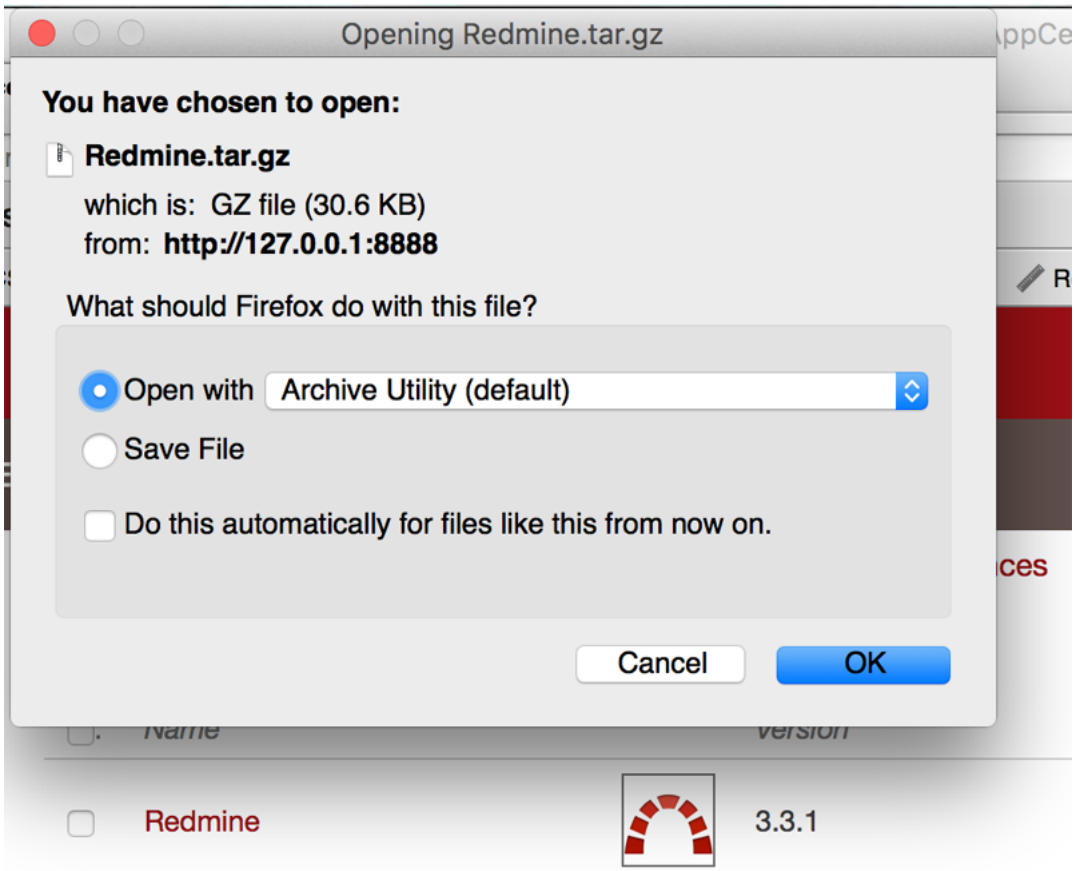
To export an existing appliance:

1. Go to your `Appliance Library`
2. Click on the `export` icon on the right hand side of the appliance template in question to export



3. This will start the export process.

4. Once the export is complete, you will be prompted to download the archive file



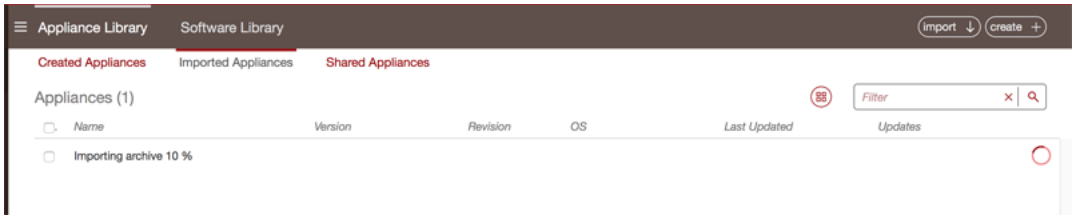
The equivalent export feature is available when editing an appliance template.

Importing

To import an archive:

1. Go to your `Appliance Library`
2. Click on the `import` button at the top right hand side of the view

3. This opens up a file browser, choose the archive to upload
4. This will start the import process.



5. Once the import is complete, a new appliance template can be found in the `Imported Appliances` sub-section of your `Appliance Library`.

Note: If you have already an appliance with the same name, version and OS type then this import will fail due to an appliance template conflict. In such situations, you will require to delete the original appliance template.

CHAPTER 3

Migrating Live Workloads

UForge AppCenter offers the capability to migrate a live system. UForge Migration will "deep scan" a live system and report back the meta-data of every file and package that makes up the running workload. It also allows you to change or add individual components prior to the final migration.

The following sub-sections describe in detail the scan steps and the differences between black box and white box migration:

Migration Process Overview

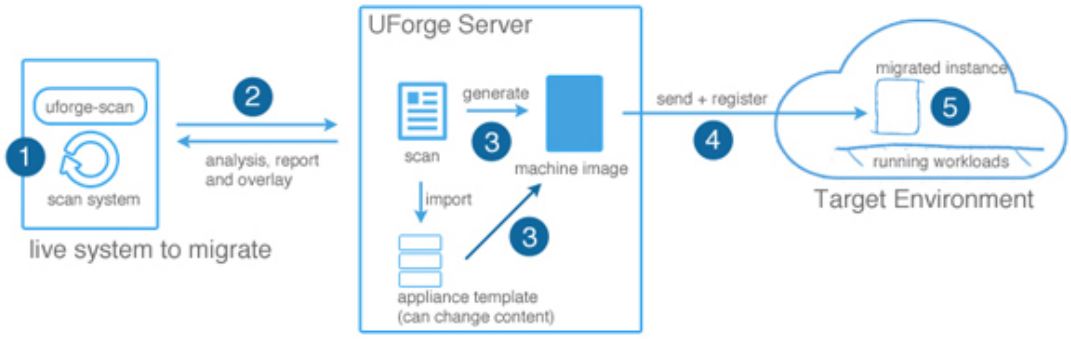
To migrate a live workload to a new target environment, you first copy the uforge-scan binary to the target environment and launch the binary. This binary analyzes the live system and sends back a report, known as a "scan" back to UForge AppCenter. You can then generate a new machine image that is compatible for the target environment. Once a machine image has been generated, the machine image can then be registered to the cloud environment and used to provision an instance. This instance will contain the same packages and files as the original running environment. This is known as "black box" migration.

You can transform a scan to an appliance template. By doing this, you have the opportunity to change the packages, files or configuration information prior to generating and registering a machine image. This is known as "white box" migration.

Warning: Whitebox migration is not supported for Windows.

The five main steps of migration are as follows:

1. The live system is "deep scanned", detecting all the files and configuration information.
2. The scan report and overlay is sent to UForge AppCenter.
3. A machine image is generated from the scan (black box migration). You can also import the scan prior to machine image generation to change the content (white box migration).
4. The machine image is uploaded and registered to the target environment.
5. The registered machine image (also known as a template in cloud terminology) can be used to provision one or more instances. These instances have near identical content to the original live system.



Blackbox Migration Process

The goal of black box migration is to reproduce a near identical copy of the currently running workload. However, there will always be small differences between the two workloads after migration is complete. When scanning a system the following information is detected:

- all the files and packages on the system (including configuration information)
- network settings including all NICs
- root and user password (encrypted)
- timezone
- keyboard settings
- kernel parameters
- users and groups
- ssh keys
- filesystem layout (partitioning)
- SELinux settings (for Linux only)

When you generate a machine image from the scan, all the information detected by the scan report is used in constructing the new machine image. However, prior to the generation starting, you will be prompted to indicate if you want to change some basic settings of the filesystem, namely the overall disk size and the swap size. You cannot set the swap size to 0. If you want to delete the swap partition, you must do this in `Advanced Partitioning` (refer to *Configuring Advanced Partitioning*).

Warning: If the IP address of the live system being scanned has a static IP address, then this IP address is preserved. Namely when the machine is migrated, the new instance has the same IP address as the original machine. In the case where the machine being scanned uses DHCP, then DHCP will be used for the migrated instance also. In this case the target environment must provide a DHCP service for an IP address to be assigned. If you wish to migrate a workload that has a static IP address, and you wish to reset the IP address or use DHCP then you should use white box migration.

Warning: Currently, UForge is not able to migrate the Yum repository GPG keys. This means that the user will have to accept the repository GPG key when the user installs or updates a package. The user will have to do this only once per repository.

When you carry out black box migration (by generating a machine image directly from a scan), the following steps are carried out:

1. You are prompted to indicate if you want to change the overall disk size.
2. Choose the machine image format to generate. Further options are provided depending upon the format chosen.
3. UForge AppCenter generates the machine image:
   - Dependency checking is SKIPPED. This is done intentionally so that UForge does not alter the package list manifest detected during the scan process.
   - Create the disk ready for installation (using the disk size and partitioning by the user if they have requested a change)
   - Install the native os packages from the scan report
   - Apply the overlay file from the scan report
   - Apply the low configuration information detected in the scan report (passwords, timezone, keyboard, etc)
   - Apply any specific libraries or configuration depending on the machine image format chosen (e.g for AWS UForge adds the required AWS libraries)
4. Register the new machine image to the target environment.
5. You can provision one or more instances from the machine image. Each instance being a near identical workload from the original.

Whitebox Migration Process

The goal of white box migration is to change the contents found during the scan of the live system prior to migration. To carry out a white box migration, the user must import the scan report as an appliance template. The import process basically transforms the meta-data of the scan report to an appliance template.

Warning: White box migration is not supported for Windows.

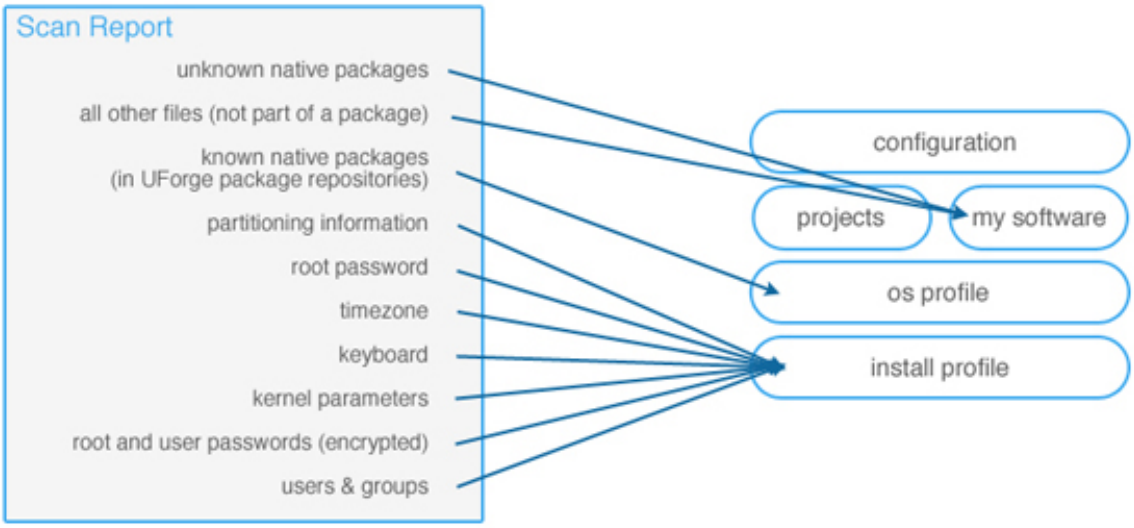
As part of this transformation process, the scan information is mapped to one or more of the appliance template layers as follows:

- Native packages that have been analyzed by the scan process and correctly found in one of UForge AppCenter package repositories are added to the `OS Profile`.
- Native packages that have been analyzed by the scan process and NOT found in any of UForge AppCenter's package repositories are added to a `My Software` component.
- All other files (including configuration files) that are not part of a native package are added to a `My Software` component.
- Partitioning table information is added to the `Install Profile`
- Root and user passwords are added to the `Install Profile` (encrypted)
- Timezone is added to the `Install Profile`
- Keyboard is added to the `Install Profile`
- Kernel parameters are added to the `Install Profile`
- Users and groups are added to the `Install Profile`

Warning: All other information found in the scan is not used (reset) and the Install Profile default information is used. This includes:

- Networking information (IP address). Furthermore, appliance templates do not support multiple NICs, therefore only the first NIC is taken into consideration.
- SELinux settings not taken into account – this information is lost
- SSH keys. You will need to manually add the ssh keys to the install profile.

Warning: Currently, UForge is not able to migrate the Yum repository GPG keys. This means that the user will have to accept the repository GPG key when the user installs or updates a package. The user will have to do this only once per repository.



Once the scan report has been imported as an appliance template, you can update the content prior to generating a machine image.

The generation process is slightly different between black box and white box migration. UForge is not generating a machine image from a scan report, rather from an appliance template. You can add and remove packages at will from the OS Profile layer. Consequently package dependencies are calculated using the list of packages in the OS Profile. Any missing packages from the OS Profile are added prior to generating the machine image.

Migration Process In Detail

The entire migration process has 5 main steps. These are:

1. Scan the source machine
2. Report the scan results to UForge AppCenter, where the platform analyzes the report. The results of this analysis are sent back to the source machine.
3. The results are used to build an "overlay" archive. This overlay is sent back to the platform.
4. The overlay archive is uncompressed, and the information is stored in the database as a `Scan`.
5. The scan can be used to generate machine images (black box migration) or imported to create a new appliance template (white box migration). The generated machine image can then be published to the target environment and instances can be provisioned.

Scanning the Source Machine

On the scan target, the `uforge-scan` binary is copied and launched as root to analyze the entire system. The scan carries out the following phases:

1. `uforge-scan` tests the connection to UForge server with the information provided by the user in the command line.
2. `uforge-scan` checks the basic information of the machine (Operating System, architecture) and the installation parameters (partitioning, timezone, keyboard, etc.).
3. Analysis of the operating system native packages installed on the system. The `uforge-scan` binary checks which packages have been installed, the state of the files in these packages etc. The scan process registers all the metadata (rights, user and groups, checksums).
4. Analysis of the files that are not part of any operating system native packages. The `uforge-scan` binary registers all the metadata (rights, user and groups, checksums).

Analysis of Report

A report is created by the `uforge-scan` binary based on all the information discovered. This report is transferred via HTTPS to UForge AppCenter.

UForge AppCenter stores all the report information. This data is then processed to identify what information is missing by UForge AppCenter to rebuild the source machine. The processing includes:

- which operating system native packages UForge AppCenter does not have in its repository or in an incremental scan.
- which files from operating system native packages have been modified compared to the official native packages in the UForge AppCenter repositories.
- which files that are not part of any OS native packages and are not in any incremental scan of the same machine.

The results of this analysis are then sent via HTTPS back to the `uforge-scan` binary on the source machine. This is basically all the information that UForge AppCenter does not have already based on the initial report received.

Build the Overlay Archive

The `uforge-scan` binary retrieves the analysis results from UForge AppCenter. These results include a list of all the packages and files UForge requires. The `uforge-scan` binary builds an overlay archive of all these packages and files.

The overlay is all the things that are missing compared to a known state (a previous scan of a machine or the operating system native packages). This overlay is a standard tar archive. Once created, it is uploaded via HTTPS to the UForge AppCenter.

The overlay is not built on the scan target but it is stream uploaded (faster and does not need any space on the scan target machine).

At this stage in the process, the `uforge-scan` binary has finished its job and no further communication between the scanned machine and UForge AppCenter is required. For this reason, the `uforge-scan` binary exits.

Note: No temporary files related to overlay remain on the scan target.

Overlay Extraction

UForge AppCenter retrieves and extracts the overlay sent by the uforge-scan binary. It then recreates all the necessary OS native packages that are not present in any of the package repositories known by UForge AppCenter. The analysis and overlay processes are now finished. All the scan metadata remain in UForge AppCenter until the scan gets deleted.

You can now carry out a black box or a white box migration. For black box migration, you generate a new machine image from the scan. For white box migration, you must first import the scan as an appliance template.

Generate an Image (Black Box Migration)

At this stage, the scan report is used to generate a new machine image. The generation tool:

1. Returns all the packages discovered on the scan target and installs them.
2. Takes the overlay and applies it on top of the built system.
3. Tunes the machine for the target environment. This is specific to the machine image format chosen. This includes injecting extra libraries and packages required by the target environment.
4. The networking information is treated differently depending on whether the IP address of the workload being migrated is using a static IP address or DHCP.
   - Static IP Addresses: The current information detected during the scan is kept. During the generation phase, this networking information is also kept. Consequently, the new machine instance has the same static IP address set.
   - Dynamic (DHCP) IP addresses: In this case, the networking information, is reset namely the IP address information is removed during the generation process, and is setup as DHCP. When the new machine instance is provisioned, the instance sends a request to the local cloud DHCP service to get a new IP address.

In the case of a migration from a para-virtualized platform to a non para-virtualized platform, UForge AppCenter injects everything that is needed to make the machine work (the kernel and its tools). Based on the packages discovered on the scan target and on the underlying operating system, UForge AppCenter calculates the most accurate kernel version to inject for your machine.

Once the image is generated, it is possible to push it to a remote environment. The image is then ready to be launched in the new environment and the migration is finished.

Import to an Appliance Template (Whitebox Migration)

At this stage, the scan is used to create a new appliance template. This allows you to change and modify the contents of the machine that has been scanned.

The process of importing:

1. Creates a template.
2. Creates an `OS Profile` and injects all the native packages.
3. Injects the overlay as a `My Software` component and is added to the appliance template.
4. Sets the scanned installation configuration information in the `Install Profile`.

It is then completely detached from the scan and you can do exactly the same things as with any other template.

If you generate an image from this template, it will go through the same steps as a standard template generation:

1. Checks all the dependencies.

2. Installs all the packages.
3. Installs all the my software components.
4. Tunes the machine for the target environment. This is specific to the machine image format chosen. This includes injecting extra libraries and packages required by the target environment.

Once the image is generated, it is possible to push it to a remote environment. The image is then ready to be launched in the new environment and the migration is finished.

Scanning the Source System

The first step in migrating your system is running a scan of the target system. This identifies the meta-data of every file and package that makes up the running workload.

You must have root access on the target system in order to complete the scan, as you will need to copy and run a binary file on the target system.

> **Warning:** UForge AppCenter does not support multi-kernels. When scanning a machine with more than one kernel, only the kernel running will be scanned and imported.

When you run a scan of a system, UForge AppCenter will differentiate between "known" data (OS packages and files that are already part of UForge AppCenter repository) and files that are "unknown". UForge AppCenter does not support more than 10 Gb of compressed "unknown" data.

> **Warning:** Any pre-install or post-install scripts on the system you are about to scan should only use ascii character set. Otherwise UForge AppCenter will return a scan error: `DB Error - invalid characters`.

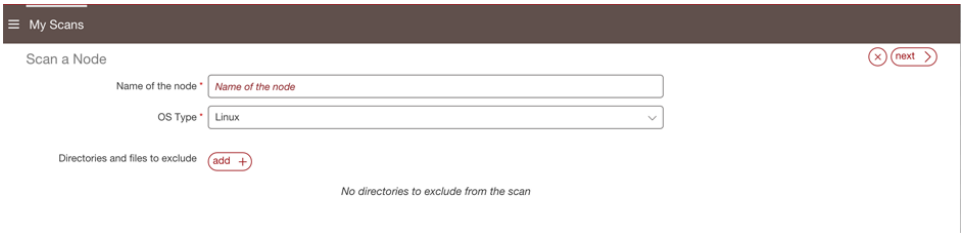
Recommendations pre-scan:

- Pre-install or post-install scripts on the system you are about to scan should only use ascii character set
- Custom packages on the live system to be scanned should not contain references to package dependencies as relative path. They should be expressed as absolute paths.
- If custom packages are installed using `--nodeps` flag, the scan process will not detect these packages. When carrying out white box migration, UForge AppCenter will check for these dependencies. If they are custom packages that are not on the live system, the generation will fail. Therefore, it is recommended to provide a custom repository with all the necessary custom packages. Otherwise, they can be added after the scan to the appliance template in `My software`.
- Image generation will fail when migrating if the source server has the same LVM volume group name as the UForge server's one. It will fail also if the volume group name set in the Partitioning Table is the same as the name of LVM volume group in UForge server.

Scanning a Linux Machine

To carry out a scan, go to the `Migration` tab:

1. Click on `scan` in the top right.
2. Enter a name for the scan of the target system you want to migrate.
3. Select `Linux` from the drop-down menu.
4. If you want to exclude certain directories or files from the scan then click `add` and enter the directory path or full pathname of the file.



5. Click `Next`. Follow the instructions on the UForge AppCenter GUI.
6. Download binary locally by clicking `Download`.
7. Copy the binary on the target environment you want to migrate.
8. Open a terminal window and login to the target environment.
9. Run the scan command on the running target environment to start a deep scan of the system you want to migrate. The binary identifies the packages, files and custom files on the system.

   By default the scan data will be saved in `/tmp`. You can modify the directory where the data will be saved using the `-t` option in order to ensure that there is enough space to save the scan data.

   You can also use API keys to run the command. In this case, in the command you copied, remove the password and enter the API keys using `-a` option for the public key and `-s` option for the secret key. For example:

```
./uforge-scan.[bin/exe] -u <username> -a <public-key> -s <secret-key> -U http://ip:port/
```

10. A report is sent to UForge AppCenter which can be used for migration. To view the progress, go back to the `Migration` tab and click `ok`.

Note: The duration of the scan depends on:

- the power of the machine in the target environment,
- the complexity of the target environment OS (number of packages installed),
- the network bandwidth between the target environment and UForge.

Scans of typical simple target environments can last about 5 to 15 minutes. In the case of larger and more complex target environments, together with poorer bandwidth, one can experience durations of up to one hour.

11. To view the details of a scan, click on the scan and refer to *Viewing a Scan*.

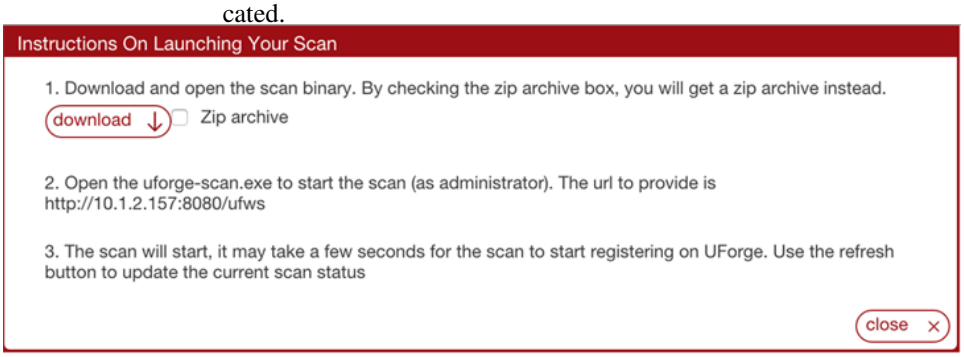
Note: Image generation will fail when migrating if the source server has the same LVM volume group name as the UForge server's one. It will fail also if the volume group name set in the Partitioning Table is the same as the name of LVM volume group in UForge server.

Scanning a Microsoft Windows Machine

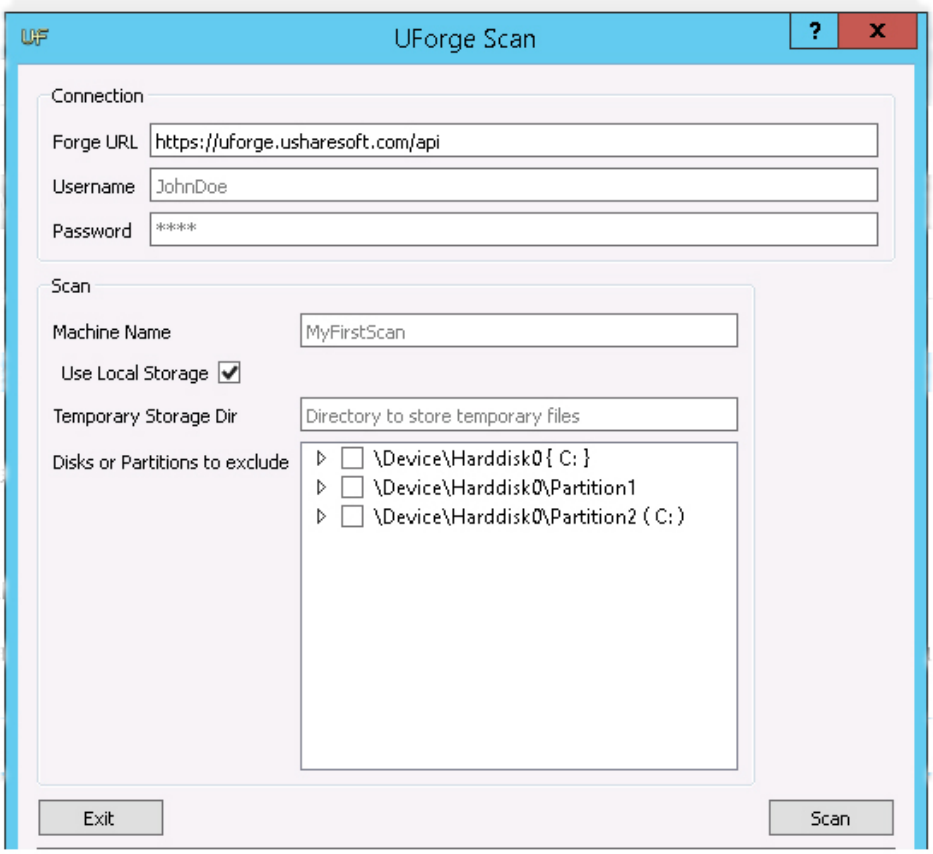
To carry out a scan, go to the `Migration` tab:

1. Click on `scan` in the top right.

2. Enter a name for the scan of the target system you want to migrate.
3. Select `Windows` from the drop-down menu and click next.
4. Select the method you want to use: if you want to scan using an exe that will launch a graphical interface, or by downloading a zip file and launching it through command-line.
    - If you are using exe the pop up will indicate the partition being scanned and time left
    - If you choose to download the zip then you will have to launch the command as indicated.

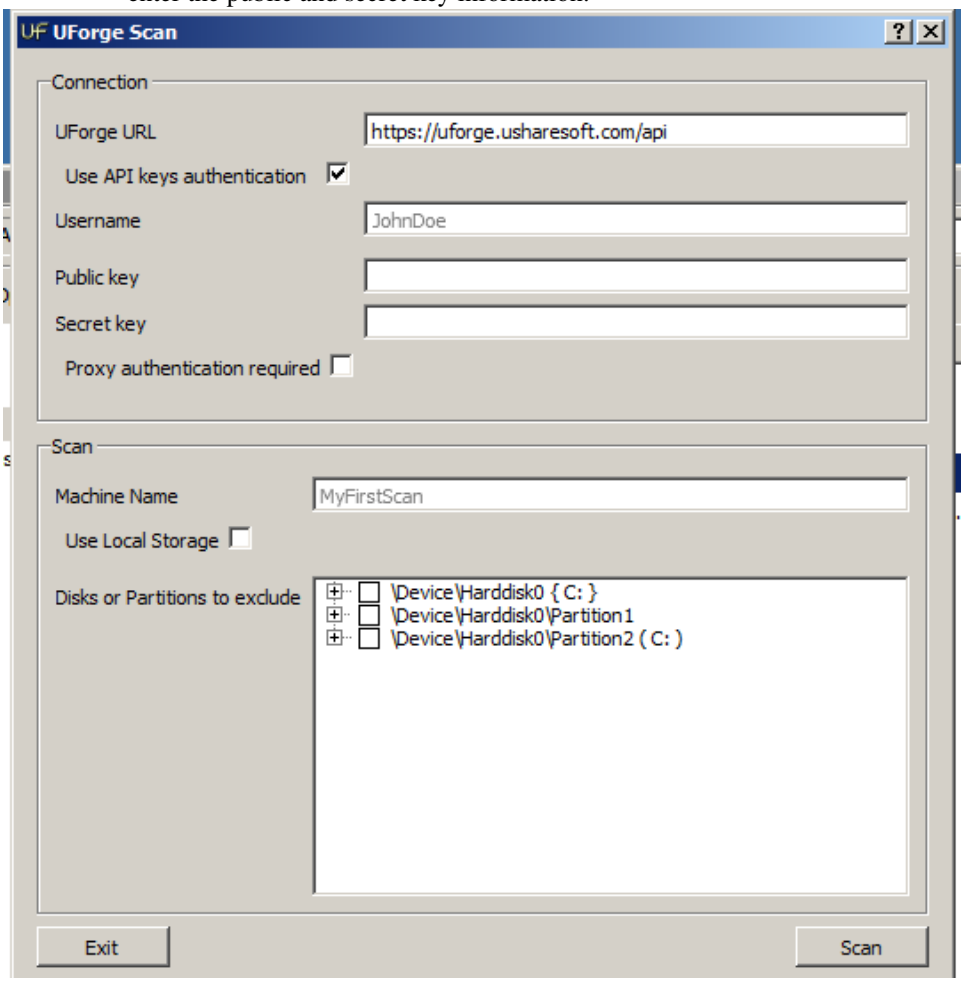


5. If you want to exclude certain directories or files from the scan then click `add` and enter the directory path or full pathname of the file.
6. Optionally you can select `Use local storage`. This means that the scan will be not be done in streaming but in 2 phases. First the data will be stored on a temporary storage drive during the scan process. This temporary storage can be a local directory or a virtual space on the network. It must be at least half the size of the machine you want to scan.



Note: If you are using local storage you will have to launch a script at the end to upload the archive to UForge AppCenter later.

7. Optionally you can use API keys. In this case, check `Use API keys authentication` and enter the public and secret key information.

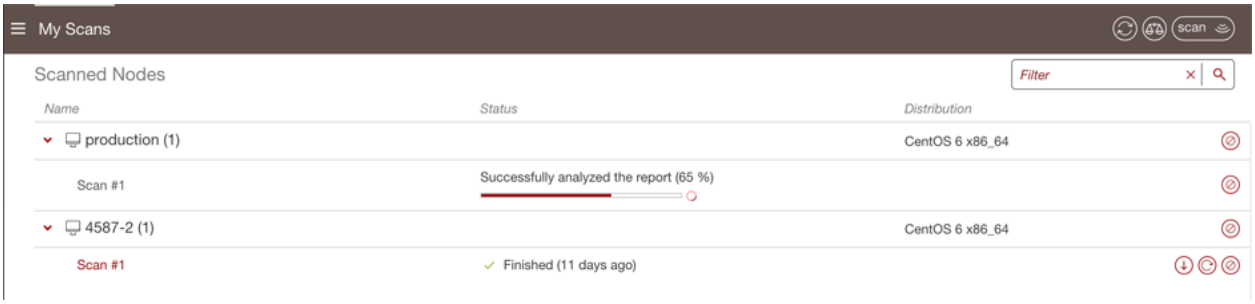


8. Click `scan` to launch the scan. A report is sent to UForge AppCenter which can be used for migration. To view the progress, go back to the `Migration` tab and click `ok`.
9. To view the details of a scan, click on the scan and refer to *Viewing a Scan*.

Viewing a Scan

Once you have run a scan, you can view the scan details.

To view the scans, go to the `Migration` tab > `My Scans`. All the scans are listed by name. If you have run the same scan with the same name, it will appear as Scan <number>.



From this page you can:

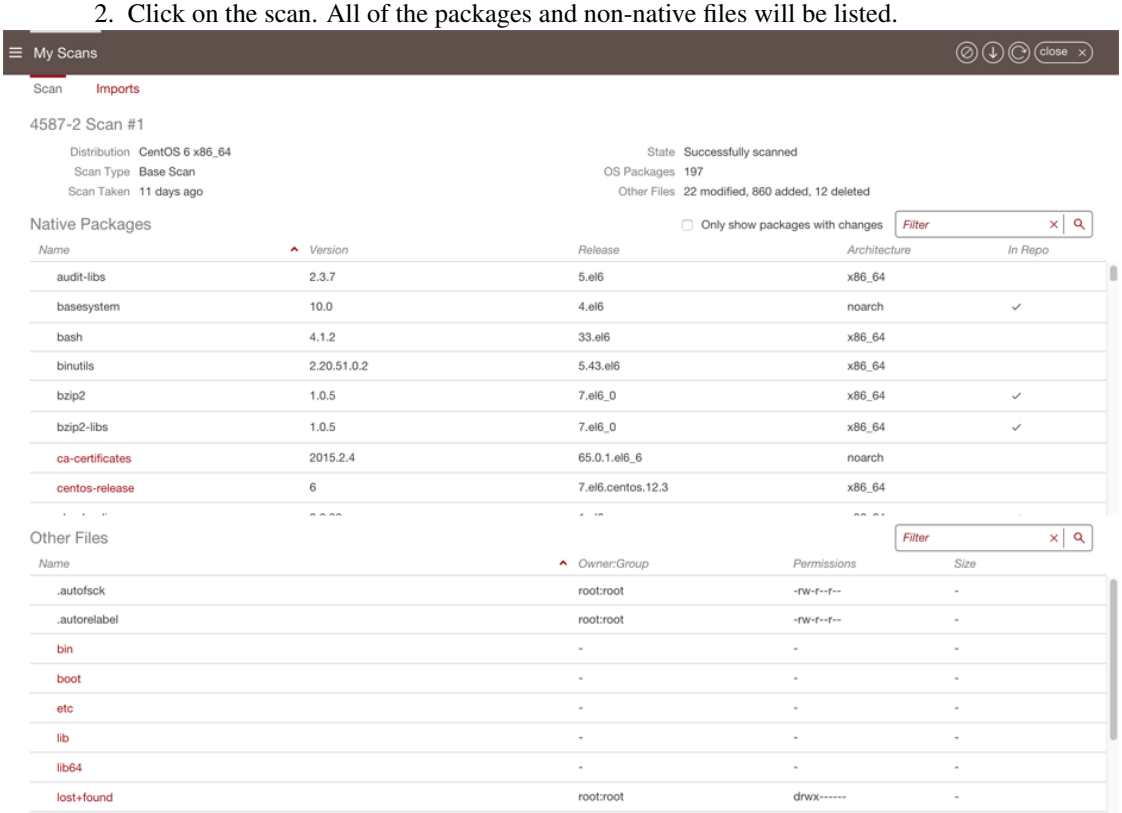
Viewing the Details of Scan

You can view the details of the scan report that will display the packages that are present as well as the filesystem detecting on running source instance.

> **Warning:** This is only supported for Linux based source instances.

To view the details of a scan:

1. Go to the `Migration` tab and click `My Scans`.
2. Click on the scan. All of the packages and non-native files will be listed.



3. You can also filter the packages that have been modified (UForge AppCenter compares the packages scanned with its repo) by checking `Only show the packages with changes`.
4. To view the more details of a package, click on the package name and then the arrow.

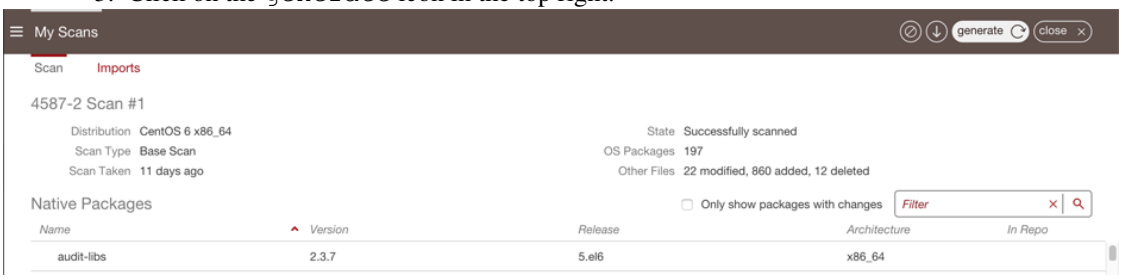
Note: The number of packages between your scanned system and the one in UForge AppCenter will differ for several reasons. First, if you had more than 1 kernel only 1 is imported into UForge AppCenter. Also, UForge AppCenter adds files for install configuration and install profile.

Generate a Machine Image

Once you have scanned a source system, you can generate a machine image directly from the scan report.

To create an image from a scan:

1. Go to the `Migration` tab.
2. Double click on the scan to view details.
3. Click on the `generate` icon in the top right.



4. Select the image format you want.
5. Click `Generate`.

Create an Appliance Template

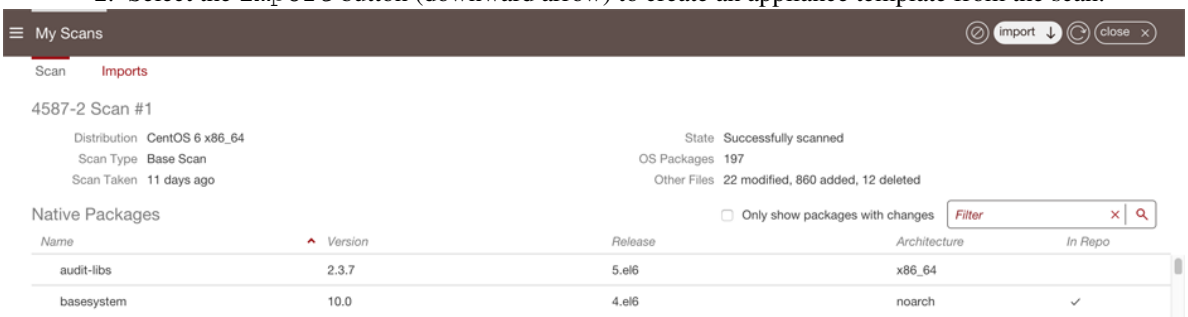
You can create an appliance template from a scan. Once you create an appliance template packages and files that are known will be listed under `OS profile`, while "unknown" packages and files will be listed under `MySoftware`.

Warning: This is only supported for Linux based source instances.

To create an appliance template from a scan:

From the Migration tab:

1. Go to the `Migration` tab > `My Scans`.
2. Select the `import` button (downward arrow) to create an appliance template from the scan.



3. Enter the appliance name and version.

4. Click `import`.

You can now generate a machine image and share it, as you would any other appliance template.

More importantly you can now change the contents of the original scanned system. If you go to the `VM Builder` tab, the new appliance template will be listed in the `imported templates` section. Double-click on it to view the details or modify it.

Changing a Configuration with "No-Console" Features

This is typically the case when migrating to Azure. Azure does not have any console facility today during first boot of the instance. When you migrate a workload in black box, no install type questions are asked on first boot. This is due to:

- root password is copied
- SSH keys are copied
- partitioning table is preserved
- keyboard is preserved
- timezone is preserved

Access to the machine would typically be done via SSH.

White box migration provides more flexibility. As soon as you import a scan, which effectively creates an appliance template, you will have access to the Install Profile. This allows you to reset and change many of the "installation"/first boot parameters including prompting the end user to provide the information (for example: ask the end user to set the root password). Any prompt to the end user is normally displayed in the console. However if the user logs into the machine for the first time via SSH, these prompts are displayed in the SSH terminal and not the console.

> **Warning:** If you decide to prompt the user for the root password, then an SSH key mechanism must already be determined (private key owned by the user and public key set in the Install Profile). Otherwise the user will be locked out of the system with no way to SSH into the machine as no password has been set yet.

Changing Configuration Information

If you want to modify configuration information of a scanned system, there are several solutions:

1. Using pre-install and post-install scripts
2. Using boot scripts
3. Using UForge Studio
4. Integrate with a Configuration Management platform

In each case, you must import the scan as an appliance template (white box migration).

Solution #1: Using pre-install and post-install scripts in the package mechanism: RPM and DEB package mechanism allows you to register scripts that are executed at various moments during the installation of the package. By packaging your middleware or application binaries as a native package you can register such scripts. These scripts are automatically taken into account as part of the machine image generation process. These packages can be added into the appliance template – either in a custom repository known by UForge AppCenter (in this case the packages are displayed as part of the repository and are added in OS Profile) or as part of a Project or My Software component.

Solution #2: Using boot scripts. UForge AppCenter allows you to add boot scripts in the appliance template. These boot scripts are executed the first time the migrated instance is provisioned. Boot scripts can also be registered to be run every time the instance is rebooted.

Solution #3: Using UForge Studio. UForge Studio provides a more comprehensive framework for executing post-configuration scenarios. UForge Studio is a stand alone product that includes a distribution and synchronization mechanism that can also be used on a single instance or even for configuring multi-node deployments. When using UForge Studio, a native package is created that includes the UForge Studio runtime and your configuration scenario. This package is added in the Configuration section of an appliance template.

Solution #4: Integrate with a Configuration Management platform: There are many 3rd party platforms including Puppet, Chef, Ansible and Saltstack that can be used to configure middleware and application layers. Once a system has been migrated or a machine image generated from an appliance template, such configuration management platforms can be used for package update and configuration. You may need to include a bootstrapping mechanism to register the instance to the configuration management platform of your choice. This bootstrapping can be done using boot scripts (see solution #2).

Adding Security Patches

When a scan is imported as an appliance template all the native packages detected from the scan are compared with the UForge AppCenter package repositories. UForge AppCenter will immediately inform you whether new package versions are available for your scan report. Using the "appliance update" feature, a graph is displayed showing you all the available updates and allows the user to update the appliance template to the latest available packages. Once the appliance template has been updated, the user can then generate the machine image and register the machine image on the target environment. The migrated instance will have the latest package updates.

Of course this is not the only way to update a migrated system. The administrator can update the live system using the standard operating system update mechanism. Depending on the operating system this will be yum, apt, yast etc. The administrator can run this update manually, or add a boot script in the appliance template that carries out the update during first boot.

This allows the administrator to decide to use other configuration management platforms (Puppet, Chef, Ansible, Saltstack, BMC Bladelogic to name a few) to manage such updates. For some of these configuration platforms though, you will need to add a boot script as part of the appliance template to bootstrap the running instance with the configuration management platform.

Changing the OS Version of Middleware

Native packages, middleware and application software can be changed or swapped out; and the user can use the `appliance update` mechanism to determine if any package updates are available that can be applied prior to generating and migrating the workload.

Note: Changing the operating system for example from CentOS to Ubuntu is not supported.

For a list of supported OSes for Migration, see the table in *Supported Operating Systems*.

Major OS versions, for example upgrading from CentOS 5.0 to CentOS 6.0 is not supported automatically, though as we have the complete list of operating system packages from the scan, a new appliance template can be constructed with the new operating system version.

This process can further be automated by using the command-line tool hammr (see hammr.io). This tool allows you to create identical machine images from a single configuration file (in JSON). The procedure would be to:

1. Scan the original system (note the scan process can be launched by hammr too)
2. Import the scan as an appliance template (this step can be done by hammr)
3. Export the appliance template using hammr. This will create an archive including a JSON file of all the metadata.

4. Update manually the major version of the operating system in the JSON file.
5. Attempt to import using the new JSON file. A new appliance template will be created with the new major operating system. Note, you may need to iterate on this, if some packages listed in the JSON file are not found (due to potential package renaming).
6. Once the import is done, re-generate which would effectively migrate the system you scanned but with a major operating system upgrade.

Qualification of any middleware and application software is strongly recommended.

Modifying the Scan Overlay

When you import a scan as an appliance template, the overlay created as part of the scan process is registered as a `My Software` component. This `My Software` component is added to the appliance template.

The `My Software` component created from the overlay contains two archives. The first includes all the native package meta-data changes (permissions, ownership changes) and data changes (due to configuration modifications through the lifetime of the live machine). The second archive includes all files that are not part of any native package.

To modify a file in this overlay, you need to download, extract, modify and re-upload it to `MySoftware` once the changes have been made.

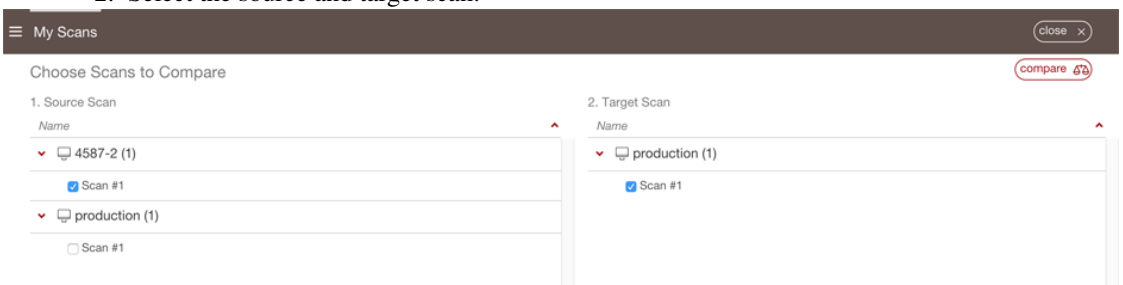
Comparing Scans

As scans are just meta-data, you can compare two scans to determine their differences. This can be used to detect the differences between two live machines (for example between staging and production) or to detect changes of the live machine over time.

> **Warning:** This is only supported for Linux based source instances.

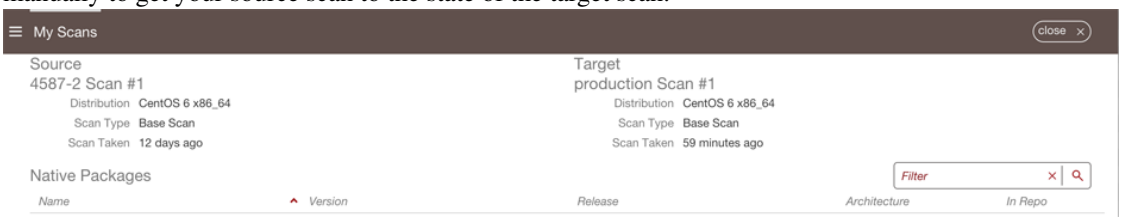
From the `Migration` tab:

1. Click on the `compare` button (balance icon) at the top right hand side of the `My Scans` page.
2. Select the source and target scan.



3. Click `compare`.

UForge lists all the differences between the two systems. The results show the changes you would need to make manually to get your source scan to the state of the target scan.



For example, if you have source scan A and target scan B, in the list, any items that are listed mean they are in scan B but not in scan A. Items that are in strikethrough mean that they were in your source scan A but not in scan B.

CHAPTER 4

Using Workspaces

UForge AppCenter allows users to create workspaces. Only users that have been given the right to create workspaces will be able to create and manage workspaces. If this is not the case, contact your UForge Administrator.

Under the `Collaboration` tab, users can create a number of workspaces, invite other users and share appliance templates with the members of their workspace. A collaboration workspace allows you to restrict the users that can see and use your appliance templates by inviting them to join your workspace. Only users who accept the invitation will be able to view your appliances.

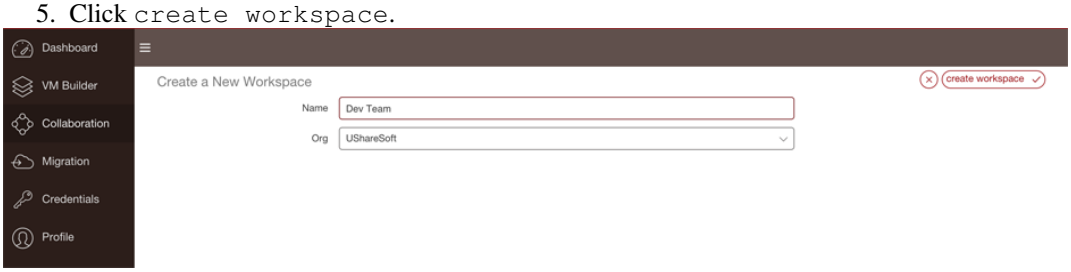
The following sub-sections go into detail on how to manage workspaces:

Creating a Workspace

To create a workspace:

1. Go to the `Collaboration` tab.
2. Click on the plus (+) sign on the top right hand, next to the workspace drop-down menu (in the example below `MySpace`).

3. Enter the workspace name.
4. Select the organization from the drop-down menu.
5. Click `create workspace`.



You can now invite users, share appliance templates, and post comments in the activity stream.

The Activity Stream

Each workspace has an Activity Stream. This is a log of the current actions being carried out by the members of the workspace (for example adding a new appliance template).

Members can also add comments to the workspace for other members of the workspace to see. For example, to ask questions or share information. Your comments will be posted in chronological order in the activity stream.

To add a comment in a workspace, do the following:

1. Go to the `Collaboration` tab.
2. If you have several workspaces, select the workspace you want from the drop-down menu at the top right.
3. Enter your comment in the square.
4. Click `post`.

You can also post comments on a specific appliance that is shared in the workspace. For more information, see *Adding a Comment to a Shared Appliance Template*.

Managing Workspace Members

Inviting Members to Your Workspace

Once you have created a workspace, you can invite users to it. By inviting users, they will be able to view appliance templates you have added to the workspace, and depending upon their access rights share their appliance templates.

To invite users to your workspace:

1. Go to the `Members` page of the `Collaboration` tab.
2. Click on `invite` in the top right.

3. Enter the email addresses of the people you want to invite to your workspace. If the email you specify is not recognized by UForge AppCenter, you will be prompted to invite them to join the platform.
4. You can modify the invitation message. This message will be included in the email inviting members to join your workspace.
5. Click `invite`.

Managing Member Access Rights

Once a user has joined your workspace, you (or another workspace administrator) can modify their status. By default, when users accept your invitation and join your workspace they will be `collaborators`.

Members of a workspace are either:

- `Guest`. A guest can read and post to the activity stream, and import appliances into their private appliance library.
- `Collaborator`. The collaborator has the same basic rights as the Guest, but can also share appliances.
- `Administrator`. This is generally the user who has created the workspace. There must be at least one administrator in a workspace, though there can be more. The administrator can invite or delete members and is able to delete a workspace. The administrator has all the same basic rights as the collaborator.

Deleting a Member

If you are a workspace administrator, you can delete any member of the workspace simply by going to the `Members` page and clicking the small symbol to the right of the member's name and info.

Sharing an Appliance Template in a Workspace

The main purpose of the workspace is to share appliance templates with a group of users.

To share an appliance template:

1. Go to the `Collaboration` tab.
2. If you have several workspaces, select the workspace you want to add your appliance template to from the drop-down menu on the top right.
3. Go to the `Shared Templates` page.
4. Click on `share` in the top right. This lists your private appliance templates.
5. Select an appliance template from your list and click next arrow.
6. Add a description.
7. Click `share` to push the appliance template to the workspace.

Once an appliance template has been shared, other users in the workspace can import it into their private appliance library. They can then modify, use and share the appliance as they wish.

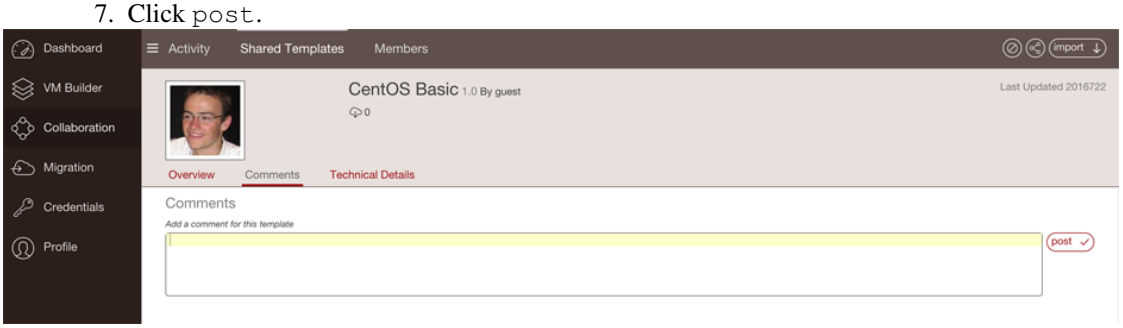
Warning: Any changes you make to your original appliance will NOT be propagated to the shared copy in the collaboration workspace. You will need to share the new version again for members of your workspace to see the changes. The older version will be overwritten.

Adding a Comment to a Shared Appliance Template

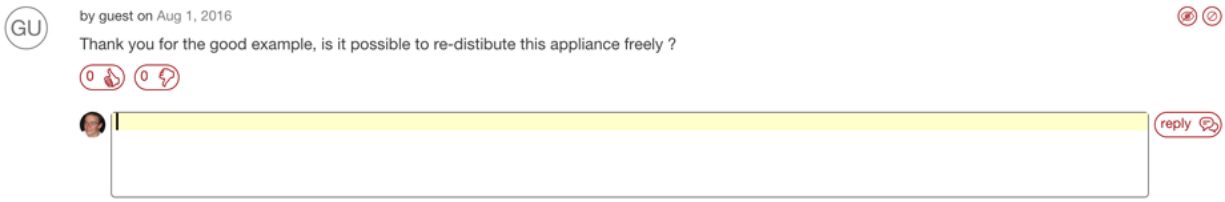
To add a comment to a specific appliance template in a workspace, do the following:

1. Go to the `Collaboration` tab.
2. If you have several workspaces, select the workspace you want from the drop-down menu on the top right.
3. Go to the `Shared Templates` page.
4. Select the appliance template you want to comment on.
5. Click on `Comments`.

6. Add your text.
7. Click `post`.



If you want to reply to a comment, enter your comment below the initial comment and click `reply`.



CHAPTER 5

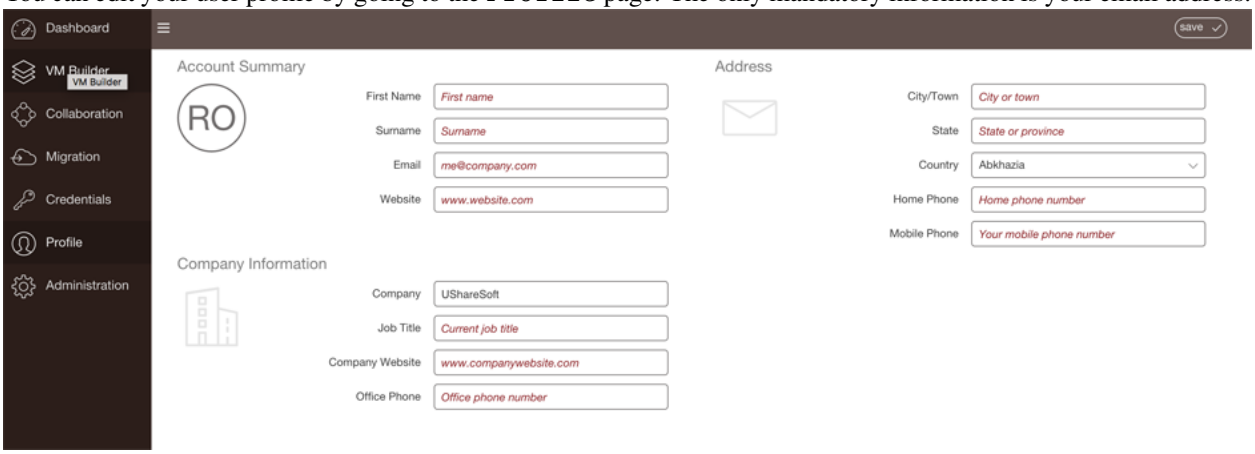
Managing Your Accounts

The `Profile` tab and `Crendentials` tab allow you to modify a large number of your personal details, including your password, Cloud Account information, API Keys and SSH keys.

The following sub-sections go into detail on how to manage your account information:

Modifying Your User Profile

You can edit your user profile by going to the `Profile` page. The only mandatory information is your email address.



You can add or modify your name, company information and address and click `save`.

To add a photo to your profile or modify the existing one:

1. Go to the `My Account` page.
2. Click on the `Profile` tab.
3. Click on the round photo icon on the left hand side.
4. Navigate to the desired image (preferably a .jpeg or .png).
5. Click `Open`.
6. Click `Save`.

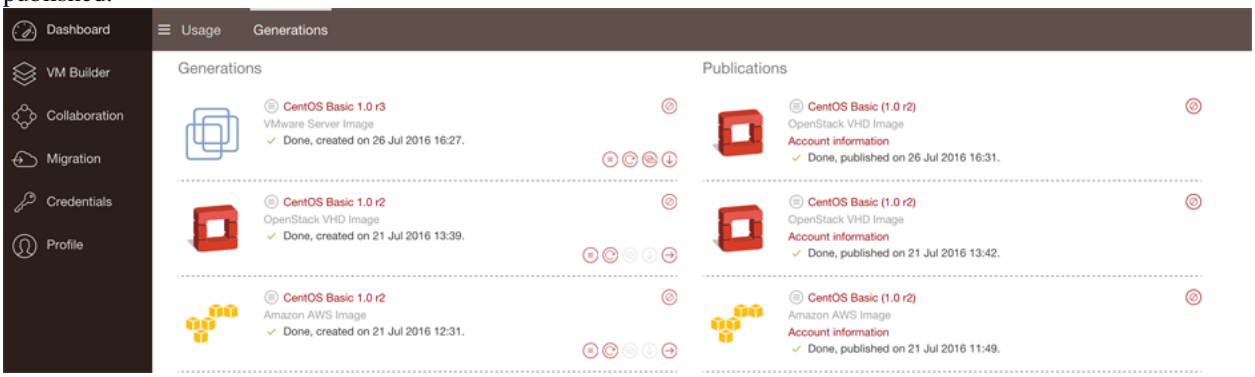
Viewing Your Statistics

When you log in to the UForge Portal, the first tab is the `Dashboard` page. This page presents your user statistics.

On the `Usage` page you can quickly see:

- `Summary`: summary of your account statistics, such as number of appliances created, image generations, number of scans and disk usage. You can also see any quotas set for your account.
- `Appliances`: the number of appliances, clearly indicating the number imported from the UForge Marketplace, Collaboration workspace, or Hammr. The number of current appliances is the number of appliances you have access to, while total includes appliances that have been deleted. You will also see on this page the number of OS types used for your various appliances.
- `Generations`: the number of images generated and published. The number of current generations is the number of images currently linked to your appliances, while total includes all the images, including the ones that have been deleted. You will also see on this page the number of OS types in use for your various appliances.
- `Scans`: the number of scans run for `Migration` service. The number of current scans is the number of scans you have access to, while total includes all the scans, including the ones that have been deleted. You will also see on this page the number of OS types in use for your various appliances.

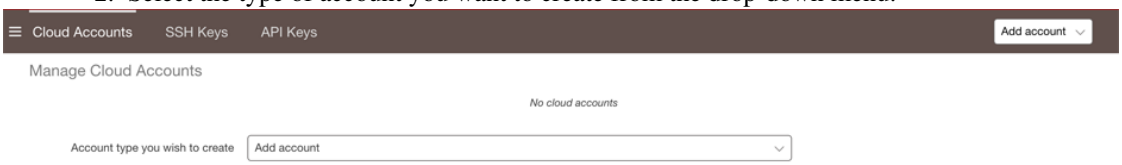
On the `Generations` page you will see a list of the machine images generated (including the ones that failed) and published.



Managing Cloud Accounts

In order to publish an image to a cloud using UForge AppCenter, you will need to add your cloud credentials to UForge. You must have a cloud account prior to setting up your credentials on the platform. Have all the information for your cloud account on hand before starting.

1. Click on the `Credentials` tab (key icon) and go to `Cloud Account`.
2. Select the type of account you want to create from the drop-down menu.



3. Scroll over any given field for more information on the mandatory information to provide for a given cloud account.

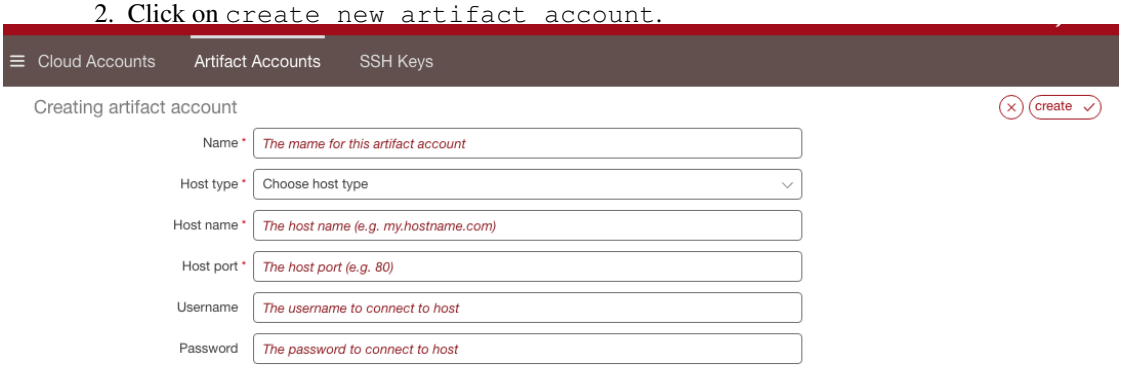
4. Click `Create` to complete.

Note: If you are an Amazon IAM user you will need to generate an X.509 Certificate following these instructions prior to creating your cloud account on UForge.

Managing Your Artifact Accounts

In order to download files for My Software, you can create an artifact account in UForge.

1. Click on the `Credentials` tab (key icon) and go to `Artifact Account`.
2. Click on `create new artifact account`.



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

Managing API Keys

API Keys are used to communicate with UForge AppCenter more securely when using the platform APIs (rather than using basic authentication).

If you have the rights, you will be able to manage your API keys. See your UForge Administrator to be given these rights, if needed.

The number of API key pairs that a user can create is set by the administrator. If you have reached your quota of key pairs, contact your administrator.

To create a key pair:

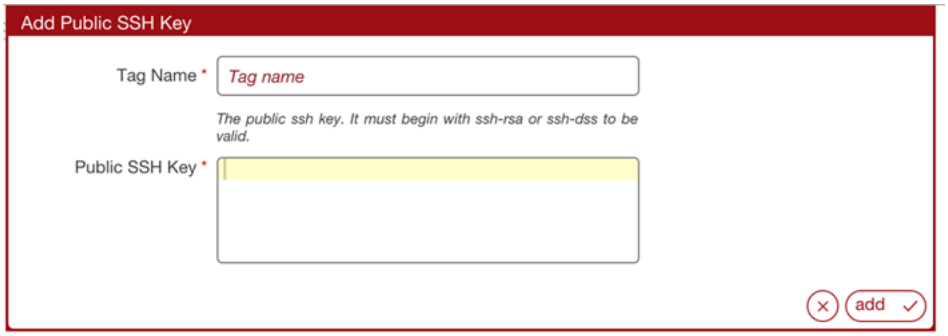
1. Click on the `Credentials` tab (key icon) and go to `API Keys`.
2. Click `create`.

Managing SSH Keys

You can manage one or more SSH keys that can be added to an appliance template.

To add an SSH key:

1. Click on the `Credentials` tab (key icon) and go to `SSH Keys`.
2. Click `create`.
3. Enter the key name and the key details.

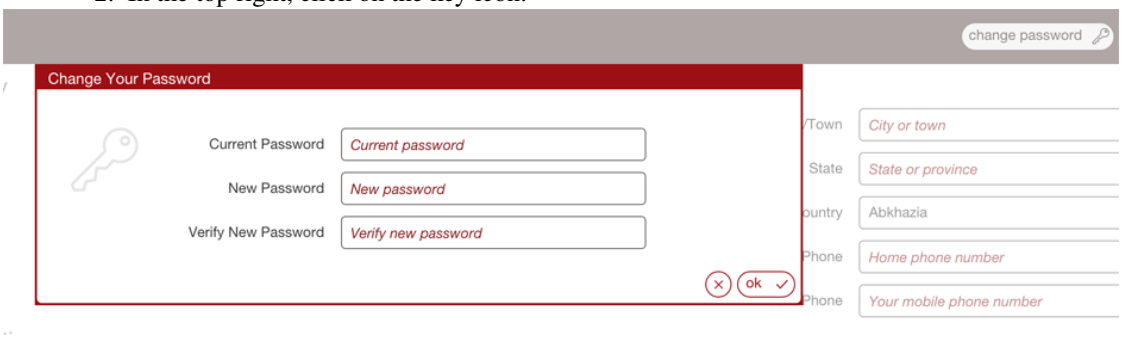


4. Click `add`.

Changing Your Password

To modify your password:

1. Click on the `Profile` tab.
2. In the top right, click on the key icon.



3. Enter your current password and your new password.
4. Click `ok`.

CHAPTER 6

Using the REST API

UForge API is a RESTful resource.

The UForge API follows the design principles of Representational State Transfer (REST). UForge platform provides a set of resources (the API), each of which is referenced with a global identifier (URI). In order to manipulate these resources, clients communicate via the standard HTTP(S) protocol and exchange representations of these resources in a specific format. The documentation notes which formats are available for each method. The API presently supports XML and JSON. To get the results in the format of your choice, the `Accept-Type` header attribute is used in the request.

To make the UForge API even easier to use, UForge has a Java SDK and Python SDK. You can create the API for other languages.

Communication with UForge is done via HTTP(S). For security reasons it is recommended to use HTTPS, however you may submit HTTP requests for debugging purposes.

- GET requests retrieve data
- POST requests create data
- PUT requests modify existing data
- DELETE request destroy existing data

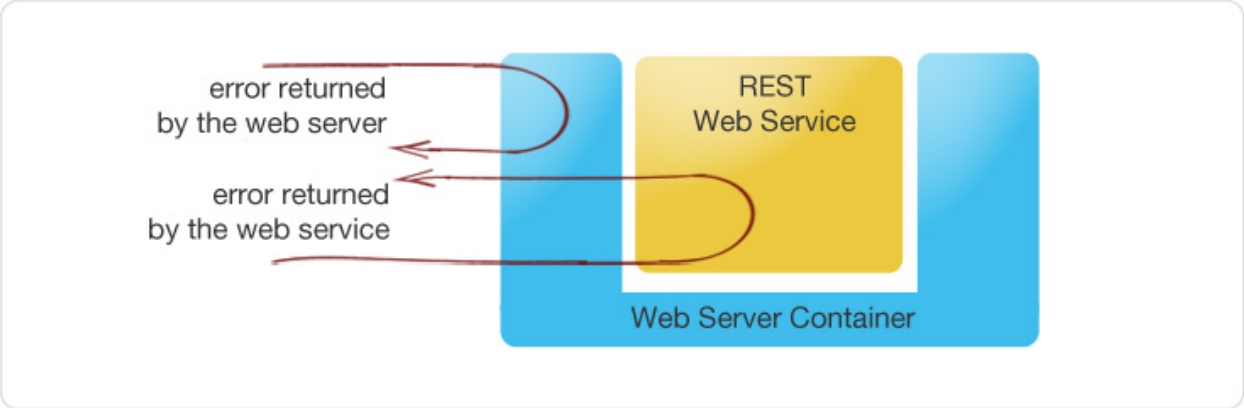
API methods that require a particular HTTP method will return an error if you do not make your request with the correct one. All HTTP methods return codified response codes.

For more information, refer to the *REST APIs documentation <apis:apis-index>*.

Response & Error Codes

The UForge API returns typical HTTP status codes for every request received. HTTP status codes in the 200 range mean that the request was successful, while the 400 and 500 range indicates an error. The status codes in the 300 range are reserved for redirection.

Some of the error codes are returned by the web server container, while all the other response codes are returned by the UForge REST web service. Errors returned by the UForge REST web service may include a more detailed error message indicating why the request failed.



Success Codes

The following codes indicate a successful connection. The response may also include a body section providing the requested information. This information is in a MIME format specified as acceptable in the request.

- `200 OK`: Success. The request was fulfilled.
- `201 Created`: Following a POST command, a new resource has been created. The new resource URI is included in the response.
- `204 No Response`: Server has received the request but there is no information to send back. This is usually the case in a DELETE command, where a resource has been deleted
- `304 Not Modified`: Used when a client does a Conditional GET Request. If the document has not been modified since the date and time specified in If-Modified-Since field, the server responds with a 304 status code and does not send the document body to the client. The purpose of this feature is to allow efficient updates of local cache information (including relevant meta-information) without requiring the overhead of multiple HTTP requests (e.g. a HEAD followed by a GET) and minimizing the transmission of information already known by the requesting client (usually a caching proxy).

Error Codes

The 4xx codes are intended for cases where the client seems to have erred, and the 5xx codes for the cases where the server identifies that the server has erred. It is impossible to distinguish these cases in general, so the difference is only informational. The UForge platform will attempt to provide a detailed error message to help the client diagnose the problem.

- `400 Bad Request`: The request has bad syntax or was inherently impossible to satisfy.
- `401 Unauthorized`: The request did not provide an acceptable authorization parameter. The client should retry the request with a suitable Authorization header.
- `403 Forbidden`: The client does not have the privileges to access this resource. Authorization will not help.
- `404 Not found`: The server did not find anything matching the resource provided in the request.
- `409 Conflict`: Following a POST command, if the resource being created already exists.
- `415 Unsupported Media Type`: The server refuses to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.
- `500 Internal Error`: The server encountered an unexpected condition which prevented it from fulfilling the request.

- `502 Bad Gateway`: UForge is down or being upgraded.
- `503 Service Unavailable`: UForge is overloaded with requests. Try again later.

Sending a Request

The UForge Platform Services are all RESTful services, where 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.

The UForge REST API uses a public and secret API key pair for authenticating each request. The public key is inserted as a query in the request URI. The secret key is then used to encode the entire URI to create a signature using HMAC_SHA1. This signature is then added to the end of the request URI.

Note that you can use Basic Authentication by adding an extra HTTP header `Authorization:Basic username:password`. However, this is less secure. We recommend this only be used on local area networks for instance.

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. Such request URLs resemble the following sample:

```
https://myuforge.example.com:443/ufws-3.3
```

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.

The Request Headers

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.

Request Example

The following is an example of a request sent to an UForge platform with hostname `10.0.0.20` using cURL to get the user `myUser`. Note that the response body (the user information) has been omitted here for clarity:

```
$ curl 'http://10.0.0.20:9090/ufws-3.3/users/myUser?apiKey=XX8Bs2prKPdFrKH_i4rsW7WR0f4FQ05IO7A8vuQUoN

* About to connect() to 10.0.0.20 port 9090 (#0)
* Trying 10.0.0.20... connected
* Connected to 10.0.0.20 (10.0.0.20) port 9090 (#0)
> GET /ufws-3.3/users/myUser HTTP/1.1
> User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8r zlib/1.2.3
> Host: 10.0.0.20:9090
> Accept: application/xml
>

< HTTP/1.1 200 OK
< X-Powered-By: Servlet/2.5
< Server: Sun GlassFish Enterprise Server v2.1.1
< Last-Modified: Thu, 21 Jul 2011 09:43:29 GMT
< ETag: "80f76a81b033572861260548dd748bb3"
< Content-Type: application/xml
```

```
< Transfer-Encoding: chunked
< Date: Thu, 21 Jul 2011 17:02:10 GMT
<
```

The example illustrates the following:

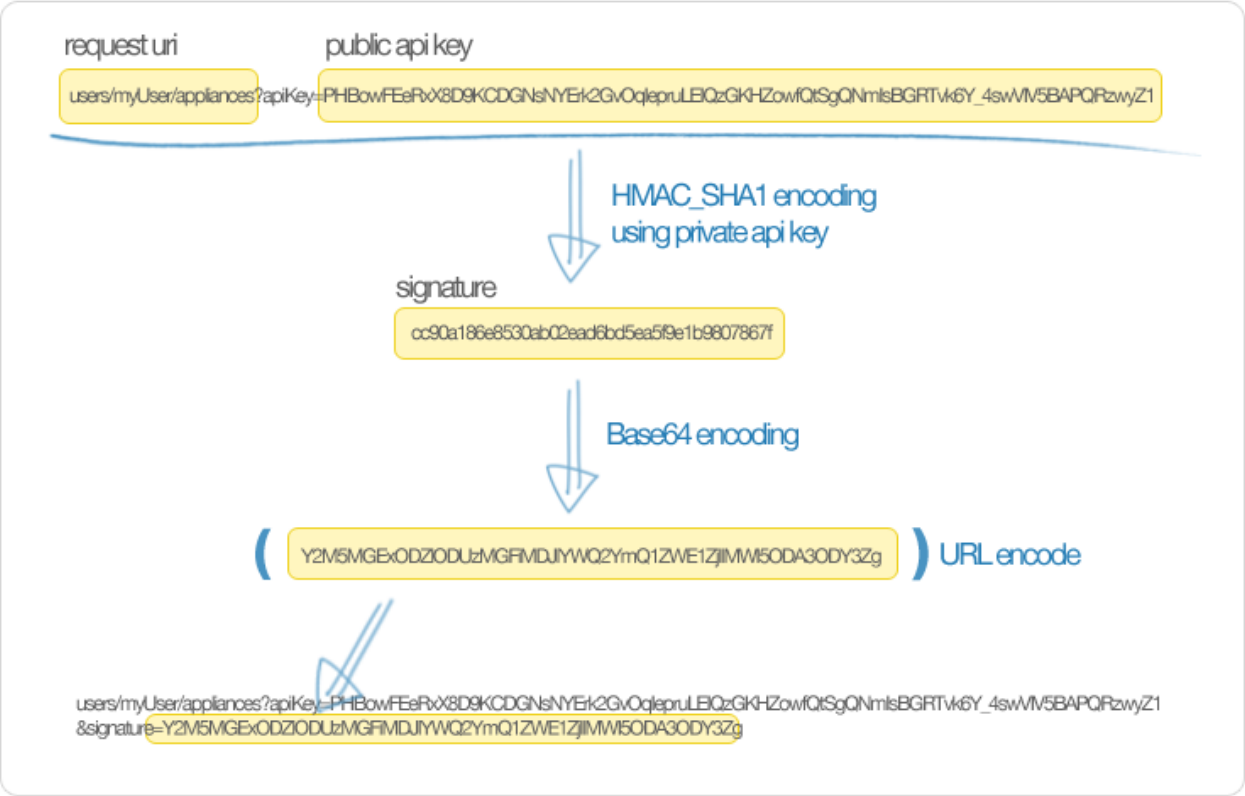
- a `GET` request is sent (cURL by default uses GET) on the resource: `/ufws-3.0/users/myUser`
- an API key is used in this case for authorization
- the `Accept header` is being used to request that the response be sent in XML. Note that, if this header is omitted, UForge sends the response in XML by default.
- the response header includes `ETag` and `Last-Modified` allowing cache validation and a conditional GET requests.

Using the API Keys

To use the UForge AppCenter APIs, it is recommended to use a public and secret API key as part of the request. This allows UForge AppCenter to correctly authenticate and authorize the request. API key pairs are managed on the `My Accounts` page under `API Key`. If you cannot see this tab, then you do not have the right to access UForge via the APIs. Contact your administrator for an initial API key pair.

The API keys are used inside and to sign each request URI to the UForge platform. The creation of a properly signed request URI is done in 5 steps:

1. Add the public API key to the end of the request URI with the query parameter `apiKey`.
2. Encrypt the request URI using `HMAC_SHA1` with your secret API key to create a signature string.
3. Encode the signature string using `Base64`.
4. URL encode the signature string.
5. Add the signature to the end of the request URI created in step 1 with the query parameter `signature`.



Query Parameters

Certain resources within the UForge AppCenter API allow query parameters (or query strings) as part of the request URI. This allows you to pass extra parameters during search requests or to restrict the response data.

To pass a query parameter in a URL, the question mark symbol (?) is used as a separator. For example:

```
http://server/uripath?query_string
```

The query string is composed of one or more field-value pairs, each separated by the equals symbol (=). The series of field pairs is separated by the ampersand symbol (&). For example:

```
http://server/uripath?field1=value1&field2=value2&field3=value3
```

CHAPTER 7

Using the Java SDK

Download and Installing the SDK

To use the UForge AppCenter Java API, you need the jar files. You can either download the jar bundle and then add the dependency jar files to your classpath, or use maven to handle all the dependencies by adding the following to your `pom.xml` file:

```
<dependency>
      <groupId>com.usharesoft</groupId>
      <artifactId>uforge-client</artifactId>
      <version>3.6</version>
</dependency>

<repository>
        <id>maven.usharesoft.com</id>
        <name>UShareSoft Repository</name>
        <url>http://maven.usharesoft.com/repository/official/</url>
        <layout>default</layout>
</repository>
```

To use the API, you must have:

- The base URL of the UForge AppCenter platform you will be communicating with, for example: `https://factory.usharesoft.com:443`
- An account on the UForge AppCenter platform you will be communicating with
- An API key pair (secret and public) for the account (or use Basic Authentication)

Communicating with UForge

The `UForgeConnector` class provides all the lower level communication with UForge AppCenter by leveraging the Jersey client API. This class creates HTTP packets with the correct header information and constructs the request URI to authenticate the request (using the secret and public keys). The response is parsed by JAXB to provide POJO Java classes of the response information.

Normally the first step is to get the user information of the account being used to authenticate. The response provides the URIs to the organizations, appliances and software this user has access to. The code below shows how to recuperate the user information. Note, as UForge is completely RESTful, when the method `login()` is used, no session is created between the client and UForge. Each request will reuse the authentication information stored in this `UForgeConnector` instance.

```
import com.usharesoft.client.common.connector.UForgeConnector;
import com.usharesoft.client.common.objects.User;
import org.apache.log4j.Logger;

private static String UFORGE_BASE_URI = "https://factory.usharesoft.com:443";
private static String USER_NAME = "myUser";
public static String SECRET_API_KEY = "b9024Ox-N5jvPqzYw8I1fYvFRuSQv9sFuNM30gNAwZ4RqY5nOt2zPdB8XyOS2h

public static String PUBLIC_API_KEY = "PHBowFEeRxX8D9KCDGNsNYErk2GvOqIepruLEIQzGKHZowfQtSgQNmIsBGRTvh

UForgeConnector connector = new UForgeConnector(UFORGE_BASE_URI, USER_NAME, PUBLIC_API_KEY, SECRET_AP
User me = connector.login();
logger.info("Successfully connected to UForge. User: " + me.getLoginName());
```

Creating an Appliance Template

An Appliance Template contains the model of the software stack. The model includes all the operating system packages, middleware and application software for generating an image that can be provisioned on a virtual or cloud platform. To create an appliance template, you need to decide which operating system to build the template from, as well as the name and version.

When creating an appliance or choosing an operating system, you must choose the organization where to create the appliance or to search an operating system. The user must be a member of the organization to have authorization to search the organization resources. By default a user will be a member of at least one organization.

The following code provides an example of constructing an appliance template.

```
// require to have the organization information in the connector when getting distributions
// and creating appliances
// get the first organizaton of the user
URI orgUri = me.getOrgUris().getUris().iterator().next();
OrgDelegate orgDelegate = new OrgDelegate(connector);
Org org = orgDelegate.get(orgUri);
connector.setOrg(org);

// Use the connector holding the user authentication information to get
// the operating system to use for constructing the appliance template
DistributionDelegate distributionDelegate = new DistributionDelegate(connector);
Distribution distribution = distributionDelegate.get(Distribution.CENTOS_5_6_I386);

// Create the appliance
ApplianceDelegate applianceDelegate = new ApplianceDelegate(connector);
Appliance appliance = new Appliance(distribution, "WordPress", "3.2.1");
appliance = applianceDelegate.create(appliance);
```

Adding an OS Profile

An Appliance Template must contain an operating system profile. This profile contains a subset of operating system packages required by the middleware and application software to run correctly. Each operating system provided by UForge contains a set of standard operating system profile templates to choose from. These contain commonly used package bundles for the operating system to run, providing the basic operating system services.

The "Minimal" OS profile contains the minimum set of packages for the operating system to run properly and provide a minimum set of networking services and administration tools.

The following code shows how to create a new OS profile from a standard OS profile template and add it to an Appliance Template.

```
// Get the minimum template from the distribution
DistribProfileTemplate osProfileTemplate = distributionDelegate.getProfile(distribution, DistribProf

// Create the os profile that will be added to the appliance from the template
DistribProfile osProfile = new DistribProfile(osProfileTemplate);
osProfile.setDistribProfileTemplate(osProfileTemplate);

// Create this os profile in the appliance
ApplianceOSProfileDelegate aospDelegate = new ApplianceOSProfileDelegate(connector);
osProfile = aospDelegate.create(this.appliance, osProfile);
```

Extra packages can be added to the appliance template's OS profile.

```
// Get the current list of packages in the os profile
PkgList pkgs = aospDelegate.getPkgs(osProfile);

// Add more packages
// php53
Pkg pkg = distributionDelegate.getPkg(distribution, "php53");
if (pkg == null) {
    logger.error("Unable to retrieve the package php53 for this distribution");
    return;
}
pkgs.addExtraPackage(pkg);

// php53-common
pkg = distributionDelegate.getPkg(distribution, "php53-common");
if (pkg == null) {
    logger.error("Unable to retrieve the package php53-common for this distribution");
    return;
}
pkgs.addExtraPackage(pkg);

// php53-cli
pkg = distributionDelegate.getPkg(distribution, "php53-cli");
if (pkg == null) {
    logger.error("Unable to retrieve the package php53-cli for this distribution");
    return false;
}
pkgs.addExtraPackage(pkg);

// php53-mysql
pkg = distributionDelegate.getPkg(distribution, "php53-mysql");
if (pkg == null) {
    logger.error("Unable to retrieve the package php53-mysql for this distribution");
    return false;
}
pkgs.addExtraPackage(pkg);

// update the os profile with the new package list
aospDelegate.updatePkgs(osProfile, pkgList);
```

Generating a Machine Image

Once you are happy with the contents of an appliance template, you can then generate a machine image to practically any hypervisor or cloud environment. The following code generates a CloudStack VHD image (for Xen hypervisor). For some image types you can select the disk size and the RAM of the virtual machine to be created. These can be updated once provisioned in the cloud environment. If you have set advanced partitioning in the installation profile, then this will be used instead for the disk size. The generation is done asynchronously; the generation status gives the progress of this generation.

```
ImageDelegate imageDelegate = new ImageDelegate(connector);
Image image = new Image(appliance, ImageFormat.CLOUDCOM_VHD_FORMAT;));
image.setCompress(true);   // create an archive (.gz)
image.setVmDiskSize(4096); // 4GB
image.setVmMemorySize(256); // 256MB

// Launch the generation
image = imageDelegate.generate(appliance, image);

// Check the generation status every 5 seconds
OpStatus status = imageDelegate.getStatus(image);
while ( status.isComplete() == false ) {
    try {
        Thread.sleep(5000);
    } catch (InterruptedException ex) {
        //Error in a thread while trying to get the status of the cloud.com image generation
    }
    // Get the status
    status = imageDelegate.getStatus(image);
}

// Generation complete!
```

Publishing an Image

UForge has connectors to many of the popular cloud platforms including Amazon, Microsoft Azure, Google Compute Engine, OpenStack, CloudStack, Eucalyptus and Flexiant to name a few. Once an image has been generated you can either download the image or publish directly to a cloud environment using your own cloud account credentials. Like generations, publishing images is asynchronous. You can get the progress of the publish from the publish status. The following code publishes a generated VHD image directly to the template library of a CloudStack environment.

```
// Create the credential information to communicate with the Cloud.com environment
CredAccount credAccount = new CredAccount();
credAccount.setType(InfraType.CLOUD_COM);
credAccount.setName("My Cloud.Com Account");
credAccount.setServerUrl(new URI("http://10.0.1.251:8080/client/api"));
credAccount.setPublicAPIKey("8pqqg0HV8ocpt6j8qYiCpDZ4cqzbtLaxCErIOpCD0r9VOjnILgahX85_J2CFvC8863en3NG
credAccount.setSecretAPIKey("9Q-vVxokmMbI_l4t7aAfbocTgoLBlnt41Xy6iLZUfC6PzAdXNy2rRegAWhBMF3mQ9jk4MtP

// Add the zone on where the image should be published
credAccount.setZoneName("zone1");

// Provide information on the image being uploaded to the Cloud.com template library
credAccount.setDisplayName("WordPress image");
credAccount.setDescription("WordPress image for the cloud.com platform");
credAccount.setPasswordEnabled(true);
```

```
credAccount.setFeaturedEnabled(false);

// Allow this image to be accessed to all cloud.com users
credAccount.setPublicImage(true);

// Publish the image
PublishDelegate publishDelegate = new PublishDelegate(connector);
PublishImage publishImage = new PublishImage();
publishImage.setCredAccount(credAccount);
publishImage.setImage(image);
publishImage = publishDelegate.publish(appliance, publishImage);

// Get the status of the publish
OpStatus status = publishDelegate.getStatus(publishImage);
while (status.isComplete() == false) {
    try {
        Thread.sleep(5000);
    } catch (InterruptedException ex) {
        // Error in a thread while trying to get the status of the iso image generation
    }
    // get the status
    status = publishDelegate.getStatus(publishImage);
}

// Publish finished
// Can check for errors
if (status.isError()) {
    // error occurred during the publish
}
```

Adding a Project from the Project Catalog

Each UForge organization provides a Project Catalog where commonly used software components can be added to an appliance template. The Project Catalog must belong to the same organization as the appliance. The following code adds some projects to an appliance template.

```
ProjectDelegate projectDelegate = new ProjectDelegate(connector);

// Add Apache HTTP Server
Project project = projectDelegate.get(distribution, "Apache HTTP server", "2.2.3", null);
if (project == null) {
    logger.error("Unable to retrieve project");
    return;
}
appliance.addProject(project);

// Add MySQL
Project project = projectDelegate.get(distribution, "MySQL5 Server", "5.0.77", null);
if (project == null) {
    logger.error("Unable to retrieve project");
    return;
}
appliance.addProject(project);

// Save the updated appliance instance
applianceDelegate.update(appliance);
```

Uploading a Software Component

You can upload your own software components to a private software library (`My Software Library`). This software can then be added to any of your appliance templates. This provides a mechanism to compliment the Project Catalog. The following code shows how to upload files, attach a license and then add it to an appliance template.

```
MySoftwareDelegate mySoftwareDelegate = new MySoftwareDelegate(connector);
File wpf = new File("wordpress-3.2.1.zip");
File lf = new File("wp-license.html");

// Create a My Software component and upload the wordpress zip file
MySoftware mySoftware = mySoftwareDelegate.upload("WordPress", "3.2.1", wpf);

// Attach a license
mySoftware = mySoftwareDelegate.uploadLicense(mySoftware, lf);

// Add the software component to the appliance
appliance.addMySoftware(mySoftware);

// Save the changes to the appliance
applianceDelegate.update(appliance);
```

Adding a Boot Script

Boot scripts can be added to the appliance template allowing initial configuration to be executed either during the first time the image is started or during every boot of the image.

The following code shows how to upload a boot script to an appliance.

```
ApplianceConfigDelegate configDelegate = new ApplianceConfigDelegate(connector);

// upload a boot script to the appliance
File bsf = new File("myscript.sh");
BootScript bootscript = new BootScript();
// only execute this boot script once during first boot
bootscript.setBootType(BootScript.FIRST_BOOT);
bootscript = configDelegate.uploadBootScript(appliance, bootscript, bsf);
```

CHAPTER 8

Using the Python SDK

Download and Installing the SDK

The Python API is supported on all major operating systems: Linux, Mac-OS, and Windows. The easiest way to install the API is using `pip`, the widely used package management system for installing and managing software packages written in Python.

Note: This API is used by the open source Hammr project. You can find more examples on how to use the API in the source code of this project.

Installing pip

If you already have `pip` installed on your system, you can skip this section.

To install or upgrade pip, download this file:

```
https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py
```

Then run the command:

```
$ python get-pip.py
```

For more information on installing pip, refer to the official pip documentation: http://www.pip-installer.org/en/latest/installing.html

Installing UForge Python API

Once `pip` has been installed, you can now install the UForge Python API packages. You may have to run this command as `sudo` or `administrator`.

See below the instructions for installing the Python API for your target platform:

For Linux

First, you need to install extra packages on your system, and then install the SDK package.

Debian

```
$ apt-get install python-dev gcc
$ pip install uforge-python-sdk
```

Red Hat and CentOS

```
$ yum install gcc python-devel
$ pip install uforge-python-sdk
```

Upgrading the SDK

To upgrade an already installed SDK, run:

```
$ pip install --upgrade uforge-python-sdk
```

For Mac OS

As a pre-requisite, you need to have XCode installed.

To install the SDK run:

```
$ export ARCHFLAGS="-Wno-error=unused-command-line-argument-hard-error-in-future"
$ pip install uforge-python-sdk
```

Upgrading the SDK

To upgrade an already installed SDK, run:

```
$ pip install --upgrade uforge-python-sdk
```

For Microsoft Windows

Run the following command:

```
c:\Python27> .\Scripts\easy_install.exe uforge-python-sdk
```

Upgrading the SDK

To upgrade an already installed SDK, run:

```
c:\Python27> .\Scripts\easy_install.exe --upgrade uforge-python-sdk
```

Communicating with UForge

The Python API provides all the lower level communication with UForge AppCenter by creating HTTP request packets with the header information to authenticate correctly.

Normally the first step is to get the user information of the account being used to authenticate. The response provides the URIs to the organizations, appliances and software this user has access to. The code below shows how to recuperate the user information. Note, as UForge is completely RESTful, when the method `login()` is used, no session is created between the client and UForge. Each request will reuse the authentication information stored in this `api` instance.

```
# Import the Uforge python API
from uforge.application import Api
# Import http client
import httplib2
import base64

# UForge API instantiation (optional: disable self signed certificate check)
client = httplib2.Http(disable_ssl_certificate_validation=True)

login='username'
passwd='password'

# Header for authentication
headers = {}
headers['Authorization'] = 'Basic ' + base64.encodestring( login + ':' + passwd )

# Create the API object
api = Api('https://10.0.0.1/ufws-3.3', client = client, headers = headers)

# Send a rquest (getting the user object)
user = api.Users(login).Get()
if user is not None:
print user.loginName + ' - ' + user.email
```

Note: All UForge Python objects are in the file `objects.py`, contained in the Python SDK. You can find all attribute names for each object type.

Creating an Appliance Template

An Appliance Template contains the model of the software stack. The model includes all the operating system packages, middleware and application software for generating an image that can be provisioned on a virtual or cloud platform. To create an appliance template, you need to decide which operating system to construct the template from, as well as the name and version.

When creating an appliance or choosing an operating system, you must choose the organization where to create the appliance or to search an operating system. The user must be a member of the organization to search the organization resources.

The following code provides an example of constructing an appliance template.

```
# TODO
```

Adding an OS Profile

An Appliance Template must contain an operating system profile. This profile contains a subset of operating system packages required by the middleware an application software to run correctly. Each operating system provided by

UForge contains a set of standard operating system profile templates to choose from. These contain commonly used package bundles for the operating system to run, providing the basic operating system services.

The "Minimal" OS profile contains the minimum set of packages for the operating system to run properly and provide a minimum set of networking services and administration tools.

The following code shows how to create a new OS profile from a standard OS profile template and add it to an Appliance Template.

```
# TODO
```

Generating a Machine Image

Once you are happy with the contents of an appliance template, you can then generate a machine image to practically any hypervisor or cloud environment. The following code generates a CloudStack VHD image (for Xen hypervisor). For some image types you can select the disk size and the RAM of the virtual machine to be created. These can be updated once provisioned in the cloud environment. If you have set advanced partitioning in the installation profile, then this will be used instead for the disk size. The generation is done asynchronously.

```
# TODO
```

CHAPTER 9

Hammr Command Line Tool

UForge AppCenter provides an open source command-line tool called `hammr` to allow users to create machine images for different environments from a single configuration file.

Hammr is a lightweight client-side tool based on Python, and can be installed on all major operating systems.

Hammr can be used as part of your "DevOps tool chain" and in conjunction with other tools such as Jenkins, Chef, Puppet and SaltStack, allowing you to easily build your machine images and maintain your live running instances. Hammr also has migration capabilities, allowing you to scan a live system, generate a machine image for a different environment as well as export it back to a configuration file for sharing.

Getting the Source Code

All the source code is available on GitHub.

Further Reading

All the documentation on installing and using hammr can be found on: http://hammr.io

CHAPTER 10

Using Python CLI

You can manage UForge AppCenter using the Python command-line interface.

Installing Python CLI

To install the Python CLI run:

```
yum install uforge-cli
```

Launching Python CLI

To launch Python CLI locally on the machine, launch:

```
/opt/UShareSoft/uforge/cli/bin/uforge -u $UFORGE_WEBSVC_LOGIN -p $UFORGE_WEBSVC_PASSWORD -U http://ws
```

When accessing from outside the machine, use: https://IP-OF-MACHINE/api

CHAPTER 11

Changelog

3.6-fp1

Release Date: 2016-10-31

New Features

- Import/Export of appliance templates in the user interface
- **Software (MySoftware) and Project bundles now consolidated. New features added including:**
  - pulling files from remote locations (HTTP, FTP endpoints) so the user no longer requires to upload software components to the platform
  - pulling files can be cached for future generations or pulled on each generation
  - file permissions added for files and directories
  - can create directory structures in a software bundle
  - can add tagging information to a software bundle
  - can add native packages from OS respositories to a software bundle
  - can add boot scripts to a software bundle
  - identify the software bundle only being supported on a subset of operating systems
- API keys can be used for authentication when running a scan for migration.
- Scan messages and error messages cleaned up and more understandable
- Japanese language localization for the UI

Bug Fixes

- 5293 Image generation error: Windows image must have at least 512 MB of memory
- 5729 Issues with migration from 3.5.1. to 3.6
- 5465 Build fails due to unreachable rpm-4.11.2.tar.bz2
- 5740 Fix DB schema checks
- 5331 AWS publish no longer works

- 5637 Windows generation from scan fails at boot
- 5427 Unable to generate a virtual machine with LVM inside a MSDOS disk
- 5291 All combo boxes are empty when a value has been selected
- 5876 Logo broken on Dashboard
- 5444 Unable to populate Fedora/RHEL distributions
- 5420 When a template is remobed from a workspace, a DELETE error appears in the log file
- 5527 Subscription info does not list the frequency of quotas
- 5494 Scan fails because of files of type c (character device file)
- 5483 The service command not found in a machine generated by UForge
- 5442 The file deletion of Project fails
- 5429 Root can disable root account in UForge CLI
- 5746 Timeout of 10 seconds for the UForge CLI is not usable
- 5563 Internal error in Migration tab
- 5558 500 Call Fail Error when generating an image from scan
- 5556 The targetformat of Amazon is not displayed when generating an image
- 5553 If a scan is deleted, the image generated from the same scan is not deleted
- 5551 Spelling mistake in UI when publishing to Flexiant
- 5549 The error of Keystone version is displayed in Keystone Server URL
- 5403 Scan fails when trying to rebuild a non repo package

CHAPTER 12

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.

UNIX is a registered trademark of the Open Group in the United States and in other countries.

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.

Other company names and product names are trademarks or registered trademarks of their respective owners.

CHAPTER 13

Copyright FUJITSU LIMITED 2017

All rights reserved, including those of translation into other languages. No part of this manual may be reproduced in any form whatsoever without the written permission of FUJITSU LIMITED

CHAPTER 14

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 15

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.