



Lab 3.1: Prepare Installation

Prepare the installation of Foreman

Make Puppetlabs repository available

Install the release package provided by the Puppetlabs repository to make it available for package installation of open source version of Puppet.

```
# yum install http://yum.puppetlabs.com/puppetlabs-release-el-7.rpm -y
```

Make EPEL repository available

Install the release package provided by the EPEL repository to make it available for package installation required as dependency.

```
# yum install http://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm -y
```

Make Foreman repository available

Install the release package provided by the Foreman repository to make it available for package installation of Foreman and its components.

```
# yum install http://yum.theforeman.org/releases/latest/el7/x86_64/foreman-release.rpm -y
```

Install foreman-installer

Install the package "foreman-installer" from the now available repositories.

```
# yum install foreman-installer -y
```



Lab 3.2: Install an All-in-one setup

Install an All-in-one setup of Foreman with DNS and DHCP

Run foreman-installer with additional parameters

To install run the following command:

```
# foreman-installer \
--foreman-proxy-dns=true \
--foreman-proxy-dns-interface=eth0 \
--foreman-proxy-dns-zone=localdomain \
--foreman-proxy-dns-reverse=0.0.10.in-addr.arpa \
--foreman-proxy-dns-forwarders=8.8.8.8 \
--foreman-proxy-dns-forwarders=8.8.4.4 \
--foreman-proxy-dhcp=true \
--foreman-proxy-dhcp-interface=eth0 \
--foreman-proxy-dhcp-gateway=10.0.0.1 \
--foreman-proxy-dhcp-range="10.0.0.100 10.0.0.200" \
--foreman-proxy-dhcp-nameservers="10.0.0.1"
```

This will output on success something similar:

Success!

- * Foreman is running at <https://foreman.localdomain>
Initial credentials are admin / PASSWORD
- * Foreman Proxy is running at <https://foreman.localdomain:8443>
- * Puppetmaster is running at port 8140
- * The full log is at `/var/log/foreman-installer/foreman-installer.log`

With the provided credentials login to '<https://foreman.localdomain>' using your browser.



Lab 3.3: Add DNS configuration to Foreman

Create the domain 'localdomain' and associate Smart proxy

Login to Foreman

With the provided credentials login to <https://foreman.localdomain> using your browser.

Navigate to 'Infrastructure > Domains'

Add the domain 'localdomain' and associate Smart proxy 'foreman.localdomain'

Click on 'New Domain' and in the dialog insert:

- DNS domain: 'localdomain'
- DNS Proxy: 'foreman.localdomain'

If the domain 'localdomain' was already created click on the domain name and change the DNS Proxy to 'foreman.localdomain'.

Press 'Submit' to store the configuration.



Lab 3.4: Add DHCP configuration to Foreman

Create the subnet 'foreman' and associate Smart proxies

Login to Foreman

With the provided credentials login to <https://foreman.localdomain> using your browser.

Navigate to 'Infrastructure > Smart Proxies'

Add the subnet 'foreman' by importing from the Smart Proxy

Select 'Import subnets' from the drop down menu next to the Smart Proxy 'foreman.localdomain' and in the dialog insert:

- Subnet tab:
 - Name: 'foreman'
 - Network address: '10.0.0.0'
 - Network mask: '255.255.0.0'
 - Gateway address: '10.0.0.1'
 - Primary DNS server: '10.0.0.1'
 - IPAM: 'DHCP'
 - Start of IP range: '10.0.0.100'
 - End of IP range: '10.0.0.200'
 - Boot mode: 'DHCP'
- Domain tab:
 - Select 'localdomain'
- Proxies tab:
 - DHCP Proxy: 'foreman.localdomain'
 - TFTP Proxy: 'foreman.localdomain'
 - DNS Proxy: 'foreman.localdomain'

Press 'Submit' to store the configuration.



Lab 4.1: Prepare PXE installation of CentOS

Prepare the installation of CentOS using PXE

Change the Installation media "CentOS mirror" to the local repo

Navigate to "Hosts > Installation media", select the entry "CentOS mirror", change the "Path" to the URL the trainer provided.

Associate the PXELinux template "Kickstart default PXELinux" with CentOS

Navigate to "Hosts > Provisioning templates" and search "Kickstart default PXELinux". Click it and in the dialog on the "Association" tab move the entry for CentOS to the "Selected Items". Click on "Submit".

Associate the Provision template "Kickstart RHEL default" with CentOS

Navigate to "Hosts > Provisioning templates" and search "Kickstart RHEL default". Click it and in the dialog on the "Association" tab move the entry for CentOS to the "Selected Items". Click on "Submit".

Associate the operating system with the Partition table "Kickstart default", Installation media "CentOS mirror", select the Templates and set parameter "enable-puppetlabs-repo" to "true"

Navigate to "Hosts > Operating systems" and search the CentOS entry. Click it and in the dialog on the "Operating System" tab change the "Minor version" to "2.1511", on the "Partition table" tab select "Kickstart default", on the tab "Installation media" select "CentOS mirror", on the "Templates" tab select the templates associated earlier and on the "Parameters" tab add the parameter "enable-puppetlabs-repo" with value "true". Click on "Submit".



Lab 4.2: Prepare PXE installation of Debian

Prepare the installation of CentOS using Debian

Change the Installation media "Debian mirror" to the local repo

Navigate to "Hosts > Installation media", select the entry "Debian mirror", change the "Path" to the URL the trainer provided.

Create the Operating system "Debian" with Major version "8", Minor version "3", Description "Debian jessie", Family "Debian", Release name "jessie"

Navigate to "Hosts > Operating systems" and click on "New operating system". In the dialog on the "Operating System" tab set the "Name" to "Debian", "Major version" to "8", "Minor version" to "3", "Description" to "Debian jessie", "Family" to "Debian", "Release name" to "jessie". Click on "Submit".

Associate the PXELinux template "Preseed default PXELinux" with Debian

Navigate to "Hosts > Provisioning templates" and search "Preseed default PXELinux". Click it and in the dialog on the "Association" tab move the entry for Debian to the "Selected Items". Click on "Submit".

Associate the Provision template "Preseed default" with Debian

Navigate to "Hosts > Provisioning templates" and search "Preseed default". Click it and in the dialog on the "Association" tab move the entry for Debian to the "Selected Items". Click on "Submit".

Associate the finish template "Preseed default finish" with Debian

Navigate to "Hosts > Provisioning templates" and search "Preseed default finish". Click it and in the dialog on the "Association" tab move the entry for Debian to the "Selected Items". Click on "Submit".

Associate the operating system with the Templates and set Architecture "x86_64", Partition table "Preseed custom LVM" and "Preseed default", Installation media "Debian mirror"

Navigate to "Hosts > Operating systems" and search the Debian entry. Click it and in the dialog on the "Operating System" tab select the "Architecture" to "x86_64", on the "Partition table" tab select "Preseed custom LVM" and "Preseed default", on the tab "Installation media" select "Debian mirror", on the "Templates" tab select the templates associated earlier. Click on "Submit".



Lab 4.3: Create a virtual machine "pxe"

Create a virtual machine "pxe" for PXE installation

Open "Virtual Machine Manager" application

Open the "Virtual Machine Manager" application from the Gnome Application menu

Select "New virtual machine" from the menu or by pressing the button

Select PXE boot

In the first step of the wizard select the network boot

Select "Linux" and "Red Hat Enterprise Linux 7.2" for CentOS or "Debian jessie" for Debian according to your preferences

In the second step choose "Linux" as operating system type and according to your preferences as version "Red Hat Enterprise Linux 7.2" for CentOS or "Debian jessie" for Debian. For later labs you will also have to choose an operating system, deploy at least one of every kind.

Keep the minimum requirements for RAM, CPU and Disk

In the third step stick with the minimum requirements for RAM and CPU because several systems will be deployed. In the fourth step keep the disk as small as possible but it will not matter as long as it is thin provisioned.

Name your virtual machine "pxe" and select the network "foreman"

Name the virtual machine "pxe" so you know which system was deployed in which lab. Select the network "foreman" instead of "default".

Create the virtual machine and immediately pause it so in the next lab the required configuration in Foreman can be created

Finish the wizard and immediately pause it before the boot times out.

If you did not pause the system in time, you will have to add the NIC as Boot media and change the boot order in the details view of the virtual machine. This change shutdowns the machine; affects the system only after powering it off. Shut it down and start it up again later for the next lab.

Verbesserung: (This change needs a shutdown of the VM, it applies only to the VM after powering it off. Shut down the system and start it up again later for the next lab.)



Lab 4.4: Configure the system "pxe" in Foreman

Configure the system "pxe" in Foreman and start installation

Open Foreman's host dialog using "Host > New Host"

On the Host tab name it "pxe" and select the Environment "production", Puppet CA "foreman.localdomain" and Puppet Master "foreman.localdomain"

On the Interface tab click edit to configure the interface with the MAC address of the system created before, identifier "eth0", select Domain "localdomain" and Subnet "foreman" and keep the suggested IP address

On the Interface tab click edit to configure the interface with the MAC address of the system created before (which could be found on the Details view on the NIC or if paused in the console), identifier "eth0", select Domain "localdomain" and Subnet "foreman" and keep the suggested IP address which will be the first one free in the range of the subnet.

On the Operating system tab select the Architecture "x86_64", Operating System, Media, Partition table depending on your choice earlier and set a Root password of your choice

Use "foreman1" if you have no preference for the root password.

Unpause the virtual machine

When you unpause the virtual machine it will get the suggested IP address and the boot media and will start the unattended installation.



Lab 4.5: Prepare Compute resource libvirt

Prepare Compute resource libvirt to install a virtual machine

Install the Compute resource using the Foreman installer

Execute the Foreman Installer with the following parameter to enable the Compute resource libvirt:

```
# foreman-installer --enable-foreman-compute-libvirt
```

This will install the required packages and restart Foreman.

Create a passphraseless ssh-key for user foreman using ssh-keygen

Switch to the user foreman and create an passphraseless ssh-key using ssh-keygen:

```
# su - foreman -s /bin/bash
$ ssh-keygen
[ENTER]
[ENTER]
```

Copy the public key to the root account of the host using ssh-copy-id

If you know the password of the account in the remote system ssh-copy-id simplifies key management.

```
$ ssh-copy-id root@host.localdomain
```

Configure the Compute resource in Foreman Web GUI

Navigate to "Infrastructure > Compute resources" and click on "New Compute Resource".

In the following dialog insert:

- Name: "host.localdomain"
- Provider: "Libvirt"
- URL: "qemu+ssh://root@host.localdomain/system"
- Display type: "VNC"
- Console password: unchecked

Validate your settings with "Test Connection" before clicking on "Submit".



Lab 4.6: Create the virtual machine "compute" from Foreman

Create the virtual machine "compute" from Foreman Web GUI and start unattended installation

Open Foreman's host dialog using "Host > New Host"

On the Host tab name it "compute" and select to deploy on the Compute resource, the Environment, Puppet CA and Master

For "Deploy on" use "host.localdomain (Libvirt)" instead of "Bare Metal" which will enable all the options of Compute resource libvirt like the virtual network configuration or virtual hardware.

On the Interface tab click edit to configure the interface with identifier "eth0", select Domain and Subnet and keep the suggested IP address, for the Libvirt options choose the virtual network "foreman"

No Mac address is required this time, but do not forget to change the Libvirt options "Network type" to "Virtual (NAT)" and "Network" to "foreman".

On the Operating system tab select the Architecture, Operating System, Media, Partition table and set a Root password.

On the Virtual Machine tab change the Storage type to "QCOW2"

Afterwards press "Submit" and Foreman will show the progress of virtual machine creation. Looking into "Virtual Machine Manager" will show you the virtual machine created with fqdn.



Lab 4.7: Install and configure the Discovery plugin

Install and configure the Discovery plugin

Run the Foreman installer to install the Discovery plugin and download the image

Add the parameters to enable the Discovery plugin and download the image to the Foreman installer

```
# foreman-installer --enable-foreman-plugin-discovery --foreman-plugin-discovery-install-images=true  
# service httpd restart
```

Adjust and deploy the PXE default configuration

Navigate to "Hosts > Provisioning Templates" and search for the "PXELinux global default". Adjust the proxy.url to "<https://foreman.localdomain>" and submit your change. Afterwards click on "Build PXE Default" to deploy it on the TFTP server.

Enable the discovery widget on the dashboard

Navigate to "Monitor > Dashboard" and select from "Manage > Add widgets" the "Discovery widget".



Lab 4.8: Create a virtual machine "discovery"

Create a virtual machine "discovery" for PXE installation

Open "Virtual Machine Manager" application

Open the "Virtual Machine Manager" application from the Gnome Application menu

Select "New virtual machine" from the menu or by pressing the button

Select PXE boot

In the first step of the wizard select the network boot

Select "Linux" and "Red Hat Enterprise Linux 7.2" for CentOS or "Debian jessie" for Debian according to your preferences

In the second step choose "Linux" as operating system type and according to your preferences as version "Red Hat Enterprise Linux 7.2" for CentOS or "Debian jessie" for Debian. For later labs you will also have to choose an operating system, deploy at least one of every kind.

Keep the minimum requirements for RAM, CPU and Disk

In the third step stick with the minimum requirements for RAM and CPU because several systems will be deployed. In the fourth step keep the disk as small as possible but it will not matter as long as it is thin provisioned.

Name your virtual machine "discovery" and select the network "foreman"

Name the virtual machine "discovery" so you know which system was deployed in which lab. Select the network "foreman" instead of "default".

Create the virtual machine and when the PXE menu appears select "(discovery)"

Finish the wizard and let it boot from the default PXE configuration when the PXE menu appears select "(discovery)" instead of "(default)". The discovery image should boot now and report status "SUCCESS" on the console, then a system appears in the Foreman Web GUI in the "Discovery widget" named after the mac address with prefix "mac".



Lab 4.9: Configure the system "discovery" in Foreman

Configure the system "discovery" in Foreman and start installation

Select the newly discovered host from the widget

On the Host tab name it "discovery" and select the Environment "production", Puppet CA "foreman.localdomain" and Puppet Master "foreman.localdomain"

On the Interface tab click edit to configure the interface add the Domain "localdomain"

All other fields are already set to the facts provided by the Discovery image.

On the Operating system tab select the Architecture "x86_64", Operating System, Media, Partition table depending on your choice earlier and set a Root password of your choice

Submit to start the installation



Lab 4.10: Install and configure the Bootdisk plugin

Install and configure the Bootdisk plugin

Run the Foreman installer to install the Bootdisk plugin

Add the parameters to enable the Bootdisk plugin

```
# foreman-installer --enable-foreman-plugin-bootdisk
```

Associate iPXE template for operating systems

Navigate to "Hosts > Provisioning Templates" and search for the iPXE templates. Associate "Kickstart default iPXE" to CentOS and "Preseed default iPXE" to Debian.

Navigate to "Hosts > Operating systems" and select the iPXE templates associated.



Lab 4.11: Reinstall the virtual machine "pxe"

Reinstall the virtual machine "pxe" from a host image

Set the Host "pxe" in "Build" mode and download the host image

Navigate to the Host view of your virtual machine "pxe" and press "Build", accept the dialog telling you it will delete reports and then download the host image.

Configure virtual machine to boot from image by adding a "CDROM" device and selecting it as boot media

Open the Details view of the virtual machine "pxe" in the Virtual Machine Manager. Add a new "CDROM" device, go to the newly added device and connect the downloaded host image. Go to the boot option, select the "CDROM" device and move it onto the first place of the boot order.

Boot and reinstall the virtual machine

When you boot from the image you will recognize it starts with a static network configuration and gets its installer via http from the configured installation media.



Lab 5.1: Proposed Solution

Import of Puppet classes

Place Puppet modules found in "/home/training" on host.localdomain into "/etc/puppet/environments/production" on foreman.localdomain

```
# scp -r host.localdomain:/home/training/puppetmodules.tar.gz /tmp
# cd /etc/puppet/environments/production
# tar xvfz /tmp/puppetmodules.tar.gz
```

Import the Puppet classes in Foreman using "Configure > Classes"

Navigate to "Configure > Classes" in the WebGUI and click on "Import from foreman.localdomain". It will query you to select the changes you want to realize, so select the Environment "production" which should show you classes and press "Update". If you want to do the optional step press "Cancel" instead!

Configure Foreman to ignore the classes from stdlib module by creating "/usr/share/foreman/config/ignored_environments.yml"

Create the file "/usr/share/foreman/config/ignored_environments.yml" with the following content.

```
:filters:
  - !ruby/regexp /^stdlib.*$/
```

This will ignore all classes starting with "stdlib" if you run the import like described above.

If you follow the Puppet Role Profile Pattern something like this could be helpful to ignore all internal classes.

```
:filters:
  - !ruby/regexp /^(?!role|profile).*/
```




Lab 5.2: Proposed Solution

Parameterize and assign Puppet classes to at least one host

Set defaults to the Smart class parameters provided by the imported class

Navigate to "Configure > Classes" and select the class "training::user". In the "Smart Class Parameter" tab insert your name as Default Value for the name for the user, add a ssh public key as Default Value for ssh_pub_key, for the parameter sudo select override and set the default to true.

Hint: To create a ssh key pair run "ssh-key-gen". The key string required for the puppet module is the second part of the pub file.

Assign the Puppet class in the host menu to one host

Select one of your hosts from the "Hosts > All Hosts" view and click "Edit". In the "Puppet Classes" tab select the class "training::user", afterwards you can see and change the parameter values in the "Parameter" tab. Press "Submit" to save your changes.



Lab 5.3: Proposed Solution

Trigger an Puppet agent run and inspect the report

Run the Puppet agent in test mode on the host you assigned the class

Login to the host you assigned the class earlier and execute the following command:

```
# puppet agent -t
```

This will run the agent in test mode (one time in foreground with verbose output) so you will see the changes configured in the puppet class.

Inspect the report of the Puppet agent run

Go back to the WebGUI and navigate to the host and select "Reports". The last report should show some applied changes and if selected it will show you the same information you saw on the console while running the agent. In addition the meta data are visualized.

Other entry points to the reports are the dashboard showing the last reports with any events in "Latest Events" and the Reports overview which filters by default only on the eventful reports.



Lab 6.1: Proposed Solution

LDAP Authentication

Configure the LDAP authentication including group synchronisation

Navigate to "Administer > LDAP Authentication" and press the "New LDAP Source" button. In the dialog add the following values:

- Name: Foreman
- Server: foreman.localdomain
- LDAP: *unchecked*
- Port: 389
- Server Type: POSIX
- Account username: *keep empty*
- Account password: *keep empty*
- Base DN: dc=localdomain
- Groups base DN: ou=groups,dc=localdomain
- LDAP filter: *keep empty*
- Automatically create account in Foreman: *checked*
- Usergroup sync: *checked*
- Login name attribute: uid
- First name attribute: givenName
- Surname attribute: sn
- Email address attribute: mail
- Photo attribute: jpegPhoto

Press "Submit" to store the configuration.

continued...

Add a administrative group to grant the administrative accounts from the LDAP privileges

Navigate to "Administer > User groups" and open the group dialog by pressing "New User group" and insert the values.

- Name: `admins`
- Admin: *checked*
- Roles: *non selected*
- External groups:
 - Name: `admins`
 - Auth source: `LDAP-Foreman`

Create the group by clicking "Submit" and afterwards try to login with the credentials "administrator / netways". Login should work, grant permissions and you should find the user in "Administer > Users". If login fails check the "LDAP Authentication", if no permissions are granted check the "User group".



Lab 6.2: Proposed Solution

Add unprivileged users

Assign the role "Viewer" to the user "viewer"

Login with the account "viewer" and password "netways" and you will have no privileges assigned to you. This will create the account and you can assign the role "Viewer" to it in "Administer > Users" view.

Create a role "Selfservice" to allow creation of new hosts and management of own hosts

Navigate to "Administer > Roles" and press "New role", name it "Selfservice" and save it by clicking "Submit". Afterwards you can add filters to the role.

Resource	Permissions	Unlimited	Search
Host/managed	build_hosts, create_hosts	X	none
Host/managed	edit_hosts, power_hosts, console_hosts, view_hosts, destroy_hosts		owner = current_user
Compute Resource	view_compute_resources, create_compute_resources_vms	X	none
Subnet	view_subnets	X	none
Domain	view_domains	X	none

The additional permissions for Compute Resource allow to create also a virtual machine on it. Subnet permissions are at least required for getting a IP address suggestion.

Assign the new role "Selfservice" to the user "selfservice"

Login with the account "selfservice" and password "netways" and you will have no privileges assigned to you. This will create the account and you can assign the role "Selfservice" to it in "Administer > Users" view.



Lab 7.1: Templates

Import the Community templates

Install the Foreman Plugin Templates using the package "tfm-rubygem-foreman_templates"

```
# yum install tfm-rubygem-foreman_templates -y
```

Run the synchronisation job with "foreman-rake templates:sync"

```
# foreman-rake templates:sync
```



Lab 7.2: Templates

Inspect DHCP reservations

Install the Foreman Plugin DHCP Browser using the package "tfm-rubygem-foreman_dhcp_browser"

```
# yum install tfm-rubygem-foreman_dhcp_browser -y
# service httpd restart
```

Inspect DHCP reservations of the subnet "foreman"

Navigate to "Infrastructure > Subnets" and press the "DHCP" button next the subnet "foreman". In the following DHCP Browser you will find one reservation for every host created earlier.



Lab 7.3: Column View

Add Architecture and Uptime to the "All Hosts" view

Install the Foreman Plugin Column View using the package "tfm-rubygem-foreman_column_view"

```
# yum install tfm-rubygem-foreman_column_view -y
```

Configure it to show the facts architecture and uptime and restart the service

```
# cp /etc/foreman/plugins/foreman_column_view.yaml.example /etc/foreman/plugins/foreman_column_view.yaml  
# service httpd restart
```




Lab 7.4: ABRT

Collect crash dumps on your Foreman server

Install the Foreman Plugin ABRT using the package "tfm-rubygem-foreman_abrt"

```
# yum install tfm-rubygem-foreman_abrt -y
# service httpd restart
```

Set up the Smart proxy by installing the package "rubygem-smart_proxy_abrt" and enabling it

Make the plugin available to the Smart proxy.

```
# yum install rubygem-smart_proxy_abrt -y
# service foreman-proxy restart
```

Let the Foreman know about the new feature so it accepts reports by executing the action "Refresh features" next to the Smart proxy in "Infrastructure > Smart Proxies".

Configure the host to automatically send bug reports to the Smart Proxy

Enable the automatic reporting of crashes.

```
# abrt-auto-reporting enabled
```

Edit the Configuration file "/etc/libreport/plugins/ureport.conf" to include the following lines.

```
URL = https://foreman.localdomain:8443/abrt
SSLVerify = yes
SSLClientAuth = puppet
```

Add the Puppet CA as trusted CA.

```
# cp /var/lib/puppet/ssl/certs/ca.pem /etc/pki/ca-trust/source/anchors/
# update-ca-trust
```

Restart the ABRT services

```
# systemctl restart abrt.service
# systemctl restart abrt-ccpp.service
```

Create a crash dump to test the setup with "will_segfault"

Create a crash dump to test.

```
# yum install will-crash -y
# will_segfault
```

Run the upload manually because the cronjob is scheduled to run every half hour.

```
# sudo -u foreman-proxy smart-proxy-abrt-send
```

You can find the "Problem reports" view as subentry of "Monitor" -> "Problems" in the host details and a widget can also be activated on the dashboard to show a trend.



Lab 7.5: OpenSCAP

Inspect the Security compliance of your system

Install the Foreman Plugin OpenSCAP using the package "tfm-rubygem-foreman_openscap"

```
# yum install tfm-rubygem-foreman_openscap -y
# service httpd restart
```

Install the Smart Proxy Plugin OpenSCAP using the package "rubygem-smart_proxy_openscap"

Make the plugin available to the Smart proxy.

```
# yum install rubygem-smart_proxy_openscap -y
```

Uncomment the following lines in "/etc/foreman-proxy/settings.d/openscap.yml".

```
:openscap_send_log_file: /var/log/foreman-proxy/openscap-send.log
:contentdir: /var/lib/openscap/content
```

Allow the access to the content directory and restart the Service

```
# chown foreman-proxy:foreman-proxy /var/lib/openscap/content
# service foreman-proxy restart
```

Let the Foreman know about the new feature so it accepts reports by executing the action "Refresh features" next to the Smart proxy in "Infrastructure > Smart Proxies".

Make the Puppet Module "foreman_scap_client" available

Install the module on the Puppet master.

```
# puppet module install isimluk-foreman_scap_client
```

Import the Puppet Class from the WebGUI ("Configure > Puppet Environments" or "Configure > Puppet Classes").

Create a Policy for CentOS 7 and assign it to a host

Create a Hostgroup "Scap" via "Configure > Host groups" with only the name set.

Content files are available per default so navigate to "Hosts > Policies" to create a "New Compliance Policy". Name it "Centos-7-Common", choose SCAP Content "Red Hat centos7 default content" and XCCDF Profile "Common Profile for General-Purpose System", schedule it "Weekly" on "Sunday" and assign it to Hostgroup "SCAP".

To view the guide click on the "Show Guide" button next to the policy.

Assign this Hostgroup to one off your CentOS 7 systems.

Execute a Puppet agent run on the host

Login to the host you assigned the Hostgroup with the Policy and execute

```
# puppet agent -t
```

continued...

Create a report on the host and upload to Smart proxy

The Puppet agent prepared a cronjob on your system, get it and execute its content.

```
# crontab -l  
# /usr/bin/foreman_scap_client 1
```

Upload the report from the Smart proxy to Foreman

On the Smart proxy is also a cronjob available which is scheduled every half hour to execute it get the report uploaded immediately.

```
# sudo -u foreman-proxy smart-proxy-openscap-send
```

Now you should find a report in the WebGUI next to the Host in the "All Hosts" view, in the Host details and "Hosts > Reports". Navigate to that report and then press the "View Report" button to inspect it.



Lab 7.6: Cockpit

Inspect your system using Cockpit integrated in Foreman

Install Cockpit and enable the Cockpit websocket

```
# yum install cockpit
# systemctl enable cockpit.socket
# systemctl start cockpit.socket
```

Install the Foreman Plugin Cockpit using the package "tfm-rubygem-foreman_cockpit"

```
# yum install tfm-rubygem-foreman_cockpit -y
# service httpd restart
```



Lab 7.7: Remote Execution

Trigger a Puppet run on a remote system

Install the Foreman Plugin Remote Execution using the Foreman Installer

This will install both the Foreman and Smart proxy Plugin, create a SSH key and restart the services.

```
# foreman-installer --enable-foreman-plugin-remote-execution\  
--enable-foreman-plugin-tasks\  
--enable-foreman-proxy-plugin-remote-execution-ssh
```

Bring out the SSH key

You can use the "ssh-copy-id":

```
# ssh-copy-id -i ~foreman-proxy/.ssh/id_rsa_foreman_proxy root@foreman.localdomain
```

Or get it via the Smart proxy:

```
# curl https://foreman.localdomain:8443/ssh/pubkey >> ~/.ssh/authorized_keys
```

Trigger the Puppet run

Before running any job make sure Foreman knows about the Smart proxy providing the features "Dynflow" and "SSH" by refreshing it.

Navigate to the host and press "Run Job". For multiple hosts the action is also available in the action menu of the "All Hosts" view. To Trigger the Puppet agent run, select the Job Category "Puppet" which has the "Puppet Run Once" as default Job. Press "Submit" to execute it and watch for the execution.



Lab 7.8: Remote Execution - Job template

Create a Job Template "ping" to run the ping command on remote hosts

Create a Job Template to run ping with default values for count and target and input field for target

Navigate to "Hosts > Job templates" and click "New Job Templates". As a template use something like this:

```
ping -c 10 <%= input("ping_target").blank? ? "8.8.8.8" : input("ping_target") %>
```

On the "Job" tab add an input with name "ping_target", input type "User input" and a description.

Run it without input and inspect the output

Select a host to run the job without any input and have a look on the output by clicking the hostname on the "Hosts" tab of the job.

Run it with input and inspect the output

Select a host to run the job, provide a different target as input and have a look on the output by clicking the hostname on the "Hosts" tab of the job.



Lab 8.1: Working with the API

Use the API to query, create and update objects

Query the API for all subnets using the URL endpoint `/api/subnets`

```
# curl -k -u admin:PASSWORD -H "Accept: version=2,application/json" \
https://foreman.localdomain/api/subnets | python -m json.tool
```

Query the API for all Debian hosts using the URL endpoint `/api/hosts` and the search `os=Debian` or `facts.osfamily=Debian`

If you want to see all systems configured in Foreman with operating system "Debian" run the following curl command. This will also include hosts with provisioning pending.

```
# curl -k -u admin:PASSWORD -H "Accept: version=2,application/json" \
https://foreman.localdomain/api/hosts?search=facts.osfamily%3DDebian | python -m json.tool
```

If you want to see all systems reported to run operating system "Debian" :

```
# curl -k -u admin:PASSWORD -H "Accept: version=2,application/json" \
https://foreman.localdomain/api/hosts?search=facts.osfamily%3DDebian | python -m json.tool
```

Create a hostgroup "training" using the API with valid defaults

In the following command the ids depend on your environment and may differ:

```
# curl -k -u admin:PASSWORD -H "Accept: version=2,application/json" -H "Content-Type: application/json" \
-X POST -d '{ "name":"training", "environment_id":"1", "puppet_ca_proxy_id":"1", "puppet_proxy_id":"1",
"subnet_id":"1", "domain_id":"1", "architecture_id":"1", "operatingsystem_id":"1", "medium_id":"1", "ptable_id":"55",
"root_pass":"start123" }' \
https://foreman.localdomain/api/hostgroups | python -m json.tool
```

Change the root password for the hostgroup "training" using the API

You can verify the change in the Audits menu afterwards with:

```
# curl -k -u admin:PASSWORD -H "Accept: version=2,application/json" -H "Content-Type: application/json" \
-X PUT -d '{ "root_pass":"St@rt123" }' \
https://foreman.localdomain/api/hostgroups/training | python -m json.tool
```



Lab 8.2: Working with the CLI

Use the CLI to prepare a new Operatingssystem entry

Create the new Operatingssystem entry for "CentOS 6.8"

```
# hammer os create --name CentOS --major 6 --minor 8 --description "CentOS 6.8" --architecture x86_64 --family "Redhat" \
--password-hash SHA256 --media "CentOS mirror" --partition-tables "Kickstart default"
```

Associate the template "Kickstart default PXELinux" and set as default template

The ids may vary depending on our environment but you can get them with the list subcommands.

```
# hammer template add-operatingsystem --name "Kickstart default PXELinux" --operatingsystem "CentOS 6.8"
# hammer os set-default-template --config-template-id 24 --id 4
```

Associate the template "Kickstart RHEL default" and set as default template

```
# hammer template add-operatingsystem --name "Kickstart RHEL default" --operatingsystem "CentOS 6.8"
# hammer os set-default-template --config-template-id 22 --id 4
```