1. Foreman Discovery 3.0 Manual

This plugin enables Foreman to do automatic bare-metal discovery of unknown nodes on the provisioning network. New nodes self-register into Foreman and upload facts collected by Facter (serial id, network interfaces, memory, disks). The registered nodes show up on Discovered Hosts page and provisioning can be initiated either manually (via UI/CLI or API) or automatically via predefined Discovery Rules.

All communication is through Foreman Proxy which is running a plugin and has direct access both to the provisioning network and Foreman instance:

```
--
Discovered Host > Foreman Proxy > Foreman
--
Foreman > Foreman Proxy > Discovered Host
--
```

1.1 Release Notes

For each Foreman version there is particular version of Foreman Discovery plugin:

Foreman version	Plugin version	Proxy version	Image version	CLI version
<= 1.2	1.0.2	N/A	N/A	N/A
= 1.3	1.1.0	N/A	0.1.1	N/A
= 1.4	1.2.0	N/A	0.3.0	N/A
= 1.5	1.3.0	N/A	0.5.0	N/A
= 1.6	1.4.0	N/A	0.6 or 2.0	N/A
= 1.7	2.0.0	0.1	2.0	0.0.1
= 1.8	3.0.0	1.0.2	2.1	0.0.1

1.1.1 Foreman Discovery plugin

3.0: New features:

- bugfixes
- Foreman 1.8 compatibility

2.0: New features:

- automatic provisioning
- ZIP-based extensions
- indirect communication via foreman proxy plugin
- support for new discovery image 2.0
- UI enhancements
- new CLI hammer plugin
- reboot option

1.1.2 Foreman Proxy Discovery plugin

1.0: Initial version.

1.1.3 Foreman Discovery Image

2.1: New features:

- runtime extensions via http/ZIP
- Foreman Discovery Proxy support

2.0: New features:

- smaller in size (only 150 MB whole RAM disk)
- faster and easier to build (dropped ovirt-node support)
- easier downstream building (in koji)
- based on centos7 and fedora19
- systemd and network manager full support
- better troubleshooting (discovery-debug script)

1.1.4 Hammer CLI Foreman Discovery plugin

0.0.1: Initial version.

2. Installation

There are three different components to install: Foreman Discovery plugin, Foreman Proxy Discovery plugin (aka Smart Proxy Discovery) and Foreman Discovery Image.

2.1 Foreman Discovery plugin

Foreman Discovery plugin is the main component that needs to be installed.

2.1.1 Installer (recommended)

As of Foreman 1.8+, the foreman-installer is able to automatically install the package and proceed with all necessary steps. To do this, re-run the installer with the following option:

```
# foreman-installer --enable-foreman-plugin-discovery
```

Important note: Executing foreman-installer will re-deploy all foreman-related configuration files. In case some changes have been made, proceed with manual download described below.

2.1.2 Manual installation

To install Foreman Discovery plugin, do the following

```
# yum install ruby193-rubygem-foreman_discovery
```

on Fedora and Red Hat systems or

```
# apt-get install ruby-foreman-discovery
```

on Debian and Ubuntu systems.

Restart of Foreman is needed when installing the package separately.

2.2 Foreman Proxy Discovery plugin

Foreman Proxy Discovery plugin must be installed on all proxy hosts on the network where discovery feature is supposed to work. It is possible to skip this step if direct communication is planned, but configuration must be changed in that case.

To install the plugin, do

```
# yum install rubygem-smart_proxy_discovery
```

on Fedora and Red Hat systems or

```
# apt-get install ruby-smart-proxy-discovery
```

on Debian and Ubuntu systems.

Foreman Proxy must be restarted after plugin installation and features must be refreshed for the given Proxy in the Foreman interface (*Infrastructure > Smart Proxies > a proxy > Refresh features*).

2.3 Foreman Discovery Image

The image is based on CentOS 7 and about 150 MB in size. There are two ways to install the image.

2.3.1 Download via installer (recommended)

As of Foreman 1.8+, the foreman-installer is able to automatically download latest stable image. For this, re-run the installer with the following option:

```
# foreman-installer \
    --foreman-plugin-discovery-install-images=true
```

Tip: It is possible to install both Discovery plugin and image in one installer run by providing both the options.

Important note: Executing foreman-installer will re-deploy all foreman-related configuration files. In case some changes have been made, proceed with manual download described below.

2.3.2 Manual download

Images are available for direct download (http://downloads.theforeman.org/discovery/) To download the latest release to the expected location, do the following on Fedora and Red Hat systems:

```
# wget http://downloads.theforeman.org/discovery/releases/latest/fdi-image-latest.tar \
  -O - | tar x --overwrite -C /var/lib/tftpboot/boot
```

On Debian systems, use `/srv/tftp/boot` instead of `/var/tftpboot/boot` .

2.3.3 Verify signatures and checksums

To verify individual ISO/tar files, import our Discovery Plugin key 7E81E7B0 (http://keys.fedoraproject.org:11371/pks/lookup?search=0x7E81E7B0&op=get) and verify individual files:

```
# wget 'http://keys.fedoraproject.org:11371/pks/lookup?search=0x7E81E7B0&op=get' -O- | gpg --impor
t
# gpg --verify --multifile *.asc
gpg: Signature made Fri 06 Nov 2015 09:42:55 AM UTC using RSA key ID 7E81E7B0
gpg: Good signature from "Foreman Discovery <foreman-xxx@googlegroups.com>"
```

To verify extracted init RAM disk and linux kernel do:

```
# cat /var/lib/tftpboot/boot/fdi-image/SHA256SUM
beb3cfba7d9fb9d71481c0c8f... initrd0.img
f03bce150d2473a28697d2988... vmlinuz0
# sha256sum /var/lib/tftpboot/boot/fdi-image/*
```

On Debian systems, use `/srv/tftp/boot` instead of `/var/tftpboot/boot` .

2.3.4 Building own image

To build a discovery image, please visit the foreman-discovery-image git repository (https://github.com/theforeman/foreman-discovery-image) and find the README for further instructions.

2.4 Hammer CLI Foreman Discovery

2.4.1 Installation

Before installing make sure you have installed the hammer_cli_foreman plugin (https://github.com/theforeman/hammer-cli/blob/master/doc/installation.md),

2.4.1.1 rpm (Fedora, CentOS, Red Hat…)

`yum install rubygem-hammer_cli_foreman_discovery`

2.4.1.2 deb (Debian, Ubuntu…)

`apt-get install ruby-hammer-cli-foreman-discovery`

2.4.1.3 gem

`gem install hammer_cli_foreman_discovery` .

2.5 Upgrade

To upgrade Foreman Discovery follow the standard procedure of upgrading all the Foreman packages.

New component Proxy Discovery plugin was introduced for Discovery 2.0 therefore there are two ways how to re-configure existing installation.

2.5.1.1 Upgrade with Proxy Discovery plugin (recommended)

Follow Manual installation > Foreman Proxy Discovery plugin from above and proceed with the next upgrade step (Reboot discovered hosts).

2.5.1.2 Upgrade without Proxy Discovery plugin

Proceed with the next step (Reboot discovered hosts) and in the next section (Configuration) make sure the kernel line in the PXELinux default template has `proxy.type` set to `foreman` and `proxy.url` points to a *Foreman* instance.

```
APPEND ... proxy.url=https://YOUR_FOREMAN proxy.type=foreman
```

2.5.3 Reboot discovered hosts

After the upgrade, if there were any existing discovered hosts, it is required to reboot and delete them. To do this, click on Reboot and Delete buttons for each one of them.

3. Configuration

The following chapter covers configuration of all the components installed.

3.1 Foreman Discovery plugin

Foreman Discovery relies on intercepting the normal boot process for machines not registered in Foreman. To achieve this, the PXE *default.cfg* file needs to be altered to instruct new machines to boot the discovery image.

3.1.1 Default PXE template

In the Foreman UI, go to Provisioning Templates, edit *PXELinux global default* template and add the following after the “LABEL local” block of options:

```
LABEL discovery
MENU LABEL Foreman Discovery
MENU DEFAULT
KERNEL boot/fdi-image/vmlinuz0
APPEND initrd=boot/fdi-image/initrd0.img rootflags=loop root=live:/fdi.iso rootfstype=auto ro rd.l
ive.image acpi=force rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 nomodeset proxy.u
rl=https://FOREMAN_PROXY_INSTANCE:8443 proxy.type=proxy
IPAPPEND 2
```

The *proxy.type* option can be either *proxy* or *foreman*. In the first case all communication goes through Smart Proxy, in the latter case the communication goes directly to Foreman (legacy mode). This is the default when not specified.

The *proxy.url* specifies URL of the Smart Proxy or Foreman depending on the previous setting. Currently only http scheme is supported, https will be added in the future. For backward compatibility, *foreman.url* is an alias for this setting.

Once the APPEND line is modified properly, set the entry to be default via the ONTIMEOUT option:

```
ONTIMEOUT discovery
```

Alternatively, the Discovery image also searches for DNS SRV record named _x-foreman._tcp. If the DNS server is configured for this (see the example for ISC BIND below), then it is not required to provide *foreman.url* at all, but *proxy.type* is still required.

Here is an example of ISC BIND configuration statement to set *foreman* A record to be used with HTTPS protocol.

```
_x-foreman._tcp SRV 0 5 443 foreman
```

You can mix both approaches and override SRV record with the command line.

It is important to keep *IPAPPEND 2* option which is key configuration option to detect interface connected to provisioning network. Also never change or remove *root* option, otherwise image will not boot properly.

It is important to know that DNS servers from DHCP are taken into acount for only for the interface that was specified via the BOOTIF option. This means when a system has multiple NICs, DNS will work for the correct interface - the one that was booted from.

3.1.2 Update PXE default template

Once the template has been updated, click the *Build PXE Default* button at the top of the Provisioning Templates page. This will instruct the TFTP proxy to rewrite the *pxelinux.cfg/default* file.

Repeat this step every time a change is made to the default template.

3.1.3 Global settings

If Locations and/or Organisations are enabled, Foreman will default to using the first Location and first Organisation for Discovered hosts. If hosts should be placed in some other Location/Organization, alter the default settings in *Administer > Settings > Discovered*.

There is also setting called *discovery_fact* which defaults to *discovery_bootif*. It specify which incoming fact should be used to get the MAC address. By default, the PXELinux BOOTIF kernel command line option is used which gives the MAC address of the interface which was booted from.

Note that *auto_discovery* option is set to *false* by default. If you want to trigger provisioning automatically with rules, you need to turn this setting on. It's recommended to try provision manually first to test before proceeding.

It is possible to add any fact reported by Facter via *discovery_fact_column* global option onto the Discovered Hosts page table as a new column. To do that, set the value of this setting to name of a fact reported. To hide the new column, set to a blank value.

3.2 Foreman Proxy Discovery plugin

Make sure *foreman_url* setting is present in the Foreman Proxy configuration file.

```
# grep foreman_url /etc/foreman-proxy/settings.yml
:foreman_url: https://FOREMAN_HOST
```

This should by done automatically by our installer. It is a good idea to check if the host responds properly and there are no firewalls blocking the communication.

3.2.1 Subnet proxy setup

All subnets with discovered nodes need this specified in Foreman so it connects via the Foreman Proxy. To do this, go to *Infrastructure > Smart Proxies* and verify if the desired proxy lists *Discovery* feature. If not, click on *Refresh features* button and it will appear immediately.

In *Infrastructure > Subnets* select the desired *Discovery Proxy* for each appropriate Subnet.

Make sure that lease pool (defined in ISC DHCP configuration) and reservation pool (defined in *Infrastructure > Subnets*) are disjoint. More on that topic in the installation scenarios (manuals/latest/index.html#3.2.3InstallationScenarios) section.

3.3 Hammer CLI Foreman Discovery plugin

The plugin must be enabled in `cli.modules.d/foreman_discovery.yml` (see Hammer config directories (https://github.com/theforeman/hammer-cli/blob/master/doc/installation.md#locations)) as follows:

```
:foreman_discovery:
    :enable_module: true
```

3.4 Permissions

The plugin will create a Role called Discovery when first started. This can be assigned to roles for non-admins to allow them to use the discovery plugin. Alternatively assign the `perform_discovery` permission to an existing Role.

4. Usage

Foreman Discovery plugin provides user interface, API and CLI.

4.1 Hardware discovery

Boot a machine in any provisioning network that was configured with the default PXE configuration above. It should register with Foreman and appear in *Hosts > Discovered Hosts*.

4.2 Manual provisioning

Select a discovered host and choose Provision. This will redirect to the normal Edit page for a Host, with the discovered data filled in where possible. Fill in the details as normal.

On save, Foreman modifies the host's PXELinux file on the TFTP server and reboots the discovered host, after which it boots into an installer for the chosen OS, and finally into the installed OS.

Delete a machine and reboot it to have it move back to the Discovery Pool.

4.3 Automatic provisioning

Starting with version 2.0, it is possible to predefine provisioning rules which will assign host group to provisioned hosts and trigger provisioning automatically. To do that, head over to *Configure > Discovery rules* and create such a rule:

- **Name** represents rule name shown to the users. It must not contain spaces or non-alphanumeric characters.
- **Search** statement is used to match discovered hosts for the particular rule. Use Scoped Search syntax to define it. Examples are shown below.
- **Host Group** is assigned to a matching host before starting provisioning process. It is very important the selected Host Group has all the required parameters set (domain, subnet, root password), otherwise provisioning process will fail.
- **Hostname** defines a pattern to assign human-readable hostnames to the matching hosts. When left empty, hostname is assigned in a form of `macMACADDRESS` by default. The same syntax as for provisioning templates is used. See below for more information and examples.
- **Hosts limit** enables to limit maximum amount of provisioned hosts per rule. If a limit was reached, the rule will not take effect until one or more hosts are deleted. Typical use case are rules per server rack or row when it is necessary to change provisioning parameters like hostname or host group per each entry.
- **Priority** puts the rules in order. Must be greater than zero and low numbers go first. Rules are always matched by priority given.
- **Enabled** flag is used for temporary shutdown of rules.
- **Organizations\Locations** discovered hosts can only auto provision from rules in the same organization/location as the discovered host.

Once some rules are defined, the good practice is to discover a host and apply the rules using Auto discover button on the host.

By default, Foreman does not trigger auto discovery automatically. This must be explicitly turned on in *Administer > Settings > Discovered > discovery_auto*.

4.3.1 Search syntax

Easiest way of testing search patterns is in Discovered hosts list, because the search box gives the same results. Typical search fields are facts, they all start with "facts.". Auto completion can be used to browse the facts as well as discovered hosts detail screen. Typical search queries:

```
facts.architecture = x86_64
facts.bios_vendor ~ 'Dell*'
facts.macaddress = "aa:bb:cc:dd:ee:ff"
facts.macaddress_eth0 = "aa:bb:cc:dd:ee:ff"
facts.ipaddress_eth1 = ~ "192.168."
```

As of Foreman 1.7, all the facts are currently simple strings, so it is not possible to do numeric comparisons.

Some important facts are extracted and converted to numbers. These are:

- cpu_count - number of CPUs
- disk_count - number of disks attached
- disks_size - total amount of disk space (in MiB)

Possible queries are:

```
cpu_count >= 8
disk_count < 10
disks_size > 1000000
```

See the searching section (manuals/latest/index.html#4.1.5Searching) for more information.

4.3.2 Hostname patterns

The target hostname template pattern has the same syntax as in Provisioning Templates (ERB). Domain will be appended automatically. A hostname based on MAC address will be used when left blank. In addition to @host attribute function rand for random integers is available. Examples:

```
application-server-<%= rand(99999) %>
load-balancer-<%= @host.facts['bios_vendor'] + '-' + rand(99999) %>
wwwsrv-<%= @host.hostgroup.name %>
minion-<%= @host.discovery_rule.name %>
db-server-<%= @host.ip.gsub('.','-') + '-' + @host.hostgroup.subnet.name %>
```

When creating hostname patterns, make sure the resulting host names are **unique**. This is very important. Hostnames must not start with numbers. A good approach is to use unique information provided by facter (MAC address, BIOS or serial ID) or to randomize the hostname somehow.

4.4 Hammer CLI Foreman Discovery plugin

Confirm your setup by running `hammer -h` and check that the discovery command is listed.

```
$ hammer -h
...
Subcommands:
 ...
 discovery                     Discovery related actions.
 ...
...
```

The actions you can use with discovery will appear as follows:

```
$ hammer discovery -h
Usage:
    hammer discovery [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:
 SUBCOMMAND                    subcommand
 [ARG] ...                     subcommand arguments

Subcommands:
 auto-provision                Auto provision a host
 delete                        Delete a discovered host
 facts                         Show a discovered host
 info                          Show a discovered host
 list                          List all discovered hosts
 provision                     Provision a discovered host
 reboot                        Reboot a host
 refresh-facts                 Refresh the facts of a host

Options:
 -h, --help                    print help
```

For example to reboot a discovered_host:

```
$ hammer discovery reboot -h
Usage:
    hammer discovery reboot [OPTIONS]

Options:
 --id ID
 --name NAME                   Name to search by
 -h, --help                    print help

$ hammer discovery reboot --id 130
Host reboot started
```

5. Extending the image

It is possible to extend the Foreman Discovery Image with custom facts, software or device drivers easily. There are two ways of doing that.

5.1 Runtime extensions

It's possible to provide a zip file containing extra code for the image to use. First, create a directory structure like:

```
.
├── autostart.d
│   └── 01_zip.sh
├── bin
│   └── ntpdate
├── facts
│   └── test.rb
└── lib
    ├── libcrypto.so.1.0.0
    └── ruby
        └── test.rb
```

`autostart.d` contains scripts that will be executed in POSIX order by the image as it starts up, before the host is registered to Foreman. `bin` is added to PATH, you can place binaries here and use them in the autostart scripts. `facts` is added to FACTERLIB so that custom facts may be configured and sent to Foreman. `lib` is added to LD_LIBRARY_PATH and `lib/ruby` is added to RUBYLIB, so that binaries in `bin` can be executed properly.

Environment variables (PATH, LD_LIBRARY_PATH, RUBYLIB and FACTERLIB) are appended to. If you need to specify the path to something explicitly in your scripts, the zip contents are extracted to `/opt/extension` on the image.

An example structure is provided in `example_zip` in this repo. You can zip up your structure with `zip -r my_extension.zip .`

You can create multiple zip files, but be aware they will be extracted to the same place on the discovery image, so files in later zips will overwrite earlier ones if they have the same filename.

To inform the image of the extensions it should use, place your zip(s) on your TFTP server along with the discovery image, and then update your PXELinux APPEND line with `fdi.zips=<path-to-zip>,<path-to-zip>`, where the paths are relative to the TFTP root. So if you have two zips at $TFTP/zip1.zip and $TFTP/boot/zip2.zip, you would use `fdi.zips=zip1.zip,boot/zip2.zip`.

5.2 Image building with extensions

The very same ZIP file with the structure described above can be injected into your own image. This is useful for network drivers when there is a chicken and egg problem.

To build a discovery image with extensions, please visit the foreman-discovery-image git repository (https://github.com/theforeman/foreman-discovery-image) and find the README for further instructions.

6 Help

Please follow our standard procedures and contacts (support.html). For problems with the image, send us the output of the following command for a running discovered node (see below how to get access to the shell):

```
# discovery-debug
```

6.1 Troubleshooting

If you find a bug, please file it in Redmine (http://projects.theforeman.org/projects/discovery/issues/new).

See the troubleshooting section (manuals/latest/index.html#7.2GettingHelp) in the Foreman manual for more info.

6.1.1 Troubleshooting plugins

If the booted machine fails to register with Foreman, then there are a number of common causes:

- Machine did not boot the correct image
  - Check your default PXE Linux template and try to re-deploy it
  - Check your pxelinux.cfg/default configuration file on the TFTP Proxy
- Machine booted image but failed to contact Foreman
  - Check the proxy.url and proxy.type options
  - Check DNS is working for that image, or use an IP in proxy.url
  - Check DHCP is handing IPs to the booted image correctly

6.1.2 Troubleshooting the image

First of all make sure your server (or VM) has more than 500 MB of memory because less memory can lead to various random kernel panic errors as the image needs to be extracted in-place (150 MB * 2).

The first virtual console is reserved for logs, all systemd logging is shown there. Particularly useful system logs are tagged with:

- discover-host - initial facts upload
- foreman-discovery - facts refresh, reboot remote commands
- nm-prepare - boot script which pre-configures NetworkManager
- NetworkManager - networking information

The root account and ssh access are disabled by default, but you can enable ssh and set root password using the following kernel command line options:

```
fdi.ssh=1 fdi.rootpw=redhat
```

Use tty2 console (or higher) to login onto a discovered host.

6.2 Contributing

Follow the same process as Foreman (contribute.html#SubmitPatches) for contributing.