ДОКУМЕНТАЦИЯ ALT LINUX TEAM

Foreman

Руководство пользователя

Редакция июль, 2023

Аннотация

Foreman — это средство автоматизации повторяющихся задач, развёртывания приложений, управления циклом жизни ваших серверов по отдельности или в облаке.

Данное руководство соответствует текущему состоянию сведений, но какие-либо окончательные правки могли не попасть в него. В случае обнаружения ошибок и неточностей в руководство вносятся изменения.

Глава 1. Установка связки Foreman-Puppet

В данном разделе описана настройка сервера.

Для создания связки Foreman-Puppet следует установить следующие подсистемы:

1. PostgreSQL
2. Puppet
3. Foreman
4. Smart Proxy

Настойка после обновления должна производится по этому же порядку.

1.1. Подготовка системы к установке сервера Foreman

В этом разделе перечислены требования для установки сервера FreeIPA. Перед установкой необходимо убедиться, что система соответствует этим требованиям.

1.1.1. Описание стенда

Описание стенда:

- Сервер Foreman — 192.168.0.148 foreman.test.alt
- Клиент — 192.168.0.181 client1.test.alt
- Клиент — 192.168.0.140 client2.test.alt

Для корректной работы сервера, необходимо, задать ему полное доменное имя (FQDN). Имя серверу можно назначить командой:

```
# hostnamectl set-hostname foreman.test.alt
```

Имена сервера и клиентов должны разрешаться через DNS-сервер или **`/etc/hosts`**.

Если в сети нет DNS-сервера на каждом хосте в файле **`/etc/hosts`** необходимо указать имя сервера и клиента:

```
# echo "192.168.0.148 foreman.test.alt foreman" >> /etc/hosts
# echo "192.168.0.181 client1.test.alt client1" >> /etc/hosts
# echo "192.168.0.140 client2.test.alt client2" >> /etc/hosts
```

1.1.2. Системные требования к серверу Foreman

Системные требования для Foreman зависят главным образом от количества запросов, которые он будет получать (количество клиентов управления конфигурацией, активности веб-интерфейса и других систем, использующих API).

На сервере должно быть установлено не менее 4 ГБ оперативной памяти и 2 ГБ дискового пространства.

1.1.3. Требования к портам

Таблица 1.1. Порты

Служба	Порт	Протокол
Для доступа к панели управления Foreman	2345	TCP
Для связи с Smart Proxy	8000	TCP
Для работы Puppet	8140	TCP

1.2. Установка PostgreSQL

Перед установкой Foreman необходимо установить и инициализировать PostgreSQL версии 11 и выше.

Важно

Если база данных перед установкой не была установлена и инициализирована, то Foreman установится некорректно. Для корректной установки придется удалить пакет, инициализировать базу данных и установить заново!

Установить PostgreSQL:

```
# apt-get install postgresql14-server
```

Версия PostgreSQL может быть другой.

Подготовить к запуску и настроить службы PostgreSQL:

- создать системные базы данных:

```
# /etc/init.d/postgresql initdb
```

- включить по умолчанию и запустить службу:

```
# systemctl enable --now postgresql.service
```

1.3. Установка Puppet Server

Установить пакеты *puppet*, *puppetserver* и *java-17-openjdk*:

```
# apt-get install puppet puppetserver java-17-openjdk
```

Версия java может быть другой.

Конфигурационные файлы Puppet находятся в каталоге **/etc/puppet/**.

Внести изменения в конфигурационный файл **/etc/puppet/puppet.conf**:

```
[main]
certname = foreman.test.alt
server = foreman.test.alt
[master]
certname = foreman.test.alt
environment = production
```

- в секции **main** находятся глобальные настройки сервиса;
- в секции **master** находятся настройки сервера Puppet;
- параметр **server** указывает клиенту, с каким сервером работать;

- параметр **certname** задает имя, с которым клиент будет обращаться к серверу.

Примечание

Следующие параметры в файле **/etc/puppet/puppet.conf** должны быть закомментированы:

```
[master]
...
#  external_nodes = /usr/lib/puppet-modules/puppetserver-foreman/files/
enc.rb
#  node_terminus = exec
```

Для возможности создания файлов на агентах нужно отредактировать файл **/etc/puppet/fileserver.conf**:

```
[files]
path /etc/puppet/code/modules/files
allow *
```

Запустить и добавить в автозагрузку сервисы:

```
# systemctl enable --now puppet
# systemctl enable --now puppetserver
```

Проверка работы агента:

```
# puppet agent -t
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for foreman.test.alt
Info: Applying configuration version '1696844601'
Info: Creating state file /var/cache/puppet/state/state.yaml
```

1.3.1. Работа с сертификатами

По умолчанию при первом запуске puppetserver выполняет функцию CA сервера, и генерирует сертификаты в каталоге **/etc/puppet/ssl**. В созданной цепочке сертификатов, сертификат CA (**ca.pem**) является самоподписанным.

Можно создать цепочку с корневым сертификатом root_key.pem, которым и подпишется сертификат CA, если перед первым запуском puppetserver выполнить команду:

```
# puppetserver ca setup --certname foreman.test.alt --subject-alt-names
foreman.test.alt
```

где в **--subject-alt-names** можно через запятую перечислить альтернативные имена хоста.

Если в сети используется несколько puppetserver, целесообразно создать подписанный сертификат для каждого из этих серверов и затем распространить сертификаты на соответствующие сервера. Для этого следует на остановленном puppetserver ca запустить команду:

```
# puppetserver ca generate --certname <имя_сервера> --subject-alt-names
<имя_хоста>,<альтернативное_имя_хоста> --ca-client
```

Важно

Включить/отключить функцию puppetserver ca можно в файле **/etc/puppetserver/services.d/ca.cfg**.

1.4. Foreman

1.4.1. Установка

Установка Foreman:

```
# apt-get install foreman puppet-theforeman-foreman puppet-theforeman-
puppetserver-foreman
```

После обновления базы пакетов необходимо перезапустить службы puppetserver и puppet:

```
# systemctl restart puppetserver
# systemctl restart puppet
```

Подготовить экземпляр Foreman, выполнив команду:

```
# railsctl setup foreman || echo "Ошибка подготовки"
Work dir is '/var/lib/foreman'
Processing rails application 'foreman' setup...
Applying variables from '/etc/sysconfig/foreman'
Renewing dependencies...
Initializing secret...
Initializing encryption key...
Initializing database...
Checking migration and seeds...
Generating API cache...
Cleaning...
Login credentials: admin / 9T7Z8SzYXMtghJFr
```

В строке *Login credentials* будут указаны логин и автоматически созданный пароль администратора.

Примечание

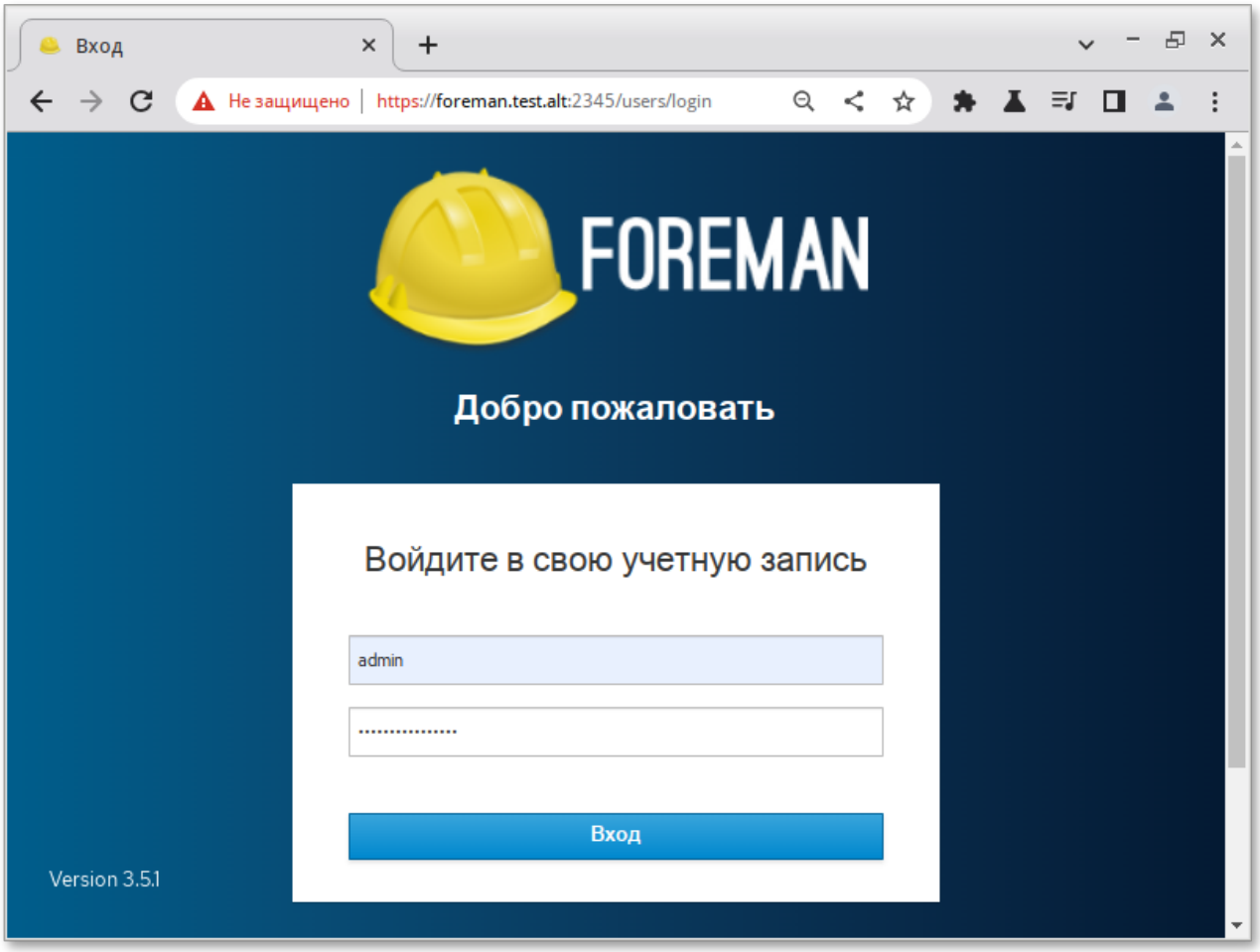
Пароль и логин администратора создаются автоматически и возвращаются скриптом настойки. Получить логин и пароль можно, выполнив следующую команду:

```
# grep "Login credentials" /var/log/foreman/ -r
/var/log/foreman/db_setup.log:Login credentials: admin / 9T7Z8SzYXMtghJFr
```

Запустить и добавить в автозагрузку службу foreman:

```
# systemctl enable --now foreman
```

Foreman будет доступен по адресу **`https://localhost:2345`**:



Примечание

Потребуется подтвердить согласие на подключение.

1.4.2. Настройка параметров

Добавить в файл **`/etc/foreman/settings.yml`** параметры:

```
:trusted_hosts: [foreman.test.alt,foreman,192.168.0.0/24]
:host_details_ui: false
```

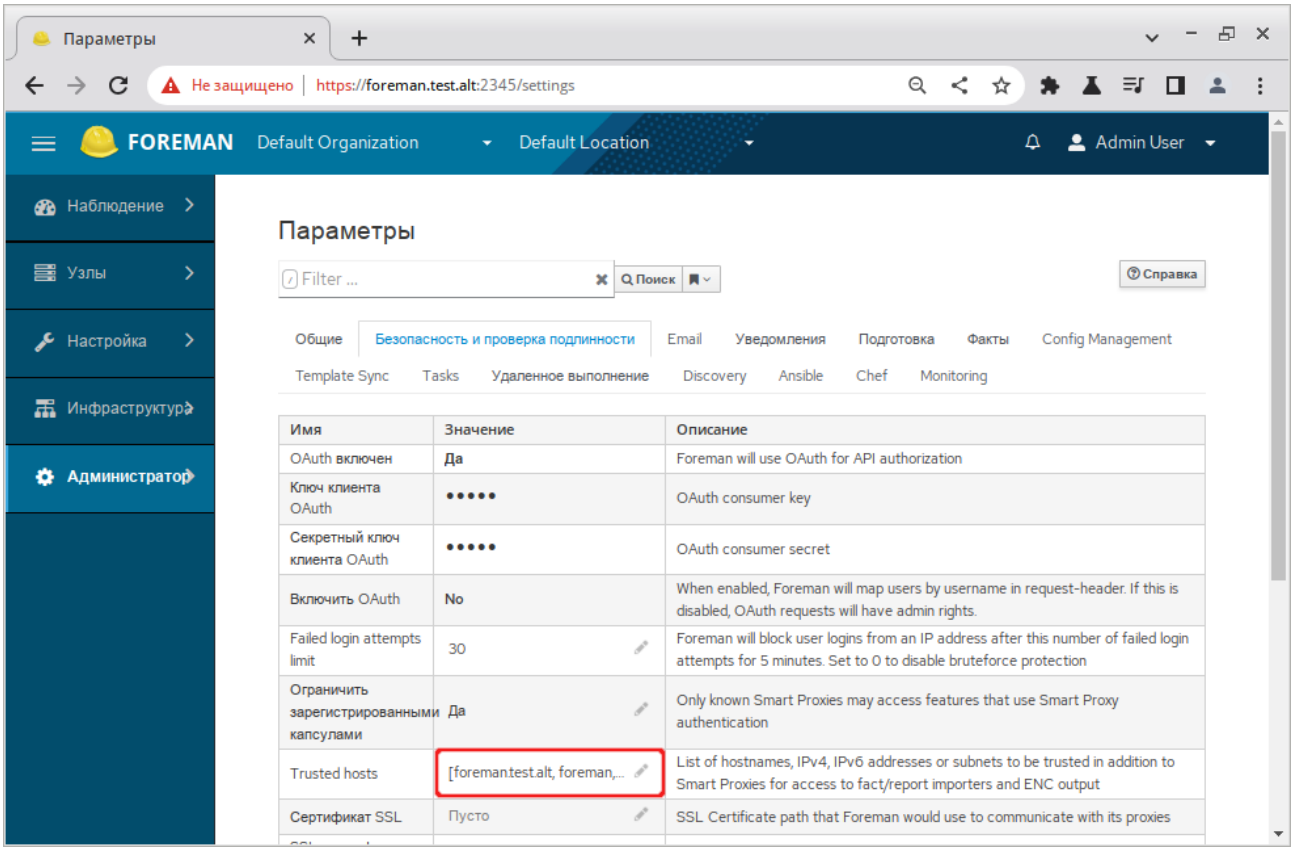
В параметре **`trusted_hosts`** необходимо указать список имён хостов, IP-адресов или подсетей, с которых будет приниматься информация о хостах.

После внесения изменений в конфигурационный файл необходимо перезапустить службу foreman:

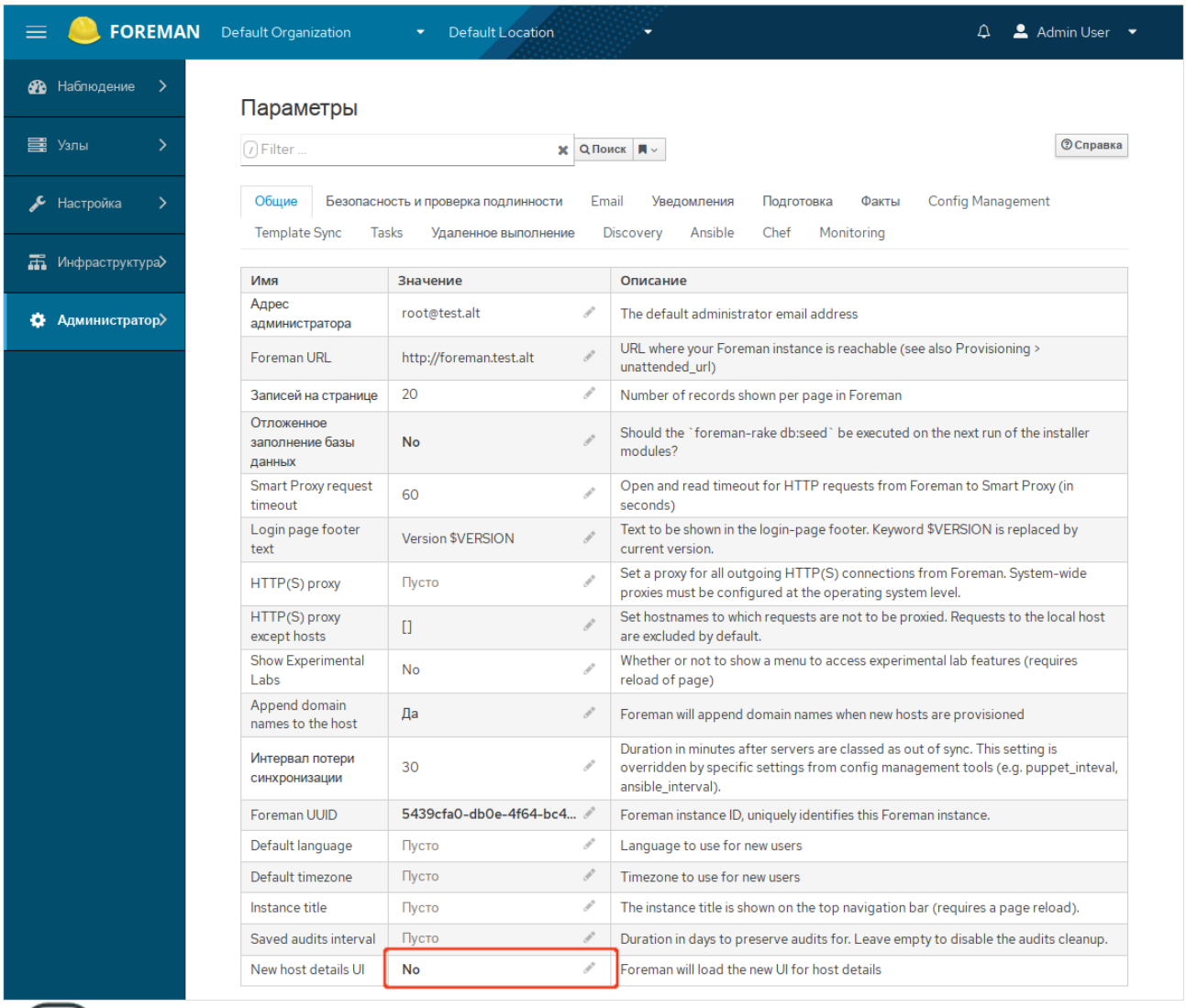
```
# systemctl restart foreman
```

Задать значение параметров также можно в веб-интерфейсе Foreman. Для этого необходимо перейти в **Администратор** → **Параметры**.

Для настройки параметра **`trusted_hosts`** выбрать вкладку **`Безопасность и проверка подлинности`**, строка **`Trusted hosts`**:



Для настройки параметра **`host_details_ui`** следует выбрать вкладку **`Общие`**, строка **`New host details UI`**.



Примечание

Настройки, определённые в конфигурационном файле, нельзя изменить в веб-интерфейсе.

1.5. Установка Smart Proxy

Smart proxy — средство выполняющее, вспомогательную для архитектуры Foreman, функцию в организации процесса ввода в работу нового хоста. Для уменьшения задержек в эксплуатации, он может быть размешен на ПК вместе со службой Foreman или на близстоящем ПК. Smart proxy также часто называется как Foreman proxy.

Примечание

Перед установкой Smart proxy необходимо убедиться, что Foreman установлен и инициализирован.

Установка пакета предоставляющего функционал прокси:

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

Пред запуском сервиса необходимо изменить содержимое следующих файлов:

- файл **/etc/smart-proxy/config/settings.d/puppetca_http_api.yml**:

```
---
# URL of the puppet master itself for API requests.
:puppet_url: https://foreman.test.alt:8140
#
# SSL certificates used to access the CA API.
:puppet_ssl_ca: /etc/puppet/ssl/certs/ca.pem
:puppet_ssl_cert: /etc/puppet/ssl/certs/foreman.test.alt.pem
:puppet_ssl_key: /etc/puppet/ssl/private_keys/foreman.test.alt.pem
```

- файл **/etc/smart-proxy/config/settings.d/puppet_proxy_puppet_api.yml**:

```
---
# URL of the puppet master itself for API requests.
:puppet_url: https://foreman.test.alt:8140
#
# SSL certificates used to access the puppet API
:puppet_ssl_ca: /etc/puppet/ssl/certs/ca.pem
:puppet_ssl_cert: /etc/puppet/ssl/certs/foreman.test.alt.pem
:puppet_ssl_key: /etc/puppet/ssl/private_keys/foreman.test.alt.pem
#
# Smart Proxy api timeout when Puppet's environment classes api is used and
classes cache is disabled
:api_timeout: 30
```

- файл **/etc/smart-proxy/config/settings.d/puppetca.yml**:

```
---
#
# PuppetCA management
# Can be true, false, or http/https to enable just one of the protocols
:enabled: http

# valid providers:
#   - puppetca_hostname_whitelisting (verify CSRs based on a hostname
whitelist)
#   - puppetca_token_whitelisting (verify CSRs based on a token whitelist)
:use_provider: puppetca_hostname_whitelisting
# Puppet version used
:puppet_version: 7.20.0
```

- файл **/etc/smart-proxy/config/settings.d/puppet.yml**:

```
---
# Can be true, false, or http/https to enable just one of the protocols
:enabled: http
# valid providers:
#   puppet_proxy_mcollective (uses mco puppet)
#   puppet_proxy_ssh         (run puppet over ssh)
#   puppet_proxy_salt        (uses salt puppet.run)
```

```
#   puppet_proxy_customrun   (calls a custom command with args)
#:use_provider: puppet_proxy_customrun
# Puppet version used
:puppet_version: 7.20.0
```

- файл **/etc/smart-proxy/config/settings.d/puppetca_hostname_whitelisting.yml**:

```
---
#
# Configuration of the PuppetCA hostname_whitelisting provider
#
:autosignfile: /etc/puppet/autosign.conf
```

- файл **/etc/smart-proxy/config/settings.d/facts.yml**:

```
---
# Can be true, false, or http/https to enable just one of the protocols
:enabled: true
```

Проверить наличие файла **/etc/puppet/autosign.conf** и установленные на него разрешения (rw-rw-r--):

```
# ls -l /etc/puppet/autosign.conf
```

Если файла нет, создать и установить на него разрешения:

```
# touch /etc/puppet/autosign.conf
# chmod 664 /etc/puppet/autosign.conf
```

Для связи Smart proxy с Foreman в файле **/etc/smart-proxy/config/settings.yml** указать следующие параметры:

```
:trusted_hosts: [foreman.test.alt,localhost]
:foreman_url: https://foreman.test.alt:2345
:foreman_ssl_ca: /etc/foreman/rootCA.pem
:foreman_ssl_cert: /etc/foreman/ssl_cert.pem
:foreman_ssl_key: /etc/foreman/ssl_key.pem
```

Примечание

Если не создавалась цепочка с корневым сертификатом root_key.pem, параметры **foreman_ssl…** в файле **/etc/smart-proxy/config/settings.yml** могут быть такие:

```
:foreman_ssl_ca: /etc/puppet/ssl/certs/ca.pem
:foreman_ssl_cert: /etc/foreman/ssl_cert.pem
:foreman_ssl_key: /etc/foreman/ssl_key.pem
```

Запуск и добавление в автозагрузку службы smart-proxy:

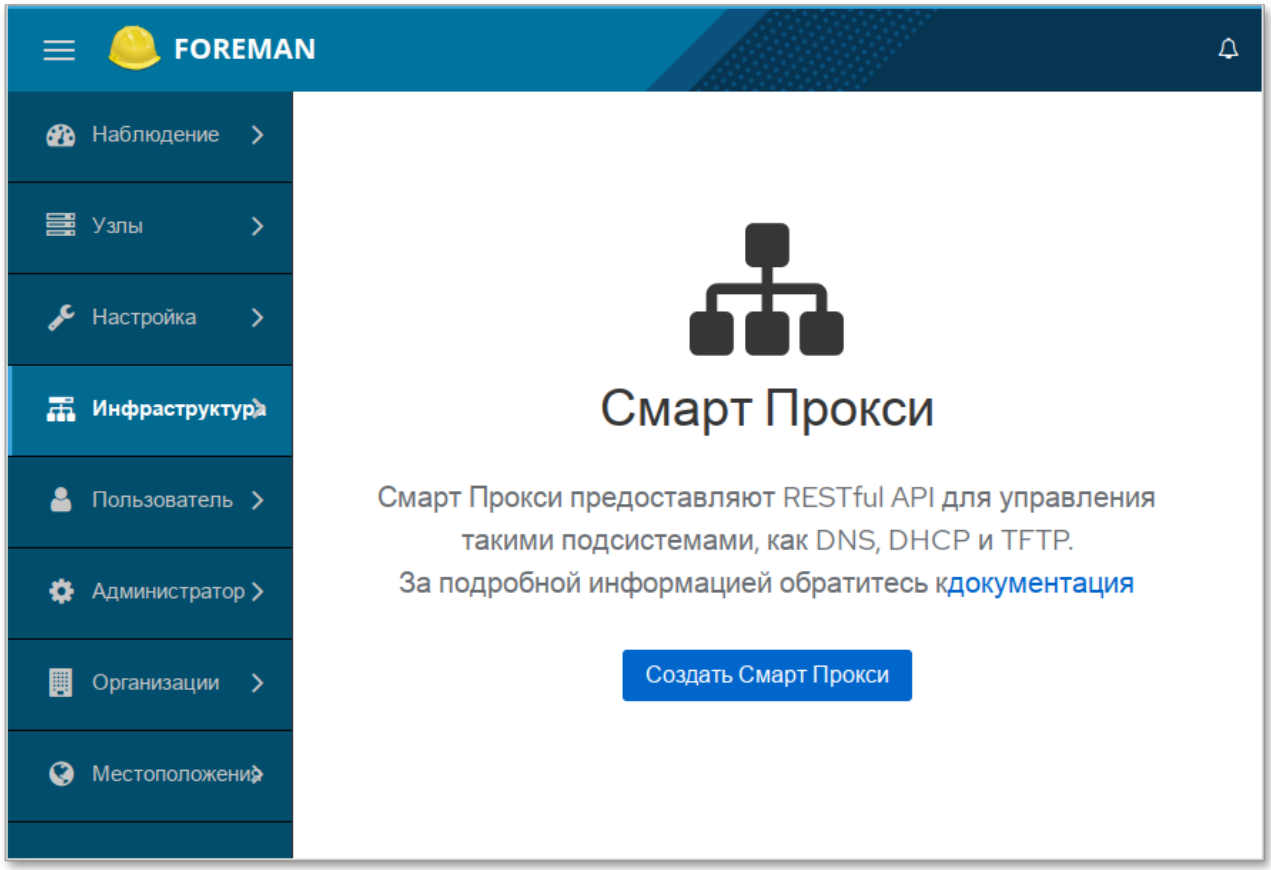
```
# systemctl enable --now smart-proxy
```

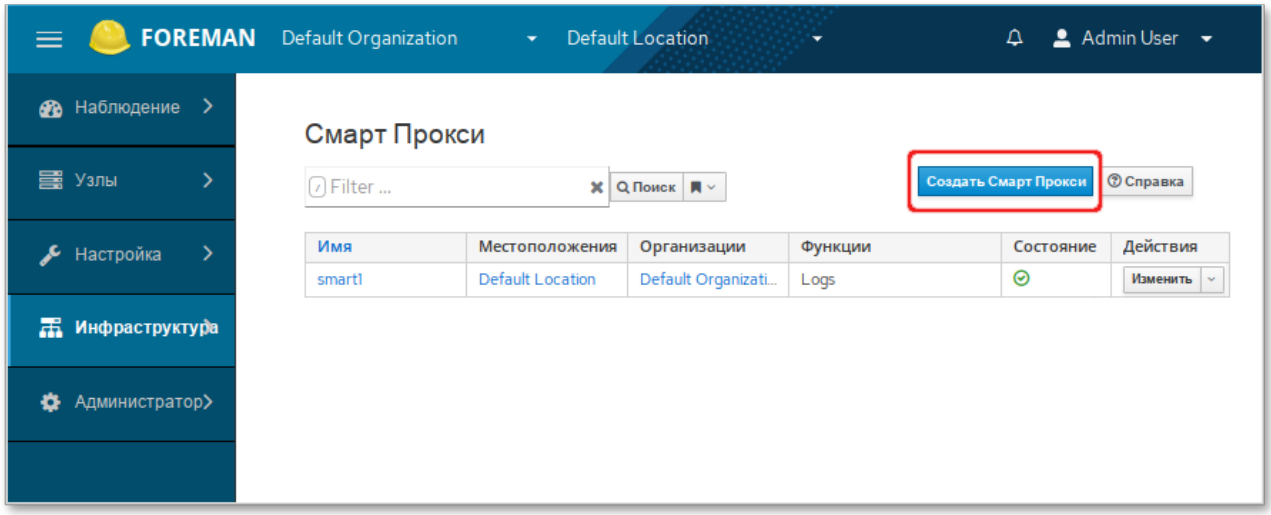
Smart proxy будет доступен для запросов Foreman по адресу **`http://localhost:8000`**. Проверка (должен быть установлен пакет *telnet*):

```
# telnet localhost 8000
```

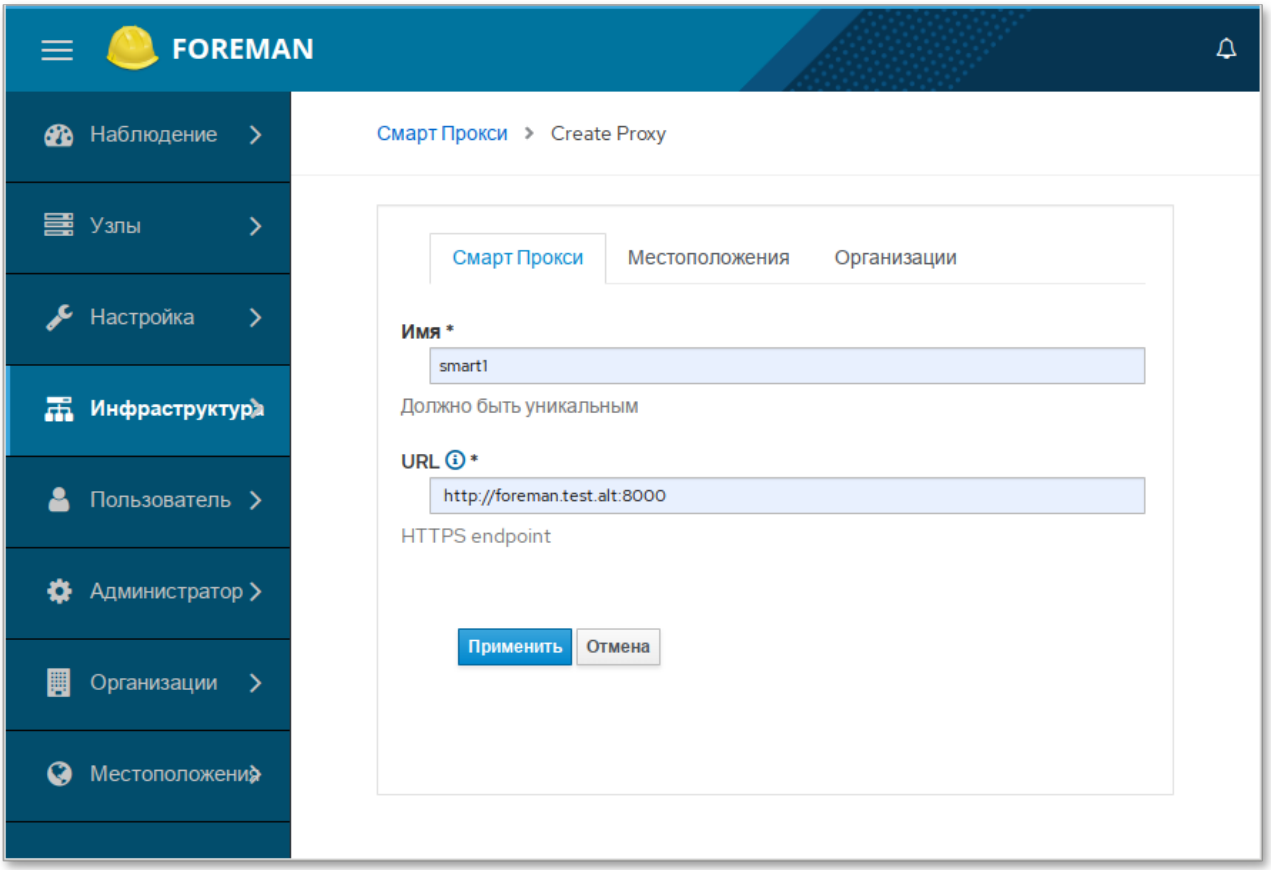
1.5.1. Настройка подключения Foreman и Smart Proxy

Настройка соединения Foreman и Smart Proxy производится в веб-интерфейсе Foreman. Следует перейти в **Инфраструктура** → **Капсулы** и нажать кнопку **`Создать Смарт Прокси`** (или перейти по прямой ссылке **`https://foreman.test.alt:2345/smart_proxies/new`**):





В открывшемся окне в поле **Имя** необходимо указать имя Smart Proxy, в поле **URL** указать адрес http://foreman.test.alt:8000 и нажать кнопку **`Применить`**:



Запустить (через перезапуск) службу smart-proxy:

```
# systemctl restart smart-proxy
```

1.6. Настройка модуля ENC для взаимодействия с Puppetserver

Создать файл **`/etc/puppet/foreman.yaml`**:

```
---
# Update for your Foreman and Puppet master hostname(s)
:url: "https://foreman.test.alt:2345"
:ssl_ca: "/etc/puppet/ssl/certs/ca.pem"
:ssl_cert: "/etc/puppet/ssl/certs/foreman.test.alt.pem"
:ssl_key: "/etc/puppet/ssl/private_keys/foreman.test.alt.pem"
# Advanced settings
#:puppetdir: "/opt/puppetlabs/server/data/puppetserver"
:puppetdir: "/var/lib/puppetserver"
:puppetuser: "puppet"
:facts: true
:timeout: 10
:threads: null
```

После того как Foreman установлен и к нему подключен(-ы) Smart_Proxy, необходимо внести изменения в файл **`/etc/puppet/puppet.conf`**.

В файл **`/etc/puppet/puppet.conf`** необходимо добавить/раскомментировать строки:

```
[master]
   ...
   external_nodes = /usr/lib/puppet-modules/theforeman-puppetserver-foreman/
files/enc.rb
   node_terminus = exec
```

После внесения изменений в конфигурационный файл Puppet необходимо перезапустить службы puppetserver и puppet:

```
# systemctl restart puppetserver
# systemctl restart puppet
```

В результате запуска сервиса puppet, текущий сервер будет добавлен в список управляемых.

Примечание

Для добавления серверной ноды в список управляемых узлов можно выполнить команду:

```
# /usr/lib/puppet-modules/theforeman-puppetserver-foreman/files/enc.rb
foreman.test.alt
```

Примечание

Если при запуске ENC-скрипта выводится сообщение об ошибке:

```
foreman.test.alt: During the fact upload the server responded with: 403
Forbidden. Error is ignored and the execution continues.
{
  "error": {"message":"Access denied","details":"Missing one of the
required permissions: upload_facts","missing_permissions":["upload_facts"]}
}
Serving cached ENC: Error retrieving node foreman.test.alt:
Net::HTTPNotFound
Check Foreman's /var/log/foreman/production.log for more information.
Unable to read from Cache file: No such file or directory @ rb_sysopen - /
var/lib/puppetserver/yaml/foreman/foreman.test.alt.yaml
```

Необходимо в файле **/etc/foreman/settings.yml** установить параметр:

```
:restrict_registered_smart_proxies: false
```

И перезапустить Foreman:

```
# systemctl restart foreman
```

Проверка работы агента:

```
# puppet agent -t
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for foreman.test.alt
Info: Applying configuration version '1691592405'
Notice: Applied catalog in 0.02 seconds
```

1.7. Добавление плагинов

1.7.1. Настройка беспарольного доступа по ssh

Генерация SSH-ключа:

```
# ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ''
```

Скопировать публичную часть SSH-ключа на клиент для пользователя user:

```
# ssh-copy-id -i ~/.ssh/id_ed25519.pub user@client1.test.alt
```

Скопировать публичную часть SSH-ключа на клиент для администратора. Для этого подключаемся к клиенту и под root копируем публичную часть ключа:

```
# ssh user@client1.test.alt
[user@client1 ~]$ su -
Password:
[root@client1 ~]# cat /home/user/.ssh/authorized_keys >>.ssh/authorized_keys
[root@client1 ~]# exit
выход
[user@client1 ~]$ exit
выход
Connection to client1.test.alt closed.
```

1.7.2. Интерактивная установка

Установить пакет *ansible*:

```
# apt-get install ansible
```

Для получения списка хостов ansible обращается к файлу **`/etc/ansible/hosts`**. Создать файл **`/etc/ansible/hosts`** со следующим содержимым:

```
[agents]
client1.test.alt
client2.test.alt
[agents:vars]
ansible_user=root
ansible_python_interpreter=/usr/bin/python3
```

Проверить работу ansible, выполнив «ping» на группу хостов agents:

```
# ansible -m ping agents
client1.test.alt | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
client2.test.alt | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Создать файл **/etc/smart-proxy/config/settings.d/ansible.yml** со следующим содержимым:

```
---
:enabled: true
:ansible_dir: /usr/lib/foreman
:working_dir: /tmp
```

1.7.3. Настройка плагина TFTP

Для задействования функции работы с TFTP-сервером необходимо создать и отредактировать файл **/etc/smart-proxy/config/settings.d/tftp.yml**. Можно взять за основу файл **/etc/smart-proxy/config/settings.d/tftp.yml.example**:

```
# cp /etc/smart-proxy/config/settings.d/tftp.yml.example /etc/smart-proxy/config/
settings.d/tftp.yml
```

Привести содержимое файла **/etc/smart-proxy/config/settings.d/tftp.yml** к виду:

```
---
# Can be true, false, or http/https to enable just one of the protocols
:enabled: true

#:tftproot: /var/lib/tftpboot
# Defines the TFTP Servername to use, overrides the name in the subnet
declaration
:tftp_servername: foreman.test.alt

# Defines the default connection timeout in seconds needed to download tftp
artifacts
# like initrd and vmlinuz. Default value 10 seconds
#:tftp_connect_timeout: 10

# Defines the default certificate action for certificate checking.
# When false, the argument --no-check-certificate will be used.
#:verify_server_cert: true
```

1.7.4. Настройка плагина DHCP

Для подключения возможности использовать DHCP необходимо создать и отредактировать файл **/etc/smart-proxy/config/settings.d/dhcp_isc.yml**. Можно взять за основу файл **/etc/smart-proxy/config/settings.d/dhcp_isc.yml.example**:

```
# cp /etc/smart-proxy/config/settings.d/dhcp_isc.yml.example /etc/smart-proxy/
config/settings.d/dhcp_isc.yml
```

Привести содержимое файла **/etc/smart-proxy/config/settings.d/dhcp_isc.yml** к виду:

```
---
#
# Configuration file for ISC dhcp provider
#

:config: /etc/dhcp/dhcpd.conf
:leases: /var/lib/dhcp/dhcpd/state/dhcpd.leases
#
# Redhat 5
#
#:config: /etc/dhcpd.conf
#
# Settings for Ubuntu
#
#:config: /etc/dhcp3/dhcpd.conf
#:leases: /var/lib/dhcp3/dhcpd.leases

# Specifies TSIG key name and secret
#:key_name: secret_key_name
#:key_secret: secret_key

:omapi_port: 7911

# use :server setting in dhcp.yml if you are managing a dhcp server which is not
localhost
```

В файле **/etc/smart-proxy/config/settings.d/dhcp.yml** необходимо указать параметры DHCP. Можно взять за основу файл **/etc/smart-proxy/config/settings.d/dhcp.yml.example**:

```
# cp /etc/smart-proxy/config/settings.d/dhcp.yml.example /etc/smart-proxy/config/
settings.d/dhcp.yml
```

Привести содержимое файла **/etc/smart-proxy/config/settings.d/dhcp.yml** к виду:

```
---
# Can be true, false, or http/https to enable just one of the protocols
:enabled: true

# valid providers:
#   - dhcp_isc (ISC dhcp server)
#   - dhcp_native_ms (Microsoft native implementation)
#   - dhcp_libvirt
:use_provider: dhcp_isc
:server: 192.168.0.148
# subnets restricts the subnets queried to a subset, to reduce the query time.
:subnets: [192.168.0.0/255.255.255.128, 192.168.0.128/255.255.255.128]

# Perform ICMP and TCP ping when searching free IPs from the pool. This makes
# sure that active IP address is not suggested as free, however in locked down
# network environments this can cause no free IPs. Enabled by default
:ping_free_ip: true
```

Если DHCP-сервер располагается на одном сервере с сервером smart-proxy, необходимо добавить пользователя _smartforeman в группы named и dhcp:

```
# gpasswd -a _smartforeman named
# gpasswd -a _smartforeman dhcp
```

Чтобы пользователь _smartproxy смог читать файл конфигурации DHCP-сервера следует установить атрибуты на файл **/etc/dhcp/dhcpd.conf**:

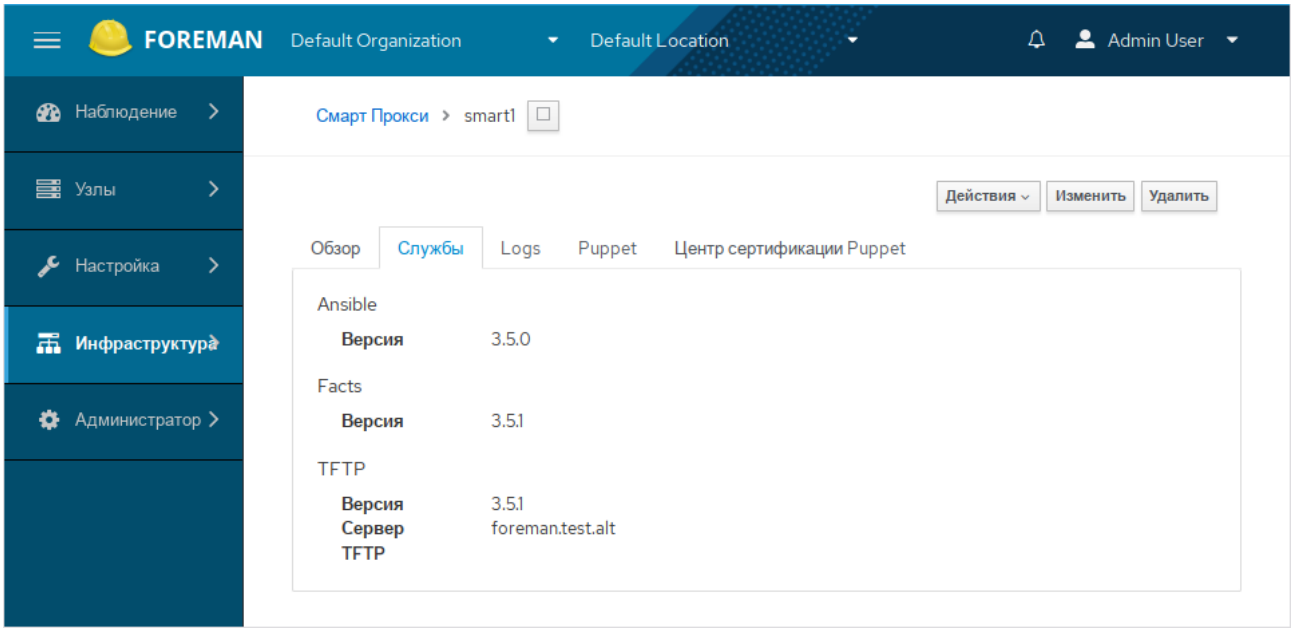
```
# chmod 644 /etc/dhcp/dhcpd.conf
```

1.7.5. Проверка версий

Проверить версию smart-proxy и подключенных плагинов можно, используя API-запрос:

```
$ curl -k -H "Accept: application/json" http://localhost:8000/version
{"version":"3.5.1","modules":
{"ansible":"3.5.0","facts":"3.5.1","tftp":"3.5.1","puppetca":"3.5.1","puppet":"3.
5.1","logs":"3.5.1"}}
```

Версии подключенных плагинов в веб-интерфейсе:



1.8. Настройка агента Puppet

Задать агенту полное доменное имя (FQDN):

```
# hostnamectl set-hostname client1.test.alt
```

Установить пакеты *puppet* и *java-17-openjdk*:

```
# apt-get install puppet java-17-openjdk
```

Версия java может быть другой.

Привести конфигурационный файл Puppet (**/etc/puppet/puppet.conf**) к виду:

```
[agent]
server = foreman.test.alt
```

Запустить службу puppet:

```
# systemctl enable --now puppet
```

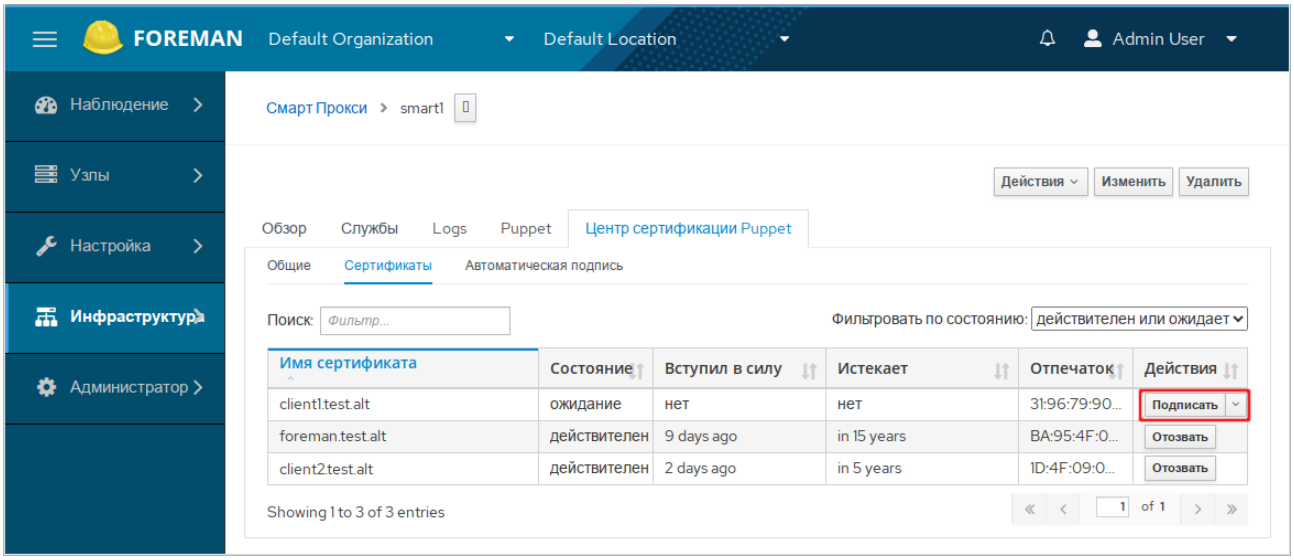
При первом запуске клиентская служба Puppet Agent отправит на Puppet Server запрос на подпись сертификата. Для просмотра списка запросов на подпись сертификата можно выполнить на сервере следующую команду:

```
# puppetserver ca list
Requested Certificates:
client1.test.alt       (SHA256)  A9:55:45:0B:47:70:B0:04:52:9B:
91:28:88:77:25:E3:95:2D:65:80:68:B2:D3:3E:73:BC:62:21:1B:8B:30:C6
```

Подписать сертификат:

```
# puppetserver ca sign --certname client1.test.alt
Successfully signed certificate request for client1.test.alt
```

Подписать сертификат можно и в веб-интерфейсе Foreman. Для этого следует перейти в **Инфраструктура** → **Капсулы** выбрать нужный Smart Proxy, перейти на вкладку **Центр сертификации Puppet** → **Сертификаты**. Здесь нужно выбрать сертификат и нажать кнопку **Подписать**:



Проверка работы агента на клиентской машине после подписания сертификата:

```
# systemctl stop puppet
# puppet agent -t
Info: csr_attributes file loading from /etc/puppet/csr_attributes.yaml
Info: Creating a new SSL certificate request for client1.test.alt
Info: Certificate Request fingerprint (SHA256): 77:C6:94:A3:DC:A2:FB:04:A6:A5:3E:
79:E3:04:79:2E:48:C4:E5:B0:1B:36:3F:10:71:59:DB:AB:C7:23:5B:19
Info: Downloaded certificate for client1.test.alt from https://foreman.test.alt:
8140/puppet-ca/v1
Info: Using environment 'production'
Info: Retrieving pluginfacts
```

```
Info: Retrieving plugin
Info: Caching catalog for client1.test.alt
Info: Applying configuration version '1696874994'
Notice: Applied catalog in 0.02 seconds
```

Повторно запустить службу:

```
# systemctl start puppet
```

1.9. Отладка

1.9.1. Puppet

Уровень журналирования Foreman устанавливается в файле **/etc/foreman/settings.yaml**:

```
:logging:
  :level: info
  :production:
    :type: file
    :layout: pattern
```

Отладочная информация Puppet добавляется в следующие файлы журнала:

- **/var/log/puppetserver/puppetserver-daemon.log**
- **/var/log/puppetserver/puppetserver.log**
- **/var/log/puppetserver/puppetserver-status.log**

Отладочная информация Puppet Agent добавляется в следующие файлы журнала:

- **/var/log/puppet/**

1.9.2. Foreman

Уровень журналирования Foreman устанавливается в файле **/etc/foreman/settings.yaml**:

```
:logging:
  :level: info
  :production:
    :type: file
    :layout: pattern
```

Отладочная информация Foreman добавляется в следующие файлы журнала:

- **/var/log/foreman/apipie_cache.log**
- **/var/log/foreman/bundle.log**
- **/var/log/foreman/db_setup.log**
- **/var/log/foreman/generators.log**
- **/var/log/foreman/key_generation.log**

- **/var/log/foreman/nginx_access.log**
- **/var/log/foreman/nginx_error.log**
- **/var/log/foreman/production.log**
- **/var/log/foreman/tmp_clear.log**

1.9.3. Smart Proxy

Уровень журналирования Smart Proxy устанавливается в файле **/etc/smart-proxy/config/settings.yml**:

```
:log_file: /var/log/smart-proxy/proxy.log
:log_level: DEBUG
```

Отладочная информация Smart Proxy добавляется в следующие файлы журнала:

- **/var/log/smart-proxy/proxy.log**
- **/var/log/foreman/run_errors.log**