ПАК «ПАТРИОТ»

СИСТЕМА УПРАВЛЕНИЯ

Руководство администратора

Версия 1.0

ОГЛАВЛЕНИЕ

Система управления «Патриот» предназначена для автоматизации задач системного администрирования парка АРМ и серверов организации.

1.1. Список терминов и сокращений

Термин, сокращение	Определение
СУ	Система управления «Патриот»
АРМ	Автоматизированное рабочее место
БД	База данных
СУБД	Система управления базами данных
ОС	Операционная система
DNS (Domain Name System)	«система доменных имён» — система для получения информации о доменах. Используется для получения IP-адреса по имени хоста (компьютера или устройства).

В данном руководстве приводятся конкретные команды которые нужно исполнить в терминале ОС, такие команды выведены специальным шрифтом на сером фоне, например:

```
#apt-get update
```

– данный символ перед командой означает что ее исполнение необходимо проводить в режиме суперпользователя, именно так обозначается приглашение командной строки в данном режиме, вводить # не нужно.

$ – данный символ перед командой означает что ее исполнение необходимо проводить в режиме пользователя, именно так обозначается приглашение командной строки в данном режим, вводить $ не нужно.

1.2. Основные понятия и определения

Сервер управления (Foreman) - это средство автоматизации повторяющихся задач, развёртывания приложений и конфигураций, мониторинга исполнения назначенных задач, web-интерфейс администратора.

Сервер конфигураций (Puppet Server) – обеспечивает распространение и хранение конфигураций, а также получает отчеты от клиентов по примененным конфигурациям, устанавливается с сервером управления.

Сервер СУБД – СУБД для хранения данных системы

Модули управления – типовые сценарии управления парком серверов и АРМ для российских ОС разработанные ООО «Галэкс Сервис»

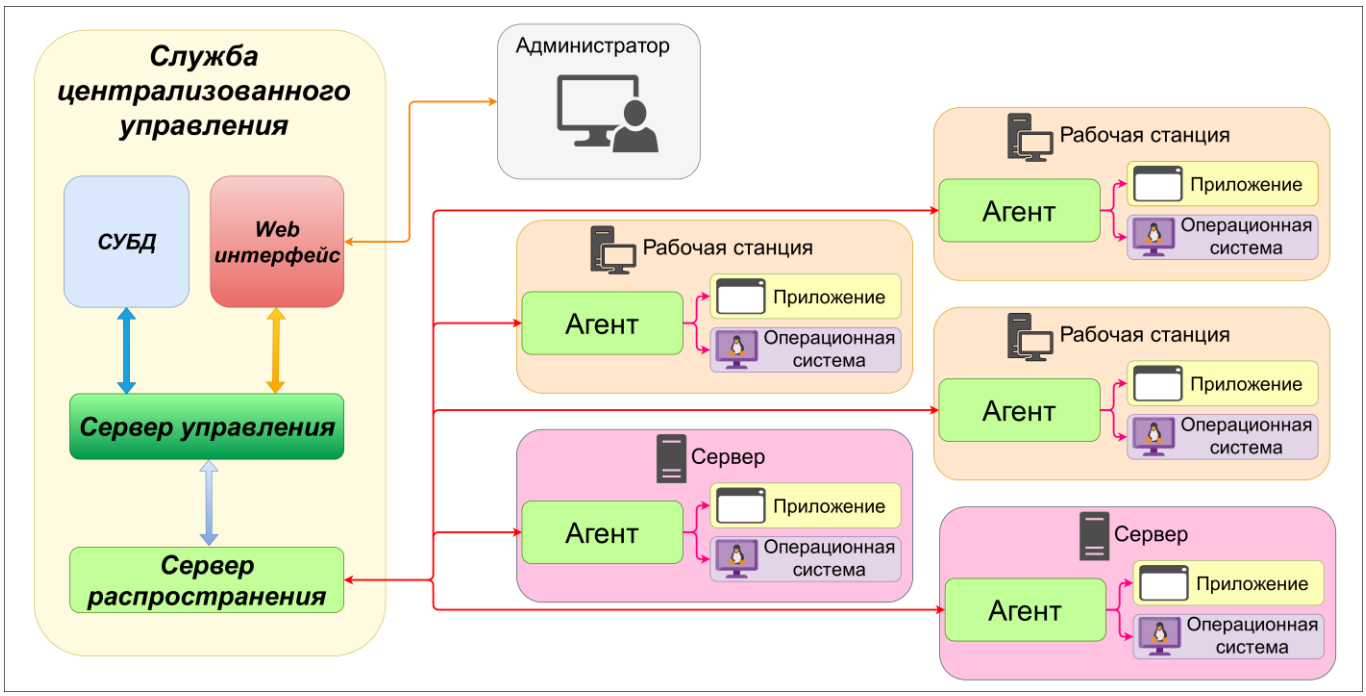


Рисунок 1 – структурная схема

Агент (puppet agent) – клиентский агент обеспечивающий применение конфигураций на конечном устройстве, агент устанавливается на каждом управляемом устройстве.

Сервер распространения (Smart proxy, капсула) – сервера, позволяющие обеспечить распределенную сеть для применения настроек, Smart Proxy получает настройки от центрального сервера управления, обеспечивает их хранение, применение на целевых АРМ, сбор отчетов, а также предоставляет локализованные службы Puppet server, Puppet CA, DHCP, DNS, TFTP.

Smart Proxy помогают масштабировать среду по мере увеличения количества управляемых АРМ для распределения нагрузки, а также используются в различных географических местоположениях.

Один экземпляр smart-proxy установлен на центральном сервере, он же является Центром сертификации Puppet для подписания сертификатов клиентов.

Факты (facts) – информация о системе, на которой установлен агент. Фактом является любой подлежащий инвентаризации параметр АРМ, например: модель ЦПУ, количество ядер, версия определенного файла. Факты делятся на встроенные - предопределенны в Puppet агенте и пользовательские - внешние факты, которые могут быть описаны самостоятельно. Факты могут использоваться в различных случаях: в манифестах для определения действия в зависимости от значения факта, формирование группы узлов на основе фактов, для получения дополнительной информации необходимой администратору и т.д. На рисунке 2 приведен пример фактов, полученных с АРМ.

▲ Имя	Значение	Origin	Получено	Действия
architecture	x86_64	P	1 месяц назад	View Chart
bios_release_date	04/10/2020	P	1 месяц назад	View Chart
bios_vendor	American Megatrends Inc.	P	1 месяц назад	View Chart
bios_version	X421DA.301	P	1 месяц назад	View Chart
blockdevice_nvme0n1_model	KINGSTON OM8PCP3512F-AB	P	1 месяц назад	View Chart
blockdevice_nvme0n1_size	512110190592	P	1 месяц назад	View Chart
blockdevices	nvme0n1	P	10 дней назад	View Chart
boardmanufacturer	ASUSTeK COMPUTER INC.	P	1 месяц назад	View Chart
boardproductname	X421DA	P	1 месяц назад	View Chart

Рисунок 1 - Пример фактов полученных с узла

Балансировщик нагрузки (HAproxy) - ПО для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений, посредством распределения входящих запросов на несколько обслуживающих серверов.

Узел (Host) – АРМ с установленной ОС управляемый СУ Патриот.

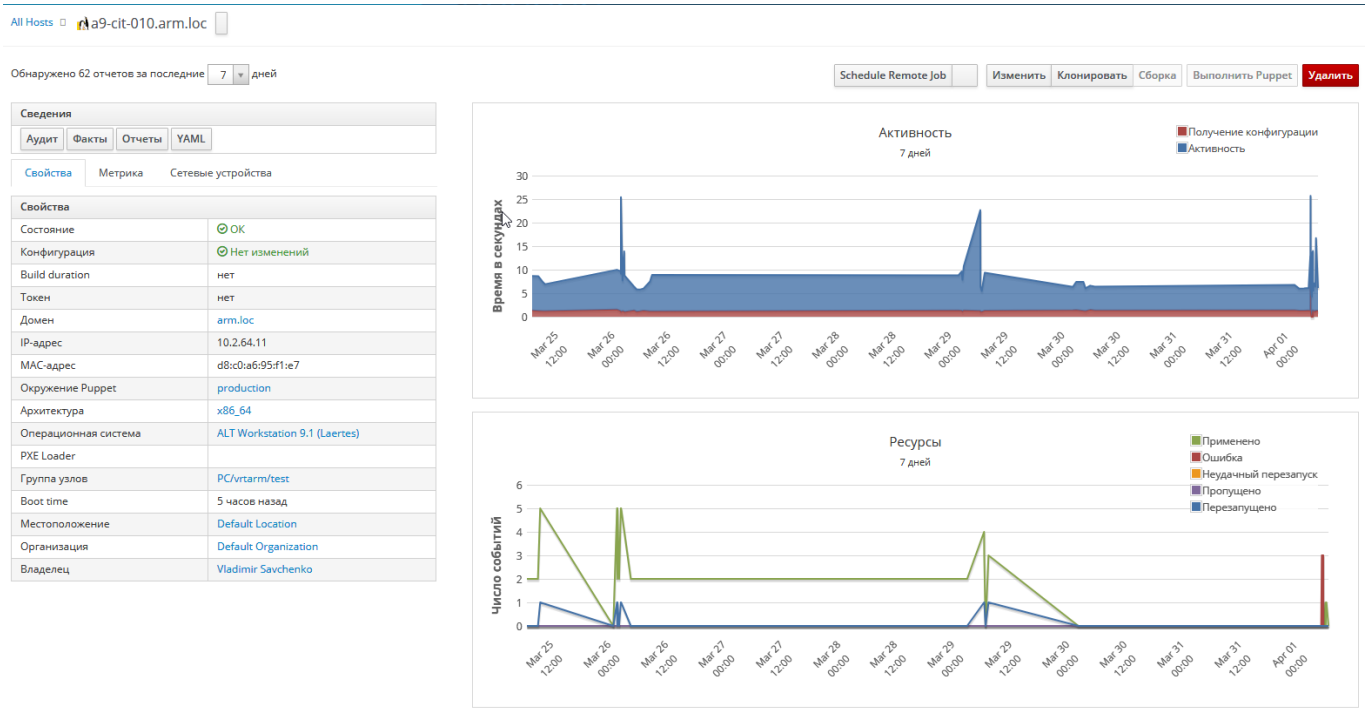


Рисунок 2 - Пример информации об узле

Полный перечень узлов доступен в разделе *Узлы (All Hosts)*. Веб интерфейс позволяет фильтровать узлы по предварительно настроенным параметрам в разделе *Закладки*. Например, если в строке поиска раздела узлы нажать *Закладки* и выбрать *«out of sync»*, то будут отображены узлы отчет от которых, поступал более 35 минут назад. Полный перечень готовых фильтров с указанием

к какому разделу он применим находится в разделе *Администратор – Закладки*. Любой существующий поисковый запрос может быть сохранен в закладки.

Группа узлов (Host Group) – объединение узлов для распространения настроек, притом узел может состоять только в одной группе узлов. По умолчанию назначение узлов в группы производится администраторов вручную.

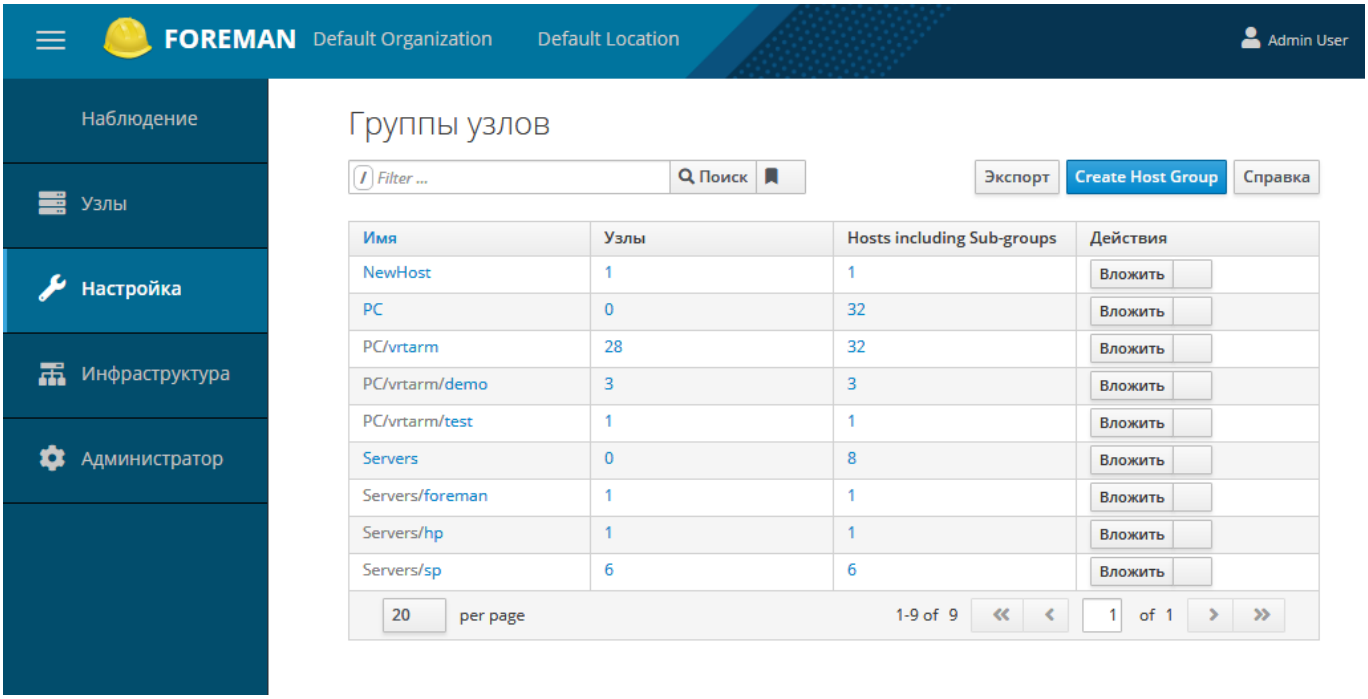


Рисунок 3 - Пример реализации группы узлов

Группы узлов могут быть вложенными для наследования параметров друг друга, что позволяет создавать иерархии групп узлов сети, соответствующие конкретным рабочим процессам (Рисунок 3). Ниже приведены примеры построения структур:

Плоская структура – структура, при которой наследование исключается, однако существует риск большого дублирования настроек между группами узлов.

Структура на основе местоположения или принадлежности к организации – структура, при которой местоположение узлов согласовано со структурой группы узлов сети. Такой подход оправдан, когда местоположение определяет применяемые настройки. Но, с другой стороны, в сложных инсталляциях с большим количеством приложений приходится переопределять применяемые настройки многократно.

Структура жизненного цикла, основанная на среде - структура, при которой первый уровень группы хостов зарезервирован для параметров специфичных для среды жизненного цикла. Второй уровень содержит определения, связанные с операционной системой, а третий уровень содержит настройки для конкретных приложений.

В сервисе распространения конфигураций обеспечен комплексный подход к делегированию полномочий, на основании организаций и местоположения. Это подход позволяет разделить администрирование парком АРМ, принадлежащего различным ведомствам или отделам.

Организация (Organization) – логическая структура для группировки ресурсов по организационному признаку для удобства управления и делегирования. Организации могут быть выстроены в иерархическую структуру. Каждый узел принадлежит только одной организации. **Местоположение (Locations)** – логическая структура для группировки ресурсов по территориальному признаку для удобства управления и делегирования. Рекомендуется использовать в рамках организации для разделения по местоположению, при необходимости. Местоположения могут иметь иерархическую структуру. Один узел может находиться только в одном местоположении.

Пользователь (User)– учетная запись для работы с системой управления. Права пользователя определяются назначенными ему ролями, а также назначенными ему организациями и местоположениями. Пользователь может быть внутренним (учетные данные хранятся внутри системы) или внешними (импортированными из LDAP).

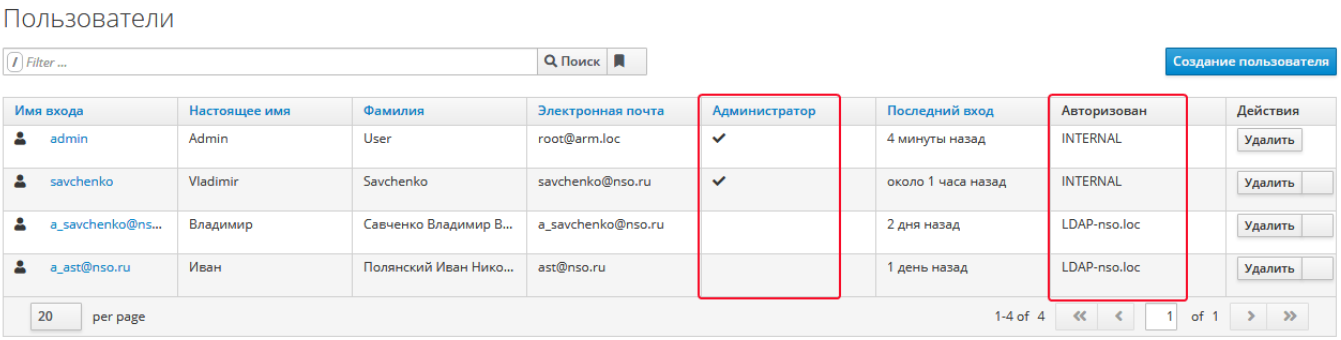


Рисунок 4 - Пример импорта пользователей из LDAP

Аутентификация LDAP -раздел, позволяющий настроить импорт пользователей из каталогов Active Directory, FreeIPA и Samba, сервис поддерживает неограниченное количество разнообразных внешних источников для импорта учетных записей (Рисунок 4).

Внимание! Для реализации соединения со службой каталогов по протоколу LDAPS требуется импортировать сертификат сервера, поместив его в каталог сертификатов в /etc/pki/ca-trust/source/anchors/ и обновить общесистемный список доверенных CA, командой:

```
#update-ca-trust
```

После перезапустить службу:

```
#systemctl restart foreman.service
```

Группа пользователей (User Groups) – объединение пользователей в группы для удобства управления и назначения ролей. Группы могут содержать:

- внутренних пользователей,
- внутренние группы,
- внешние группы из LDAP.

Роль (Role) – перечень действий, доступных для выполнения или просмотра (Рисунок 5). Система имеет ряд предустановленных ролей и механизм самостоятельного создания ролей. Роль может быть назначена на пользователя или группу пользователей.

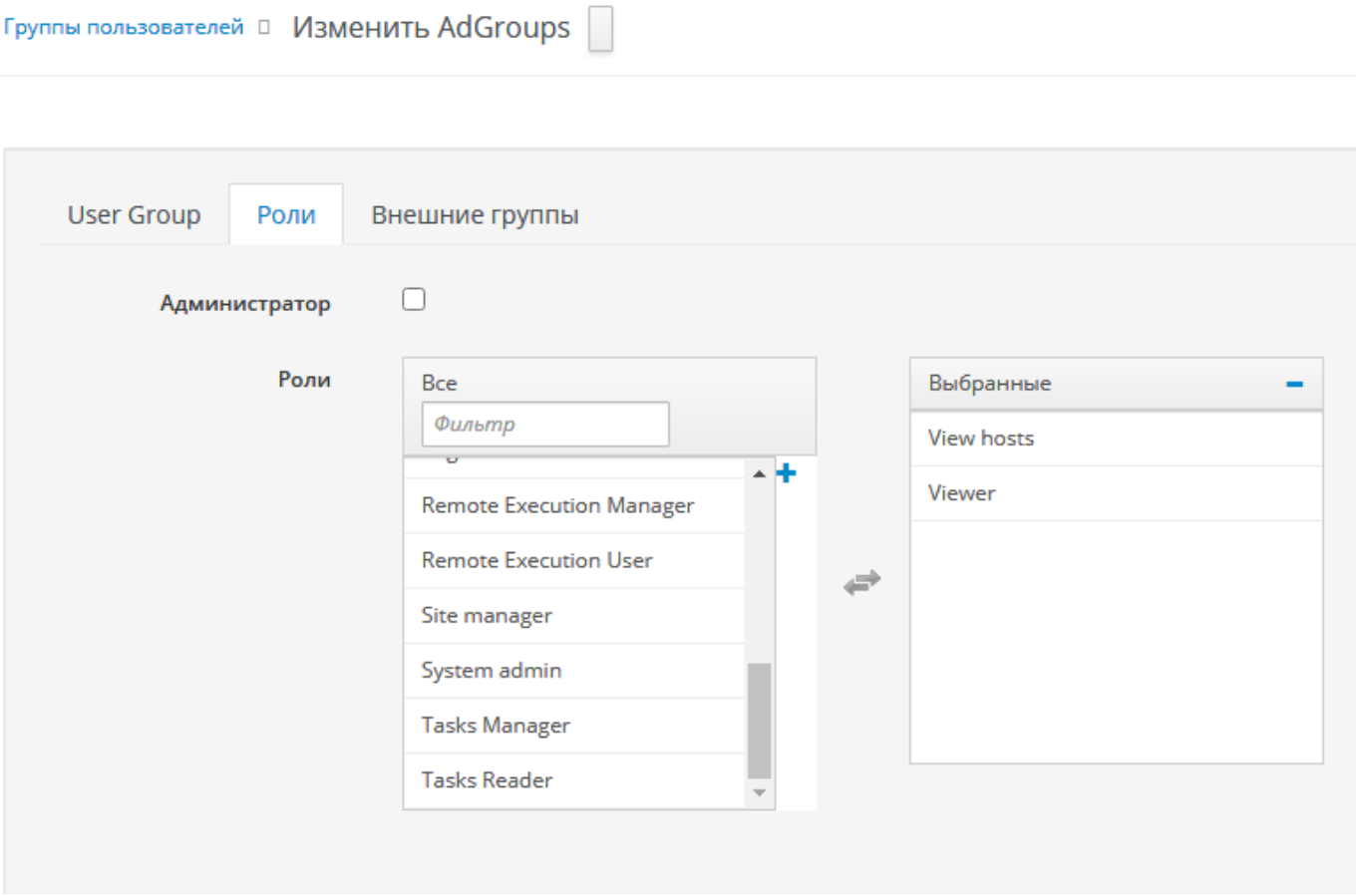
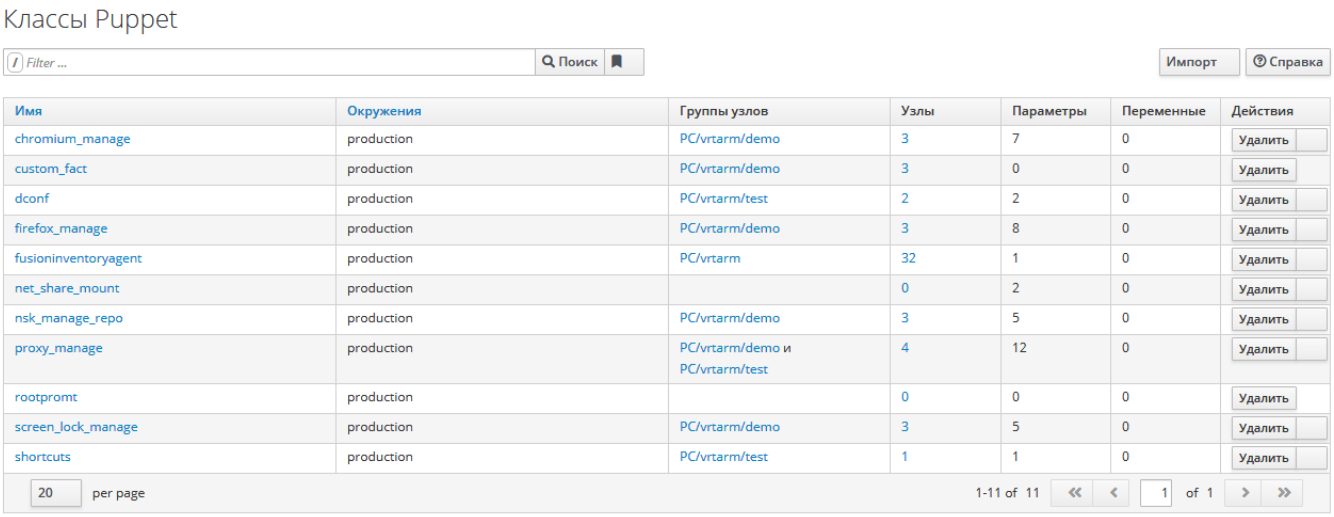


Рисунок 5 - Пример назначений ролей группе пользователей

Модули управления (Puppet Classes) – набор настроек, применяемых на узел или группу узлов. Содержит в своем составе модуль, написанный на языке Puppet, и значения переменных, определенных в веб-интерфейсе Foreman в настройках модуля (раздел смарт-параметры, Smart Parameters). Класс Puppet может быть назначен только на группу узлов или непосредственно на узел (Рисунок 6). Назначить класс на домен, организацию или местоположения - нельзя. Назначать класс на конкретный узел не рекомендуется, т. к. усложняется управление настройками.



Имя	Окружения	Группы узлов	Узлы	Параметры	Переменные	Действия
chromium_manage	production	PC/vrtarm/demo	3	7	0	Удалить
custom_fact	production	PC/vrtarm/demo	3	0	0	Удалить
dconf	production	PC/vrtarm/test	2	2	0	Удалить
firefox_manage	production	PC/vrtarm/demo	3	8	0	Удалить
fusioninventoryagent	production	PC/vrtarm	32	1	0	Удалить
net_share_mount	production		0	2	0	Удалить
nsk_manage_repo	production	PC/vrtarm/demo	3	5	0	Удалить
proxy_manage	production	PC/vrtarm/demo и PC/vrtarm/test	4	12	0	Удалить
rootpromt	production		0	0	0	Удалить
screen_lock_manage	production	PC/vrtarm/demo	3	5	0	Удалить
shortcuts	production	PC/vrtarm/test	1	1	0	Удалить

Рисунок 6 - Пример классов Puppet и назначения их на группы узлов

Для получения полного перечня переменных классов и количества переопределений значений переменных для них, служит страница Smart Class Parameters.

Окружения (Puppet Environments) – это независимые наборы конфигурации, которые хранятся в отдельных каталогах. Каждый узел должен находиться в каком-то одном окружении. У каждого окружения свои модули и манифесты.

Группы конфигураций (Config Groups) -способ назначить несколько классов на узел или группу узлов, при этом задание переменных класса все равно происходит внутри класса, а не внутри группы.

Требования к ресурсам и установка системы управления описаны в документе: «Руководство по установке»

После установки веб интерфейс системы доступен по адресу: http://ip:2345, логин и пароль пользователя генерируется при установке системы.

Значение переменных по умолчанию для класса puppet задается в разделе Смарт-параметры - Default Behavior. В этом разделе определяется или переопределяется значение по умолчанию для всех. Если требуется определить значение переменной в зависимости от атрибутов узла или его принадлежности группе узлов необходимо задействовать режим переопределения значения в разделе Specify Matchers.

Переопределение значения может быть произведено на следующих уровнях:

- Узел;
- Группа узлов;
- Операционная система, учитывается имя и версия;
- Домен узла;
- Организация;

- Местоположение;
- Имя факта.

Перечень переопределений доступен в разделе Specify Matchers. Задается оно в виде: **Тип атрибута – Значение** (Рисунок 8).

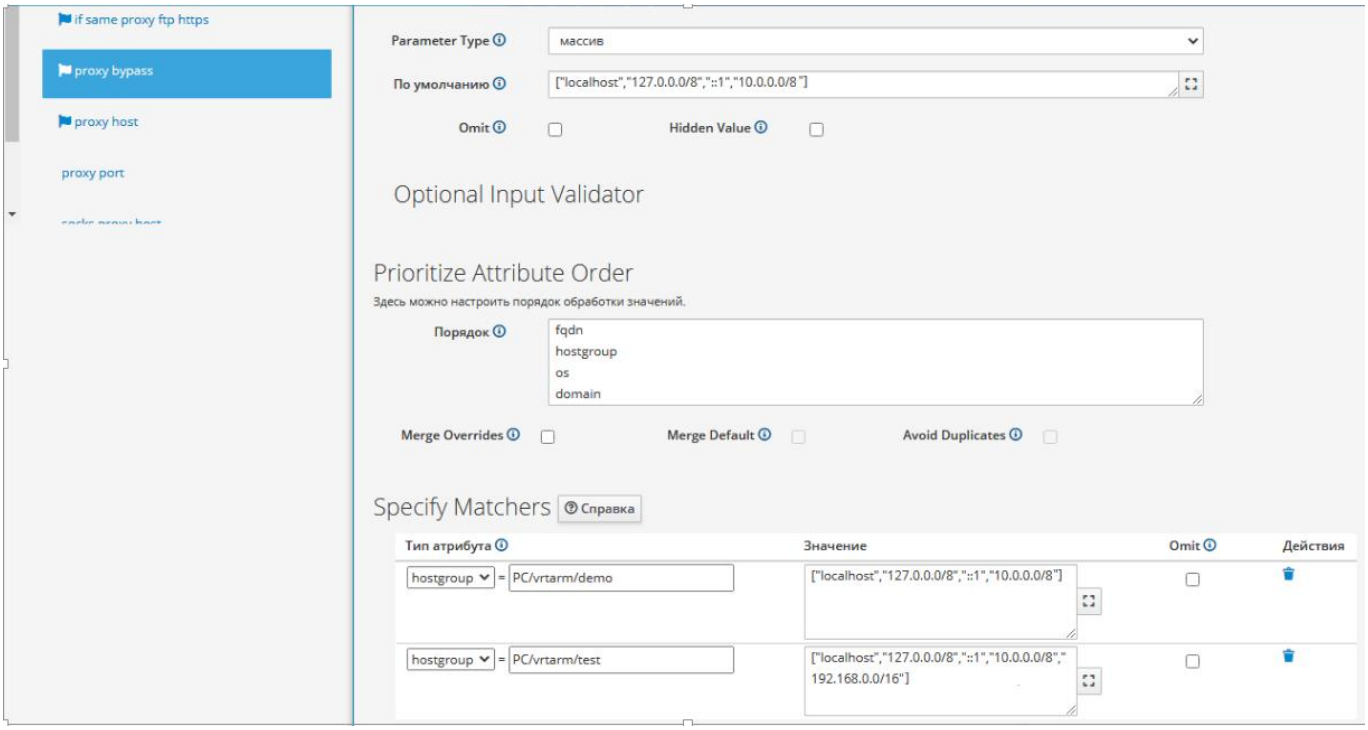


Рисунок 8 - Пример задания значения параметра с переопределением для разных групп узлов

Переопределение работает следующим образом:

- Перебираются все переопределения в порядке, указанном в Prioritize Attribute Order до первого совпадения;
- Если совпадений не найдено, используются значения по умолчанию;
- При необходимости, администратор может изменить порядок применения настроек для конкретной переменной. Например, указать что более высокий приоритет имеет организация, а не узел или группа узлов. Происходит это изменением порядка списка ключей, более приоритетный атрибут помещается выше.

4.1. Управление пользователями и группами.

Модуль позволяет управлять локальными учетными записями пользователей, группами на целевых АРМ.

Функционал модуля:

- Управление локальными пользователями (создание, удаление, домашняя папка, SSH-ключи, состав групп, пароли);
- Управление локальными группами;

- Сопоставление локальных и доменых групп;
- Управление настройками sudo.

Поддерживаемые ОС: Альт, Альт СП.

Модуль состоит из нескольких подмодулей (классов Foreman) описание которых приведено ниже.

4.1.1. Управление локальными пользователями

Функционал подмодуля:

- Создание локальных пользователей;
- Удаление локальных пользователей;
- Смена пароля пользователя;
- Назначение групп пользователя;
- Назначение домашнего каталога;
- Управление ssh ключами пользователя;

Название класса в Foreman: manage_accounts.

Принцип работы:

Для пользователя пароль задается в виде хэша Blowfish. Что бы создать такой хэш можно использовать следующую команду в терминале:

```
htpasswd -bnBC 12 user <Password>
```

Пакет htpasswd при необходимости ставится командой:

```
#apt-get install apache2-htpasswd
```

Внимание! Если задать, что пользователь должен быть создан, но не задать его пароль, то пользователь будет без пароля!

Внимание! При задании ключей SSH система управления контролирует перечень ключей пользователя. Если ключ не указан в перечне ключей ssh_authkeys, он будет удален. Также отдельно можно для конкретного ключа указать, что его нужно удалить.

Перечень параметров:

Настройка (переменная)	**Тип**	**Значение**	**Описание**
Username	Строка		Имя создаваемого пользователя
ensure	строка	*Present (по умолчанию)*	Создать пользователя
		absent	Удалить пользователя

Настройка (переменная)	Тип	Значение	Описание
uid	Целое число		ID пользователя. Если не задано создается автоматически.
gid	Целое число или строка		Первичная группа пользователя. Если не задано создается автоматически.
groups	Массив		Перечень групп, куда входит пользователь.
shell	строка	*/bin/bash (по умолчанию)*	Оболочка пользователя по умолчанию
pwhash	Строка		Захэшированный пароль пользователя
home	строка	*/home/$username*	Домашний каталог пользователя
managehome	Логическая переменная	*True (по умолчанию)*	Определяет создавать ли пользовательский каталог
sync_home	Логическая переменная	*False (по умолчанию)*	Определяет использовать ли шаблон при создании домашнего каталога.
sync_home_src	Строка		Источник, откуда копировать шаблон домашнего каталога
home_owner	Строка	*$username (по умолчанию)*	Владелец домашнего каталога
home_group	Строка		Группа, которой принадлежит домашний каталог пользователя
home_perm	Строка	*0700*	Права доступа на домашний каталог пользователя
manage_ssh_authkeys	Логическая переменная	*False (по умолчанию)*	Управлять или нет ключами SSH.
ssh_authkeys	Хэш-таблица		SSH ключи для пользователя. Переменная действительна только если manage_ssh_authkeys = true
Описание полей хэш-таблицы ssh_authkeys			
Название ключа	Строка		Имя пользователя и узла для ключа. Может быть произвольным, но должно быть уникальным внутри хэш-таблицы

Настройка (переменная)	Тип	Значение	Описание
user	Строка		Имя пользователя, для которого задается ключ.
ensure	строка	*Present (по умолчанию)*	Создать ключ
		absent	Удалить ключ
type	Строка	*ssh-rsa (по умолчанию)*	Тип ключа. Возможные значения: 'ssh-dss', 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519'
Key	Строка		Ключ SSH

Пример использования:

```
test1:
 ensure: present
 pwhash: "$2y$05$lI9y4ChxJNQicf5zoGZt2lGR4n."
 home_perms: '0700'
 groups:
 - wheel
 - uucp
 - proc
 - audio
 - users
 home_group: users
 manage_ssh_authkeys: true
 ssh_authkeys:
   host@lan:
     user: test1
     ensure: absent
     type: ssh-rsa
     key: AAAAB3NzaC1yc2E8PfyyzSQAiN/f7rqgwjDpbUiEJzxtm/j9pqquMCDeBX+1ZwRh1
   lan@host:
     user: test1
     ensure: present
     type: ssh-rsa
```

```
    key: AAAAB3NzaC1yc2EAAAABJQAAAZpQ39cIW6yUPZJzpeo/KpfpStr4Jdw==
test2:
 ensure: present
 pwhash: "$2y$05$lI9yoGZt2lGR4n."
 home_perms: '0700'
test3:
 ensure: absent
```

Результат выполнения примера:

1. cat /etc/group | grep <username> – проверка пользователя и его групп
2. su <username> – вход в систему под пользователем. Проверка пароля

4.1.2. Управление локальными группами

Функционал подмодуля:

- Создание локальных групп;
- Удаление локальных групп;

Название класса в Foreman: manage_accounts.

Принцип работы:

Данный модуль создает и удаляет группы согласно заданному перечню. Перечень пользователей входящих в группу контролируется через управление пользователем.

Перечень параметров:

Настройка (переменная)	Тип	Значение	Описание
Groupname	Строка		Имя создаваемой группы
ensure	строка	*Present (по умолчанию)*	Создать группу
		absent	Удалить группу
gid	Целое число		ID группы. Если не задано создается автоматически.

Пример использования:

```
supergroup:
 ensure: present
```

Результат выполнения примера:

На целевом АРМ будет создана группа supergroup, gid группы будет присвоен автоматически.

4.1.3. Сопоставление доменных и локальных групп

Функционал подмодуля:

- Сопоставление доменных и локальных групп.

Название класса в Foreman: manage_accounts.

Принцип работы:

Для реализации функции используется библиотека для службы переключения имен и набор инструментов для администрирования ролей и привилегий - libnss-role. Данная библиотека должна присутствовать на клиенте, для ее развертывания можно использовать модуль установки приложений. Подробнее с функциями библиотеки можно ознакомится на ресурсе: https://github.com/Etersoft/libnss-role/blob/master/README-ru.md

Служба переключения имен (NSS) – соединяет компьютер с различными источниками общих баз данных конфигурации и механизмов разрешения имен. Например: /etc/passwd, /etc/group, /etc/hosts.

Модуль ролей – это модуль для службы переключения имён NSS. Модуль реализует возможность добавления групп в группы. Для администрирования модуля ролей реализованы специальные вспомогательные утилиты, которые рассматривают все группы поделёнными на две категории: роли и привилегии.

Управление ролями проводится с помощью конфигурационных файлов в каталоге **/etc/role.d** и файле **/etc/role**.

Модуль контролирует содержимое каталога **/etc/role.d** поддерживая, состояние 3 файлов по умолчанию. А также через шаблон задает состояние файла **/etc/role**.

Перечень параметров:

<имя роли>: <перечень групп, входящих в роль.>

По одной роли в строке.

Пример использования:

```
- "Domain Users:users"
- "Domain Admins:localadmins"
- "RDP:tsusers,fuse"
- "RDPAdmins:tsusers,tsadmins,fuse"
- "rdp users 087@ARM.LOC:tsusers"
```

Результат выполнения примера:

На целевом АРМ выполнить можно отследить применение параметров просмотрев файл с группами:

```
#cat /etc/role
```

в котором должны значится заданные группы и пользователи.

4.1.4. Управление правами на sudo

Функционал подмодуля:

- Управление конфигурационным файлом программы sudo

Название класса в Foreman: manage_accounts.

Принцип работы:

1. Через шаблон удерживается состояние файла /etc/sudoers. Поменять через Foreman его содержимое нельзя!
2. Содержимое каталога **/etc/sudoers.d** полностью удаляется, кроме файла **/etc/sudoers.d/sudo_custom**, его содержимое задается через шаблон. Этим файлом можно управлять через Foreman.

Перечень параметров:

Настройка (переменная)	Тип	Значение	Описание
if_manage_sudo	Логическая переменная	*False (по умолчанию)*	Определяет управляет ли модуль настройками sudo. Если true, то ставит пакет sudo, состояние файла /etc/sudoers не контролируется, каталог /etc/sudoers.d и его содержимое - контролируется. Внимание! Если значение переменной if_manage_sudo = false, остальные параметры игнорируются.
if_manage_sudoers	Логическая переменная	*False (по умолчанию)*	Определяет контролируется ли состояние файла /etc/sudoers.
sudo_priv	Массив		Массив с перечислением прав, которые нужно задать для sudo

Пример использования:

```
- "user ALL=(ALL) ALL"
- "%group ALL=(ALL) NOPASSWD: ALL"
```

Для наделения группы правами на **sudo** перед именем группы нужно указать знак **%**.

Результат выполнения примера:

Задаются параметры в **/etc/sudoers.d/sudo_custom** из переменной **sudo_priv.** Проверить можно в соответствующем файле:

```
cat /etc/sudoers.d/sudo_custom
```

4.2. Настройка обновления операционной системы

Модуль позволяет управлять обновлениями ОС и прикладного ПО на целевых АРМ.

Функционал модуля:

- Управление списком репозиториев на локальной машине;

- Настройка локального репозитория для получения обновления;
- Запуск команды установки обновлений.

Поддерживаемые ОС: Альт, Альт СП, Astra Linux, РЕД ОС.

Модуль состоит из нескольких подмодулей (классов Foreman) описание которых приведено ниже.

4.2.1. Управление перечнем подключенных репозиториев

Функционал подмодуля:

- Управление списком репозиториев на целевом АРМ.

Название класса в Foreman: manage_repo.

Принцип работы:

В ОС Альт список репозиториев для пакетного менеджера задается в файле /etc/apt/sources.list, либо в любом файле с расширением list (например, mysources.list) в каталоге /etc/apt/sources.list.d/. Больше информации о репозиториях можно получить из документации к дистрибутивам https://docs.altlinux.org/ru-RU/index.html.

- Модуль управляет репозиториями внося изменения в файл /etc/apt/sources.list;
- Модуль удаляет файлы из каталога /etc/apt/sources.list.d/ и удерживает /etc/apt/sources.list в заданном состоянии;
- Ранее подключенные репозитории удаляются;
- После добавления репозиториев выполняет команду: apt-get update на АРМ к которым применен.

Перечень параметров для работы с репозиторием для ОС Альт:

Настройка (переменная)	Тип	Значение	Описание
server_url	*строковая*		Сервер репозиториев в формате: ***протокол://сервер/путь.*** Значение по умолчанию: не определено. Если значение не задано, модуль не отработает и выдаст ошибку. Применительно к развернутому модулю Система обновлений переменная равна **ftp://alt-mirror.arm.loc/mirror**

main_repo_chanel	*строковая*	*main*	Подключение текущей ветки
		stable	Подключение стабильной ветки
		history	Подключение исторической ветки
repo_arches	*строковый массив*	*['x86_64','noarch','x86_64-i586'] (по умолчанию)*	Перечень архитектур репозитория.
additional_repos	*Переменная типа массив хэш-таблиц*		Перечень дополнительных репозиториев (в формате: каталог репозитория - название репозитория - перечень платформ см.ниже).

additional_repos - переменная типа массив хэш-таблиц, позволяет задать перечень дополнительных репозиториев в формате: каталог репозитория, название репозитория, перечень платформ:

repo_dir - каталог на сервере, где расположен дополнительный репозиторий;

repo_name - название дополнительного репозитория;

arches - перечень архитектур дополнительного репозитория. Переменная типа массив.

Пример использования:

в формате YAML для подключения 2-х репозиториев: 265697 и customalt9:

```
- repo_dir: '265697'
  repo_name: 'task'
  arches:
  - 'x86_64'
  - 'x86_64-i586'
- repo_dir: 'customalt9'
  repo_name: 'customalt9'
  arches:
  - 'x86_64'
  - 'x86_64-i586'
  - 'noarch'
```

Или в формате JSON для этих же данных:

```
[{"repo_dir":"265697","repo_name":"task","arches":["x86_64","x86_64-i586"]},{"repo_dir":"customalt9","repo_name":"customalt9","arches":["x86_64","x86_64-i586","noarch"]}]
```

Результат выполнения примера:

На целевом АРМ подключены репозитории:

rpm <$server_url> 265697/x86_64 task

rpm <$server_url> 265697/x86_64-i586 task

rpm <$server_url> customalt9/x86_64 customalt9

rpm <$server_url> customalt9/x86_64-i586 customalt9

rpm <$server_url> customalt9/noarch customalt9

Перечень параметров для работы с репозиторием для ОС Astra:

repos - перечень основных репозиториев для ОС Astra. Переменная типа массив хэш-таблиц. Содержит в себе параметры:

- server_url - сервер репозиториев (в формате: протокол://сервер). Например: http://download.astralinux.ru.
- type_repo - тип репозитория. Переменная типа массив строк. Для каждой ОС свой набор, подробный перечень нужно уточнять в документации к дистрибутиву. Например: ["os","updates","extras"] для CentOS, ["main", "restricted"] для Ubuntu.

Внимание! В заголовке запрещены пробелы. Рекомендуется для разделения слов используется дефис.

additional_repos - перечень дополнительных репозиториев для ОС Astra. Переменная типа массив хэш-таблиц. Содержит в себе параметры:

- server_url - сервер репозиториев (в формате: протокол://сервер/путь). Например: http://download.astralinux.ru/astra.

Внимание! Некоторые репозитории изменяют стандартную структуру репозитория, поэтому во избежание неудачного добавления репозитория используется формат протокол://сервер/путь, а не протокол://сервер.

Для репозитория ОС Astra со стандартным набором и порядком параметров deb http://download.astralinux.ru/debian stretch main параметр server_url имеет вид http://download.astralinux.ru/debian.

- type_repo - тип репозитория. Обязателен для Debian-подобных ОС. Переменная типа массив строк. Например:["puppet7"].

Внимание! В заголовке запрещены пробелы. Рекомендуется для разделения слов используется дефис.

- code_name - указывает кодовое имя репозитория для Debian-подобных ОС. Переменная типа строка. По умолчанию не требуется. Необходима только если нужно добавить репозиторий другой ОС (например подключить к ОС Астра репозиторий Debian) или

репозиторий другой версии этой ОС (например, обновить Debian с вервии 9 на 10). Например: "stretch".

Пример использования Repos:

Пример в формате JSON для подключения репозитория для Astra 1.6:

```
[{"server_url":"http://download.astralinux.ru","type_repo":["main","contrib","non-free"]}]
```

По итогам работы модуля будут такие репозитории:

```
deb http://download.astralinux.ru/astra/main smolensk main contrib non-freedeb
http://download.astralinux.ru/astra/devel smolensk main contrib non-freedeb
http://download.astralinux.ru/astra/updates smolensk main contrib non-freedeb
http://download.astralinux.ru/update-dev smolensk main contrib non-free
```

Additional repos:

Пример в формате JSON для подключения к ОС Astra репозитория от Debian 9 (Stretch), для установки дополнительного ПО:

```
[{"server_url":"http://download.astralinux.ru/debian","type_repo":["main"],"code_name":"stretch"}]
```

По итогам работы модуля будут такие репозитории:

```
deb http://download.astralinux.ru/debian/ stretch main
```

4.2.2. Установка обновлений операционной системы

Подмодуль предназначен для выполнения команды на обновление целевых АРМ.

Функционал подмодуля:

- Обновление ядра;
- Обновление ПО;
- Управление перезагрузкой после обновления.

Название класса в Foreman: remote execution.

Принцип работы:

После того как на ОС настроены основные и дополнительные репозитории можно провести обновление ОС. Обновление проводиться по команде системного администратора. Для этого используются возможности плагина Remote Execution:

- На вкладке **Узлы – Все узлы** выбираем нужные узлы для обновления;
- По кнопке **Действия** выбираем **Schedule Remote Job**;
- Во вновь открывшемся окне выбираем в категории: **Commands**. В шаблоне задания: **Run command - SSH Default**;
- Выбираем расписание запуска: немедленно (по умолчанию)
  - В поле command, указываем параметры обновления (см.перечень)

- Применить;
- На вновь открывшемся окне изучаем результаты выполнения команды.

Перечень параметров:

Параметр	Для ОС Альт	Для ОС Astra
Выполнить обновление ОС	apt-get update && apt-get dist-upgrade	apt update && astra-update -A -r
Выполнить обновление ОС с перезагрузкой	apt-get update && apt-get dist-upgrade && reboot	apt update && astra-update -A -r && reboot
Выполнить обновление ОС и ядра	apt-get update && apt-get dist-upgrade && update-kernel	apt update && astra-update -A -r && update-kernel
Выполнить обновление ОС и ядра с перезагрузкой	apt-get update && apt-get dist-upgrade && update-kernel && reboot	apt update && astra-update -A -r && update-kernel && reboot

* Следует иметь в виду, что при указании команды reboot, задание, может не успеть вернуть статус исполнения перед перезагрузкой.

4.3. Управление ПО

Модуль позволяет управлять ПО на целевых АРМ.

Функционал модуля:

- Установка пакетов ПО по указанному перечню;
- Удаление пакетов ПО по указанному перечню.

Поддерживаемые ОС: Альт, Альт СП.

Модуль состоит из нескольких подмодулей (классов Foreman) описание которых приведено ниже.

4.3.1. Установка ПО

Модуль устанавливает пакеты программного обеспечения на целевые АРМ по указанному перечню.

Функционал модуля:

Установка программного обеспечения на целевые АРМ согласно заданному списку.

Название класса в Foreman: installpackages.

Принцип работы:

- Списком задаются пакеты, которые должны быть установлены на целевых АРМ.

- При применении настроек отсутствующие пакеты будут установлены из подключенных репозиториев, проверка установленных приложений на соответствие списку будет осуществляться каждый интервал применения классов Puppet.

Перечень параметров:

packages_list - массив, содержащий имена пакетов, необходимых для установки. Принимает значения в формате: ["package1", "package2", "package3"]

Пример использования:

в web интерфейсе Foreman задается packages_list ["htop", "gotop"]

Результат выполнения примера:

1. На целевых АРМ производится обновление кэша apt-get update;
2. На целевых АРМ устанавливаются пакеты из $packages_list командой apt-get install $packages_list;
3. При невозможности установки заданных в массиве пакетов или не верном названии пакета, модуль не установит пакеты и выдаст ошибку;
4. Установлены htop и gotop на целевые АРМ.

4.3.2. Удаление ПО

Модуль удаляет пакеты программного обеспечения на целевые АРМ по указанному перечню.

Функционал модуля:

Удаление программного обеспечения на целевых АРМ, согласно заданному списку.

Название класса в Foreman: removepackages.

Принцип работы:

- Списком задаются пакеты, подлежащие удалению на целевых АРМ.
- При применении настроек отсутствующие пакеты будут удалены, проверка приложений на соответствие списку будет осуществляться каждый интервал применения классов Puppet.

Перечень параметров:

packages_list - массив, содержащий имена пакетов, которые необходимые удалить. Принимает значения в формате: ["package1", "package2", "package3"].

Пример использования:

в web интерфейсе Foreman задается packages_list - ["apt-indicator", "openntpd","virtualbox-common","virtualbox-guest-common","virtualbox-guest-common-vboxguest","virtualbox-guest-common-vboxvideo","kernel-modules-virtualbox-addition-std-def","kernel-modules-virtualbox-addition-un-def"]

Результат выполнения примера:

Пакеты "apt-indicator", "openntpd","virtualbox-common","virtualbox-guest-common","virtualbox-guest-common-vboxguest","virtualbox-guest-common-vboxvideo","kernel-modules-virtualbox-addition-std-def","kernel-modules-virtualbox-addition-un-def" будут удалены на целевых АРМ

4.4. Управление точным временем

Модуль позволяет задавать на целевых АРМ часовой пояс ОС и сервера источники точного времени.

Функционал модуля:

- Устанавливает необходимый часовой пояс в ОС;
- Задает перечень серверов источников точного времени;
- Устанавливает необходимые пакеты сервисов;

Название класса в Foreman: tz_chrony

Поддерживаемые ОС: Альт, Альт СП.

Принцип работы:

Установка часового пояса производится путем создания симлинка файла необходимой зоны, содержащегося в каталоге /usr/share/zoneinfo к /etc/locatime. Источники точного времени задаются в конфигурационном файле сервиса chrony – chrony.conf.

Перечень параметров:

1. **timezone** - Часовой пояс в формате tzdata. Строковая переменная. Значение по умолчанию: 'Asia/Novosibirsk'. Полный список мировых локаций (с координатной привязкой), с которыми проассоциированы мировые часовые пояса, можно найти в файле zone.tab, который обычно находится в директории tzdata: /usr/share/zoneinfo/zone.tab.
2. **time_servers** - Перечень серверов источников точного времени, массив. Если передан пустой массив, значения не применятся на систему.

Пример использования:

```
- 10.10.1.2
- 10.10.28.34
- ntp.ix.ru
- ntp.sstf.nsk.ru
```

или

```
["10.10.1.2","10.10.28.34", "ntp.ix.ru", "ntp.sstf.nsk.ru"]
```

Результат выполнения примера:

Проверить текущий часовой пояс на целевом АРМ: timedatectl status | grep «Time zone»

Проверить сервера точного времени на целевом АРМ:

для chrony – mcedit /etc/chrony.conf, задано значение server <time_server> iburst

Для ntp – mcedit /etc/ntp.conf

4.5. Перезапуск управляемых клиентов

Перезапуск сервера — это разовая операция, которая не может быть корректно реализована средствами Puppet. Поэтому для реализации задачи используются возможности плагина Remote Execution.

Для выполнения перезапуска или выключения сервера нужно выполнить следующие действия:

- На вкладке Узлы – Все узлы выбираем нужные нам узлы;
- По кнопке Действия выбираем Schedule Remote Job;
- Во вновь открывшемся окне выбираем в категории: **Power**. В шаблоне задания: **Power Action - SSH Default;**
- Выбираем значения параметра action:
- **restart** – перезагрузка сервера в течение минуты, после получения команды.
- **shutdown** – немедленное выключение.
- Выбираем расписание запуска: **немедленно** (по умолчанию), в указанное время или повторять регулярно;
- После выбора всех параметров выбираем **Применить**.

* Внимание! Операция выключения и перезагрузки не успевают возвратить результат выполнения операции, поэтому сообщение об ошибке не несет информации. Контролировать выполнение операции можно с помощью команды ping.

4.6. Управление настройками DNS

Модуль позволяет провести настройки параметров DNS целевого АРМ.

Функционал модуля:

- Конфигурирование DNS целевого АРМ;
- Проверка успешности разрешения имен.

Название класса в Foreman: manage_resolv_conf.

Поддерживаемые ОС: Альт, Альт СП, Astra Linux, РЕД ОС.

Принцип работы модуля:

В начале работы модуль определяет используемый на целевом АРМ дистрибутив, если дистрибутив не определен работа модуля завершается с ошибкой. При поддерживаемом дистрибутиве модуль осуществляет настройку DNS согласно заданным параметрам. В каждом дистрибутиве свой подход к настройке сети и DNS.

Для ОС семейства Альт редактируется файл resolv.conf и выполняется /sbin/resolvconf -u;

Для Astra редактируется файл: resolv.conf и перезапускает службу networking;

Для РЕД ОС: редактируется файл /etc/NetworkManager/conf.d/dns-servers.conf и перезапускает службу NetworkManager (https://wiki.archlinux.org/title/NetworkManager).

Тестирование проводится следующим образом: с помощью команды /usr/bin/getent hosts осуществляется попытка разрешить имя test_host в IP-адрес. Если операция осуществляется успешно код выхода 0, и сообщение об ошибке не будет. В противном случае применение политики закончится не удачно, в интерфейсе Foreman отобразится сообщение об ошибке. Таким образом если задать в переменной test_host не существующее имя хоста и IP-адреса DNS-серверов указаны корректно, выполнение команды завершится не удачно. Поэтому нужно указывать корректный узел.

Перечень параметров:

dns_servers - перечень DNS-серверов, массив строк. Значение по умолчанию: пусто.

search_domains - перечень доменов поиска, массив строк. Значение по умолчанию: пусто.

test_host - имя узла на примере которого тестируется работы DNS. Если параметр не задан тестирование не проводится.

4.7. Удаленный запуск команд

Модуль позволяет производить выполнение команд на целевом узле.

Функционал модуля:

- Модуль позволяет удалённо отсылать команды или скрипты на удаленный АРМ или группу АРМ

Название класса в Foreman: Remote execution.

Поддерживаемые ОС: Альт, Альт СП, Astra Linux, РЕД ОС.

Принцип работы модуля:

- На вкладке **Узлы – Все узлы** указываем узлы;
- По кнопке **Действия** выбираем **Schedule Remote Job**;
- Во вновь открывшемся окне выбираем в категорию: **Command**s. В шаблоне задания: **Run Command - SSH Default;**
- В разделе **Command** указываем **команды**, которые необходимо выполнить на удаленном узле. Команды, можно, указывать построчно или в формате bash-скрипта;
- **Выбираем расписание запуска**: немедленно (по умолчанию), в указанное время или повторять регулярно;
- После выбора всех параметров выбираем **Применить**;
- На вновь открывшемся окне изучаем **результаты выполнения команды**.

4.8. Подключение нового клиента

Модуль позволяет удаленно подключить к системе управления нового клиента.

Функционал модуля:

- Модуль позволяет удалённо подключить АРМ к системе управления конфигурациями

Поддерживаемые ОС: Альт, Альт СП, Astra Linux, РЕД ОС.

Название класса в Foreman: Remote execution.

Принцип работы модуля:

Обязательным условием для удаленного подключения является наличие установленного открытого ключа SSH сервера Foreman для пользователя, от имени которого проводится настройка. Если пользователь не root, то он должен иметь право повысить свои привилегии в системе с помощью sudo без ввода пароля.

Удаленное подключение нового сервера к системе управления производится следующим образом:

- На вкладке **Узлы – Все узлы** нажимаем кнопку **Create Host;**
- Во вновь открывшемся окне **указываем имя узла**, **окружение – production**, выбираем мастер сервер и центр сертификации из списка;
- На вкладке **Операционная система** выбираем **архитектуру**, **Операционную систему** и снимаем галку возле пункта **Сборка - Разрешить подготовку узла;**
- На вкладке **Интерфейсы** жмем кнопку **Изменить** и указываем **MAC-адрес, Домен** и **Адрес IPv4** нового узла;
- Жмем кнопку **Применить**. Если какой-то параметр заполнен не верно система сообщит об ошибке;
- Повторно переходим на вкладку **Узлы – Все узлы** и выбираем вновь созданный узел;
- По кнопке **Действия** выбираем **Schedule Remote Job;**
- Во вновь открывшемся окне выбираем в категорию: **Puppet**. В шаблоне задания: **Puppet Agent Install - SSH Default;**
- Выбираем расписание запуска: немедленно (по умолчанию);
- После выбора всех параметров выбираем **Применить;**
- На вновь открывшемся окне изучаем результаты выполнения команды;
- Через некоторое время узел должен передать факты на систему, и они должны появиться в свойствах узла в Foreman.

4.9. Инициация применения политик со стороны сервера

Функционал модуля:

Модуль позволяет инициировать применение политик системы управления со стороны сервера.

Поддерживаемые ОС: Альт, Альт СП, Astra Linux, РЕД ОС.

Название класса в Foreman: Remote execution.

Принцип работы модуля:

Агент puppet на стороне клиента регулярно (в соответствии с настройками) проводит обращения к серверу. Если требуется немедленно получить политики находясь на клиенте можно выполнить команду:

```
puppet agent -t
```

Для инициации применения политик со стороны сервера можно воспользоваться возможностями плагина Remote Execution.

- На вкладке **Узлы – Все узлы** выбираем нужные нам узлы;
- По кнопке **Действия** выбираем **Schedule Remote Job;**
- Во вновь открывшемся окне выбираем в категорию: **Puppet**. В шаблоне задания: **Puppet Run Once - SSH Default;**
- Выбираем расписание запуска: немедленно (по умолчанию);
- После выбора всех параметров выбираем. **Применить;**
- На вновь открывшемся окне изучаем результаты выполнения команды.

4.10. Управление прокси сервером

Модуль позволяет произвести задать системный прокси сервера для целевых АРМ.

Функционал модуля:

- Задание прокси-сервера для протокола HTTP;
- Задание прокси-сервера для протокола HTTPS;
- Задание прокси-сервера для протокола FTP;
- Задание прокси-сервера для протокола SOCKS;
- Перечень узлов исключений.

Поддерживаемые ОС: Альт, Альт СП.

Название класса в Foreman: proxy_manage.

Принцип работы модуля:

Заданные настройки прокси сервера в модуле вносятся в системные переменные окружения и в настройки Dconf. Различные приложения берут настройки из этих двух источников. Например, при выставлении параметра в браузере Firefox **использовать системные настройки**, браузер использует системные переменные, в то время как Chromium использует настройки определенные в Dconf.

- Модуль задает и удерживает состояние настроек прокси в файле /etc/sysconfig/network;
- Модуль задает и удерживает состояние настроек прокси в разделе /system/proxy системы Dconf;
- Модуль блокирует возможность изменения настроек через Dconf;
- Системные настройки не применяются к текущему сеансу пользователя, для применения настроек пользователь должен войти в новый сеанс. Применение настройки Dconf не требует выхода из системы.

Перечень параметров:

Настройка (переменная)	**Тип**	**Значение**	**Описание**
if_proxy_manage	*логическая*	*True*	Модуль управляет настройками прокси на АРМ.
		False (по умолчанию)	Модуль не управляет настройками, значение переменных подмодуля не принимаются во внимание
if_same_proxy_ftp_https	*логическая*	*True*	Использовать настройки прокси протокола HTTP для протоколов HTTPS, FTP
		False (по умолчанию)	Настройки прокси для протоколов HTTPS и FTP задаются отдельно
dconf_db	*строковая*	*puppet (по умолчанию)*	каталог в */etc/dconf/db/*, в котором будет храниться конфигурация Dconf
proxy_host	*строковая*		DNS-имя или IP-адрес прокси сервера для протокола HTTP.
proxy_port	*целочисленная*	*8080 (по умолчанию)*	Порт прокси-сервера, числовое значение от 1 до 65535 для протокола HTTP.
https_proxy_host	*строковая*		DNS-имя или IP-адрес прокси сервера для протокола HTTPS.
https_proxy_port	*целочисленная*	*8080 (по умолчанию*	Порт прокси-сервера, числовое значение от 1 до 65535 для протокола HTTPS.
ftp_proxy_host	*строковая*		DNS-имя или IP-адрес прокси сервера для протокола FTP.
ftp_proxy_port	*целочисленная*	*8080 (по умолчанию*	Порт прокси-сервера, числовое значение от 1 до 65535 для протокола FTP.
socks_proxy_host	*строковая*		DNS-имя или IP-адрес прокси сервера для протокола SOCKS.
socks_proxy_port	*целочисленная*	*8080 (по умолчанию*	Порт прокси-сервера, числовое значение от 1 до 65535 для протокола SOCKS.

Настройка (переменная)	Тип	Значение	Описание
proxy_bypass	*строковая*		Переменная содержит перечень узлов, для которых настройки прокси сервера не применяются. Значениями могут быть имена узлов, доменов (например, *.foo.com), IP-адреса (как IPv4, так и IPv6), а также сетевые адреса с маской подсети (например, 192.168.0.0/24). Пример описания: ["localhost","127.0.0.0/8","::1","10.10.0.0/16","hp01.arm.loc",".arm.loc",".nso.loc"]

4.11. Управление настройками браузеров Chromium и Chromium-gost

Модуль управляет настройками браузера Chromium и Chromium-gost.

Функционал модуля:

- Управление списком доверенных сайтов;
- Управление версиями SSL и TLS;
- Установка запрета пользователю на установку расширений;
- Установка расширений согласно списку;
- Удаление расширений согласно списку.

Поддерживаемые ОС: Альт, Альт СП.

Название класса в Foreman: chromium_manage.

Принцип работы модуля:

Модуль формирует из заданных настроек файл на целевой системе /etc/${browser}/policies/managed и удерживает его в заданном состоянии. Применение файла политик можно проконтролировать на целевом АРМ пройдя по адресу: chrome://policy

Перечень параметров:

Настройка (переменная)	Тип	Значение	Описание
if_browser_manage	*логическая*	*True*	Модуль управляет настройками браузера на АРМ.
		False (по умолчанию)	Модуль не управляет настройками, настройки возвращаются к настройкам по умолчанию.
browser	*строковая*	*chromium*	Модуль управляет настройками Chromium

		chromium-gost (по умолчанию)	Модуль управляет настройками Chromium-gost
if_browser_install	*логическая*	*true*	Модуль производит установку управляемого браузера при отсутствии его на целевой ОС.
		False (по умочанию)	Модуль не производит действий по установке браузеров на целевой ОС.
auth_server	*строковая*	**.nso.ru,*.nso.loc,*.arm.loc (по умолчанию)*	Переменная указывает перечень доверенных сайтов. Пример значения: *.nso.ru,*.nso.loc,*.arm.loc.
ssl_version_min	*строковая*		Минимальная версия протокола TLS, поддерживаемая браузером. Возможные значения: tls1, tls1.1 или tls1.2.
store_mode	*строковая*	*allowed*	Пользователю разрешено устанавливать расширения
		removed	Пользователю запрещено устанавливать расширения. Существующие расширения продолжат работу.
		blocked	Пользователю запрещено устанавливать расширения. Существующие расширения будут удалены.
install_extension	*хэш-таблица*		В переменной указан перечень расширений и действий над ними. Описание данной переменной с примерами приведено далее.

install_extension - Переменная типа хэш-таблица, в которой указан перечень расширений и действий над ними. Хэш-таблица состоит из следующих параметров:

- **ID расширения.** Узнать ID можно на странице установки расширения в Интернет-магазин Chrome. Длина каждого идентификатора – 32 символа. Подробнее можно ознакомиться на портале разработчика chromium: https://support.google.com/chrome/a/answer/7517525?hl=ru
- **installation_mode** - действия с расширением. Возможные действия: allowed, removed, blocked, force_installed или normal_installed. Подробнее можно ознакомиться на портале разработчика Chromium https://support.google.com/chrome/a/answer/9867568?hl=ru;
  - allowed - пользователи могут установить это расширение;
  - blocked - пользователи не могут установить это расширение;
  - removed - (для Chromium 75 и более поздних версий) пользователи не могут установить это расширение. Если пользователи установили это расширение ранее, оно будет удалено;

- force_installed - расширение устанавливается в браузеры пользователей автоматически. Пользователи не могут его удалить. В этом режиме также необходимо указать ссылку для скачивания расширения (параметр update_url);
- normal_installed - расширение устанавливается в браузеры пользователей автоматически. Пользователи могут его отключить. В этом режиме также необходимо указать ссылку для скачивания расширения (параметр update_url);
- update_url - URL для скачивания расширения при его установке. Требуется только для установки расширения. Если не указан, расширение не будет установлено. В большинство случаев равно: update_url: https://clients2.google.com/service/update2/crx.

Пример использования переменной install_extension в формате YAML:

```
cjpalhdlnbpafiamejdnhcphjbkeiagm:
  installation_mode: force_installed
  update_url: https://clients2.google.com/service/update2/crx
gjceecgpmolmpdeidmfehcfepdfmmffl:
  installation_mode: blocked
hdokiejnpimakedhajhdlcegeplioahd:
  installation_mode: removed
```

Пример использования переменной install_extension в формате JSON:

```
"cjpalhdlnbpafiamejdnhcphjbkeiagm": {
     "installation_mode": "force_installed",
     "update_url": "https://clients2.google.com/service/update2/crx"
   },
"gjceecgpmolmpdeidmfehcfepdfmmffl": {
     "installation_mode": "blocked"
   },
"hdokiejnpimakedhajhdlcegeplioahd": {
     "installation_mode": "removed"
   }
```

Результат выполнения примера :

1. Установится расширение uBlock Origin. Расширение загрузится с серверов Google.
2. Расширение Lingvo Translator+ будет заблокировано для установки и использования.
3. Расширение LastPass: Free Password Manager будет удалено, и установка его будет запрещена.

4.12. Управление настройками браузера Firefox

Модуль управляет настройками браузера Firefox.

Функционал модуля:

- Управление списком доверенных сайтов;
- Управление версиями SSL и TLS;
- Установка запрета пользователю на установку расширений;
- Установка расширений согласно списку;
- Удаление расширений согласно списку.

Поддерживаемые ОС: Альт, Альт СП.

Название класса в Foreman: firefox_manage.

Принцип работы модуля:

Модуль формирует из заданных настроек файл /etc/firefox/policies/policies.json для целевого АРМ и удерживает его в заданном состоянии. Применение файла политик можно проконтролировать на целевом АРМ открыв страницу в браузере: about:policies.

Перечень параметров:

Настройка (переменная)	**Тип**	**Значение**	**Описание**
if_browser_manage	*логическая*	*True*	Модуль управляет настройками браузера на АРМ.
		False (по умолчанию)	Модуль не управляет настройками, при этом настройки обнуляются.
browser	*строковая*	*firefox*	Модуль управляет настройками firefox
		firefox-esr (по умолчанию)	Модуль управляет настройками firefox-esr
if_fox_install	*логическая*	*true*	Модуль производит установку управляемого браузера при отсутствии его на целевой ОС.
		False (по умочанию)	Модуль не производит действий по установке браузеров на целевой ОС.
auth_server	*строковая*	**.nso.ru,*.nso.loc,*.arm.loc (по умолчанию)*	Переменная указывает перечень доверенных сайтов. Пример значения: *.nso.ru,*.nso.loc,*.arm.loc.
ssl_version_min	*строковая*		Минимальная версия протокола TLS, поддерживаемая браузером. Возможные значения: tls1, tls1.1, tls1.2 или tls1.3.
ssl_version_max	*строковая*		Максимальная версия протокола TLS, поддерживаемая браузером. Возможные значения: tls1, tls1.1, tls1.2 или tls1.3.
default_extension_settings	*хэш-таблица*		Параметры установки расширений по умолчанию. Более детальное описание данной переменной приведено далее.

Настройка (переменная)	Тип	Значение	Описание
install_extension	*хэш-таблица*		В переменной указан перечень расширений и действий над ними. Более детальное описание данной переменной приведено далее.

default_extension_settings - параметры установки расширений по умолчанию:

- **blocked_install_message** - сообщение выводимое при установке расширения;
- **install_sources** - перечень источников, с которых разрешена установка расширений. В настоящий момент браузер выдает ошибку при установке этой переменной, несмотря на то что браузером заявлена его поддержка. Значение переменной не применяется к браузеру;
- **installation_mode** - действия с расширением. Поддерживаются только значения: allowed и blocked;
- **allowed_types** - перечень допустимых типов для установки. Возможные значения: "extension", "theme", "dictionary", "locale". В настоящий момент значение переменной не применяется к браузеру.

install_extension - Переменная типа хэш-таблица, в которой указан перечень расширений и действий над ними. Имеет следующие параметры:

- **Имя расширения.** Узнать ID или имя можно на странице **about:support** (если оно уже установлено) или с помощью расширения AMO - https://github.com/mkaply/queryamoid/releases/ . Подробнее об установке расширения можно узнать https://github.com/mozilla/policy-templates#extensionsettings;
- **installation_mode** - действия с расширением. Далее дано описание возможных действий.. Подробнее про режимы работы - https://github.com/mozilla/policy-templates#extensionsettings;
    - allowed - пользователи могут установить это расширение;
    - blocked - пользователи не могут установить это расширение;
    - force_installed - расширение устанавливается в браузеры пользователей автоматически. Пользователи не могут его удалить. В этом режиме также необходимо указать ссылку для скачивания расширения (параметр install_url). Не может быть указан для параметра по умолчанию;
    - normal_installed - расширение устанавливается в браузеры пользователей автоматически. Пользователи могут его отключить. В этом режиме также необходимо указать ссылку для скачивания расширения (параметр install_url). Не может быть указан для параметра по умолчанию;

- **install_url** - URL для скачивания расширения при его установке. Если параметр не указан, расширение не будет установлено. Не может быть указан для параметра по умолчанию.

Пример использования переменной default_extension_settings:

```
blocked_install_message: Ваш администратор запретил установку расширений.
installation_mode: blocked
```

или в формате JSON:

```
{
  "blocked_install_message": "Ваш администратор запретил установку расширений.",
  "installation_mode": "blocked",
}
```

Результат выполнения примера:

Пример заблокирует установку любых расширений, кроме тех, что указаны явно в параметре installation_mode.

Пример использования переменной **install_extension**:

```
{
  "https-everywhere@eff.org": {
    "installation_mode": "force_installed",
    "install_url": "https://addons.mozilla.org/firefox/downloads/file/3574076/latest.xpi"
  },
  "bing@search.mozilla.org": {
    "installation_mode": "blocked",
  },
  "mailru@search.mozilla.org": {
    "installation_mode": "blocked"
  },
  "uBlock0@raymondhill.net": {
    "installation_mode": "force_installed",
    "install_url": "https://addons.mozilla.org/firefox/downloads/latest/ublock-origin/latest.xpi"
  }
}
```

Результат выполнения примера:

1. Установится расширение https-everywhere;
2. Строка поиска Bing будет отключена;
3. Строка поиска Mail.ru будет отключена;
4. Расширение uBlock Origin будет установлено.

4.13. Управление корневыми сертификатами

Модуль позволяет устанавливать и обновлять сертификаты в общесистемном хранилище доверенных сертификатов.

Функционал модуля:

- Установка сертификатов в общесистемное хранилище доверенных сертификатов;
- Обновление сертификатов в общесистемном хранилище доверенных сертификатов.

Поддерживаемые ОС: Альт, Альт СП.

Название класса в Foreman: cert_manage.

Принцип работы модуля:

Модуль удерживает список установленных сертификатов согласно заданному перечню по именам файлов. Сертификаты располагаются в каталоге на сервере Foreman /etc/puppet/code/environments/production/modules/cert_manage/files. Если файлы сертификатов отсутствуют на сервере при применении политики puppet, будет выдана ошибка.

Перечень параметров:

name_cert массив с именами файлов

Пример использования:

```
name_cert ["minsvyaz.cer","qualifiedgisca2020.cer","qualifiedgiscav3.cer"]
```

Результат выполнения примера:

На конечный АРМ установлены сертификаты minsvyaz.cer","qualifiedgisca2020.cer","qualifiedgiscav3.cer"

4.14. Управление блокировкой рабочего стола

Модуль позволяет произвести настройку параметров хранителя экрана и блокировки рабочего стола графической среды Mate.

Функционал модуля:

- Управление блокировкой сеанса;
- Задание времени бездействия АРМ;
- Активация хранителя экрана;
- Выбор хранителя экрана.

Поддерживаемые ОС: Альт, Альт СП.

Название класса в Foreman: screen_lock_manage.

Принцип работы:

Настройка параметров проводится через низкоуровневую систему управления настройками - Dconf. Подробнее с возможностями и принципом работы Dconf, а также с редактором dconf-editor можно ознакомиться на сайте разработчика ОС Альт https://www.altlinux.org/Dconf. В Mate настройки хранителя собраны в приложении mate-screensaver-preference, которое является пользовательским интерфейсом к части настроек Dconf.

В Mate реализована следующая логика активации блокировки сеанса:

1. При бездействии N минут (параметр dconf - idle_delay) АРМ считается простаивающим, может быть включен хранитель экрана.
2. Через M минут (параметр dconf - lock-delay) нахождения в режиме простоя, может быть активирована блокировка сеанса пользователя.

Модуль производит настройку параметров в следующих ветках dconf:

- /org/mate/desktop/session/
- /org/mate/screensaver/

Модуль блокирует возможность изменения настроек посредством файла, который содержит перечень удерживаемых настроек Dconf. Файл 50-screenlock.erb располагается в каталоге /etc/dconf/db/<$dconf_db>.d/locks/. Значение lock-delay также задано в файле 50-screenlock.erb равным 0 и не управляется средствами модуля. Т. е. блокировка сеанса осуществляется одновременно с переходом АРМ в состояние простоя.

Перечень параметров:

Настройка (переменная)	**Тип**	**Значение**	**Описание**
if_screenlock_manage	*логическая*	*True*	Модуль управляет настройками хранителя экрана. Все заданные переменные будут применены.
		False (по умолчанию)	Модуль не управляет настройками, применяются значения по умолчанию, заданные в Dconf, значение переменных подмодуля не принимаются во внимание.
dconf_db	*строковая*	*Puppet (по умолчанию)*	Переменная задает имя каталога, в котором будет хранится конфигурация Dconf, каталог будет размещен в /etc/dconf/db/
if_lock_enabled	*логическая*	*True (по умолчанию)*	Блокировать сеанс, когда запущен хранитель экрана

		false	Блокировка сеанса отключена
idle_delay	*целочисленная*	*5 (по умолчанию, допустимые 1-120)*	Продолжительность в минутах простоя компьютера перед включением хранителя экрана
if_screensaver_enable	*логическая*	*true (по умолчанию)*	Хранитель экрана включен
		false	Хранитель экрана выключен
mode	*строковая*	*blank-only (по умолчанию)*	черный экран, значение переменной themes игнорируется
		single	использовать одну тему, указанную в переменной themes
		random	выбрать случайную тему из переменной themes
themes	*массив*	*screensavers-cosmos-slideshow (по умолчанию)*	Перечень тем для хранителя. Модуль работает с темами, которые расположены на АРМ в каталоге /usr/share/applications/screensavers/. Пустое значение не допустимо, модуль выдаст при работе ошибку. Если указать несуществующий хранитель экрана, то будет черный экран. Перечень, базовых скринсейверов: screensavers-cosmos-slideshow, screensavers-footlogo-floaters, screensavers-gnomelogo-floaters, screensavers-personal-slideshow, screensavers-popsquares.

Пример использования

```
if_screenlock_manage - true
dconf_db - Puppet
if_lock_enabled - false
```

idle_delay - 1
if_screensaver_enable - true
mode - blank-only

Результат выполнения примера

при таких параметрах через минуту включится хранитель экрана, но экран не будет заблокирован

4.15. Управление сетевыми дисками

Модуль позволяет произвести настройку параметров хранителя экрана и блокировки рабочего стола графической среды Mate.

Функционал модуля:

- Управление блокировкой сеанса;
- Задание времени бездействия АРМ;
- Активация хранителя экрана;
- Выбор хранителя экрана.

Поддерживаемые ОС: Альт, Альт СП.

Название класса в Foreman: net_share_mount.

Принцип работы:

Подключение разделяемых файловых ресурсов осуществляется при помощи модуля pam_mount, который позволяет при входе доменного пользователя автоматически монтировать ресурс без повторного ввода пароля. Общее описание настроек, осуществляемых модулем на сайте разработчика: https://www.altlinux.org/SSSD/AD#Через_pam_mount

Действия, производимые модулем:

- Проверяется является ли АРМ членом домена. Если АРМ не входит в домен, то модуль прекращает свою работу.
- Проверяется, поддерживаема ли ОС этим модулем. Если ОС не входит в поддерживаемые, то модуль прекращает свою работу.
- Модуль производит установку пакетов 'cifs-utils', 'pam_mount' при их отсутствии.
- Модуль производит настройку файлов /etc/pam.d/system-auth-sss, /etc/security/pam_mount.conf.xml.

Перечень параметров:

Настройка (переменная)	**Тип**	**Значение**	**Описание**

if_net_share_manage	*логическая*		True - модуль настраивает конфигурационные файлы и удерживает их состояние: /etc/security/pam_mount.conf.xml – файл параметров монтирования сетевых дисков и /etc/pam.d/system-auth-sss файл параметров авторизации пользователя. False -Модуль не управляет конфигурационными файлами. Существующие файлы остаются без изменений, но более модуль не удерживает их состояние.
shares	*строковый массив*		Переменная хэш-таблица, описывает перечень монтируемых сетевых дисков. Если в параметр передан пустой массив сетевые папки будут удалены из перечня монтирования.

shares – переменная хэш-таблица, описывает перечень монтируемых сетевых дисков. Если в параметр передан пустой массив сетевые папки будут удалены из перечня монтирования.

Имеет следующие параметры:

- server - сервер, с которого производится монтирование диска.
- path - название каталога для монтирования с сервера.
- mountpoint - точка монтирования сетевого диска на клиенте.

Пример использования:

```
[
  {
    "server": "dc01.testlab.smb",
    "path": "sysvol",
    "mountpoint": "~/sysvol"
  },
```

```
  {
    "server": "dc01.testlab.smb",
    "path": "netlogon",
    "mountpoint": "~/netlogon"
  },
  {
    "server": "dc01.testlab.smb",
    "path": "User Data",
    "mountpoint": "~/userData"
  }
]
```

Результат выполнения примера:

По итогам работы модуля у пользователя будут смонтированы следующие диски:

- sysvol с сервера dc01.testlab.smb в каталог пользователя ~/sysvol,
- netlogon с сервера dc01.testlab.smb в каталог пользователя ~/netlogon,
- userdata с сервера dc01.testlab.smb в каталог пользователя ~/userData.

4.16. Управление сетевыми принтерами

Модуль управляет принтерами на целевых АРМ.

Функционал модуля:

- Установка сетевого принтера;
- Установка драйвера сетевого принтера;
- Назначение принтера по умолчанию;
- Удаление сетевого принтера.

Поддерживаемые ОС: Альт, Альт СП.

Название класса в Foreman: printer_manage.

Перечень параметров:

Управление принтерами осуществляется переменной **printers_list** - типа массив, элементами которой являются хэш-таблица, с параметрами принтера и действий с ним. Хэш-таблица имеет следующие поля:

- **printer_name** - название принтера, как он будет отображаться пользователю.
- **action** - действия с принтером. Возможные значения:
- **remove** – удалить принтер из системы;
- **install** - установить принтер в систему.

Для удаления принтера достаточно указать его название. Для установки необходимо указать переменные URL и driver, их описание ниже.

- **URL** - протокол и адрес принтера для подключения. Доступные протоколы можно получить в документации к принтеру. Как правило подходят протоколы IPP и HTTP. Переменная обязательна для установки принтера.

Ниже приведен пример возможных протоколов и адресов:

```
http://hostname:631/ipp/
http://hostname:631/ipp/port1
ipp://hostname/ipp/
ipp://hostname/ipp/port1
lpd://hostname/queue
socket://hostname
socket://hostname:9100
```

- **driver** - Название драйвера: имя файла драйвера, без расширения ppd. Сам файл драйвера должен лежать в каталоге модуля files. Путь до каталога /etc/puppet/code/environments/production/modules/printer_manage/files сервера Foreman. Если поле отсутствует, используется драйвер по умолчанию everywhere. Перед установкой принтера файл драйвера копируется с сервера Foreman на локальную АРМ в каталог /usr/share/cups/model.
- **default** - логическая переменная, выбран принтер по умолчанию или нет.

  Если в массиве в нескольких местах выбран параметр default = true, то принтером по умолчанию станет последний принтер из массива с таким параметром.

Пример использования переменной printers_list:

```
[
  {
    "printer_name": "VersaLink",
    "action": "install",
    "URL": "http://prn-versa.arm.loc/ipp",
    "driver": "xrxB7030",
    "default": false
  },
    {
    "printer_name": "Flor4prn",
    "action": "install",
    "URL": "ipp://10.10.3.13/ipp",
    "default": true
```

```
  },
  {
    "printer_name": "MFU",
    "action": "remove"
  }
]
```

или в таком формате:

```
- printer_name: "VersaLink",
  action: "install",
  URL: "http://prn-versa.arm.loc/ipp",
  driver: "xrxB7030",
  default: false
- printer_name: "Flor4prn",
  action: "install",
  URL: "ipp://10.10.3.13/ipp",
  driver: "everywhere",
  default: true
- printer_name: "Versa",
  action: "remove"
```

Результат выполнения примера:

1. Установлен принтер с названием VersaLink, доступный по сети по адресу: prn-versa.arm.loc. Для него будет установлен драйвер от принтера модели Xerox VersaLink B7030.
2. Установлен принтер с названием Flor4prn, доступный по сети по адресу: 10.10.3.13. Для него будет установлен драйвер по умолчанию.
3. Принтер с названием Versa будет удален из системы.

В связи с особенностями работы системы CUPS и доступных для нее команд, нельзя просто поменять параметры принтера, например сменить драйвер. Любая перенастройка должна проходить через повторную установку/удаление принтера.

4.17. Управление ярлыками на рабочем столе

Модуль управляет ярлыками на рабочем столе пользователя.

Функционал модуля:

- Создание ярлыков для рабочего стола;
- Удержание заданной конфигурации.

Поддерживаемые ОС: Альт, Альт СП.

Название класса в Foreman: shortcuts.

Принцип работы:

- Модуль создает ярлыки по заданным параметрам и размещает в каталоге /usr/share/Desktop.
- Модуль обновляет ярлыки только при входе пользователя в сеанс. Создается в каталоге /etc/X11/profile.d/ скрипт shared_desktop_icons_sync.sh, который запускается при входе пользователя в сеанс и синхронизирует ярлыки каталога /usr/share/Desktop в пользовательский каталог рабочего стола.
- Модуль устанавливает пакет rsync (при его отсутствии в ОС) и удаляет пакет shared-desktop-icons (т. к. конфликтует с ним).
- Копирует иконки приложений из каталога на сервере Foreman puppet:///modules/shortcuts/icons в локальный каталог АРМ: /usr/share/icons/custom_icons. Скопированные иконки можно использовать в параметрах ярлыков.
- Удаленные ярлыки автоматически восстанавливаются на рабочем столе, если они есть в каталоге /usr/share/Desktop при следующем входе пользователя в систему.
- Для полного удаления ярлыка на рабочем столе, без его автовосстановления его нужно удалить с рабочего стола и из каталога /usr/share/Desktop. И отключить создание с помощью puppet.

Перечень параметров:

shorts_setting - переменная типа хэш-таблица, в которой описаны параметры создаваемых ярлыков. Формат хэш-таблицы имеет следующий вид:

Имя ярлыка - называние ярлыка на рабочем столе. Обязательный параметр.

type - тип ярлыка. Обязательный параметр. Допустимо одно из значений:

application – ярлык запускает приложение,

terminal – ярлык запускает приложение в терминале,

link – ярлык открывает заданный файл, веб-страницу или другую ссылку, в приложении заданном по умолчанию для данного типа в Mate.

source - командная строка или URL (зависит от типа ярлыка). Обязательный параметр.

icon - иконка для ярлыка. Не обязательный параметр. Абсолютный путь до ярлыка в файловой системе.

Пример использования переменной shorts_setting:

```
Портал совместной работы:
  type: application
  source: "/usr/bin/chromium-browser http://r7.arm.loc"
  icon: "/usr/share/icons/custom_icons/r7.png"
Правительство НСО:
  type: application
```

```
  source: "/usr/bin/firefox http://www.nso.ru/"
  icon: "/usr/share/icons/mate/48x48/apps/bluetooth.png"
Chromium Web Browser:
  type: application
  source: chromium %U
  icon: chromium
Yandex:
  type: link
  source: "http://yandex.ru"
htop:
  type: terminal
  source: "htop"
  icon: "htop"
```

Результат выполнения примера:

1. Ссылка на "Портал совместной работы" по адресу http://r7.arm.loc будет всегда открываться в браузере Chromium. Будет использована иконка, находящаяся на сервере Foreman и скопированная на локальную систему.
2. Ссылка на сайт правительства Новосибирской области всегда будет открываться в браузере Firefox, будет использована локальная иконка.
3. Браузер Chromium, со стандартным значком.
4. Ссылка на сайт Яндекса, со значком по умолчанию. Ссылка будет открываться в браузере по умолчанию.
5. Ярлык на консольное приложение htop, со стандартным значком.

По всем вопросам, связанным с СУ «Патриот» вы можете обратиться:

- Электронная почта patriot@galex.ru – круглосуточно;
- Телефон 8(3852)501680 (доб.1370) – 9-18, пн-пт, МСК +4;
- Телеграмм канал https://t.me/tpsupatriot.