

# Платформа «Простор»

## Руководство по установке

Версия 2.0<sup>бета</sup>

для Astra Linux Common Edition

**Лаборатория 50**

---

## Платформа «Простор»: Руководство по установке

Настоящий документ является руководством по установке и настройке платформы «Простор», а также необходимых для её функционирования компонентов Astra Linux Common Edition.

© ООО «Лаборатория 50», 2014-2018

Дата публикации: 30 октября 2018 г.

## Аннотация

Руководство содержит информацию и инструкции для автоматизированной установки платформы «Простор» в составе:

- Простор.Кластер;
- Простор.Управление;
- Простор.VDI;
- Простор.Хранение.

Предлагаем вам короткое наставление по использованию нашего руководства.

1. Сверьтесь с входными требованиями к аппаратной части и необходимому для установки программному обеспечению в разделе [«Общие сведения»](#).
2. Ознакомьтесь с общим алгоритмом установки, необходимыми данными и возможными вариантами разворачивания комплекса в разделе [«Планирование установки»](#).
3. Перед началом установки выполните предварительную настройку оборудования по разделу [«Подготовка целевых узлов»](#).
4. Подготовьте файлы конфигурации устанавливаемого кластера. Доступные настройки приведены в разделе [«Конфигурация кластера»](#).
5. Выполните необходимые шаги для проверки полученной конфигурации кластера и запустите установку и настройку платформы «Простор» по инструкции раздела [«Установка кластера»](#).

Ваши замечания и пожелания по содержанию руководства и процедуре установки присылайте на почту [team@lab50.net](mailto:team@lab50.net).

*Команда «Лаборатории 50»*

## Содержание

История изменений .....	6
Принятые обозначения .....	7
1 Общие сведения .....	8
1.1 Назначение .....	8
1.1.1 Простор.Кластер .....	8
1.1.2 Простор.Управление .....	8
1.1.3 Простор.VDI .....	8
1.1.4 Простор.Хранение .....	8
1.2 Состав платформы .....	9
1.2.1 Простор.Кластер .....	9
1.2.2 Простор.Управление .....	9
1.2.3 Простор.VDI .....	10
1.2.4 Простор.Хранение .....	11
1.3 Требования к аппаратным средствам .....	11
1.4 Требования к программному обеспечению .....	12
2 Планирование установки .....	13
2.1 Проверка входных данных .....	13
2.2 Варианты установки .....	14
2.2.1 Интегрированная панель управления .....	14
2.2.2 Внешняя панель управления .....	15
3 Подготовка целевых узлов .....	17
3.1 Проверка IPMI .....	17
4 Установка системы управления конфигурацией .....	18
5 Конфигурация кластера .....	19
5.1 Общие сведения .....	19
5.2 Состав кластера .....	19
5.3 Сценарий установки .....	21
5.3.1 Запрос дополнительных данных .....	21
5.3.2 Роли .....	22
5.3.3 Типовые сценарии .....	24
5.4 Параметры .....	24
5.4.1 Базовые настройки .....	24
5.4.2 Репозитории установки .....	25
5.4.3 Настройки сети .....	25
5.4.4 Настройка синхронизации времени .....	27
5.4.5 Настройка системы управления кластером .....	27
5.4.6 Фенсинг .....	29
5.4.7 Служба управления .....	30
5.4.8 Веб-сервер .....	30
5.4.9 Веб-терминал .....	31

---

5.4.10	Панель управления .....	31
5.4.11	Настройка параметров Гластера .....	31
5.4.12	Настройка монтирования Гластер-тома .....	33
5.4.13	Виртуализация .....	34
5.5	Проверка файлов конфигурации .....	34
6	Установка кластера .....	35
6.1	Предварительная проверка .....	35
6.2	Установка .....	35
7	Сообщения системному программисту .....	37
Приложение 1	Формат YAML .....	38
Приложение 2	Конфигурационные файлы .....	40
Приложение 3	Роли .....	42

## История изменений

2018-09-20      Добавлена аннотация.

## Принятые обозначения

В данном руководстве принят ряд обозначений.

В документе могут встречаться следующие типы замечаний:

### Примечание

Примечание или полезная подсказка.

### Важно

Существенные нюансы, которые вы должны принять во внимание, прежде чем начнёте какое-либо действие.

### Предупреждение

Важная информация о проблемах безопасности или риске потери данных.

В документе используются два варианта приглашения командного интерпретатора.

#### **\$ команда**

Команды, начинающиеся с приглашения \$, может выполнять любой пользователь, в том числе root.

#### **# команда**

Команды, начинающиеся с приглашения #, должен выполнять пользователь root. Но вы можете использовать утилиту **sudo** чтобы запускать их под другой учётной записью.

# 1 Общие сведения

## 1.1 Назначение

Платформа «Простор» предназначена для создания надёжной ИТ-инфраструктуры, обеспечивающей непрерывную доступность прикладных приложений и сервисов.

### 1.1.1 Простор.Кластер

Предназначен для обеспечения непрерывной доступности прикладных приложений и сервисов. При отказе любого сервиса или узла целиком кластер автоматически перезапустит сервис на работающем узле с минимальным прерыванием для клиентов кластера.

Простор.Кластер гарантирует целостность данных при переносе пользовательских сервисов между узлами одного кластера. В случае необходимости, для предотвращения порчи данных отказавшие узлы будут автоматически исключены из кластера. Узлы могут быть либо виртуальными машинами, либо работать на физическом оборудовании. Для достижения высокой готовности программно-аппаратного комплекса требуется специальная конфигурация аппаратных средств, обеспечивающая отказоустойчивость каждой из подсистем кластера.

### 1.1.2 Простор.Управление

Центральное звено управления компонентами платформы. Включает в себя: службу управления, веб-панель, интерфейс командной строки, программный REST-интерфейс.

Веб-панель управления предоставляет администраторам все основные функции по управлению «Простором»: контроль состояния и управление узлами, виртуальными машинами, кластерной файловой системой и другими ресурсами кластера высокой готовности. Архитектура панели спроектирована без единой точки отказа и не зависит от внешних СУБД или разделяемых хранилищ. Панель управления доступна при наличии хотя бы одного исправного узла.

### 1.1.3 Простор.VDI

Предоставляет следующие возможности:

- подключение к рабочим столам из Linux, Windows или через веб-интерфейс;
- бесшовная миграция – перенос с одного узла кластера на другой без останова гостевой операционной системы и отключения пользователей;
- создание мгновенных снимков состояния;
- подключение устройств USB в гостевую ОС;
- работа со специальными томами Г кластер и коммерческими СХД.

### 1.1.4 Простор.Хранение

Необходимо для создания отказоустойчивых хранилищ данных с общим доступом. В основе кластерной файловой системы лежит Gluster.



Gluster — это распределённая параллельная линейно-масштабируемая файловая система с возможностью защиты от сбоев. С помощью InfiniBand RDMA или TCP/IP Gluster может объединить хранилища данных, находящиеся на разных серверах, в одну параллельную сетевую файловую систему.

## 1.2 Состав платформы

### 1.2.1 Простор.Кластер

В основе Простор.Кластер лежит программное обеспечение Pacemaker и Corosync.

Corosync — это система группового обмена информацией для отказоустойчивых кластеров. Corosync обеспечивает синхронизацию заданного набора данных на узлах кластера. С помощью данного компонента реплицируется между узлами База Данных Кластера (CIB).

Pacemaker — это системный уровень системы управления кластером. Pacemaker состоит из нескольких ключевых компонентов.

*База данных кластера (CIB)* описывает конфигурацию кластера и текущего состояния всех ресурсов. Храниться в формате XML. Содержимое CIB автоматически синхронизируется по всему кластеру и используется для вычисления наилучшего состояния кластера и способов его достижения.

Состоянием кластера управляет Выбранный Координатор (Designated Controller — DC). В состав Pacemaker входит *диспетчер кластерных ресурсов* (Cluster Resource Manager — CRM). Pacemaker централизует все кластерные решения, выбирая один из экземпляров службы CRM в качестве главного. При отказе выбранного узла или процесса CRM на нем будет выбран другой.

Вычисленное наилучшее состояние кластера Выбранный Координатор воплощает путём передачи сообщений:

- на своем узле — *диспетчеру локальных ресурсов* (Local Resource Management — LRM);
- на других узлах — диспетчеру кластерных ресурсов, которые с свою очередь иницируют диспетчер локальных ресурсов.

Все узлы сообщают результаты своих операций Выбранному Координатору (DC). На основе ожидаемых и фактических результатов DC либо выполняет действия, необходимые для ожидания завершения предыдущей операции, либо прерывает обработку и на основе полученных результатов вычисляет новое наилучшее состояние кластера.

В некоторых случаях может потребоваться выключение узла для защиты общих данных или полного восстановления ресурсов. Для этого в состав Pacemaker входит служба фенсинга (STONITH).

### 1.2.2 Простор.Управление

Основной задачей Простор.Управление является предоставление операторам практического и надёжного интерфейса управления кластером при установке, настройке и эксплуатации.

Простор.Управление координирует взаимодействие между различными компонентами платформы: Простор.Кластер, Простор.VDI и Простор.Хранение.

Состоит из следующих компонентов:

- служба управления;
- графический интерфейс пользователя;
- интерфейс командной строки.

Служба управления является центральным звеном системы управления. Функционирует на каждом узле кластера. Служба подключается к используемым компонентам, например, защищённой виртуализации, посредством прикладным программных интерфейсов. Службы, запущенные на узлах кластера, синхронизируют свое состояние посредством программы Cegosync. Архитектура системы управления является полностью децентрализованной. При отказе службы управления на любом узле система управления в целом остаётся функциональной.

Взаимодействие со службой управления происходит через REST ППИ по протоколу HTTP. Интерфейс является открытым и может использоваться программами пользователя.

Интерфейсы пользователя (командный и графический) подключаются к службе управления по REST-интерфейсу.

### 1.2.3 Простор.VDI

Простор.VDI реализован на основе гипервизора KVM, входящего в состав операционной системы Astra Linux Common Edition. В состав Простор.VDI входит эмулятор оборудования (QEMU), обеспечивающий возможность запуска различных гостевых операционных систем. Каждая гостевая операционная система представляет собой отдельный процесс базовой операционной системы. Процесс KVM имеет изолированное от других гостевых процессов адресное пространство. Гипервизор использует аппаратные возможности для виртуализации процессора (Intel VT и AMD-V) и памяти (AMD NPT и Intel EPT).

QEMU представляет собой платформу виртуализации, которая позволяет эмулировать широкий спектр оборудования (включая диски, графические адаптеры, сетевые устройства). Любые запросы ввода/вывода, которые делает гостевая операционная система, перехватываются и направляются в пользовательский режим для эмулирования с помощью процесса QEMU.

В состав Простор.VDI входит система управления виртуализацией Libvirt. Данный компонент предназначен для упрощения операций над виртуальными машинами и предназначен для работы с различными гипервизорами, в т.ч. KVM.

Libvirt полностью берёт на себя взаимодействие с процессом QEMU и предоставляет командную оболочку virsh с универсальным набором операций над виртуальными машинами. В состав Простор.VDI входит служба Libvirtd, который управляет процессом и виртуальной машиной в целом через интерфейс управления QEMU.

## 1.2.4 Простор.Хранение

Компонент предназначен для создания общих отказоустойчивых хранилищ данных. В основе Простор.Хранения лежит файловая система GlusterFS.

GlusterFS – это распределённая параллельная линейно-масштабируемая файловая система с возможностью защиты от сбоев. С помощью InfiniBand RDMA или TCP/IP GlusterFS может объединить хранилища данных, находящиеся на разных серверах, в одну параллельную сетевую файловую систему.

Центральным звеном Глостера является служба управления `glusterd`. Функционирует на каждом узле кластера. После запуска и первоначальной настройки `glusterd`, из совокупности узлов кластера образуется доверенный пул хранения. Каждый узел пула контролирует элементарные блоки хранения – *кирпичи*. Кирпичи используются для построения логических элементов хранения – *томов*. Для каждого кирпича запускается процесс `glusterfsd`.

Комбинируя шаблоны организации кирпичей, Глостер предоставляет различные варианты избыточности и производительности: зеркалирование, чередование, с арбитром и без, и т.д.

При подключении клиента к тому Глостера, процесс `glusterfs` клиента связывается с процессом `glusterd` серверов. `Glusterd` отправляет конфигурацию тома и его кирпичей, с помощью которой процесс `glusterfs` клиента сможет напрямую работать с процессом `glusterfsd` каждого кирпича.

Процесс `glusterfs` состоит из стека *клиентских трансляторов*, задаваемого конфигурацией тома. Транслятор – это модуль Глостера, отвечающей за некоторую часть функциональности томов. Помимо клиентских трансляторов, существуют *серверные трансляторы*, работающие на стороне сервера.

## 1.3 Требования к аппаратным средствам

Для установки ПО Простор.Кластер с помощью системы управления конфигурацией необходим компьютер со следующими характеристиками:

1. Совместимость с ОС Линукс.
2. Объем оперативной памяти не менее 2 ГБ.
3. Монитор с разрешением как минимум 1280x1024.
4. Клавиатура.
5. Мышь или сенсорный экран.

Рекомендуемые требования к техническим характеристикам оборудования кластера высокой готовности:

1. Серверы на базе Intel/AMD совместимые с Astra Linux Common Edition версии 2.12.
2. Аппаратный массив RAID 6/10.
3. Оперативная память каждого сервера не менее 32 ГБ для работы с виртуальными машинами, 8 ГБ без VM.
4. У каждого сервера должно быть не менее 4 сетевых интерфейсов, обеспечивающих сеть 1 Гбит/с и аппаратную поддержку 802.1q (VLAN).
5. Каждый сервер должен быть оснащён двумя блоками питания.

6. 2 УРП (типа APC 7920) с удаленным администрированием и обеспечивающих отключение серверов от сети питания.
7. Серверы должны быть оснащены интерфейсом удалённого управления IPMI.
8. 4 управляемых коммутатора с 24 портами.
9. 2 источника бесперебойного питания.
10. КМВ с монитором, клавиатурой и мышью.

Минимальные требования к техническим характеристикам оборудования кластера высокой готовности:

1. Серверы на базе Intel/AMD совместимые с Astra Linux Common Edition версии 2.12.
2. Оперативная память каждого сервера должна быть не менее 16 ГБ.
3. У каждого сервера должно быть не менее двух сетевых интерфейсов, обеспечивающих сеть 1 Гбит/с.
4. Серверы должны быть оснащены интерфейсом удалённого управления IPMI.
5. 2 управляемых коммутатора с 24 портами.
6. КМВ с монитором, клавиатурой и мышью.

## 1.4 Требования к программному обеспечению

Система управления конфигурацией представляет собой набор сценариев и модулей для инструментария управления конфигурацией Ansible. Для корректной работы программного комплекса необходима версия Ansible 1.9.

Установку Простор.Кластер с помощью системы управления конфигурацией можно производить с любой Линукс-совместимой операционной системы с установленным ПО Ansible подходящей версии.

Для работы системы управления конфигурацией Ansible в Astra Linux Common Edition требуется следующий набор пакетов:

- openssh-client;
- python;
- python-crypto;
- python-httplib2;
- python-jinja2;
- python-paramiko;
- python-markupsafe;
- python-yaml.

## 2 Планирование установки

Данное руководство описывает порядок настройки кластера высокой готовности на базе платформы «Простор». Перед установкой необходимо выполнить все предварительные шаги по настройке аппаратной части кластера и установке операционной системы Astra Linux Common Edition на узлах.

Перед началом установки необходимо зафиксировать следующие входные данные:

- количество серверов в кластере;
- устройства фенсинга;
- параметры (адрес, доменное имя) используемых сетей;
- вариант установки ПО.

Автоматизированная установка происходит следующим образом. Администратор готовит целевую конфигурацию кластера, состоящую из:

- описание состава физической или виртуальной инфраструктуры;
- параметров кластера (включая входные данные);
- сценария установки.

Подготовленный сценарий последовательно интерпретируется утилитой Ansible. Ansible выполняет операции сценария на целевых машинах. Для подключения к целевому узлу и выполнения операций сценария используется протокол SSH. При необходимости передачи файлов между целевой и установочной машиной используется протокол SFTP.

Использование Ansible в качестве основы системы управления конфигурацией позволяет предъявлять минимальные требования к составу программного обеспечения на целевых узлах. Фактически требуется только сервер OpenSSH.

Процесс установки с помощью системы управления конфигурацией разбит на следующие основные шаги:

1. Настройка аппаратных средств (разд. 3).
2. Установка и настройка ОС Astra Linux Common Edition на серверы кластера.
3. Установка системы управления конфигурацией (разд. 4).
4. Подготовка (описание) конфигурации кластера (разд. 5).
5. Установка компонентов платформы с помощью системы управления конфигурацией (§ 6.2).

### 2.1 Проверка входных данных

Перед установкой Простор.Кластера все аппаратные средства (вычислительные, сетевые, фенсинг) должны быть предварительно установлены и настроены.

Список необходимых данных перед установкой кластера высокой готовности:

1. Устройства фенсинга:
  - 1) настройка статического IP-адреса управляемых распределителей питания (если используются);
  - 2) настройка статического IP-адреса интерфейса IPMI серверов кластера (если используются).
2. Сетевое оборудование:

- 1) адреса и параметры используемых сетей (как минимум сети данных и сети управления).
  - 2) использование распределенного агрегирования;
  - 3) настройка (R)STP.
3. Серверы (узлы) кластера:
- 1) сетевые имена узлов кластера;
  - 2) IP-адреса узлов кластера в сети данных и сети управления;
  - 3) учётная запись и пароль, которые будут использоваться для установки ПО.

## 2.2 Варианты установки

Платформа позволяет в значительных пределах варьировать состав и структуру устанавливаемых программных компонентов. Для упрощения задачи инсталляции система управления конфигурации включает два типовых варианта установки:

- с интегрированной панелью управления;
- с вынесенной панелью управления на отдельный узел.

### 2.2.1 Интегрированная панель управления

В этом варианте установки веб-панель управления функционирует на узлах кластера. Готовность обеспечивается на счёт одновременной работы экземпляров панели на всех узлах кластера. Пользователь подключается к панели по сетевому имени плавающего IP-адреса. При отказе узлов плавающий адрес автоматически переносится на работающий узел. В случае сбоя или отказа плавающего сетевого адреса панель управления автоматически переключится на доступный работающий узел. Структурная схема данного варианта установки показана на [рис. 2.1](#).

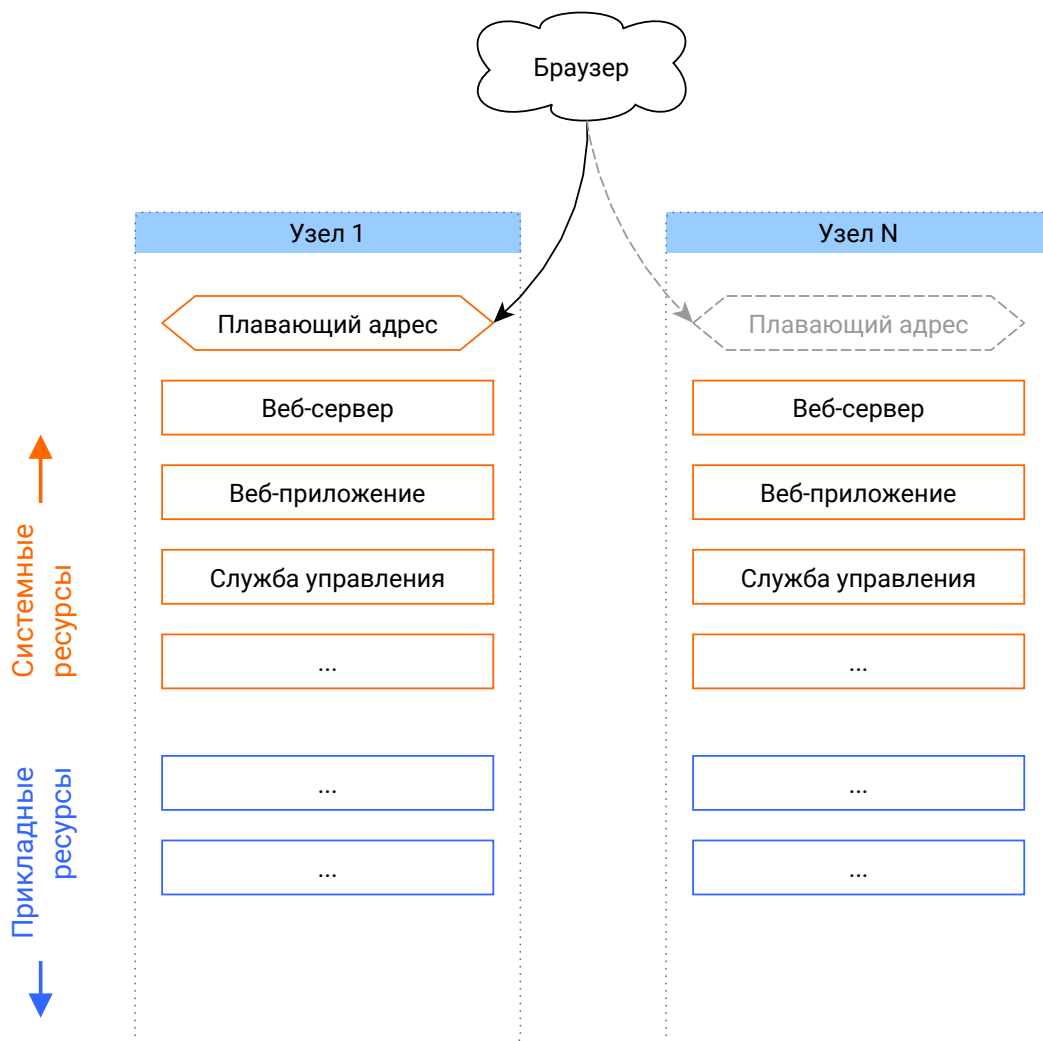


Рисунок 2.1. Интегрированная панель

Достоинства:

- высокая доступность веб-панели управления;
- не требуется дополнительных серверов.

Недостатки:

- с панели управления возможно управлять только одним кластером;
- на каждом узле кластера необходим веб-сервер Apache.

## 2.2.2 Внешняя панель управления

Вариант установки подразумевает функционирование панели управления на отдельном узле (сервере). Веб-приложение панели подключается к одному или нескольким узлам кластера через сеть управления. Пользователь подключается к узлы с установленной панелью управления. Высокая доступность функций управления достигается за счёт автоматического переключения веб-приложения панели на работающие узлы кластера. Структурная схема данного варианта установки показана на [рис. 2.2](#).

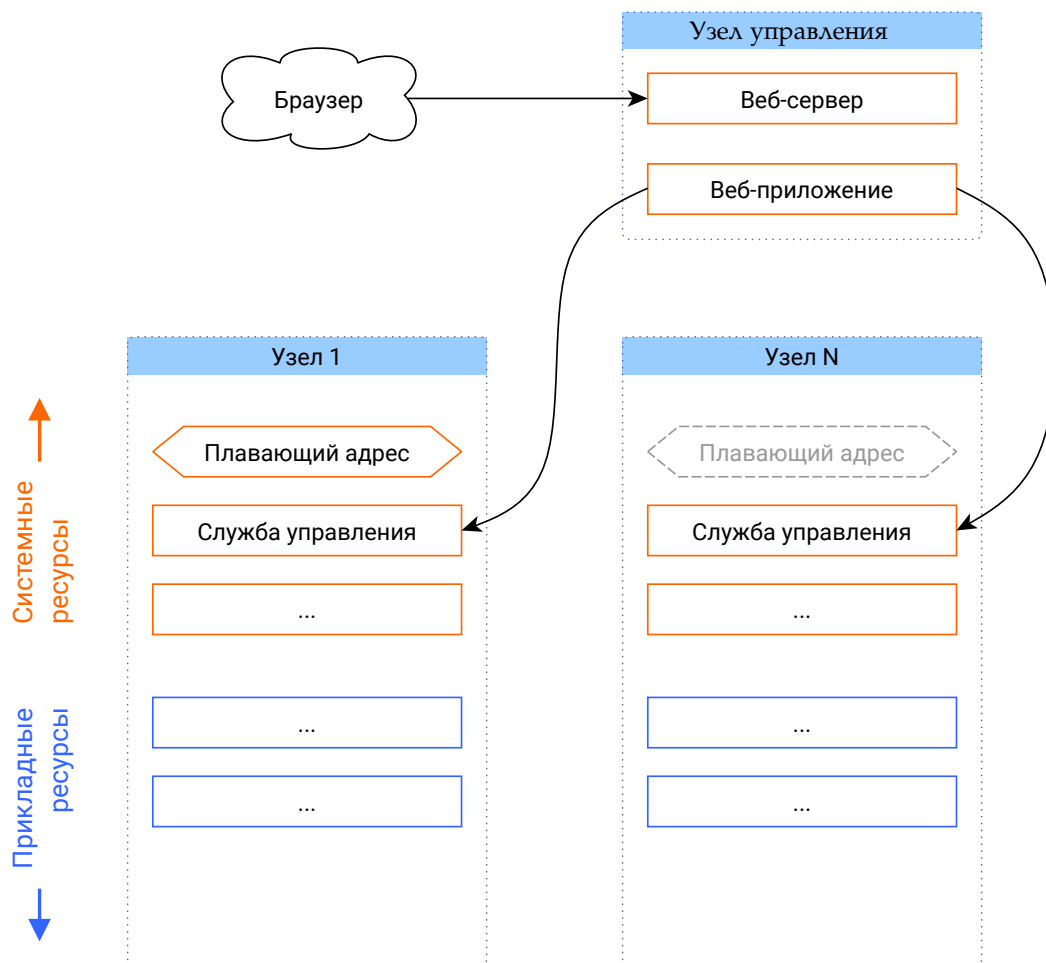


Рисунок 2.2. Интегрированная панель

**Достоинства:**

- возможность управления несколькими кластерами через одну панель;
- возможность удаления некоторых компонентов (например, веб-сервер) с узлов кластера.

**Недостатки:**

- не обеспечивается высокая доступность веб-приложения панели управления;
- необходим доступ к сети управления кластера.

Обеспечение высокой готовности веб-приложения панели управления возможно путём установки данного компонента на несколько внешних серверов. При этом переход в браузере на резервный узел необходимо будет выполнять вручную.



### 3 Подготовка целевых узлов

Подготовка целевых узлов (серверов) кластера включает в себя настройку BIOS, установку и первоначальную настройку операционной системы.

Выполните следующие проверки настроек BIOS в целях подготовки серверов для использования в кластере высокой готовности:

- 1) включение сервера после подачи питания;
- 2) включение функций виртуализации (при необходимости).

Имена всех сетевых узлов должны быть разрешимы на компьютере, с которого будет разворачиваться Простор.Кластер. Должны быть настроены следующие доменные имена для каждого узла кластера: *hostname*, *hostname.домен*, *hostname.pub*, *hostname.cntrl*. Служебные доменные имена *.pub* и *.cntrl* используются только внутри кластера и в процессе установки. Между доменными именами и IP-адресами должно быть следующее соответствие:

- *hostname* – IP-адрес, к которому будет подключаться по SSH система установки;
- *hostname.домен* – IP-адрес в публичной сети для внешнего доступа;
- *hostname.pub* – IP-адрес в публичной сети для внутреннего доступа;
- *hostname.cntrl* – IP-адрес в сети управления.

#### Примечание

Система управления Ansible используется SSH для удалённого выполнения необходимых операций. Компьютер, с которого будет производиться инсталляция кластера высокой готовности может подключаться к узлам кластера по SSH как по публичной сети, так и по сети управления. В зависимости от выбранного варианта необходимо настроить разрешение доменного имени *hostname*.

#### 3.1 Проверка IPMI

При использовании IPMI для фенсинга в кластере необходимо проверить его работоспособность. Выполнить её можно с помощью команды *ipmitool* пакета *ipmitool*. Выполнить проверку следует для каждого IPMI-интерфейса узла кластера.

Проверка стандартного интерфейса IPMI 2.0 (к примеру, сервера Intel, Supermicro) выполняется командой:

```
$ ipmitool -I lanplus -H <сетевой адрес> -U <пользователь> \  
-P <пароль> chassis power status
```

В случае правильных параметров подключения программа напечатает состояние сервера, например:

```
Chassis Power is on
```

В противном случае необходимо настроить IPMI на сервере или проверить параметры подключения.

## 4 Установка системы управления конфигурацией

Для установки системы управления конфигурацией необходимы компьютер с установленной и настроенной ОС, а также установленным ПО Ansible версии 1.9.x.

Скопируйте с поставочного носителя платформы «Простор» систему управления конфигурацией на локальный носитель. Если система поставляется в виде архива, то распакуйте его командой

```
$ tar xzf hac-deploy-x.y.tar.gz -C <каталог назначения>
```

Перед началом установки необходимо проверить доступность серверов кластера по сетевому имени.

Если не используется внешний сервер DNS, тогда настройку доменных имён необходимо выполнить на узле установки с помощью файла /etc/hosts. Пример возможных настроек для кластера с двумя сетями:

```
# Установка будет производиться через публичную сеть

# Публичная (общая) сеть: 10.0.0.0/24, домен lab50
10.0.0.2 node1.lab50 node1.pub node1
10.0.0.3 node2.lab50 node2.pub node2
10.0.0.4 node3.lab50 node3.pub node3

# Сеть управления: 172.16.0.0/24
172.16.0.2 node1.cntrl
172.16.0.3 node2.cntrl
172.16.0.4 node3.cntrl
```

Система управления конфигурацией может использовать на целевых узлах как учётную запись суперпользователя (root), так и любую другую, при необходимости. При использовании обычной учётной записи необходимо убедиться, что пользователь состоит в группе sudo.

Для операционной системы Astra Linux Common Edition мы рекомендуем использовать учётную запись из группы astra\_admin. Пользователи из этой группы могут запускать команды с **sudo** без ввода пароля.

После настройки разрешения имён для правильной работы системы управления Ansible следует перенести ключ SSH с компьютера установки на целевые узлы. При отсутствии ключа у пользователя его можно создать с помощью команды

```
$ ssh-keygen -q -N ""
```

Перенос ключей необходимо выполнить на учётную запись, которая будет использоваться для операций на целевых узлах. Например:

```
$ ssh-copy-id root@node1
```

Такую команду необходимо выполнить для каждого сервера кластера.

## 5 Конфигурация кластера

### 5.1 Общие сведения

Автоматизированная установка происходит на основе следующих подготовленных данных:

- состав (физической или виртуальной инфраструктуры);
- параметры кластера (включая параметры настройки оборудования);
- сценарий установки.

Подготовленный сценарий последовательно интерпретируется утилитой Ansible. Ansible выполняет операции сценария на целевых узлах через SSH.

#### Состав кластера

Состав целевой системы задаётся специальным файлом, описывающим аппаратные средства кластера и базовые параметры узлов. Файл состава является обычным текстовым файлом в INI-формате. Располагается в каталоге системы управления конфигурацией. Подробное описание приведено в § 5.2.

#### Параметры кластера

Конфигурационные файлы задают параметры устанавливаемого кластера высокой готовности. Параметры задаются в формате YAML (описание в прил. 1). Конфигурационные файлы могут быть разных видов. Стандартное место расположения – подкаталог `vars/`. Подробное описание вариантов приведено в прил. 2.

#### Сценарий установки

Сценарий установки задаёт состав устанавливаемых компонентов на каждый узел, сервер или группу узлов (серверов), задаваемых в файле состава. Компоненты устанавливаются в порядке перечисления. В терминах Ansible компонент называется *ролью*. Роль задаёт последовательность действий по установке и настройке. Данные роли составляются также в формате YAML (описание в прил. 1). Описание роли приведено в прил. 3.

### 5.2 Состав кластера

Состав целевой системы описывается специальным файлом, используемым системой управления Ansible. Стандартное имя файла и расположение – `cluster` в корневом каталоге системы управления конфигурацией. Файл состава является обычным текстовым файлом в INI-формате:

```
[имя группы узлов №1]
имя-узла-№1
имя-узла-№2
...

[имя группы узлов №2]
имя-узла-№1 параметр=значение
имя-узла-№2
...
```

Названия групп могут произвольно задаваться пользователем. Существуют стандартные группы, используемые в стандартной поставке системы управления конфигурацией. В качестве примера можно использовать поставляемый файл `cluster.sample`. Стандартные группы:

### nodes

Обязательная группа, содержит узлы кластеры высокой готовности. Узлы должны указываться без домена. Для каждого узла должны быть указаны следующие параметры:

- `control_ip` – адрес в сети управления;
- `public_ip` – адрес в сети данных;
- или один параметр `ip`, если последний октет IP-адресов совпадает во всех сетях.

При использовании фенсинга также должны быть указаны следующие параметры: `ipmi_ip`, `pdu_outlet`. Назначение этих параметров приведено в § 5.4.6.

Пример:

```
[nodes]
node1 control_ip=172.16.0.2 public_ip=10.0.0.2
node2 control_ip=172.16.0.3 public_ip=10.0.0.3
node3 control_ip=172.16.0.4 public_ip=10.0.0.4
```

или

```
[nodes]
node1 ip=2
node2 ip=3
node3 ip=4
```

Вы можете использовать следующие полезные параметры:

- `ansible_host`, `ansible_port`, `ansible_user` – для указания параметров SSH-подключения.

### dashboard

Эта группа необходима только в случае установки панели на отдельной машине (§ 2.2.2). Группа машин, на которых устанавливается панель управления. Если требуется, чтобы панель была установлена на узлах КВГ (интегрированный вариант установки), следует применить роль `dashboard` к узлам КВГ (в группе `nodes`).

В составе системы управления конфигурацией поставляется файл `cluster.sample`, который можно использовать как основу.

Пример файла `cluster`:

```
[nodes]
node1 ip=11 ipmi_ip=172.16.0.111
node2 ip=12 ipmi_ip=172.16.0.112
node3 ip=13 ipmi_ip=172.16.0.113

[dashboard]
admin-machine
```

## 5.3 Сценарий установки

Последовательность действий по установке и настройке кластера задаётся сценарием установки. Сценарий — это *playbook* в системе управления конфигурацией Ansible. Сценарии описываются на языке YAML (прил. 1).

В системе управления типовой сценарий задаёт перечень необходимых ролей для целевых узлов. Тем не менее, администратор волен составлять сценарии по своему усмотрению. В этом случае рекомендуем вам обратиться к руководству по системе управления Ansible, доступному по адресу [docs.ansible.com](http://docs.ansible.com) [http://docs.ansible.com].

Типовой минимальный сценарий установки кластера высокой готовности (с интегрированной панелью управления):

```
- hosts: nodes
  user: user
  sudo: true
  any_errors_fatal: true
  vars_files:
    - vars/config.yml

  roles:
    - base
    - network
    - ntp
    - pacemaker
    - mgmt
    - apache
    - webssh
    - dashboard
    - { role: glusterfs, volume: "{{ volume_vms }}" }
    - {
      role: glusterfs_client,
      mountpoint: "{{ virtualization.share }}",
      volume: "{{ volume_vms }}"
    }
    - vm-server
```

### 5.3.1 Запрос дополнительных данных

Если по требованиям безопасности запрещено указывать пароли или другие важные данные открытым текстом в файлах конфигурации, возможен вариант запроса параметров при запуске сценария.

Список запрашиваемых параметров задаётся конструкцией `vars_prompt` в сценарии. Каждый элемент списка `vars_prompt` содержит описание запрашиваемого параметра.

#### **name**

Название параметра.

#### **prompt**

Пояснительный текст, который будет выведен при запросе данных.

#### **private**

Необязательный параметр. Флаг скрытия вводимых данных. Полезен при задании паролей.

**confirm**

Необязательный параметр. Флаг необходимости повторного ввода и сравнения данных. Полезен при задании паролей.

Пример запроса пароля администратора Astra Linux Directory для подключения к внешнему серверу:

```
...
vars_files: ...
vars_prompt:
  - name: "ald_password"
  - prompt: "Input ALD administrator password"
  - private: yes
  - confirm: no
...
```

**5.3.2 Роли**

По целевому назначению большинство ролей не являются универсальными, а предназначены для настройки узлов кластера или серверов администрирования. Список ролей системы управления конфигурацией приведен в табл. 5.1.

Таблица 5.1. Роли

Название	Описание
<b>Универсальные роли</b>	
base	Базовая настройка целевых узлов
ntp	Настройка синхронизации времени
ald-client	Настройка подключения к домену Astra Linux Directory на рабочих местах операторов
glusterfs-client	Настройка подключения к отказоустойчивой файловой системе Gluster
dashboard	Развёртывание панели управления кластером высокой готовности
<b>Роли для настройки узлов кластера</b>	
network	Управление сетевой конфигурацией узлов кластера
racemaker	Развёртывание базового ПО
mgmtd	Развёртывание службы управления кластером высокой готовности
webssh	Доступ в узлам кластера через веб-терминал
glusterfs	Развёртывания на узлах кластера отказоустойчивой файловой системы Gluster
vm-server	Развёртывание комплекса защищённой виртуализации
<b>Роли для настройки узлов администрирования (рабочих мест оператора)</b>	
network-client	Управление сетевой конфигурацией рабочих мест операторов и администраторов

Название	Описание
firefox	Настройка веб-браузера для подключения к панели управления

Данный набор описывает все необходимые операции по установке и настройке Простор.Кластер, Простор.Управление, Простор.VDI, Простор.Хранение на серверах кластера и администрирования. Каждая роль предназначена для настройки для определённых функциональных аспектов целевой системы. При работе с ролями следует учитывать неявные взаимосвязи между ними.

Зависимости между ролями (порядок применения) приведён на рисунках 5.1, 5.3 и 5.2. Звёздочкой отмечены необязательные роли. В конкретной поставке платформы они могут отсутствовать. Необходимость роли ald-client определяется типом выбранной аутентификации.

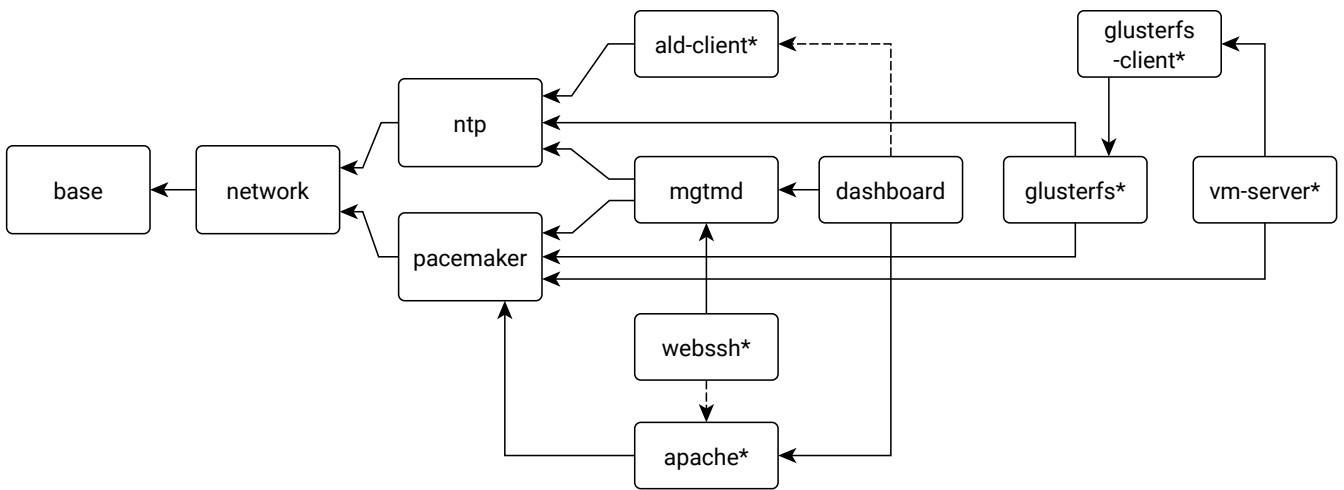


Рисунок 5.1. Порядок применения ролей узлов кластера

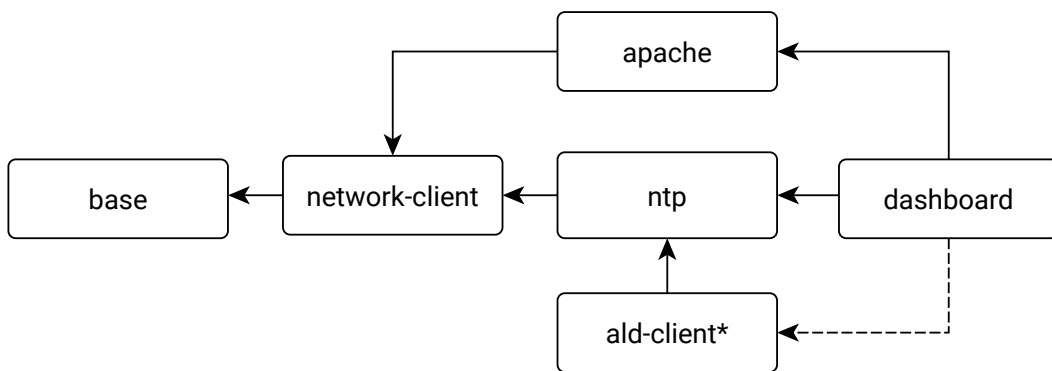


Рисунок 5.2. Порядок применения ролей узла управления

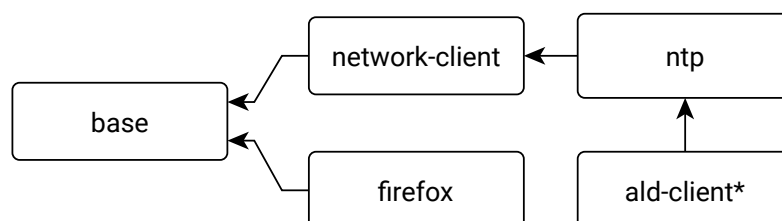


Рисунок 5.3. Порядок применения ролей для рабочих мест оператора

### 5.3.3 Типовые сценарии

Система управления конфигурацией содержит в своем составе несколько типовых сценариев (§ 2.2), пригодных для установки и настройка кластера высокой готовности.

#### **cluster.yml**

Сценарий базовой установки кластера высокой готовности в варианте внешней панели управления. В качестве сервиса аутентификации может использоваться как локальная схема через PAM, так и внешний сервер Astra Linux Directory.

#### **dashboard.yml**

Сценарий установки панели управления на сервер администрирования.

#### **operator.yml**

Вспомогательный сценарий для настройки рабочего места (APM) оператора или администратора кластера.

#### **playbooks/uninstall.yml**

Сценарий удаления компонентов с узлов кластера. Может быть полезен при тестировании сценариев и параметров кластера. После удаления компонентов данным сценарием можно производить установку компонентов Простора в приблизительно «чистовом» варианте.

## 5.4 Параметры

Конфигурационные файлы задают параметры устанавливаемого кластера высокой готовности. Параметры задаются в формате YAML (описание в прил. 1). Конфигурационные файлы могут быть разных видов. Стандартное место расположения — подкаталог `vars/`. Подробное описание вариантов приведено в прил. 2.

### Примечание

В поставке системы управления конфигурацией есть пример файла параметров:  
`vars/cluster.yml.sample`

### 5.4.1 Базовые настройки

Базовые настройки включают параметры, обязательные для конфигурации. Для них не существует стандартных значений. Базовые настройки используются всеми ролями.

Базовые настройки сосредоточены в блоке `common`:

#### **domain**

Сетевой домен, который будут использовать все службы и сервисы, разворачиваемые на кластере. При использовании Astra Linux Directory или «Высокодоступного домена «Прокаталог» должен совпадать с доменом ALD.

#### **public**

Параметры внешнего подключения.



**host**

Сетевое имя узла, который будет использоваться для подключения внешних клиентов. Должен соответствовать плавающему IP.

**ip**

Адрес плавающего IP, который будет использоваться для подключения внешних клиентов к ресурсам кластера.

Пример:

```
common:
  domain: lab50.net

public:
  host: cluster
  ip: 10.0.0.50
```

## 5.4.2 Репозитории установки

Списком репозитория установки оперирует роль `base`. Может быть указано произвольное количество необходимых репозитория. Для указания репозитория используется стандартный формат APT (`/etc/apt/sources.list`).

Адреса репозитория задаются в блоке `repository` в формате

```
название: репозиторий
```

Стандартные настройки:

```
repository:
# Адрес репозитория с дистрибутивом Астры Common Edition
os: "deb http://mirror.yandex.ru/astra/current/orel/repository/ orel main contrib non-free"

# Адрес репозитория с дистрибутивом КВГ
hac: "deb http://packages.lab50.net/hac/ ce main non-free"

# Пакет со списком ключей, которыми подписываются репозитории
repository_keyring: lab50-archive-keyring
```

Для доступа к репозиториям могут использоваться различные варианты, например:

- через HTTP/FTP: `deb http://сайт/каталог ...;`
- файловая система: `deb file://путь ...;`
- по SSH (необходимо перенести ключ с помощью `ssh-copy-id`): `deb ssh://user@host/путь ....`

Пакет со списком ключей, которыми подписываются репозитории, задаётся с помощью параметра `repository_keyring`. Стандартное значение: `lab50-archive-keyring`. Это пакет содержит ключи публичных репозитория и всех продуктов «Лаборатории 50».

## 5.4.3 Настройки сети

Настройки сети производятся ролью `network`. Роль выполняет следующие основные операции:

- 1) сетевая конфигурация ОС Astra Linux Common Edition;
- 2) настройка `/etc/hosts` на целевых узлах;

3) применение сетевой конфигурации на узле.

В ходе применения сетевой конфигурации узлы при необходимости могут автоматически перезагружаться.

Сетевая конфигурация сосредоточена в блоке `network`. В настоящей версии системы управления конфигурации возможна настройка двух сетей: сети управления (внутренней сети кластера) и сети данных (внешней сети кластера). Через сеть данных производится взаимодействие с внешним миром, в то время как к внутренней сети имеет доступ только кластер.

Описание настроек сети управления находится в блоке `network.control`, сети данных — в блоке `network.public`. Оба блока полностью идентичны по составу параметров.

Каждый блок настроек содержит следующие параметры.

#### **address**

Первые три октета (A.B.C) IP-адреса сети. Например: «192.168.1».

#### **dhcp**

Флаг использования внешнего сервера DHCP для получения адресов узлами. Значения `true/false`.

#### **bonding**

Настройка использования агрегирования (бондинга) сетевых интерфейсов. Агрегирование будет настроено со схеме резервирования одного канала. Значение: список интерфейсов (из двух элементов). Если агрегирование не используется, оставьте поле пустым. Пример: «[eth0, eth2]».

#### **interface**

Сетевой интерфейс Линукс, подключенный к соответствующей сети. При использовании агрегирования это будет `bondx`.

#### **bridge**

*Необязательный параметр.* Если необходим сетевой мост, например, для виртуальных машин тогда укажите имя моста. В этом случае ему будет присваиваться адрес в соответствующей сети.

#### **debconf**

Дополнительные настройка сетевого интерфейса в формате Дебьян, которые будут перенесены в файл конфигурации `/etc/network/interfaces`. Блок может содержать любые корректные параметры и их значения. Например:

```
netmask: "255.255.255.0"  
gateway: "10.0.0.1"  
dns-nameservers: "10.0.0.2"
```

Пример конфигурации:

```
network:  
# Сеть управления  
control:  
  address: '172.16.0'  
  dhcp: false  
  bonding: [eth1, eth2]
```

```
interface: bond0
debconf:
  netmask: 255.255.255.0
  gateway:
  dns-search:
  dns-nameservers:

# Сеть данных
public:
  address: '10.10.50'
  dhcp: false
  bonding: [eth0, eth3]
  interface: bond1
  bridge: br0
  debconf:
    netmask: 255.255.255.0
    gateway:
    dns-search:
    dns-nameservers:
```

#### 5.4.4 Настройка синхронизации времени

Для корректной работы практически любого сервиса в кластере необходима синхронизация времени на всех узлах кластера и рабочих местах операторов. При использовании в любом качестве домена Astra Linux Directory, синхронизация времени является обязательным требованием.

Для настройки синхронизации времени предусмотрена роль `ntp`. Эта роль может применяться как на узлах кластера, так и клиентских машинах. Роль предусматривает возможность настройки синхронизации от внешнего сервера, так и использование в качестве источника времени одного из узлов кластера. Для синхронизации используется внешняя сеть.

Настройки синхронизации времени находятся в блоке `ntp`. В этом блоке должны перечисляться адреса серверов времени для синхронизации. Для использования узла в качестве сервера *NTP*, его необходимо перечислить в этом списке *первым*. При установке кластера первый сервер в списке будет использоваться для начальной синхронизации времени.

Для данной роли нет стандартных настроек.

Пример:

```
ntp:
- 0.ru.pool.ntp.org
- 1.ru.pool.ntp.org
- 2.ru.pool.ntp.org
- 3.ru.pool.ntp.org
```

#### 5.4.5 Настройка системы управления кластером

Система управления кластером `Racemaker` — это базовый элемент кластера высокой готовности. Для настройки кластера предназначена роль `racemaker`. Основные функции роли:

- 1) установка и настройка `Corosync` и `Racemaker`;

- 2) создание необходимых ресурсов;
- 3) создание дополнительных ресурсов;
- 4) настройка фенсинга (будет описана в § 5.4.6).

Настройки системы управления кластером сосредоточены в блоках `corosync`, `resources`, `fencing`.

В блоке `corosync` задаются сетевые настройки кластера.

#### **mcastaddr**

Адрес для синхронизации узлов кластера. Должен быть из сети `multicast`. Стандартное значение: `239.255.1.1`.

#### **mcastport**

Порт для синхронизации узлов кластера. Стандартное значение: `4001`.

### **Примечание**

Если у вас в одной сети управления находятся несколько кластеров (например, в целях тестирования), у каждого кластера должна быть уникальная комбинация адреса и порта.

Роль `rasemaker` создаёт только один обязательный ресурс: плавающий IP (параметры описаны в § 5.4.1). Для каждого программного компонента из состава платформы «Простор», входящего в дистрибутив, при установке автоматически создаются необходимые ресурсы кластера. Однако, в системе управления конфигурацией предусмотрена возможность создания дополнительных пользовательских ресурсов. Эти ресурсы будут созданы ролью `rasemaker` после настройки и запуска всех обязательных компонентов.

Существует возможность создания следующих типов дополнительных ресурсов: простые, группы и клоны. Дополнительные ресурсы описываются в блоке `resources`. Эти ресурсы являются необязательными для установки кластера.

#### **list**

Список простых ресурсов.

#### **groups**

Список групп. Состав каждой группы должен описываться в блоки `resources`.

#### **clones**

Список клонов. Каждый элемент списка должен содержать параметр `resource` — название исходного ресурса. Свойства клона можно указывать с помощью параметра `meta`.

Пример задания ресурсов:

```
resources:
  list:
    - myresource
    - res1
    - res2
```

```
groups:
  - mygroup

clones:
  - resource: myresource
    meta:
      target-role: Started
      interleave: true

mygroup: [ res1, res2 ]
```

Стандартные параметры:

```
resources: []

corosync:
  # Multicast-адрес используемый Corosync для синхронизации
  # 226.95.1.1, 226.94.1.1
  mcastaddr: 239.255.1.1
  # Порт для синхронизации
  mcastport: 4001
```

### 5.4.6 Фенсинг

Фенсинг настраивается ролью `racemaker`. Система управления конфигурацией поддерживает следующие устройства фенсинга:

- управляемые распределители питания (PDU);
- интерфейсы IPMI;
- для виртуальных узлов через интерфейс Libvirt;

При использовании УРП предполагается, что каждый сервер оснащён двумя блоками питания, т.е. устройств УРП также два.

Настройки фенсинга задаются блоком `fencing`.

#### **pdu**

Необязательный блок. При использовании УРП должен содержать список устройств, каждое из которых описывается следующими параметрами:

##### **name**

Название устройства (ASCII).

##### **ip**

IP-адрес устройства.

#### **ipmi**

Необязательный блок. При использовании IPMI должен содержать настройки подключения:

##### **login**

Логин для аутентификации.

##### **password**

Пароль для аутентификации.

## virsh

Необязательный блок. При использовании виртуальных узлов и системы виртуализации Libvirt должен содержать:

### host

Сервер виртуализации, на котором работают виртуальные узлы.

### login

Учетная запись для подключения по SSH.

### password

Учетная запись для подключения по SSH.

Пример настроек фенсинга:

```
fencing:
  pdu:
    - name: left
      ip: 172.16.0.5
    - name: right
      ip: 172.16.0.6

  ipmi:
    login: 'ADMIN'
    password: 'ADMIN'
```

## IPMI

При использовании фенсинга через IPMI помимо настроек доступа необходимо указывать IP-адрес интерфейса IPMI каждого узла. Адрес задаётся с помощью параметра `ipmi_ip` для каждого узла в файле состава (§ 5.2).

## УРП

При использовании фенсинга через управляемые распределители питания помимо адресов устройств необходимо указывать номер розетки сервера в УРП. Предполагается, что в двух УРП каждый сервер включается в одинаковые розетки. Номер розетки задаётся параметром `pdu_outlet` для каждого узла в файле состава (§ 5.2).

### 5.4.7 Служба управления

Для настройки службы управления на узлах предназначена роль `mgmt.d`. Роль автоматически устанавливает и настраивает необходимые программные компоненты.

### 5.4.8 Веб-сервер

Для настройки веб-сервера Apache предназначена роль `apache`. Роль автоматически устанавливает и настраивает необходимые программные компоненты. Если роль устанавливается на узлы кластера, будет автоматически создан ресурс для веб-сервера.

Веб-сервер требуется для работы панели управления в любом варианте установки (интегрированном или внешнем).

### 5.4.9 Веб-терминал

Веб-терминал (роль `webssh`) позволяет подключаться к узлам кластера по SSH из панели управления. Роль автоматически устанавливает и настраивает необходимые программные компоненты.

### 5.4.10 Панель управления

Для настройки панели управления кластером служит роль `dashboard`. Панель может работать как на узлах кластера (вариант интегрированной установки – § 2.2.1), так и на сторонних машинах (вариант внешней установки – § 2.2.2). Конфигурация задаётся блоком параметров `dashboard`:

#### **access**

Блок настроек аутентификации и авторизации пользователей.

#### **admins**

Группа для роли администратора. Стандартное значение: `administrators`.

#### **users**

Группа для роли операторов. Стандартное значение: `users`.

#### **auth**

Механизм аутентификации и авторизации. Значения: `ald` (стандартное), `ram`, `none` (без авторизации).

#### **cache**

Путь размещения кэша панели. Стандартное значение: `/tmp/hac-control-cache`.

### Примечание

Параметры `access.admins` и `access.users` должны соответствовать создаваемым группам в роли `ald-client`.

Стандартные параметры:

```
# Панель управления
dashboard:
  access:
    admins: administrators
    users: users
    auth: none
  cache: /tmp/hac-control-cache
```

### 5.4.11 Настройка параметров Глостера

Установка и настройка сервера распределенной файловой системы производится ролью `glusterfs`. Перед использованием системы управления конфигурацией необходимо спланировать размещение и параметры томов Глостера:

- 1) сеть используемую Глостером для работы;

- 2) устройства на которых будут размещаться кирпичи;
- 3) тип и параметры создаваемого тома.

### Используемая сеть

Простор.Кластер предполагает наличие как минимум двух сетей. Сеть управления используется для внутренней синхронизации кластера, а также передачи важных данных. Выделение сети управления в отдельную физическую сеть служит для изоляции от возможных проблем, которые могут быть вызваны неправильной работой прикладных программ или неверными настройками пользователей.

В публичной сети работают все пользовательские сетевые приложения, происходит доступ к Панели управления кластером и т.д. При наличии только двух сетей, распределенная файловая система должна быть настроена на использование публичной сети.

Производительные кластерные инсталляции могут выделять специальную сеть для работы систем хранения данных, в т.ч. Гластер.

Сеть, которая будет использоваться распределенной файловой системой заедается адресами серверов Гластер. Например, для использования публичной сети должны использоваться IP-адреса или имена узлов в общем домене. Данные имена используются только внутри самого кластера, поэтому нет необходимости для их настройки во вне (например, внешних серверов DNS).

### Создание томов

Настройки томов задаются переменной `volumes`. Переменная перечисляет тома, каждый из которых содержит следующие параметры:

#### **name**

Название тома.

#### **brick**

Расположение (каталог) кирпича на сервере.

#### **device**

Блочное устройство на котором будет располагаться кирпич. В процессе установки оно будет отформатировано. Запись для автоматического монтирования к каталогу, указываемому параметром `brick`, будет добавлена в `/etc/fstab`.

#### **transport**

Транспорт, который будет использовать Гластером. Стандартное значение: `tcp`.

#### **force**

Флаг принудительного создания тома. Необходим в некоторых случаях, например, при расположении кирпича на одном устройстве с другими данными. Стандартное значение: `false`.

#### **replica**

Флаг создания тома с зеркалированием. Стандартное значение: `false`.



**replica\_count**

Количество одновременных реплик (копий) для тома. Стандартное значение: количество узлов в кластере.

**disperse**

Флаг создания тома с чередованием. Стандартное значение: false.

**disperse\_count**

Количество кирпичей в чередовании. Стандартное значение: 0.

**group**

Тип оптимизации производительности. Необязательный параметр.

Пример.

```
# Единственный том: зеркало с тройным резервированием
# для работы виртуальных машин
volumes:
- name: vms
  brick: /srv/brick/vms
  group: virt
  replica: yes
  replica_count: 3
  force: yes
```

Стандартные параметры:

```
glusterfs:

# force
force: false

# ping-timeout 20s
ping_timeout: 20

# Отключение NFS: on/off
disable_nfs: on

quorum:
  ratio: 51%
```

## 5.4.12 Настройка монтирования Глестер-тома

Роль `glusterfs_client` создаёт ресурс для монтирования заданного тома на узлах кластера. Настройки передаются через параметры роли:

**volume**

Название Глестер-тома.

**mountpoint**

Точка монтирования тома.

Для монтирования нескольких томов используйте роль последовательно для каждого тома.

Пример.

```
- { role: glusterfs_client,  
    volume: "myvolume",  
    mountpoint: "/srv/myvolume" }
```

### 5.4.13 Виртуализация

Для настройки виртуализации служит роль `vm-server`, конфигурация задаётся в блоке параметров `virtualization`. Параметры:

#### **share**

Путь к каталогу с общим хранилищем. Стандартное значение: `/srv/vms`.

#### **backup**

Путь к каталогу резервного копирования VM.

#### **hugepages**

Использование больших страниц. Должно быть указано количество резервируемых страниц.

Стандартные параметры:

```
virtualization:  
  
# Путь к каталогу с общим хранилищем.  
share: /srv/vms  
  
# Путь к каталогу резервного копирования VM.  
#backup:  
  
# Использование больших страниц. Должно быть указано  
# количество резервируемых страниц.  
#hugepages: 512
```

## 5.5 Проверка файлов конфигурации

Для проверки синтаксиса созданных файлов конфигурации и сценария следует воспользоваться командой **ansible-playbook** с аргументов `--syntax-check`:

```
$ ansible-playbook --syntax-check config.yml
```

## 6 Установка кластера

### 6.1 Предварительная проверка

Система управления конфигурацией Ansible имеет полезный режим тестового прогона сценария. В этом режиме не производится реальных действий на целевых узлах. Этот режим удобен для проверки созданной конфигурации и понимания выполняемых шагов при установке.

Тестовый режим включается с помощью параметра `--check`:

```
$ ansible-playbook --check cluster.yml
```

Проверке подлежат все используемые сценарии, как установки узлов, так и настройки рабочих станций и серверов управления.

#### Примечание

Перед выполнением проверки необходимо выполнить все подготовительные шаги для запуска установки, в том числе копирование ключей ssh (разд. 3).

### 6.2 Установка

Запуск установки комплекса кластера высокой готовности должен производиться после выполнения всех подготовительных шагов, включая настройку оборудования и описания конфигурации кластера.

Система управления конфигурацией запускается командой **ansible-playbook**. Команде должен быть указан файл сценария, например:

```
$ ansible-playbook --become-user=учетная запись cluster.yml
```

Команда **ansible-playbook** допускает указание следующих аргументов:

**-i <файл>**

Файл состава. Без явного указания файла используется файл `stage`.

**--step**

Пошаговое выполнение сценария.

**--ask-become-pass**

Пароль пользователя на целевых узлах для использования `sudo`.

**-v, -vv, -vvv, -vvvv**

Вывод дополнительной (отладочной) информации при работе.

В течение работы системы управления конфигурацией оператору выводятся сообщения о выполняемых задачах в следующем формате:

```
TASK: [ роль | название задачи ]
результат1: [узел1]
результат2: [узел2]
...
```

Каждая задача выполняется на одном или более узлах с соответствующим результатом. Результат выполнения может быть следующим:

- ok – задача (команда) успешно выполнена;
- changed – задача успешно выполнена, на целевом узле были внесены изменения;
- skipping – задача пропущена для данного узла (в силу условий);
- failed – при выполнении задачи произошла ошибка.

После выполнения всех задач сценария выводится сводная таблица по всем целевым узлам, например:

```
node1           : ok=139  changed=105  unreachable=0  failed=0
node2           : ok=125  changed=87   unreachable=0  failed=0
...
```

Установка является успешной при условии отсутствия неуспешных задач (failed) на всех узлах.

После успешного завершения работы системы управления конфигурацией кластер полностью готов к использованию.

## 7 Сообщения системному программисту

Сообщения системы управления конфигурацией выдаются в едином формате и имеют описание на русском языке. Формат сообщений описан в § 6.2.

Все сообщения сохраняются в файле журнала `install.log`.

## Приложение 1. Формат YAML

YAML – формат сериализации данных, концептуально близкий к языкам разметки, но ориентированный на удобство ввода-вывода типичных структур данных многих языков программирования.

Синтаксис YAML минималистичен, особенно по сравнению с синтаксисом XML. В стандарте указывается, что большое влияние на синтаксис оказал стандарт RFC 822.

YAML позволяет описывать все основные структуры данных: переменные, списки, ассоциативные массивы (словари). Служебные слова Ansible приносят возможность использования таких конструкций как циклы и т.п.

### Основы синтаксиса

Основные синтаксические правила формата YAML:

- Сценарии YAML используют печатаемые символы Юникод в формате UTF-8 и UTF-16.
- Отступы из пробелов (символы табуляции не допускаются) используются для обозначения структуры.
- Комментарии начинаются с символа «решётка» (#), могут начинаться в любом месте строки и продолжаются до конца строки.
- Списки обозначаются начальным дефисом (-) с одним членом списка на строку, либо члены списка заключаются в квадратные скобки ([ ]) и разделяются запятой и пробелом (,).
- Ассоциативные массивы оформляются двоеточием с пробелом (:) в виде «ключ: значение», по одной паре ключ-значение на строку, либо в виде пар, заключённых в фигурные скобки и разделённых запятой и пробелом (,).
- Строки записываются без кавычек, однако могут быть заключены в одиночные или двойные кавычки. Внутри двойных кавычек могут быть использованы экранированные символы в стиле языка Си, начинающиеся с обратного слэша (\).

### Структуры данных

Структуры данных могут оформляться в виде блочного или строчного представления. Блочное представление выглядит следующим образом:

```
# Список элементов
--- # Может использоваться для разделения сценариев
- base
- network
- ald

---
# Словарь
host: cluster
ip: 192.168.1.1
```

Его эквивалентом в строчном представлении будет является:

```
# Список элементов
---
[ base, network, ald]
---
# Словарь
{ host: cluster, ip: 192.168.1.1 }
```

## Приложение 2. Конфигурационные файлы

Конфигурационные файлы задают параметры устанавливаемого кластера высокой готовности. Существует несколько основных способов задания параметров:

- командная строка запуска установки;
- задание в сценариях;
- через включение самостоятельных файлов конфигурации;
- специальные файлы конфигурации групп и узлов.

### Задание в командной строке

Параметры конфигурации кластера можно задавать при запуске инсталляции с помощью параметра `--extra-vars`:

```
$ ansible-playbook setup.yml --extra-vars "var1=value1 var2=value2"
```

### Параметры в сценариях

В простейшем случае параметры кластера могут задаваться напрямую в запускаемых сценариях:

```
---
- hosts: nodes
  user: root
  vars:
    common:
      master_node: '{{ groups.nodes[0] }}'
      domain: npdb
      public:
        host: cluster
        ip: 10.0.0.1
```

Недостатком этого способа является локальный характер задания параметров. Они будут использоваться только в данном сценарии и для данной группы.

### Файлы конфигурации

Файлы конфигурации задаются в формате YAML. Расположение файлов может быть произвольно, однако рекомендуемым местом расположения является подкаталог `var/`. Подготовленные файлы могут включаться из сценариев с помощью блока `vars_files`:

```
---
- hosts: nodes
  user: root
  vars_files:
    - vars/main.yml
    - vars/network.yml
...
```

Указать использование определённого файла конфигурации можно прямо из командной строки:



```
$ ansible-playbook setup.yml --extra-vars "@vars/cfg.yml
```

Мы рекомендуем этот способ.

## Специальные файлы конфигурации

Специальные файлы конфигурации задают параметры для определённых групп и узлов. Подобно обычным файлам конфигурации они задаются в формате YAML. Специальные файлы удобны для задания общих или стандартных параметров.

Файлы конфигурации для групп располагаются в каталоге `group_vars/`. Имена файлов должны совпадать с названиями групп. Например, `group_vars/clients`. Вы можете использовать файл `all`, который будет использоваться для всех групп. Значения параметров из более специфических групп переопределяют значения из общих групп (например, `all`).

Файл конфигурации для конкретных узлов располагаются в каталоге `host_vars/`. Имена файлов должны совпадать с названиями узлов.

## Приоритет параметров конфигурации

В случае, когда конфигурационные параметры заданы в нескольких источниках, они применяются в следующем приоритете (по убыванию):

- 1) командная строка;
- 2) внешние файлы конфигурации заданные в блоке `vars_files`;
- 3) параметры в блоке `vars`;
- 4) файлы конфигурации узлов в `host_vars/`;
- 5) файлы конфигурации групп в `group_vars/`;
- 6) стандартные значения заданные в ролях.

### Примечание

Будьте внимательны при работе с несколькими YAML-файлами конфигурации. При переопределении конкретного параметра, принадлежащего определённому блоку, вы должны задать (повторить) все параметры данного блока.

## Приложение 3. Роли

Ролью называется типовой набор задач, охватывающий определённую структурную часть целевой системы. Роль объединяет задачи, переменные, триггеры, зависимости, в единую логическую структуру, которая может использоваться достаточно самостоятельно. Роли располагаются в каталоге `roles/`.

Каждая роль обладает собственной структурой, включающей описания следующих элементов:

- 1) задачи (операции) (`tasks/`);
- 2) необходимые файлы (`files/`);
- 3) шаблоны файлов (`templates/`);
- 4) триггеры (`handlers/`);
- 5) данные (`vars/`);
- 6) стандартные значения параметров (`defaults/`);
- 7) зависимости (`meta/`).

Каждый элемент роли располагается в собственном подкаталоге роли.