

ООО «Астра Консалтинг»

Астра ИС МД (Инфраструктурные Сервисы)

Модуль Astra Scylla

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРЕДОСТАВЛЕННОГО ДЛЯ
ПРОВЕДЕНИЯ ЭКСПЕРТНОЙ ПРОВЕРКИ**

Москва, 2026

Список терминов и сокращений

Таблица 1. Список терминов и сокращений

Термин, сокращение	Определение, расшифровка
ПО	Программное обеспечение
ЦОД	Центр обработки данных

Содержание

1	Установка Astra Scylla с использованием инсталлятора.....	4
1.1	Установка Ansible:	4
1.2	Установка необходимых Python-модулей.....	4
2	Инструкция по установке Astra Scylla с использованием плейбуков Ansible.....	5
2.1	Шаг 1: Настройка файла инвентаря	5
2.2	Шаг 2: Настройка плейбука.....	5
2.3	Шаг 3: Настройка переменных.....	5
2.4	Настройка метода установки в Ansible	7
2.5	Шаг 4: Запуск плейбука.....	7
2.6	Дополнительные шаги	7
3	Проверка состояния кластера с помощью nodetool status	8
3.1	Выполнение команды nodetool status	8
3.2	Пример вывода команды nodetool status	8
3.3	Разбор вывода команды nodetool status.....	8
3.4	Анализ результатов.....	9
4	Описание плейбуков.....	10
5	Заключение	11

1 Установка Astra Scylla с использованием инсталлятора.

Для использования инсталлятора требуется предварительная подготовка виртуальных машин и серверов, включая настройку операционной системы, установку необходимых пакетов и инструментов, а также настройку сетевых подключений.

1.1 Установка Ansible:

- Убедитесь, что Ansible версии 2.9 и выше установлен.

Рекомендуется использовать Python 3.8 и выше.

```
sudo apt update  
sudo apt install -y ansible python3-pip
```

1.2 Установка необходимых Python-модулей

```
pip3 install --user ansible
```

2 Инструкция по установке Astra Scylla с использованием плейбуков Ansible

2.1 Шаг 1: Настройка файла инвентаря

Создайте файл `inventory.yml` с IP-адресами ваших узлов:

```
[Astra Scylla_nodes]
x.x.x.x rack_number=1
x.x.x.x rack_number=2 seed=true
x.x.x.x rack_number=3

[Astra Scylla_nodes:vars]
cl=my_cluster_name

[Astra Scylla_seeds]
x.x.x.x rack_number=2 seed=true

[all:vars]
dc=my_datacenter

[all:vars]
ansible_connection=ssh
ansible_user=astra
ansible_ssh_private_key_file=~/.ssh/id_rsa
```

2.2 Шаг 2: Настройка плейбука

Основной плейбук для запуска установки находится в файле `start.yml`:

2.3 Шаг 3: Настройка переменных

Файл `roles/Astra Scylla_install/vars/main.yml` содержит основные переменные для настройки кластера:

cluster_name: `"{{ hostvars[inventory_hostname]['cl'] }}"` -Имя кластера. В данном случае оно берется из переменной `cl` хоста, указанного в `inventory_hostname`.

listen_address: "{{ inventory_hostname }}" - Адрес, на котором Astra Scylla будет слушать запросы от других узлов кластера. Обычно это IP-адрес или имя хоста текущего узла.

rpc_address: "{{ inventory_hostname }}" - Адрес, на котором Astra Scylla будет принимать RPC (Remote Procedure Call) запросы. Это может быть полезно для мониторинга и управления. Здесь также используется IP-адрес или имя хоста текущего узла.

api_address: "0.0.0.0" - Адрес, на котором Astra Scylla будет слушать API запросы. Значение 0.0.0.0 означает, что API будет доступен на всех сетевых интерфейсах.

endpoint_snitch: "GossipingPropertyFileSnitch" - Снитч (snitch), который используется для определения местоположения данных внутри кластера. В данном случае используется GossipingPropertyFileSnitch, который автоматически обновляет информацию о топологии кластера.

hinted_handoff_enabled: true - Флаг, указывающий, включена ли функция hinted handoff. Hinted handoff помогает восстановить данные, если узел был недоступен во время записи.

incremental_backups: true - Флаг, указывающий, включены ли инкрементальные бэкапы. Инкрементальные бэкапы позволяют сохранять только изменения с момента последнего полного бэкапа.

tombstone_warn_threshold: 1000 - Пороговое значение для предупреждений о tombstones (метках удаления). Если количество tombstones превышает это значение, система будет выдавать предупреждения.

tombstone_failure_threshold: 100000 - Пороговое значение для ошибок, связанных с tombstones. Если количество tombstones превышает это значение, система будет выдавать ошибки.

compaction_large_partition_warning_threshold_mb: 1000 - Пороговое значение для предупреждений о больших партициях в мегабайтах. Если размер партиции превышает это значение, система будет выдавать предупреждения.

compaction_large_row_warning_threshold_mb: 1000 - Пороговое значение для предупреждений о больших строках в мегабайтах. Если размер строки превышает это значение, система будет выдавать предупреждения.

compaction_large_cell_warning_threshold_mb: 1000 - Пороговое значение для предупреждений о больших ячейках в мегабайтах. Если размер ячейки превышает это значение, система будет выдавать предупреждения.

compaction_rows_count_warning_threshold: 100000 - Пороговое значение для предупреждений о количестве строк. Если количество строк превышает это значение, система будет выдавать предупреждения.

compaction_collection_elements_count_warning_threshold: 100000 - Пороговое значение для предупреждений о количестве элементов коллекций. Если количество элементов превышает это значение, система будет выдавать предупреждения.

dst_path: "/tmp/Astra Scylla-packages" - Путь, куда будут загружаться пакеты Astra Scylla. В данном случае это /tmp/Astra Scylla-packages.

dc: "my_datacenter" - Название датацентра, в котором находится текущий узел.

rack: "{{ hostvars[inventory_hostname]['rack_number'] }}" - Номер стойки, в которой находится текущий узел. Здесь значение берется из переменной rack_number хоста, указанного в inventory_hostname.

2.4 Настройка метода установки в Ansible

В Ansible плейбуке, метод установки Astra Scylla можно настроить с помощью переменной Astra Scylla_install_method, которая задается в файле main.yml, расположенном в каталоге defaults. Переменная Astra Scylla_install_method поддерживает два значения:

- **"install_from_repo"** — установка Astra Scylla из репозитория. Этот метод автоматически настроит ваш сервер на использование корпоративного Sonatype Nexus Repository Astra Scylla, откуда будут устанавливаться необходимые пакеты.
- **"install_from_packages"** — установка Astra Scylla из локальных пакетов. Этот метод требует, чтобы у вас уже были скачаны необходимые пакеты на сервер, и установка будет производиться непосредственно из них.

Пример записи в main.yml:

```
Astra Scylla_install_method: "install_from_repo"  
или "install_from_packages"
```

2.5 Шаг 4: Запуск плейбука

Для запуска установки выполните команду:

```
ansible-playbook -i inventory.yml start.yml
```

2.6 Дополнительные шаги

После успешного завершения плейбука убедитесь, что все узлы работают корректно, проверив статус службы Astra Scylla:

```
systemctl status scylla-server
```

3 Проверка состояния кластера с помощью nodetool status

После завершения установки и настройки кластера Astra Scylla, вы можете проверить состояние всех узлов с помощью утилиты nodetool. Эта утилита входит в состав Astra Scylla и предоставляет различные команды для управления и мониторинга кластера.

3.1 Выполнение команды nodetool status

Команда **nodetool status** отображает информацию о текущем состоянии всех узлов в кластере. Для выполнения этой команды на одном из узлов введите:

```
nodetool status
```

3.2 Пример вывода команды nodetool status

Ниже приведен пример вывода команды nodetool status для трёхузлового кластера:

```
Datacenter: my_datacenter
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address          Load           Tokens         Owns          Host ID
Rack
UN  x.x.x.x           3.21 GB        256            33.3%        11111111-2222-3333-
4444-1
UN  x.x.x.x           3.14 GB        256            33.3%        66666666-7777-8888-
9999-2
UN  x.x.x.x           3.45 GB        256            33.3%        aaaaaaaaa-bbbb-cccc-
dddd-3
```

3.3 Разбор вывода команды nodetool status

- **Datacenter:** my_datacenter — имя центра обработки данных (ЦОД), указанное в конфигурации.
- **Status:** UN — показывает состояние узла:
 - U (Up) — узел работает.
 - N (Normal) — узел работает в обычном режиме.
 - Другие состояния могут включать D (Down), L (Leaving), J (Joining), и M (Moving).
- **Address:** IP-адрес узла.
- **Load:** Текущий объём данных, который хранится на узле.

- **Tokens:** Количество токенов, закреплённых за узлом. Обычно это 256 по умолчанию.
- **Owns:** Процент данных, которые принадлежат узлу в рамках распределённого хранилища.
- **Host ID:** Уникальный идентификатор узла.
- **Rack:** Номер стойки, к которой относится узел.

3.4 Анализ результатов

- Если все узлы отображаются со статусом UN, это означает, что кластер Astra Scylla работает корректно, и все узлы успешно подключены и активны.
- Если какой-либо узел имеет статус DN (Down/Normal), это указывает на проблему с доступностью этого узла. В этом случае необходимо проверить сетевое соединение, статус службы Astra Scylla на этом узле, а также логи для диагностики проблемы.

4 Описание плейбуков

- ``configure.yml``: Настраивает конфигурационные файлы для Astra Scylla, создаёт и копирует необходимые файлы.
- ``install_from_packages.yml``: Устанавливает Astra Scylla из локальных пакетов.
- ``install_from_repo.yml``: Устанавливает Astra Scylla из репозитория.
- ``service.yml``: Управляет службой Astra Scylla, включая настройки и запуск.
- ``handlers/main.yml``: Содержит обработчики для перезапуска, старта и остановки службы Astra Scylla.

5 Заключение

После развертывания и настройки кластера с помощью Ansible и запуска Astra Scylla, регулярная проверка состояния узлов с помощью `nodetool status` является ключевым шагом в мониторинге здоровья и производительности кластера. Этот инструмент позволяет вам оперативно обнаружить проблемы и принять необходимые меры для их устранения.

Если все узлы отображаются как работающие (UN), это означает, что ваш кластер функционирует корректно, и данные распределены между узлами равномерно.