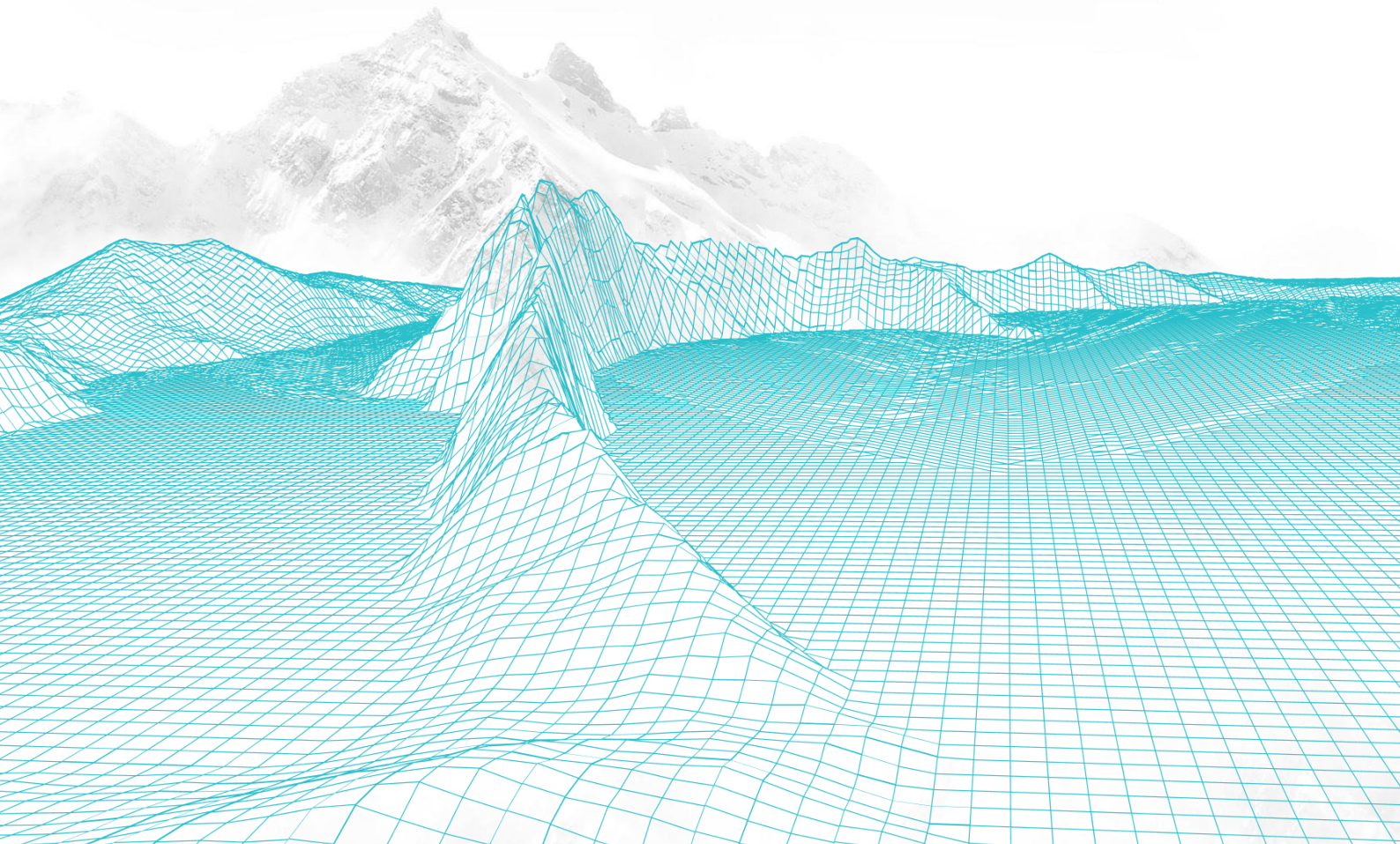
Программное обеспечение
FORTEIA

РУКОВОДСТВО ПО УСТАНОВКЕ

Москва 2026



Содержание

1. ВВЕДЕНИЕ

В данном документе описано развёртывание отказоустойчивого кластера FORTEIA на трёх серверах с установленной заранее ОС Ubuntu.

Все компоненты шлюза работают внутри единого контейнера. Образ контейнера с ПО FORTEIA поставляется на флеш-накопителе. Развёртывание выполняется с компьютера администратора по SSH.

Для корректной работы СХД, программное обеспечение FORTEIA должно быть установлено на каждом узле кластера.

2. ОПИСАНИЕ СИСТЕМЫ

Программное обеспечение FORTEIA разработано для использования в составе программно-аппаратных комплексов систем хранения данных.

Оно обеспечивает централизованное и надёжное хранение информации, а также по протоколу S3 предоставляет доступ к данным, размещённым на внешних файловых хранилищах, предоставленных через NFS (Network File System).

Высокая отказоустойчивость ПО достигается за счёт горизонтального масштабирования - повышение пропускной способности объектного хранилища путём добавления узлов. Надёжность хранения обеспечивается за счёт репликации объектов на несколько независимых NFS-папок.

3. УСТАНОВКА ПО FORTEIA

3.1. Требования к оборудованию и сети

3.1.1 Узлы кластера шлюза

Кластер состоит из трех серверов, работающих под операционной системой Ubuntu Server 22.04/24.04 LTS.

Конфигурация серверов:

- CPU: x86-64, 16+ ядер с частотой от 2ГГц;
- RAM: не менее 256 ГБ;
- SSD-диск объёмом не менее 512ГБ для операционной системы и логов;
- SSD-диск объёмом не менее 2 ТБ для хранения метаданных;
- Три сетевых интерфейса (или VLAN) с отдельными IP-адресами:
  - управляющая сеть (1 Гбит/с) – для SSH, GUI и внутреннего обмена между узлами;
  - фронтенд-сеть (100 Гбит/с) – для приёма S3-запросов от клиентов;
  - бэкенд-сеть (100 Гбит/с) – для подключения внешних NFS-серверов (СХД).

На серверах должны быть открыты следующие порты:

- 27017 – MongoDB (внутри управляющей сети)
- 8941 – REST API шлюза (доступ администратора, можно на управляющей сети)
- 8080 – S3 API (фронтенд-сеть)
- 8950 – Web-интерфейс
- 8085 – RPC между узлами (управляющая сеть)
- 2049 – NFS (бэкенд-сеть, доступ к внешним шарам)

3.1.2 Внешние NFS-серверы

Один или несколько NFS-серверов (СХД), экспортирующих каталоги, которые будут использоваться как бэкенд для хранения объектов.

На каждом NFS-сервере должны быть созданы экспорты (например, **/exports/export1**, **/exports/export2** и т.д.) с параметрами: **rw,sync,no_subtree_check,no_root_squash**.

NFS-серверы должны быть доступны по бэкенд-сети с узлов шлюза.

3.2. Подготовка серверов шлюза

На каждом из трёх узлов выполните следующие шаги:

1. Создайте пользователя (например, s3admin) и разрешите ему выполнять sudo без пароля:

sudo adduser s3admin

sudo usermod -aG sudo s3admin

echo "s3admin ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/s3admin

2. Откройте порты:

sudo ufw allow 27017/tcp

sudo ufw allow 8941/tcp

sudo ufw allow 8080/tcp

sudo ufw allow 8950/tcp

sudo ufw allow 8085/tcp

sudo ufw allow 2049/tcp

sudo ufw enable

3. Отформатируйте и смонтируйте диск /dev/sdb в каталог /mnt/s3data. В этом каталоге будут храниться метаданные.

- Проверьте имя диска:

  sudo lsblk

Перед созданием нового раздела убедитесь, что на диске /dev/sdb нет нужных данных. При необходимости очистите таблицу разделов:

sudo wipefs -a /dev/sdb

- Создайте раздел:

  sudo parted /dev/sdb mklabel gpt
  sudo parted /dev/sdb mkpart primary 0% 100%

- Отформатируйте раздел в ext4:

  sudo mkfs.ext4 /dev/sdb1

- Создайте точку монтирования:

  sudo mkdir -p /mnt/s3data

- Добавьте в /etc/fstab для автоматического монтирования:

  echo "/dev/sdb1 /mnt/s3data ext4 defaults 0 2" | sudo tee -a /etc/fstab

- Выполните монтирование файловой системы:

  sudo mount /mnt/s3data

- Создайте каталоги для метаданных:

  sudo mkdir -p /mnt/s3data/mongodb /mnt/s3data/tmp

- Назначьте пользователя s3admin владельцем каталога /mnt/s3data:

  sudo chown -R s3admin:s3admin /mnt/s3data

4. Настройте SSH-доступ с машины администратора:

- Сгенерируйте ключ:

  ssh-keygen -t rsa -b 4096 -f ~/.ssh/cluster_key -N ""

- Скопируйте публичный ключ на каждый узел:

  ssh-copy-id -i ~/.ssh/cluster_key.pub
  s3admin@<IP_управляющей_сети_узла>

- Проверьте доступ:

ssh -i ~/.ssh/cluster_key s3admin@<IP_управляющей_сети_узла> hostname

- Запишите IP-адреса каждого узла для доступа из управляющей сети – они потребуются в конфигурации MongoDB. Обозначим их как:

  NODE1_MGMT_IP=192.168.1.10
  NODE2_MGMT_IP=192.168.1.11
  NODE3_MGMT_IP=192.168.1.12

3.3. Установка Podman и загрузка образа

На каждом узле установите Podman и загрузите образ с флешки.

1. Установка Podman

sudo apt update sudo apt install -y podman

2. На администраторской машине подключите флешку с ПО FORTEIA и скопируйте файл s3-gateway.tar в каталог, например, ~/cluster-deploy/. Затем передайте его на каждый узел:

scp -i ~/.ssh/cluster_key ~/cluster-deploy/s3-gateway.tar s3admin@<IP_управляющей_сети_узла>:/tmp/

3. Загрузите образа в Podman на каждом узле:

podman load -i /tmp/s3-gateway.tar

Проверьте, что образ появился:

podman images

Запомните имя и тег (например, localhost/s3-gateway:latest). В дальнейшем используйте его в командах запуска.

3.4. Установка пакета nfs-common

sudo apt install -y nfs-common

3.5. Создание конфигурационных файлов s3gw.json

На каждом узле требуется свой файл конфигурации. Подготовьте три файла на администраторской машине (например, s3gw-node1.json, s3gw-node2.json, s3gw-node3.json).

Шаблон файла конфигурации (пример для node1 с IP адресом из управляющей сети - 192.168.1.10):

```
{
"CommonOptions": {
"ClusterStateUpdateRate": 3,
"ErrorMessages": "/etc/errors_msg.yaml",
"EventStoreTime": 5,
"FSMountPoint": "/var/swarm/fs",
"HealthMonitorConfig": "/etc/health_monitor.json",
"Id": 1,
"NodeType": 1,
"PWScoreMin": 0,
"ReplicasCount": 1
},
"DBOptions": {
"Hosts": [
{ "IP": "192.168.1.10", "Port": 27017 },
{ "IP": "192.168.1.11", "Port": 27017 },
{ "IP": "192.168.1.12", "Port": 27017 }
],
"Name": "swarm",
"ReplicaSet": "rs0",
"ServerSelectionMS": 2000,
"SocketTimeoutMS": 5000,
"WTimeoutMS": 4000,
"ClientPoolMaxSize": 300
```

```
},
"LogManagerOptions": {
"FillSpeedBorder": 512,
"FillSpeedCheckInterval": 600,
"LogStorageDays": 30,
"MaxLogUse": 26843545600,
"Facility": "all",
"AuditLogStorageDays": 90
},
"MetricsOptions": {},
"NetOptions": {
"HostId": "192.168.1.10",
"RPCPort": 8085
},
"NodeOptions": { "Hosts": [] },
"Notification": { "RemoveRecordTimeout": 864000 },
"RESTOptions": {
"Host": { "IP": "0.0.0.0", "Port": 8941 }
},
"S3Options": {
"BlockSize": 10485760,
"Host": { "IP": "0.0.0.0", "Port": 8080 },
"LogDeliveryPeriodS": 3600,
"MaxThreads": 32,
"UseSSL": false,
"S3FSEntryPoint": "/var/swarm/fs",
"S3IOType": "fs",
"SelfDiagnosticInterval": 5
},
"AuthOptions": {
"TokenLifetime": 86400,
"SessionLifetime": 86400,
```

"MaxSessionCount": 100,

"PWScoreMin": 0

}

}

Для node2 (IP 192.168.1.11) измените:

"Id": 2

"HostId": "192.168.1.11"

Остальное без изменений.

Для node3 (IP 192.168.1.12) измените:

"Id": 3

"HostId": "192.168.1.12"

Остальное без изменений.

Скопируйте файлы на соответствующие узлы:

scp -i ~/.ssh/cluster_key s3gw-node1.json s3admin@192.168.1.10:/tmp/s3gw.json

scp -i ~/.ssh/cluster_key s3gw-node2.json s3admin@192.168.1.11:/tmp/s3gw.json

scp -i ~/.ssh/cluster_key s3gw-node3.json s3admin@192.168.1.12:/tmp/s3gw.json

На каждом узле переместите файл в каталог данных и установите права:

ssh -i ~/.ssh/cluster_key s3admin@192.168.1.10 "sudo mv /tmp/s3gw.json /mnt/s3data/s3gw.json && sudo chown s3admin:s3admin /mnt/s3data/s3gw.json"

Повторите для остальных узлов.

3.6. Запуск контейнеров

Если внутри контейнера процесс работает не от root, убедитесь, что UID пользователя в контейнере соответствует UID пользователя s3admin на хосте. Проверить это можно командой: **podman exec s3gw id**

На каждом узле запустите контейнер с параметрами:

--network host – использует сеть хоста (все интерфейсы доступны).
-v /mnt/s3data/mongodb → /var/lib/mongodb (данные MongoDB).
-v /mnt/s3data/tmp → /var/swarm/fs (временная файловая система, позже сюда будут монтироваться NFS).
-v /mnt/s3data/s3gw.json → /var/swarm/fs/s3gw.json (конфигурация).

Пример команды (на каждом узле, от пользователя s3admin):

podman run -d
--name s3gw
--network host
--privileged
--pids-limit=0
--security-opt=seccomp=unconfined
-v /mnt/s3data/mongodb:/var/lib/mongodb
-v /mnt/s3data/tmp:/var/swarm/fs
-v /mnt/s3data/s3gw.json:/var/swarm/fs/s3gw.json
localhost/s3-gateway:latest

Замените **localhost/s3-gateway:latest** на реальное имя образа.

Проверьте запуск **podman ps.**

3.7. Устранение неполадок с контейнером

Если что-то пошло не так:

1. Остановить и удалить контейнер:

podman stop s3gw && podman rm s3gw

2. Очистить данные MongoDB (если требуется):

sudo rm -rf /mnt/s3data/mongodb/*

3. Повторить запуск контейнера.

3.8. Инициализация реплика-сета MongoDB

После запуска всех трёх контейнеров подключитесь к MongoDB на первом узле (node1) и инициализируйте реплика-сет.

ssh -i ~/.ssh/cluster_key s3admin@192.168.1.10

podman exec -it s3gw mongosh --port 27017

В оболочке выполните:

```
rs.initiate({
_id: "rs0",
members: [
{ _id: 0, host: "192.168.1.10:27017" },
{ _id: 1, host: "192.168.1.11:27017" },
{ _id: 2, host: "192.168.1.12:27017" }
]
})
```

Через 10–15 секунд проверьте статус:

rs.status()

Убедитесь, что один узел стал PRIMARY, два других – SECONDARY.