

# Действия после сбоев и ошибок при эксплуатации

## 1. Типовые ошибки StarVault

Ошибка при входе в StarVault `Authentication failed: ldap operation failed: failed to bind as user` может возникать в следующих ситуациях:

- ошибка в учетных данных сервисной учетной записи, с помощью которой выполняется подключение к службе каталогов и поиск пользователя в каталоге LDAP-сервера. Проверьте, что параметры сервисной учетной записи указаны верно, а в логах LDAP-сервера фиксируется успешный вход данной учетной записи;
- ошибка в учетных данных учетной записи, с помощью которой вы выполняете вход. Проверьте, что учетная запись активна, пароль установлен, не просрочен и не требует замены;
- ошибка в настроенных фильтрах, по которым выполняется поиск пользователя. Проверьте настроенные фильтры. Для локализации проблемы с фильтром рекомендуется использовать в тестовых целях как можно более простой фильтр без комплексных условий. Информационное сообщение после входа в StarVault `no LDAP groups found in groupDN; only policies from locally-defined groups available` может возникать в следующих ситуациях:
- ошибка в настроенных фильтрах, по которым выполняется поиск групп и определение принадлежности пользователя к группам. Проверьте настроенные фильтры. Для локализации проблемы с фильтром рекомендуется использовать в тестовых целях как можно более простой фильтр без комплексных условий.

## 2. Проверка конфигурации фильтров LDAP

Для диагностики фильтров LDAP удобно использовать утилиту `ldapsearch`. С помощью данной утилиты вы можете конструировать и направлять запросы к LDAP-серверу. В ответе LDAP-сервера вы можете проверить, насколько возвращаемая информация соответствует используемым вами фильтрам:

1. Проверьте, что в вашем каталоге пользователей (User DN) доступны все необходимые пользователи:

```
ldapsearch -x -LLL -h '<Имя или IP-адрес LDAP-сервера>' \
-p '<Порт LDAP-сервера>' \
-D '<имя сервисного аккаунта>' \
-w '<пароль сервисного аккаунта (Bindpass)>' \
-b "<Каталог пользователей (User DN)>" \
'<Атрибут имени пользователя (User Attribute)>' \
distinguishedName \
-s sub "(objectClass=user)"
```

BASH | 

2. Выберите из списка любого пользователя и используйте его атрибут `distinguishedName` для дальнейших проверок. Получите список групп, в которые входит данный пользователь:

```
ldapsearch -x -LLL -h '<Имя или IP-адрес LDAP-сервера>:<Порт LDAP-сервера>' \
\
-D '<имя сервисного аккаунта>' \
-w '<пароль сервисного аккаунта (Bindpass)>' \
-b "<Каталог групп (Group DN)>" \
'<Фильтр групп (Group Filter)>'
```

BASH | 

## 3. Восстановление Etcd

В данном разделе представлены процедуры восстановления работоспособности кластера Etcd.

Повреждение данных Etcd может произойти по многочисленным причинам, например:

- При выходе из строя узлов виртуализации, повреждении серверного оборудования или его компонентов, программных сбоев или потери сетевой связанности.
- При отказе программных и аппаратных систем хранения данных и ошибках записи данных на файловую систему.
- При **жестком** отключении мастер-узлов.
- При удалении критически важных данных Etcd.

Как правило, при повреждении базы данных сервис Etcd на узле переходит из состояния `active` в состояние `activating / failed`, а в системе мониторинга регистрируются события `etcdMembersDown`. При сохранении кворума кластера Etcd сервер Nova API продолжает работать, поскольку выполняет на своей стороне балансировку запросов. При этом после повреждения данных мастер-узел не сможет самостоятельно подключиться к прежнему кластеру, и может потребоваться выполнить его восстановление.

После оценки проблем с кластером Etcd воспользуйтесь наиболее подходящей процедурой восстановления или обратитесь в техническую поддержку.

## 3.1. Общие практики

Etcd не всегда требует наличия резервной копии (снапшота) для восстановления одного участника кластера.

Существует два основных сценария решения проблем в зависимости от того, что произошло с участником кластера Etcd:

- **Перезапуск неисправного участника кластера/** Если участник кластера перестал отвечать на запросы, сервис Etcd находится в состоянии `activating /failed` или `active`, но при этом не отвечает, то необходимо начать решение проблемы с перезапуска сервиса Etcd на узле. После перезапуска узел должен автоматически добавиться в существующих кластер и синхронизировать свое состояние с остальными участниками.
- **Восстановление неисправного участника кластера.** Если проблема не решается перезапуском сервиса Etcd, а данные Etcd повреждены, вы можете полностью восстановить узел. Для этого вам необходимо удалить неисправного участника из кластера Etcd, удалить поврежденные данные и передобавить участника в кластер. После подключения восстановленный участник кластера получит актуальную копию данных и синхронизирует свое состояние с остальными участниками.

Восстановление полного кластера Etcd из резервной копии необходимо в том случае, когда 2 и более участников кластера неисправны. В данной ситуации также перестает отвечать сервер Nova API, поэтому важно иметь возможность получить резервные копии Etcd независимо от работоспособности Nova API. Данный сценарий требует полной остановки сервисов Etcd на каждом мастер-узле, восстановления резервной копии на каждый мастер-узел и повторный запуск кластера Etcd.

При полном восстановлении кластера Etcd выполняется откат всех данных Nova на время последней успешной резервной копии. При этом все клиенты Nova (kubelet, CNI, CSI и др.) могут испытывать проблемы и конфликты с локальными данными при подключении к восстановленному кластеру. Это означает, что после успешного восстановления и запуска кластера Etcd могут потребоваться ручные действия для дальнейшего восстановления работоспособности Nova.

## 3.2. Восстановление одного неисправного узла Etcd

В данном разделе описывается процесс восстановления отдельного неисправного участника кластера Etcd. Процедура подразумевает, что кластер Nova остается работоспособным, однако один из участников кластера Etcd перестает отвечать на запросы.

### Необходимые условия

- ✓ У вас есть доступ к кластеру с учетной записью, имеющей роль **cluster-admin** в Nova.
- ✓ Вы установили утилиту `kubectl` для работы с Nova.

✓ У вас есть доступ по SSH на мастер-узлы кластера.

## Процедура

1. Подключитесь по SSH на один из исправных мастер-узлов.
2. Установите переменные окружения для подключения к Etcd:

```
source <(cat /etc/etcd.env | grep -v "^#\|^$" | sed 's/^/export /')
```

BASH | 

- ### 3. Определите неисправный узел Etcd.

```
/opt/etcd/etcdctl endpoint health --cluster --write-out=table
```

BASH | 

### Пример вывода:

```
{
  "level": "warn",
  "ts": "2024-06-14T12:22:37.426365+0300",
  "logger": "client",
  "caller": "v3@v3.5.10/retry_interceptor.go:62",
  "msg": "retrying of unary invoker failed",
  "target": "etcd-endpoints://0xc000325340/172.31.100.105:2379",
  "attempt": 0,
  "error": "rpc error: code = DeadlineExceeded desc = latest balancer error: last connection error: connection error: desc = \"transport: Error while dialing: dial tcp 172.31.100.105:2379: connect: connection refused\""
}
```

BASH | 

```
+-----+
|          ENDPOINT          | HEALTH |      TOOK      |          ERROR          |
+-----+-----+-----+-----+
| https://172.31.100.97:2379 |   true |    7.856832ms   |                        |
| https://172.31.100.106:2379 |   true |    8.094994ms   |                        |
| https://172.31.100.105:2379 | false | 5.000964147s   | context deadline       |
| exceeded                   |        |                  |                         |
+-----+-----+-----+-----+
Error: unhealthy cluster
```

В примере выше неисправным является узел, имеющий адрес подключения `https://172.31.100.105:2379`.

- #### 4. Получите дополнительные сведения о неисправном узле Etcd:

```
/opt/etcd/etcdctl member list --write-out=table
```

BASH | [ ]

### Пример вывода:

```
/opt/etcd/etcdctl member list --write-out=table
```

BASH | 

```
+-----+-----+-----+-----+
|          ID          | STATUS | NAME |          PEER ADDRS          |
| CLIENT ADDRS        | IS LEARNER |      |
+-----+-----+-----+-----+
| 51fc1c73bfe7f25f | started | etcd3 | https://172.31.100.106:2380 |
https://172.31.100.106:2379 |      false |      |
| 56be76cb75b1242d | started | etcd1 | https://172.31.100.97:2380  |
https://172.31.100.97:2379  |      false |      |
| e42bdaa6ace5c691 | started | etcd2 | https://172.31.100.105:2380 |
https://172.31.100.105:2379 |      false |      |
+-----+-----+-----+-----+
```

Сохраните информацию о неисправном узле:

- ID - идентификатор узла;
- NAME - имя узла;
- PEER ADDR - адрес подключения узла.

5. Удалите неисправный узел из кластера Etcd:

```
/opt/etcd/etcdctl member remove <ID>
```

BASH | 

Пример вывода:

```
/opt/etcd/etcdctl member remove e42bdaa6ace5c691
```

BASH | 

```
Member e42bdaa6ace5c691 removed from cluster dae05cd826ec589
```

6. Добавьте неисправный узел повторно в кластер:

```
/opt/etcd/etcdctl member add <NAME> --peer-urls=<PEER ADDR>
```

BASH | 

Пример вывода:

```
/opt/etcd/etcdctl member add etcd2 --peer-urls=https://172.31.100.105:2380
```

BASH | 

```
Member 4cafbd8e176044a1 added to cluster dae05cd826ec589
```


```
ETCD_NAME="etcd2"
```

```
ETCD_INITIAL_CLUSTER="etcd2=https://172.31.100.105:2380,etcd3=https://172.31.100.106:2380,etcd1=https://172.31.100.97:2380"
```

```
ETCD_INITIAL_ADVERTISE_PEER_URLS="https://172.31.100.105:2380"
ETCD_INITIAL_CLUSTER_STATE="existing"
```

## 7. Проверьте список участников кластера Etcd:

```
/opt/etcd/etcdctl member list --write-out=table
```

BASH | 

### Пример вывода:

```
/opt/etcd/etcdctl member list --write-out=table
```

BASH | 

ID	STATUS	NAME	PEER ADDRS
CLIENT ADDRS	IS LEARNER		
4cafbdb8e176044a1	unstarted		https://172.31.100.105:2380
false			
51fc1c73bfe7f25f	started	etcd3	https://172.31.100.106:2380
https://172.31.100.106:2379	false		
56be76cb75b1242d	started	etcd1	https://172.31.100.97:2380
https://172.31.100.97:2379	false		

Вновь добавленный узел находится в состоянии `unstarted`.

## 8. Подключитесь на неисправный узел по SSH и проверьте статус сервиса Etcd:

```
systemctl status etcd
```

BASH | 

### Пример вывода:

```
● etcd.service - etcd
Loaded: loaded (/etc/systemd/system/etcd.service; enabled; vendor preset: disabled)
Active: activating (auto-restart) (Result: exit-code) since Thu 2024-06-13 19:01:42 MSK; 8s ago
Process: 243655 ExecStart=/opt/etcd/etcd (code=exited, status=2)
Main PID: 243655 (code=exited, status=2)
```

BASH | 

## 9. Проверьте журнал событий сервиса Etcd на неисправном узле:

```
journalctl -u etcd
```

BASH | 

### Пример вывода:

```
journalctl -u etcd
```

BASH | 

```
Jun 13 19:03:14 nova-master-3.nova.internal etcd[244118]: \
{"level":"info","ts":"2024-06-13T19:03:14.605421+0300","caller":"embed/etcd.go:309","msg":"starting an etcd server","etcd-version":"3.5.10","git-sha"> Jun 13 19:03:14
nova-master-3.nova.internal etcd[244118]: \
{"level":"warn","ts":"2024-06-13T19:03:14.605518+0300","caller":"fileutil/fileutil.go:53","msg":"check file permission","error":"directory "/var/lib> Jun 13 19:03:14
nova-master-3.nova.internal etcd[244118]: \
{"level":"panic","ts":"2024-06-13T19:03:14.607293+0300","caller":"backend/backend.go:189","msg":"failed to open database","path":"/var/lib/etcd/membe> Jun 13 19:03:14 nova-
master-3.nova.internal etcd[244118]: panic: failed to open database Jun 13
19:03:14 nova-master-3.nova.internal etcd[244118]: goroutine 110 [running]:
Jun 13 19:03:14 nova-master-3.nova.internal etcd[244118]:
go.uber.org/zap/zapcore.(*CheckedEntry).Write(0xc000108840, \{0xc000532580,
0x2, 0x2\}) Jun 13 19:03:14 nova-master-3.nova.internal etcd[244118]:
go.uber.org/zap@v1.17.0/zapcore/entry.go:234 +0x49b
```

10. Остановите сервис Etcd на неисправном узле:

```
systemctl stop etcd
```

BASH | 

11. Удалите директорию с данными Etcd на неисправном узле:

```
rm -rf /var/lib/etcd
```

BASH | 

12. Измените параметр состояния кластера для запуска сервиса Etcd:

```
sed -i
's/ETCD_INITIAL_CLUSTER_STATE=new/ETCD_INITIAL_CLUSTER_STATE=existing/g'
/etc/etcd.env
```

BASH | 

13. Запустите сервис Etcd на неисправном узле:

```
systemctl start etcd
```

BASH | 

14. Проверьте состояние сервиса Etcd:

```
systemctl status etcd
```

BASH | 

Пример вывода:

```
● etcd.service - etcd
Loaded: loaded (/etc/systemd/system/etcd.service; enabled; vendor preset:
disabled)
```


BASH | 

```
Active: active (running) since Fri 2024-06-14 12:37:55 MSK; 30s ago
Main PID: 96408 (etcd)
    Tasks: 9 (limit: 102296)
Memory: 221.6M
CGroup: /system.slice/etcd.service
        └─96408 /opt/etcd/etcd

Jun 14 12:37:55 nova-master-2.nova.internal etcd[96408]:
{"level":"info","ts":"2024-06-14T12:37:55.065848+0300","caller":"etcdmain/main.go:44","msg":"notifying init daemon"}
Jun 14 12:37:55 nova-master-2.nova.internal etcd[96408]:
{"level":"info","ts":"2024-06-14T12:37:55.065967+0300","caller":"etcdmain/main.go:50","msg":"successfully notified init daemon"}
Jun 14 12:37:55 nova-master-2.nova.internal systemd[1]: Started etcd.
```

Статус сервиса Etcd должен быть `active`.

15. Проверьте список участников кластера Etcd:

```
/opt/etcd/etcdctl member list --write-out=table BASH | 
```

ID	STATUS	NAME	PEER ADDRS	CLIENT ADDRS	IS LEARNER
4cafbdb8e176044a1	started	etcd2	https://172.31.100.105:2380	https://172.31.100.105:2379	false
51fc1c73bfe7f25f	started	etcd3	https://172.31.100.106:2380	https://172.31.100.106:2379	false
56be76cb75b1242d	started	etcd1	https://172.31.100.97:2380	https://172.31.100.97:2379	false

Вновь добавленный узел находится в состоянии `started`.

16. Проверьте статус участников кластера Etcd:

```
/opt/etcd/etcdctl endpoint status --cluster --write-out=table BASH | 
```

Пример вывода:

```
/opt/etcd/etcdctl endpoint status --cluster --write-out=table BASH | 
```

Endpoint	Health	Latency
172.31.100.105:2379	healthy	0.0001s
172.31.100.106:2379	healthy	0.0001s
172.31.100.97:2379	healthy	0.0001s



ENDPOINT			ID	VERSION	DB SIZE	IS
LEADER	IS LEARNER	RAFT TERM	RAFT INDEX	RAFT APPLIED INDEX	ERRORS	
https://172.31.100.105:2379	4cafbdb8e176044a1	3.5.10	151 MB			
false	false	5	376222	376222		
https://172.31.100.106:2379	51fc1c73bfe7f25f	3.5.10	151 MB			
false	false	5	376222	376222		
https://172.31.100.97:2379	56be76cb75b1242d	3.5.10	151 MB			
true	false	5	376222	376222		

## 4. Восстановление кластера узлов Etcd

В данном разделе описывается процесс восстановления полного кластера Etcd. Процедура подразумевает, что кластер Nova не работоспособен, а сервер Nova API не отвечает на запросы. Вам потребуется наличие актуальной резервной копии для восстановления кластера в прежнее состояние.



Восстановление кластера Etcd в исходное состояние с помощью резервной копии является крайним случаем при восстановлении работоспособности Nova. Если сервер Nova API отвечает на запросы, значит Etcd доступен и полное восстановление не требуется. Достаточно восстановить неисправного участника кластера.

### Необходимые условия

- ✓ У вас есть доступ к кластеру с учетной записью, имеющей роль cluster-admin в Nova.
- ✓ Вы установили утилиту kubectl для работы с Nova.
- ✓ У вас есть доступ по SSH на мастер-узлы кластера.
- ✓ У вас есть актуальная резервная копия мастер-узлов.
- ✓ У вас есть ключ шифрования резервных копий, если резервное копирование выполнялось с помощью модуля Data Protection.



Если вы используете модуль Data Protection, то восстановите актуальную резервную копию Etcd из последней доступной резервной копии с помощью Kopia следуя процедуре Восстановление резервных копий мастер-узлов). Если резервные копии хранятся на NFS-хранилище, то вам будет необходимо скопировать их на каждый мастер-узел.

При восстановлении кластера Etcd вам потребуется открыть 3 SSH-сессии (по одной на каждый мастер-узел).

В качестве примера в руководстве используются следующие имена и IP-адреса мастер-узлов:

NAME	InternalIP
nova-master-1.nova.internal	172.31.100.97
nova-master-2.nova.internal	172.31.100.105
nova-master-3.nova.internal	172.31.100.106

Резервная копия Etcd располагается на каждом мастер-узле в директории /tmp.

## Процедура

1. Подключитесь по SSH к каждому мастер-узлу кластера Nova.
2. Проверьте статус сервисов Etcd на каждом мастер-узле и остановите сервис, если он запущен:

```
systemctl status etcd  
systemctl stop etcd
```

BASH | 

### Пример вывода:

```
● etcd.service – etcd  
Loaded: loaded (/etc/systemd/system/etcd.service; enabled; vendor preset:  
disabled)  
Active: activating (auto-restart) (Result: exit-code) since Mon 2024-06-17  
16:47:47 MSK; 2s ago  
Process: 134875 ExecStart=/opt/etcd/etcd (code=exited, status=2)  
Main PID: 134875 (code=exited, status=2)
```

BASH | 

3. Убедитесь, что параметр состояния кластера ETCD\_INITIAL\_CLUSTER\_STATE имеет значение new на каждом мастер-узле:

```
cat /etc/etcd.env | grep ETCD_INITIAL_CLUSTER_STATE  
  
ETCD_INITIAL_CLUSTER_STATE=new
```

BASH | 

Если значение отличается, измените его на new в файле /etc/etcd.env .

4. Установите переменные окружения для подключения к Etcd на первом мастер-узле:

```
source <(cat /etc/etcd.env | grep -v "^#\|^$" | sed 's/^/export /')
```

BASH | 

5. Удалите директорию с данными Etcd на каждом мастер-узле:

```
rm -rf /var/lib/etcd
```

BASH | 

## 6. Распакуйте резервную копию Etcd на каждом мастер-узле:

```
cd /tmp/ && tar -zxvf /tmp/etcd_snapshot_nova-v5.1.1_k8s-v1.27.11_2024-06-14_081538.db.tar.gz
```

BASH |

Достаточно использовать один снапшот Etcd и скопировать его на каждый мастер-узел.

## 7. Выполните восстановление снапшота Etcd на каждом мастер-узле:

```
ETCDCTL_API=3 /opt/etcd/etcdctl snapshot restore \
  --data-dir=/var/lib/etcd \
  --name=<MEMBER_NAME> \
  --initial-cluster=etcd1=https://<MEMBER_1_IP_ADDRESS>:2380,etcd2=https://<MEMBER_2_IP_ADDRESS>:2380,etcd3=https://<MEMBER_3_IP_ADDRESS>:2380 \
  --initial-cluster-token=k8s_etcd \
  --initial-advertise-peer-urls=https://<MEMBER_1_IP_ADDRESS>:2380 \
  /tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-v1.27.11_2024-06-14_081538.db
```

BASH |

MEMBER\_X\_IP\_ADDRESS - IP-адрес мастер-узла.

Пример команды для первого мастер-узла:


```
ETCDCTL_API=3 /opt/etcd/etcdctl snapshot restore \
  --data-dir=/var/lib/etcd \
  --name=etcd1 \
  --initial-cluster=etcd1=https://172.31.100.97:2380,etcd2=https://172.31.100.105:2380,etcd3=https://172.31.100.106:2380 \
  --initial-cluster-token=k8s_etcd \
  --initial-advertise-peer-urls=https://172.31.100.97:2380 \
  /tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-v1.27.11_2024-06-14_081538.db
```

BASH |  

```
2024-06-19T20:02:10+03:00 info snapshot/v3_snapshot.go:260
restoring snapshot {"path": "/tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-v1.27.11_2024-06-14_081538.db", "wal-dir": "/var/lib/etcd/member/wal",
"data-dir": "/var/lib/etcd", "snap-dir": "/var/lib/etcd/member/snap"}
2024-06-19T20:02:11+03:00 info membership/store.go:141 Trimming
membership information from the backend...
2024-06-19T20:02:11+03:00 info membership/cluster.go:421 added
member {"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-peer-id": "4c357ea1a12149d0", "added-peer-peer-urls":
["https://172.31.100.97:2380"]}
2024-06-19T20:02:11+03:00 info membership/cluster.go:421 added
member {"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-peer-id": "e42bdaa6ace5c691", "added-peer-peer-urls":
["https://172.31.100.105:2380"]}
```


```
2024-06-19T20:02:11+03:00 info membership/cluster.go:421 added
member {"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-
peer-id": "e7b44abd88b7fad0", "added-peer-peer-urls":
["https://172.31.100.106:2380"]}
2024-06-19T20:02:11+03:00 info snapshot/v3_snapshot.go:287 restored
snapshot {"path": "/tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-
v1.27.11_2024-06-14_081538.db", "wal-dir": "/var/lib/etcd/member/wal",
"data-dir": "/var/lib/etcd", "snap-dir": "/var/lib/etcd/member/snap"}
```

Пример команды для второго мастер-узла:

```
BASH | 
ETCDCTL_API=3 /opt/etcd/etcdctl snapshot restore \
  --data-dir=/var/lib/etcd \
  --name=etcd2 \
  --initial-
cluster=etcd1=https://172.31.100.97:2380,etcd2=https://172.31.100.105:2380,e
tcd3=https://172.31.100.106:2380 \
  --initial-cluster-token=k8s_etcd \
  --initial-advertise-peer-urls=https://172.31.100.105:2380 \
  /tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-v1.27.11_2024-06-
14_081538.db
```

```
2024-06-19T20:02:48+03:00 info snapshot/v3_snapshot.go:260
restoring snapshot {"path": "/tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-
v1.27.11_2024-06-14_081538.db", "wal-dir": "/var/lib/etcd/member/wal",
"data-dir": "/var/lib/etcd", "snap-dir": "/var/lib/etcd/member/snap"}
2024-06-19T20:02:48+03:00 info membership/store.go:141 Trimming
membership information from the backend...
2024-06-19T20:02:49+03:00 info membership/cluster.go:421 added
member {"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-
peer-id": "4c357ea1a12149d0", "added-peer-peer-urls":
["https://172.31.100.97:2380"]}
2024-06-19T20:02:49+03:00 info membership/cluster.go:421 added
member {"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-
peer-id": "e42bdaa6ace5c691", "added-peer-peer-urls":
["https://172.31.100.105:2380"]}
2024-06-19T20:02:49+03:00 info membership/cluster.go:421 added
member {"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-
peer-id": "e7b44abd88b7fad0", "added-peer-peer-urls":
["https://172.31.100.106:2380"]}
2024-06-19T20:02:49+03:00 info snapshot/v3_snapshot.go:287 restored
snapshot {"path": "/tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-
v1.27.11_2024-06-14_081538.db", "wal-dir": "/var/lib/etcd/member/wal",
"data-dir": "/var/lib/etcd", "snap-dir": "/var/lib/etcd/member/snap"}
```

Пример команды для третьего мастер-узла:

```
BASH | 
ETCDCTL_API=3 /opt/etcd/etcdctl snapshot restore \
  --data-dir=/var/lib/etcd \
  --name=etcd3 \
```

```

--initial-
cluster=etcd1=https://172.31.100.97:2380,etcd2=https://172.31.100.105:2380,e
tcd3=https://172.31.100.106:2380 \
--initial-cluster-token=k8s_etcd \
--initial-advertise-peer-urls=https://172.31.100.106:2380 \
/tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-v1.27.11_2024-06-
14_081538.db

2024-06-19T20:04:01+03:00  info    snapshot/v3_snapshot.go:260 restoring
snapshot {"path": "/tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-
v1.27.11_2024-06-14_081538.db", "wal-dir": "/var/lib/etcd/member/wal",
"data-dir": "/var/lib/etcd", "snap-dir": "/var/lib/etcd/member/snap"}
2024-06-19T20:04:02+03:00  info    membership/store.go:141 Trimming
membership information from the backend...
2024-06-19T20:04:02+03:00  info    membership/cluster.go:421  added member
{"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-peer-id":
"4c357ea1a12149d0", "added-peer-peer-urls": ["https://172.31.100.97:2380"]}
2024-06-19T20:04:02+03:00  info    membership/cluster.go:421  added member
{"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-peer-id":
"e42bdaa6ace5c691", "added-peer-peer-urls": ["https://172.31.100.105:2380"]}
2024-06-19T20:04:02+03:00  info    membership/cluster.go:421  added member
{"cluster-id": "dae05cd826ec589", "local-member-id": "0", "added-peer-id":
"e7b44abd88b7fad0", "added-peer-peer-urls": ["https://172.31.100.106:2380"]}
2024-06-19T20:04:02+03:00  info    snapshot/v3_snapshot.go:287 restored
snapshot {"path": "/tmp/opt/backup/etcd_snapshot_nova-v5.1.1_k8s-
v1.27.11_2024-06-14_081538.db", "wal-dir": "/var/lib/etcd/member/wal",
"data-dir": "/var/lib/etcd", "snap-dir": "/var/lib/etcd/member/snap"}

```

8. Запустите сервис Etcd на каждом мастер-узле подряд и проверьте его статус:

```

systemctl start etcd
systemctl status etcd

```

BASH | 

Пример вывода:

```

● etcd.service - etcd
Loaded: loaded (/etc/systemd/system/etcd.service; enabled; vendor preset:
disabled)
Active: active (running) since Wed 2024-06-19 20:05:17 MSK; 59s ago
Main PID: 134154 (etcd)
Tasks: 10 (limit: 102296)
Memory: 56.8M
CGroup: /system.slice/etcd.service
└─134154 /opt/etcd/etcd

```

BASH | 

Статус сервиса Etcd должен быть active.

9. Проверьте статус участников кластера Etcd:

```
/opt/etcd/etcdctl endpoint status --cluster --write-out=table
```

BASH | 

Пример вывода:

```
+-----+-----+-----+-----+-----+
|          ENDPOINT          |          ID          | VERSION | DB SIZE | IS LEADER | IS LEARNER | RAFT TERM | RAFT INDEX | RAFT APPLIED INDEX | ERRORS |
+-----+-----+-----+-----+-----+
| https://172.31.100.97:2379 | 4c357ea1a12149d0 | 3.5.10 | 151 MB | true | false | 2 | 6489 | 6489 | |
| https://172.31.100.105:2379 | e42bdaa6ace5c691 | 3.5.10 | 151 MB | false | false | 2 | 6489 | 6489 | |
| https://172.31.100.106:2379 | e7b44abd88b7fad0 | 3.5.10 | 151 MB | false | false | 2 | 6489 | 6489 | |
+-----+-----+-----+-----+-----+
```

BASH | 

На данном этапе восстановление кластера Etcd завершено. Далее необходимо выполнить проверку работоспособности среды Nova.

# Типы событий безопасности

В данном разделе приведены таблицы событий компонентов Nova, логи которых могут отправляться в OpenSearch:

## 1. Neuvector

Тип события	Событие	Визуализация события в OpenSearch
События безопасности	Обнаружение угроз	Да
	Нарушения политики сетевой безопасности	Нет
	Попытки несанкционированного доступа	Да
	Статистика по CVE	Да
Сетевая активность	Логи соединений между контейнерами	Нет
	Анализ поведения сетевого трафика	Нет
	Логи сканирования уязвимостей	Нет
События контейнеров	Запуск и остановка контейнеров	Да
	Ошибки контейнеров	Да
	Инциденты переполнения	Нет
Обновления политик	Создание и изменение обеспечительных политик	Нет
	Аудит изменений политик безопасности	Нет

## 2. StartVault Audit Log:

Тип события	Событие	Визуализация события в OpenSearch
-------------	---------	-----------------------------------

Аудит доступа	Вход и выход из системы пользователей	Да
	Изменение ролей и привилегий	Да
Системные события	Запуск/остановка сервисов	Да
	Логи обновлений системы	Да
Изменения конфигурации	Настройка системы и приложения	Нет
	Резервное копирование и восстановление данных	Нет
События безопасности	Нарушения попыток доступа	Да
	Изменения в политике безопасности	Да

### 3. Kubelet

Тип события	Событие	Визуализация события в OpenSearch
Операции управления	Запуск и завершение работы подов	Да
	Проблемы со статусами контейнеров	Да
Ошибки и предупреждения	Недостаток ресурсов (ЦП, память)	Нет
	Ошибки подключения к API серверу	Нет
Статистика производительности	Использование ресурсов подами	Нет
	Время запуска контейнеров	Нет
Информационные события	Регистрация узлов	Да
	Обновления конфигураций	Да

### 4. Audit Log Kubernetes

Тип события	Событие	Визуализация события в OpenSearch
-------------	---------	-----------------------------------



Пользовательские действия	Создание, обновление, удаление ресурсов (Pods, Deployments)	Нет
	Авторизация и аутентификация пользователей	Нет
Изменения в конфигурации	Обновления конфигурации кластера	Да
	Изменения политик безопасности сетей	Нет
События на уровне кластера	Разрешения и запреты на проведение операций	Да
	Доступ к секретам и конфигурационным данным	Да
События администрирования	Создание и удаление пользовательских ролей и привилегий	Да
	Управление ресурсами кластера	Да

## 5. Audit Log Scheduler

Тип события	Событие	Визуализация события в OpenSearch
События планирования	Pending Pods: Логи Pods, которые находятся в состоянии ожидания и по каким причинам их нельзя запланировать	Да
	Successful Schedules: успешно запланированные Pods и информация о том, на каком узле они были размещены	Да
Решения о привязке (Bindings):	Логи о привязке Pods к узлам, включая выбранные узлы и альтернативные варианты, которые не были выбраны	Да

Изменения в конфигурации	Изменения в политиках планировщика, таких как конфигурации affinities/anti-affinities, resource constraints и другие настройки	Нет
--------------------------	--	-----