Consul ?????? ???????????

Когда мы говорим о service mesh, в первую очередь следует упомянуть о Consul, разработанную HashiCorp.

Consul — это один из инструментов с открытым исходным кодом, который широко используется для обнаружения сервисов для нескольких эфемерных или неэфемерных ресурсов.

Но следует упомянуть о многих системах service mesh - Istio, Linkerd, Traefik Mesh, Open Service Mesh (OSM), Nginx Service Mesh (NSM), Kuma.

Consul выполняет не только роль service mesh (далее SM), но также у него есть богатый функционал, который отличает его от других сервисов.

- health checking
- load balancing
- Key-value store
- Web-UI

Сам по себе Consul представляет собой бинарник, который запускается с различными параметрами. Это позволяет оперативно развернуть consul.

Например, рассмотрим часть безопасности консула, которая не включена в настройку или конфигурацию базового консула. Итак, для безопасности мы упомянем ACL.

Consul ACL (Список контроля доступа) — это опция, которая была добавлена в consul версии 1.4.0 в соответствии с официальной документацией. Этот параметр очень важен с точки зрения безопасности, поскольку он обеспечивает несколько уровней безопасности, когда мы настраиваем или получаем доступ к консулу для нескольких упомянутых нами параметров. Давайте обсудим, как работает consul ACL. Он делит свою работу на две части:

1. Consul token

2. Policy

Токен консула используется в качестве механизма аутентификации между клиентом и сервером, что означает, что он предоставляет доступ только в том случае, если кто-то проходит аутентификацию с помощью действительного токена консула ACL. Каждый токен ведет себя как отдельный объект, когда мы пытаемся создать его на сервере. Когда мы настраиваем токен консула, консул запрашивает несколько вещей, таких как ограничение

параметра токена, описание и политика.

Политика является одним из важных аспектов списка контроля доступа консула после создания токена, поскольку она обеспечивает часть авторизации списка контроля доступа консула. Короче говоря, политика обеспечивает доступ или тип разрешения для любого токена ACL консула, а также сообщает, как будет вести себя конкретный токен ACL консула и его границы разрешений.

5ca7d4ba8060a6b5a6a419714914eb14.png

Существует несколько вариантов, предоставляемых консулом ACL, таких как роли консула, идентификаторы служб, идентификаторы узлов и методы привязки, которые представлены в разных версиях консула, но на данный момент мы рассматриваем только токены политики и консула для базовых требований аутентификации и авторизации.

Все параметры, предоставляемые консулом, охватываются консулом ACL, что означает, что политика консула ACL предоставляет границы разрешений для всех параметров, таких как ключ-значение, обнаружение службы, проверка работоспособности и т.д.

Политика consul ACL написана на языке HCL, который является основным языком большинства инструментов Hashicorp, таких как terraform, vault и Nomad. Политика Consul использует язык HCL для реализации политик для токенов. Ниже представлен формат создания политики Consul ACL:

```
<resource> "<segment>" {
policy = "<policy disposition>"
}
```

Давайте разберем этот формат и узнаем о каждой части этого формата политики Consul ACL.

Ресурс указывает тип опции, предоставляемой консулом, например сервис, ключ-значение.

Следующие правила можно настроить для нескольких ресурсов консула, таких как:
- acl
- agent
- event
- key
- keyring
- node
- operator
- query
- service
- session

Их даже больше, и список слишком велик, и да, он поддерживает почти все функции, которые предоставляет консул.

**** применим только для некоторых ресурсов консула, таких как ключ-значение, в противном случае для многих сервисов мы можем оставить его пустым, используя две двойные кавычки.

<policy disposition> означает, какой доступ ему нужен, например:
- read
- write
- list
- deny

Если мы рассмотрим все вышеперечисленные пункты, посмотрите пример ниже для получения дополнительных разъяснений:

key_prefix "r`edis/" {`
```
    policy = "read"
}
```

CONSUL ACL: BOOTSTRAPPING

Итак, чтобы убедиться, используйте последний консул. Чтобы проверить, поддерживает ли текущая версия консула в вашей системе ACL. Введите consul –version и проверьте, присутствует ли подкоманда для ACL.

```
$ consul --help
Usage: consul [--version] [--help] []
Available commands are:
acl Interact with Consul's ACLs
```

Также вы можете проверить веб-интерфейс, только если пользовательский интерфейс включен для консула. В разделе ACCESS CONTROLS нажмите Tokens (чтобы попасть на веб-интерфейс consul введите адрес: http://<consul_ip>:8500/

255b4839a3166cd5dfdfad0638f41f1c.png

Чтобы реализовать Consul ACL, он в основном состоит из двух шагов:

1. Включить ACL
2. Создание токена Bootstrap

Если мы попытаемся bootstrap token, он выдаст ошибку, что поддержка ACL отключена.

```
$ consul acl bootstrap
Failed ACL bootstrapping: Unexpected response code: 401 (ACL support disabled)
```

Чтобы включить ACL, необходимо добавить следующие строки в файл конфигурации consul.hcl или же создать файл с любым названием но с расширением .hcl и положить его в /etc/consul.d

```
acl = {
    enabled = true
    default_policy = "deny"
    enable_token_persistence = true
}
```

После добавления сохраните файл, перезапустите службу и используйте команду consul acl bootstrap, чтобы включить консул ACL.

Как только вы выполните команду вы получите токен начальной загрузки, который состоит из следующих идентификаторов/токенов.
- AccessorID
- SecretID

SecretID отвечает за аутентификацию, и этот secretID является загрузочным токеном, что означает, что это токен администратора, который может делать что угодно или имеет полные привилегии. Храните этот токен начальной загрузки в надежном месте. В противном случае любой может получить доступ или изменить что-либо, используя этот токен начальной загрузки.

Как только вы создадите загрузочный токен, это активирует consul ACL, для которого требуется аутентификация на основе токенов. Итак, используйте приведенную выше команду с умом.

Получите доступ к пользовательскому интерфейсу, используя URL-адрес консула, и он запросит вход в консул, потому что по умолчанию анонимный доступ отключен, что означает ни чтение, ни запись.

5f29ff46985b6699a91e458dbc97bb5b.png

a488a3d86c6d402fc5b18d46fb8d373a.png

После того, как вы вошли в систему, он предоставит вам доступ в соответствии с сгенерированным токеном и разрешением, прикрепленным к токену ACL.

030c71885d389178a06e41430c4a2b02.png

Вы можете создать несколько токенов консула и соответствующих политик.

Вышеуказанные шаги пользовательского интерфейса также можно выполнить с помощью CLI, что означает, что когда мы включаем ACL, он в основном добавляет механизм

аутентификации как для CLI, так и для пользовательского интерфейса. Итак, чтобы получить доступ или изменить ресурсы консула, нам нужно добавить токены консула соответственно.

Поскольку ACL были включены, вам потребуется использовать токен для выполнения любых дополнительных операций. Например, даже для проверки списка участников потребуется токен

```
$ consul members -token 6e2aff60-f047-3158-58ea-ae2b1afc24a6
Node  Address            Status  Type    Build   Protocol  DC    Partition  Segment
web1  192.168.0.21:8301  alive   server  1.11.1  2         home  default    <all>
```

Для облегчения работы с токеном можно вынести его в переменные, но это не рекомендуется из соображений безопасности

```
$ export CONSUL_HTTP_TOKEN=6e2aff60-f047-3158-58ea-ae2b1afc24a6
```

Обратите внимание, токен начальной загрузки можно создать только один раз, после создания токена начальной загрузки загрузка будет отключена. После начальной загрузки системы ACL токенами ACL можно управлять через API ACL

9c332a9d4e3fa90c297ff89901e68deb.png

Сброс ACL

Бывают ситуации когда теряется secretID и встает вопрос что же делать. Тут к нам на помощь приходит официальная документация касательно восстановления ACL.

Для начала нам надо определить какой из серверов является лидером (это не относится к схеме где у вас всего один сервер Consul)

```
$ curl 192.168.0.21:8500/v1/status/leader
```

Результатом выполнения команды станет адрес сервера лидера. Дальше нам надо определить номер индекса ACl необходимого для сброса

```
$ consul acl bootstrap
Failed ACL bootstrapping: Unexpected response code: 403 (Permission denied: ACL bootstrap no
longer allowed (reset index: 635911))
```

И в конце запишем индекс сброса в файл сброса начальной загрузки (в нашем случае индекс будет равен 635911

```
$ echo 635911 >> <consul-data-dir>/acl-bootstrap-reset
```

Результатом будет сброс всех токенов и возможность создать новый. Обязательно запиишите его.

Это очень краткий обзор consul ACL, потому что Consul ACL — очень обширная тема и имеет множество вариантов, но мы рассмотрели только некоторые аспекты. Мы обсудили, почему консул ACL важен для безопасности консула. Consul ACL состоит из двух частей: токена и политики, где токен используется в качестве механизма аутентификации, а политика используется в качестве механизма авторизации. Мы обсудили загрузку consul ACL с нуля, которая включает в себя несколько шагов и проверок. Политика Consul ACL предназначена не для предоставления доступа, а для отключения или отказа в доступе к ресурсу.

Revision #1
Created 2025-06-27 08:53:40 UTC by Антон Сергеевич Абраменко
Updated 2025-06-27 08:54:16 UTC by Антон Сергеевич Абраменко