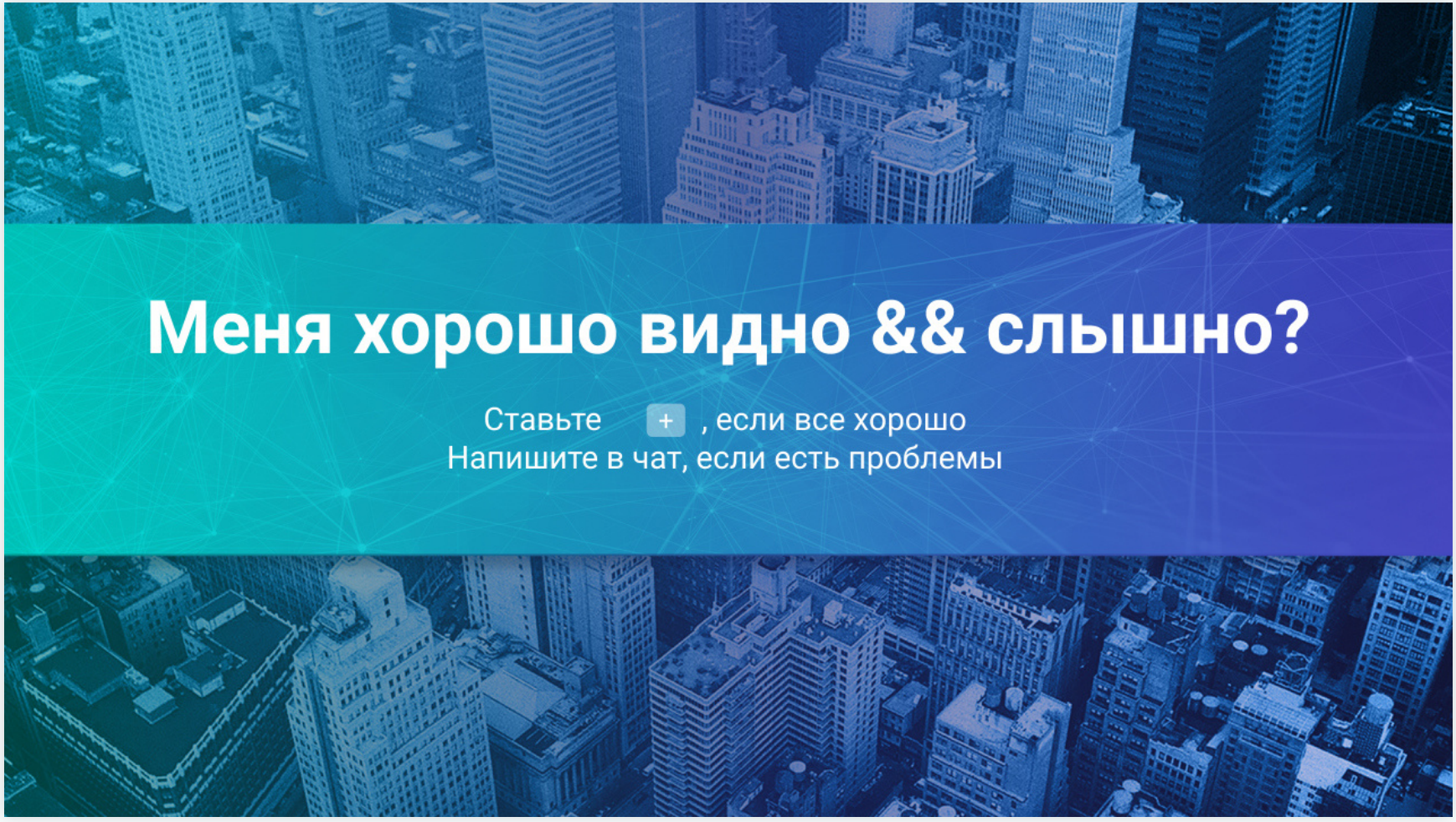

OTUS
ОНЛАЙН-ОБРАЗОВАНИЕ
Онлайн-образование


Меня хорошо видно && слышно?
Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

НЕ ЗАБЫТЬ ВКЛЮЧИТЬ ЗАПИСЬ!!!

Linux High Availability. HashiCorp Consul.

Цель занятия

- познакомиться с HashiCorp Consul;
- настроить DNS с healthcheck на базе HashiCorp Consul.

План занятия

HashiCorp Consul:

- какие задачи может решить?
- установка и настройка;
- кластеризация и федерация;
- резервное копирование и восстановление;
- обнаружение сервисов (service discovery) с помощью DNS;
- KV store и consul-template.

HashiCorp Consul

Consul - ПО для обеспечения совместной работы и конфигурирования приложений в изменяющейся, распределенной инфраструктуре.

- поддерживает высокую доступность "из коробки";
- поддерживает распределённые схемы развёртывания;
- готов к работе в схемах с несколькими ЦОД.

Официальный сайт: https://www.consul.io/
Начинать знакомство отсюда:
https://learn.hashicorp.com/collections/consul/getting-started

Чтобы что? Для чего?

ПО, созданное как реакция на проблемы, возникающие при переходе к микросервисной архитектуре, которой свойственны высокая динамичность и распределённость сервисов. Сервисов и их экземпляров может быть так много, что ручное управление становится неэффективно. Проблемы:

- обнаружение сервисов
- разрозненная конфигурация
- изоляция и сегментация трафика

Что же даёт нам Consul:

- service discovery (даёт "месторасположение" приложения);
- kv store (хранилище для конфигурации приложений);
- consul connect (какие сервисы к каким могут обращаться);
- encryption (все соединения могут быть защищены).

HashiCorp Consul

Фактически, представляет собою исполняемый бинарный файл (написан на Go). Для широкораспространённых дистрибутивов есть в виде подготовленных пакетов.

```
[mbfx@ht-p1-0 ~]$ rpm -qil consul
...
/etc/consul.d/consul.hcl
/usr/bin/consul
/usr/lib/systemd/system/consul.service
```

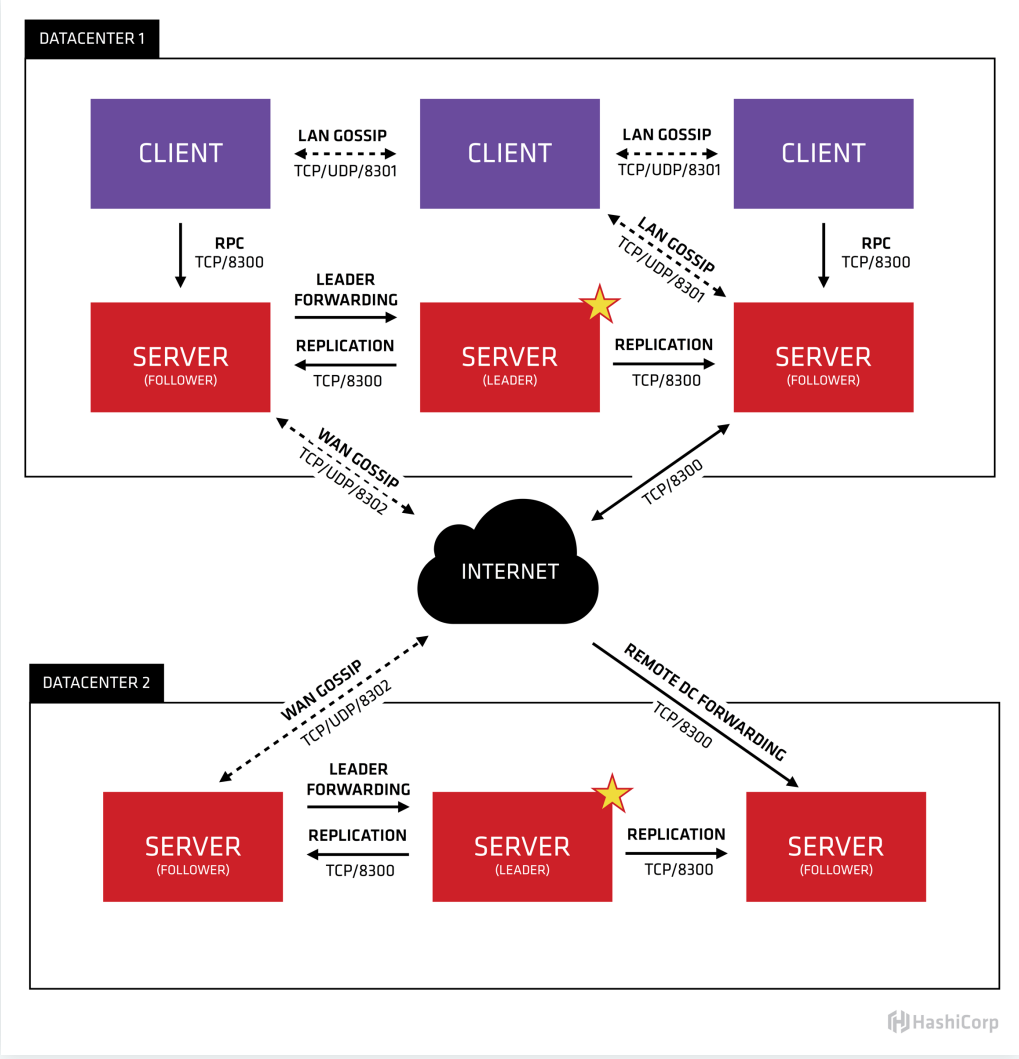
Скачать можно здесь: https://www.consul.io/downloads.html
Или собрать руками: https://www.consul.io/docs/install#compiling-from-source

```
consul agent -dev -data-dir=/tmp/consul
```

Терминология

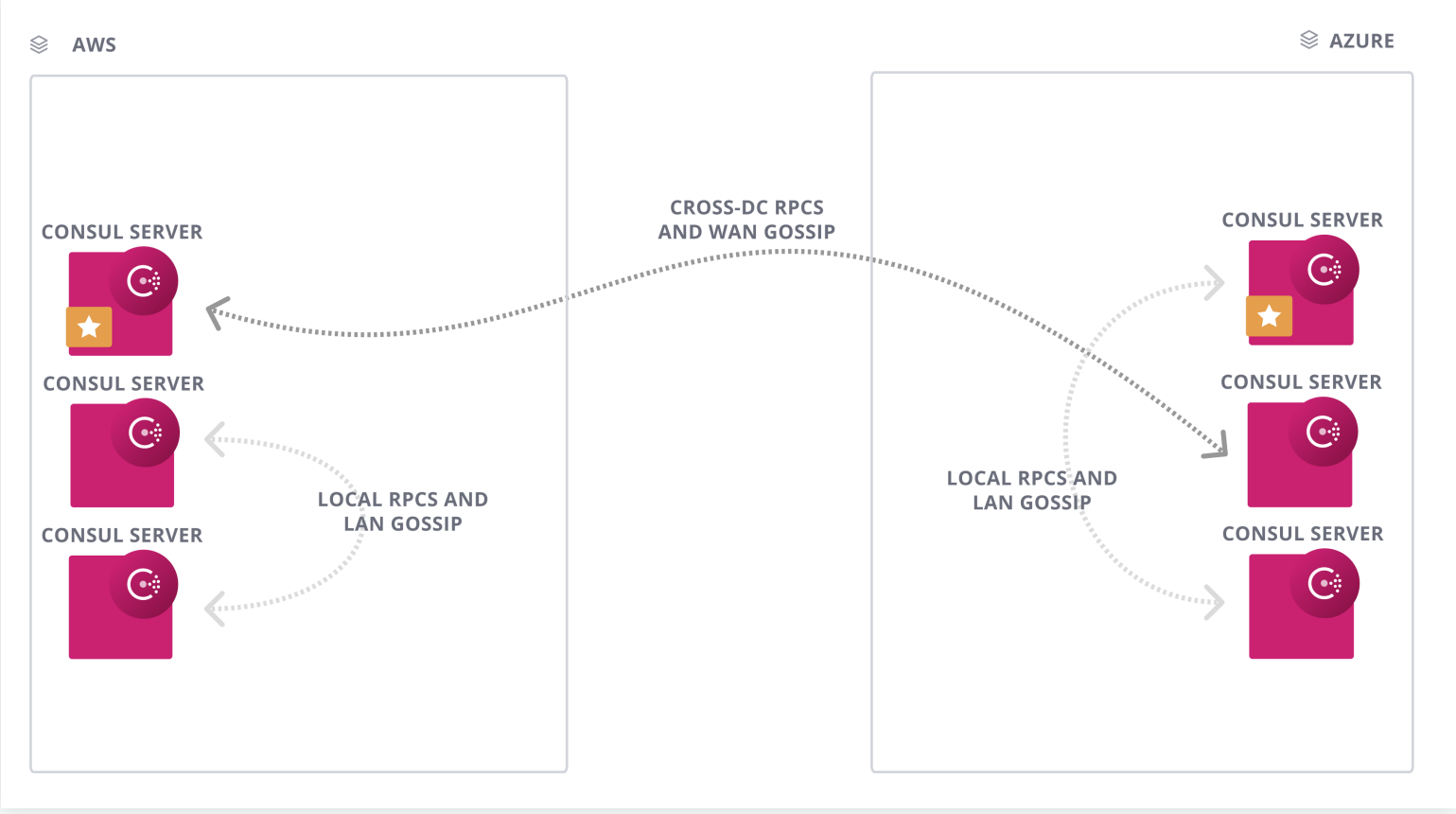
- agent - запущенный демон (может быть client или server):
- client - перенаправляет запросы на server используя RPC;
- server:
  - определяет кворум и состояние кластера;
  - отвечает на запросы RPC;
  - отвечает за обмен данными с лидером кластера;
  - отвечает за обмен данными с другими datacenter.
- datacenter - определяется как группа хостов в защищенном сетевом периметре с низкой задержкой и высокой пропускной способностью каналов;
- consensus - согласие в отношении выбранного лидера (протокол RAFT - https://raft.github.io/)
- gossip - отвечает за членство в кластере, обнаружение отказов и распространение событий. (https://www.serf.io/docs/internals/gossip.html)
  - LAN gossip
  - WAN gossip
- RPC (Remote Procedure Call) - выполняем запросы на сервере.

Архитектура кластера



multi-cluster федерация

Классическая федерация - несколько кластеров Consul из нескольких ЦОД могут обмениваться информацией.



Базовая конфигурация кластера

Конфигурационные файлы в /etc/consul.d/. Конфигурация может быть описана в HCL и/или JSON. Для проверки можно использовать `consul validate /etc/consul.d/`

```
client_addr = "0.0.0.0" # for test only
bootstrap = true
ui = true
server = true
log_file = "/var/log/consul/"
log_rotate_duration = "24h"
data_dir = "/opt/consul"
datacenter = "dc1"
retry_join = ["ht-p1-0", "ht-p1-1". "ht-p1-2" ]
```

Сетевые порты для работы: https://www.consul.io/docs/install/ports
Ну и, конечно же, всё можно сделать вручную.

Service Discovery

Service Discovery

Обнаружение сервисов - процесс поддержки актуальной информации о расположении и состоянии сервисов; с его помощью можно автоматизировать процессы:

- получения адресов экземпляров нужного сервиса;
- анонсирования новых сервисов и их эксземпляров;
- изменения настроек сервисов;
- выведение из работы сервисов и их эксземпляров, в т.ч. в случае неполадок.

Обнаружение сервисов consul реализуется следующим образом:

- сервис регистрируется в "реестре" (запросом к HTTP API или файлом конфигурации);
- при DNS запросе к агенту отдаётся список IP-адресов (A-запись) и портов (SRV-запись) тех хостов, на которых расположен сервис;
- при соответствующем HTTP-запросе к API то же, но можно получить дополнительную информацию из kv store;
- сервис можно разрегистрировать запросом к HTTP API или изменив конфигурацию consul;
- можно привязать к сервису healthcheck, который в случае failure/crirical будет убирать сервис из списка отдаваемых по DNS и HTTP API.

Запросы к DNS

Структура домена consul

```
<node>.node[.datacenter].<domain>
[tag.]<service>.service[.datacenter].<domain>
```

```
dig @ht-p1-0 -p 8600 ht-p1-0.node.consul.
dig @ht-p1-0 -p 8600 -x 192.168.69.50
dig @ht-p1-0 -p 8600 webserver.service.consul.
dig @ht-p1-0 -p 8600 webserver.service.consul. SRV
dig @ht-p1-0 -p 8600 nginx.webserver.service.consul.
```

Запросы к HTTP

Описание HTTP API: https://www.consul.io/api-docs/index

```
curl http://localhost:8500/v1/agent/members\?pretty
curl http://localhost:8500/v1/agent/checks\?pretty
curl http://localhost:8500/v1/agent/services\?pretty
curl http://localhost:8500/v1/catalog/nodes\?pretty
curl http://localhost:8500/v1/catalog/services\?pretty
curl http://localhost:8500/v1/catalog/service/consul\?pretty
```

```
curl -XGET http://localhost:8500/v1/kv/nginx/config/workers
curl -XPUT http://localhost:8500/v1/kv/nginx/config/workers -d 10
curl -XDELETE http://localhost:8500/v1/kv/nginx/config/workers
```

Запросы к HTTP

Описание HTTP API: https://www.consul.io/api-docs/index

```
payload.json
{
  "Datacenter": "dc1",
  "Node": "demo1"
}
```

```
curl -X PUT \
 --data @payload.json \
 http://localhost:8500/v1/catalog/register
curl -X PUT \
 --data @payload.json \
 http://localhost:8500/v1/catalog/deregister
curl -X PUT \
 -d '{"Datacenter": "dc1", "Node": "demo1"}' \
 http://localhost:8500/v1/catalog/deregister
```

Зарегистрируем сервис

```
{ "service": {
 "id": "webserver",
 "name": "webserver",
 "tags": [ "nginx" ],
 "port": 80,
 "meta": {
 "meta": "nginx webserver" },
 "enable_tag_override": false,
 "check": {
 "id": "webserver_up",
 "name": "nginx healthcheck",
 "http": "http://localhost/healthcheck",
 "interval": "10s",
 "timeout": "2s" }
} }
```

healthcheck

Т.к. нам в работе нужны "здоровые" сервисы, то нам надо проверять их состояние. Варианты:

- скрипты (0 - OK, 1 - Warning, 2 - Critical);
- HTTP (К примеру, GET: 2xx - OK, 429 - Warning, другие - Failure);
- TCP (соединение установлено - OK, иначе - Critical);
- alias (связываем с уже существующей проверкой).

Проверим наш сервис

```
dig @ht-p1-0 -p 8600 webserver.service.consul.
curl http://localhost:8500/v1/catalog/services\?pretty
curl http://localhost:8500/v1/catalog/service/webserver\?pretty
```

```
while true;
do curl -s $(dig +short @localhost -p 8600 webserver.service.consul. | h
sleep .5;
done
```

```
systemctl stop nginx
while true;
do dig +short @localhost -p 8600 webserver.service.consul. | wc -l;
sleep .5;
done
```

- Есть Consul UI, если его включить.

Configuration management

Управление конфигурацией

Инструменты:

- kv store
- consul-template (https://github.com/hashicorp/consul-template)

Если приложение самостоятельно умеет работать с kv хранилищем, то прекрасно (patroni, например, использует такой инструмент consul, как session). Иначе, подменяем конфигурационные файлы с помощью consul-template.

Работа с kv store

Инструменты:

```
consul kv put nginx/config/workers 10
consul kv get nginx/config/workers
consul kv put -flags=42 nginx/test/test abcd1234
consul kv put -flags=42 nginx/test/test2 abcd1234
consul kv get --detailed nginx/test/test
consul kv get -recurse
consul kv delete nginx/test/test
consul kv delete -recurse nginx/test
consul kv put nginx/config/workers 8
consul kv get nginx/config/workers
```

- Можно читать и изменять, используя Consul UI.

consul-template

Демон, который формирует файлы из подготовленных шаблонов и сведений, получаемых при опросе кластера Consul. В дополнение к этому, он может выполнять некоторые команды, когда сформирован очередной шаблон; к примеру, может перезапускать сервис.
https://github.com/hashicorp/consul-template

Для написания шаблонов используется Template Language, который включает 4 группы функций:

- API functions (обращение к Consul);
- scratchpad (промежуточное хранение данных);
- helper functions (вспомогательные функции для разбора данных, форматирования и т.п.);
- math functions.

Конфигурация consul-template

Базовые настройки, т.к. при установке бинарника вообще ничего нет.

```
consul {
address = "localhost:8500"
retry {
  enabled  = true
  attempts = 12
  backoff  = "250ms"
} }
template {
source = "/etc/nginx/nginx.ctmpl"
destination = "/etc/nginx/nginx.conf"
perms = 0600
command = "systemctl reload nginx"
}
```

consul-template на примере nginx

Сконфигурируем nginx:

```
upstream consulWebservice {
{{ range service "webserver" }}
server {{.Address}}:{{.Port}};{{end}}
}
...
listen {{ key "nginx/config/listen_port" }} default_server;
root {{ key "nginx/config/root_path" }};
...
location / {
proxy_pass http://$consulWebservice;
}
```

и попробуем:

```
consul-template -config=/etc/consul-template.d/config.hcl -once
```

Резервное копирование

Резервное копирование

На что обратить внимание:

- важна директория data-dir на server'ax;
- на client ничего интересного нет;
- бекап выполняется получанием snapshot с leader;
- snapshot не выполняется, если кластер degraded;
- snapshot не выполняется, если нет leader;
- можно снять не с leader, но данные будут устаревшими;
- при snapshot часть самых свежих данных может не попасть в снимок, т.к. они ещё не были согласованы (около 100ms), но для аварийного восстановления это, обычно, уже не имеет значения :)

Резервное копирование

```
consul snapshot save backup.snap #consistent
consul snapshot save -stale backup.snap #stale
consul snapshot inspect backup.snap
curl -XDELETE http://localhost:8500/v1/kv/nginx/config/workers
consul snapshot restore backup.snap
```

Ещё возможности

Пара слов о возможностях

ACL - для обеспечения безопасности путём установки политик по умолчанию и выдачи токенов как для конкретных операций с Consul, так и для доступа к сервисам.
https://learn.hashicorp.com/tutorials/consul/access-control-setup-production

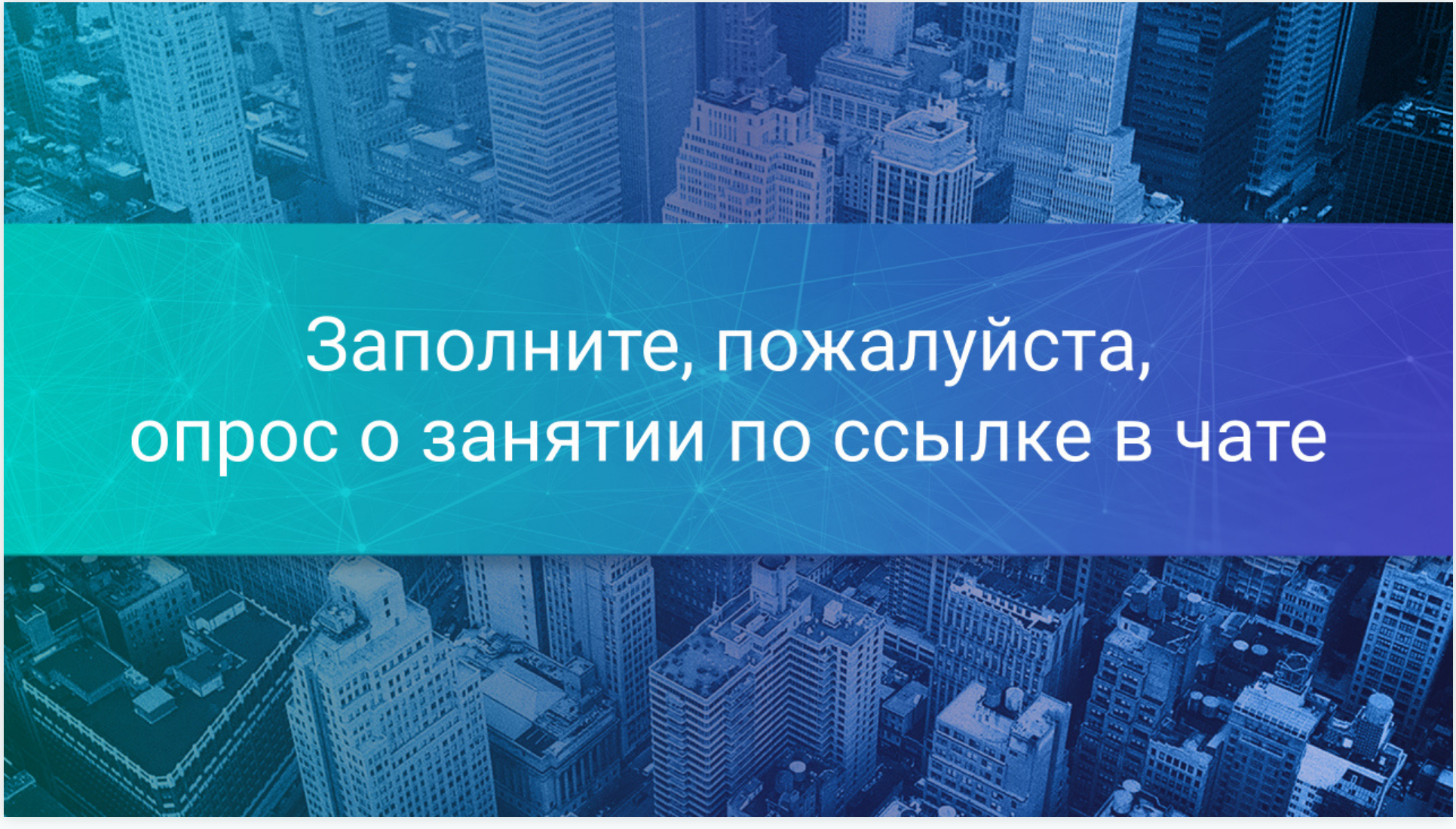
connect & intentions - защита межсервисной коммуникации и контроль над ней (кому с кем можно общаться)
https://learn.hashicorp.com/tutorials/consul/get-started-service-networking

Защита служебной информации:
https://learn.hashicorp.com/tutorials/consul/gossip-encryption-secure
https://learn.hashicorp.com/tutorials/consul/tls-encryption-secure

Рефлексия

Отметьте 3 пункта, которые вам запомнились с вебинара

Что вы будете применять в работе из сегодняшнего вебинара?


Заполните, пожалуйста,
опрос о занятии по ссылке в чате