

ФИНАЛЬНОЕ ЗАДАНИЕ

Итак, пришло время последнего рывка в рамках текущего практикума. За время прохождения программы вы узнали многое про Kubernetes — время приступить к более комплексной рабочей задаче, где всё как в реальной жизни: мало конкретики, размытое описание желаемого результата и свобода творчества.

Готовы? Приступаем!

[БЕТА](#)[МЕНЮ](#)[Часть 1](#)[Часть 2](#)[Часть 3](#)

Задание

ЗАДАНИЕ

У вас есть приложение <https://github.com/librespeed/speedtest> (кому-то оно может быть знакомо из практикума по Docker).

Ситуация типичная: вам дали репозиторий, который нужно задеплоить в кубер. Информации ноль, позвонить и узнать, как работает приложение некому (если автор не вкоммитил свой номер телефона где-то в репозитории).

Требуется итоговый набор манифестов, чтобы кто-то сделал `kubectl apply -f .` и приложение заработало.

Для удобства помещайте все объекты в разные файлы.

Требования к настройкам приложения

- Приложение должно быть запущено во frontend-режиме с хранением телеметрии, обфускацией ID и **без возможности** редактирования IP-адресов (для этого придётся немного поизучать репозиторий приложения и найти настройки — см. Dockerfile и docker/entrypoint.sh).
- Конфигурация должна производиться через переменные окружения (кроме тех случаев, когда нам явно требуется файл).
- СУБД — MySQL, пример развёртывания можно взять из [документации](#).
По желанию можно развернуть провиционер дисков (либо просто не создавайте их и данные не пишутся в файловую систему контейнера).

Требования к СУБД

- Разверните СУБД MySQL в namespace `db`.
- На всех манифестах, касающихся СУБД, должна присутствовать метка `app=db`.
- Имя манифеста с основной нагрузкой - `mysql` (тип вы без труда определите сами).
- Определите, где нужны `secrets`, а где `configmaps`, и создайте их. Присвойте имена `mysql-configmap` и `mysql-secret`.
- Создайте 2 сервиса: `clusterip` с именем `mysql` и `headless` с именем `mysql-headless`.
- Не забудьте инициализировать структуру базы из [sql-скрипта](#). Его нужно смонтировать внутри контейнера. Чтобы узнать куда, посмотрите описание на ваш образ на hub.docker.com

Требования при работе с kubernetes

- Приложение должно быть развёрнуто в namespace `final`.
- На всех манифестах, касающихся приложения, должна присутствовать метка `app.kubernetes.io/instance=librespeed`.
- Имя деплоймента `librespeed`.
- Образ (registry.rebrainme.com/workshops/middle/kubernetes-local/newplatform_autochecks/speedtest:5.3) должен скачиваться с нашего закрытого registry. Токен `pull-creds` пароль `glpat-vnpL_iv7WxdzB2QQuKEE` для регистрии `registry.rebrainme.com`. Секрет должен называться `registrysecret`.
- В качестве списка серверов необходимо использовать файл `servers.json`. Он должен быть смонтирован в ваш под в `/servers.json`. В качестве имени конфигмапы используйте `librespeed-servers`.

o

servers.json

```
[
  {
    "name": "Helsinki, Finland (1) (Hetzner)",
    "server": "//fi1.backend.librespeed.org/",
    "dlURL": "garbage.php",
    "ulURL": "empty.php",
    "pingURL": "empty.php",
    "getIpURL": "getIP.php"
  },
  {
    "name": "Helsinki, Finland (2) (Hetzner)",
    "server": "https://fi2.backend.librespeed.org:8443/",
    "dlURL": "garbage.php",
    "ulURL": "empty.php",
    "pingURL": "empty.php",
    "getIpURL": "getIP.php"
  },
  {
    "name": "Helsinki, Finland (3) (Hetzner)",
    "server": "//fi.openspeed.org",
    "dlURL": "garbage.php",
    "ulURL": "empty.php",
    "pingURL": "empty.php",
    "getIpURL": "getIP.php"
  },
]
```

```
{
  "name": "Helsinki, Finland (4) (Hetzner)",
  "server": "https://ls.fexiven.de:8443/",
  "dlURL": "garbage.php",
  "ulURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Helsinki, Finland (5) (Hetzner)",
  "server": "//fast.kabi.tk/",
  "dlURL": "garbage.php",
  "ulURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Paris, France (Hurricane Electric)",
  "server": "//librespeed.louifox.house/",
  "dlURL": "backend/garbage.php",
  "ulURL": "backend/empty.php",
  "pingURL": "backend/empty.php",
  "getIpURL": "backend/getIP.php"
},
{
  "name": "Paris, France (online.net)",
  "server": "//fr1.backend.librespeed.org/",
  "dlURL": "garbage.php",
  "ulURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Falkenstein, Germany (Hetzner)",
  "server": "//de4.backend.librespeed.org/",
  "dlURL": "garbage.php",
  "ulURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Nuremberg, Germany (1) (Hetzner)",
  "server": "//de1.backend.librespeed.org/",
  "dlURL": "garbage.php",
  "ulURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Nuremberg, Germany (2) (netcup)",
  "server": "//de2.backend.librespeed.org/",
  "dlURL": "garbage.php",
  "ulURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Nuremberg, Germany (3) (Hetzner)",
  "server": "//de3.backend.librespeed.org/",
  "dlURL": "garbage.php",
  "ulURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
}
```

```
"name": "Nuremberg, Germany (4) (Hetzner)",
"server": "//de5.backend.librespeed.org/",
"dURL": "garbage.php",
"uURL": "empty.php",
"pingURL": "empty.php",
"getIpURL": "getIP.php"
},
{
  "name": "Nuremberg, Germany (5) (Hetzner)",
  "server": "https://speedtest.ciapa.tech/",
  "dURL": "garbage.php",
  "uURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Bari, Italy (GARR)",
  "server": "//st-be-ba1.infra.garr.it",
  "dURL": "garbage.php",
  "uURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Bologna, Italy (GARR)",
  "server": "//st-be-bo1.infra.garr.it",
  "dURL": "garbage.php",
  "uURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Roma, Italy (GARR)",
  "server": "//st-be-rm2.infra.garr.it",
  "dURL": "garbage.php",
  "uURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Vilnius, Lithuania (Time4VPS)",
  "server": "//librespeed.fosssdaily.xyz/",
  "dURL": "garbage.php",
  "uURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Las Vegas, United States (BuyVM)",
  "server": "https://lv1.backend.librespeed.nixnet.services/",
  "dURL": "garbage.php",
  "uURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "New York, United States (BuyVM)",
  "server": "https://ny1.backend.librespeed.nixnet.services/",
  "dURL": "garbage.php",
  "uURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Roost, Luxembourg (BuyVM)",
```

```
"server": "https://lux1.backend.librespeed.nixnet.services/",
"dlURL": "garbage.php",
"uLURL": "empty.php",
"pingURL": "empty.php",
"getIpURL": "getIP.php"
},
{
  "name": "Borås, Sweden (Linux.Pizza)",
  "server": "//se1.speedtest.linux.pizza/",
  "dlURL": "garbage.php",
  "uLURL": "empty.php",
  "pingURL": "empty.php",
  "getIpURL": "getIP.php"
},
{
  "name": "Amsterdam, Netherlands",
  "server": "http://spd-nlsrv.hostkey.com/",
  "id": 1,
  "dlURL": "garbage",
  "uLURL": "empty",
  "pingURL": "empty",
  "getIpURL": "getIP"
},
{
  "name": "Helsinki, Finland",
  "server": "http://spd-fisrv.hostkey.com/",
  "id": 2,
  "dlURL": "garbage",
  "uLURL": "empty",
  "pingURL": "empty",
  "getIpURL": "getIP"
},
{
  "name": "NewYork, USA",
  "server": "http://spd-uswb.hostkey.com/",
  "id": 3,
  "dlURL": "garbage",
  "uLURL": "empty",
  "pingURL": "empty",
  "getIpURL": "getIP"
},
{
  "name": "Frankfurt, Germany",
  "server": "http://spd-desrv.hostkey.com/",
  "id": 4,
  "dlURL": "garbage",
  "uLURL": "empty",
  "pingURL": "empty",
  "getIpURL": "getIP"
},
{
  "name": "Reykjavik, Iceland",
  "server": "http://spd-icsrv.hostkey.com/",
  "id": 5,
  "dlURL": "garbage",
  "uLURL": "empty",
  "pingURL": "empty",
  "getIpURL": "getIP"
},
{
  "name": "Türkiy, Istanbul",
  "server": "http://spd-tr.hostkey.com/",
  "id": 6,
  "dlURL": "garbage",
  "uLURL": "empty",
```

```
"pingURL": "empty",  
"getIpURL": "getIP"  
}  
]
```

- Для хранения env-переменных используйте конфигмап **librespeed-env** , загрузите их через **envFrom** .
- Для хранения секретных данных используйте **secret** с именем **librespeed-secret** .
- Используя Ingress, опубликуйте приложение, отключите форсирования **https** . Имя для ingress-объекта — **librespeed-ingress** .
- Проверьте, что Ingress пересылает дополнительные заголовки **X-forwarded-For** и **X-real-ip** (на веб-интерфейсе приложения вы должны увидеть ваш реальный ip).
- Добавьте любые объекты в kubernetes, какие считаете нужным. Аргументируйте свой выбор.

СОЗДАТЬ ОКРУЖЕНИЕ >>>



ФИНАЛЬНОЕ ЗАДАНИЕ

Перед отправкой на проверку

Протестируйте работоспособность приложения.

Проверьте, что в СУБД создалась запись с данными вашего приложения.

Сделайте соответствующие скриншоты:

вывода команды **kubectl get pods, services, deployments, statefulsets, secrets, configmaps** (для обоих неймспейсов);
содержимого СУБД;
работающего приложения, с измеренной скоростью.

Оформите задание в репозиторий.

Отправляйте на проверку.

ФИНАЛЬНОЕ ЗАДАНИЕ

Редактор Предпросмотр



Приложите требуемые скриншоты и ссылку на репозиторий (проверьте, что он открыт).

ПРИКРЕПИТЬ МАТЕРИАЛ >>>

(не более 10 файлов; размер каждого не более 10 МБ)

ОТПРАВИТЬ >>>

ПОЙДЁМ ДАЛЬШЕ >>>