Автор cert-manager достаточно шустро выпускает новые версии и очень любит вносить изменения в процедуру установки своего продукта.

К счастью, он наконец-то сделал tutorial, в котором описал процедуру установки - начиная установкой с помощью манифестов или Helm, и заканчивая разворачиванием тестового приложения и созданием ingress. И что самое ценное - поддерживает его в актуальном состоянии.

Так что, когда будете устанавливать cert-manager на свой боевой кластер, сходите по ссылке и ознакомьтесь с актуальной информацией. А сейчас будем ставить cert-manager в учебный кластер по следующей инструкции:

Устанавливаем CRD от cert-manager, helm-репозиторий от автора cert-manager и сам chart

```
kubectl apply --validate=false -f
https://raw.githubusercontent.com/jetstack/cert-manager/release-0.12/deploy/manif
ests/00-crds.yaml

kubectl create namespace cert-manager

helm repo add jetstack https://charts.jetstack.io

helm repo update

helm install cert-manager \
 --namespace cert-manager \
 --version v0.12.0 \
 --set ingressShim.defaultIssuerName=letsencrypt \
 --set ingressShim.defaultIssuerKind=ClusterIssuer \
 jetstack/cert-manager
```

Проверяем, что все pod'ы запустились. У всех должен быть статус `Running`. Pod с webhook в версии 0.12 у меня поднимался около двух с половиной минут, потому что он хочет смонтировать секрет с сертификатом `cert-manager-webhook-tls`, а этот сертификат создается не очень быстро.

Куплено благодаря Skladchik.com

```
kubectl -n cert-manager get pod
```

Теперь можно создать тестовые ресурсы и проверить, как выписываются сертификаты. Для этого перейдем в каталог с материалами практики

```
8.cert-manager
```

```
cd ~
git clone git@gitlab.slurm.io:edu/slurm.git
cd ~/slurm/practice/8.cert-manager/
```

Проверяем работу cert-manager, выпустив самоподписанный сертификат:

```
kubectl apply -f test-resources.yaml
```

После применения манифеста должен появиться ресурс типа **certificate** с именем **selfsigned-cert**. Посмотрим его описание и манифест:

```
kubectl -n cert-manager-test describe certificate selfsigned-cert
```

```
kubectl -n cert-manager-test get certificate selfsigned-cert -o yaml
```

В `Events`сертификате должно быть написано `Certificate issued successfully`, а в манифесте, в поле:`status.message` `Certificate is up to date and has not expired`

А собственно содержимое секретного ключа и сертификата лежит в секрете **selfsigned-cert-tls**

```
kubectl -n cert-manager-test describe secret selfsigned-cert-tls
```

```
kubectl -n cert-manager-test get secret selfsigned-cert-tls -o yaml
```

И наконец, посмотрим на сущность **Issuer**, согласно которой и был выпущен сертификат

```
kubectl -n cert-manager-test get issuer -o yaml
```

Как видно из поля `spec` манифеста - данный Issuer указывает cert-manager создавать selfsigned сертификат.

Удаляем тестовый namespace, и после этого продолжим

```
kubectl delete ns cert-manager-test
```

Для выпуска сертификатов от Let's Encrypt надо создать сущность ClusterIssuer, в которой указать url ACME сервера, e-mail и секрет. В нём будет храниться информация для авторизации в letsencrypt.

Для учебных целей применяем манифест, в котором указан url от тестового stage ACME сервера

```
kubectl apply -f clusterissuer-stage.yaml
```

В файле `clusterissuer.yaml` находится Issuer, который выписывает рабочие сертификаты от lestencrypt, вы можете использовать его в своих кластерах, только не забудьте поменять e-mail в манифесте. На учебном кластере этот манифест применять не надо, потому что лимиты от letsencrypt достаточно жесткие.

Исправляем поля host в манифесте `tls-ingress.yaml` (укажите **ваш** номер студента в двух местах) и применим его:

```
kubectl apply -f tls-ingress.yaml -n default
```

Проверяем созданный сертификат и секреты

```
kubectl get certificate my-tls -o yaml
kubectl get secret my-tls -o yaml
```

Убедимся, что сертификат подписан stage CA от letsencrypt

```
curl https://my.sXXXXXX.edu.slurm.io -k -v

* Server certificate:
*       subject: CN=my.sXXXXXX.edu.slurm.io
*       start date: Jun 17 14:20:23 2019 GMT
*       expire date: Sep 15 14:20:23 2019 GMT
*       common name: my.sXXXXXX.slurm.io
*       issuer: CN=Fake LE Intermediate X1
```

В поле issuer должно быть написано **CN=Fake LE Intermediate X1**. Код ответа 503 в нашем случае - это нормально.

Если все получилось, удаляем всё, что создали в процессе практических занятий:

```
helm uninstall cert-manager -n cert-manager
```

```
kubectl delete ingress my-ingress-nginx
```

PS: Эта страница периодически обновляется, чтобы оставаться актуальной и рассказывать об установке самой свежей версии cert-manager. Но перед обновлением cert-manager в своем кластере обязательно читайте документация о том, что поменялось и каким образом провести обновление. В версии 0.11 было как минимум три изменения, ломающие обратную совместимость:

- изменилась apiVersion CRD ресурсов cert-manager;
- поменялся формат манифестов Issuer и Cluster Issuer;
- поменялось название annotation (для ingress с устаревшей аннотацией сертификаты не будут создаваться, но если сертификат уже выписан, то он продолжит обновляться).

Куплено
благодаря
Skladchik.com