在上一节我们了解了如何制作一个 Scrapy 的 Docker 镜像，本节课我们来介绍下如何将镜像部署到 Kubernetes 上。

Kubernetes

Kubernetes 是谷歌开发的，用于自动部署，扩展和管理容器化应用程序的开源系统，其稳定性高、扩展性好，功能十分强大。现在业界已经越来越多地采用 Kubernetes 来部署和管理各个项目，

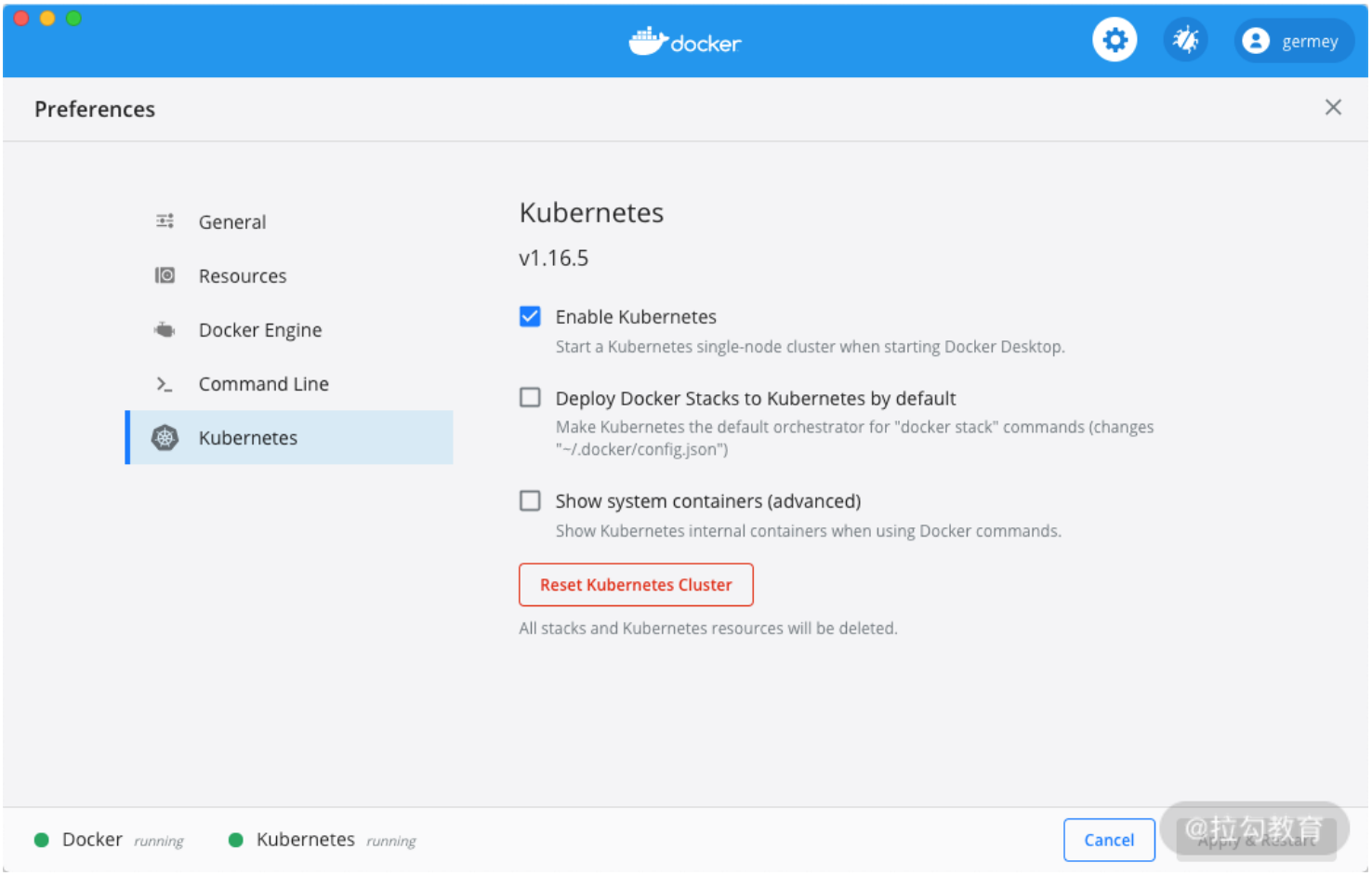
如果你还不了解 Kubernetes，可以参考其官方文档来学习一下： https://kubernetes.io/。

准备工作

如果我们需要将上一节的镜像部署到 Kubernetes 上，则首先需要我们有一个 Kubernetes 集群，同时需要能使用 kubectl 命令。

Kubernetes 集群可以自行配置，也可以使用各种云服务提供的集群，如阿里云、腾讯云、Azure 等，另外还可以使用 Minikube 等来快速搭建，当然也可以使用 Docker 本身提供的 Kubernetes 服务。

比如我这里就直接使用了 Docker Desktop 提供的 Kubernetes 服务，勾选 Enable 直接开启即可。



kubectl 是用来操作 Kubernetes 的命令行工具，可以参考 https://kubernetes.io/zh/docs/tasks/tools/install-kubectl/ 来安装。

如果以上都安装好了，可以运行下 kubectl 命令验证下能否正常获取节点信息：

```
kubectl get nodes
```

运行结果类似如下：

```
NAME             STATUS   ROLES    AGE   VERSION
docker-desktop   Ready    master   75d   v1.16.6-beta.0
```

部署

要部署的话我们需要先创建一个命名空间 Namespace，这里直接使用 kubectl 命令创建即可，Namespace 的名称这里我们取名为 crawler。

创建命令如下：

```
kubectl create namespace crawler
```

运行结果如下：

```
namespace/crawler created
```

如果出现上述结果就说明命名空间创建成功了。接下来我们就需要把 Docker 镜像部署到这个 Namespace 下面了。
Kubernetes 里面的资源是用 yaml 文件来定义的，如果要部署一次性任务或者为我们提供服务可以使用 Deployment，更多详情可以参考 Kubernetes 对于 Deployment 的说明：https://kubernetes.io/docs/concepts/workloads/controllers/deployment/。

新建 deployment.yaml 文件如下：

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: crawler-quotes
  namespace: crawler
  labels:
    app: crawler-quotes
spec:
  replicas: 1
  selector:
    matchLabels:
      app: crawler-quotes
  template:
    metadata:
      labels:
        app: crawler-quotes
    spec:
      containers:
        - name: crawler-quotes
          image: germey/quotes
          env:
            - name: MONGO_URI
              value: <mongo>
```

这里我们就可以按照 Deployment 的规范声明一个 yaml 文件了，指定 namespace 为 crawler，并指定 container 的 image 为我们已经 Push 到 Docker Hub 的镜像 germey/quotes，另外通过 env 指定了环境变量，注意这里需要将 `<mongo>` 替换成一个有效的 MongoDB 连接字符串，如一个远程 MongoDB 服务。

接下来我们只需要使用 kubectl 命令即可应用该部署：

```
kubectl apply -f deployment.yaml
```

运行完毕之后会提示类似如下结果：

```
deployment.apps/crawler-quotes created
```

这样就说明部署成功了。如果 MongoDB 服务能够正常连接的话，这个爬虫就会运行并将结果存储到 MongoDB 中。

另外我们还可以通过命令行或者 Kubernetes 的 Dashboard 查看部署任务的运行状态。

如果我们想爬虫定时运行的话，可以借助于 Kubernetes 提供的 cronjob 来将爬虫配置为定时任务，其运行模式就类似于 crontab 命令一样，详细用法可以参考：https://kubernetes.io/zh/docs/tasks/job/automated-tasks-with-cron-jobs/。

可以新建 cronjob.yaml，内容如下：

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: crawler-quotes
  namespace: crawler
spec:
  schedule: "0 */1 * * *"
  jobTemplate:
    spec:
      template:
        spec:
          restartPolicy: OnFailure
          containers:
            - name: crawler-quotes
              image: germey/quotes
              env:
                - name: MONGO_URI
                  value: <mongo>
```

注意到这里 kind 我们不再使用 Deployment，而是改成了 CronJob，代表定时任务。`spec.schedule` 里面定义了 crontab 格式的定时任务配置，这里代表每小时运行一次。其他的配置基本一致，同样注意这里需要将 `<mongo>` 替换成一个有效的 MongoDB 连接字符串，如一个远程 MongoDB 服务。
接下来我们只需要使用 kubectl 命令即可应用该部署：

```
kubectl apply -f cronjob.yaml
```

运行完毕之后会提示类似如下结果：

```
cronjob.batch/crawler-quotes created
```

出现这样的结果这就说明部署成功了，这样这个爬虫就会每小时运行一次，并将数据存储到指定的 MongoDB 数据库中。

总结

以上我们就简单介绍了下 Kubernetes 部署爬虫的基本操作，Kubernetes 非常复杂，需要学习的内容很多，我们这一节介绍的只是冰山一角，还有更多的内容等待你去探索。