让 Kubernetes 跑在 Docker 中

@张晋涛

目录

Docker 和 Kubernetes 概览

Docker

可将应用程序以 Docker 镜像的方式进行交付

```
(MoeLove) ➜  ~ docker run --rm -d redis:alpine
ffc0ba5f02daf2ed1f3bcad03f5aba0293f963c65304d4316948d9b67e969d1c
(MoeLove) ➜  ~ docker exec $(docker ps -ql) redis-cli ping
PONG
```

Docker

良好的 UX

```
(MoeLove) ➜  ~ docker --help
Management Commands:
config      Manage Docker configs
container   Manage containers
context     Manage contexts
image       Manage images
network     Manage networks
system      Manage Docker
volume      Manage volumes
year*       A docker plugin, Happy New Year! (Jintao Zhang, v1.0.0)
```

Docker

提供服务发现机制

```
(MoeLove) ➜  ~ docker network create foo
018a9af99a3aaf7eb0a3e02d17619e404f57f6b571ddf12c8bc3c5901bc47561
(MoeLove) ➜  ~ docker run --rm -d --name bar --network foo redis:alpine
d28a92dc88380aed034628e377a619721a0eb0a90da05fbfd2b27d41551ba406
(MoeLove) ➜  ~ docker run --rm --name baz --network foo redis:alpine redis-cli -h bar ping
PONG
```

Docker

- 安装： https://docs.docker.com/get-docker/
- 起步： https://docs.docker.com/get-started/overview/
- Dockerfile： https://docs.docker.com/engine/reference/builder/
- docker-compose： https://docs.docker.com/compose/reference/overview/

Docker -> Kubernetes

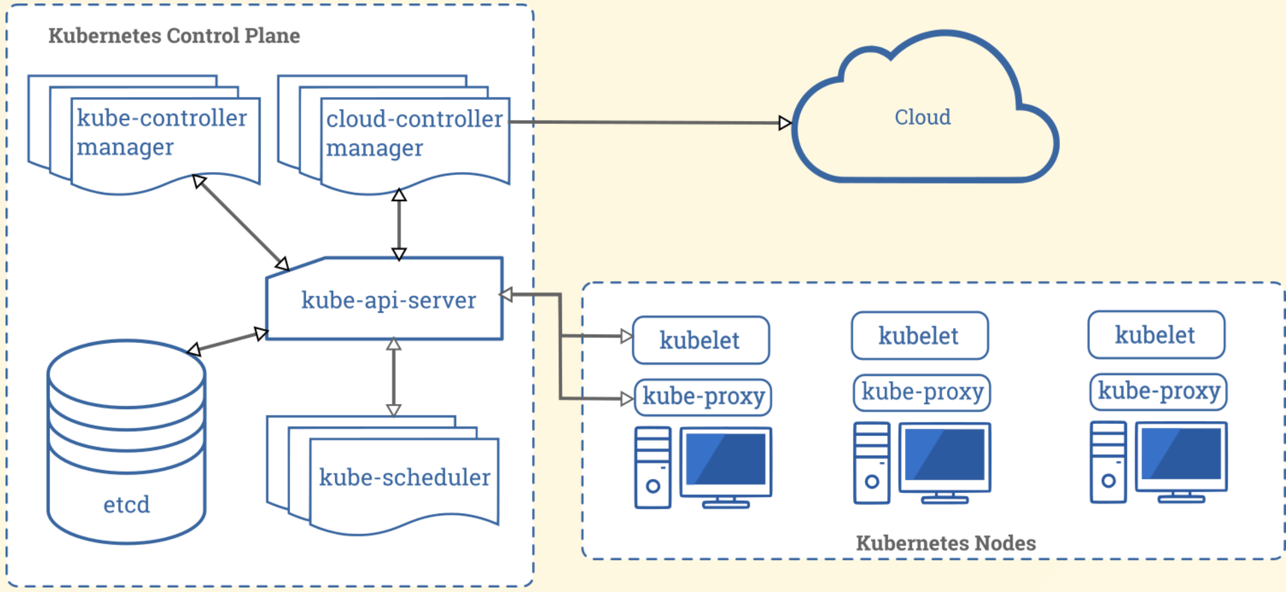
- 跨节点组网
- 灵活调度
  - 节点负载
  - 亲和性/反亲和性
  - ..
- 故障自愈
- ...

Kubernetes

- CNI（Container Network Interface）
- CSI（Container Storage Interface）
- CRI（Container Runtime Interface）
- 部署，扩缩容
- 监控
- 负载均衡
- 日志记录

Kubernetes

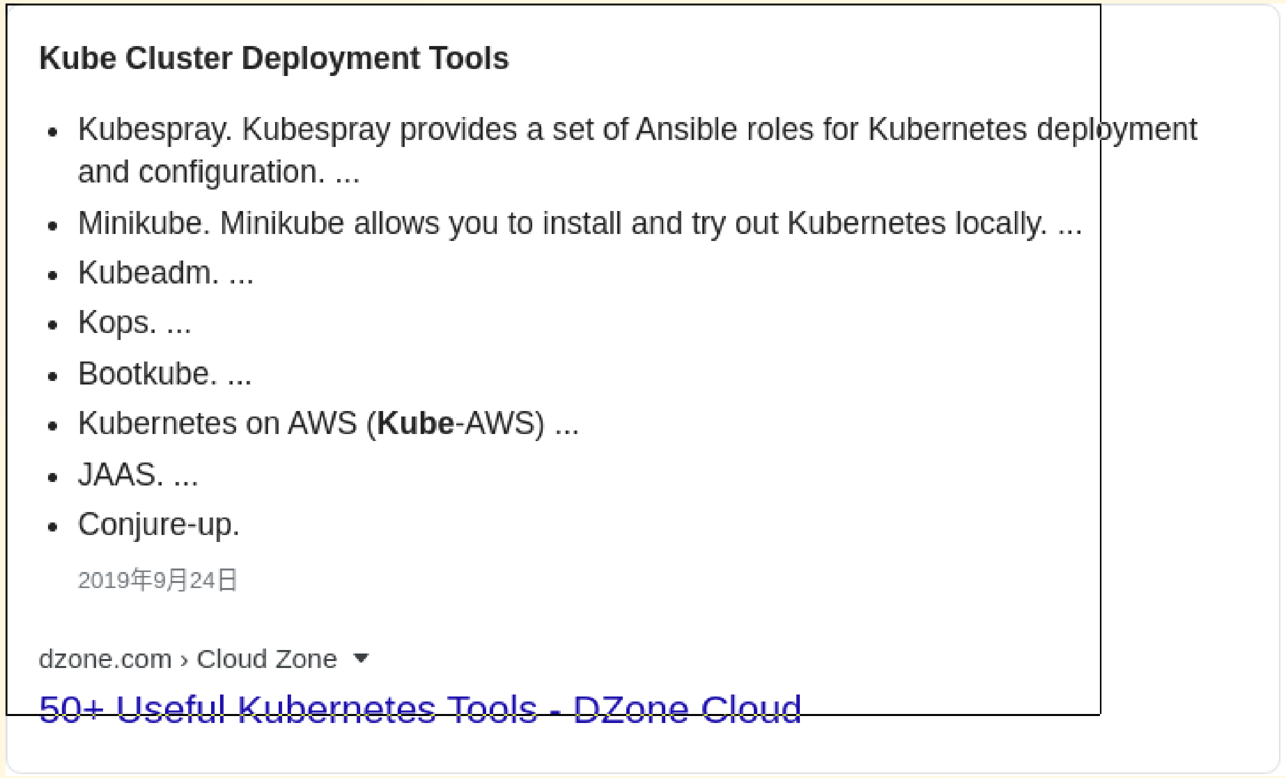
- Control Plane
  - kube-api-server
  - kube-controller-manager
  - kube-scheduler
  - etcd
- Node
  - kubelet
  - kube-proxy



Kubernetes 安装

- Kubeadm
- Kubespray
- Kops
- ...

ref: https://dzone.com/articles/50-useful-kubernetes-tools



Kube Cluster Deployment Tools

- Kubespray. Kubespray provides a set of Ansible roles for Kubernetes deployment and configuration. ...
- Minikube. Minikube allows you to install and try out Kubernetes locally. ...
- Kubeadm. ...
- Kops. ...
- Bootkube. ...
- Kubernetes on AWS (**Kube**-AWS) ...
- JAAS. ...
- Conjure-up.

2019年9月24日

dzone.com › Cloud Zone ▾

50+ Useful Kubernetes Tools - DZone Cloud

Kubernetes 安装

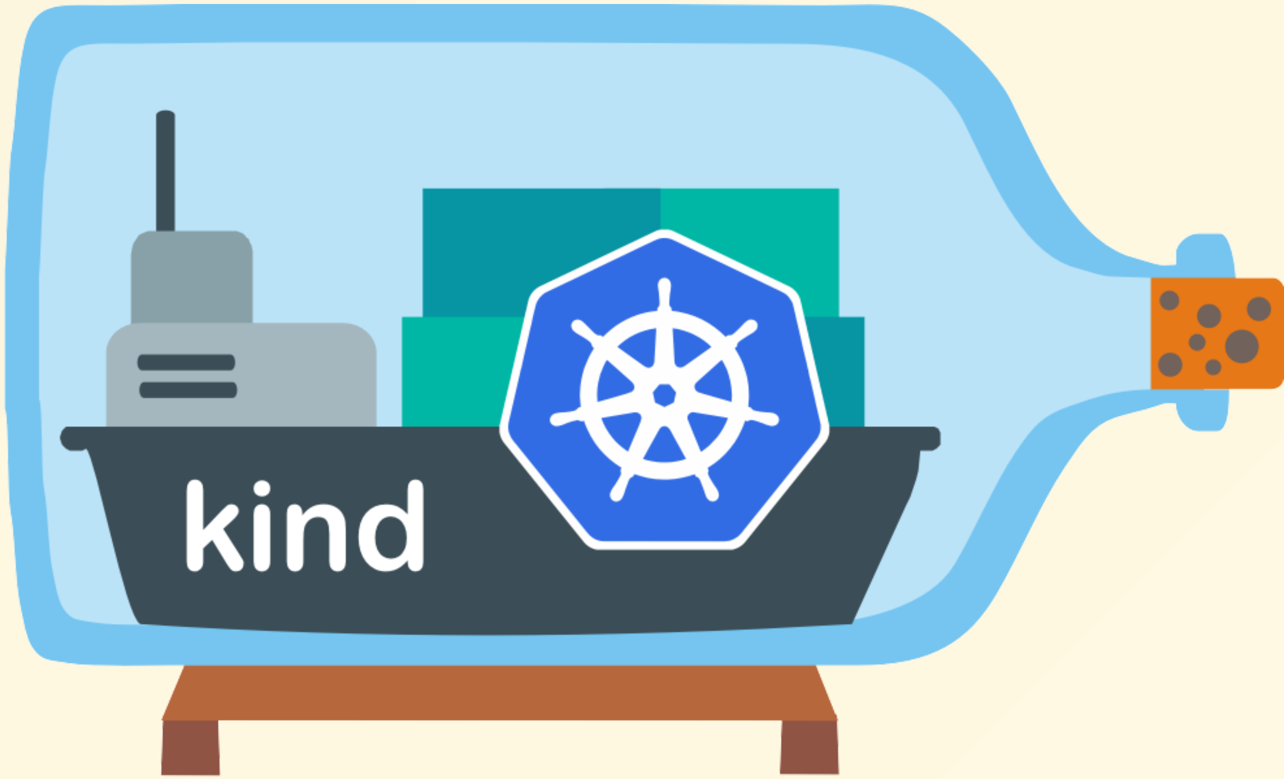
how to install Kubernetes

找到约 35,200,000 条结果 （用时 0.52 秒）

KIND 项目介绍及实践

KIND 介绍

- 使用 Docker 容器作为 Node
- 使用 kubeadm 进行集群初始化
- 支持多节点/HA 模式
- 内置 StorageClass
- 用于 K8S 自身的测试
- https://github.com/kubernetes-sigs/kind



KIND 安装

```
(MoeLove) ➜  ~ GO111MODULE="on" go get sigs.k8s.io/kind@v0.8.1
(MoeLove) ➜  ~ kind version
kind v0.8.1 go1.14.2 linux/amd64
```

或

```
curl -Lo ./kind "https://kind.sigs.k8s.io/dl/v0.8.1/kind-$(uname)-amd64"
chmod +x ./kind
mv ./kind /some-dir-in-your-PATH/kind
```

KIND 启动 K8S 集群

```
(MoeLove) ➜  ~ time kind create cluster
Creating cluster "kind" ...
✓ Ensuring node image (kindest/node:v1.18.2) 🖼
✓ Preparing nodes 📦
✓ Writing configuration 📜
✓ Starting control-plane 🕹️
✓ Installing CNI 🔌
✓ Installing StorageClass 💾
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 🙂
kind create cluster  1.18s user 0.77s system 5% cpu 33.001 total
```

KIND 实践

获取集群节点

```
(MoeLove) ➜  ~ docker exec kind-control-plane kubectl --kubeconfig=/etc/kubernetes/admin.conf get nodes
NAME                 STATUS   ROLES    AGE   VERSION
kind-control-plane   Ready    master   10m   v1.18.2
```

或（本地已安装好 `kubectl` ）

```
(MoeLove) ➜  ~ kubectl get nodes
NAME                 STATUS   ROLES    AGE   VERSION
kind-control-plane   Ready    master   10m   v1.18.2
```

KIND 实践

部署一个 Redis

```
(MoeLove) ➜  ~ docker exec kind-control-plane kubectl --kubeconfig=/etc/kubernetes/admin.conf create deploy redis --image="redis:alpine"
deployment.apps/redis created
(MoeLove) ➜  ~ docker exec kind-control-plane kubectl --kubeconfig=/etc/kubernetes/admin.conf get deploy -o wide
NAME    READY   UP-TO-DATE   AVAILABLE   AGE    CONTAINERS   IMAGES         SELECTOR
redis   1/1     1            1           4m9s   redis        redis:alpine   app=redis
(MoeLove) ➜  ~ docker exec kind-control-plane kubectl --kubeconfig=/etc/kubernetes/admin.conf get pods
NAME                     READY   STATUS    RESTARTS   AGE
redis-55f59fdd8f-qpf9f   1/1     Running   0          4m17s
```

KIND 实践

- 文档： https://kind.sigs.k8s.io/
- Slack： #kind

使用 KIND 作为本地测试环境

示例代码

```python
import tornado.ioloop
import tornado.web

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write("Hello, world")
def make_app():
    return tornado.web.Application([
        (r"/", MainHandler),
    ])
if __name__ == "__main__":
    app = make_app()
    app.listen(8888)
    tornado.ioloop.IOLoop.current().start()
```

示例代码

```
(MoeLove) ➜  kind-python tree
.
├── app.py
└── requirements.txt

0 directories, 2 files
```

容器化

构建镜像，并推送至镜像仓库

```
FROM python:3.8-alpine

WORKDIR /code
COPY requirements.txt .

RUN pip install -r requirements.txt
COPY app.py .
CMD [ "python", "./app.py" ]
EXPOSE 8888
```

部署至 K8S 集群中

```
(MoeLove) ➜  kind-python ls
app.py  deploy.yaml  Dockerfile  requirements.txt
(MoeLove) ➜  kind-python kubectl apply -f deploy.yaml
deployment.apps/kind-python created
(MoeLove) ➜  kind-python kubectl get pods -l app=kind-python
NAME                           READY   STATUS    RESTARTS   AGE
kind-python-66659564dd-zl7qd   1/1     Running   0          96s
```

让 Kubernetes 跑在 Docker 中

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: kind-python
  name: kind-python
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kind-python
  template:
    metadata:
      labels:
        app: kind-python
    spec:
      containers:
      - image: taobeier/kind-python
        name: kind-python
```

本地访问

```
(MoeLove) ➜  kind-python kubectl port-forward deploy/kind-python 8888:8888
Forwarding from 127.0.0.1:8888 -> 8888
Forwarding from [::1]:8888 -> 8888
```

```
(MoeLove) ➜  kind-python http :8888
HTTP/1.1 200 OK
Content-Length: 12
Content-Type: text/html; charset=UTF-8
Date: Sun, 19 Jul 2020 05:35:55 GMT
Etag: "e02aa1b106d5c7c6a98def2b13005d5b84fd8dc8"
Server: TornadoServer/6.0.4

Hello, world
```

扩展

- 使用 CI/CD pipeline 自动化完成部署
- 通过 KIND 配置文件直接使用 Service 地址访问

总结

- Docker 和 Kubernetes 的基础
- KIND 项目
- 容器化 Python 项目并部署至 KIND 集群中

Thanks