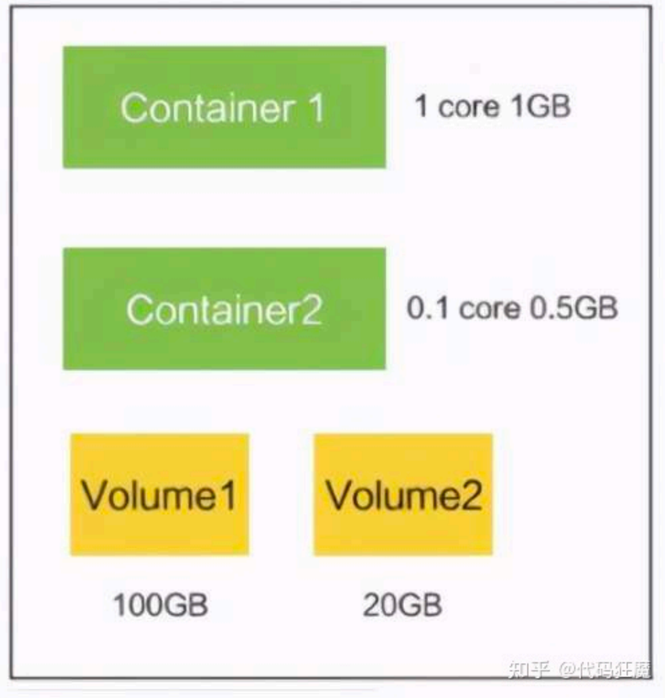
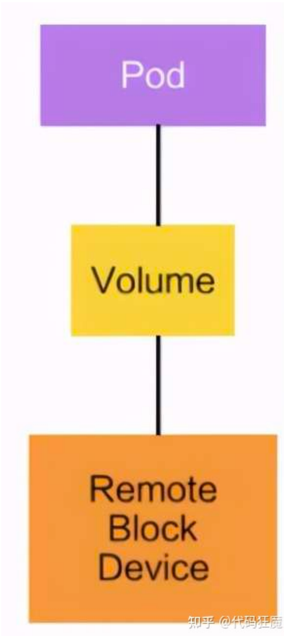
K8s

核心概念

- Pod
  - Pod是最小调度单元
  - Pod里面会包含一个或多个容器（Container）
  - Pod内的容器共享**存储及网络**，可通过localhost通信
- Volume
  - 存储卷，在Pod中可以声明卷来问**访问文件系统**
  - Volume也是一个**抽象层**，其具体的后端存储可以是本地存储、NFS网络存储、云存储（阿里云盘、AWS云盘、Google云盘等）、分布式存储（比如说像 ceph、GlusterFS ）

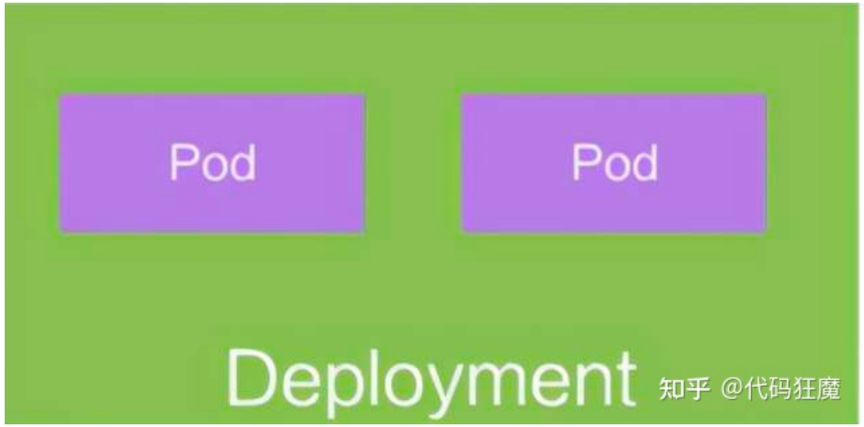




K8s

核心概念

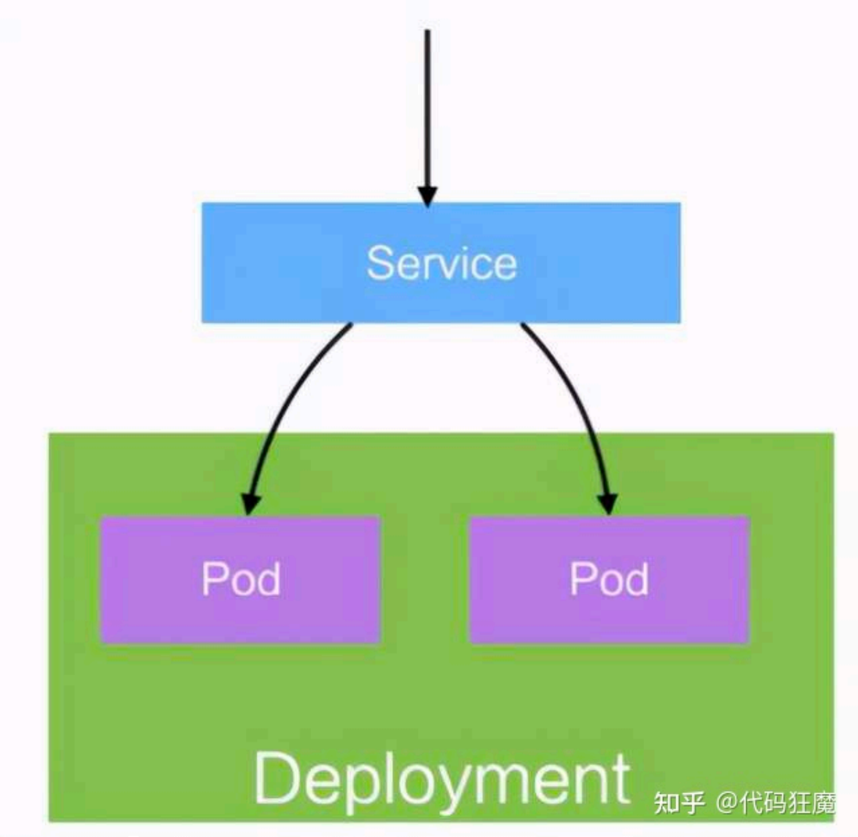
- Deployment
  - Pod 这个抽象上更为上层的一个**抽象**，它可以定义一组 Pod 的副本数目、以及这个 Pod 的版本。
  - 一般大家用 Deployment 这个抽象来做应用的真正的管理，而 Pod 是组成 Deployment 最小的单元
    - 定义一组Pod的副本数量，版本等
    - 通过控制器维护Pod的数目
    - 自动恢复失败的Pod



K8s

核心概念

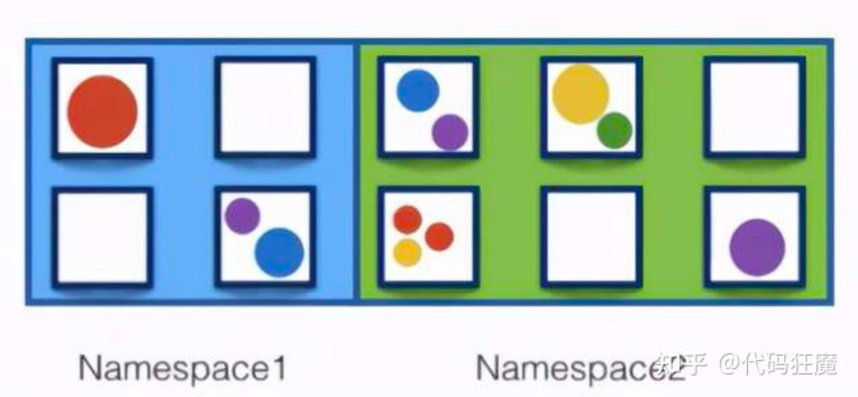
- Service
  - Pod是不稳定的，**IP是会变化的**，所以需要一层抽象来屏蔽这种变化，这层抽象叫做Service
  - 支持多种访问方式ClusterIP（对集群**内部访问**）NodePort（对集群**外部访问**）LoadBalancer（集群**外部负载均衡**）



K8s

核心概念

- Namespace
  - 用来做资源的逻辑隔离
    - 比如上面的Pod、Deployment、Service都属于资源
    - **不同**Namespace下资源**可以重名。同一**Namespace下资源名**需唯一**



K8s

核心概念

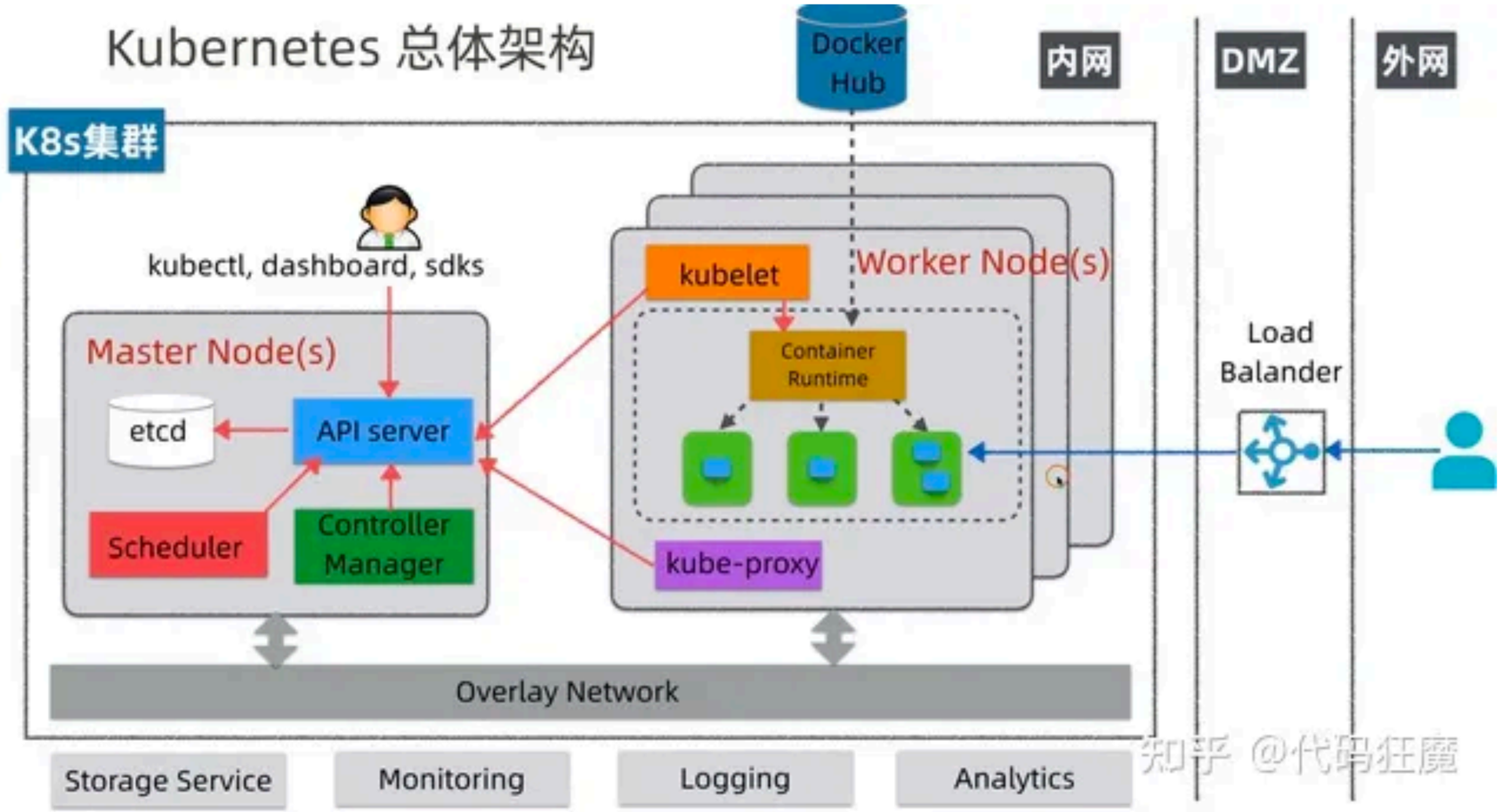
- 实例
  - kubectl apply -f https://github.com/fission/fission/releases/download/v1.17.0/fission-all-v1.17.0-minikube.yaml

```yaml
# Source: fission-all/templates/controller/svc.yaml
apiVersion: v1
kind: Service
metadata:
  name: controller
  labels:
    svc: controller
    application: fission-api
    chart: "fission-all-v1.17.0"
spec:
  type: NodePort
  ports:
  - port: 80
    targetPort: 8888
    nodePort: 31313
  selector:
    svc: controller
```

```yaml
# Source: fission-all/templates/controller/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: controller
  labels:
    chart: "fission-all-v1.17.0"
    svc: controller
    application: fission-api
spec:
  replicas: 1
  selector:
    matchLabels:
      svc: controller
      application: fission-api
  template:
    metadata:
      labels:
        svc: controller
        application: fission-api
      annotations:
        prometheus.io/scrape: "true"
        prometheus.io/path: "/metrics"
        prometheus.io/port: "8080"
    spec:
      containers:
      - name: controller
        image: "index.docker.io/fission/fission-bundle:v1.17.0"
        imagePullPolicy: IfNotPresent
        command: ["/fission-bundle"]
        args: ["--controllerPort", "8888"]
        env:
        - name: FISSION_FUNCTION_NAMESPACE
          value: "fission-function"
        - name: DEBUG_ENV
          value: "false"
        - name: PPROF_ENABLED
          value: "false"
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
```

K8s

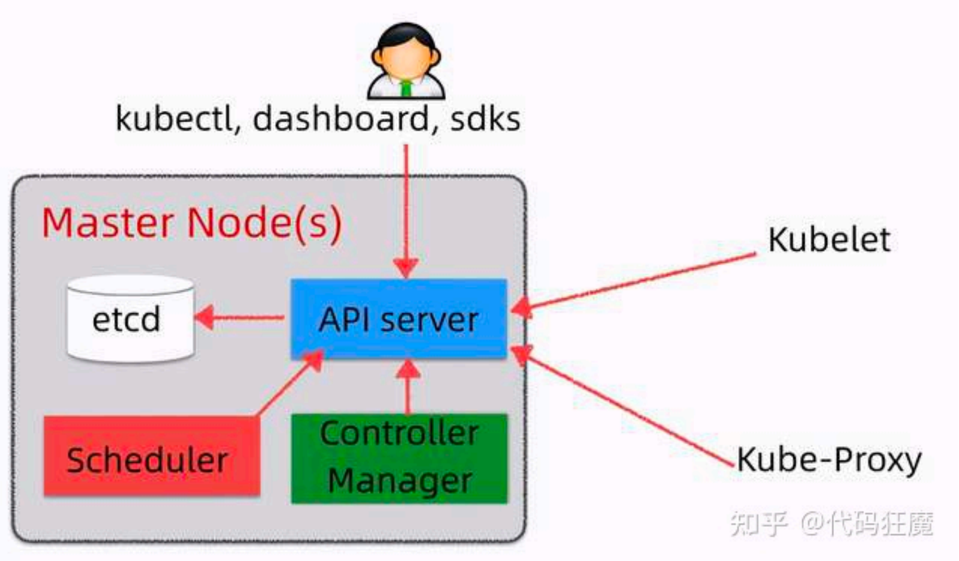
总体架构


Kubernetes 总体架构
Docker Hub
内网
DMZ
外网
K8s集群
kubectl, dashboard, sdks
Master Node(s)
etcd
API server
Scheduler
Controller Manager
kubelet
Worker Node(s)
Container Runtime
kube-proxy
Load Balander
Overlay Network
Storage Service
Monitoring
Logging
Analytics
知乎 @代码狂魔

K8s

架构

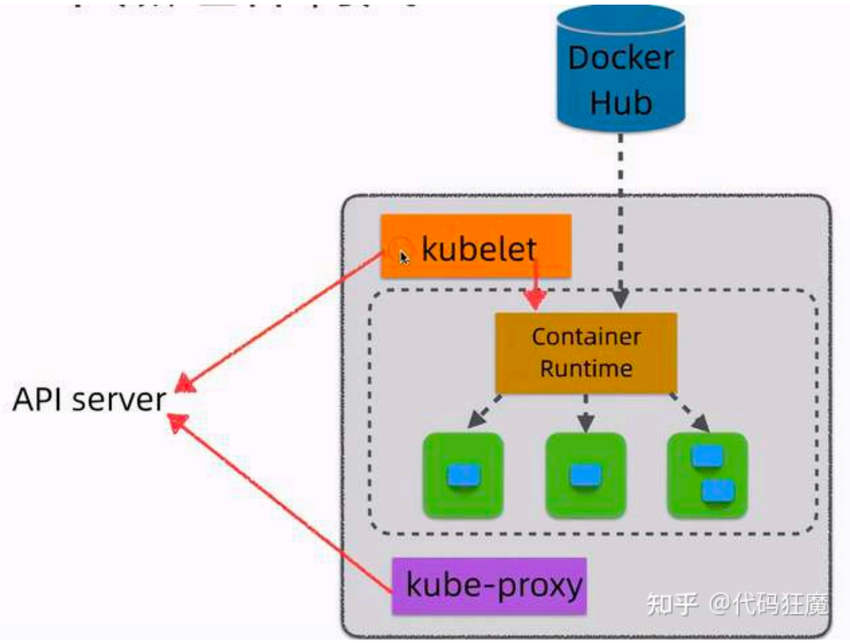
- etcd
  - 分布式KV数据库，用于**保存集群中的相关数据**；
- API Server
  - **集群统一入口**，以restful风格操作，同时交给etcd存储（是**唯一能访问etcd的组件**）；
  - 提供认证、授权、访问控制、API注册和发现等机制，可以通过**kubectl命令行工具，dashboard可视化面板，或者sdk等访问**
- Scheduler
  - 节点的调度，**选择node节点部署应用**
- Controller Manager
  - 处理集群中常规后台任务，一个资源对应一个控制器，同时监控集群的状态，**确保实际状态和最终状态一致**



K8s

架构

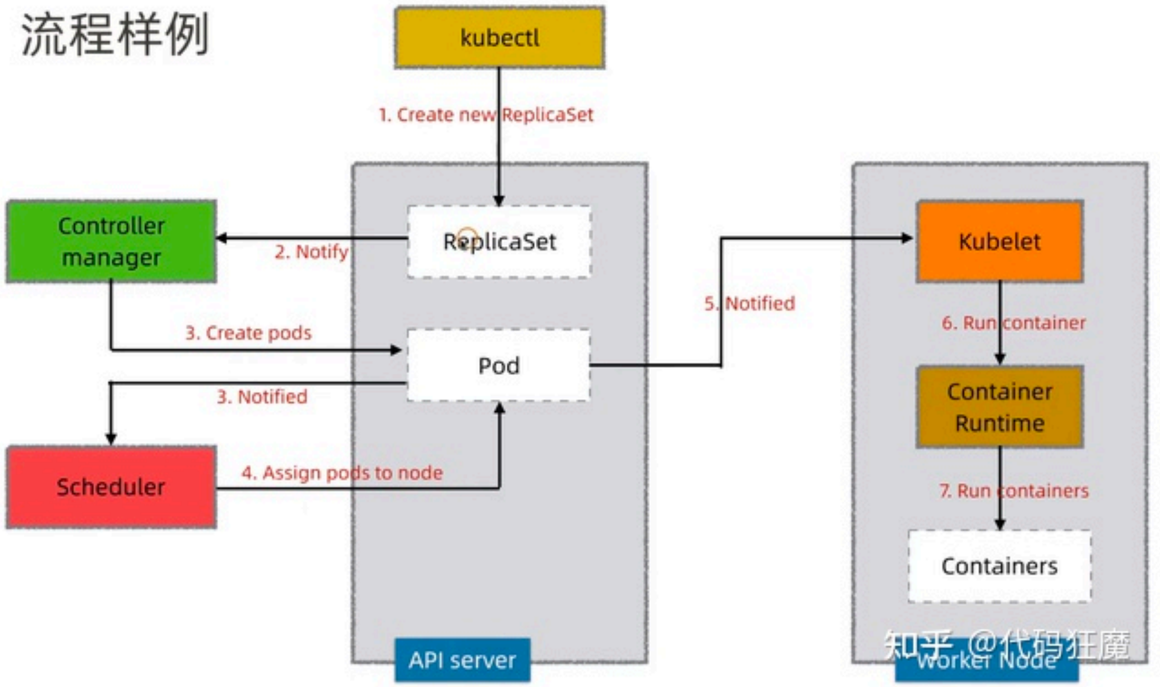
- kubelet：相当于Master派到node节点**代表，管理本机容器**，上报数据给API Server
- Container Runtime：容器运行时，K8S支持多个容器运行环境：Docker、Containerd、CRI-O以及**任何实现Kubernetes CRI (容器运行环境接口) 的软件**
- kube-proxy：实现**服务（Service）抽象组件**，屏蔽PodIP的变化和负载均衡



K8s

流程

- 通过kubectl命令行工具向API Server发送一个请求：创建ReplicaSet，API Server会将此请求存储在**etcd**中
- Controller Manager会接受到一个通知
- Controller Manager发现**现在的集群状态和预期的状态不一致**，因此需要创建Pod，此信息会通知到Scheduler
- Scheduler会**选择空闲的Worker节点**，然后通过API Server更新Pod的定义
- API Server会通知到Worker节点的上的kubelet
- kubelet指示当前节点上的Container Runtime运行对应的容器
- Container Runtime下载镜像并启动容器



K8s

安装

- kind
  - https://kind.sigs.k8s.io/docs/user/quick-start/
- minikube

k8s

实用命令

- kubectl get pod
  - 查看镜像的拉取情况
    - Running, ErrImagePull

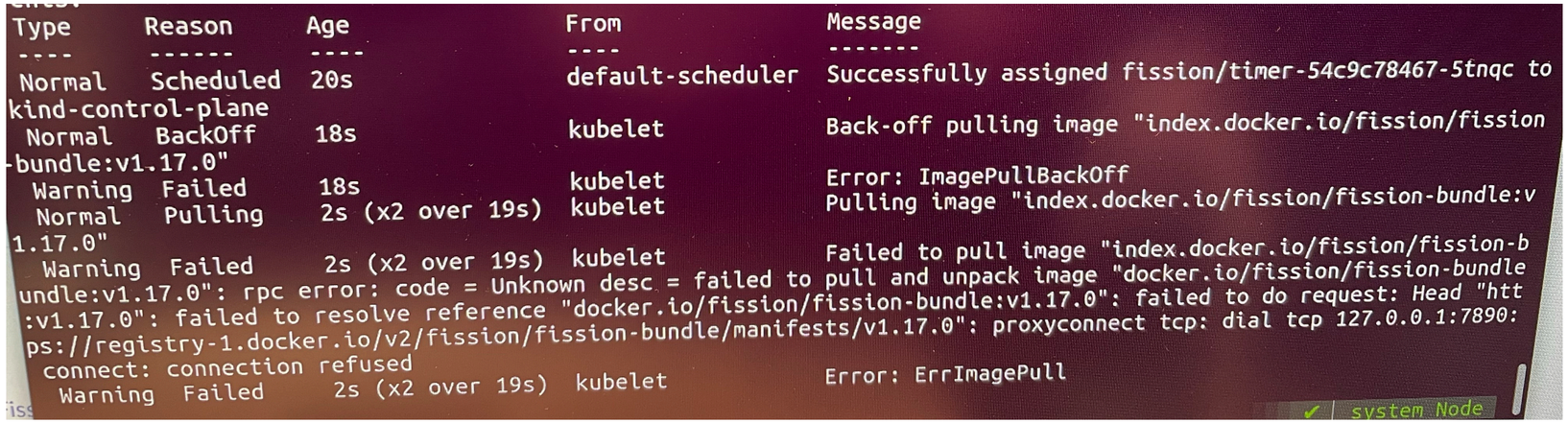
```
~ kubectl get pod                                  1 x | base Py | kind-kind/fission ○
NAME                                                READY   STATUS      RESTARTS      AGE
buildermgr-67977f7fd-9dcgc                          1/1     Running     7 (21m ago)   14d
controller-8476846849-rvrhs                         1/1     Running     7 (21m ago)   14d
executor-6d6f795b8f-qqz9g                           1/1     Running     8 (21m ago)   14d
fission-v1-17-0-fission-v1.17.0-160-hdvd9           0/1     Completed   0             14d
fission-v1-17-0-fission-v1.17.0-776-z4wjx           0/1     Completed   0             14d
kubewatcher-748d558d8d-jlw5b                        1/1     Running     7 (21m ago)   14d
mqtrigger-keda-788cc8c9cf-ckmp6                     1/1     Running     6 (21m ago)   14d
router-5695b85dbf-drdvj                             1/1     Running     7 (21m ago)   14d
storagesvc-94f978db4-rdfl5                          1/1     Running     4 (22m ago)   14d
timer-74f7f6896b-6828n                              1/1     Running     7 (21m ago)   14d
```

```
~ kubectl get pod                                  ✔ | system Node | kind-kind/fission ○
NAME                                                READY   STATUS              RESTARTS   AGE
buildermgr-c6fc9d69-bc9cx                           0/1     ErrImagePull        0          18h
controller-5d949b66-kzdd6                           0/1     ErrImagePull        0          18h
executor-5b64885578-qdjsn                           0/1     ContainerCreating   0          18h
fission-v1-17-0-fission-v1.17.0-160-k4mtc           0/1     ErrImagePull        0          18h
fission-v1-17-0-fission-v1.17.0-776-vrcc8           0/1     ErrImagePull        0          18h
kubewatcher-69f8885cd5-4f25d                        0/1     ImagePullBackOff    0          18h
mqtrigger-keda-857cf76bb6-qnzfv                     0/1     ImagePullBackOff    0          18h
router-7b4c556884-slrwj                             0/1     ImagePullBackOff    0          18h
storagesvc-66b576b85-ntgzh                          0/1     ImagePullBackOff    0          18h
timer-54c9c78467-5tnqc                              0/1     ErrImagePull        0          18h
```

k8s

实用命令

- kubectl describe pod
  - 如果pod显示镜像拉取有错误，可以通过此命令排查



参考

- https://zhuanlan.zhihu.com/p/382229383?utm_id=0

fission

安装 & 运行

- 参考
  - https://shimo.im/docs/pmkxQwl8yLUmZaAN
  - https://fission.io/docs/installation/

fission 中一些实用的命令

- 有助于排查问题
  - fission pkg info --name <pkg-name>
    - ⚠️ 输入的日志不是顺序的；查看的时候要注意
  - fission support dump
    - 会把所有的fission组件的日志打包下载到本地；
    - 调查函数部署/运行出现的问题时很有用；