实验二：Service job in cloud

Jiahua Wang

Part A：认识 Kubernetes 集群

命令运行结果

见附录图 4，图 5

对 Kubernetes 集群的认识

用途

Kubernetes 是一个用于管理容器化的工作负载和服务的自动化运维平台，它消除了容器化应用程序在部署、伸缩时涉及到的许多手动操作，实现了容器集群的自动化部署、自动扩缩容、维护等功能

架构

Kubernetes 集群主要包括控制平面组件（Control Plane Components）和多个 Node 组件，此外还可以安装插件。用户可以通过 kubectl 等接口访问集群

- 控制平面组件为集群做出全局决策，例如编排调度，检测和响应集群事件。主要包括 kube-apiserver，etcd，kube-scheduler，kube-controller-manager，cloud-controller-manager
- Node 组件维护运行的 Pod 并提供 Kubernetes 运行环境，主要包括 kubelet，kube-proxy，容器运行时（Container Runtime）。容器运行时包括多个最小的可部署的计算单元 Pod
- 主要插件包括 DNS，Web 界面，容器资源监控，集群层面日志

功能

- 服务发现：使用 DNS 名称或自己的 IP 地址来暴露容器
- 负载均衡：当进入容器的流量很大，均衡负载并分配网络流量，从而使部署稳定
- 存储编排：允许用户自动挂载选择的存储系统
- 自动部署和回滚：以受控的速率将实际状态更改为用户期望的状态
- 自动完成装箱计算：将容器按实际情况调度到节点上，以最佳方式利用资源
- 自我修复：重新启动失败的容器、替换容器、杀死不响应用户定义的运行状况检查的容器
- 密钥与配置管理：存储和管理敏感信息，在不重建容器镜像的情况下部署和更新密钥和应用程序配置

etcd 的功能

etcd 是强一致性、高可用的的分布式键值存储，用作 Kubernetes 的所有集群数据的后台数据库，内部采用 Raft 协议作为一致性算法。在 Kubernetes 集群中，etcd 主要用于共享配置和服务发现。共享配置即 etcd 通过 watch 机制实时发布配置的变化。服务发现即 etcd 保存服务的 IP 和监听端口，不同的服务通过名字找到对方并建立连接。

Part B：在 Kubernetes 上搭建博客系统

名词含义与用途

- Service：将运行在一组 Pods 上的应用程序公开为网络服务的抽象方法，它提供一致的 IP 地址和端口来访问底层 Pod。 外部用户和内部 Pod 都使用 Service 与其他 Pod 通信，这样使一组 Pod 无需跟踪另一组 Pod 的 IP 就可以使用其服务，实现了解耦。
- PersistentVolumeClaim（PVC）：表达用户对存储的请求，PVC 消耗 PV（PersistentVolume）的资源，可以请求特定的大小和访问模式，例如可以加载一个读写实例或者多个只读实例，而无须感知这些实例背后具体的存储实现。PVC 需要指定归属于某个 Namespace，在同一个 Namespace 的 Pod 才可以指定对应的 PVC。
- Deployment：定义及管理多副本应用的对象，用于部署无状态应用，解决 Pod 的生命周期管理、调度、多副本问题，为 Pod 和 ReplicaSet 提供声明式的更新能力。可以根据用户需求将 Pod 调度到目标机器上，并监控容器的运行状态，也可以快速地实现扩容。

了解集群状态，创建博客

见附录图 6，图 7，图 8，图 9

部署 Kubernetes 应用与部署传统应用的不同之处

- 部署速度：传统部署由于是在物理机上运行，速度慢；Kubernetes 部署速度更快，效率更高
- 资源利用：传统部署无法将剩余的资源分配给其他的应用，Kubernetes 部署灵活分配资源，提高资源利用率
- 资源隔离：传统部署中，如果某个程序占用大量资源，会导致其他程序性能下降；Kubernetes 部署中资源隔离性高
- 部署难度：传统部署需要手动配置环境；Kubernetes 部署可以通过命令行工具自动完成
- 跨平台一致性：传统部署对物理机环境有严格要求，Kubernetes 部署可以跨平台
- 持续集成/持续交付：Kubernetes 部署更加敏捷

附录

```
root@ecs:~# minikube version
minikube version: v1.23.1
commit: 9e2f8cb489d9b3e871ba206d40ae92c7521b7e76-dirty
```

图 1：检查 minikube 版本

```
root@ecs:~# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceeec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

图 2：运行 hello world 镜像，验证 Docker 引擎安装成功

```
root@ecs:~# kubectl version
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.1", GitCommit:"632ed300f2c34f6d6d15ca4cef3d3c7073412212"
, GitTreeState:"clean", BuildDate:"2021-08-19T15:45:37Z", GoVersion:"go1.16.7", Compiler:"gc", Platform:"linux/amd64"}
The connection to the server localhost:8080 was refused - did you specify the right host or port?
```

图 3：检查 kubectl 版本

```
root@ecs:~# minikube start --image-mirror-country='cn' --registry-mirror="https://79b750902857414ea937fcfb4ce87137.mirror.swr.
myhuaweicloud.com" --force
😄  minikube v1.23.1 on Ubuntu 18.04 (amd64)
❗  minikube skips various validations when --force is supplied; this may lead to unexpected behavior
✨  Automatically selected the docker driver. Other choices: none, ssh
🛑  The "docker" driver should not be used with root privileges.
💡  If you are running minikube within a VM, consider using --driver=none:
📘    https://minikube.sigs.k8s.io/docs/reference/drivers/none/
✅  Using image repository registry.cn-hangzhou.aliyuncs.com/google_containers
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.22.1 on Docker 20.10.8 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔎  Verifying Kubernetes components...
    ▪ Using image registry.cn-hangzhou.aliyuncs.com/google_containers/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

图 4：启动 minikube 集群

```
root@ecs:~# kubectl get pod --all-namespaces
NAMESPACE     NAME                               READY   STATUS    RESTARTS   AGE
kube-system   coredns-7d89d9b6b8-4bpnc           1/1     Running   0          67s
kube-system   etcd-minikube                      1/1     Running   0          78s
kube-system   kube-apiserver-minikube            1/1     Running   0          81s
kube-system   kube-controller-manager-minikube   1/1     Running   0          78s
kube-system   kube-proxy-blxbm                   1/1     Running   0          67s
kube-system   kube-scheduler-minikube            1/1     Running   0          78s
kube-system   storage-provisioner                1/1     Running   0          76s
```

图 5：检查组件运行情况

```
root@ecs:~/experiment2# kubectl get service --all-namespaces
NAMESPACE     NAME              TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)                  AGE
default       kubernetes        ClusterIP      10.96.0.1       <none>        443/TCP                  35m
default       wordpress         LoadBalancer   10.104.182.56   <pending>     80:30681/TCP             23m
default       wordpress-mysql   ClusterIP      None            <none>        3306/TCP                 23m
kube-system   kube-dns          ClusterIP      10.96.0.10      <none>        53/UDP,53/TCP,9153/TCP   35m
```

图 6：部署博客系统后集群中运行的所有 service 的状态

```
root@ecs:~/experiment2# kubectl get deployment --all-namespaces
NAMESPACE     NAME              READY   UP-TO-DATE   AVAILABLE   AGE
default       wordpress         1/1     1            1           24m
default       wordpress-mysql   1/1     1            1           24m
kube-system   coredns           1/1     1            1           36m
```

图 7：部署博客系统后集群中运行的所有 deployment 的状态

```
root@ecs:~/experiment2# kubectl get pod --all-namespaces
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
default       wordpress-c9cdf4bcb-5htqp          1/1     Running   0             22m
default       wordpress-mysql-575f8bcc5d-hcxp4   1/1     Running   0             22m
kube-system   coredns-7d89d9b6b8-4bpnc           1/1     Running   1 (25m ago)   34m
kube-system   etcd-minikube                      1/1     Running   0             34m
kube-system   kube-apiserver-minikube            1/1     Running   1 (23m ago)   34m
kube-system   kube-controller-manager-minikube   1/1     Running   0             34m
kube-system   kube-proxy-blxbm                   1/1     Running   0             34m
kube-system   kube-scheduler-minikube            1/1     Running   0             34m
kube-system   storage-provisioner                1/1     Running   3 (23m ago)   34m
```

图 8：部署博客系统后集群中运行的所有 pod 的状态

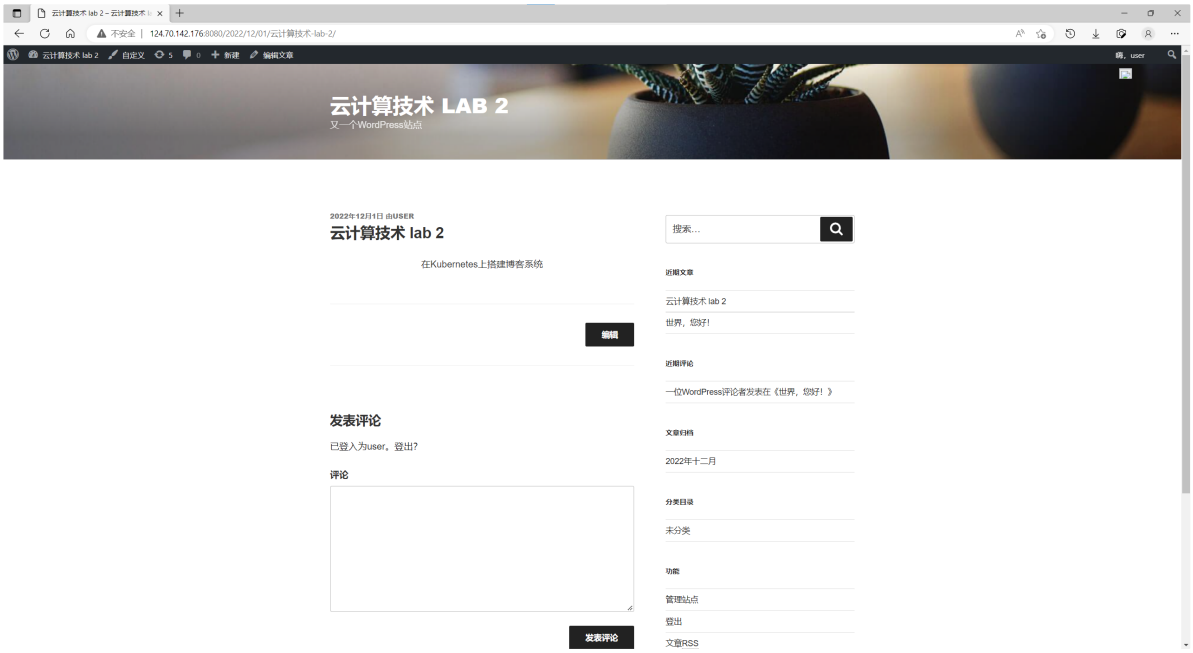
云计算技术 lab 2 – 云计算技术
不安全 | 124.70.142.176:8080/2022/12/01/云计算技术-lab-2/
云计算技术 lab 2
自定义
5
0
新建
编辑文章
嗨，user
云计算技术 LAB 2
又一个WordPress站点
2022年12月1日 由USER
云计算技术 lab 2
在Kubernetes上搭建博客系统
编辑
发表评论
已登入为user，登出?
评论
发表评论
搜索…
近期文章
云计算技术 lab 2
世界，您好!
近期评论
一位WordPress评论者发表在《世界，您好!》
文章归档
2022年十二月
分类目录
未分类
功能
管理站点
登出
文章RSS

图 9：创建的第一个博客