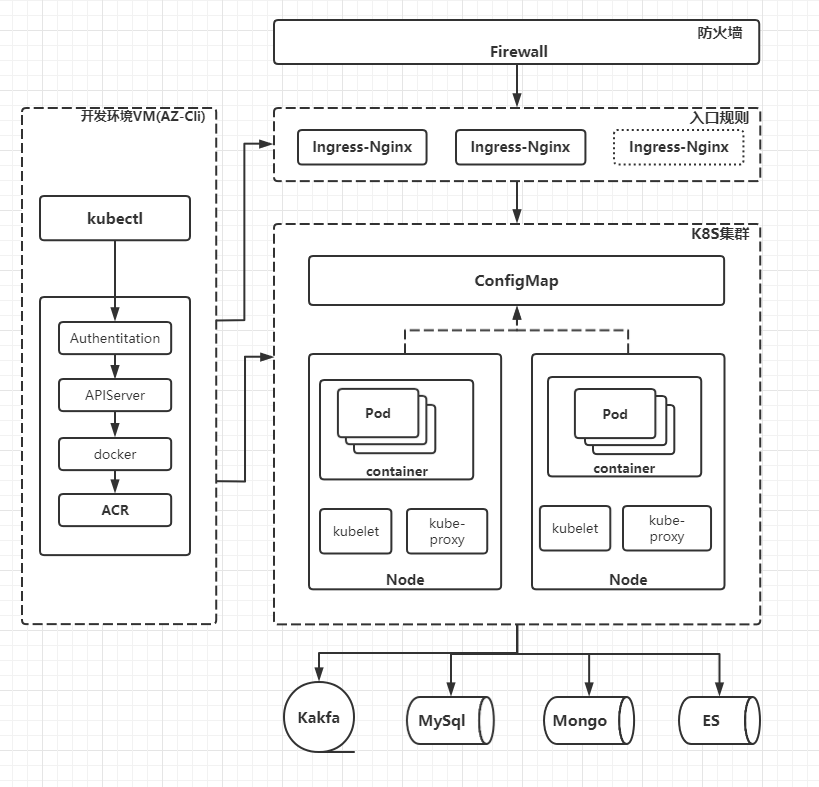
# 架构图

****

# Dashboard

**KubeSphere**

<http://120.240.0.4:30880/login>

登录账号：admin

密码： Gangtise@12345

**Kuboard**

<http://100.0.0.4:32567/dashboard>

登录token:

eyJhbGciOiJSUzI1NiIsImtpZCI6IkFPOUVTblVhRVZXamNvbGF0RWhIem94amp2MWRnRzIzcFJYOGtZWEw4cXcifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJrdWJvYXJkLXVzZXItdG9rZW4tZzJndHYiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoia3Vib2FyZC11c2VyIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiZDlhZTc4YTAtYjg1Mi00YmIyLTkzYTUtOWE3OGQ4NDdkMGI2Iiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1YmUtc3lzdGVtOmt1Ym9hcmQtdXNlciJ9.hWtfy3ftAfw0dF7qxWEzyR6U7m\_lcIzgO-oP\_x\_9FGmFscwh0bnk0zOig5aMR-PBMLYq8NGpMCRWIONosA1LazneMHo7vAWlzgmEfqXHVchLV4xSr-0rE9z\_CZDKCr\_-P2W\_Jill2LN96l8hdZTPW4Rt6TxUR6ixGLd7OFe\_h5wLPlWpE244Gtrfvc6XoPOUt2A1qjS298YYaBuxFXZsV68jFJeICpR56OYyWdr-B\_C69c6dkvPCsjgB0obir9auUUUFKca0TcrMxJGhlr5zQ4D74QA80mV2R8dy1cjD4TwUSxt85B5H\_EkcadWh9SzUGOOruZ7\_orajDTqB07TQV\_WnR-MyzUSHoQ1mLvd91Cd1zpLFRE4K74AsOFMiwXokKYM\_Wp4dQwJ1KMeOYh176rU2p5yJ5hWDsyiWP9T4KTGFebSmQ0-ZuK9bv\_upNhDbWC6L501fyVAnYqEM03ywK6C423tnAoXQnApUlzPo8UKZOYerKv3-ljDNpvJXKHm5ijd-6EcrQcHHfzzJQ4dLL08-xEsVw\_sQe\_WWQqUWzQ\_8hOQkT2Ms53rM317MvmFZxXKqteoyIkYhn5FyhBZNcPNu0MNiHTg7RMSgR8HG0qvfZuGZmK4MSUFXjQg7qZUpxZP5ANkzbyRDvhXJ1mFY8EWviB\_NJ5rkqG-fQMZA5Uk

# 环境安装

## Docker

### 安装docker

#### docker 开启2375端口，提供外部访问docker

1、编辑docker文件：/usr/lib/systemd/system/docker.service

命令：vim /usr/lib/systemd/system/docker.service

修改ExecStart行，增加内容 -H tcp://0.0.0.0:2375 ，修改后如下：

ExecStart=/usr/bin/dockerd -H tcp://0.0.0.0:2375 -H fd:// --containerd=/run/containerd/containerd.sock

2、重新加载docker配置

systemctl daemon-reload // 1，加载docker守护线程

systemctl restart docker // 2，重启docker

## yum

### yum包

wget http://vault.centos.org/7.1.1503/os/x86\_64/Packages/yum-3.4.3-125.el7.centos.noarch.rpm

wget http://vault.centos.org/7.1.1503/os/x86\_64/Packages/yum-metadata-parser-1.1.4-10.el7.x86\_64.rpm

wget http://vault.centos.org/7.1.1503/os/x86\_64/Packages/yum-plugin-aliases-1.1.31-29.el7.noarch.rpm

wget http://vault.centos.org/7.1.1503/os/x86\_64/Packages/yum-plugin-fastestmirror-1.1.31-29.el7.noarch.rpm

wget http://vault.centos.org/7.1.1503/os/x86\_64/Packages/yum-plugin-protectbase-1.1.31-29.el7.noarch.rpm

wget http://vault.centos.org/7.1.1503/os/x86\_64/Packages/yum-utils-1.1.31-29.el7.noarch.rpm

### 安装yum

依次执行以下命令

rpm -ivh yum-\*

rpm -ivh yum-metadata-parser-1.1.4-10.el7.x86\_64.rpm

rpm -ivh yum-3.4.3-161.el7.centos.noarch.rpm yum-plugin-fastestmirror-1.1.31-50.el7.noarch.rpm

## 安装python3.6

1、通过 gcc --version 查看，没安装的先安装gcc，yum -y install gcc

2、安装其它依赖包

cat /etc/redhat-release

CentOS Linux release 7.6.1810 (Core)

配epel源\_阿里

wget -O /etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo

安装python3.6.8

yum install python36 -y

安装pip3

# 搜索pip3的安装包名称

yum whatprovides pip3

# 安装pip3

yum install python36-pip -y

验证

[root@node1 ~]# python3 --version

Python 3.6.8

## 安装Azure CLI（yum）

需要依赖python3.6以上版本

<https://docs.microsoft.com/zh-cn/cli/azure/install-azure-cli-yum?view=azure-cli-latest>

历史版本下载：<https://packages.microsoft.com/yumrepos/azure-cli/>

使用2.0.78版本，太高了，SSH连接节点时无法连接。



rpm格式文件安装：

#rpm -ivh xxxx.rpm

### 先设置中国区环境

#az cloud set --name AzureChinaCloud

#az login

### 安装 kubectl

#az aks install-cli

group: gangtise-k8sGroup

name : aksCluster

### 将 kubectl 配置为连接到 Kubernetes 群集

#az aks get-credentials --resource-group gangtise-k8sGroup --name aksCluster

az aks get-credentials --resource-group gangtise-k8sGroup --name gangtiseCluster

az aks get-credentials --resource-group gangtiseProEnv --name productionEnvironment

### 验证到群集的连接

kubectl get nodes

## 创建 Azure 容器注册表

### 创建

#az acr create --resource-group gangtise-k8sGroup --name gangtiseACR --sku Basic

### 登录到容器注册表

#az acr login --name gangtiseACR

### 查询

# az acr list --resource-group gangtise-k8sGroup --query "[].{acrLoginServer:gangtiseacr.azurecr.cn}" --output table

### 为现有的 AKS 群集配置 ACR 集成

#az aks update -n aksCluster -g gangtise-k8sGroup --attach-acr gangtiseACR

### 映像从 ACR 部署到 AKS 确保你具有正确的 AKS 凭据

# az aks get-credentials -g gangtise-k8sGroup -n aksCluster

## Docker镜像

### 登录到注册表

# az acr login --name gangtiseACR

### 使用容器注册表的 acrloginServer 地址标记本地xxx 映像

# docker tag springdemo:1.0 gangtiseacr.azurecr.cn/springdemo:1.0

### 将映像推送到注册表

# docker push gangtiseacr.azurecr.cn/springdemo:1.0

### 查看注册表中的映像

# az acr repository list --name gangtiseACR --output table

### 查看特定映像的标记

# az acr repository show-tags --name gangtiseACR --repository xxx --output table

### 将映像导入 ACR

#az acr import -n <myContainerRegistry> --source dockerhub.azk8s.cn/library/nginx:latest --image nginx:v1

## 使用SSH连接到(AKS) 群集节点

### 配置

# CLUSTER\_RESOURCE\_GROUP=$(az aks show --resource-group gangtise-k8sGroup --name aksCluster --query nodeResourceGroup -o tsv)

# SCALE\_SET\_NAME=$(az vmss list --resource-group $CLUSTER\_RESOURCE\_GROUP --query [0].name -o tsv)

# ssh-keygen -m PEM -t rsa -b 2048

# az vmss extension set \

--resource-group $CLUSTER\_RESOURCE\_GROUP \

--vmss-name $SCALE\_SET\_NAME \

--name VMAccessForLinux \

--publisher Microsoft.OSTCExtensions \

--version 1.4 \

--protected-settings "{\"username\":\"azureuser\", \"ssh\_key\":\"$(cat ~/.ssh/id\_rsa.pub)\"}"

# az vmss update-instances --instance-ids '\*' \

--resource-group $CLUSTER\_RESOURCE\_GROUP \

--name $SCALE\_SET\_NAME

### 在AKS群集中运行帮助器 pod

# kubectl run --generator=run-pod/v1 -it --rm aks-ssh --image=debian

### 将终端会话连接到该容器后，使用 apt-get 安装 SSH 客户端

# apt-get update && apt-get install openssh-client -y

### 新开一个终端 辅助证书到pod中

# kubectl cp ~/.ssh/id\_rsa $(kubectl get pod -l run=aks-ssh -o jsonpath='{.items[0].metadata.name}'):/id\_rsa

### 查看 AKS 群集节点的专用 IP 地址

# kubectl get nodes -o wide

### 进入辅助容器

# kubectl exec -it aks-ssh -- /bin/bash

### 返回建立的aks-ssh

# ssh -i id\_rsa [azureuser@120.240.0.4](mailto:azureuser@120.240.0.4)

## AKS节点中安装HELM

Helm 官网 ： <https://helm.sh/docs/>

1. SSH方式进入AKS各节点
2. 安装

# sudo snap install helm --classic

1. 验证

# helm version

查找

helm search repo stable

更新图表

helm repo update

运行 Helm 图表

## AKS外部安装HELM

下载二进制文件

解压

tar -zxvf

移动到/usr/local/bin目录下

mv linux-amd64/helm /usr/local/bin/

添加执行权限

chmod +x /usr/local/bin/helm

测试使用可以使用(获取客户端的版本号)

helm version

helm repo add stable <https://kubernetes-charts.storage.googleapis.com/>

## 安装和使用 Istio

<https://github.com/istio/istio/releases/tag/1.6.3>

ISTIO\_VERSION=1.6.3

curl -sL "https://github.com/istio/istio/releases/download/$ISTIO\_VERSION/istio-$ISTIO\_VERSION-linux-amd64.tar.gz" | tar xz

cd istio-$ISTIO\_VERSION

sudo cp ./bin/istioctl /usr/local/bin/istioctl

sudo chmod +x /usr/local/bin/istioctl

在 AKS 上安装 Istio 组件

## Paping

wget <http://www.updateweb.cn/softwares/paping_1.5.5_x86-64_linux.tar.gz>

tar zxvf paping\*.tar.gz

./paping –p 30379 –c 10 100.0.0.66

## psping (Windows)

psping 100.0.0.66:30379

## nslookup

sudo apt install yum

yum -y install bind-utils

nslookup redis-0.default

# AKS操作

## 缩放节点池

### 获取 myResourceGroup 资源组中名为 myAKSCluster 的群集的节点池名称

# az aks show --resource-group gangtise-k8sGroup --name aksCluster --query agentPoolProfiles

[

{

"count": 1,

"maxPods": 110,

"name": "nodepool1",

"osDiskSizeGb": 30,

"osType": "Linux",

"storageProfile": "ManagedDisks",

"vmSize": "Standard\_DS2\_v2"

}

]

### 使用 az aks scale 命令缩放群集节点

# az aks scale --resource-group gangtise-k8sGroup --name aksCluster --node-count 3 --nodepool-name xxxxx

## 升级AKS群集

### 检查是否有可用的AKS群集升级

az aks get-upgrades --resource-group gangtise-k8sGroup --name aksCluster

### 升级

az aks upgrade --resource-group gangtise-k8sGroup --name aksCluster --kubernetes-version 1.16.9

### 若要确认升级是否成功，请使用 [az aks show](https://docs.microsoft.com/cli/azure/aks?view=azure-cli-latest#az-aks-show) 命令

az aks show --resource-group gangtise-k8sGroup --name aksCluster --output table

## 命名空间

### 创建

# kubectl create namespace spring-app

### 查看

kubectl get namespaces

### 删除

kubectl delete namespaces new-namespace

删除一个namespace会自动删除所有属于该namespace的资源。default和kube-system命名空间不可删除。

## Configmap

### 创建

# kubectl create configmap myconfig --from-file=myconfig.yaml -n test

### 查看

kubectl get configmap --all-namespaces

kubectl describe configmap special-config

### 删除

#kubectl delete configmap myconfigmap

或#kubectl delete -f myconfigmap.yml

## Volume

### 查看PersistentVolume资源详情

# kubectl get pv task-pv-volume -o wide

### 查看PersistentVolumeClaim资源状态

# kubectl get pvc task-pv-claim

### 删除

kubectl delete pvc task-pv-claim

kubectl delete pv task-pv-volume

### 进入task-pv-pod的shell中

kubectl exec -it task-pv-pod -- /bin/bash

### 查看集群中的StorageClass信息

kubectl get StorageClass --all-namespaces

## 内部负载均衡

<https://docs.azure.cn/zh-cn/aks/internal-lb>

## NGINX

<https://github.com/kubernetes/ingress-nginx/releases>

### 安装

helm install my-nginx-ingress stable/nginx-ingress \

--namespace spring-app \

--set controller.replicaCount=2

--set controller.nodeSelector."beta\.kubernetes\.io/os"=linux \

--set defaultBackend.nodeSelector."beta\.kubernetes\.io/os"=linux

--set defaultBackend.image.repository=gcr.azk8s.cn/google\_containers/defaultbackend-amd64

### 卸载

helm -n spring-app uninstall gts-nginx-ingress

### 查询

kubectl get service -l app=nginx-ingress --namespace spring-app

### 删除将流量定向到示例应用的入口路由

kubectl delete -f hello-world-ingress.yaml

重装：

helm install my-nginx-ingress stable/nginx-ingress \

--namespace gts-ingress \

--set controller.replicaCount=2

--set controller.nodeSelector."beta\.kubernetes\.io/os"=linux \

--set defaultBackend.nodeSelector."beta\.kubernetes\.io/os"=linux

--set defaultBackend.image.repository=gcr.azk8s.cn/google\_containers/defaultbackend-amd64

## REDIS

### 重启集群

1. flushdb
2. cluster reset

### 检查 redis-cli

cluster info

CLUSTER NODES

cluster slots

cluster addslots 11992 12382 14041 14386

### zuul routes url跨namespace服务调用

进入pod, 查看 /etc/resolv.conf

http://gts-userdata-service.spring-app.svc.cluster.local:12001/