**Note:**

All commands running on terminal will be bolded, and all references from Internet will be given hyperlink.

Some parameters of Commands need to be modified, whose font color will be red.

**Prerequisite:**

1. Install az cli

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli>

1. Set Azure region as China

**az cloud set --name AzureChinaCloud**

1. Log in

**az log in**

**Part 1: Create a Kubernetes Service**

**Note: For this document, the name of Kubernetes is [test], and the name of resource group is [cpqtest]**

1. 图形用户界面, 文本, 应用程序, 电子邮件

   描述已自动生成Click to create a service
2. Add a service

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. 图形用户界面, 文本, 应用程序, 电子邮件

   描述已自动生成Configuration

**Part 2: Create SSH connection for AKS**

Reference: <https://docs.microsoft.com/zh-cn/azure/aks/ssh>

1. Create an RSA key

**ssh-keygen -m PEM -t rsa -b 4096**

**文本

描述已自动生成**

****

****

1. Connect to the aks cluster

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**az account set --subscription 78796284-b042**

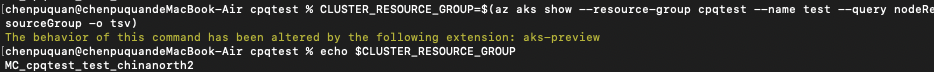
**az aks get-credentials --resource-group cpqtest --name test**

1. Obtain the resource group name of the AKS

**CLUSTER\_RESOURCE\_GROUP=$(az aks show --resource-group cpqtest --name test --query nodeResourceGroup -o tsv)**

1. Check for resource group name

**echo $CLUSTER\_RESOURCE\_GROUP**

****

1. Obtain the scale set name

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



If you get this error, try the command below:

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

1. Check for the scale set name

**echo $SCALE\_SET\_NAME**

****

1. Add SSH keys to cluster nodes

i.

**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)\"}"**

ii.

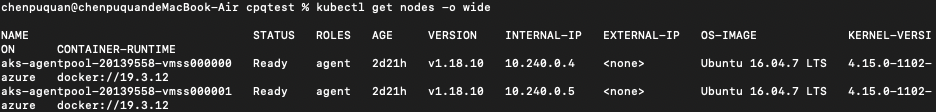
**az vmss update-instances --instance-ids '\*' \**

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

**--name $SCALE\_SET\_NAME**

1. Get the information from nodes

**kubectl get nodes -o wide**

****

1. Create SSH container and install SSH service

i.

**kubectl run -it --rm aks-ssh --image=debian**

**只能创建临时容器来建立ssh连接**

**kubectl exec -it --rm aks-ssh --image=debian 创建永久ssh容器被更新取消，已失效**

ii.

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

**文本

描述已自动生成**

1. Open a new terminal, and run command, copy your private SSH key into the helper pod. This private key is used to create the SSH into the AKS node.

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

1. Return to the terminal session to your container, update the permissions on the copied id\_rsa private SSH key so that it is user read-only

**chmod 0600 id\_rsa**

1. Connect the node with SSH

**ssh -i id\_rsa azureuser@10.240.0.4**

1. Reconnect

**kubectl attach aks-ssh -c aks-ssh -i -t // 已失效**

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

**Part 3: Deployment of NFS**

Reference:

<https://docs.microsoft.com/zh-cn/azure/aks/azure-nfs-volume>

1. Create namespace

**kubectl create namespace jhub && kubectl create namespace jhubauth && kubectl create namespace jhubauth-gpu-dev && kubectl create namespace gitlab**

1. Check for namespace

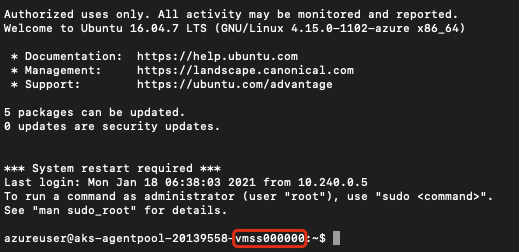
**kubectl get namespace**

**表格

描述已自动生成**

1. Connect to the Virtual Machine, set the first machine as server

**ssh -i id\_rsa azureuser@10.240.0.4**

1. Create a new folder, and install NFS on this server machine

**mkdir data**

**sudo apt-get install nfs-kernel-server -y && \**

**sudo chmod -R 777 /home/azureuser/data && \**

**mkdir -p /home/azureuser/data/nfs/mysql/data && \**

**mkdir -p /home/azureuser/data/nfs/jupyter && \**

**mkdir -p /home/azureuser/data/nfs/gitlab/postgresql && \**

**mkdir -p /home/azureuser/data/nfs/gitlab/gitlab && \**

**sudo chmod -R 777 /home/azureuser/data/nfs && \**

**sudo sed -i '$a/home/azureuser/data/nfs \*(rw,no\_root\_squash,sync,no\_subtree\_check)' /etc/exports && \**

**sudo /etc/init.d/nfs-kernel-server start && sudo /etc/init.d/nfs-kernel-server restart && \**

**showmount -e 10.240.0.4**

1. Pull image from docker (Password: newland@123)

**docker login --username=423815514@qq.com registry.cn-beijing.aliyuncs.com && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/nfs-client-provisioner:v1.0.0 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/mysql:5.7.28 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/jupyterhub\_auth:1.0.0 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/gitlab:11.5.1 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/redis:v1.0.0 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/postgresql:10 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/k8s-hub:nlai && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/cuda-cudnn-opencv-caffe-nnie-mapper:waste-v1.0.0 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/nbshare-base:0.1.6-mem && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/rabbish\_cpu:v1.0**

1. Exit, and connect to the rest machine, then set them as client

i.

**exit**

****

**ssh -i id\_rsa azureuser@10.240.0.5**

****

**sudo apt install nfs-common -y && mkdir /home/azureuser/data && sudo chmod 777 -R /home/azureuser/data && mkdir -p /home/azureuser/data/nfs && sudo mount -t nfs 10.240.0.4:/home/azureuser/data/nfs /home/azureuser/data/nfs && showmount -e 10.240.0.4**

ii. Pull image from docker

**docker login --username=423815514@qq.com registry.cn-beijing.aliyuncs.com && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/nfs-client-provisioner:v1.0.0 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/mysql:5.7.28 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/jupyterhub\_auth:1.0.0 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/gitlab:11.5.1 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/redis:v1.0.0 && \**

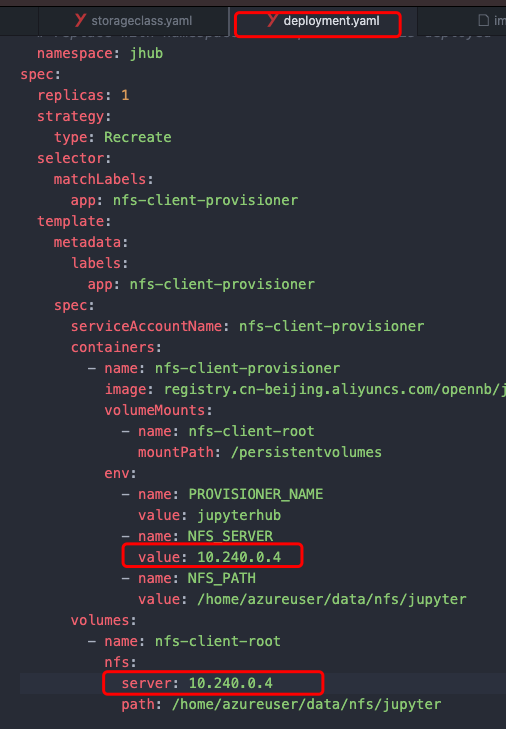
**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/postgresql:10 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/k8s-hub:nlai && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/cuda-cudnn-opencv-caffe-nnie-mapper:waste-v1.0.0 && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/nbshare-base:0.1.6-mem && \**

**docker pull registry.cn-beijing.aliyuncs.com/opennb/jupyterhub/rabbish\_cpu:v1.0**

1. Use config file for Deployment
2. **cd azure/chart/storage**
3. modify server ip address, according to the ip address of your server machine
4. **kubectl apply -f .**
5. **kubectl get pod -n jhub**

****

**Part 3: Open Ports**

**Reference:**

[**https://docs.microsoft.com/zh-cn/azure/virtual-machines/windows/nsg-quickstart-portal**](https://docs.microsoft.com/zh-cn/azure/virtual-machines/windows/nsg-quickstart-portal)

1. Search for vmss

图形用户界面

描述已自动生成

1. Choose your vmss



1. Configuration

图形用户界面

描述已自动生成

1. in

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. out

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**Part 4: Load Balance**

1. 图形用户界面, 文本, 应用程序, 电子邮件

   描述已自动生成choose load balance
2. Running condition

**图形用户界面, 文本, 应用程序

描述已自动生成**

1. Create ports

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. Authorization

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. Jupyterhub

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. Gitlab

图形用户界面, 应用程序

描述已自动生成

1. MySQL

图形用户界面, 应用程序, 电子邮件

描述已自动生成

1. Finished

图形用户界面, 应用程序, 表格

描述已自动生成

1. Config Load Balance for every port

图形用户界面, 文本, 应用程序

描述已自动生成

2. 文本

   中度可信度描述已自动生成图形用户界面, 应用程序

   描述已自动生成Finished

图形用户界面, 文本, 应用程序

描述已自动生成

**Part 4: MySQL Configuration**

1. cd azure/chart/auth/mysql
2. modify **mysql-svc.yaml** if you use different port for mysql

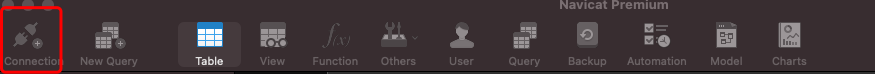
文本

描述已自动生成

1. kubectl apply -f .
2. kubectl get pod -n jhubauth



1. **电子设备的屏幕截图

   描述已自动生成**Download and use navicat

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成 **user: root**

**password: newlandedu**

1. Create a new database

电脑的截图

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

1. Restore data from nb3 database file

**截图里有图片

描述已自动生成**

**图形用户界面, 应用程序

描述已自动生成**

****

**屏幕的截图

描述已自动生成**

**Part 5: Auth Configuration**

1. cd azure/chart/auth
2. modify deploy-jhubauth.yaml

文本

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. kubectl apply -f.
2. kubectl get pod -n jhubauth



**Part 6: Jupyterhub Configuration**

1. cd azure/chart
2. 文本

   描述已自动生成modify config.yaml

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

1. **helm upgrade --install jhub ./Chart/jupyterhub --namespace jhub --version=0.8.2 --values config.yaml**

文本

描述已自动生成

1. **kubectl get pod -n jhub**



1. Use your public ip address to log in

图形用户界面, 应用程序

描述已自动生成

额外命令：

如果要删除原先jupyterhub设置再覆盖新的：

**helm del jhub -n jhub**

**helm upgrade --install jhub ./Chart/jupyterhub --namespace jhub --version=0.8.2 --values config.yaml**

覆盖新的设置前，需要删除原有的设置, ：

**kubectl delete -f.**

**kubectl apply -f.**

运行镜像的时候出现none type error, datetime error 等错误时，运行

./patch

图形用户界面

低可信度描述已自动生成

挂载磁盘，扩容缩容，自行参考别的文档。

kubectl logs -f jupyter-chenpuquan -n jhub