Documentation Links

These links are provided for your convenience. Neither the Author nor Pluralsight manage or provide this information and are not responsible for the information available on the referenced sites.

Install Azure CLI (az)

https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest

az Docs

https://docs.microsoft.com/en-us/cli/azure/reference-index?view=azure-cli-latest

AKS Docs

https://docs.microsoft.com/en-us/azure/aks/

AKS Pricing

https://azure.microsoft.com/is-is/pricing/details/kubernetes-service/

Monitoring AKS

https://docs.microsoft.com/en-us/azure/azure-monitor/insights/container-insights-overview

Azure Container Registry SKUs

https://docs.microsoft.com/en-us/azure/container-registry/container-registry-skus

Install AWS CLI (aws)

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html

aws Docs

https://docs.aws.amazon.com/cli/index.html

Install aws-iam-authenticator

https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html

Install eksctl

https://eksctl.io/introduction/installation/

EKS Docs

https://docs.aws.amazon.com/eks/?id=docs_gateway

EKS Pricing

https://aws.amazon.com/eks/pricing/

Monitoring EKS

https://docs.aws.amazon.com/eks/latest/userguide/logging-monitoring.html

Serverless Pods (Fargate)

https://docs.aws.amazon.com/eks/latest/userguide/fargate.html

Install Google Cloud SDK (gcloud)
https://cloud.google.com/sdk/docs/quickstarts

gcloud Docs
https://cloud.google.com/sdk/gcloud/reference/

GKE Docs
https://cloud.google.com/kubernetes-engine/docs/

GKE Pricing
https://cloud.google.com/kubernetes-engine/pricing

Monitoring GKE
https://cloud.google.com/monitoring/kubernetes-engine/

GCR Hostnames
https://cloud.google.com/container-registry/docs/overview#registry_name

Additional Resources
These commands are provided for convenience. Verify with docs and your values.

AKS Sample Commands
```
#configure cli
az login

#create resource group
az group create --name=kube-demo --location=westus

#register namespaces
az provider register --namespace Microsoft.Network
az provider register --namespace Microsoft.Compute
az provider register --namespace Microsoft.OperationsManagement

#create private registry in resource group
az acr create --resource-group kube-demo --location westus --name
globomanticsdemoregistry --sku Basic

#build docker image
docker build -t globomanticsdemoregistry.azurecr.io/examples/demo:1.0 .
docker tag <image-id> globomanticsdemoregistry.azurecr.io/examples/demo:1.0

#upload image to registry
az acr login --name globomanticsdemoregistry
docker push globomanticsdemoregistry.azurecr.io/examples/demo:1.0

#create cluster (3 nodes)
az aks create --resource-group=kube-demo --name=demo-cluster --node-vm-
size=Standard_D1 --generate-ssh-keys
```

```
#grant AKS pull access to ACR
#Get the id of the service principal configured for AKS
CLIENT_ID=$(az aks show -g kube-demo -n demo-cluster --query
"servicePrincipalProfile.clientId" --output tsv)
#Get the ACR registry resource id
ACR_ID=$(az acr show -g kube-demo -n globomanticsdemoregistry --query "id" --output
tsv)
#Create role assignment
az role assignment create --assignee $CLIENT_ID --role acrpull --scope $ACR_ID

#configure kubectl
az aks get-credentials --resource-group=kube-demo --name=demo-cluster

#create deployment
kubectl create deployment demo-app --
image=globomanticsdemoregistry.azurecr.io/examples/demo:1.0

#expose service
kubectl expose deployment demo-app --type=LoadBalancer --port 5000 --target-port 5000

#scale pods up (3 pods)
kubectl scale deployment demo-app --replicas=3
kubectl get pods
while true; do sleep 0.1; curl http://<external-ip>:5000/; echo -e;done

#scale to 3 nodes
az aks scale --name demo-cluster --node-count 3 --resource-group kube-demo

#update deployment
docker build -t globomanticsdemoregistry.azurecr.io/examples/demo:2.0 .
docker push globomanticsdemoregistry.azurecr.io/examples/demo:2.0
kubectl set image deployment/demo-app
demo=globomanticsdemoregistry.azurecr.io/examples/demo:2.0

#remove load balancer
kubectl delete service demo-app

#delete cluster
az aks delete --name demo-cluster --resource-group kube-demo

#delete images
az acr repository delete --name globomanticsdemoregistry --image examples/demo:2.0

#remove resource group - will get rid of container registry and container images
az group delete --name kube-demo
```

EKS Sample Commands

```
#configure awscli
aws configure

#create cluster (only supported in certain regions)
eksctl create cluster --name demo-cluster --region us-east-1 --zones=us-east-1a,us-
east-1b,us-east-1d

#create repository
aws ecr create-repository --repository-name demo

#build / tag image
docker build -t <aws_account_id>.dkr.ecr.us-east-1.amazonaws.com/<image>:<tag> .
docker build -t 481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:1:0 .

#upload image to registry
aws ecr get-login --region us-east-1 --no-include-email
docker push 481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:1.0

#create deployment
kubectl create deployment demo-app --image=481978552537.dkr.ecr.us-east-
1.amazonaws.com/demo:1.0

#expose service
kubectl expose deployment demo-app --type=LoadBalancer --port 5000 --target-port 5000

#scale pods
kubectl scale deployment demo-app --replicas=3

#hit pods
kubectl get service
while true; do sleep 0.1; curl http://aec84399f1b5f11ea833d0a247407c60-1248700768.us-
east-1.elb.amazonaws.com:5000/; echo -e;done

#get nodegroup info
eksctl get nodegroup --cluster=demo-cluster

#scale nodes
eksctl scale nodegroup --cluster=demo-cluster --nodes=3 --name=ng-e56250ca

#update deployment
docker build -t 481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:2.0 .
docker push 481978552537.dkr.ecr.us-east-1.amazonaws.com/demo:2.0
kubectl set image deployment/demo-app-ec2 demo=481978552537.dkr.ecr.us-east-
1.amazonaws.com/demo:2.0

#create fargate cluster
eksctl create cluster --name demo-cluster-fargate --region us-west-2 --zones=us-east-
1a,us-east-1b,us-east-1d --fargate
```

```
#remove load balancer
kubectl delete service demo-app

#delete cluster
eksctl delete cluster --name demo-cluster

#delete images
aws ecr list-images --repository-name demo

#delete 2.0 image
aws ecr batch-delete-image --repository-name demo --image-ids
imageDigest=sha256:f00e429b3dd2a6f5b158ca035e8d16afc779b1db0b813a2c9b9f47060d32b1a3

#delete entire repository --force is to delete images in repo if present
aws ecr delete-repository --repository-name demo --force
```

GKE Sample Commands

```
#get project id
glcoud projects list

#create an environment variable with the projectID
export PROJECT_ID=<your-project-id>

#build and tag image
docker build -t gcr.io/${PROJECT_ID}/demo:1.0 .

#test image
docker run -p 5000:5000 <image-id>

#push to google cloud container registry
gcloud auth configure-docker
docker push gcr.io/${PROJECT_ID}/demo:1.0

#create cluster with 3 nodes in node pool
gcloud container clusters create demo-cluster --num-nodes=3

#create a deployment with one pod
kubectl create deployment demo-app --image=gcr.io/${PROJECT_ID}/demo:1.0

#create LoadBalancer to access app
kubectl expose deployment demo-app --type=LoadBalancer --port 5000 --target-port 5000

#scale to 3 pods
kubectl scale deployment demo-app --replicas=3

#get external ip
kubectl get services
```

```
#hit service with curl
while true; do sleep 0.1; curl http://<external-ip>:5000/; echo -e;done

#change app
docker build -t gcr.io/${PROJECT_ID}/demo:2.0 .
docker push gcr.io/${PROJECT_ID}/demo:2.0
kubectl set image deployment/demo-app demo=gcr.io/${PROJECT_ID}/demo:2.0

#scale node pool up/down
gcloud container clusters resize demo-cluster --num-nodes 1
gcloud container clusters resize demo-cluster --num-nodes 3

#cleanup
kubectl delete service demo-app
gcloud container clusters delete demo-cluster
gcloud container images delete gcr.io/${PROJECT_ID}/demo:1.0
gcloud container images delete gcr.io/${PROJECT_ID}/demo:2.0
```