Getting Started with EKS Exercise Files

Code

These folders contain a snapshot of the code used in the Getting Started with EKS course by Craig Golightly on Pluralsight.

`managedkube` - this folder has the staging examples for the Terraform infrastructure code. You can get updated information about the Terraform modules and examples by visiting the ManagedKube GitHub repository.

`sample-app` - this folder has the sample applicaiton that gets deployed into the EKS cluster.

Resources

GitHub - https://github.com/

AWS - https://console.aws.amazon.com/

AWS Service Quotas - https://console.aws.amazon.com/servicequotas/home

Terraform - https://www.terraform.io/

Terraform Cloud - https://www.terraform.io/cloud

Grafana Labs (Prometheus, Grafana, Loki) - https://grafana.com/docs/

Locust (load test) - https://locust.io/

Helm - https://helm.sh/

ManagedKube GitHub repo - https://github.com/ManagedKube/kubernetes-ops

GitHub actions - https://github.com/features/actions

kubectl - https://kubernetes.io/docs/reference/kubectl/overview/

Working with Git Branches course - https://www.pluralsight.com/courses/git-branches-working

Designing for Complexity on AWS course (AWS Organizations and AWS SSO) - https://www.pluralsight.com/courses/designing-complexity-aws

Designing for Advanced Security within AWS course - https://www.pluralsight.com/courses/designing-advanced-security-aws

Commands

Generate a config to connect to your cluster with `kubectl`

```
aws eks --region us-east-1 update-kubeconfig --name staging
```

`kubectl` commands

```
# cluster information
kubectl cluster-info

# list nodes
kubectl get nodes

# list pods for all namespaces
kubectl get pods --all-namespaces

# check ingress
kubectl get ingress --all-namespaces

# check certificates
kubectl get Issuers,ClusterIssuers,Certificates,CertificateRequests,
Orders,Challenges --all-namespaces

# install metrics server
kubectl apply -f https://github.com/kubernetes-sigs/metrics-server
/releases/latest/download/components.yaml

# get metrics for nodes
kubectl top nodes

# get metrics for pods
kubectl top pods --all-namespaces

# get deployments in sample-app namespace
kubectl get deployments -n sample-app

# get information about hpa in sample-app namespace
kubectl describe hpa -n sample-app
```

Terraform commands

```
# log into your Terraform Cloud account
terraform login

# initialize a Terraform directory
terraform init

# run a plan
terraform plan

# run an apply
terraform apply

# In order to delete the infrastructure queue Terraform Destroy
# plans from Terraform Cloud. Destroy workspaces in this order:
kubernetes-ops-staging-sample-app
kubernetes-ops-staging-helm-grafana-loki-stack
kubernetes-ops-staging-helm-kube-prometheus-stack
kubernetes-ops-staging-helm-external-dns
kubernetes-ops-staging-helm-ingress-nginx
kubernetes-ops-staging-helm-cert-manager
kubernetes-ops-staging-25-eks-cluster-autoscaler
*kubernetes-ops-staging-5-route53-hostedzone
kubernetes-ops-staging-20-eks
kubernetes-ops-staging-10-vpc
```

*Note that DNS records for `api.k8s` and `grafana.k8s` subdomains are NOT automatically deleted from the hosted zone when the Terraform destroy plans are run on `kubernetes-ops-staging-sample-app` and `kubernetes-ops-staging-helm-grafana-loki-stack` respectively, so you need to delete those 4 records before running destroy on `kubernetes-ops-staging-5-route53-hostedzone`

Git commands

```
# check what is new and what is staged for commit
git status

# create a branch called sample-app
git checkout -b sample-app

# show what has changed
git diff

# add a file called main.tf
git add main.tf

# add all changes
git add *

# commit what you have added with a comment
git commit -m "adding new files"

# add and commit all changes
git commit -am "committing everything"

# push the sample-app branch to remote
git push origin sample-app

# switch back to main branch
git switch main

# pull changes merged on remote main to local main (run on main)
git pull

# list branches
git branch

# delete the local branch named "sample-app"
git branch -d sample-app
```

Loki query to get logs from the `kube-system` namespace

```
{namespace="kube-system"}
```

Locust commands

```
# install requirements (run in /sample-app folder)
pip install -r requirements.txt

# run locust test against endpoint
locust --host https://api.k8s.staging.globomantics.net -f sample_load.py
```

Secret values to define in TF Cloud and GitHub

```
# for TF Cloud and GitHub to access your AWS account
AWS_ACCESS_KEY_ID
AWS_SECRET_ACCESS_KEY

# for GitHub action to access TF Cloud
TF_API_TOKEN_STAGING
```