

Esse projeto tem como proposta construir utilizando a ferramenta Terraform e recursos do Google Cloud Platform (GCP), um Cluster Kubernetes Privado (GKE), uma Virtual Private Cloud (VPC) e inserir uma aplicação simples, Hello Monks, inserida no repositório DockerHub, neste Cluster e expor essa aplicação de forma segura utilizando o NGINX ingress controller.

3.6 Google Cloud Platform (GCP)

1. Inicialmente é necessário criar um projeto com o nome escolhido e coletar o seu "Id". Doc
2. Para executar os próximos passos é necessário ativar as seguintes APIs: Kubernetes Engine(GKE), Compute Enginee (CE) e Identity-Aware Proxy (IAP). Doc
3. Agora para acesso do Terraform ao projeto é necessário criar uma Service Account para o Terraform e aplica a role de Project Edit. Doc
4. Gere uma *Key.json* para a Service Account criada no item anterior e faça o donwload da mesma para a pasta Terraform Doc
5. Aplique a role de **iap.tunnel.user** no seu usuário Doc.

3.7 Terraform

O Git instalado é um pré-requisito para o Terraform possa acessar e fazer downloads dos módulos necessários. No diretório "Terraform":

1. Altere o id do projeto coletado anteriormente no arquivo de variáveis Terraform.tfvars.
2. No diretório Terraform execute os seguintes comandos:
\$ terraform init
\$ terraform plan
\$ terraform aplay
obs.: Espere o processo terminar, isso pode demorar um pouco

3.8 Google Cloud Platform (GCP)

Para executar os próximos passos certifique-se que o processo anterior terminou sem erros.

1. Acesse o Host criado no recurso Compute Enginee, como nomeado no arquivo host.tf
2. Utilize o Cloud Shell da VM Host
3. Baixe o kubectl na Instância VM
\$ sudo apt-get install kubectl
4. É necessário fazer a autentificação para isso utilize o comando
\$ gcloud auth login

5. Caso seja necessário instale o SDK de autenticação do GKE
\$ sudo apt-get install google-cloud-sdk-gke-gcloud-auth-plugin
6. O tunelamento/jump para o Cluster GKE pode ser feito
"ex: gcloud container clusters get-credentials monks-cluster --zone us-central1-c --project monksproject"

Ainda no Cloud Shell da instância VM e conectado ao cluster, execute os itens a seguir para a implementação dos objetos do NGINX ingress Controller:

1. Instale o HELM
\$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
\$ chmod 700 get_helm.sh
\$./get_helm.sh
2. Faça o download do repositório e aplique o Chart do Ingress Nginx Controller
\$ helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
\$ helm repo update
\$ helm install monks ingress-nginx/ingress-nginx
3. Certifique-se de que o Nginx ingress Controller foi instalado e anote o EXTERNAL-IP do service do ingress-nginx-controller
\$ kubectl get pod
\$ kubectl get svc monks-ingress-nginx-controller

Agora para aplicar os objetos Kubernetes da aplicação é necessário fazer o downloads dos arquivos monks.yaml e ingress_monks.yaml na VM Host e seguir os próximos passos:

1. Aplique os objetos Kubernetes do arquivo monks.yaml
\$ kubectl apply -f monks.yaml
2. Verifique o Status dos Objetos
\$ kubectl get all -n monks-ns
3. Para aplicar o Ingress é necessário mudar o Host no arquivo ingress_monks.yaml para o endereço EXTERNAL-IP copiado no item anterior
obs.: Utilize o nip.io - ex: "34.122.88.204.nip.io"
\$ kubectl apply -f ingress_monks.yaml
4. Verifique os status dos objetos
\$ kubectl get all -n monks-ns

Em seu Browser digite o endereço utilizado no Host - ex: "34.122.88.204.nip.io" e verifique o acesso.

3.9 Deletar o projeto do cluster

Caso queira excluir o projeto siga os próximos passos:

1. Na pasta do Terraform execute o seguinte comando
\$ terraform destroy
obs. Espere o processo terminar, isso pode demorar um pouco
2. No console do GCP entre em Faturamento e desative o projeto criado no campo "ações"
3. No console do GCP entre em IAM e Administração, Configurações e encerre o projeto.