

Kubernetes

Criando Imagens personalizadas com o Docker

. Criando uma imagem personalizada

→ No PowerShell com Docker instalado:

- cd app-2-kubernetes (pasta com arquivos index.html e dockerfile)
- docker build . -t moiseslcar/app-html:1.0
- docker images
- docker login

Username: moiseslcar
Password: Mo692764@
Login Succeeded

- docker push moiseslcar/app-html:1.0

* Checar no hub.docker.com.

. Criando um deployment de um aplicativo

- .
- .
- .

. Criando um Load Balancer por YAML

* Para o ‘apiVersion’ no arquivo yaml:

Pod	v1
Deployment	apps/v1
Service	v1

Exemplo: *app-html-lb.yaml*

```
apiVersion: v1
kind: Service
metadata:
  name: app-html-lb
spec:
  selector:
    app: app-html
  ports:
    - port: 80
      targetPort: 80
  type: LoadBalancer
```

Serviços de Acesso para Kubernetes Pods

Serviços de acesso

. Criando um NodePort

→ Baixar repositório exemplo do Denilson:

- git clone <https://github.com/denilsonbonatti/k8s-pod-exemplo1-dio.git>

→ Entrar na pasta criada e checar com o VScode os arquivos:

‘dockerfile, index.php, pod.yml e nodeport.php’

→ Gerar a imagem: (* Verificar se docker está ativo.. “- docker -v”)

- docker build . -t moiseslcar/myapp-php:1.0

→ Subir para o hub do Docker:

- docker push moiseslcar/myapp-php:1.0

→ Verificar imagem em hub.docker.com (*Imagem pronta p/ ser utilizada)

→ Conectar no cluster da GCP:

- gcloud container clusters get-credentials meu-cluster --zone us-central1-c --project boreal-depth-369023 (já conectado.. Ver nodes.. -kubectl get nodes)

→ Criar o pod:

- kubectl apply -f .\pod.yml

- kubectl get pods

→ Ver detalhes (ip's) de cada node do cluster GCP:

- gcloud compute instances list ou

- kubectl get nodes -o wide

→ Gerar a porta de acesso com o nodeport.yml:

- kubectl apply -f .\nodeport.yml

* Para funcionar, liberar a porta no firewall.. comando específico:

- gcloud compute firewall-rules create my-app-php -allow tcp:30734

* “30734” - porta gerada aleatoriamente pelo nodeport.yml

* Checar no navegador com qualquer ip dos pods:porta.. (Exemplo:
34.121.145.40:30734)

. Executando aplicações no Pod

→ Acessando e executando arquivo da aplicação em execução:

- kubectl get pods
 - kubectl exec --stdin --tty myapp-php -- /bin/bash
- ```
root@myapp-php:/var/www/html# apt install nano
nano index.php
```
- Alterar e checar no navegador, em qualquer pod.
- ```
# exit (Sair do container)
```

. Deployment e Service em um único arquivo YAML

...

. Encaminhamento de porta

* Exemplo com imagem do mysql, liberando a porta 3306 e acessando o banco c/ um cliente MySql (SQLPro) - criando tabela, inserindo um registro e dando um select.

. Criando conexão com um banco de dados

- git clone <https://github.com/denilsonbonatti/k8s-database-exemplo.git>
- > Entrar na pasta 'database' e gerar a imagem (mudar para o hub do docker moiseslcar):
 - docker build . -t moiseslcar/meubanco:1.0
- Subir para o hub do Docker:
 - docker push moiseslcar/meubanco:1.0
- > Ver em hub.docker.com/repositories
- * No arquivo db-deployment.yml, além do deployment, sobe tbm um serviço de conexão ao banco.
 - > Subir o deployment do bd:
 - kubectl apply -f ./db-deployment.yml
 - > Entrar na pasta 'backend' e gerar a imagem (sempre mudar p/ gerar a imagem no dockerhub moiseslcar):
 - docker buil . -t moiseslcar/php:1.0
 - Subir para o hub do Docker:

```
- ducker push moiseslcar/php:1.0
--> Subir o deployment do backend, onde sobe tbm um seviço NodePort:
- kubectl apply -f .\php-deployment.yml
* Verificar as regras de firewall:
- cloud compute firewall-rules list
--> Liberar a porta(do serviço NodePort) no firewall:
- cloud compute firewall-rules create backend --allow tcp:30005
** Database e Backend ok **

--> Executar o frontend isoladamente na própria máquina, executando o index.html da pasta frontend.

* Pegar o ip de qualquer pod e substituir no link da URL no arquivo js.js:
- kubectl get nodes -o wide
--> Executar o index.html (Inserir uns dados de exemplo no formulário..)
--> Checar se dados constam mesmo no banco - conectar no pod mysql:
- kubectl get pods
- kubectl exec --tty --stdin mysql-56b44f87f5-8pqhz -- /bin/bash
bash-4.2# mysql -u root -h 127.0.0.1 -p
Password: Senha123
mysql> use meubanco;
mysql> select * FROM mensagens;
+-----+-----+-----+
| id | nome | mensagem |
+-----+-----+-----+
| 27 | Mosley Tribes | Testando inserção no banco.. |
| 431 | Justin Verbal | K8s bombandoo.. |
+-----+-----+-----+
2 rows in set (0.00 sec)

** Se matar o pod, se matar o deployment, tudo se perde. **

mysql> exit
bash-4.2# exit
```

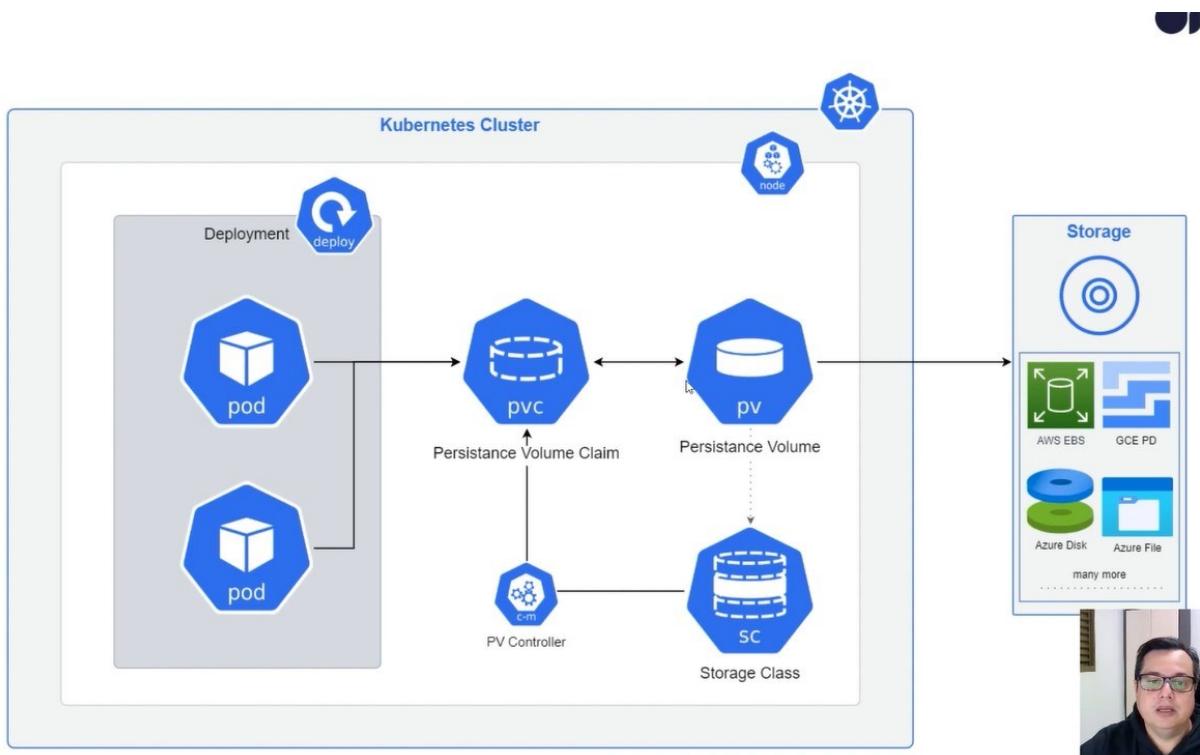
Persistência de Dados em Clusters Kubernetes

Persistência de dados

. Introdução ao PV e PVC

--> PersistentVolume (PV)

--> PersistentVolumeClaim (PVC)



. Criando os arquivos de configuração de PV e PVC

--> Testes localmente com o minikube, ainda sem o arquivo PV:

- . Deploy de uma imagem do MySql:5.6, criando tabela, inserindo registro, matando o deploy e o pod – Subindo novamente o deploy do mySql, conectando ao banco e conferindo que não perde os dados.

- . Criando os arquivos de configuração de PV e PVC parte 2

```
--> pv.yml, pvc.yml  
--> Meios de acesso: ReadWriteOnce, ReadOnlyMany e ReadWriteMany  
* Exemplos com o minikube.
```

- . Atribuindo a um deploy um PVC

```
..  
--> Testes com o minikube local.
```

- . Armazenamento persistente em nuvem

Provisionar PersistentVolumes dinamicamente

<https://cloud.google.com/kubernetes-engine/docs/concepts/persistent-volumes>

```
# pvc-pod-demo.yaml  
apiVersion: v1  
kind: PersistentVolumeClaim  
metadata:  
  name: pvc-demo  
spec:  
  accessModes:  
    - ReadWriteOnce  
  resources:  
    requests:  
      storage: 30Gi  
  storageClassName: standard-rwo
```

--> Criar arquivo *mysql-deploy.yml* com o serviço do PVC acima e o Deployment (do arquivo Armazenamento de dados local.txt do material) e fazer as alterações:

```
* storage: 10Gi  
* local --> pvc-gcp  
* replicas: 6  
* volumes:  
  - name: pvc-gcp  
    PersistentVolumeClaim:  
      ClaimName: pvc-gcp2
```

--> Testes com GCP

** Este ‘`accessModes:- ReadWriteOnce`’, não funciona para este tipo de disco na nuvem, pois de acordo com a documentação, é acessível apenas pelo mesmo nó.

- . Utilizando um servidor NFS Cloud Filestore

** Fazer o exemplo com Filestore da GCP para ver o arquivo persistido, replicado em todos os pods e em todos os nós do cluster.

5800/6111

Criando o Deploy de uma Aplicação

Projeto

- . Apresentação ppt com detalhes para desenvolver o projeto.. no material.

--> git clone <https://github.com/denilsonbonatti/k8s-projeto1-app-base.git>

** Persistir no banco.

Trabalhando com Branches no GitHub

Conceitos iniciais sobre reverter commits

Principais conceitos sobre Branches

- . Introdução ao curso
- . O que são branches
- . Entendendo na prática sobre branches
- . Guiando-se nas branches com a tag HEAD
- . Comando checkout e merge

Visualização de histórico

- . Comando stash e seus subcomandos
- . Comando git log

- . Subcomandos específicos com `git log`

Entenda como reverter commits

- . Conceitos iniciais sobre reverter commits

--> `git revert x git reset`

- `git reset (-- soft`
- `-- mixed`
- `-- hard)`

- . Comandos para reverter commits - Parte 1

- . Comandos para reverter commits - Parte 2

- `git reset --hard HEAD~1` (Perda de código**)

- `git status`

- `git log -oneline` (para ver linha a linha)

- `git revert` (sem as flags, pode ser orientado à HEAD ou à um commit específico)

Estruturando commits

- . Conceitos iniciais sobre estruturação de commits

Porque me importar ?

Melhor legibilidade do histórico

Amigável para novos desenvolvedores

Amigável ao versionamento semântico

--> Commits Atômicos (Organizados)

Estrutura

Assunto

Corpo

Rodapé

- Estrutura básica de um commit

- . Praticando estruturação de commits - Parte 1
 - > Criar repositório com read-me e fazer o clone no gitbash
 - > Issues no repositório, no github (criando uma nova issue)
 - git config --global --list (Lista as configurações do git global)
-
- . Conceitos avançados sobre commits
 - . Usando vim (editor de código)
 - . Verificando detalhes no github sobre commits (descrição, corpo e footer)
 - . Commits Semânticos
 - . Semantic Versioning (3 . 2 . 7 - MAJOR . MINOR . PATCH) - <https://semver.org/>
 - . Conventional Commits - <https://www.conventionalcommits.org/>

Git e GitHub focado em PullRequest

Forks

- . Apresentação do curso

- Forks
- Permissões
- Templates de issues e Pull Request
- Meu primeiro Pull Request
- Aliases

- git remote -v (mostra pra onde o repositório tá apontado)

- . Forks

Permissões

- . Permissões

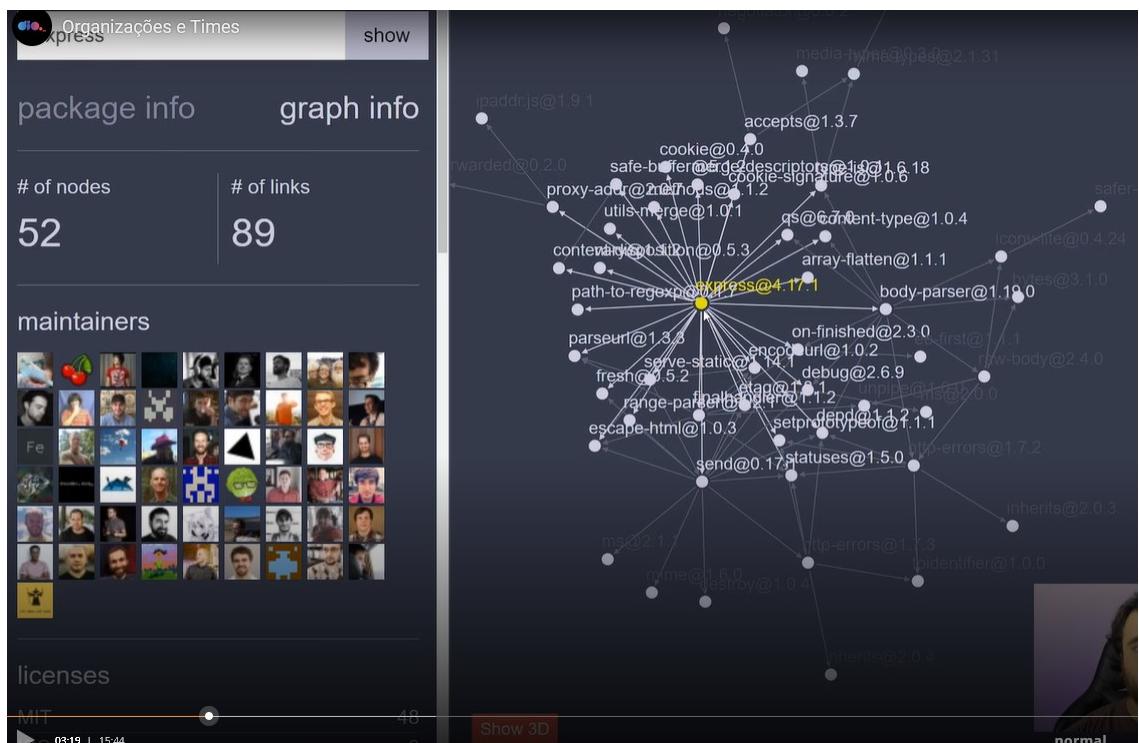
--> Adicionando colaboradores ao projeto (repositório) no github

- . Exemplo: Clonando projeto, alterando arquivos e fazendo push na master.

Colaboradores x Pull Request

Organizações e Times

- Organizações e Times
 - Ferramenta para visualizar os pacotes do npm - <https://npm.anvaka.com>



* Quantas pessoas ligadas diretamente à um projeto.

- . Sempre analisar bem o readme, por exemplo, antes de fazer um pull request em um projeto ou se já tem uma issue criada semelhante.

--> Criando nova Organização, criando time, adicionando colaboradores

Pull Request

. Fazendo o primeiro Pull Request

--> Projeto “aprenda-git” no repositório do instrutor, criado para reunir conhecimento e para a colaboração**

** Tarefa: Submeter um pull request no projeto

- Criar um arquivo md com sugestões, fazer o commit (pull request e conferir).

Issues

- . Utilizando Issues

Issues => Discussões

--> Aba Issues em ‘Settings’ - Criar / Adicionar templates (Bug report, Feature request..)

** Mais uma ferramenta de organização que o github propõe para ser adotada no dia a dia.

Alias

- . Configurando Aliases

- git config --global alias.s status
- git config --global --list (aparece “alias.s=status”)
- git s (para testar)

** Pode ser usado para qualquer comando, para facilitar o dia a dia.

- git config --global --unset alias.b (deletar alias)
- git config --global --list (para conferir)

6200 / 7750

Trabalhando com Git e Gitlab na prática

Conceitos básicos sobre Git

- . Introdução e objetivos do curso

--> trabalhando com Versionamento e o *IntelliJ* também

* Com Osnir Cunha

- <https://linkedin.com/in/osnircunha>
- <https://github.com/osnircunha>
- <https://gitlab.com/osnircunha>

- . Conceitos sobre Git e Fluxo de trabalho

Versionamento

- VCS - Version Control System
 - São sistemas gerenciadores de versões
 - Mantém o registro de modificações do código
- São importantes para:
 - Manter controle das mudanças:
 - Quem? O que? Quando? Por que?
 - Disponibilizar código mais recente

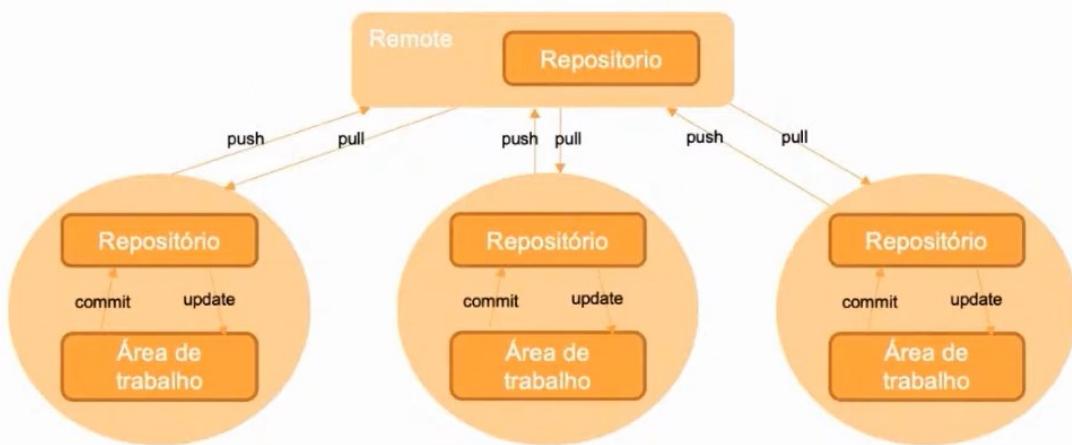
GIT

dio.

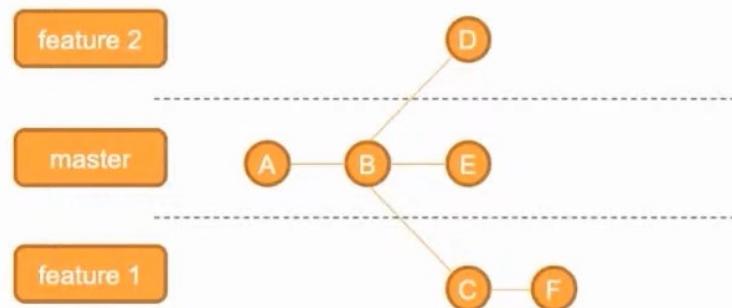
- DVCS – Distributed Version Control System
- Cada cópia do repositório contém o histórico completo de todas as alterações
- Open source
- Otimizado para desempenho

GIT - Remote

dio.



GIT - Branch



Fluxo de trabalho



** Praticar!!!

. Passo a passo para criar um repositório

- `git init` (Cria um repositório com o nome da pasta onde se está)

- `git status`

On branch master

```
No commits yet
```

. Configurações e commits

- git config user.name "michael Jackson" (se não usar '--global', a configuração é só no repositório)
- git config user.mail "michaeljackson@inter.com.br"
 - * usuário configurado.
- > gitignore e 1º commit
- git log (mostra últimos commits e detalhes)
- > Atalhos para os comandos do git
- git acm "Adiciona nova funcionalidade...." (por exemplo)

. Criando branches

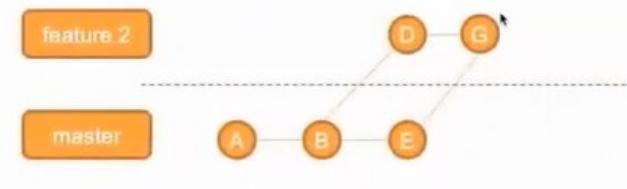
- git checkout -b feature1 (Cria e já muda para a branch criada)
- git status
- git l (l abreviado de 'log')

. Sobre merge e rebase

- git status
- git checkout feature2
- git log --graph --oneline --all (mostra tudo.. Todos commits)
- git merge master

GIT Merge

Merge master > feature 2



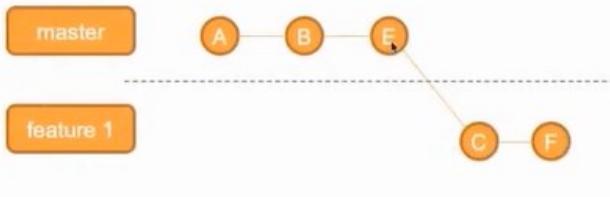
- git log --graph --oneline --all

--> Git Rebase (Refaz a base, reescreve a árvore de commits - muito cuidado, depende o caso)

- git checkout feature1
- git rebase master

GIT Rebase

Rebase master > feature1



Como utilizar o GitLab para seus projetos

- . Introdução ao GitLab

GitLab

- Gerenciamento de repósitorios
 - Similar ao Github e Bitbucket
 - Open source
 - Organização de repositórios em grupos
 - CI/CD
- .
- . Interface da plataforma e criação de projeto
 - . Apresentação do projeto no IntelliJ
 - . Como realizar o merge request
 - . Desenvolvendo uma pipeline e GitLab CI
 - . Visualizando a execução da pipeline
 - . Dúvidas e comentários finais

gitlab.com/dio-git_and_gitlab

Criando seu 1º Repositório no GitHub Para Compartilhar Seu Progresso

- . Projeto realizado ainda no curso de github
- . Projeto Entregue

Deployment e Roolback em Clusters Kubernetes

Deployment e Roolback

. Histórico de deployment

- * Histórico de atualizações de versões de uma aplicação, por exemplo.
- * Caso necessário, existem várias formas de voltar à uma versão anterior.
--> Não recomendado usar ‘remove deploy’ e sim, fazer um ‘apply’ da imagem com a versão nova, que substitui os pods gradualmente.

. History e undo

- kubectl apply -f deploy.yml --record ([mantém registrado o histórico de deployments](#))
- kubectl get pods
- kubectl rollout deployment httpd [nome do deploy] ([mostra o histórico](#))
- * Sempre usar número de versão na descrição do arquivo yml, para melhor identificação.
- kubectl rollout undo deployment [nome do deploy] (volta à versão)
- kubectl rollout history deployment [nome] ([lista histórico de deploy](#))
- * Cada undo vai voltando à anterior.
- kubectl rollout undo deployment [nome] --to-revision=1 ([volta pro primeiro deployment](#))
- ** Com o ‘--to-revision=x’, pode voltar à um deploy específico.

. Organizando o histórico de deployment

- * Se atentar para o nome do arquivo `yml` e imagem utilizada (informar num da versão)

- . **Secrets**

Secrets

Um Secret é um objeto que contém uma pequena quantidade de informação sensível, como senhas, tokens ou chaves. Este tipo de informação poderia, em outras circunstâncias, ser colocada diretamente em uma configuração de Pod ou em uma imagem de contêiner. O uso de Secrets evita que você tenha de incluir dados confidenciais no seu código.

Secrets

Secrets podem ser criados de forma independente dos Pods que os consomem. Isto reduz o risco de que o Secret e seus dados sejam expostos durante o processo de criação, visualização e edição ou atualização de Pods.

Exemplo

```
apiVersion: v1
kind: Secret
metadata:
  name: my-secret
type: Opaque
data:
  ROOT_PASSWORD: Senha123
```

* `secrets.yml`

CI-CD utilizando Kubernetes

CI-CD utilizando Kubernetes

- . O que é um CICD

CI/CD com Gitlab, Docker e Kubernetes

O que é Deploy?

↓

A implantação (Deploy) envolve mover o software de um ambiente controlado para outro. Um ambiente é um subconjunto de infraestrutura de TI usado para uma finalidade específica.

Integração contínua (CI)

↓

A integração contínua é uma prática de desenvolvimento de software em que os desenvolvedores, com frequência, juntam suas alterações de código em um repositório central. Depois disso, criações e testes são executados. Os principais objetivos da integração contínua são encontrar e investigar erros mais rapidamente, melhorar a qualidade do software e reduzir o tempo necessário para validar e lançar novas atualizações de software.

Fonte: <https://aws.amazon.com/pt/devops/what-is-devops/>

Entrega contínua (CD)

↓

A entrega contínua é uma prática de desenvolvimento de software em que alterações de código são criadas, testadas e preparadas automaticamente para liberação para produção. Ela expande com base na integração contínua, pela implantação de todas as alterações de código em um ambiente de teste e/ou ambiente de produção, após o estágio de criação. Quando a integração contínua for implementada adequadamente, os desenvolvedores sempre terão um artefato de criação pronto para ser implantado, e que passou por um processo de teste padronizado.



```
--> Criando novo projeto no GitLab - app-cicd-mdio
--> Logando no GitLab com PowerShell (comandos sugeridos no projeto criado no
    GitLab)
--> Criando pasta local .\cicd-mdio\ e entrando na pasta
--> Criando index.html básico na pasta com VSCode e rodando os comandos,
    conforme sugestão para fazer um push ('Push na existing folder'):
- git init --initial-branch=main
```

```
- git remote add origin https://gitlab.com/moiseslcar/app-cicd-mdio.git
```

```
- git add .  
- git commit -m "Initial commit"  
- git push -u origin main  
* usuario e senha  
*conferir no GitLab
```

. Criando uma pipeline de criação de imagens

```
--> Criar pasta app e jogar index.html pra dentro
```

```
* Criar o dockerfile na mesma
```

```
app > 📄 dockerfile > ...  
1  FROM httpd:latest  
2  
3  WORKDIR /usr/local/apache2/htdocs/  
4  
5  COPY index.html /usr/local/apache2/htdocs/  
6  
7  EXPOSE 80
```

```
--> Criar o arquivo cicd padrão do GitLab - .gitlab-ci.yml
```

```
index.html app dockerfile app .gitlab-ci.yml cicd-mdio X
.gitlab-ci.yml
1 stages:
2   - build
3
4 build_images:
5   stage: build
6   image: docker:20.10.16
7
8   services:
9     - docker:20.10.16-dind
10
11 variables:
12   DOCKER_TLS_CERTDIR: "/certs"
13
14 before_script:
15   - docker login -u $REGISTRY_USER -p $REGISTRY_PASS
16
17 script:
18   - docker build -t moiseslcar/app-cicd-mdio:1.0 app/
19   - docker push moiseslcar/app-cicd-mdio:1.0
20
```

```
--> Criar as varáveis REGISTRY_USER E REGISTRY_PASS no GitLab
--> Realizar o commit (na pasta .\cicd-mdio)
- git add .
- git commit -m "Criacao do ci/cd"
- git push -u origin main
* Ver a pipeline no GitLab e a imagem criada no dockerhub automaticamente.

. Criando uma VM do tipo Bastion para a gerênci a do cluster
.
.
-> Já conectado com putty na VM criada no GCP:
- gcloud config set account moiseslcar@gmail.com
- gcloud init
- y
* Copiar link gigante, jogar no browser para logar na gcp
```

- * Então instalar kubectl:
 - sudo apt-get install kubectl
- * Instalar auth plugin:
 - sudo apt-get install google-cloud-sdk-gke-gcloud-auth-plugin
- * Copiar link de conexão SSH ao cluster Kubernetes:
 - gcloud container clusters get-credentials meu-cluster --zone us-central1-c --project boreal-depth-369023
- ** Login: gcp
 - > ok.. Testar:
 - kubectl get nodes
 - pwd

.

Criando um pipeline de deployment no cluster Kubernetes

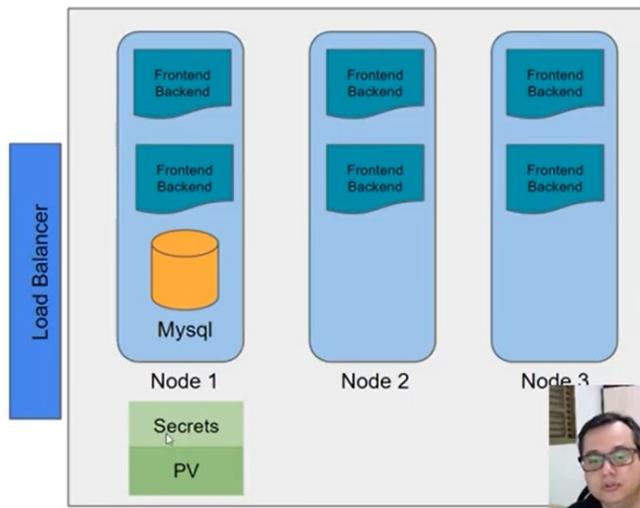
- * Na VM Bastion criada na GCP, instalar o git:
 -

--> Prática

6440 / 7750

Criando um pipeline de Deploy com GitLab e Kubernetes

Arquitetura do cluster e aplicação



Banco de dados

id	int
nome	varchar
email	varchar
comentario	varchar

<https://gitlab.com/denilsonbonatti/kubernetes-projeto2-dio>

--> Prática - Projeto - ENTREGAR

Google Cloud Foundation
Introdução ao Google Cloud Platform

Conteúdo mais teórico, análise geral

The screenshot shows a dark-themed learning interface with three main sections:

- O que é Cloud Computing?**
 - Sobre o Expert
 - Objetivos
 - Definição e Benefícios
 - Quando não usar Cloud
 - Tipos de Serviços
- Regiões e Zonas**
 - Definição de Regiões e Zonas
 - Pontos de Presença
 - Google Cloud Networking
 - Latência
 - Produtos
 - Responsabilidade Compartilhada
 - Service Level Agreement (SLA)
 - Documentação
 - Conclusão
- Materiais de Apoio e Questionário**
 - Materiais de apoio

A red button at the bottom of the 'Regiões e Zonas' section contains the text "CLIQUE AQUI PARA CONCLUIR ESSA AULA".

Gestão de identidades com Boas Práticas de Segurança

Pontos Importantes e Gestão de Identidades

- Pontos Importantes
- Gestão de Identidade

CLIQUE AQUI PARA CONCLUIR ESSA AULA

Questionário

- Certifique seu conhecimento

Hands On: Provisionando Usuários Cloud Identity

Hands On: Provisionando Usuários Cloud Identity

- Registro de Domínio
- Cloud Identity
- Admin Console
- Ativando Console GCP
- Criação de Projetos

CLIQUE AQUI PARA CONCLUIR ESSA AULA

Questionário

- Certifique seu conhecimento

Gestão de Recursos na GCP

Gestão de recursos

- Introdução e Políticas de Organização

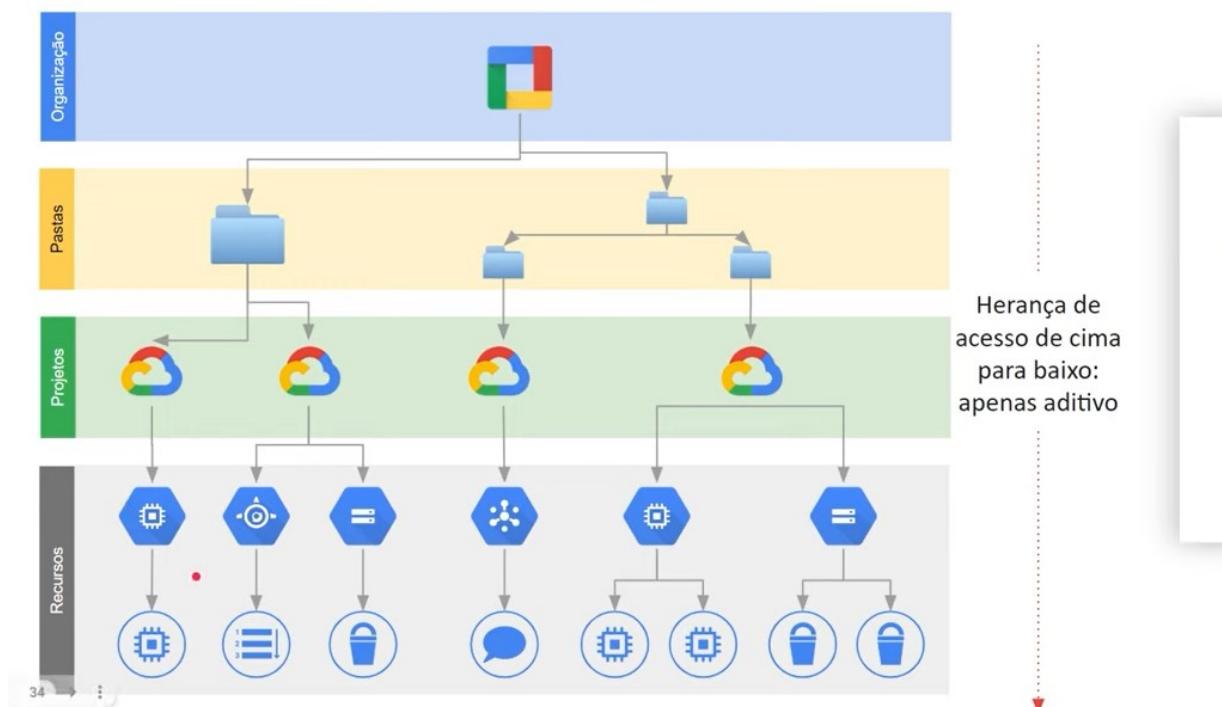
Objetivos

- Organizar nomes e recursos na nuvem
- Garantir processos e projetos estruturados

Motivações de uso

- Desejo ter uma hierarquia de recursos para poder controlar e descobrir quem é o responsável.
- Desejo ter controle e restrições dos recursos.
- Desejo ter meus projetos de forma organizada e segura com informações claras.
- Desejo automatizar processos para evitar erros humanos e criar padrões programáveis via código.

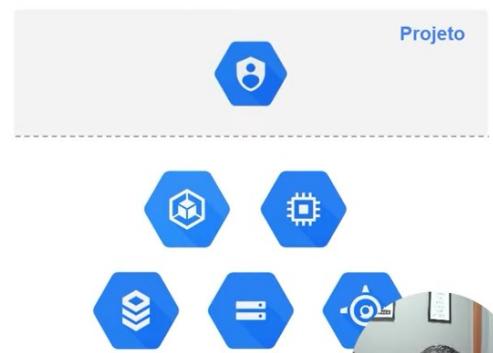
Hierarquia da Organização



A política efetiva para um recurso é a união da política definida neste recurso e a política herdada de seu pai.

Projetos

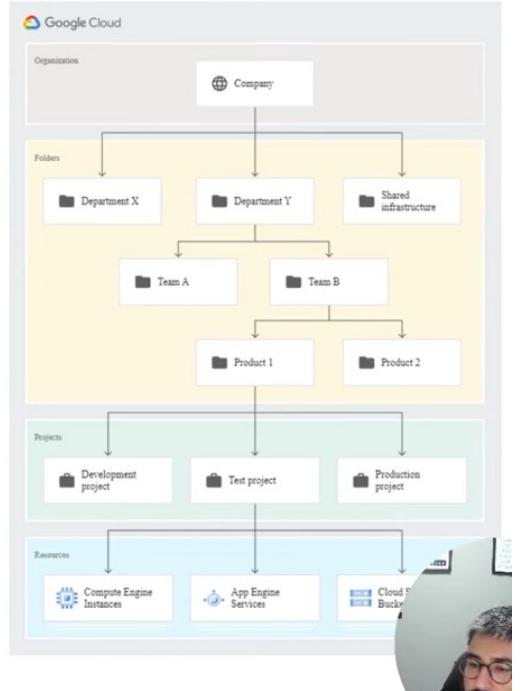
- Onde estão nossos workloads e produtos que estamos utilizando Ex: Compute Engine, Storage etc..
- Projetos são completamente isolados uns dos outros
- Utilizamos para agrupamentos de workloads divididos por uma lógica de negócio ex: financeira, jurídica, inovação.
- Os projetos podem fazer parte de uma organização
- O provisionamento é simples e gratuito



Pastas / Folder

As pastas são nodes no [Cloud Platform Resource Hierarchy](#). Uma pasta pode conter projetos, outras pastas ou uma combinação de ambos. As organizações podem usar pastas para agrupar projetos no nó da organização em uma hierarquia. Por exemplo, sua organização pode conter vários departamentos, cada um com seu próprio conjunto de recursos do Google Cloud. Com as pastas, você pode agrupar esses recursos por departamento ou agrupar recursos que compartilham políticas comuns do IAM. Cada pasta pode conter várias pastas ou recursos. No entanto, uma determinada pasta ou recurso pode ter somente um pai.

<https://cloud.google.com/resource-manager/docs/creating-managing-folders>

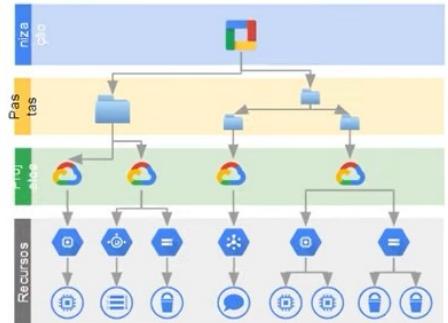


** Slides salvo na pasta do curso E:\DIO\GCP

Organização

O recurso Organização representa uma organização e é o nó **raiz na hierarquia de recursos do Google Cloud**. Ele é o ancestral hierárquico dos recursos de projeto e de pastas. O recurso **Organização** está intimamente associado a uma conta do Google Workspace ou do Cloud Identity. Quando um usuário com uma conta do Google Workspace ou do Cloud Identity cria um projeto do Google Cloud, um recurso Organização é provisionado automaticamente para ele.

Quando um usuário gerenciado cria um projeto, o requisito é que ele esteja em alguma organização. Se um usuário especificar uma organização e tiver as permissões corretas, o projeto será atribuído a ela.



<https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy#organizations>

--> Políticas da Organização Recomendadas

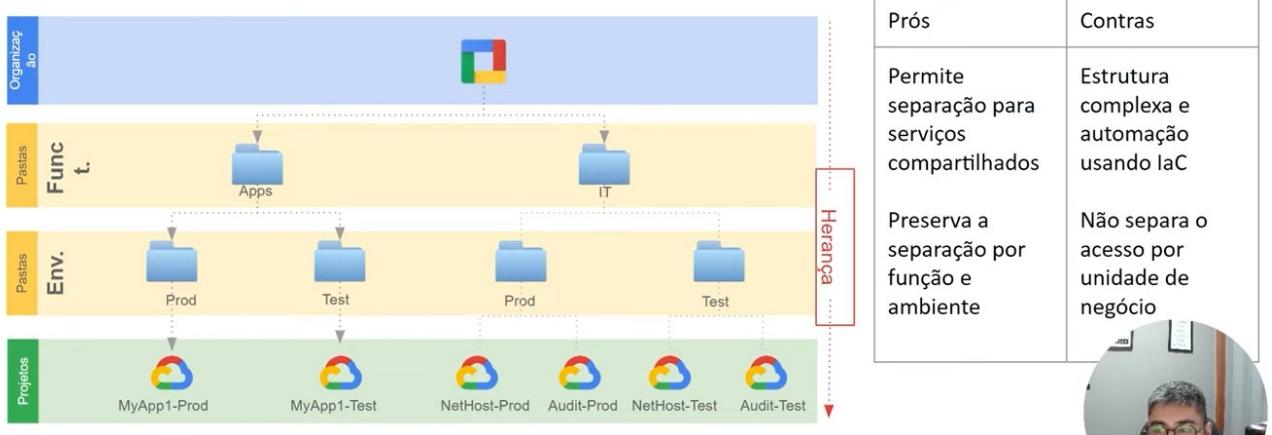
** Contúdo mais teórico.. Segue no Slide - E:\DIO\GCP (com links da GCP)

- Design da Organização

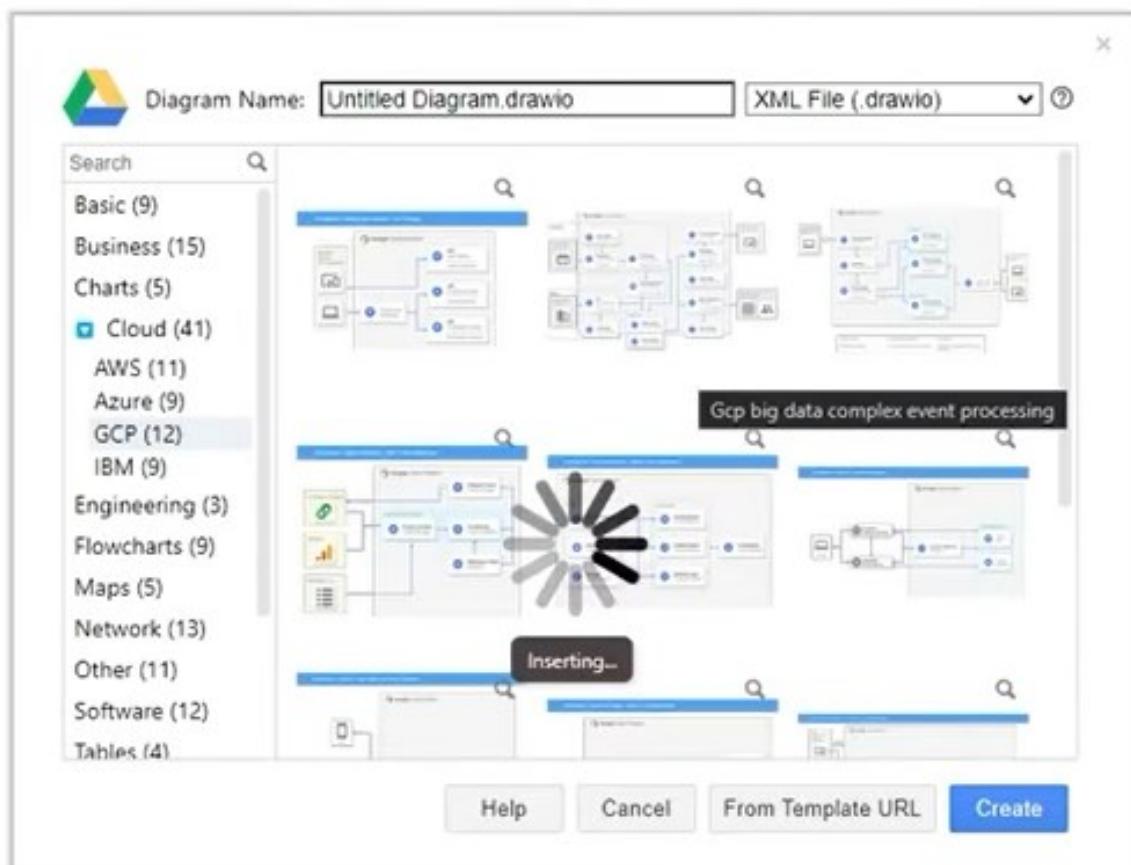
Design da Organização

	Poucos projetos	Muitos projetos
Complexidade do projeto	Baixa	Alta
Complexidade do IAM	Alta	Baixa
Tráfego de rede entre projetos	Menos frequente – N/A	Mais frequente – VPN/shared VPC
Menor privilégio	Mais difícil	Menos difícil

Hierarquia Baseada em Funções

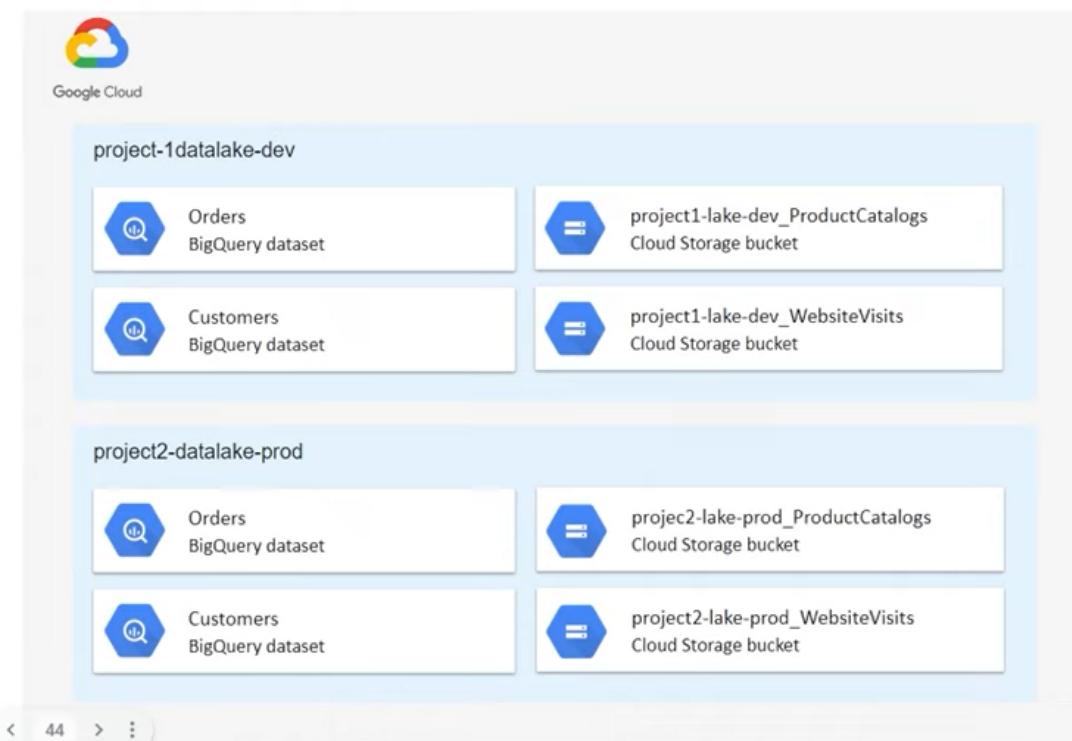


** Dica de ferramenta: draw.io

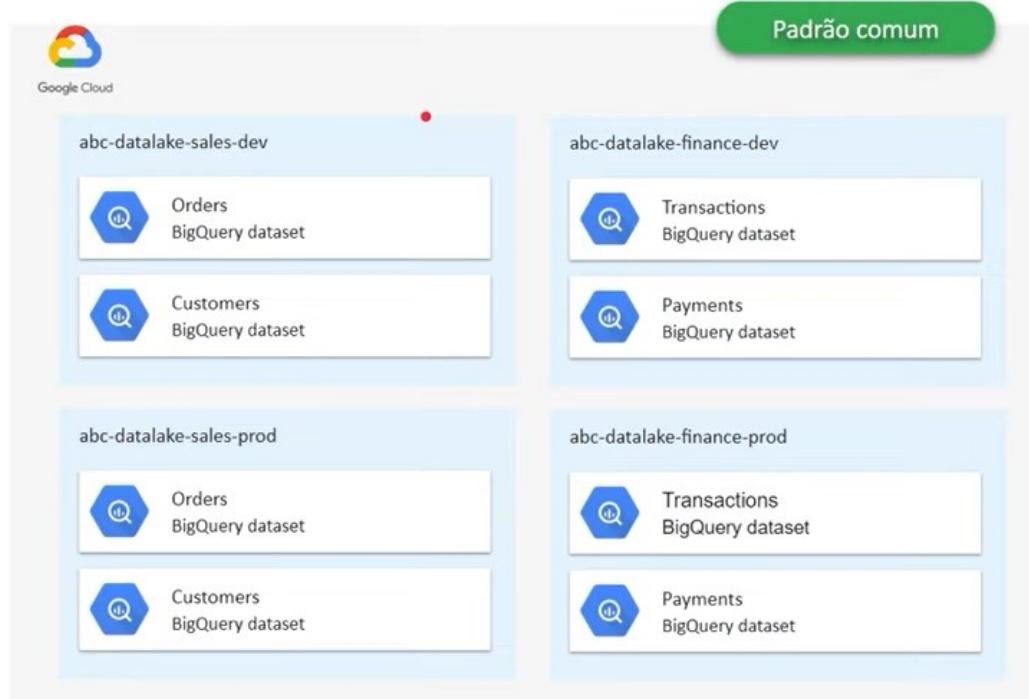


Um projeto para cada ambiente

Anti-pattern



**Um projeto para cada aplicação
e Ambiente**



Design: Nome do Projeto

	Nome do projeto	Número do projeto	ID do projeto
Atribuído	Pelo usuário	Automaticamente	Pelo usuário
Único globalmente	Não	Sim	Sim
Mútavel	Sim	Não	Não
Considerações de design	Nenhuma	Nenhuma	<p>Prefixo com o nome da empresa para ajudar a garantir exclusividade global</p> <p>Não inclua atributos que podem mudar no futuro, por exemplo: nome da equipe, tecnologia usada</p>

Organização do Projeto

- Um projeto por aplicativo ou serviço para cada ambiente
- Possível atributos adicionais: classificação de dados
- Esquema de nomenclatura de ID de projeto consistente para todos os projetos da empresa

Projeto por aplicação ou serviço
"empresa-departamento-produto-dev"

Dev

Projeto por aplicação ou serviço
"empresa-departamento-produto-test"

Test

Projeto por aplicação ou serviço
"empresa-departamento-produto-prod"

Prod

Estrutura de pastas

Níveis

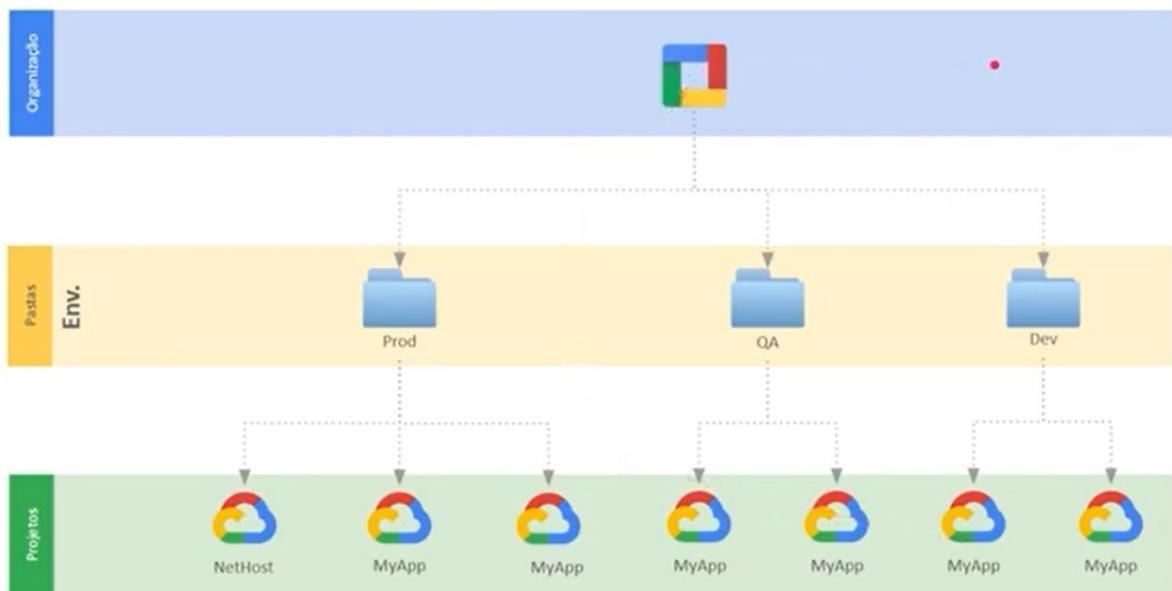
Na maior parte dos casos, três ou quatro níveis de pasta são suficientes

Extensível

A estrutura de pasta é flexível e facilmente extensível. Você pode **começar de forma simples**, e adicionar mais níveis quando necessário

- Hierarquia baseada em Ambientes

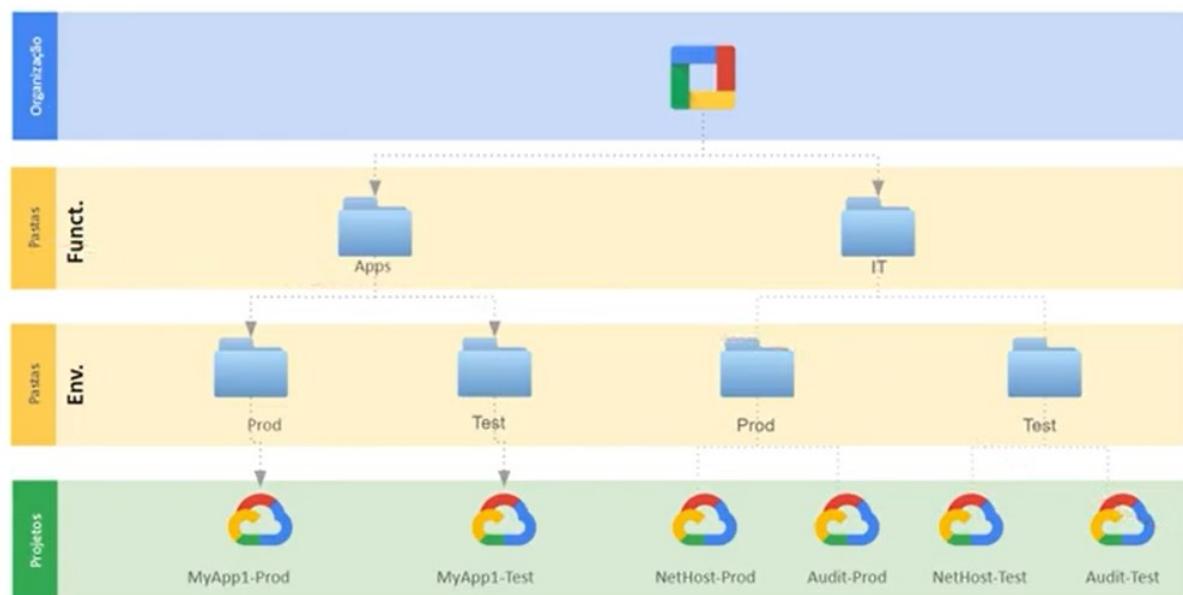
Hierarquia Baseada em Ambientes



* Estrutura bem simples, com seus prós e contras.

- . Hierarquia baseada em Funções

Hierarquia Baseada em Funções



- . Modelos Operacionais

Modelos Operacionais

- Centralizado: time operacional
- Descentralizado: engenheiros/ dev + time operacional
- Criação de Projetos: automatizado
- Cotas de Recursos
- Monitoramento de cotas: Cloud Operations

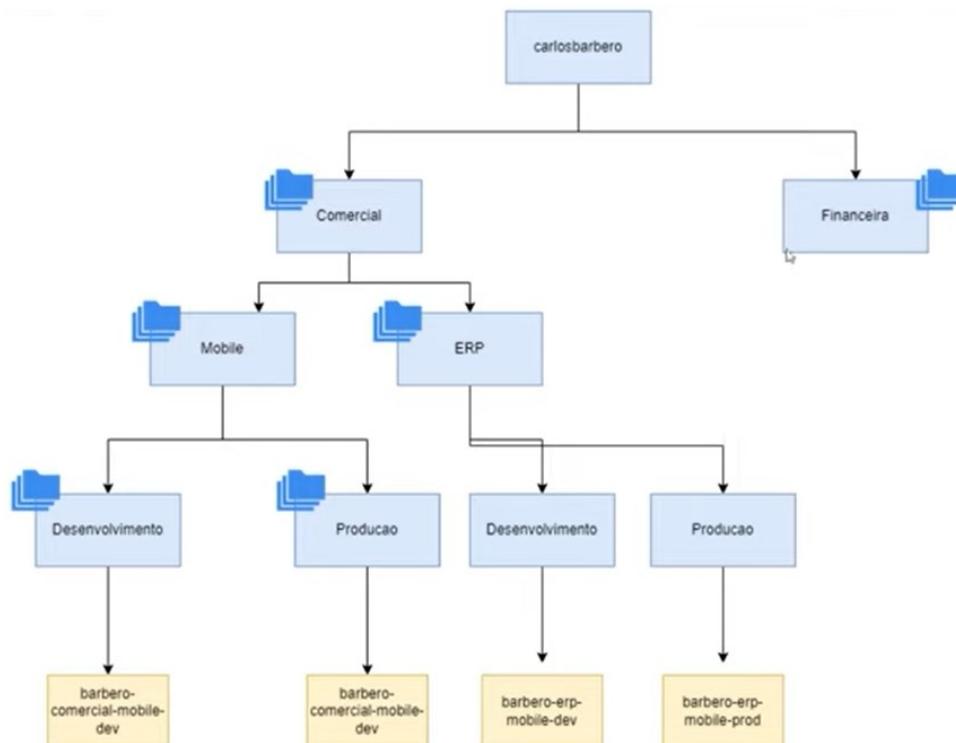
Exemplo Prático Organizando nossos projetos



Organizando Projetos na GCP

Organizando Nossos Projetos

- Desenhando Nossa Organização
- > draw.io



- Folder e Projetos

--> Conciliar Diagrama com Hierarquia de pastas na GCP

Nome	ID	Acessado pela última vez	Status
carlosbarbero.com.br	540829645030	29 de junho de 2022	
Comercial	104700958243	29 de junho de 2022	
Mobile	557081810182	29 de junho de 2022	
Producao	739780430969	29 de junho de 2022	
barbero-comercial-mobile-prod	barbero-comercial-mobil...	29 de junho de 2022	
Desenvolvimento	424067285780	29 de junho de 2022	
barbero-comercial-mobile-dev	barbero-comercial-mobil...	29 de junho de 2022	
projeto1	projeto1-354018	29 de junho de 2022	
ERP	951468414199	29 de junho de 2022	
Producao	475654499292	29 de junho de 2022	
barbero-erp-prod	barbero-erp-prod	29 de junho de 2022	
Desenvolvimento	720043405108	29 de junho de 2022	
barbero-erp-dev	barbero-erp-dev	29 de junho de 2022	

. Automação de processos com Terraform - Parte 1

<https://cloud.google.com/docs/terraform>

--> Preparando a parte ‘Financeira’ com o Terraform – pastas, subpastas e projetos

* Criar a Service account no projeto da GCP e a chave.

Projeto com Terraform:

```
main.tf
provider "google" {
  project = "barbero-comercial-mobile-dev"
  region  = "us-central1"
  zone    = "us-central1-c"
  credentials = "${file("serviceaccount.yaml")}"
}

resource "google_folder" "Financeiro" {
  display_name = "Financeiro"
  parent       = "organizations/540829645030"
}

resource "google_folder" "SalesForce" {
  display_name = "SalesForce"
  parent       = google_folder.Financeiro.name
}
```

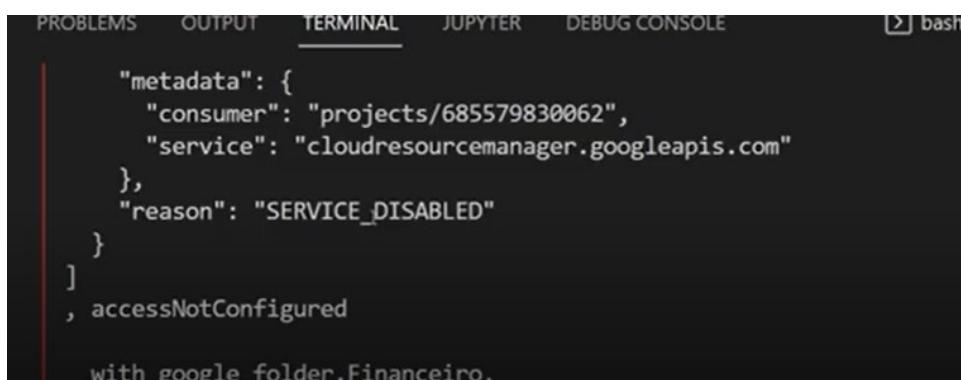
Projeto do instrutor

- > Arquivo `main.tf` com toda configuração Terraform - criação de pastas, subpastas e projetos com o id da billing account, folder-id, `False` para rede default, ..
- * Sempre necessário informar um billing account ao criar um projeto.
- > Arquivo `serviceaccount.yaml` com a chave (Copiar da chave da Service Account GCP)
- > Arquivo `.gitignore` com o `serviceaccount.yaml` para não expor a chave no github e gitlab)

. Automação de processos com Terraform – Parte 2

- `terraform init`
- `terraform plan`
- `Terraform apply`
- `yes`

* Notar erro – `Cloudresourcesapi = Disable`

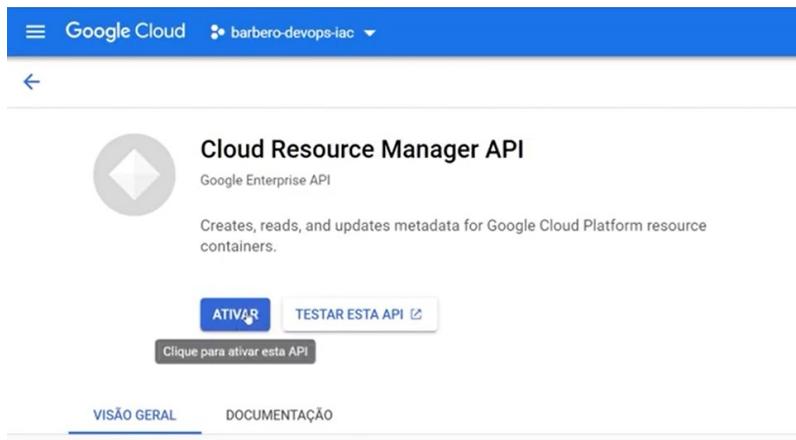


```
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE bash
"metadata": {
    "consumer": "projects/685579830062",
    "service": "cloudresourcemanager.googleapis.com"
},
"reason": "SERVICE_DISABLED"
]
, accessNotConfigured

with google.folder.Financeiro.
```

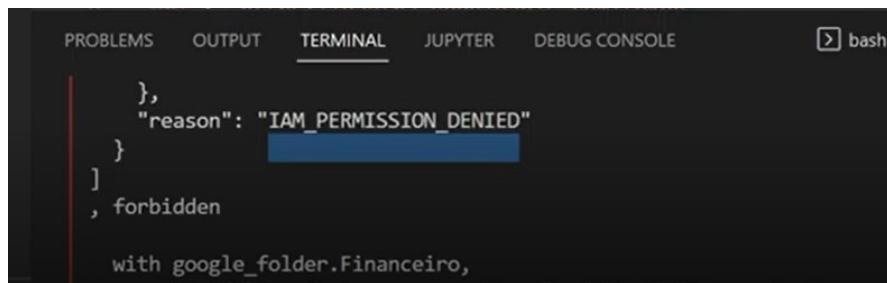
* No GCP, para todos os recursos vai ter um API associada - todos os recursos em cloud são através de API's

--> Então ativar a ‘Cloud Resource Manager API’ no projeto na GCP



- `terraform apply (novamente)`
- `yes`

* Notar o erro - *Iam Permission Denied*



```
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE bash
},
  "reason": "IAM_PERMISSION_DENIED"
}
],
, forbidden

with google_folder.Financeiro,
```

* Adicionar usuário com o email da service account no IAM, dentro do projeto, com a permissão ‘Administrador de recursos – Criador de pastas’

- `terraform apply` (novamente)

- `yes`

* Erro – Google billing api

Api de serviço - ativar ‘Cloud Billing API’

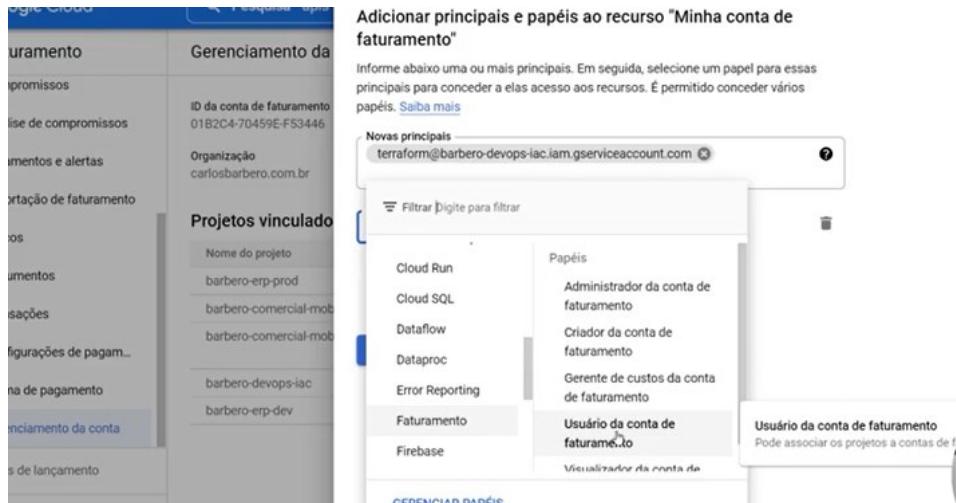
. Automação de processos com Terraform – Parte 3, 4 e 5

- `terraform apply` (novamente)

- `yes`

* Erro – conta de faturamento

* Adicionar participante no gerenciamento da conta de faturamento, com a permissão ‘Faturamento – Usuário da conta de faturamento’



- `terraform apply` (novamente)

- `yes`

* Erro – project creator

* Dar permissão ao usuário do terraform (no IAM) para, além de ‘criador de pastas’, criar projetos - ‘Administrador de recursos – Criador do projeto’.

- `terraform apply` (novamente)

- `yes`

* Depois de mais uns ajustes do instrutor nos seus projetos deletados e conta de faturamento, tudo certo – pastas e projetos criados.

. Admin Console - Terraform

--> Questões sobre grupos de acesso

> Console do workspace (Terraform com Google workspace)

> Console do GCP

* Implementação de Terraform para Admin Console do Google Workspace

> Baixar projeto no github do instrutor:

<https://github.com/digitalinnovationone/trilha-gcp-fundations-terraform-adminconsole>

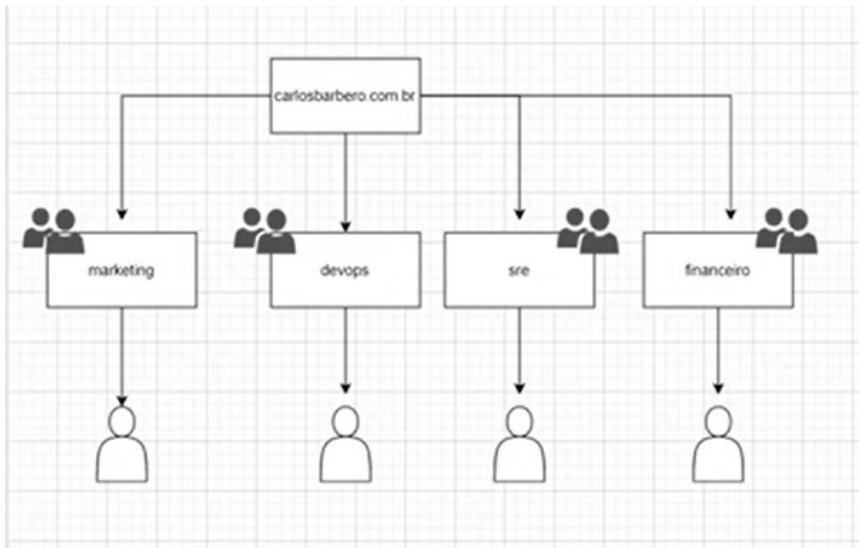
--> **Seguir o passo a passo** – mas antes, desenhar o diagrama do nosso projeto de usuários (Draw.io) - Figura*.

Em seguida, com o projeto baixado, seguir instruções:

> Habilitar Admin SDK API no projeto Devops

> Tela de permissão (Tela de consentimento)

> Criar a service account ‘terraformworkspace’, fazer ativações e criar chave (tipo JSON) e colar no arq. serviceaccount.yaml. ...



--> Após rodar o terraform e ok, conferir usuários e grupo no \Google Admin

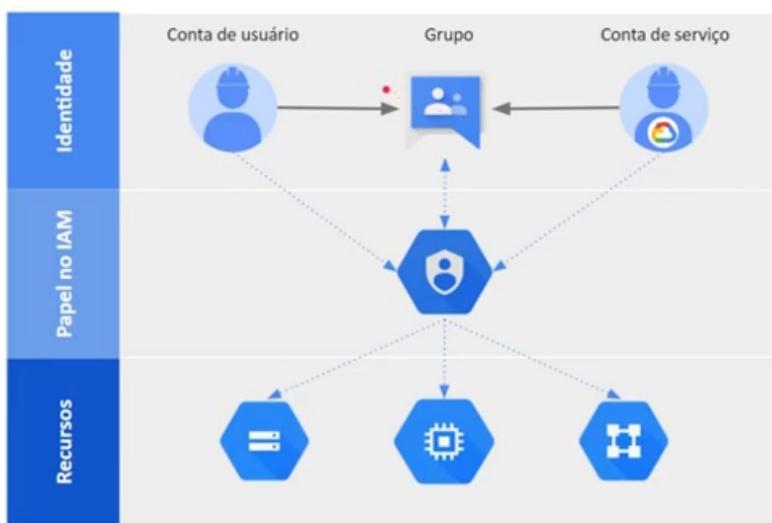
* Deletar então os recursos no GCP.

Gestão de Acessos e Segurança na GCP

Gestão de Acessos

- . Políticas do Cloud IAM

Políticas do Cloud IAM



O IAM permite que você adote o princípio de segurança de privilégio mínimo, para conceder apenas o acesso necessário aos recursos.

As políticas do IAM podem ser aplicadas nos níveis de Organização, Pasta e Projeto. Para determinadas soluções do Google Cloud, elas também podem ser aplicadas no nível do recurso.



- . Cloud IAM e Política de Recursos

Como o Cloud IAM Funciona?

Com o IAM, você gerencia o controle de acesso definindo *quem* (identidade) tem *qual acesso* (papel) a *que recurso*. As instâncias de máquina virtual do Compute Engine, os clusters do Google Kubernetes Engine (GKE) e os buckets do Cloud Storage são todos recursos do Google Cloud. As organizações, pastas e projetos que você usa para organizar seus recursos também são recursos.

No IAM, a permissão para acessar um recurso não é concedida *diretamente* ao usuário final. Em vez disso, as permissões são agrupadas em *papéis*, que são concedidos a *principais* autenticados. No passado, o IAM muitas vezes se referia aos principais como *membros*. Algumas APIs ainda usam esse termo.

Uma *política de permissão*, também conhecida como *política do IAM*, define e aplica os papéis concedidos aos principais. Cada política de permissão é anexada a um recurso. Quando um principal autenticado tenta acessar um recurso, o IAM verifica a política do recurso para determinar se a ação é permitida.

Condições Cloud IAM



Quem ?



pode fazer o
que



em qual
recurso



sob quais
condições

O que são Contas de Serviço?

- Usado por **aplicativos** que precisam fazer uso de **APIs do Google Cloud**.
- Autenticação baseada em **pares de chaves**
- Gerenciado por projeto no **Cloud Console**, mas pode ser adicionado a grupos no **Admin Console**.
- Nova Conta de Serviço
- Conta de Serviço default.

. Políticas recomendadas para Proteger Contas de Serviço

- ➔ Documentação do GCP
- ➔ Nunca criar chaves públicas para as service accounts default
- ➔ Limitar os privilégios às suas service accounts

. Cloud Security Command Center

Cloud Security Command Center



O [Security Command Center](#) é o serviço centralizado de vulnerabilidade e relatórios de ameaças do Google Cloud. O Security Command Center ajuda a fortalecer a postura de segurança ao avaliar a superfície de segurança e ataque de dados, fornecer inventário e descoberta de ativos e identificar configurações incorretas, vulnerabilidades e ameaças, além de ajudar você a mitigar e corrigir riscos.



- ➔ Documentação GCP
 - . Nível Standart (Gratuito)
 - . Nível Premium
 - . Relatórios a nível de segurança

.Políticas da Organização

Políticas de Organização

O serviço de política da organização restringe as configurações de recursos permitidas. As políticas podem ser aplicadas à organização, pastas e projetos. Requer função IAM: política da organização / administrador da política da organização. [LINK](#)

[Policy Intelligence](#) : O Policy Intelligence utiliza os recursos de ML e IA do Google para ajudar as empresas compreender e gerenciar suas políticas de IAM para reduzir o risco

→ Documentação do GCP

Redes e Monitoramento na GCP

Redes e Monitoramento

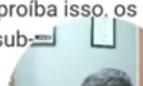
. Redes VPC

Redes VPC

Uma rede de nuvem privada virtual (VPC) é uma versão virtual de uma rede física, implementada dentro da rede de produção do Google, usando [Andromeda](#). Uma rede VPC fornece o seguinte:

- Fornece conectividade para suas [instâncias de máquina virtual \(VM\) do Compute Engine](#), incluindo [clusters do Google Kubernetes Engine \(GKE\)](#), [ambiente flexível do App Engine](#) e outros produtos do Google Cloud baseados em VMs do Compute Engine.
- Oferece balanceamento de carga TCP/UDP interno nativo e sistemas de proxy para平衡amento de carga de HTTP(S) internos.
- Conecta-se a redes locais usando túneis do Cloud VPN e anexos do Cloud Interconnect.
- Distribui o tráfego dos平衡adores de carga externos do Google Cloud para back-ends.

Os projetos podem conter várias redes VPC. A menos que você crie uma política organizacional que proíba isso, os novos projetos começam com uma rede padrão (uma rede VPC de modo automático) que tem uma sub-rede (sub-rede) em cada região.



- . Infraestrutura de rede global

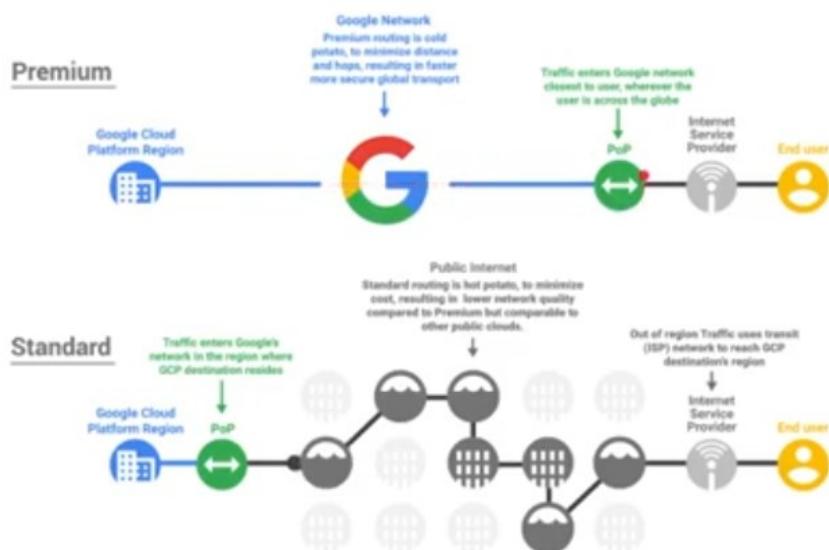
Infraestrutura de Rede Global



<https://cloud.google.com/about/locations#network>

- . Camada de Rede

Camada de Rede



<https://gweb-network-tier-demo-archive.firebaseio.com/>

. Proteção e Criptografia

Proteção e Criptografia

Primary focus
of this document

Application [Red Bar] Google Cloud Platform services

Platform [Green Bar] Database and file storage: protected by AES256 or AES128 encryption

Infrastructure [Yellow Bar] Distributed file system: data chunks in storage systems protected by AES256 encryption with integrity

Block storage [Yellow Bar]

Hardware [Blue Bar] Storage devices: protected by AES256 or AES128 encryption

<https://cloud.google.com/docs/security/encryption/default-encryption?hl=pt-br>
<https://cloud.google.com/docs/security/encryption-in-transit>

➔ Muito recurso na documentação do GCP (links)

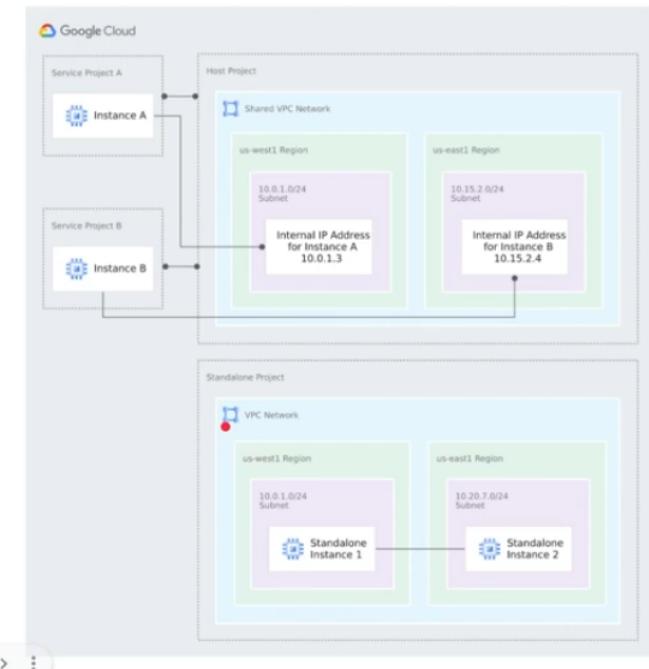
. Conceitos VPC

Conceitos VPC



<https://cloud.google.com/vpc/docs/vpc?hl=pt-br>

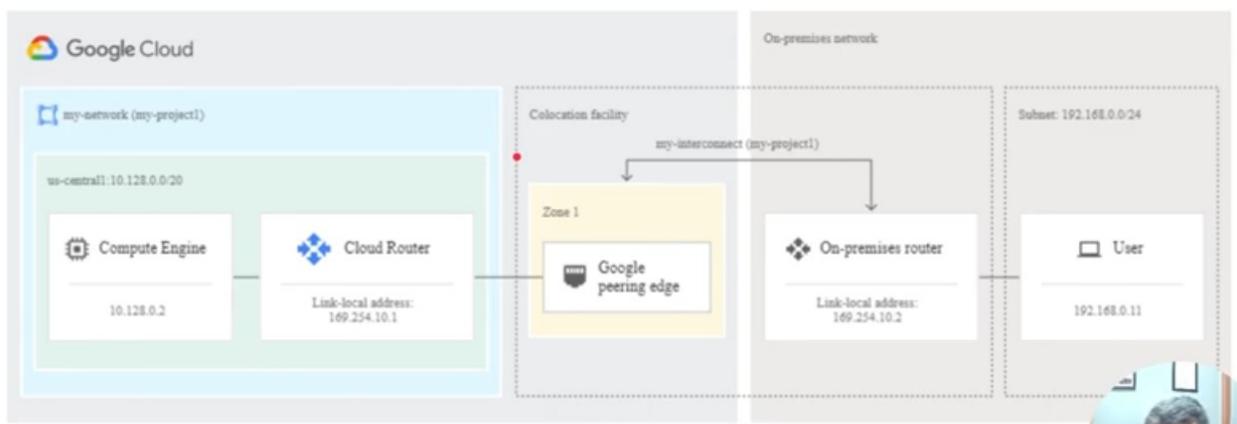
Conceitos Shared VPC



https://cloud.google.com/vpc/docs/shared-vpc#use_cases

- Google Cloud Interconnect

Google Cloud Interconnect



<https://cloud.google.com/network-connectivity/docs/interconnect/concepts/dedicated-overview#how-it-works-dedicated>



- Monitoramento e Registros

Google Cloud Operations



<https://cloud.google.com/stackdriver/docs>

Activity

The screenshot shows the Google Cloud Activity page. The top navigation bar includes 'Google Cloud', a dropdown for 'barbero-devops-iac', a search bar with 'Pesquisa' placeholder, and various icons for account and settings. The left sidebar has 'Visão geral do Cl...' and 'Ver todos os produtos' buttons. The main area has tabs for 'PAINEL', 'ATIVIDADE' (which is selected), and 'RECOMENDAÇÕES'. A date filter '07/07/2022' is applied. The activity list shows 14 entries from 14:44, all related to 'google.longrunning.Operations.GetOperation' and executed by 'carlos@carlosbarbero.com.br'. Each entry has a dropdown arrow icon.

14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	Concluída:google.api.serviceusage.v1.ServiceUs...	carlos@carlosbarbero.com.br executou google.api.serviceusage.v1.Servic...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼
14:44	google.longrunning.Operations.GetOperation	carlos@carlosbarbero.com.br executou google.longrunning.Operations.Get...	▼

<https://cloud.google.com/identity-platform/docs/activity-logging>

<https://console.cloud.google.com/home/activity?project=barbero-devops-iac>

Google Cloud Loggin

The screenshot shows the Google Cloud Logging interface. On the left, there's a sidebar with 'Operations' and 'Registros' sections. The main area has a 'Buscador de registros' header with tabs for 'Consulta' (selected), 'Recente (0)', 'Salvo (0)', 'Sugeridos (0)', 'Biblioteca', and buttons for 'REFINAR ESCOPO' and 'Projeto'. Below this are filters for 'Últimos 1 hora', 'Pesquisar todos os campos', 'Recurso', 'Nome do registro', and 'Gravidade'. A toggle switch 'Exibir consulta' is also present. The results section shows a single entry with a timestamp '13 de jul, 16:08'. At the bottom, there are buttons for 'Correlacionar por' and other options.

<https://cloud.google.com/logging/docs/how-to?hl=pt-br>

Gestão de Dados e Controle de Custos na GCP

Gestão de Dados

- . Google Cloud KMS

Objetivo

Desejo armazenar dados com segurança, descobrir o quanto sensível eles são e gerenciar quem pode acessá-los, com o objetivo de manter os dados seguros cumprindo as regulamentações exigidas em meu país.

Google Cloud KMS

Key Management Service (KMS) hospedado na nuvem Permite que você gerencie a criptografia para seus serviços em Cloud da mesma maneira que você faz on-premises.

Sua empresa pode gerar, usar, rotacionar e destruir chaves de criptografia AES256. O Cloud KMS está integrado ao IAM e ao Cloud Audit Logging para que você possa gerenciar permissões em chaves individuais e monitorar como elas são usadas.

Funcionalidades

- AES256 key
- Criptografar e descriptografar via API
- Rotação de chave automatizada e à vontade
- Atraso para destruição de chave
- Alta disponibilidade global
- Suporta criptografia de software e hardware

Hardware encryption

- Serviço baseado em nuvem de Hardware Security Module (HSM)
- [FIPS 140-2 Level 3 certified](#)
- Chaves de criptografia de hosts
- Executa operações criptográficas

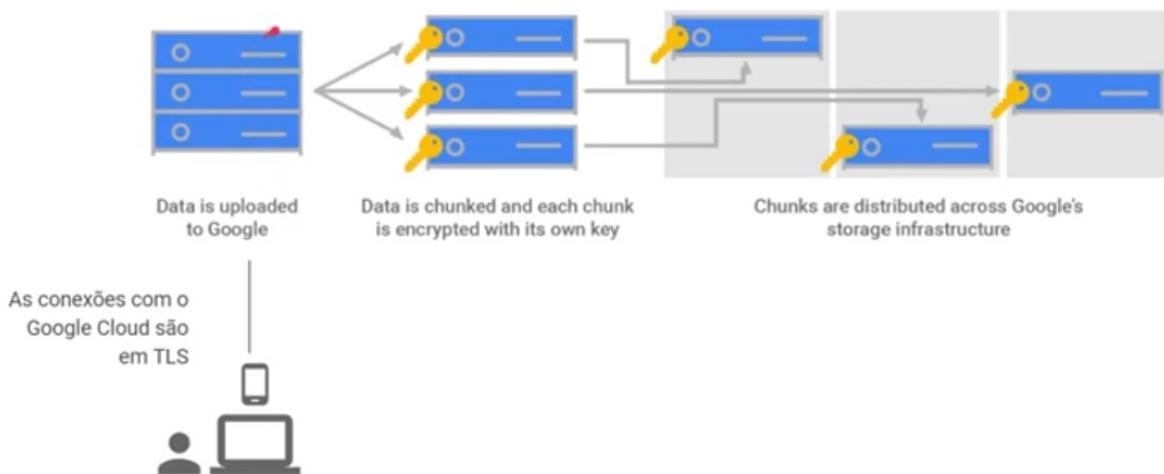


<https://cloud.google.com/security-key-management?hl=pt-br#section-5>

➔ Documentação oficial com leitura fundamental e laboratórios.

. Criptografia Default

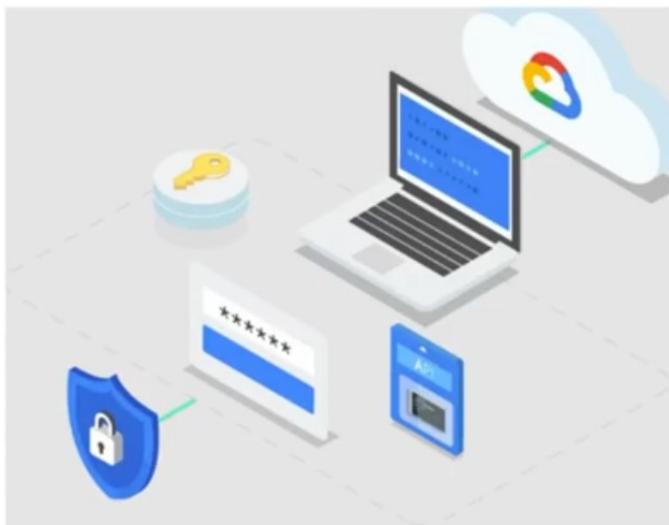
Criptografia Default



<https://cloud.google.com/docs/security/encryption/default-encryption#:~:text=Google%20uses%20the%20Advanced%20Encryption,to%202015%20that%20use%20AES128.>

- . Secret Manager

Secret Manager

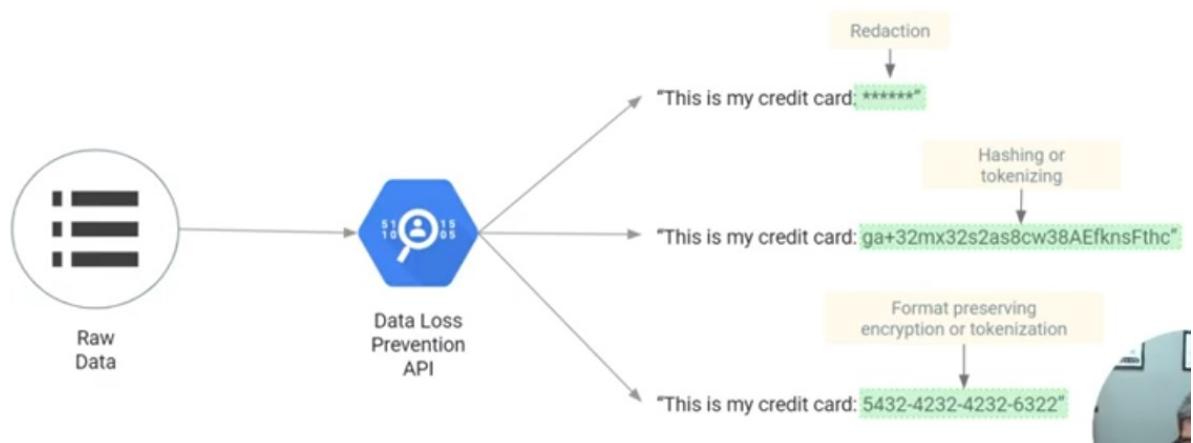


<https://cloud.google.com/secret-manager>

- . Google Cloud DLP

Google Cloud DLP

Fornece uma API programática para detecção de dados PII



➔ Dados PII = Dados pessoais do usuário

<https://cloud.google.com/dlp#section-1>

Google Cloud DLP

ID	Job Title	Phone	Comments
359740	Senior Engineer	307-964-0673	Please email them at jane@imadethisup.com
981587	VP, Engineer	713-910-6787	none
394091	Lawyer	692-398-4146	Updated phone to: 692-398-4146
986941	Senior Ops Manager	294-967-5508	none
490456	Junior Ops Manager	791-954-3281	Tried to verify account with their SSN 222-44-5555

=

ID (FPE)	Job Title	Phone	Comments
438422	Engineer	307- ### - ####	Please email them at [Found Email]
530375	Engineer	713- ### - ####	none
496534	Lawyer	692- ### - ####	Updated phone to: 692- ### - ####
242348	Ops	294- ### - ####	none
593887	Ops	791- ### - ####	Tried to verify account with their SSN [Found SSN]

- . Google Cloud Storage

Google Cloud Storage



Armazenamento de objetos para empresas de todos os tamanhos. Armazene qualquer quantidade de dados. Recupere-os quantas vezes quiser.

<https://cloud.google.com/storage>

Classes de armazenamento para qualquer carga de trabalho

Armazene dados em várias classes de armazenamento e economize sem precisar sacrificar o desempenho. É possível começar com uma classe que corresponde ao uso atual e depois reconfigurar para economizar.

[Standard Storage](#): recomendado para dados que precisam ser acessados com muita frequência, incluindo sites, streaming de vídeos e apps para dispositivos móveis.



[Nearline Storage](#): baixo custo. Recomendado para dados que podem ser armazenados por no mínimo 30 dias, incluindo backup de dados e conteúdo multimídia infrequente.

[Coldline Storage](#): custo muito baixo. Recomendado para dados que podem ser armazenados por no mínimo 90 dias, incluindo recuperação de desastres.

[Archive Storage](#): o custo mais baixo de todos. Recomendado para dados que podem ser armazenados por no mínimo 365 dias, incluindo arquivos regulatórios.

Controle de Custos

- . FinOps - Parte 1

Controle de Custos

FinOps

FinOps é uma prática cultural e disciplina de gerenciamento financeiro em nuvem em evolução que permite que as organizações obtenham o máximo valor comercial, ajudando as equipes de engenharia, finanças, tecnologia e negócios a colaborar em decisões de gastos orientadas por dados.

(Definition Updated: November 2021 by the FinOps Foundation
Technical Advisory Council

<https://cloud.google.com/resources/cloud-finops-getting-started-whitepaper>
<https://cloud.google.com/learn/what-is-finops#section-7>
<https://www.finops.org/introduction/what-is-finops/>

03/01/2023 – 6940 / 7750

Conceituando IAC e CI/CD

- . Infraestrutura como Código: IAC

IAC

Ferramentas	Iac é uma ferramenta de automação de processos, voltada para provisionamento de recursos.
Gestão	Para fazer alterações na infraestrutura, os engenheiros de automação devem fazer uma solicitação no sistema para implantar na produção.
Controle de versão de código	Toda o código é configurado e armazenado em ferramentas de controles de versão como por exemplo git
Processo de implantação	Através de pipelines



Benefícios

- Controlado pelos commits no git.
- Declarativo
- Infraestrutura diferencial entre o estado desejado e o estado atual.
- Módulos reutilizáveis em uma organização

. Produtos

Produtos



Google
Cloud Deployment
Manager



<https://cloud.google.com/deployment-manager/docs>

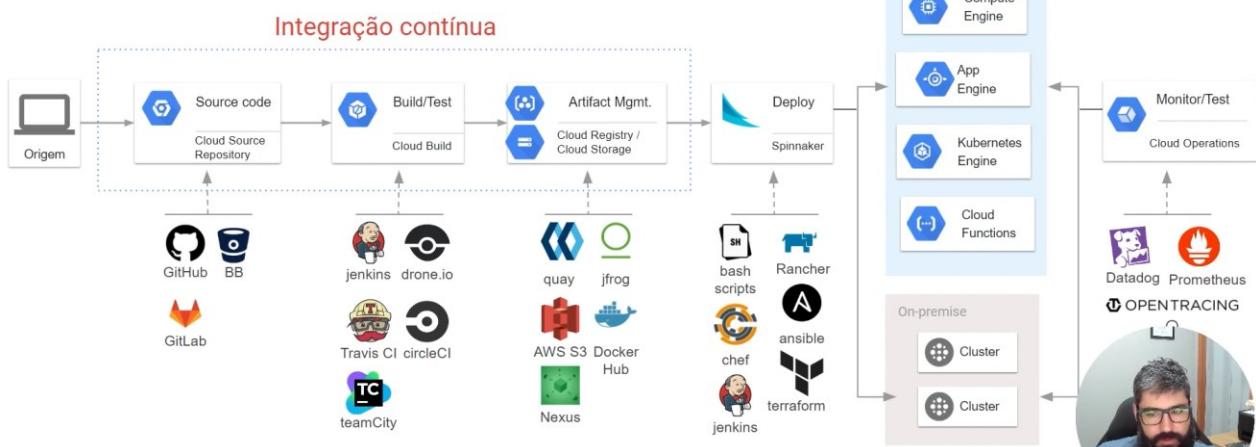
<https://cloud.google.com/docs/terraform>

> Google Cloud Deployment Manager - somente para cloud do Google *

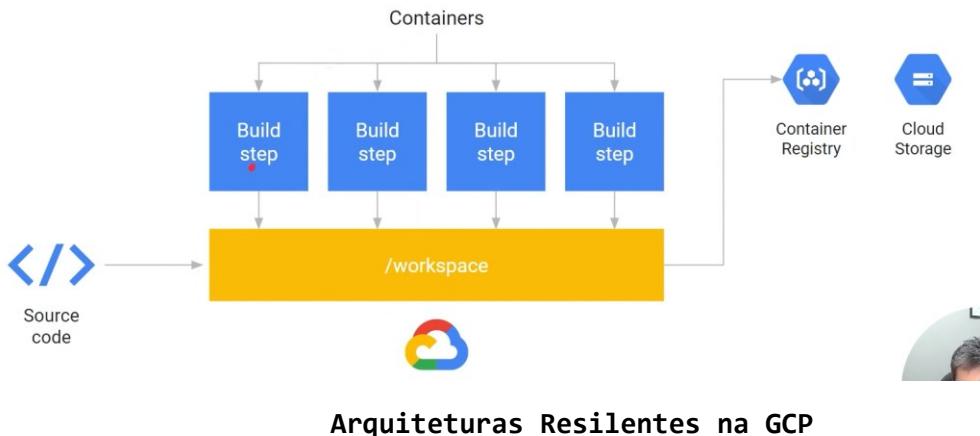
. Integração e Entrega \contínua (CI/CD)



Ferramentas



Google Cloud Build



Arquiteturas Resilientes na GCP

- Disponibilidade e Nível de Serviço

Nível de Serviço

SLI

Indicador

Quantidade mensurável que representa o que é importante para os usuários.

Frequentemente relacionado a confiabilidade ou desempenho.

Exemplo:

- Taxa de erro do Load balancer

SLO

Objetivo

O alvo que você deseja que seu SLI alcance

Exemplo:

- Taxa de erro do Load balancer <0.01%

SLA

Acordo

Consequências quando o SLO não é atendido. Aplicável apenas para serviços GA.

Freqüentemente, um Crédito Financeiro: porcentagem da fatura mensal que será creditada em futuras faturas mensais.

Vinculado à definição de tempo de inatividade e período mínimo de interrupção que é considerado tempo de inatividade.

Exemplo:

- 99,0% a <99,99%: 10% de crédito
- 95,0 a <99,0%: 25% de crédito
- Tempo de inatividade mínimo = 60s

Arquiteturas Resilientes

1. Redundância geográfica
2. Serviços gerenciados
3. Alta disponibilidade
4. Automatizado

- . Gestão de Incidentes

O que devo informar?

- Recurso
- Nome
- Região
- Localização
- Logs
- Links
- Resumo do que está acontecendo.
- Escolher a Prioridade

04/01/2023

Desenhando Sua Organização de Pastas e Projetos e Grupos de Acessos na Google Cloud Platform

> Entregar Projeto com **Draw.io**

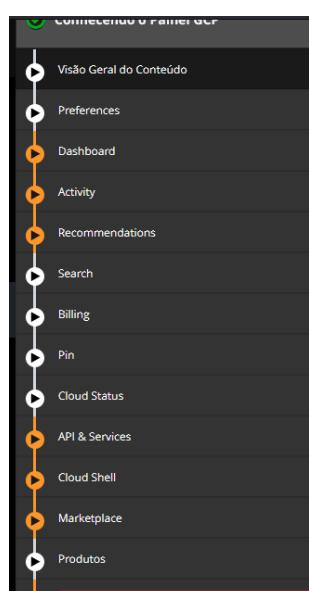
Definição de Orçamento de Billing na Google Cloud Platform

> Entregar Projeto com **print das config no console**

Exportação do Billing na Google Cloud Platform

> Entregar projeto com **print das Config de Exportação da Billing ao BigQuery**

Conhecendo o Painel da GCP



Personalizando Dashboard e Trocando as Prefeências de Idioma

- > Entregar Projeto com **print do Dashboard personalizado**

Utilizando o Cloud Shell

- > Entregar Projeto com **print da aplicação java testada no Cloud Shell**

04/01/23 _ 7200/7750

DEVOPS NA NUVEM COM GOOGLE CLOUD PLATAFORM (GCP)

Introdução ao Conceito de Devops e Preparação do Ambiente GCloud SDK

Introdução

- . Apresentação Inicial
- . O que é Devops
 - Uma cultura de desenvolvimento que mistura Ferramentas, Processos e Pessoas.
 - * Ver documentação do GCP sobre Devops

GCloud SDK

- . Introdução
 - * Documentação
- . Gcloud SDK Instalação Linux
 - > No exemplo, é instalado o Gcloud SDK em uma VM Ubuntu do próprio Google e realizado autenticação.
- . Gcloud SDK Instalação Windows
 - > Baixar instalador e executar - Uso no PowerShell
- . Gcloud SDK Create Compute Engine
 - > Exemplo usando a linha de comando sugerida pelo console ao criar uma nova instância de VM na GCP, usando o Cloud Shell.
- . Gcloud SDK Comandos e Documentação

Trabalhando com Cloud Build

Cloud Build

- . Introdução
 - Plataforma de CI/CD que controla tudo. Não precisa se preocupar com as VMs , updates...

. Exemplo Cloud Build Basic Config - Parte 1

Links dos exemplos do Google no Github -

github.com/GoogleCloudPlatform/cloud-build-samples
github.com/GoogleCloudPlatform/cloud-builders

. Exemplo Cloud Build Basic Config - Parte 2

* Após a alteração do cloudbuild.yaml, subir o build no PowerShell com o comando:

- `gcloud builds submit --config=cloudbuild.yaml`

. Exemplo Cloud Build Maven

> Usando exemplo de aplicação java, conferindo também o arquivo cloudbuild.yaml e os arquivos java.

. Cloud Build Substitutions - Exemplo Python

* Observação sobre variáveis

> Usando um exemplo do Python, subindo a aplicação no cloud run, pelo cloud build.

> Comando com novo parâmetro:

- `gcloud builds submit --config=cloudbuild.yaml --substitutions=_BUCKET_NAME=projeto1pytest`

→ URL para testar o “Hello World” - no Cloud Run..

. Cloud Build Gatilhos / Triggers - Exemplo NodeJs

> Gatilhos a partir de um git push, disparando o cloud build.

15/01/23 - 7400 / 7750

Trabalhando com Terraform

Terraform

. Introdução

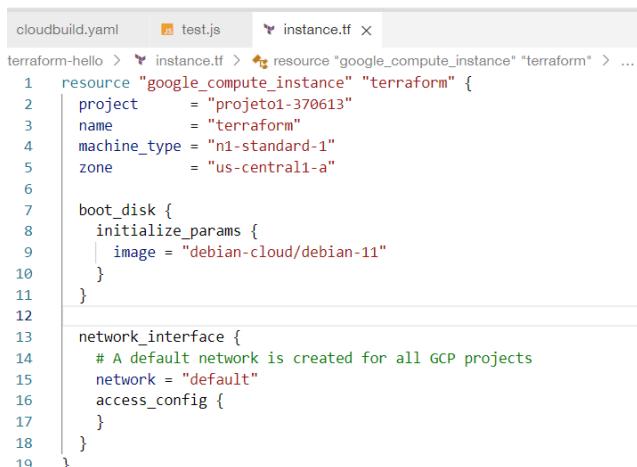
> Documentação Terraform - install, config..

. Terraform hello

* Terraform já vem instalado no Cloud Shell

> Como configura o terraform.. arquivos .tf

- Criação de um `instance.tf` básico para provisionar uma VM na GCP:
 - `terraform init` (pasta `terraform-hello`)
 - `terraform plan`
 - `terraform apply`
 - `terraform destroy`



```
cloudbuild.yaml  test.js  instance.tf ×  
terraformer-hello > instance.tf > resource "google_compute_instance" "terraformer" {  
  1   resource "google_compute_instance" "terraformer" {  
  2     project      = "projeto1-370613"  
  3     name         = "terraformer"  
  4     machine_type = "n1-standard-1"  
  5     zone         = "us-central1-a"  
  6  
  7     boot_disk {  
  8       initialize_params {  
  9         image = "debian-cloud/debian-11"  
10       }  
11     }  
12  
13     network_interface {  
14       # A default network is created for all GCP projects  
15       network = "default"  
16       access_config {  
17       }  
18     }  
19   }
```

```

resource "google_compute_instance" "terraform" {
  project      = "projeto1-370613"
  name         = "terraform"
  machine_type = "n1-standard-1"
  zone         = "us-central1-a"
  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-11"
    }
  }
  network_interface {
    # A default network is created for all GCP projects
    network = "default"
    access_config {
    }
  }
}

```

> Conferir a Compute Engine na GPC e o *destroy*, que elimina a VM.

. Estado do Terraform

> Salvando o estado no Cloud Storage - [Arquivo `main.tf` na pasta `terraform-exemplo2`](#)

- . Criar o bucket *projeto1terraform*
- `terraform init`
- `terraform apply`

```

terraform {
  required_providers {
    google = {
      source = "hashicorp/google"
    }
  }
  backend "gcs" {
    bucket = "projeto1terraform"
    prefix = "terraform/state"
  }
}
provider "google" {
  project = "projeto1-370613"
  region  = "us-central1"
  zone    = "us-central1-c"
}
resource "google_compute_network" "vpc_network" {
  name        = "terraform-network"
}
resource "google_compute_instance" "vm_instance" {
  name        = "terraform-instance"
  machine_type = "f1-micro"
  tags = ["prod"]
  boot_disk {

```

```

    initialize_params {
      image = "debian-cloud/debian-11"
    }
  }
  network_interface {
    # A default network is created for all GCP projects
    network = "default"
    access_config {
    }
  }
}

```

. Variáveis e Saídas

> Adicionado arquivos *variables.tf* e *outputs.tf*

. O que é legado

> Para projetos que não foram criados com o Terraform - importar o terraform no legado - ferramenta para exportar o projeto existente para o terraform

. Exportando um projeto

> Documentação - <https://cloud.google.com/docs/terraform/resource-management/export?hl=pt-br>

```

- gcloud components install config-connector
- gcloud beta resource-config bulk-export \
--project=projeto1-370613 \
--resource-format=terraform
> para salvar na pasta, incluir o path (terraformgcp):
    - gcloud beta resource-config bulk-export \
    --path=terraformgcp \
    --project=projeto1-370613 \
    --resource-format=terraform

```

→ Conferir a estrutura de pastas e arquivos .tf de nossos projetos na GCP, exportado para dentro da pasta '*terraformgcp*'.

. Importando os estados

> Documentação - <https://cloud.google.com/docs/terraform/resource-management/import?hl=pt-br>

```

- gcloud beta resource-config terraform generate-import
terrafromexport
    > Vai gerar os arquivos de módulos e import .tf
    - terraform init
    - ./import_terraform.....

```

. Como fazer um apply para checar os Estados alterados

> alterar alguma coisa (nome do rótulo da VM) e fazer o terraform apply, para ver as mudanças aplicadas na VM no GCP.

. Gerenciando o Estado do Terraform com Google Cloud Storage

> Pegar o exemplo sugerido no github da Dio, criar o *backend.tf* com a config do bucket criado na GCP.

```

- terraform init
- terraform apply

```

> Pronto! Estado salvo em um bucket no Cloud Storage.

Cloud Deploy GCP

Cloud Deploy

• Introdução

> Skaffold

The screenshot shows the Google Cloud Pipeline interface for a project named 'barbero-devops-lab'. The pipeline is titled 'PIPELINE DE ENTREGA my-demo-app-1 (us-central1)'. The main view displays a flow diagram with two stages: 'qsdev' (green) and 'qspod' (red). An arrow points from 'qsdev' to 'qspod', labeled 'Promover' (Promote). Below the diagram, there are three sections: 'Implantações' (Deployments) with 2 in the last 30 days, 'Frequência de implantação' (Deployment frequency) with 'Nenhum lançamento' (No releases), and 'Taxa de falha na implantação' (Failure rate in deployment) at 100% over the last 30 days. A large button at the bottom right says 'CLIQUE AQUI PARA CONCLUIR ESSA AULA' (Click here to finish this lesson).

This is a screenshot of a course navigation menu titled 'Cloud Deploy'. It lists four lessons: 'Introdução', 'Skaffold e Deploy - Parte 1', 'Skaffold e Deploy - Parte 2', and 'Skaffold e Deploy - Conclusão'. At the bottom is a red button labeled 'CLIQUE AQUI PARA CONCLUIR ESSA AULA'.

This is a screenshot of a course navigation menu titled 'Jenkins'. It lists five lessons: 'Introdução' (highlighted with a green checkmark), 'Instalação Através do Marketplace', 'Configurando o Jenkins - Parte 1', 'Configurando o Jenkins - Parte 2', and 'Exemplo Prático com o Jenkins'. At the bottom is a red button labeled 'CLIQUE AQUI PARA CONCLUIR ESSA AULA'.

This is a screenshot of a course navigation menu titled 'Spinnaker'. It lists five lessons: 'O que é Spinnaker', 'Instalação GCP e Configuração - Parte 1', 'Instalação GCP e Configuração - Parte 2', 'Criando um Cluster no Contexto do Spinnaker', and 'Deploy Pipeline Staging e Prod'. At the bottom is a red button labeled 'CLIQUE AQUI PARA CONCLUIR ESSA AULA'.

Criando Pipeline de CI/CD com Cloud Build e Terraform

Projeto prático

> Desafio de implantação e compilação, configurando a trigger no Cloud Build com Terraform e salvando o estado no Cloud Storage

- . Editar o script no exemplo2 - link:

<https://github.com/digitalinnovationone/terraform-gcp/tree/main>

- . Submeter os prints de cada etapa de configuração do Pipeline