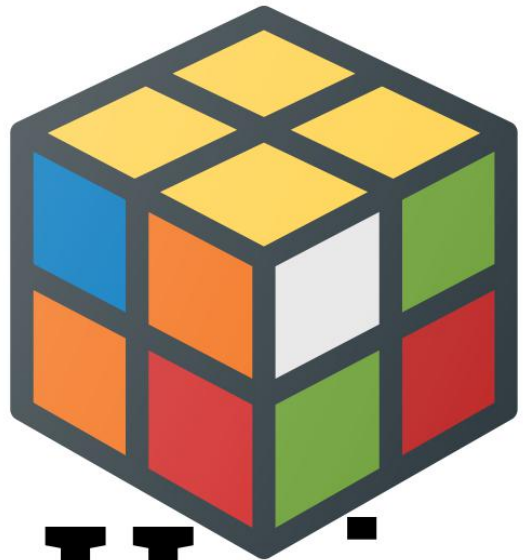


Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Atualizando e Desfazendo Deployments

Atualizando e Desfazendo Deployments

O desenvolvimento de qualquer aplicação é somente o passo inicial de um projeto, seja ele qual for.

O ciclo de vida de uma aplicação inclui atualizações (updates) e durante estas atualizações coisas ruins, muito ruins, podem acontecer sendo necessário desfazer estas atualizações (rollback).

Felizmente a ferramenta Deployment do Kubernetes nos fornece recursos para gerenciar isso de forma inteligente.

Para que possamos compreender como funciona estes recursos precisamos antes falar de **Versionamento e Rollout**

Atualizando e Desfazendo Deployments

Versionamento e Rollout

Quando realizamos o deploy (publicação) com o Deployment do Kubernetes é disparado um recurso chamado “rollout”.

Atualizando e Desfazendo Deployments

Versionamento e Rollout

A cada execução do“rollout” é criada uma nova revisão da publicação.

Revisão 1

Atualizando e Desfazendo Deployments

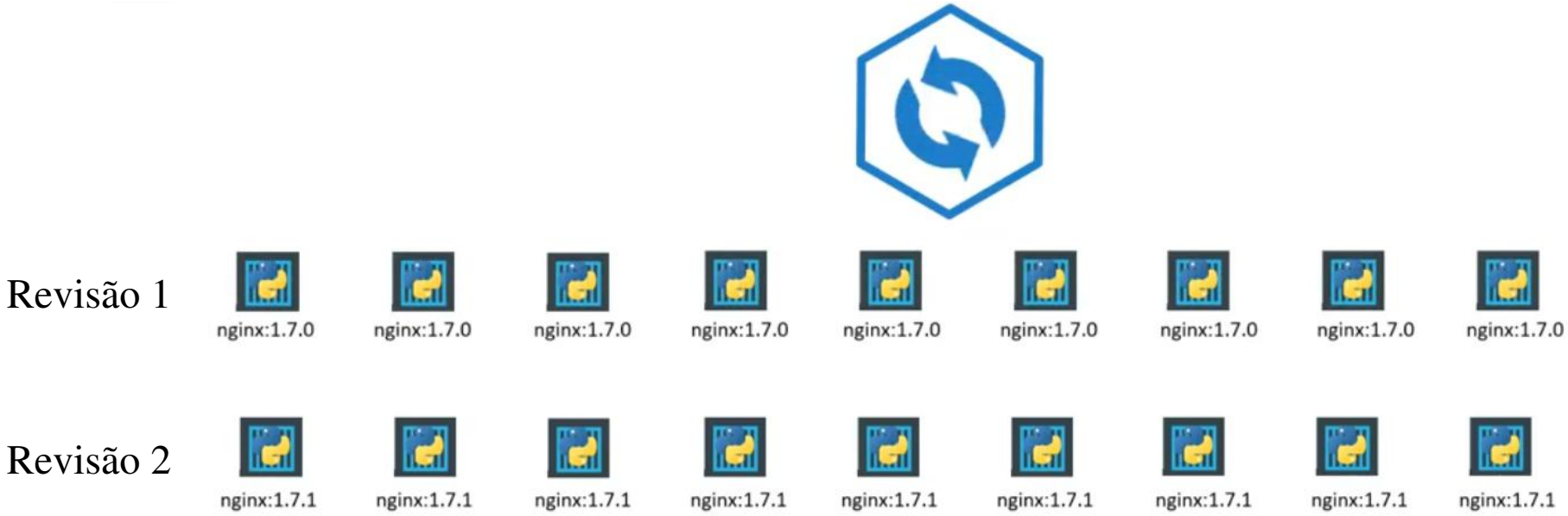
Versionamento e Rollout

A cada execução do“rollout” é criada uma nova revisão da publicação.

Atualizando e Desfazendo Deployments

Versionamento e Rollout

A cada execução do“rollout” é criada uma nova revisão da publicação.

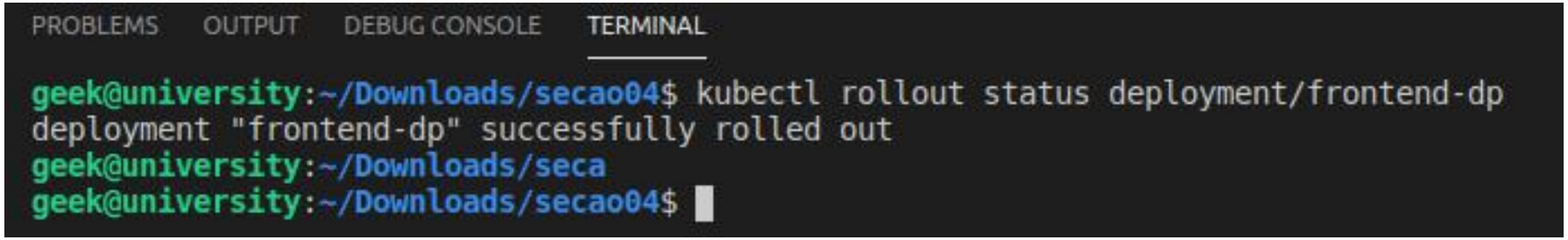


Este processo de revisão, criado pelo “rollout” ajuda o Kubernetes a manter o rastro de publicações da aplicação, fazendo com que facilmente possa ser realizado um “rollback” para qualquer das reviões anteriores em caso de problemas.

Atualizando e Desfazendo Deployments

Kubectl

Podemos checar o status do rollout a qualquer momento com o comando:

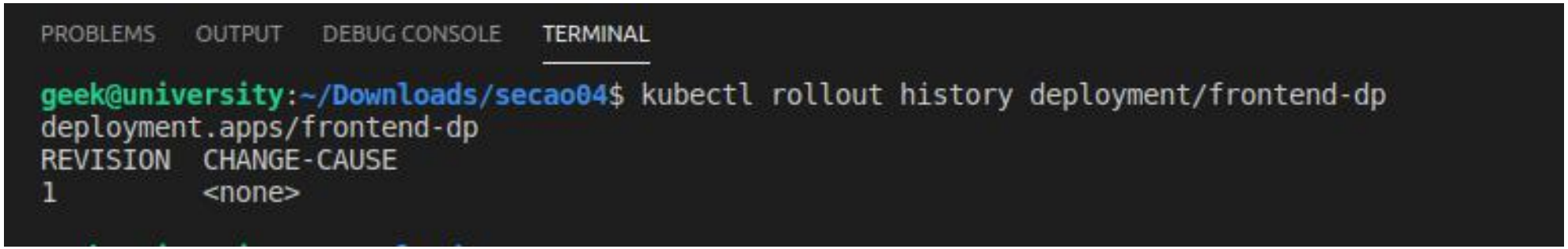


OBS: Note que o parâmetro informado para o deployment é deployment/nome-do-deployment pois podemos ter vários deployments no nosso cluster.

Atualizando e Desfazendo Deployments

Kubectl

Podemos consultar o histórico do rollout a qualquer momento com o comando:



OBS: Note que estamos na revisão 1 do rollout, pois apenas publicamos a aplicação sem termos realizado nenhuma nova publicação de atualização.

Atualizando e Desfazendo Deployments

Estratégias de (Deployment) Publicação

Atualmente existem 2 estratégias de deployment no Kubernetes.

```
geek@university:~/Downloads/secao04$ kubectl describe deployment frontend-dp
Name:                   frontend-dp
Namespace:              default
CreationTimestamp:      Sun, 08 Nov 2020 20:35:36 -0300
Labels:                 app=frontend-app
                        type=frontend
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               type=frontend
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=frontend-app
           type=frontend
  Containers:
   nginx-container:
    Image:        nginx
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Progressing    True    NewReplicaSetAvailable
  Available      True    MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   frontend-dp-79fdffd664 (3/3 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  15h   deployment-controller  Scaled up replica set frontend-dp-79fdffd664 to 3
```

Se consultarmos os detalhes do nosso deployment, veremos que a estratégia de deployment usada por padrão é a RollingUpdate.

Atualizando e Desfazendo Deployments

Estratégias de (Deployment) Publicação

Estratégia Recreate

Imagine que tenhamos uma aplicação publicada em 5 instâncias de pods para dividir a carga de trabalho.

Atualizando e Desfazendo Deployments

Estratégias de (Deployment) Publicação

Estratégia Recreate

Acabamos de finalizar o desenvolvimento de uma nova atualização e precisamos republicar a aplicação.

Atualizando e Desfazendo Deployments

Estratégias de (Deployment) Publicação

Estratégia Recreate

Na estratégia Recreate, primeiro damos um “shutdown” (desligamos) em todas as instâncias.

Atualizando e Desfazendo Deployments

Estratégias de (Deployment) Publicação

Estratégia Recreate

E então realizamos a publicação em novas instâncias.

Atualizando e Desfazendo Deployments

Estratégias de (Deployment) Publicação

Estratégia Recreate

O problema desta abordagem é que durante este tempo seus clientes/usuários ficarão sem acesso à aplicação.

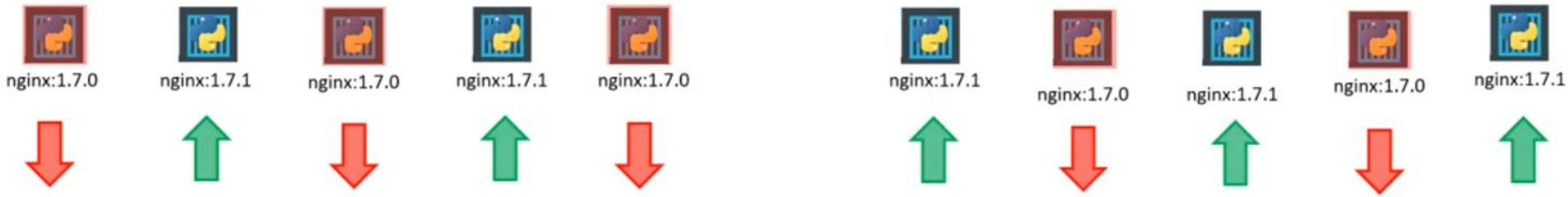
OBS: Felizmente já vimos que não é esta a estratégia padrão usada pelo Kubernetes. ;)

Atualizando e Desfazendo Deployments

Estratégias de (Deployment) Publicação

Estratégia Rolling Update

Esta estratégia atualiza uma por uma (de cada vez) as instâncias de pods do nosso cluster, fazendo com que a aplicação fique sempre disponível, mantendo desta forma a alta disponibilidade.



OBS: Felizmente já vimos que esta é a estratégia padrão usada pelo Kubernetes. ;)

```
Labels:                 app=frontend-app
                        type=frontend
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               type=frontend
Replicas:               3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=frontend-app
           type=frontend
```

Atualizando e Desfazendo Deployments

Como ocorrem as atualizações?

OBS: Uma atualização não é apenas atualização da sua aplicação, mas pode ser de versão do Docker, versão do Kubernetes, configurações do cluster, do replicaset, labels, número de réplicas, ou seja, qualquer mudança ocorrida após a publicação original.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend-dp
  labels:
    app: frontend-app
    type: frontend
spec:
  template:
    metadata:
      name: frontend-pod
      labels:
        app: frontend-app
        type: frontend
    spec:
      containers:
        - name: nginx-container
          image: nginx:1.19.4
  selector:
    matchLabels:
      type: frontend
  replicas: 3
```

Veja neste exemplo que fiz uma mudança apenas na versão do nginx a ser usada como imagem.

Atualizando e Desfazendo Deployments

Kubectl

Então podemos aplicar a atualização com o comando:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

geek@university:~/Downloads/secao04$ kubectl apply -f deployments/dp.yaml
Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply
deployment.apps/frontend-dp configured
geek@university:~/Downloads/secao04$
```

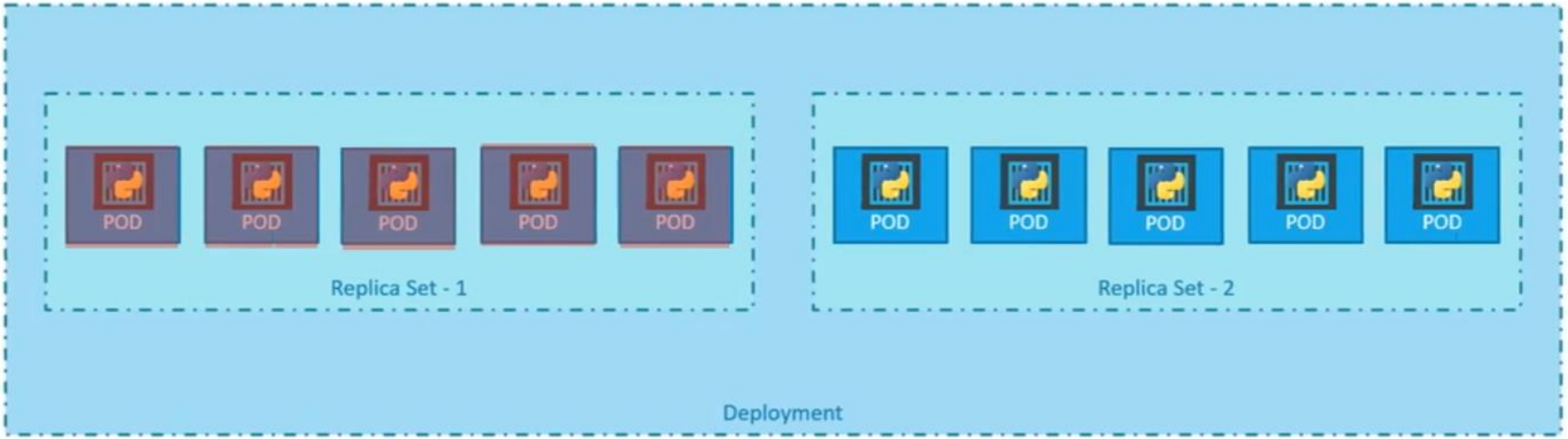
OBS: Note o alerta que o Kubernetes está dando para que nas próximas vezes, sempre que usarmos o comando “create” para criar qualquer objeto, devemos informar o parâmetro --save-config

Exemplo:

kubectl create -f pods/meu-pod.yaml --save-config

Atualizando e Desfazendo Deployments

Como o Kubernetes gerencia a atualização por baixo dos panos?



A ferramenta Deployment criar um novo ReplicaSet idêntico ao original e vai fazendo a atualização, um por um, se a estratégia for Rolling Update, ou todos de uma vez se a estratégia for Recreate.

Atualizando e Desfazendo Deployments

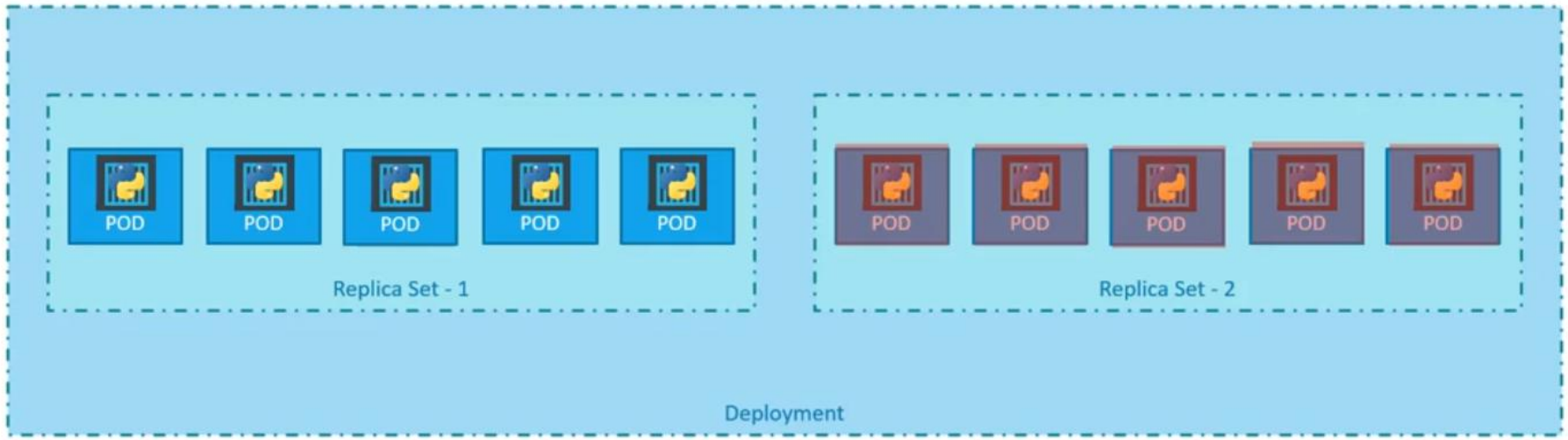
E se a atualização apresentar problemas? Rollback!

Lembra-se que a cada deploy (publicação) realizado é criada uma nova versão de revisão pelo Rollout?

Podemos desfazer a atualização realizada com o comando:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

geek@university:~/Downloads/secao04$ kubectl rollout undo deployment/frontend-dp
deployment.apps/frontend-dp rolled back
```



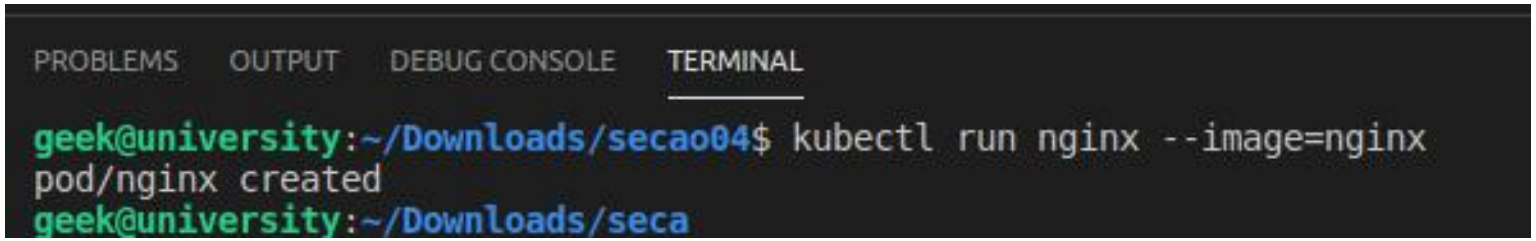
O rollback irá destruir as instâncias com a atualização e criar novamente instâncias com a versão anterior.

OBS: Tudo isso de acordo com a estratégia de atualização adotada.

Atualizando e Desfazendo Deployments

Kubectl

Dica: Uma forma de publicação mais direta usando o kubectl é através do comando:



Este comando não apenas cria um pod mas também um ReplicaSet.

Esta é uma forma rápida e direta mas não é a recomendada, pois o ideal é termos os arquivos com definição para que tenhamos controle do que e como está sendo criado.

Atualizando e Desfazendo Deployments

Comandos úteis:

kubectl create -f deployment.yaml --save-config

kubectl get deployments

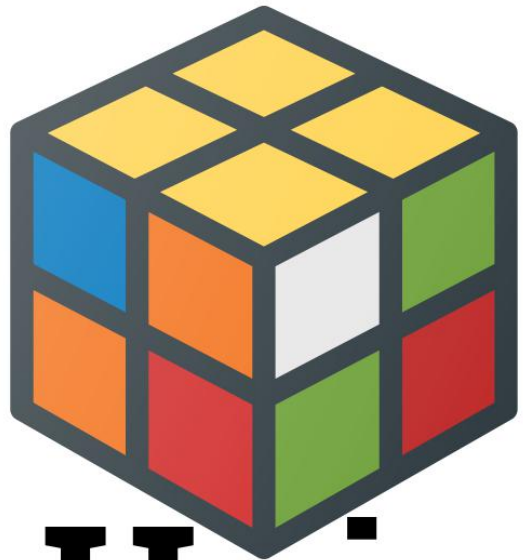
kubectl apply -f deployment.yaml

kubectl rollout status deployment/nome-do-deployment

kubectl rollout history deployment/nome-do-reployment

kubectl rollout undo deployment/nome-do-deployment

OBS: Na próxima aula iremos fazer o exercício prático disso tudo!



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br