ESCOLA SUPERIOR DE REDES

Orquestração de Containers

Gestão do ciclo de vida de aplicações

Tópicos abordados

- Rollout e rollbacks (rollout undo)
- Alterando o CMD e ENTRYPOINT de containers
- Exportando variáveis de ambiente
- ConfigMaps
- Secrets
- Init-Containers
- HPA - Horizontal POD Autoscaler

Revisando *Deployments*

Como vimos, *Deployments* possibilitam uma forma ***declarativa de atualização para Pods e ReplicaSets***

Deployments são o construto padrão a ser utilizado ao ***disponibilizar aplicações***

Porque oferece o recurso de ***rollout*** e ***rollback***

Visualizando o *status* de *rollouts*

vagrant@s2-master-1: ~ | fbs@DESKTOP-A1M9PI8: ~

```
root@s2-master-1:~# kubectl apply -f deploy-nginx.yaml ; while true ; do kubectl rollout status de
ployment/deploy-nginx ; sleep 3 ; done
deployment.apps/deploy-nginx created
Waiting for deployment "deploy-nginx" rollout to finish: 0 of 4 updated replicas are available...
Waiting for deployment "deploy-nginx" rollout to finish: 1 of 4 updated replicas are available...
Waiting for deployment "deploy-nginx" rollout to finish: 2 of 4 updated replicas are available...
Waiting for deployment "deploy-nginx" rollout to finish: 3 of 4 updated replicas are available...
deployment "deploy-nginx" successfully rolled out
deployment "deploy-nginx" successfully rolled out
^C
root@s2-master-1:~#
```

Atualizando a imagem de *deployments*

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-nginx
spec:
  replicas: 4
  selector:
    matchLabels:
      app: deploy-nginx
  template:
    metadata:
      labels:
        app: deploy-nginx
    spec:
      containers:
      - name: deploy-nginx
        image: nginx:latest
```

Registrando o histórico com a flag --record

```
root@s2-master-1:~# k apply -f manifests/deploy-nginx.yaml --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/deploy-nginx configured
```

Usando o comando set image

```
root@s2-master-1:~# k set image deploy/deploy-nginx deploy-nginx=nginx:1.15.1 --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/deploy-nginx image updated
```

Visualizando o histórico de *deployments*

```
root@s2-master-1:~# k rollout history deployment deploy-nginx
deployment.apps/deploy-nginx
REVISION  CHANGE-CAUSE
1         kubectl apply --filename=manifests/deploy-nginx.yaml --record=true
2         kubectl set image deploy/deploy-nginx deploy-nginx=nginx:1.15.1 --record=true
```

Visualizando os detalhes de cada revisão

```
root@s2-master-1:~# k rollout history deployment deploy-nginx --revision=1
deployment.apps/deploy-nginx with revision #1
Pod Template:
  Labels:       app=deploy-nginx
        pod-template-hash=5c646dc86b
  Annotations:  kubernetes.io/change-cause: kubectl apply --filename=manifests/deploy-nginx.yaml --record=true
  Containers:
   deploy-nginx:
    Image:      nginx:latest
    Port:       <none>
    Host Port:  <none>
    Environment:        <none>
    Mounts:     <none>
  Volumes:      <none>

root@s2-master-1:~# k rollout history deployment deploy-nginx --revision=2
deployment.apps/deploy-nginx with revision #2
Pod Template:
  Labels:       app=deploy-nginx
        pod-template-hash=5bb9649cdc
  Annotations:  kubernetes.io/change-cause: kubectl set image deploy/deploy-nginx deploy-nginx=nginx:1.15.1 --record=true
  Containers:
   deploy-nginx:
    Image:      nginx:1.15.1
    Port:       <none>
    Host Port:  <none>
    Environment:        <none>
    Mounts:     <none>
  Volumes:      <none>
```

Realizando o *rollback* para um versão anterior

revison = 3

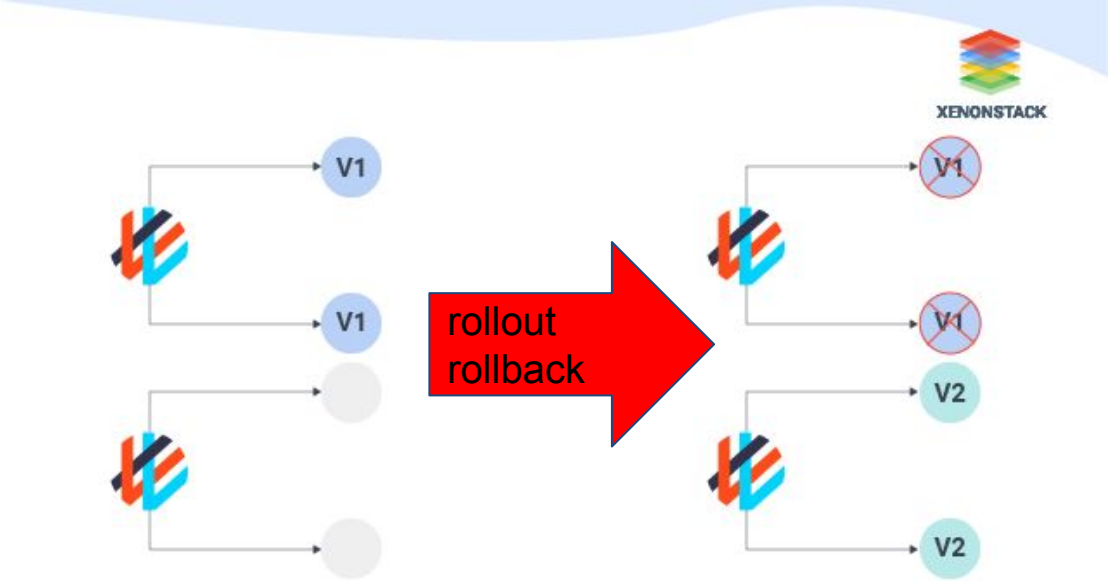
```
root@s2-master-1:~# k apply -f manifests/deploy-nginx.yaml --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/deploy-nginx configured
root@s2-master-1:~# k rollout history deployment deploy-nginx
deployment.apps/deploy-nginx
REVISION  CHANGE-CAUSE
1         kubectl apply --filename=manifests/deploy-nginx.yaml --record=true
2         kubectl set image deploy/deploy-nginx deploy-nginx=nginx:1.15.1 --record=true
3         kubectl apply --filename=manifests/deploy-nginx.yaml --record=true

root@s2-master-1:~# k set image deploy/deploy-nginx deploy-nginx=nginx:1.17.1 --record
Flag --record has been deprecated, --record will be removed in the future
deployment.apps/deploy-nginx image updated
root@s2-master-1:~# k rollout history deployment deploy-nginx
deployment.apps/deploy-nginx
REVISION  CHANGE-CAUSE
1         kubectl apply --filename=manifests/deploy-nginx.yaml --record=true
2         kubectl set image deploy/deploy-nginx deploy-nginx=nginx:1.15.1 --record=true
3         kubectl apply --filename=manifests/deploy-nginx.yaml --record=true
4         kubectl set image deploy/deploy-nginx deploy-nginx=nginx:1.17.1 --record=true

root@s2-master-1:~# k describe deployment deploy-nginx | grep Image
    Image:        nginx:1.17.1
root@s2-master-1:~# k rollout undo deployment deploy-nginx
deployment.apps/deploy-nginx rolled back
root@s2-master-1:~# k describe deployment deploy-nginx | grep Image
    Image:        nginx:1.16.1
root@s2-master-1:~# k rollout history deployment deploy-nginx
deployment.apps/deploy-nginx
REVISION  CHANGE-CAUSE
1         kubectl apply --filename=manifests/deploy-nginx.yaml --record=true
2         kubectl set image deploy/deploy-nginx deploy-nginx=nginx:1.15.1 --record=true
4         kubectl set image deploy/deploy-nginx deploy-nginx=nginx:1.17.1 --record=true
5         kubectl apply --filename=manifests/deploy-nginx.yaml --record=true
```

Estratégias de *rollout*

Recreate

RollingUpdate

Recreate

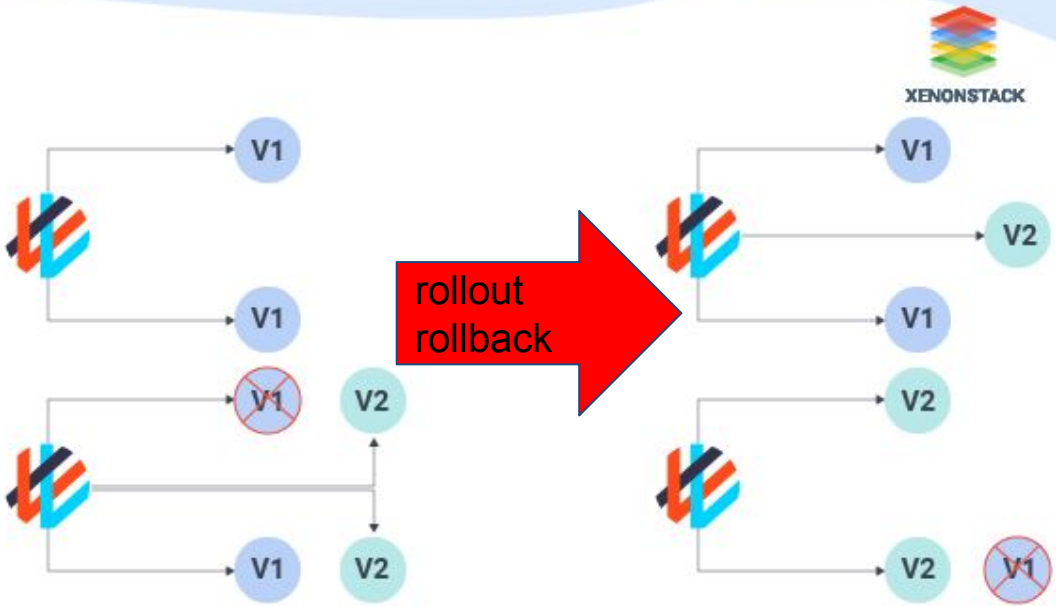


Fácil configuração

Estado renovado ao final do processo

Alto impacto ao usuário, ocasionando ***downtime***

RollingUpdate - PADRÃO



Atualização sem ***downtime***

Mantém duas versões da app ao mesmo tempo

Mais demorada

Configuração do *RollingUpdate*

MaxSurge

Máximo número de *pods* que podem ser criados acima do valor desejado

MaxUnavailable

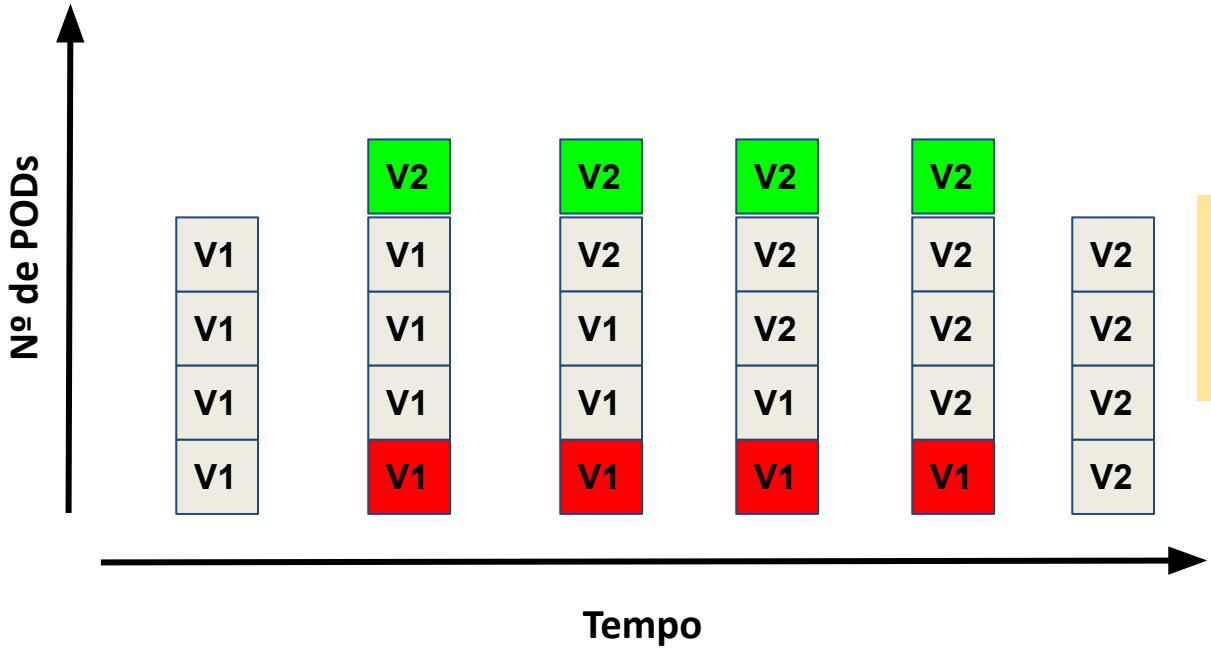
Máximo número de *pods* indisponíveis durante o *rollout*

Configuração do *RollingUpdate*

PODs = 4

MaxSurge = 25% -> 1

MaxUnavaiable = 25% -> 1



No máximo 5 PODs (4 desired + 1 surge)

No mínimo 3 PODs (4 desired - 1 unavailable)

Outras estratégias de rolling deployment

https://www.weave.works/blog/kubernetes-deployment-strategies

Configuração do *RollingUpdate*

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-nginx-rollingupdate
spec:
  replicas: 4
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
  selector:
    matchLabels:
      app: deploy-nginx-rollingupdate
  template:
    metadata:
      labels:
        app: deploy-nginx-rollingupdate
    spec:
      containers:
      - name: deploy-nginx
        image: nginx:latest
```

Configuração do *RollingUpdate*

```
root@s2-master-1:~# k apply -f manifests/deploy-nginx-rollingupdate.yaml ; while true ; do k rollout status deployme
nt deploy-nginx-rollingupdate ; sleep 1 ; done
deployment.apps/deploy-nginx-rollingupdate configured
Waiting for deployment spec update to be observed...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 0 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 0 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 1 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 2 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 2 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 2 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 2 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 2 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 3 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 3 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 3 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 3 out of 4 new replicas have been updated...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "deploy-nginx-rollingupdate" rollout to finish: 3 of 4 updated replicas are available...
deployment "deploy-nginx-rollingupdate" successfully rolled out
```

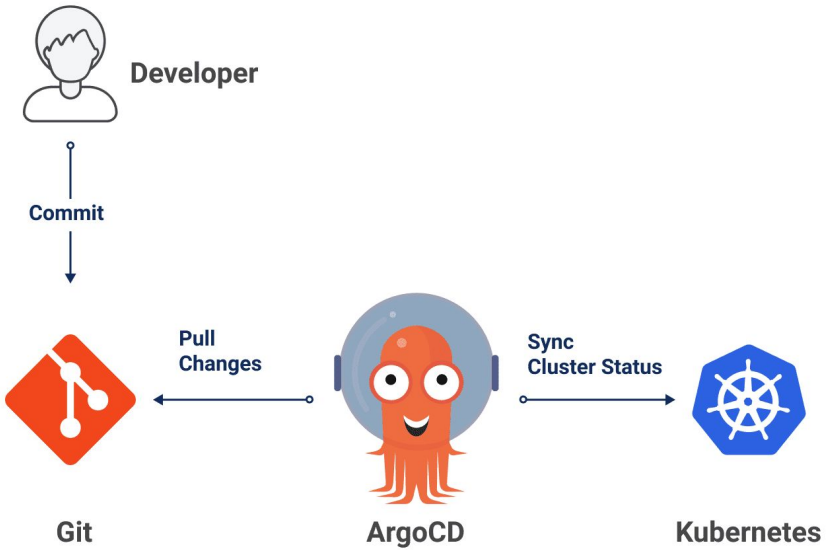
O que acharam do rollout/rollback do k8s?

Quanto tempo demoraria um rollout/rollback sem k8s mas com container?

Quanto tempo demoraria um rollout/rollback sem k8s e sem container?

Acharam que isso é tudo?

Acharam que isso é tudo no k8s?



GitOps
git repo é a fonte da verdade

https://argo-cd.readthedocs.io

Alterando o CMD de containers

```
root@s2-master-1:~# k run alpine-1 --image=alpine
pod/alpine-1 created
root@s2-master-1:~# k get pod
NAME       READY   STATUS      RESTARTS   AGE
alpine-1   0/1     Completed   0          8s
root@s2-master-1:~# k get pod
NAME       READY   STATUS             RESTARTS      AGE
alpine-1   0/1     CrashLoopBackOff   1 (7s ago)    15s
```

```
root@s2-master-1:~# k run alpine-2 --image=alpine -- sleep 360
pod/alpine-2 created
root@s2-master-1:~# k get pod
NAME       READY   STATUS      RESTARTS      AGE
alpine-1   0/1     Completed   4 (49s ago)   103s
alpine-2   1/1     Running     0             11s
root@s2-master-1:~# k exec alpine-2 -- ps aux
PID   USER     TIME  COMMAND
    1 root      0:00 sleep 360
    7 root      0:00 ps aux
```

```
FROM scratch
ADD alpine-minirootfs-3.18.2-x86_64.tar.gz /
CMD ["/bin/sh"]
```

altera o CMD do container

Alterando o ENTRYPOINT de containers

```
root@s2-master-1:~# k run nginx --image=nginx:alpine
pod/nginx created
root@s2-master-1:~# k get pod
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          8s
root@s2-master-1:~# k exec nginx -- ps aux
PID   USER     TIME  COMMAND
    1 root      0:00 nginx: master process nginx -g daemon off;
   30 nginx     0:00 nginx: worker process
   31 root      0:00 ps aux
```

```
ENTRYPOINT ["/docker-entrypoint.sh"]

EXPOSE 80

STOPSIGNAL SIGQUIT

CMD ["nginx", "-g", "daemon off;"]
```

```
root@s2-master-1:~# k run nginx-2 --image=nginx:alpine --command -- sleep 360
pod/nginx-2 created
root@s2-master-1:~# k get pod
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   0          2m46s
nginx-2   1/1     Running   0          10s
root@s2-master-1:~# k exec nginx-2 -- ps aux
PID   USER     TIME  COMMAND
    1 root      0:00 sleep 360
    7 root      0:00 ps aux
```

altera o ENTRYPOINT do container

Via kubectl - forma imperativa

```
root@s2-master-1:~# k run app-color --image=fbscarel/myapp-color -l app=app-color --port=80 --env="COLOR=red"
pod/app-color created
root@s2-master-1:~# k get pod
NAME        READY   STATUS              RESTARTS   AGE
app-color   0/1     ContainerCreating   0          4s
root@s2-master-1:~# k create service nodeport app-color --tcp=80 --node-port=31000
service/app-color created
root@s2-master-1:~# k get svc
NAME        TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
app-color   NodePort   10.111.106.38   <none>        80:31000/TCP   5s
root@s2-master-1:~# curl localhost:31000/color; echo
Hostname: app-color ; Color: red
root@s2-master-1:~# k delete pod app-color --force
Warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resour
ce may continue to run on the cluster indefinitely.
pod "app-color" force deleted
root@s2-master-1:~# k run app-color --image=fbscarel/myapp-color -l app=app-color --port=80 --env="COLOR=blue"
pod/app-color created
root@s2-master-1:~# curl localhost:31000/color; echo
Hostname: app-color ; Color: blue
```

Via YAML

```
root@s2-master-1:~# k run app-color --image=fbscarel/myapp-color -l app=app-color --port=80 --env="COLOR=blue" --dry-run=client -o yaml > manifests/app-color.yaml
root@s2-master-1:~# cat manifests/app-color.yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    app: app-color
  name: app-color
spec:
  containers:
  - env:
    - name: COLOR
      value: blue
    image: fbscarel/myapp-color
    name: app-color
    ports:
    - containerPort: 80
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
```

Variáveis de ambiente no kubernetes

https://kubernetes.io/docs/tasks/inject-data-application/define-environment-variable-container/

ConfigMaps

Objeto utilizado para armazenar dados não-confidenciais em pares ***chave-valor***

Podem ser consumidos como variáveis de ambiente, argumentos CLI ou arquivos montados em ***volumes***

Permitem desacoplar a configuração da imagem do container, facilitando ***portabilidade***

Para dados sensíveis, deve-se utilizar ***Secrets***

Criando *ConfigMaps* via YAML

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-color-cm
data:
  COLOR: pink
```

Criando *ConfigMaps* via imperativa

```
root@s2-master-1:~# k create cm app-color-cm --from-literal="COLOR=pink"
configmap/app-color-cm created
root@s2-master-1:~# k describe cm app-color-cm
Name:         app-color-cm
Namespace:    lab
Labels:       <none>
Annotations:  <none>

Data
====
COLOR:
----
pink

BinaryData
====

Events:  <none>
```

Utilizando *ConfigMaps* na configuração de *Pods*

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: app-color
  name: app-color
spec:
  containers:
  - env:
    - name: COLOR
      valueFrom:
        configMapKeyRef:
          name: app-color-cm
          key: COLOR
    image: fbscarel/myapp-color
    name: app-color
    ports:
    - containerPort: 80
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: app-color-envfrom
  name: app-color-envfrom
spec:
  containers:
  - name: app-color
    image: fbscarel/myapp-color
    ports:
    - containerPort: 80
    envFrom:
    - configMapRef:
        name: app-color-cm
```

Utilizando *ConfigMaps* na configuração de *Pods*

```
root@s2-master-1:~# k get pod
NAME                  READY   STATUS    RESTARTS   AGE
app-color-envfrom     1/1     Running   0          2m51s
app-color-valuefrom   1/1     Running   0          12m
root@s2-master-1:~# k get cm
NAME               DATA   AGE
app-color-cm       1      69m
kube-root-ca.crt   1      10d
root@s2-master-1:~# k describe cm app-color-cm
Name:         app-color-cm
Namespace:    lab
Labels:       <none>
Annotations:  <none>

Data
====
COLOR:
----
pink

BinaryData
====

Events:  <none>
```

Utilizando *ConfigMaps* na configuração de *Pods*

```
root@s2-master-1:~# k create service nodeport app-color-envfrom --tcp=80 --node-port=31000
service/app-color-envfrom created
root@s2-master-1:~# k create service nodeport app-color-valuefrom --tcp=80 --node-port=32000
service/app-color-valuefrom created
root@s2-master-1:~# k get svc
NAME                  TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
app-color-envfrom     NodePort    10.111.88.157   <none>        80:31000/TCP   18s
app-color-valuefrom   NodePort    10.111.3.238    <none>        80:32000/TCP   5s
root@s2-master-1:~# curl localhost:31000/color; echo
Hostname: app-color-envfrom ; Color: pink
root@s2-master-1:~#
root@s2-master-1:~# curl localhost:32000/color; echo
Hostname: app-color-valuefrom ; Color: pink
```

Secrets

Objetos utilizados para armazenar informações ***sensíveis***

P.ex.: senhas, *tokens* Oauth ou chaves SSH

Armazenados em codificação ***base64*** (i.e. não cifrados)

Segurança baseada na visibilidade dos elementos (opcionalmente, **criptografia**/**RBAC**)

Tipos de *Secrets*

Builtin Type	Usage
`Opaque`	arbitrary user-defined data
`kubernetes.io/service-account-token`	service account token
`kubernetes.io/dockercfg`	serialized `~/.dockercfg` file
`kubernetes.io/dockerconfigjson`	serialized `~/.docker/config.json` file
`kubernetes.io/basic-auth`	credentials for basic authentication
`kubernetes.io/ssh-auth`	credentials for SSH authentication
`kubernetes.io/tls`	data for a TLS client or server
`bootstrap.kubernetes.io/token`	bootstrap token data

Criando *Secrets* via YAML

```
root@s2-master-1:~# k apply -f manifests/daytona-secret.yaml
secret/daytona-secret-yaml created
root@s2-master-1:~# k describe secrets daytona-secret-yaml
Name:         daytona-secret-yaml
Namespace:    lab
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
DBPASS:  5 bytes
DBUSER:  4 bytes
```

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: daytona-secret-yaml
data:
  DBPASS: YmVhY2g=
  DBUSER: cm9vdA==
```

Criando *Secrets* via imperativa

```
root@s2-master-1:~# k create secret generic secret-teste --from-literal="user=fulano" --from-literal="password=senha"
secret/secret-teste created
root@s2-master-1:~# k get secrets
NAME                  TYPE     DATA   AGE
daytona-secret-yaml   Opaque   2      3m30s
secret-teste          Opaque   2      7s
```

Visualizando *Secrets*

```
root@s2-master-1:~# k get secrets secret-teste -o json
{
    "apiVersion": "v1",
    "data": {
        "password": "c2VuaGE=",
        "user": "ZnVsYW5v"
    },
    "kind": "Secret",
    "metadata": {
        "creationTimestamp": "2023-08-04T08:15:52Z",
        "name": "secret-teste",
        "namespace": "lab",
        "resourceVersion": "297408",
        "uid": "f7518b03-70f8-4239-9cf2-8486af16fd87"
    },
    "type": "Opaque"
}
root@s2-master-1:~# echo c2VuaGE= | base64 -d; echo
senha
root@s2-master-1:~# echo ZnVsYW5v | base64 -d; echo
fulano
```

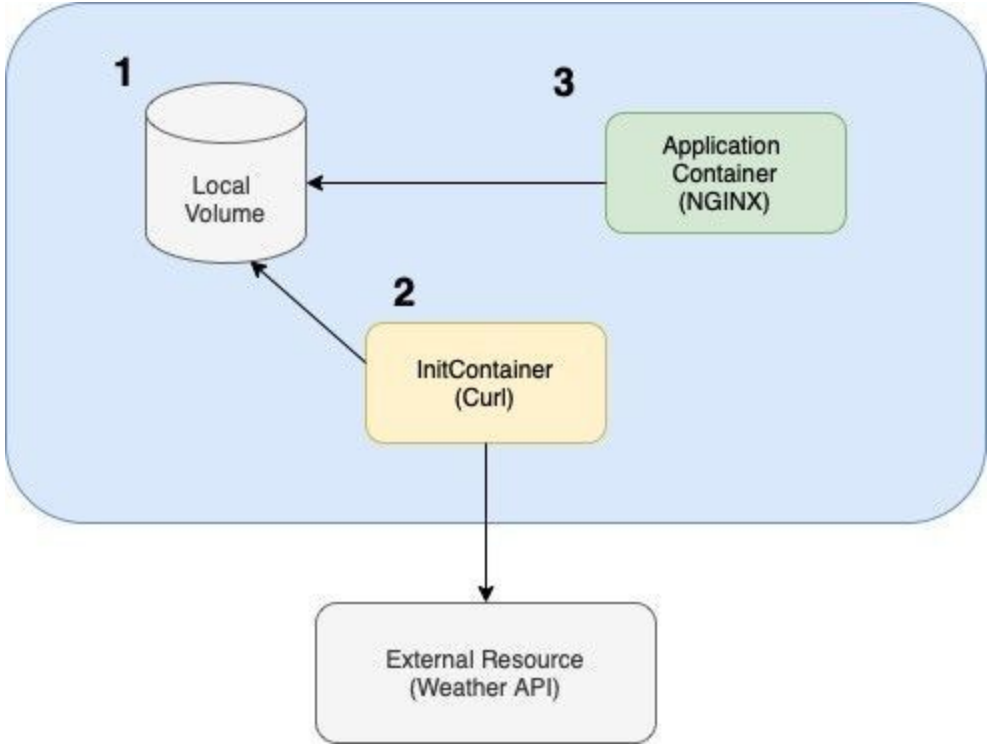
Visualizando *Secrets*

```
root@s2-master-1:~# k get secrets secret-teste -o json | jq '.data | map_values(@base64d)'
{
  "password": "senha",
  "user": "fulano"
}
```

Utilizando *Secrets* na configuração de *Pods*

```yaml
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: daytona-app
  name: daytona-app
spec:
  containers:
  - image: fbscarel/myapp-mysql
    name: daytona-app
    env:
      - name: DBHOST
        valueFrom:
          configMapKeyRef:
            name: daytona-cm
            key: DBHOST
    envFrom:
    - secretRef:
        name: daytona-secret-yaml
```

Init-container



Init Containers

Containers especializados que executam antes de containers de aplicação em um *pod*

Executados antes dos containers “normais”, até sua conclusão

Em havendo mais de um *init container*, executam sequencialmente

O próximo *init container* é iniciado apenas se o anterior obteve suceso

Init Containers: por quê usar?

Podem conter utilitários ou código especializado para configuração do ambiente

Podem fazer a função de *builder* e/ou *deployer*

Podem ter uma visão diferenciada do *filesystem*, acessando *secrets* invisíveis aos outros containers

Permitem a introdução de bloqueio/*delay* para checagem de pré-condições

Init Containers: exemplo

vagrant@s2-master-1: ~ | fbs@DESKTOP-A1M9PI8: ~

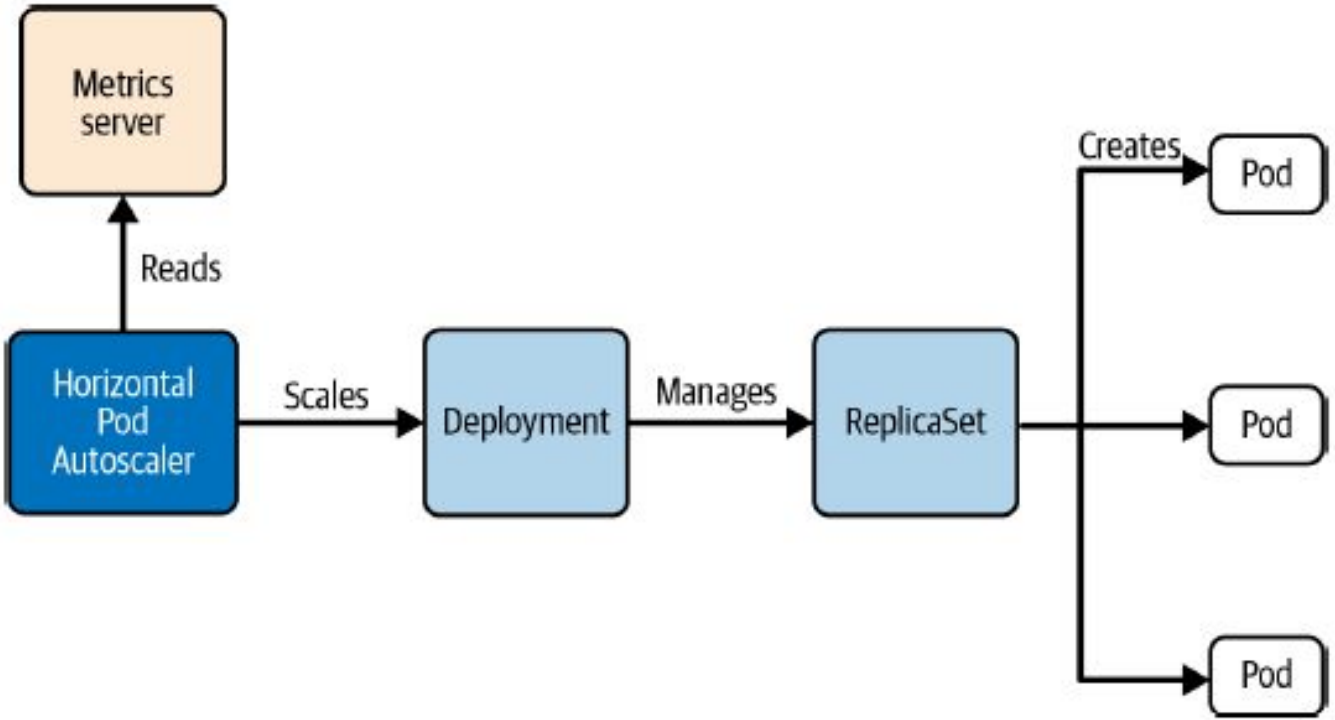
```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  labels:
    app: myapp
spec:
  containers:
  - name: myapp-container
    image: busybox:1.28
    command: ['sh', '-c', 'echo The app is running! && sleep 3600']
  initContainers:
  - name: init-myservice
    image: busybox:1.28
    command: ['sh', '-c', "until nslookup myservice.$(cat /var/run/secrets/kubernetes.io/serviceac
count/namespace).svc.cluster.local; do echo waiting for myservice; sleep 2; done"]
  - name: init-mydb
    image: busybox:1.28
    command: ['sh', '-c', "until nslookup mydb.$(cat /var/run/secrets/kubernetes.io/serviceaccount
/namespace).svc.cluster.local; do echo waiting for mydb; sleep 2; done"]
~
~
                                                                    19,0-1        All
```

Horizontal Pod Autoscaler

Escala automaticamente o número de *pods* em um ***deployment, replicaset ou statefulset***

Observa métricas de uso de CPU ou outras para determinar demanda

HPA - Horizontal POD Autoscaler



Exemplo:

```yaml
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: php-hpa
spec:
  maxReplicas: 5
  minReplicas: 1
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: php-hpa
  targetCPUUtilizationPercentage: 50
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: php-hpa
spec:
  selector:
    matchLabels:
      app: php-hpa
  replicas: 1
  template:
    metadata:
      labels:
        app: php-hpa
    spec:
      containers:
      - name: php-hpa
        image: fbscarel/php-sqrt
        resources:
          limits:
            cpu: 200m
          requests:
            cpu: 100m
```

Exemplo:

```
root@s2-master-1:~# k apply -f manifests/php-deploy.yaml
deployment.apps/php-hpa created
root@s2-master-1:~# k autoscale deploy php-hpa --cpu-percent=50 --min=1 --max=5 --dry-run=client -o yaml > manifests/php-hpa.y
aml
root@s2-master-1:~# k apply -f manifests/php-hpa.yaml
horizontalpodautoscaler.autoscaling/php-hpa created
root@s2-master-1:~# k get hpa
NAME      REFERENCE            TARGETS         MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   <unknown>/50%   1         5         0          13s
root@s2-master-1:~# k get deployments.apps php-hpa
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
php-hpa   1/1     1            1           4m12s
root@s2-master-1:~#
root@s2-master-1:~# k run load-generator --image=busybox -- /bin/sh -c 'while true; do wget -q -O- http://php-hpa; done'
pod/load-generator created
```

Exemplo:

```
root@s2-master-1:~# k get hpa
NAME      REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa    1%/50%    1         5         1          3m36s
root@s2-master-1:~#
root@s2-master-1:~# k get hpa
NAME      REFERENCE             TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa    187%/50%   1         5         1          3m51s
root@s2-master-1:~#
root@s2-master-1:~# k get hpa
NAME      REFERENCE             TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa    119%/50%   1         5         4          4m7s
root@s2-master-1:~#
root@s2-master-1:~# k get hpa
NAME      REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa    57%/50%   1         5         4          4m23s
root@s2-master-1:~#
root@s2-master-1:~# k get hpa
NAME      REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa    42%/50%   1         5         5          4m46s
root@s2-master-1:~#
root@s2-master-1:~# k get hpa
NAME      REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa    45%/50%   1         5         5          5m7s
root@s2-master-1:~#
root@s2-master-1:~# k get hpa
NAME      REFERENCE             TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa    45%/50%   1         5         5          5m34s
```

Exemplo:

```
root@s2-master-1:~# k delete pod load-generator
pod "load-generator" deleted
^Croot@s2-master-1:~k get pod
NAME                     READY   STATUS        RESTARTS   AGE
load-generator           1/1     Terminating   0          2m50s
php-hpa-b89f98cf7-fxnnq  1/1     Running       0          2m21s
php-hpa-b89f98cf7-nsrj7  1/1     Running       0          25m
php-hpa-b89f98cf7-ntzzq  1/1     Running       0          2m21s
php-hpa-b89f98cf7-sxr2d  1/1     Running       0          2m21s
php-hpa-b89f98cf7-z7nsb  1/1     Running       0          96s
root@s2-master-1:~# k get pod
NAME                     READY   STATUS    RESTARTS   AGE
php-hpa-b89f98cf7-fxnnq  1/1     Running   0          2m36s
php-hpa-b89f98cf7-nsrj7  1/1     Running   0          25m
php-hpa-b89f98cf7-ntzzq  1/1     Running   0          2m36s
php-hpa-b89f98cf7-sxr2d  1/1     Running   0          2m36s
php-hpa-b89f98cf7-z7nsb  1/1     Running   0          111s
```

Exemplo:

```
root@s2-master-1:~# while true; do k get hpa; echo '----'; sleep 20; done
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   18%/50%   1         5         5          21m
----
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         5          22m
----
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         5          22m
----
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         5          22m
----
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         5          23m
```

```
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         5          24m
----
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         5          24m
----
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         5          24m
----
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         5          25m
```

```
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         2          26m
----
NAME      REFERENCE            TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
php-hpa   Deployment/php-hpa   1%/50%    1         5         1          27m
```

Tarefa 5

As atividades práticas desta sessão podem ser obtidas em formato HTML via:

https://bit.ly/ads19-tarefas-s5

ESCOLA SUPERIOR DE REDES

Gestão do ciclo de vida de aplicações

RNP ORGANIZAÇÃO SOCIAL DO MCTI

MINISTÉRIO DO TURISMO

MINISTÉRIO DA DEFESA

MINISTÉRIO DA SAÚDE

MINISTÉRIO DA EDUCAÇÃO

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES

PÁTRIA AMADA BRASIL GOVERNO FEDERAL