WTR

WORKSHOP
DE TECNOLOGIA DE REDES DO POP-RS
2021

Oficina de introdução ao Kubernetes na prática

21 SET

César Loureiro

Início: 14 h

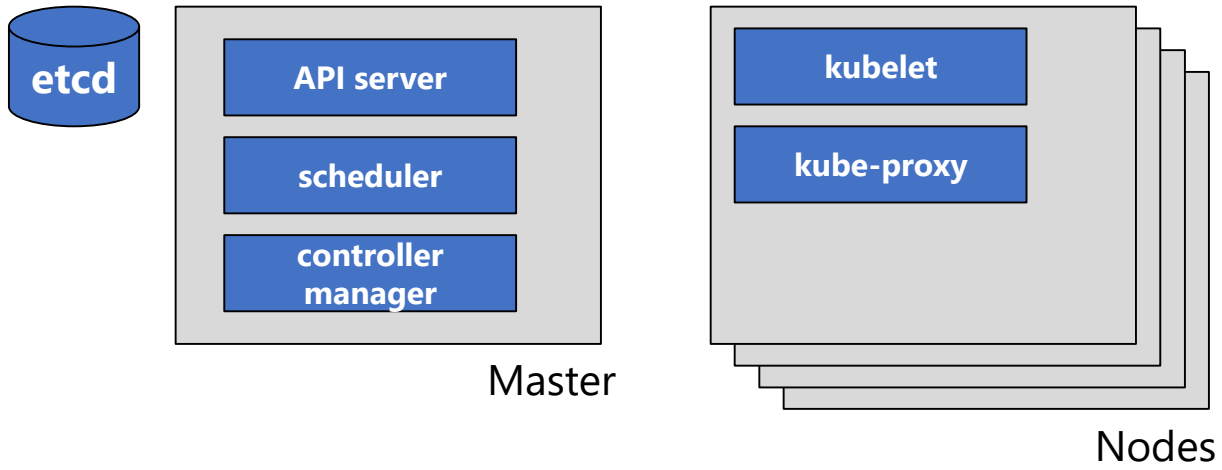
PoP-RS
Ponto de Presença da RNP no Rio Grande do Sul

RNP
ORGANIZAÇÃO SOCIAL DO MCTI

Kubernetes é um gerenciador de containers *open source*

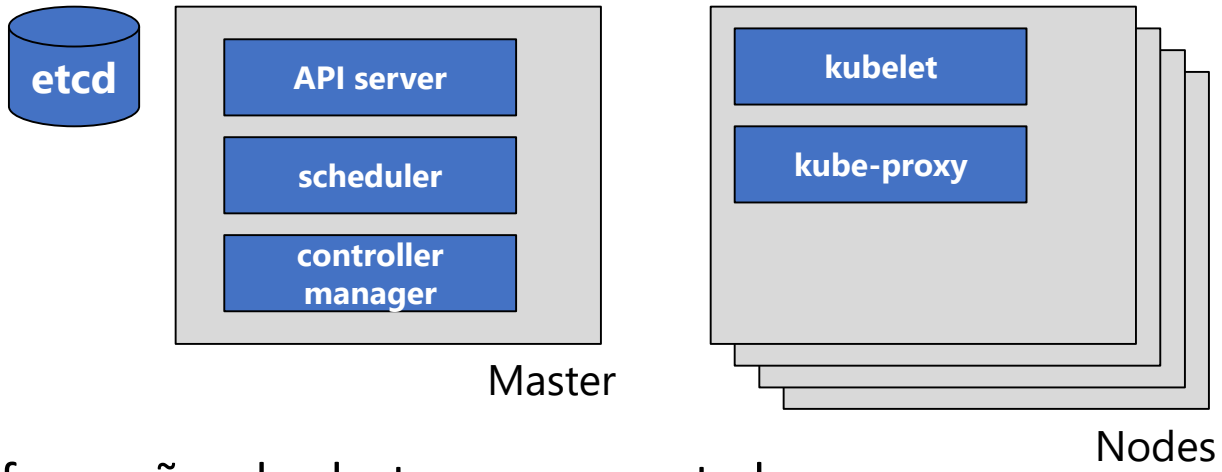
- Provê a orquestração dos containers, de rede, dos volumes (discos) e dos demais recursos para que as aplicações executem com disponibilidade e escalabilidade
- Não possui nenhum run-time nativo: Precisa que seja instalado no mínimo um gerenciador de rede e o run-time dos containers (Docker?, ContainerD?, CRI-O?)

Kubernetes
Arquitetura



- API server: front-end do plano de controle. Ela processa solicitações internas e externas
- Scheduler: Ele analisa quais os recursos de que um pod necessita, como CPU e memória e aloca o POD a algum node
- Controller manager: executa o controle do cluster, verifica se os pods estão em execução, se os serviços estão disponíveis, etc.

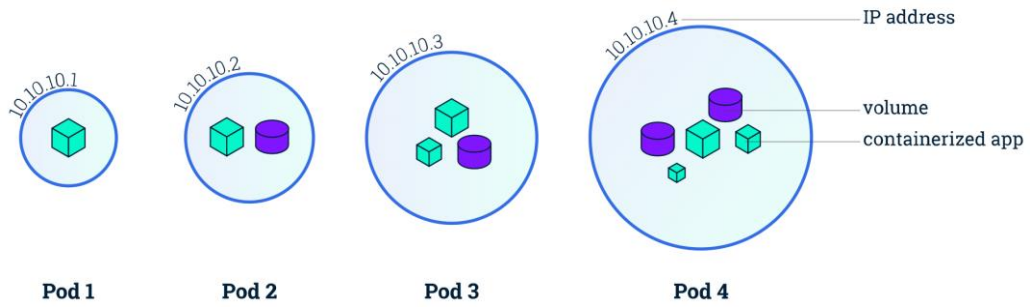
Kubernetes Arquitetura



- etcd: é um BD de “chave:valor” com todas as informações do cluster, como o estado dos serviços e as configurações
- Kubelet: Quando o plano de controle precisa que algo aconteça em um nó, o kubelet realiza a ação
- kube-proxy: encaminhamento de pacotes TCP/UDP interno e externo

Kubernetes
Principais objetos

- Pod
  - Menor unidade de computação
  - Consiste em um objeto que executa uma instância de uma aplicação. Possui um endereço IP interno único e pode possuir acesso a volumes e outros objetos.



- Deployment
  - O Deployment define a criação das instâncias do seu aplicativo (Pods). Depois de criar um Deployment, o Master do Kubernetes agenda a criação dos Pods nos nós do Cluster.

Execução de aplicações em kubernetes

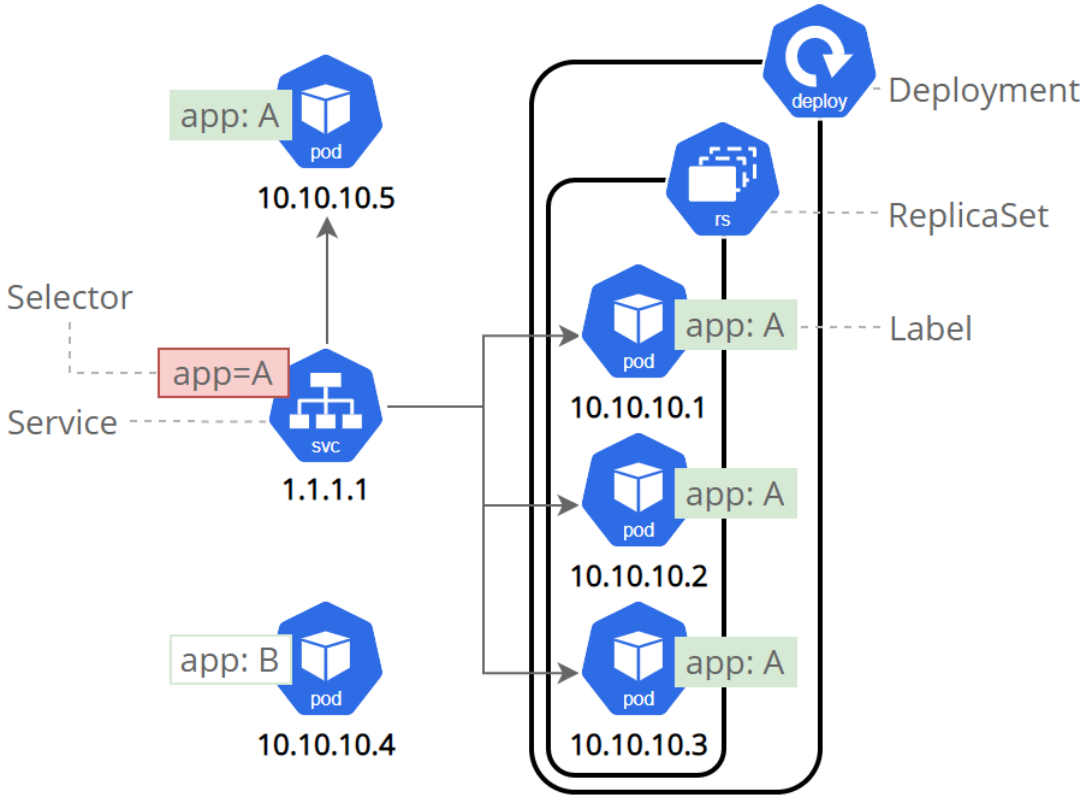

```
D:\Users\Cesar Loureiro>kubectl get services -o wide
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE   SELECTOR
kubernetes          ClusterIP      10.96.0.1       <none>         443/TCP          18d   <none>
meuapache-service3  LoadBalancer   10.110.246.4    200.132.0.75   80:31543/TCP     17d   app=meuapache
nginx               ClusterIP      10.109.220.40   200.132.0.77   80/TCP,443/TCP   9d    app=nginx,tier=backend
php                 ClusterIP      10.105.199.85   <none>         9000/TCP         9d    app=php,tier=backend

D:\Users\Cesar Loureiro>kubectl get pods -o wide
NAME                                  READY   STATUS    RESTARTS   AGE     IP           NODE         NOMINATED NODE   READINESS GATES
meu-nginx-5d464b9b67-m75pk            1/1     Running   1          9d      10.38.0.6    k8s-node02   <none>           <none>
meuapache-deployment-6dc575d4cd-fmm8m 1/1     Running   0          3d19h   10.40.0.5    k8s-node01   <none>           <none>
meuapache-deployment-6dc575d4cd-mc26t 1/1     Running   0          3d19h   10.38.0.7    k8s-node02   <none>           <none>
meuapache-deployment-6dc575d4cd-tj99d 1/1     Running   0          3d19h   10.40.0.6    k8s-node01   <none>           <none>
php-7bcf88555c-tngxk                  1/1     Running   1          9d      10.40.0.2    k8s-node01   <none>           <none>

D:\Users\Cesar Loureiro>kubectl get deploy -o wide
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS           IMAGES                          SELECTOR
meu-nginx             1/1     1            1           9d    nginx                nginx:1.7.9                     app=nginx,tier=backend
meuapache-deployment  3/3     3            3           17d   meuapache-container  cesarloureiro/aplicacao:v8      app=meuapache
php                   1/1     1            1           9d    php                  php:7-fpm                       app=php,tier=backend
```

Kubernetes
Principais objetos

- Services
  - Atribui um IP (interno / externo ao cluster) e uma Porta (interna / externa ao cluster)
  - Também chamado de frontend, é a exposição da aplicação para através da rede

Tipos:

- ClusterIP - Expõe o serviço utilizando um IP interno do Cluster, mas pode ser atribuido a um externalIP
- Node Port - Comunicação externa, mas através de portas altas (acima de 3000)
- LoadBalance - solicita IP ao provedor de serviço, ou balanceador externo (usado em Núvens Públicas)
- Externalname - redirecionamento através de DNS



Execução de aplicações em kubernetes


```
D:\Users\Cesar Loureiro>kubectl get services -o wide
NAME                TYPE           CLUSTER-IP      EXTERNAL-IP    PORT(S)          AGE   SELECTOR
kubernetes          ClusterIP      10.96.0.1       <none>         443/TCP          18d   <none>
meuapache-service3  LoadBalancer   10.110.246.4    200.132.0.75   80:31543/TCP     17d   app=meuapache
nginx               ClusterIP      10.109.220.40   200.132.0.77   80/TCP,443/TCP   9d    app=nginx,tier=backend
php                 ClusterIP      10.105.199.85   <none>         9000/TCP         9d    app=php,tier=backend

D:\Users\Cesar Loureiro>kubectl get pods -o wide
NAME                                  READY   STATUS    RESTARTS   AGE     IP           NODE         NOMINATED NODE   READINESS GATES
meu-nginx-5d464b9b67-m75pk            1/1     Running   1          9d      10.38.0.6    k8s-node02   <none>           <none>
meuapache-deployment-6dc575d4cd-fmm8m 1/1     Running   0          3d19h   10.40.0.5    k8s-node01   <none>           <none>
meuapache-deployment-6dc575d4cd-mc26t 1/1     Running   0          3d19h   10.38.0.7    k8s-node02   <none>           <none>
meuapache-deployment-6dc575d4cd-tj99d 1/1     Running   0          3d19h   10.40.0.6    k8s-node01   <none>           <none>
php-7bcf88555c-tngxk                  1/1     Running   1          9d      10.40.0.2    k8s-node01   <none>           <none>

D:\Users\Cesar Loureiro>kubectl get deploy -o wide
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS           IMAGES                          SELECTOR
meu-nginx            1/1     1            1           9d    nginx                nginx:1.7.9                     app=nginx,tier=backend
meuapache-deployment 3/3     3            3           17d   meuapache-container  cesarloureiro/aplicacao:v8      app=meuapache
php                  1/1     1            1           9d    php                  php:7-fpm                       app=php,tier=backend
```

Exemplo de YAML

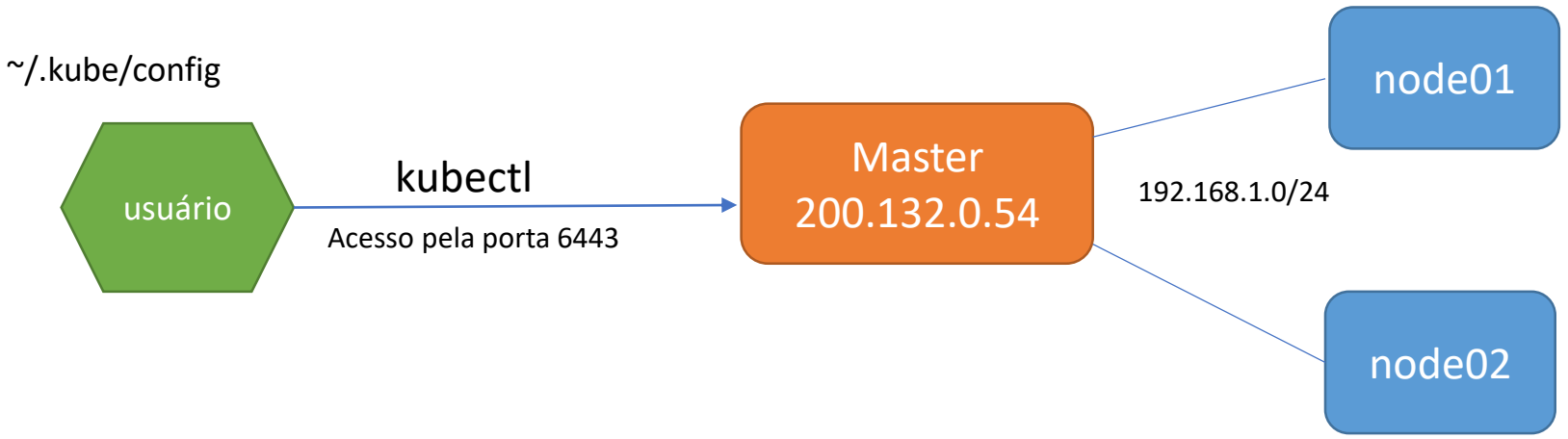
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: meuapache-deployment
  labels:
    app: meuapache
spec:
  replicas: 6
  selector:
    matchLabels:
      app: meuapache
  template:
    metadata:
      labels:
        app: meuapache
    spec:
      containers:
      - name: meuapache-container
        image: cesarloureiro/aplicacao:v8
        ports:
        - containerPort: 80
      imagePullSecrets:
        - name: regcred
```

deploy-meuapache.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: meuapache-service3
spec:
  selector:
    app: meuapache
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: LoadBalancer
  externalIPs:
  - 200.132.0.75
```

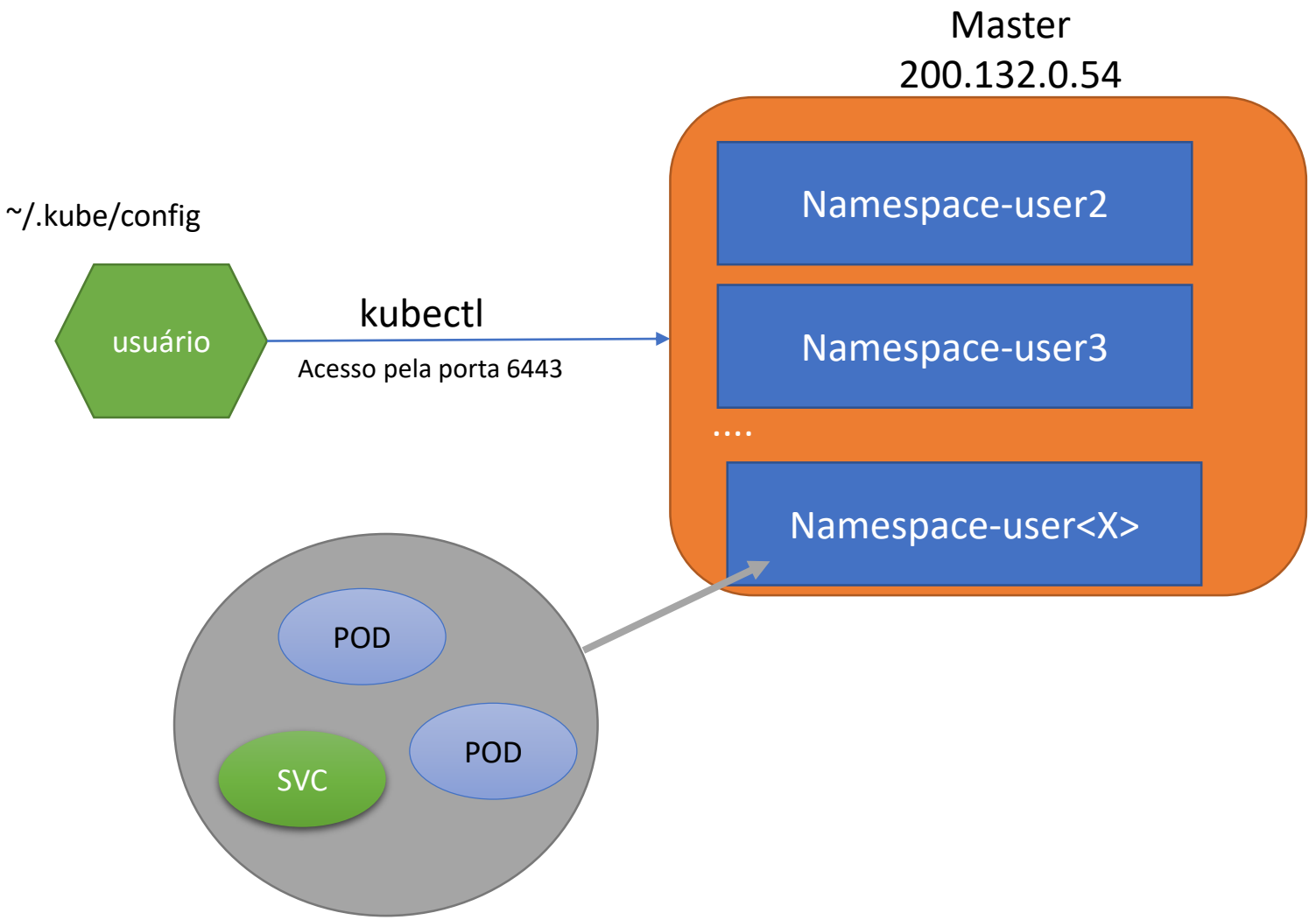
service-meuapache.yaml

Estrutura do laboratório



IPs disponíveis para
Serviços: 200.238.30.0/26

Estrutura do laboratório



Mão na massa

- http://www.pop-rs.rnp.br/kubernetes

Acesso ao cluster

- Cada usuário deverá criar uma pasta /.kube em seu home_directory e salvar seu arquivo “config” que contem as chaves de acesso ao servidor do Master do kubernetes.

Windows

xcopy <arquivo-conf> "%USERPROFILE%/.kube/config”

Selecione a opção arquivo na pergunta

Linux

```
mkdir ~/.kube/
cp  <arquivo-conf>  ~/.kube/config
```

Exemplo 1

- Realizar o deploy de um nginx com sua página padrão na porta 80 (HTTP) em um IP externo do cluster.

Criação do Deployment:
#kubectl create deployment hello --image=nginx

Verificar sua execução:
#kubectl get pods --output=wide

** Criar um serviço:
#kubectl create service clusterip hello --tcp=80:80

Editar um serviço para expor sua aplicação para a Internet:
#kubectl edit service hello

Incluir as duas linhas a seguir logo abaixo do endereço IP de "clusterIP".
Substituindo o <X> pelo seu Número

```
  externalIPs:
  - 200.238.30.<X>
```

** Exemplo do arquivo**

```
---------->
spec:
  clusterIP: 10.98.117.163
  clusterIPs:
  - 10.98.117.163
  externalIPs:
  - 200.238.30.99
  ipFamilies:
  - IPv4
<----------
```

Validação

- Realizar o deploy de um nginx com sua página padrão na porta 80 (HTTP) em um IP externo do cluster.

Visualize o deployment, o serviço, e o pod
#kubectl get deploy --output=wide
#kubectl get services --output=wide
#kubectl get pods --output=wide

Veja a descrição do pod em exercução
#kubectl describe pod <nome_pod>

#Acesse seu serviço via navegador:
http://200.238.30.<X>

Acesse o terminal do pod
#kubectl exec --stdin --tty <nome do pod> -- /bin/bash

Altere o conteúdo do index.htm
#apt-get update
#apt-get install vim
#vim /usr/share/nginx/html/index.html

Teste via web

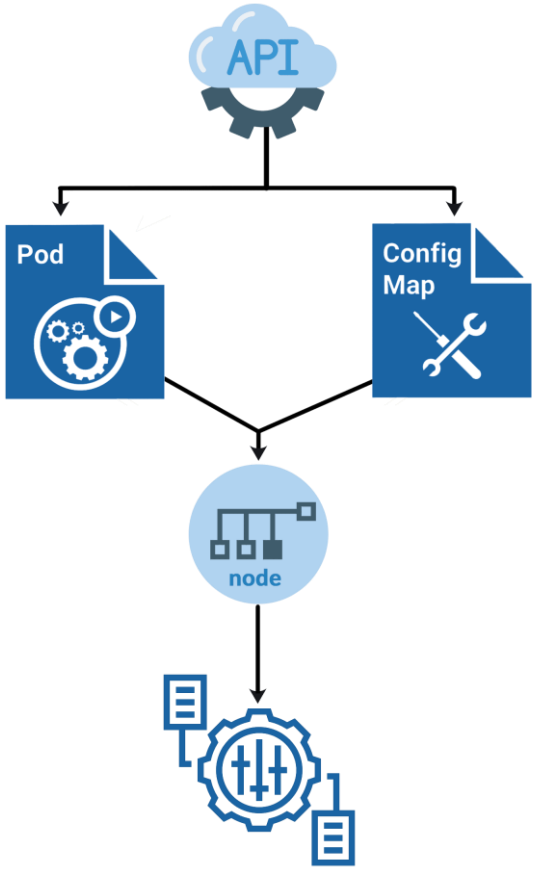
Escale o número de pods para 3
#kubectl scale deployment/hello --replicas=3

Teste via web

Apague tudo:
#kubectl delete deploy hello
#kubectl delete sevices hello

Kubernetes
mais objetos importantes

- ConfigMap
  - É um objeto que contem conjuntos de chave-valor para armazenamento de configurações, que serão utilizados pelos Pods.
    - Exemplo: Configuração do Apache
  - Outros tipos de objetos:
    - Namespaces, secrets, roles, rolebinds, ingress,



Exemplo 2

- Realizar o deploy de um aplicação nginx através de arquivos yaml, criando um index.html da imagem do container utilizando o objeto **ConfigMap**

Criação index.html
echo "<html><head><title>Exemplo 2 </title></head>
<body>Exemplo 2</body></html>" > .\index.html

Importar o index.html para o configmap:
kubectl create configmap cm-exemplo2 --from-file=.\index.html

Olhar seu conteudo
kubectl get configmap cm-exemplo2 -o yaml

Criar o deploy da aplicação
notepad app-exemplo2.yaml

Exemplo 2 - aplicação

- Realizar o deploy de um aplicação nginx através de arquivos yaml, alterando o index.html da imagem do container utilizando o objeto **ConfigMap**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-exemplo2
spec:
  selector:
    matchLabels:
      app: app-exemplo2
  template:
    metadata:
      labels:
        app: app-exemplo2
```

```
spec:
      containers:
      - name: container-exemplo2
        image: nginx:latest
        volumeMounts:
        - name: html
          mountPath: /usr/share/nginx/html
      volumes:
      - name: html
        configMap:
          name: cm-exemplo2
```

#kubectl apply -f .\app-exemplo2.yaml

Exemplo 2 - serviço

- Realizar o deploy de um serviço para expor a porta 80 da aplicação criada anteriormente

#notepad svc-exemplo2.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: svc-exemplo2
  labels:
spec:
  selector:
    app: app-exemplo2
  ports:  # expondo o nginx
  - name: http
    port: 80    # porta externa utilizada no externalIP
    protocol: TCP
externalIPs:
  - 200.238.30.<X>
```

#kubectl apply -f .\svc-exemplo2.yaml

WTR

WORKSHOP
DE TECNOLOGIA DE REDES DO POP-RS
> 2021

22
SET

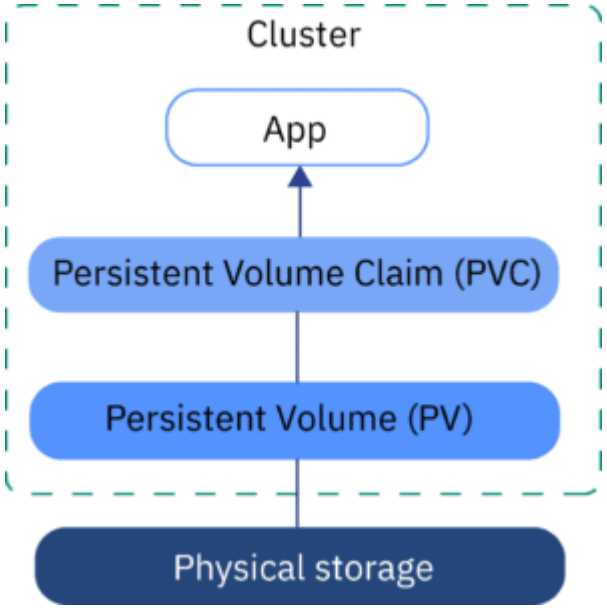
cafe?

PoP-RS
Ponto de Presença da
RNP no Rio Grande do Sul

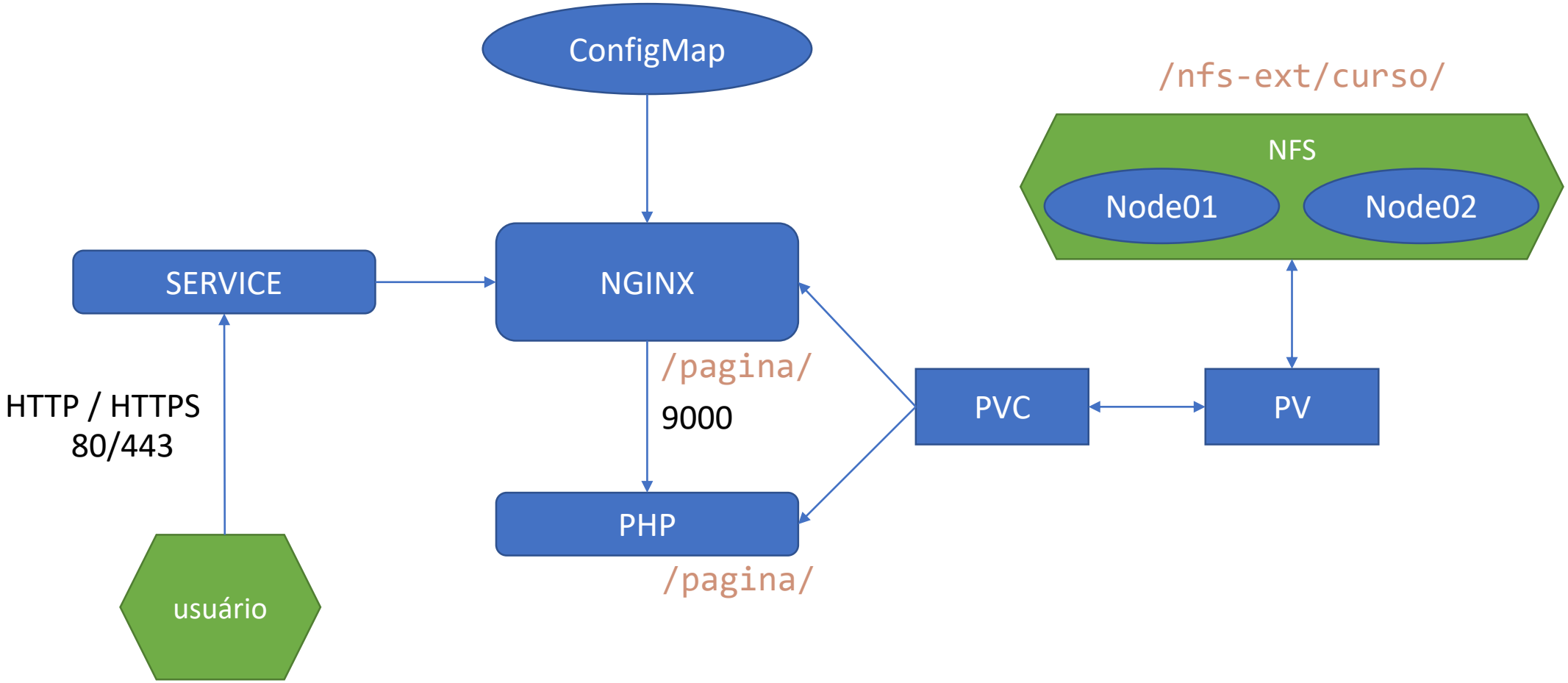
RNP
ORGANIZAÇÃO SOCIAL DO MCTI

Kubernetes
mais objetos importantes

- Volumes
  - PersistentVolume
    - PVs são recursos dentro um cluster, normalmente disponibilizado em um Array externo ou por NFS entre os nodes
  - PersistentVolumeClaim
    - PVCs são requisições para esses recursos e também atuam como validação da solicitação desses recursos
  - StorageClass
    - É a definição do tipo de PV, onde pode definir o tempo de retenção dos dados, ou diferentes velocidades de disco(hardware) para diferentes storageClass



Exemplo 3


ConfigMap
/nfs-ext/curso/
NFS
Node01
Node02
SERVICE
NGINX
/pagina/
9000
HTTP / HTTPS
80/443
PVC
PV
PHP
/pagina/
usuário

Obrigado (a)!

César Loureiro
cesar.loureiro@pop-rs.rnp.br

WTR
WORKSHOP
DE TECNOLOGIA DE REDES DO POP-RS
2021

APOIO nic.br

REALIZAÇÃO PoP-RS Ponto de Presença da RNP no Rio Grande do Sul

RNP ORGANIZAÇÃO SOCIAL DO MCTI

MINISTÉRIO DO TURISMO
MINISTÉRIO DA DEFESA
MINISTÉRIO DA SAÚDE
MINISTÉRIO DAS COMUNICAÇÕES
MINISTÉRIO DA EDUCAÇÃO
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES

PÁTRIA AMADA BRASIL GOVERNO FEDERAL