# Armazenamento e Controladores

Pedro Horchulhack

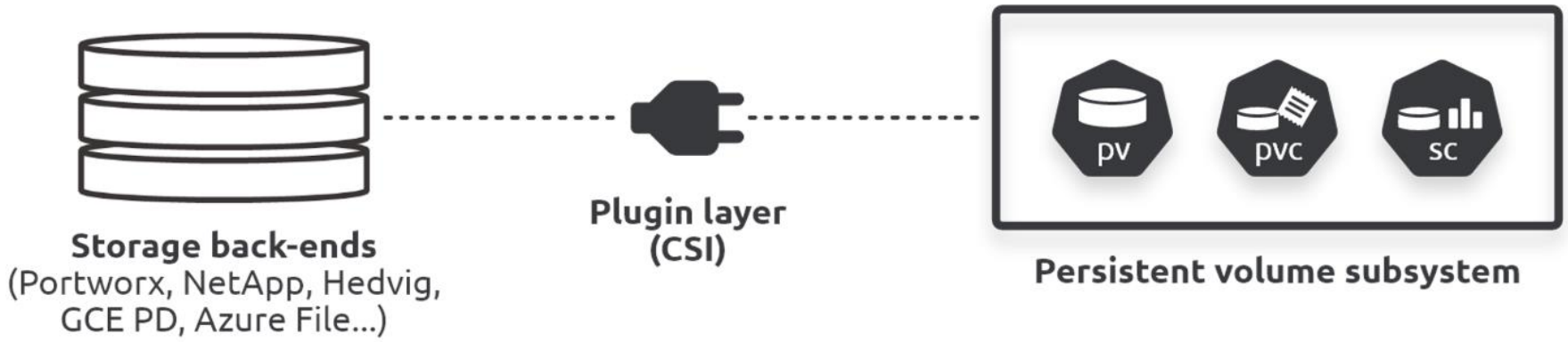
# Agenda

1. Armazenamento
2. Controladores
3. Atividade

# Armazenamento

# Armazenamento

- O armazenamento de dados é imprescindível em qualquer aplicação real
- Como visto em Docker, é criado um sistema de arquivos próprio para lidar com arquivos do host para os contêineres
- No Kubernetes temos algo semelhante, o Subsistema de Volumes Persistentes




Fonte: POULTON, Nigel; JOGLEKAR, Pushkar. **The Kubernetes book**. Version 4, March 2019. United Kingdom: Nigel Poulton, 2019.

# Armazenamento

- O Kubernetes cria uma abstração para o sistema de arquivos
  - Recurso denominado Volume (Persistentes e Efêmeros)

- Temos três recursos da sua API para permitir trabalhar com persistência de dados
  - Persistent Volumes (PV)
  - Persistent Volume Claims (PVC)
  - Storage Classes (SC)

# Armazenamento

- Persistent Volume (PV)
  - É como o Kubernetes repereseNta o armazenamento externo
- Persistent Volume Claims (PVC)
  - Atuam como “tickets” que concedem acesso do PV aos Pods
- Storage Classes (SC)
  - Separação lógica/tierização de PVs

# Armazenamento

- Podemos provisionar um Volume Persistente de duas formas:
  - Estática (Existem na API do Kubernetes)
  - Dinâmica (Baseado em StorageClasses)

- Tipos de Volumes Persistentes:

| Nome | Descrição |
|---|---|
| Container Storage Interface (CSI) | Define uma interface padrão para sistemas de arquivos |
| Fibre Channel (FC) | Permite armazenamento via fibra ótica |
| hostPath | Monta um diretório do nó do Pod |
| SCSI over IP (ISCSI) | Implementação do protocolo iSCSI |
| Local | Representa um disco, partição ou diretório |
| Network File System (NFS) | Implementação do protocolo NFS |

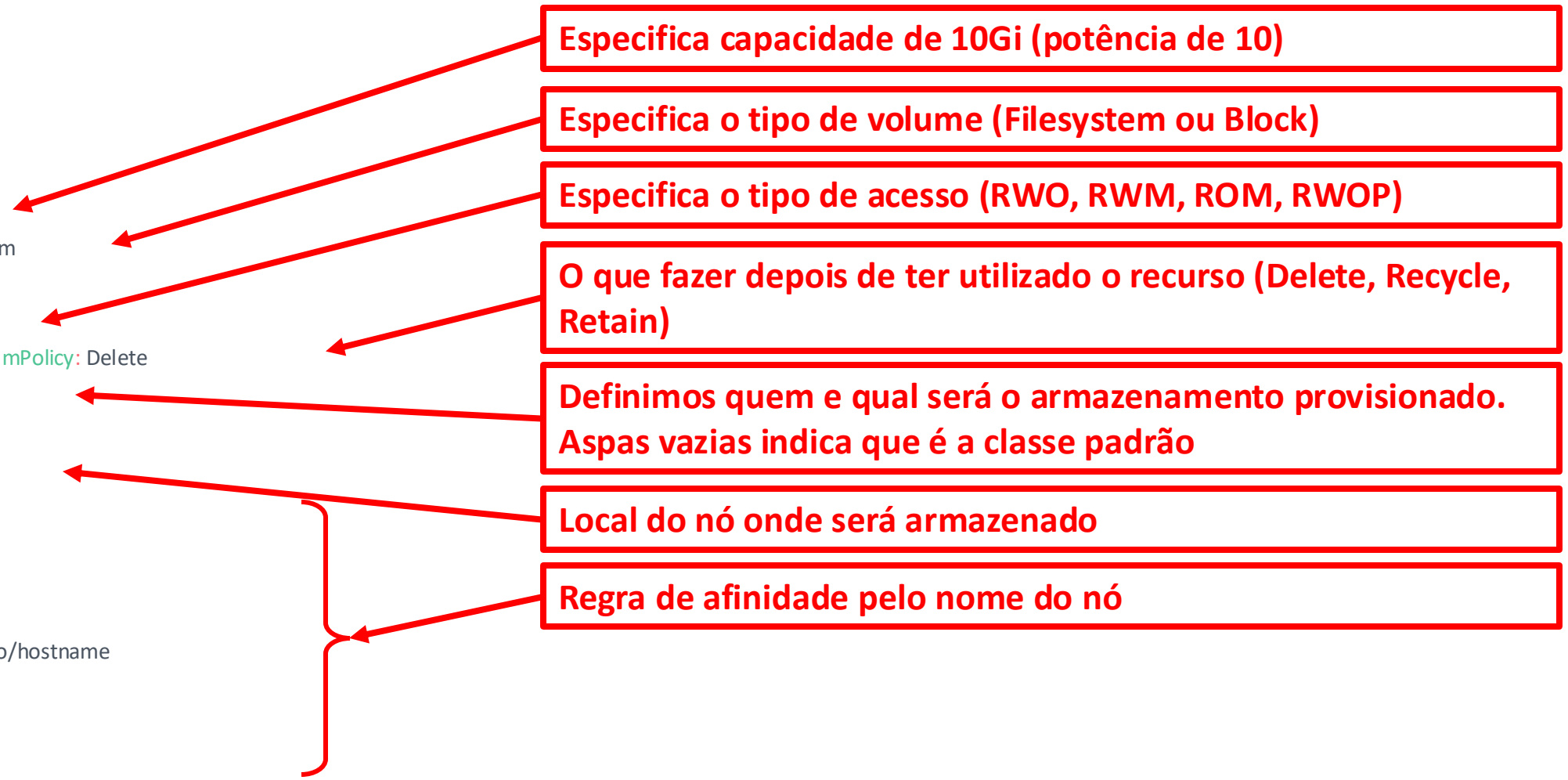
# Armazenamento – PersistentVolume

- Um PersistentVolume local é estático
  - No provisionamento dinâmico ele pode ser criado sob demanda

- Esses volumes estão sujeitos a disponibilidade dos nós do cluster
  - O diretório/partição/disco devem existir para funcionar!

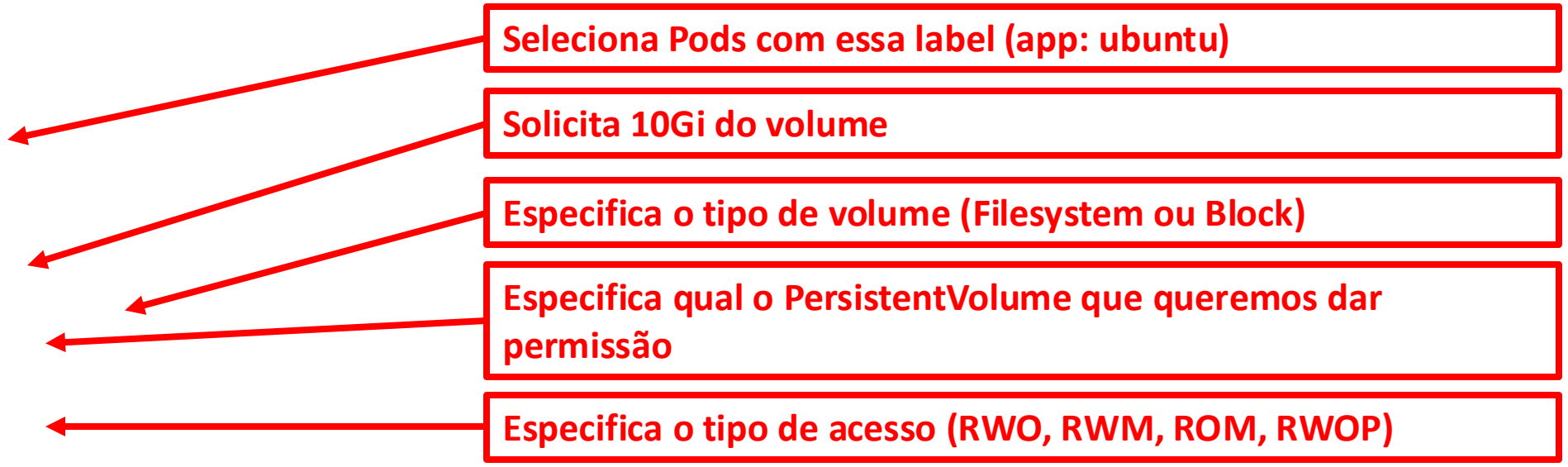
# Armazenamento – PersistentVolume

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
 name: pod-pv
spec:
 capacity:
  storage: 10Gi
 volumeMode: Filesystem
 accessModes:
  - ReadWriteMany
 persistentVolumeReclaimPolicy: Delete
 storageClassName: ""
 local:
  path: /tmp/teste
 nodeAffinity:
  required:
   nodeSelectorTerms:
    - matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
         - secplab00
```



# Armazenamento – PersistentVolumeClaim

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pod-pvc
spec:
  selector:
    matchLabels:
      app: ubuntu
  resources:
    requests:
      storage: 10Gi
  volumeMode: Filesystem
  volumeName: pod-pv
  accessModes:
    - ReadWriteMany
  storageClassName: ""
```



# Armazenamento – Pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu
  labels:
    app: ubuntu
spec:
  containers:
    - image: ubuntu:22.04
      name: ubuntu
      command: ["tail", "-f", "/dev/null"]
      volumeMounts:
        - mountPath: /mnt
          name: data
  volumes:
    - name: data
      persistentVolumeClaim:
        claimName: pod-pvc
```

**Monta o PV de nome data no diretório `/mnt` do contêiner**

**Indica o nome do volume (para ser usado pelos contêineres) e qual o nome do claim (pod-pvc)**

# Armazenamento – Pod

- Testar implementação dos slides anteriores
  - Acessar Pod
  - Criar arquivo no diretório compartilhado
  - Sair do Pod
  - Verificar o diretório do Persistent Volume

# ConfigMaps e Segredos

# ConfigMaps e Segredos

- ConfigMap é outro tipo de objeto no Kubernetes, porém voltado ao armazenamento de configurações **fora** de um Pod

- Podemos utilizar para armazenar, por exemplo:
  - Variáveis de ambiente
  - Arquivos de configuração
  - Hostnames
  - Portas de serviços
  - Nomes de contas
  - ...

- **Não utilizamos ConfigMaps para armazenar informações sensíveis!!!**

# ConfigMaps e Segredos

- Para o armazenamento de informações sensíveis, temos o objeto Secret
  - Ainda assim é inseguro! Ele não é encriptado *at rest*. No entanto, podemos contornar isso com um objeto EncryptionConfiguration (Não será abordado)
  - Os dados são encriptados no tráfego

- Ambos (ConfigMap e Segredos) são uma lista de entredas chave/valor

# ConfigMaps e Segredos

Definição de um simples ConfigMap

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: meu-configmap
data:
  SERVER_ADDR: "127.0.0.1"
immutable: true
```

# ConfigMaps e Segredos

Associação de ConfigMap à um Pod

Aqui carregamos os dados como variáveis de ambiente!

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: busybox
spec:
  containers:
    - name: busybox
      image: busybox
      command: ["tail", "-f", "/dev/null"]
      envFrom:
        - configMapRef:
            name: meu-configmap
```

# ConfigMaps e Segredos

Definição de um Secret

- Convertemos valores para base64

```
echo -n 'admin' | base64
echo -n 'p@ssw0rd' | base64
```

```
apiVersion: v1
kind: Secret
metadata:
  name: meu-segredo
type: Opaque
data:
  usuario: YWRtaW4=
  senha: cEBzc3cwcmQ=
```

# ConfigMaps e Segredos

Associação de Segredo à um Pod

Aqui carregamos os dados como variáveis de ambiente!

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: busybox
spec:
  containers:
    - name: busybox
      image: busybox
      command: ["tail", "-f", "/dev/null"]
      envFrom:
        - secretRef:
            name: meu-segredo
```

# Referências

# Referências


1. https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands
2. https://microk8s.io/docs/getting-started
3. https://kubernetes.io/docs/concepts/services-networking/service/
4. https://kubernetes.io/pt-br/docs/concepts/overview/components/
5. https://kubernetes.io/pt-br/docs/concepts/overview/working-with-objects/
6. https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
7. https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/
8. https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/
9. https://kubernetes.io/docs/concepts/workloads/controllers/deployment/

# Referências


10. https://kubernetes.io/docs/tasks/configmap-secret/managing-secret-using-config-file/
11. https://kubernetes.io/docs/concepts/configuration/secret/
12. POULTON, Nigel; JOGLEKAR, Pushkar. **The Kubernetes book**. Version 4, March 2019. United Kingdom: Nigel Poulton, 2019.
13. RICE, Liz. **Container security: fundamental technology concepts that protect containerized applications**. First edition. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, 2020.

Contato: p.horchulhack@pucpr.br