

Mãos à obra: Permissões

Uma vez que criamos o volume chamado **volume-mysql**, temos que configurar a quantidade de recursos que nosso Pod com o banco de dados terá permissão de consumir. Para isso, crie um novo arquivo chamado **permissoes.yaml**.

Como já vimos, a primeira linha desse nosso arquivo seria responsável por informar a versão da API do Kubernetes responsável pela esquematização do objeto **PersistentVolumeClaim**:

```
apiVersion: v1
```

Na sequência, vamos informar que iremos configurar o tipo de objeto **PersistentVolumeClaim** e daremos um nome de identificação como sendo **configuracao-mysql**

```
kind: PersistentVolumeClaim
metadata:
  name: configuracao-mysql
```

Posteriormente, devemos especificar como será o tipo de acesso, bem como o armazenamento que iremos requisitar para a nossa aplicação. Vamos configurar o modo de acesso como sendo de leitura e escrita somente para a máquina que receber o Pod com o banco de dados e um armazenamento de 3Gi

```
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

Na sequência, devemos informar que essa configuração do **PersistentVolumeClaim** deverá ser utilizada pelo nosso Pod com o banco de dados para acessar o volume chamado **volume-mysql**. Volte ao arquivo **statefulset.yaml** e coloque dentro da chave **volumes** o nome do **PersistentVolumeClaim** que terá as configurações de consumo de recursos desse volume chamado **volume-mysql**

```
volumes:
  - name: volume-mysql
    persistentVolumeClaim:
      claimName: configuracao-mysql
```

Uma vez que nossa configuração do banco foi realizada, precisamos fazer com que a aplicação web acesse esse banco. Essa será nossa próxima tarefa.