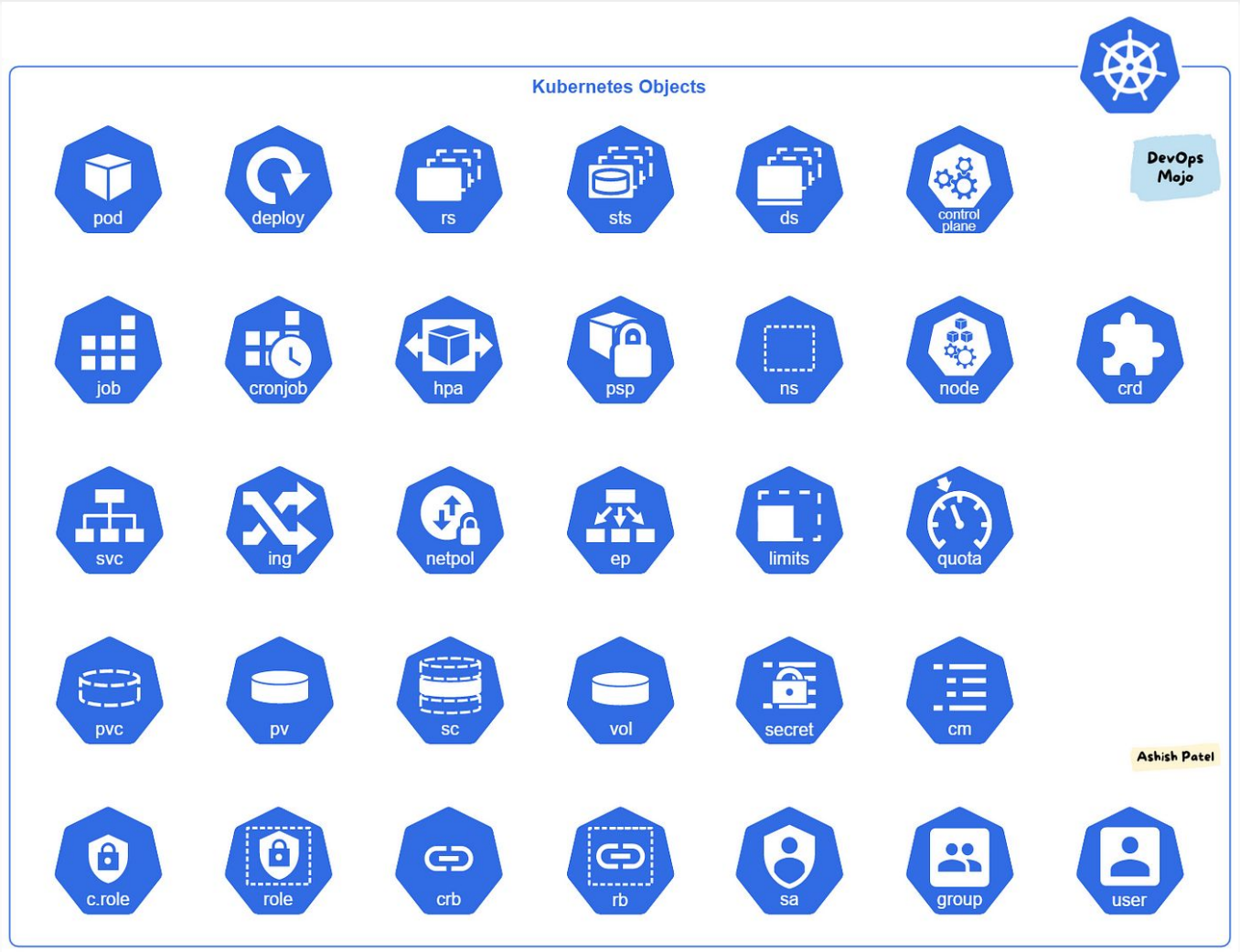
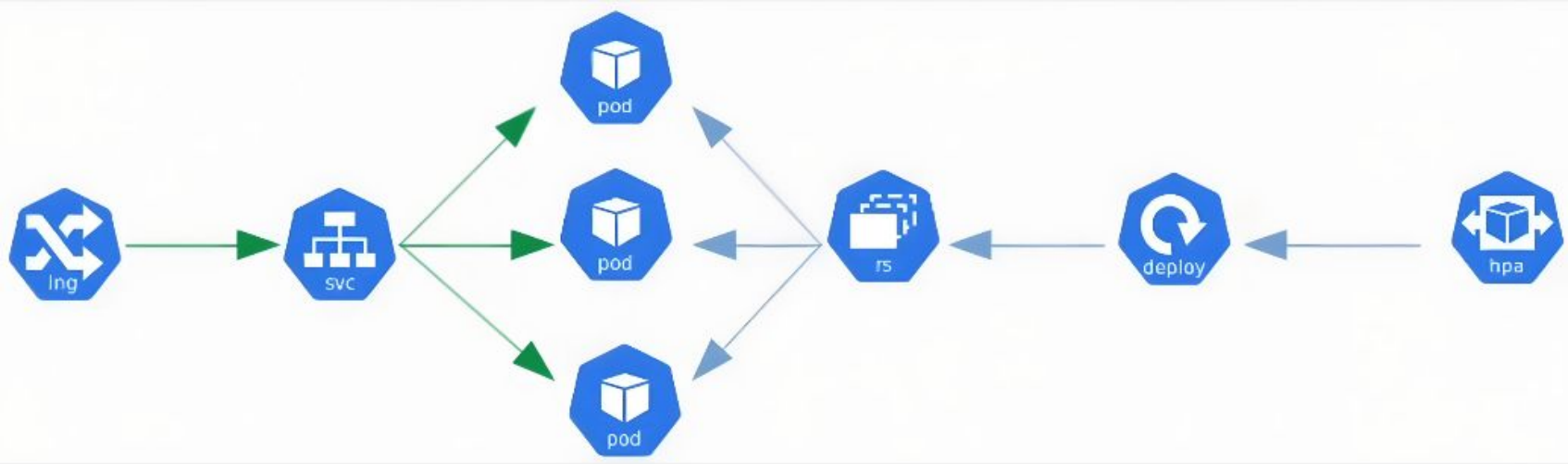
Orquestração de containers com Kubernetes

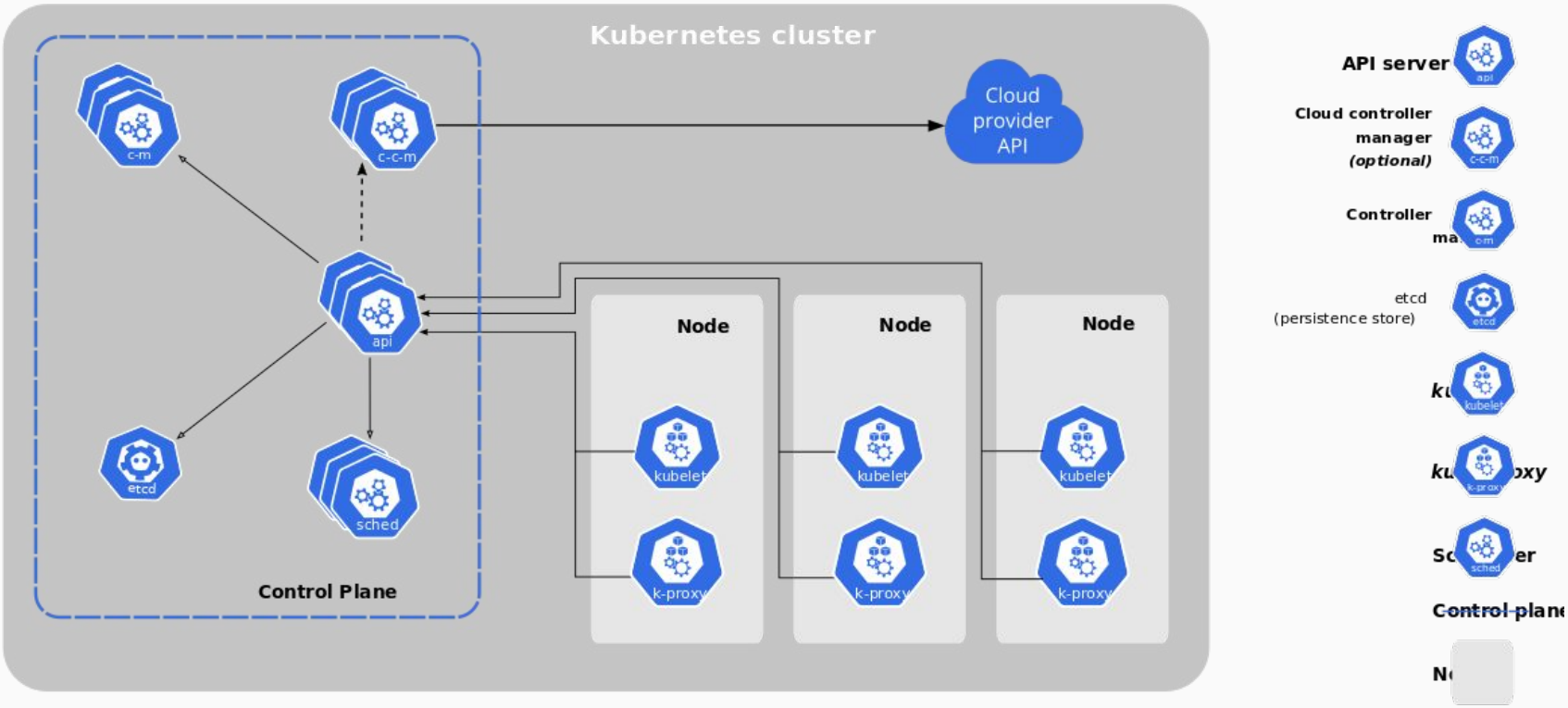
Igor Follador e Jaisson Bassanesi

Kubernetes e seus componentes





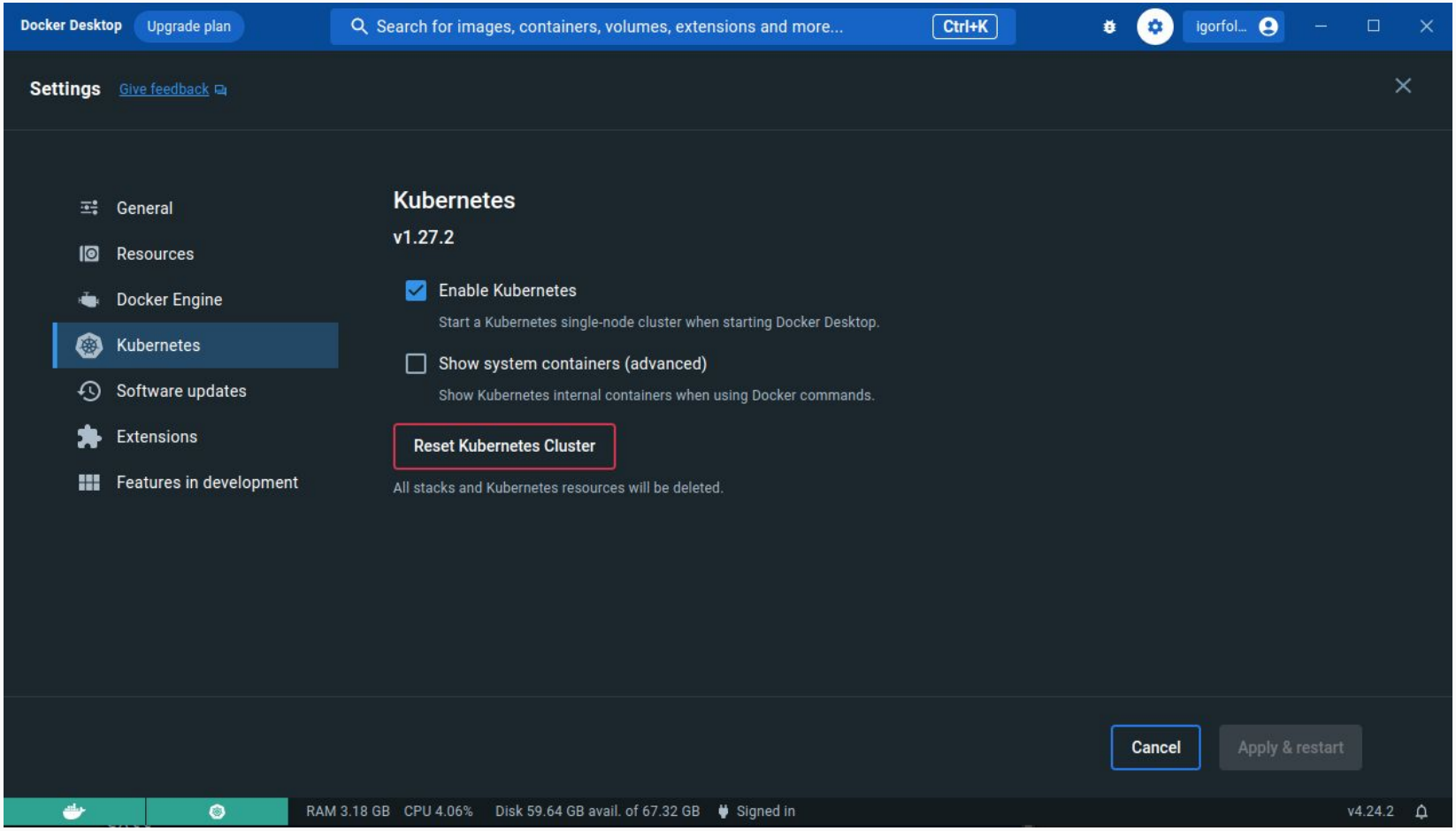
Simples diagrama de sistema orquestrado com Kubernetes



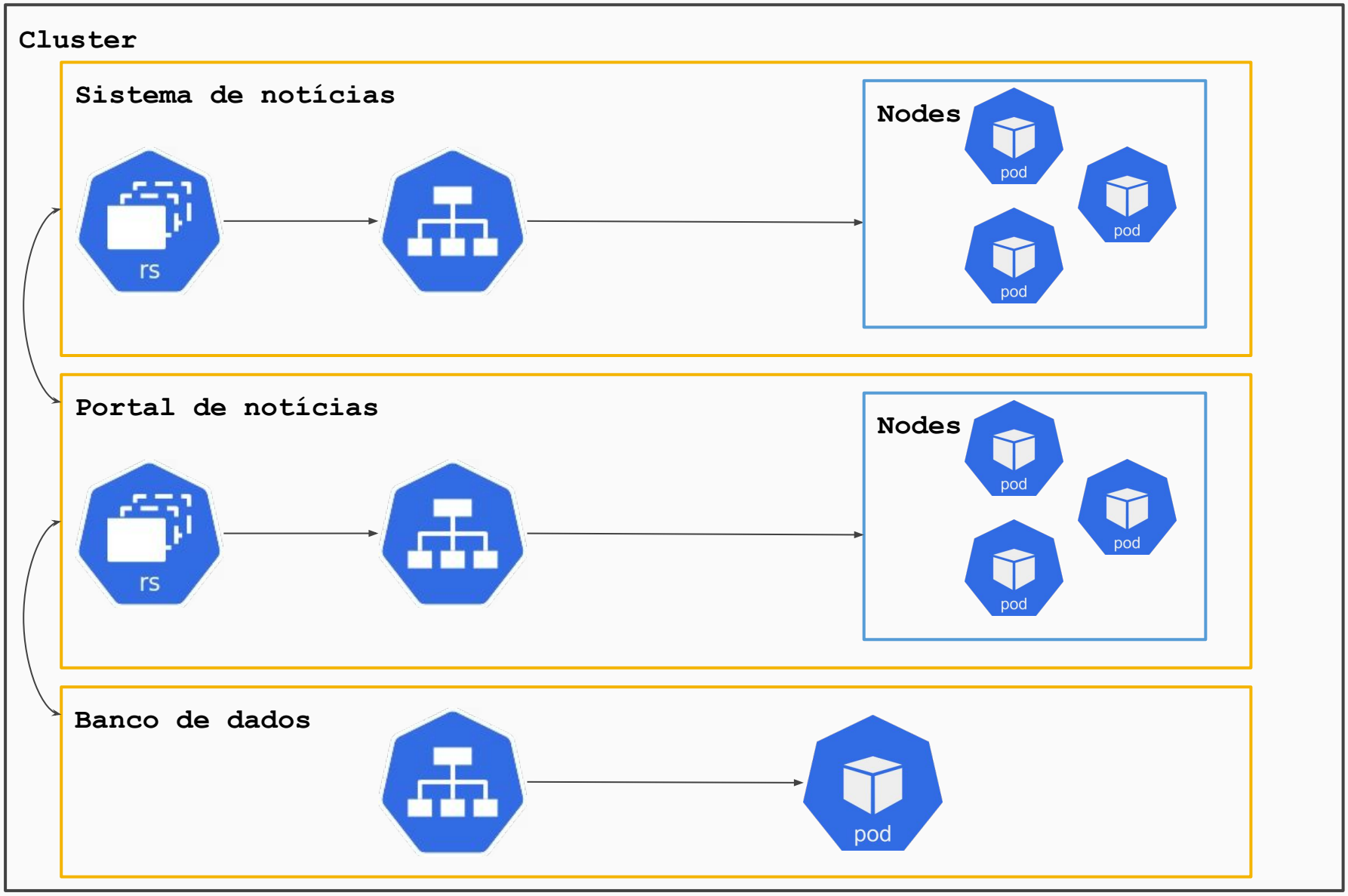
Simples diagrama de sistema orquestrado com Kubernetes

Montando o primeiro cluster
pré-requisitos

- Instalar o Docker Desktop;
- Habilitar o Kubernetes no Docker Desktop;



Montando o primeiro cluster
passo 0 - visão do cluster



Montando o primeiro cluster
passo 1 - criando o Pod do portal

```
# portal-noticias.yaml
apiVersion: v1
kind: Pod
metadata:
  name: portal-noticias
  labels:
    app: portal-noticias
spec:
  containers:
    - name: portal-noticias-container
      image: aluracursos/portal-noticias:1
      ports:
        - containerPort: 80
```

Montando o primeiro cluster
passo 1 - criando o Pod do portal

kubectl apply -f portal-noticias.yaml
kubectl get pods - - watch

```
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f portal-noticias.yaml
pod/portal-noticias created
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get pods --watch
NAME              READY   STATUS    RESTARTS   AGE
portal-noticias   1/1     Running   0          8s
```

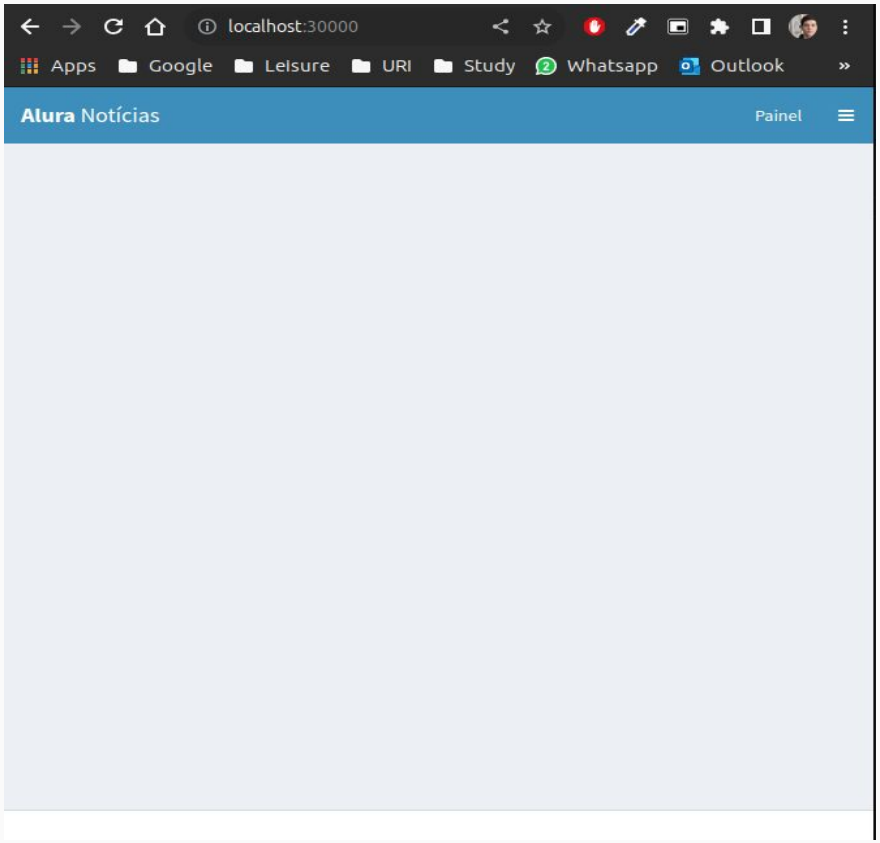
Montando o primeiro cluster
passo 2 - criando o Service do portal (NodePort)

```
# svc-portal-noticias.yaml
apiVersion: v1
kind: Service
metadata:
  name: svc-portal-noticias
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 30000
  selector:
    app: portal-noticias
```

Montando o primeiro cluster
passo 2 - criando o Service do portal (NodePort)

```
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f svc-portal-noticias.yaml
service/svc-portal-noticias created
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get services
NAME                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes            ClusterIP   10.96.0.1        <none>        443/TCP        109m
svc-portal-noticias   NodePort    10.101.151.165   <none>        80:30000/TCP   16s
```

kubectl apply -f svc-portal-noticias.yaml
kubectl get services



Montando o primeiro cluster
passo 3 - criando o Pod do sistema de cadastro

```
# sistema-noticias.yaml
apiVersion: v1
kind: Pod
metadata:
  name: sistema-noticias
  labels:
    app: sistema-noticias
spec:
  containers:
    - name: sistema-noticias-container
      image: aluracursos/sistema-noticias:1
      ports:
        - containerPort: 80
```

passo 3 - criando o Pod do sistema de cadastro

kubectl apply -f sistema-noticias.yaml
kubectl get pods

```
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f sistema-noticias.yaml
pod/sistema-noticias created
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get pods
NAME               READY   STATUS    RESTARTS   AGE
portal-noticias    1/1     Running   0          7m8s
sistema-noticias   1/1     Running   0          8s
```

Montando o primeiro cluster
passo 4 - criando o Service do sistema de cadastro

```
# svc-sistema-noticias.yaml
apiVersion: v1
kind: Service
metadata:
  name: svc-sistema-noticias
spec:
  type: NodePort
  ports:
    - port: 80
      nodePort: 30001
  selector:
    app: sistema-noticias
```

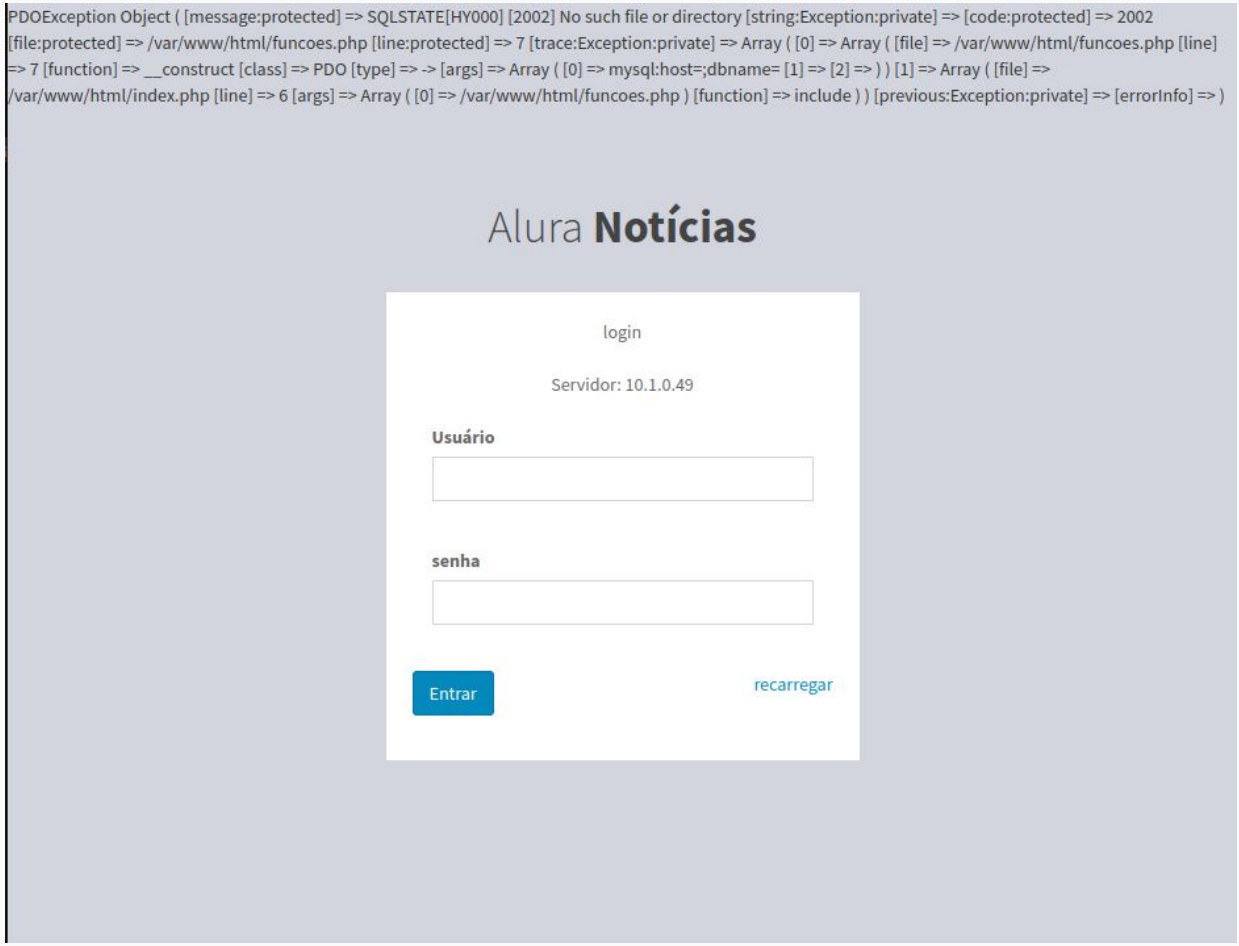
Montando o primeiro cluster
passo 4 - criando o Service do sistema

kubectl apply -f svc-sistema-noticias.yaml
kubectl get svc

```
folladorr@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f svc-sistema-notic
ias.yaml
service/svc-sistema-noticias created
folladorr@dell-3590:~/GitHub/alura-kubernetes$ kubectl get svc
NAME                    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
 AGE
kubernetes              ClusterIP   10.96.0.1        <none>        443/TCP
 151m
svc-portal-noticias     NodePort    10.101.151.165   <none>        80:30000/TCP
 42m
svc-sistema-noticias    NodePort    10.101.251.192   <none>        80:30001/TCP
 13s
```

Montando o primeiro cluster
passo 4 - criando o Service do sistema

kubectl apply -f svc-sistema-noticias.yaml
kubectl get svc

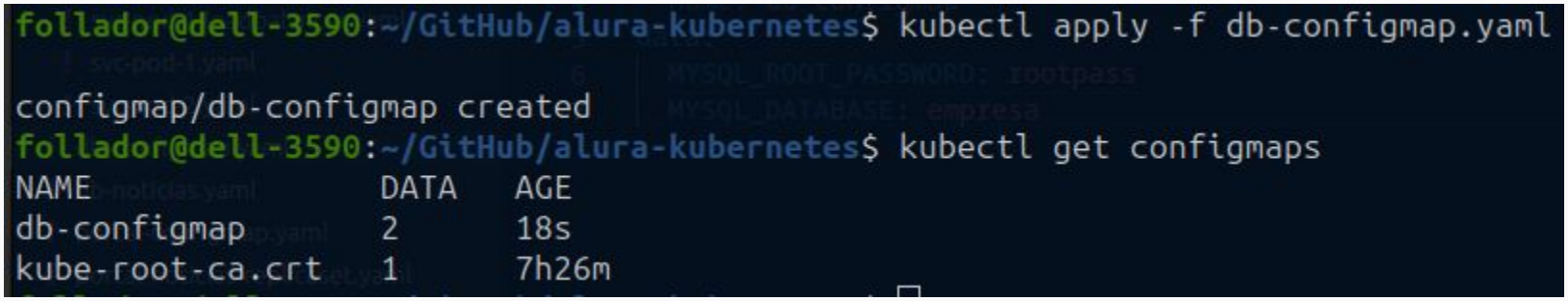


Montando o primeiro cluster
passo 5 - criando ConfigMap para a base de dados

```
# db-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: db-configmap
data:
  MYSQL_ROOT_PASSWORD: minhaSenha
  MYSQL_DATABASE: empresa
```

passo 5 - criando ConfigMap para a base de dados

kubectl apply -f db-configmap.yaml
kubectl get configmaps



Montando o primeiro cluster
passo 6 - criando Pod da base de dados

```
# db-noticias.yaml
apiVersion: v1
kind: Pod
metadata:
  name: db-noticias
  labels:
    app: db-noticias
spec:
  containers:
    - name: db-noticias-container
      image: aluracursos/mysql-db:1
      ports:
        - containerPort: 3306
      envFrom:
        - configMapRef:
            name: db-configmap
```

Montando o primeiro cluster
passo 6 - criando Pod da base de dados

kubectl apply -f db-noticias.yaml
kubectl get pods

```
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f db-noticias.yaml
pod/db-noticias created
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get pods
NAME                READY   STATUS    RESTARTS      AGE
db-noticias         1/1     Running   0             9s
portal-noticias     1/1     Running   1 (22m ago)   5h58m
sistema-noticias    1/1     Running   1 (22m ago)   5h51m
```

Montando o primeiro cluster
passo 7 - criando Service da base de dados

```
# svc-db-noticias.yaml
apiVersion: v1
kind: Service
metadata:
  name: svc-db-noticias
spec:
  type: ClusterIP
  ports:
    - port: 3306
  selector:
    app: db-noticias
```

passo 7 - criando Service da base de dados

kubectl apply -f svc-db-noticias.yaml

```
folladorr@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f svc-db-noticias.yaml
service/svc-db-noticias created
```

Montando o primeiro cluster
passo 8 - criando ConfigMap para o sistema

```
# sistema-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: sistema-configmap
data:
  HOST_DB: svc-db-noticias:3306
  USER_DB: root
  PASS_DB: minhaSenha
  DATABASE_DB: empresa
```

passo 8 - adicionando ConfigMap no Pod do sistema

```
# sistema-noticias.yaml
apiVersion: v1
kind: Pod
metadata:
  name: sistema-noticias
  labels:
    app: sistema-noticias
spec:
  containers:
    - name: sistema-noticias-container
      image: aluracursos/sistema-noticias:1
      ports:
        - containerPort: 80
      envFrom:
        - configMapRef:
            name: sistema-configmap
```

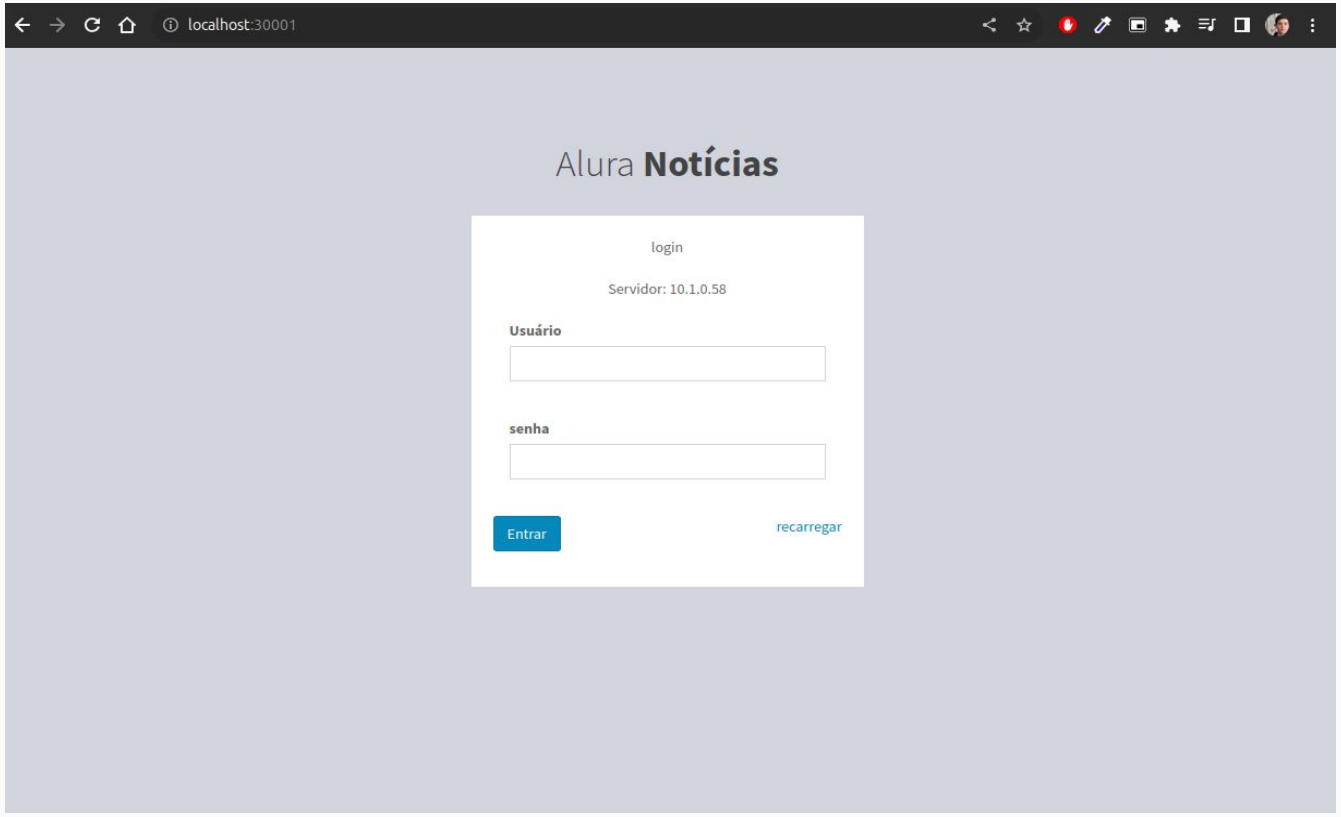
passo 8 - adicionando ConfigMap no Pod do sistema

kubectl apply -f sistema-configmap.yaml
kubectl delete pod sistema-noticias
kubectl apply -f sistema-noticias.yaml

```
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f sistema-configmap.yaml
configmap/sistema-configmap created
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl delete pod sistema-noticias
pod "sistema-noticias" deleted
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f sistema-noticias.yaml
pod/sistema-noticias created
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get pods
NAME               READY   STATUS    RESTARTS      AGE
db-noticias        1/1     Running   0             15m
portal-noticias    1/1     Running   1 (37m ago)   6h13m
sistema-noticias   1/1     Running   0             7s
```

Montando o primeiro cluster
passo 8 - adicionando ConfigMap no Pod do sistema

kubectl apply -f sistema-configmap.yaml
kubectl delete pod sistema-noticias
kubectl apply -f sistema-noticias.yaml



Montando o primeiro cluster
passo 9 - criando ConfigMap para o portal

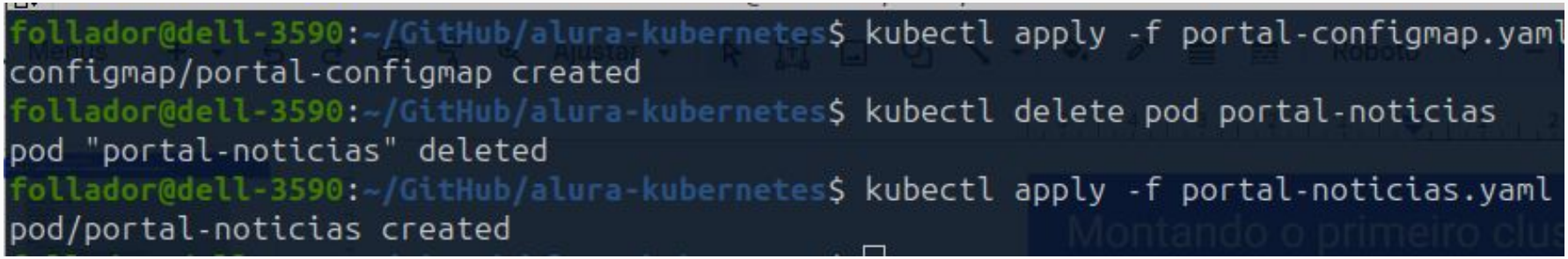
```
# portal-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: portal-configmap
data:
  IP_SISTEMA: http://localhost:30001
```

Montando o primeiro cluster
passo 9 - adicionando ConfigMap no Pod do portal

```
# portal-noticias.yaml
apiVersion: v1
kind: Pod
metadata:
  name: portal-noticias
  labels:
    app: portal-noticias
spec:
  containers:
    - name: portal-noticias-container
      image: aluracursos/portal-noticias:1
      ports:
        - containerPort: 80
      envFrom:
        - configMapRef:
            name: portal-configmap
```

passo 9 - adicionando ConfigMap no Pod do sistema

kubectl apply -f portal-configmap.yaml
kubectl delete pod portal-noticias
kubectl apply -f portal-noticias.yaml



passo 10 - criando ReplicaSet para o sistema

```
# sistema-noticias-replicaset.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: sistema-noticias-replicaset
spec:
  template:
    metadata:
      name: sistema-noticias
      labels:
        app: sistema-noticias
    spec:
      containers:
        - name: sistema-noticias-container
          image: aluracursos/sistema-noticias:1
          ports:
            - containerPort: 80
          envFrom:
            - configMapRef:
                name: sistema-configmap
  replicas: 3
  selector:
    matchLabels:
      app: sistema-noticias
```

Montando o primeiro cluster
passo 10 - criando ReplicaSet para o sistema

kubectl apply -f sistema-noticias-replicaset.yaml
kubectl get pods
kubectl get replicasets

```
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f sistema-noticias-replicaset.y
aml
replicaset.apps/sistema-noticias-replicaset created
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
db-noticias                         1/1     Running             0          39m
portal-noticias                     1/1     Running             0          4m
sistema-noticias                    1/1     Running             0          24m
sistema-noticias-replicaset-7xtvp   0/1     ContainerCreating   0          5s
sistema-noticias-replicaset-xtsfg   0/1     ContainerCreating   0          5s
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get replicasets
NAME                          DESIRED   CURRENT   READY   AGE
sistema-noticias-replicaset   3         3         3       20s
```

Montando o primeiro cluster
passo 11 - criando ReplicaSet para o portal

```yaml
# portal-noticias-replicaset.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: portal-noticias-replicaset
spec:
  template:
    metadata:
      name: portal-noticias
      labels:
        app: portal-noticias
    spec:
      containers:
        - name: portal-noticias-container
          image: aluracursos/portal-noticias:1
          ports:
            - containerPort: 80
          envFrom:
            - configMapRef:
                name: portal-configmap
  replicas: 3
  selector:
    matchLabels:
      app: portal-noticias
```

Montando o primeiro cluster
passo 10 - criando ReplicaSet para o sistema

kubectl apply -f portal-noticias-replicaset.yaml
kubectl get pods
kubectl get rs

```
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl apply -f portal-noticias-replicaset.ya
ml
replicaset.apps/portal-noticias-replicaset created
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
db-noticias                         1/1     Running             0          41m
portal-noticias                     1/1     Running             0          6m34s
portal-noticias-replicaset-9kwws    0/1     ContainerCreating   0          5s
portal-noticias-replicaset-rfjdd    0/1     ContainerCreating   0          4s
sistema-noticias                    1/1     Running             0          26m
sistema-noticias-replicaset-7xtvp   1/1     Running             0          2m39s
sistema-noticias-replicaset-xtsfg   1/1     Running             0          2m39s
follador@dell-3590:~/GitHub/alura-kubernetes$ kubectl get rs
NAME                          DESIRED   CURRENT   READY   AGE
portal-noticias-replicaset    3         3         3       13s
sistema-noticias-replicaset   3         3         3       2m48s
```