Roteiro da implantação e teste

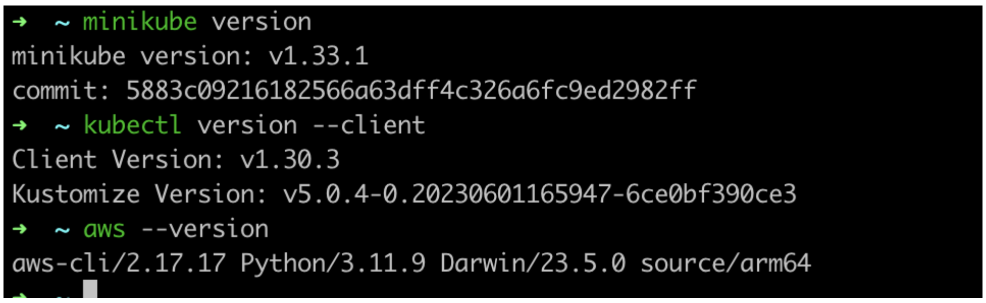
As características do computador usado para realizar a implantação e teste do procedimento detalhado neste documento foram:

- Modelo do CPU: Apple M1
- Número de cores: 8
- Memória RAM: 16 Gb

1. Pré-requisitos

Clone o repositório do GitHub com os arquivos que serão usados para a implantação.
Instalar os seguintes paquetes:

- minikube
- kubectl
- aws-cli

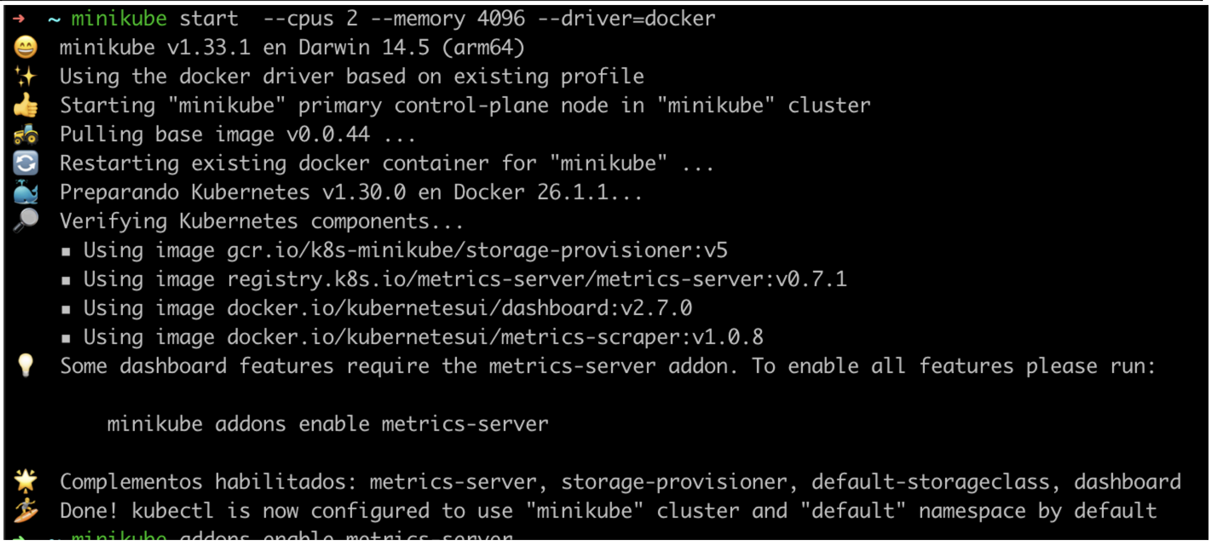


```
➜  ~ minikube version
minikube version: v1.33.1
commit: 5883c09216182566a63dff4c326a6fc9ed2982ff
➜  ~ kubectl version --client
Client Version: v1.30.3
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
➜  ~ aws --version
aws-cli/2.17.17 Python/3.11.9 Darwin/23.5.0 source/arm64
```

2. Cluster Kubernetes usando minikube

Iniciamos o Minikbe

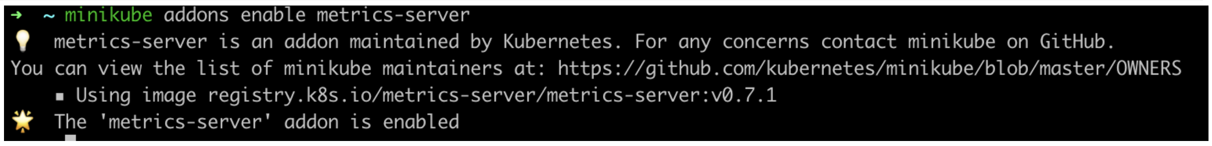
```
$  minikube start  --cpus 2 --memory 4096 --driver=docker
```



```
➜  ~ minikube start  --cpus 2 --memory 4096 --driver=docker
😄  minikube v1.33.1 en Darwin 14.5 (arm64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.44 ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparando Kubernetes v1.30.0 en Docker 26.1.1...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
    ▪ Using image registry.k8s.io/metrics-server/metrics-server:v0.7.1
    ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
    ▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
💡  Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server

🌟  Complementos habilitados: metrics-server, storage-provisioner, default-storageclass, dashboard
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
➜  ~ minikube addons enable metrics-server
```

Habilitamos as métricas

```
$  minikube addons enable metrics-server
```



```
➜  ~ minikube addons enable metrics-server
💡  metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
    ▪ Using image registry.k8s.io/metrics-server/metrics-server:v0.7.1
🌟  The 'metrics-server' addon is enabled
```

Criamos um deployment usando o arquivo deployment.yaml

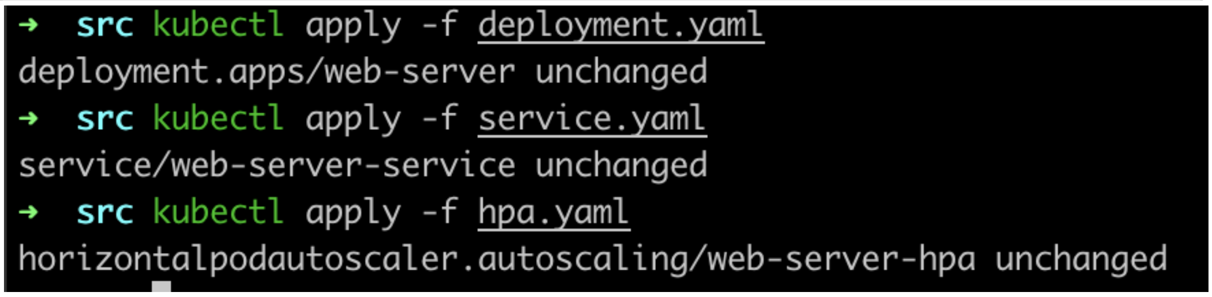
```
$  kubectl apply -f deployment.yaml
```

Exporemos o serviço usando o arquivo service.yaml

```
$  kubectl apply -f service.yaml
```

Configurar o autoescalamento horizontal usando o arquivo hpa.yaml

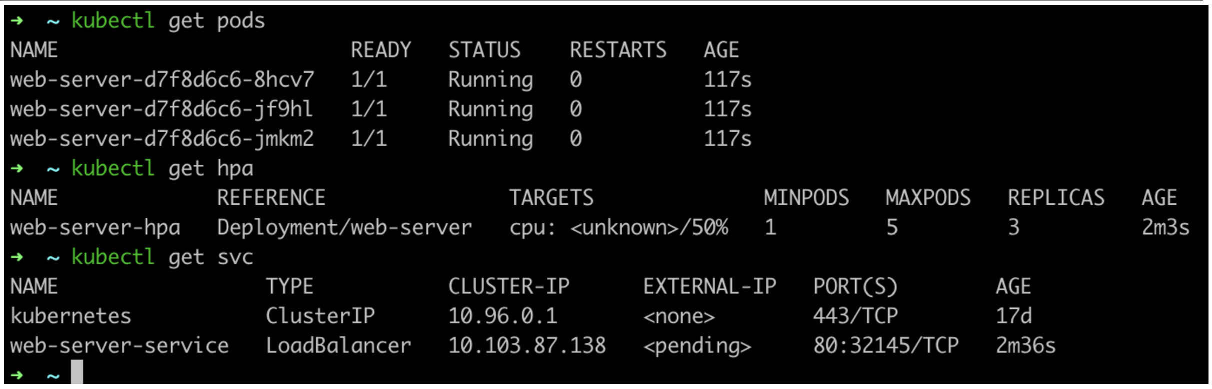
```
$  kubectl apply -f hpa.yaml
```



```
➜  src kubectl apply -f deployment.yaml
deployment.apps/web-server unchanged
➜  src kubectl apply -f service.yaml
service/web-server-service unchanged
➜  src kubectl apply -f hpa.yaml
horizontalpodautoscaler.autoscaling/web-server-hpa unchanged
```

Verificamos o deployment

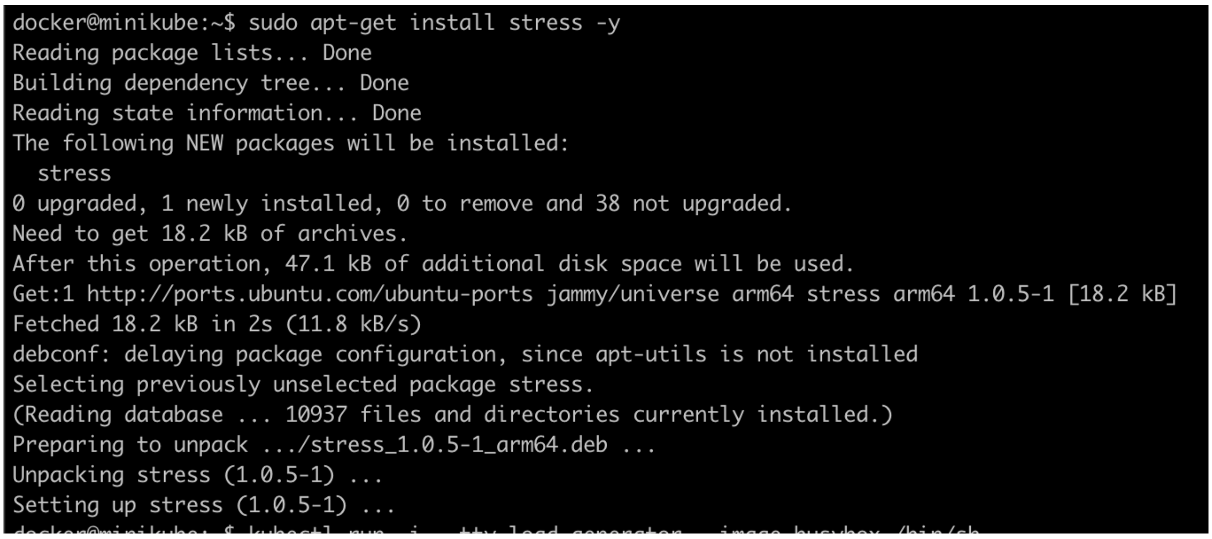
```
$  kubectl get pods
$  kubectl get svc
$  kubectl get hpa
```



```
➜  ~ kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
web-server-d7f8d6c6-8hcv7    1/1     Running   0          117s
web-server-d7f8d6c6-jf9hl    1/1     Running   0          117s
web-server-d7f8d6c6-jmkm2    1/1     Running   0          117s
➜  ~ kubectl get hpa
NAME             REFERENCE                TARGETS             MINPODS   MAXPODS   REPLICAS   AGE
web-server-hpa   Deployment/web-server    cpu: <unknown>/50%   1         5         3          2m3s
➜  ~ kubectl get svc
NAME                 TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)         AGE
kubernetes           ClusterIP      10.96.0.1       <none>        443/TCP         17d
web-server-service   LoadBalancer   10.103.87.138   <pending>     80:32145/TCP    2m36s
➜  ~
```

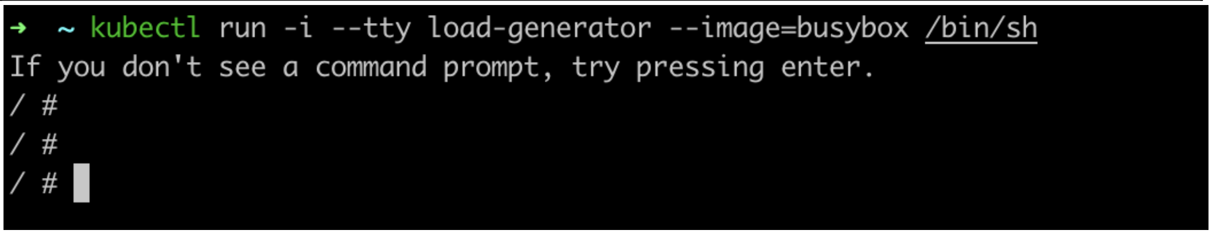
Geramos carga no cluster para testar o HPA

```
$  minikube ssh
docker@minikube:~$ sudo apt-get update
docker@minikube:~$ sudo apt-get install stress -y
```


```
docker@minikube:~$ sudo apt-get install stress -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  stress
0 upgraded, 1 newly installed, 0 to remove and 38 not upgraded.
Need to get 18.2 kB of archives.
After this operation, 47.1 kB of additional disk space will be used.
Get:1 http://ports.ubuntu.com/ubuntu-ports jammy/universe arm64 stress arm64 1.0.5-1 [18.2 kB]
Fetched 18.2 kB in 2s (11.8 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package stress.
(Reading database ... 10937 files and directories currently installed.)
Preparing to unpack .../stress_1.0.5-1_arm64.deb ...
Unpacking stress (1.0.5-1) ...
Setting up stress (1.0.5-1) ...
```

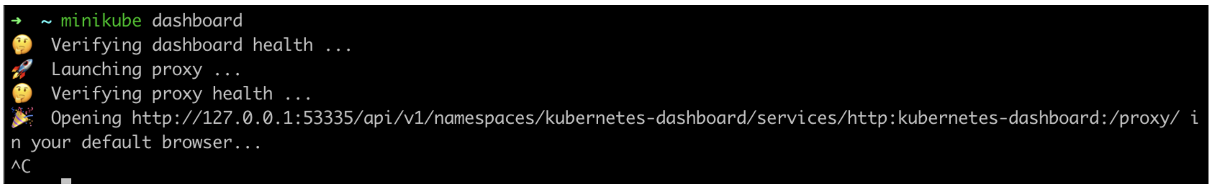
Geramos carga no cluster para testar o HPA

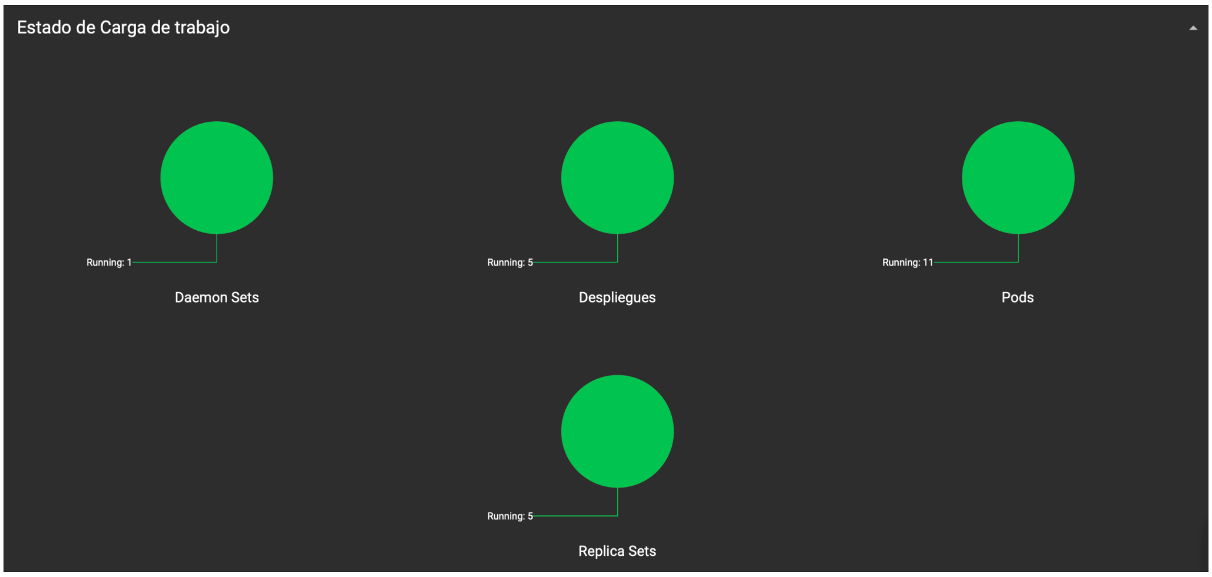
```
$  kubectl run -i --tty load-generator --image=busybox /bin/sh
#  while sleep 0.01; do wget -q -O- http://web-server-service; done
```

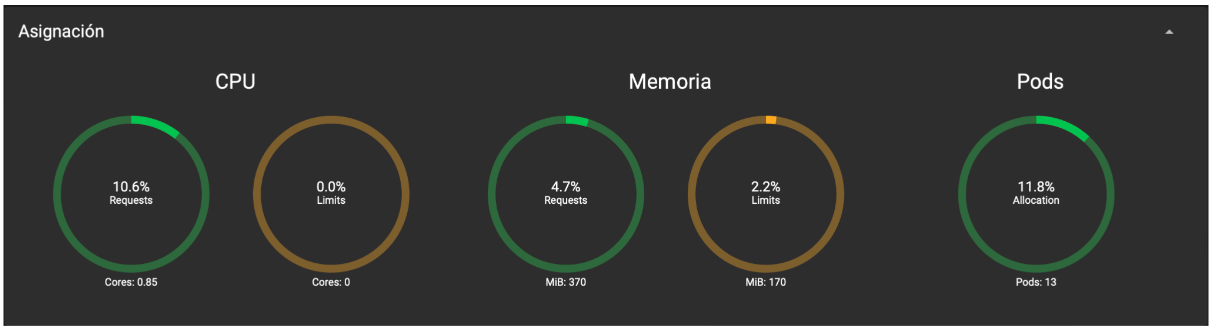

```
➜  ~ kubectl run -i --tty load-generator --image=busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
/ #
/ #
```

Verificamos o autoescalamento horizontal no dashboard

```
$  minikube dashboard
```


```
➜  ~ minikube dashboard
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:53335/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ i
n your default browser...
^C
```

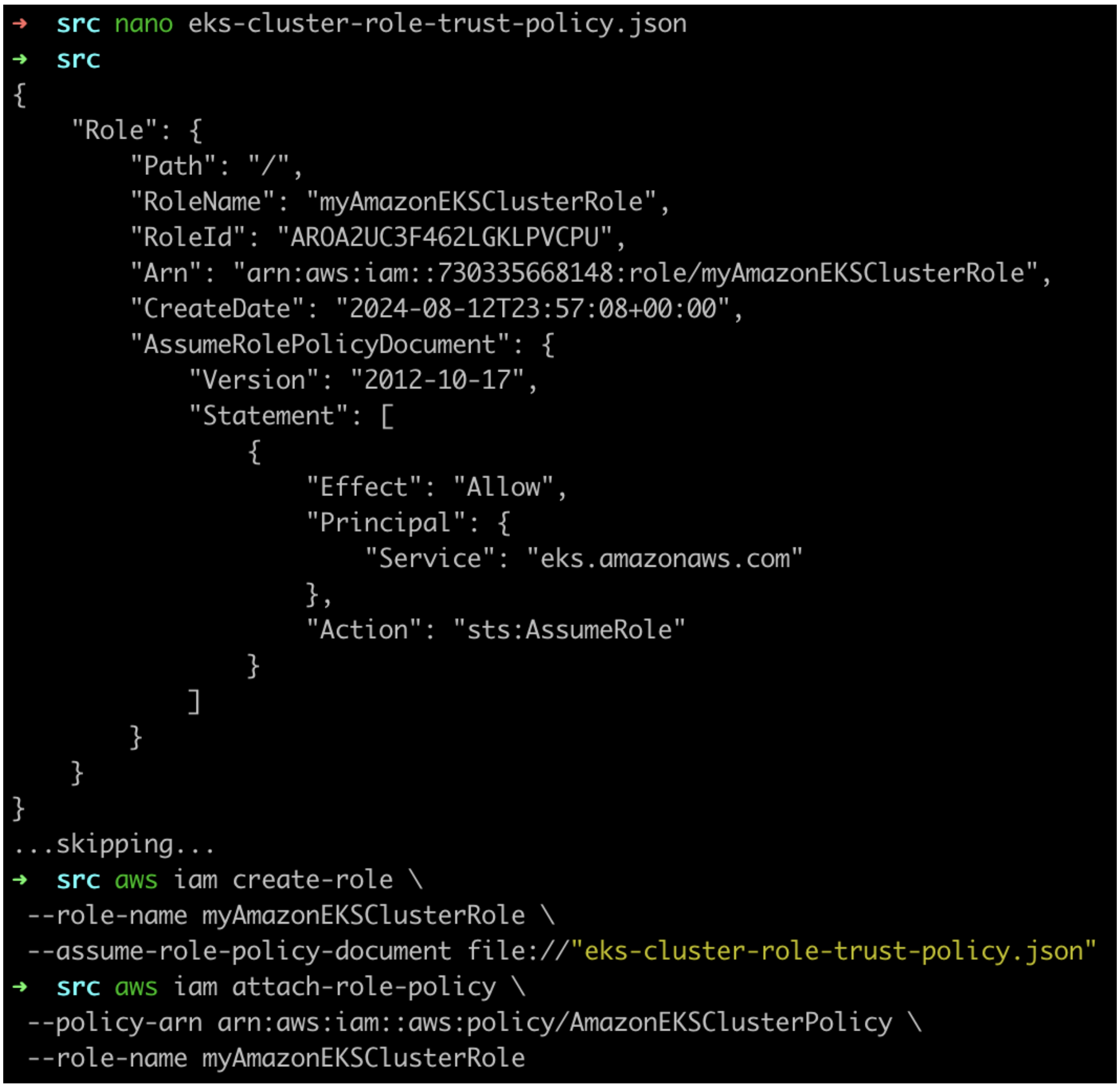




3. Cluster AWS EKS

Criar o perfil IAM com as políticas de gerenciamento. Usar o arquivo eks-cluster-role-trust-policy.json e executar os seguintes comandos

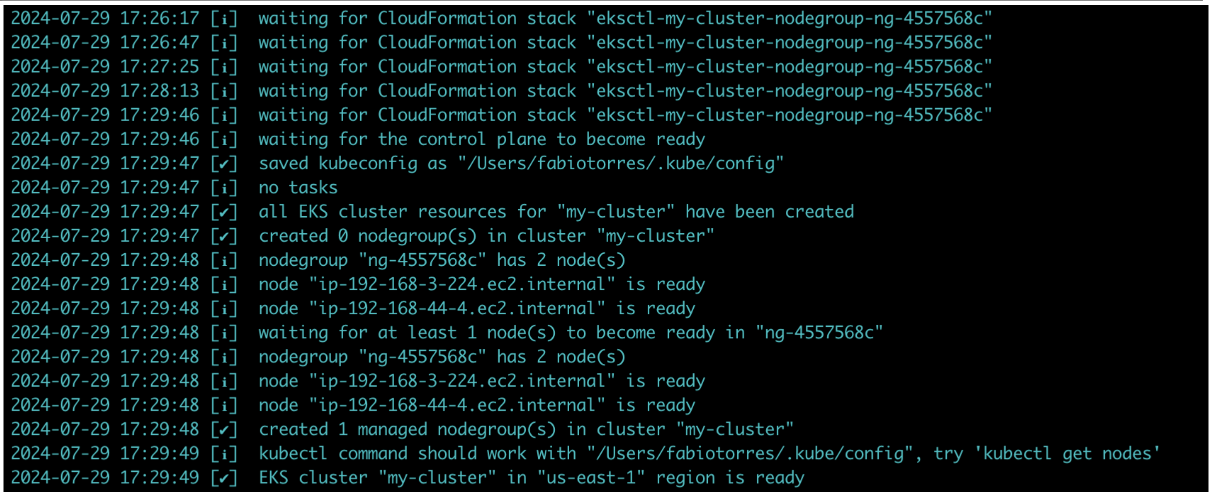
```
$ aws iam create-role \
 --role-name myAmazonEKSClusterRole \
 --assume-role-policy-document file://"eks-cluster-role-trust-policy.json"

$ aws iam attach-role-policy \
 --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy \
 --role-name myAmazonEKSClusterRole
```



```
➜  src nano eks-cluster-role-trust-policy.json
➜  src
{
    "Role": {
        "Path": "/",
        "RoleName": "myAmazonEKSClusterRole",
        "RoleId": "AROA2UC3F462LGKLPVCPU",
        "Arn": "arn:aws:iam::730335668148:role/myAmazonEKSClusterRole",
        "CreateDate": "2024-08-12T23:57:08+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "eks.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
...skipping...
➜  src aws iam create-role \
 --role-name myAmazonEKSClusterRole \
 --assume-role-policy-document file://"eks-cluster-role-trust-policy.json"
➜  src aws iam attach-role-policy \
 --policy-arn arn:aws:iam::aws:policy/AmazonEKSClusterPolicy \
 --role-name myAmazonEKSClusterRole
```

Criamos o cluster e nós no AWS EKS usando o seguinte comando

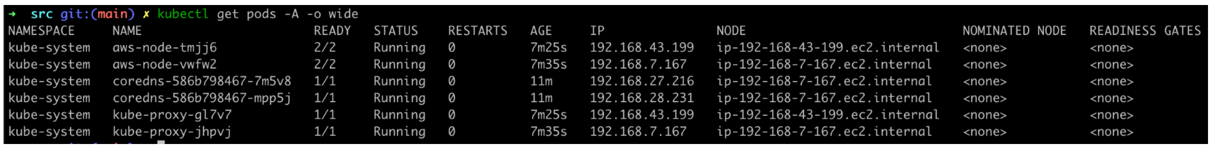
```
$ aws eks update-kubeconfig --region us-east-1 --name my-cluster
```



```
2024-07-29 17:26:17 [i]  waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-ng-4557568c"
2024-07-29 17:26:47 [i]  waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-ng-4557568c"
2024-07-29 17:27:25 [i]  waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-ng-4557568c"
2024-07-29 17:28:13 [i]  waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-ng-4557568c"
2024-07-29 17:29:46 [i]  waiting for CloudFormation stack "eksctl-my-cluster-nodegroup-ng-4557568c"
2024-07-29 17:29:46 [i]  waiting for the control plane to become ready
2024-07-29 17:29:47 [✔]  saved kubeconfig as "/Users/fabiotorres/.kube/config"
2024-07-29 17:29:47 [i]  no tasks
2024-07-29 17:29:47 [✔]  all EKS cluster resources for "my-cluster" have been created
2024-07-29 17:29:47 [✔]  created 0 nodegroup(s) in cluster "my-cluster"
2024-07-29 17:29:48 [i]  nodegroup "ng-4557568c" has 2 node(s)
2024-07-29 17:29:48 [i]  node "ip-192-168-3-224.ec2.internal" is ready
2024-07-29 17:29:48 [i]  node "ip-192-168-44-4.ec2.internal" is ready
2024-07-29 17:29:48 [i]  waiting for at least 1 node(s) to become ready in "ng-4557568c"
2024-07-29 17:29:48 [i]  nodegroup "ng-4557568c" has 2 node(s)
2024-07-29 17:29:48 [i]  node "ip-192-168-3-224.ec2.internal" is ready
2024-07-29 17:29:48 [i]  node "ip-192-168-44-4.ec2.internal" is ready
2024-07-29 17:29:48 [✔]  created 1 managed nodegroup(s) in cluster "my-cluster"
2024-07-29 17:29:49 [i]  kubectl command should work with "/Users/fabiotorres/.kube/config", try 'kubectl get nodes'
2024-07-29 17:29:49 [✔]  EKS cluster "my-cluster" in "us-east-1" region is ready
```

Verificar os pods do EKS criado

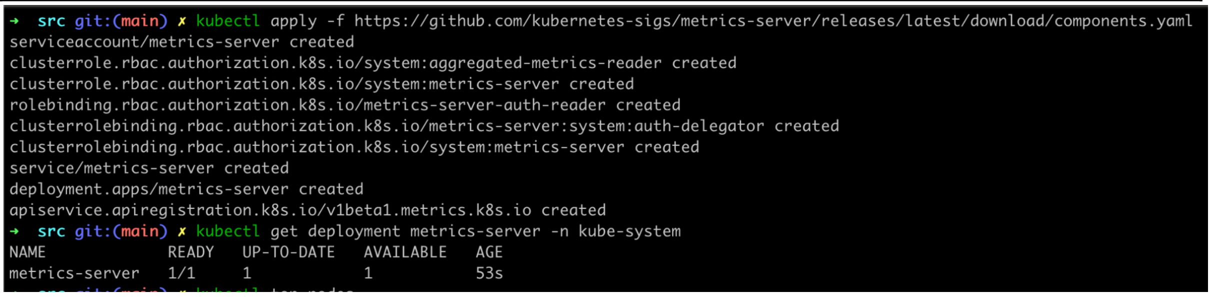
```
$ kubectl get pods -A -o wide
```



```
➜  src git:(main) ✗ kubectl get pods -A -o wide
NAMESPACE     NAME                       READY   STATUS    RESTARTS   AGE     IP               NODE                             NOMINATED NODE   READINESS GATES
kube-system   aws-node-tmjj6             2/2     Running   0          7m25s   192.168.43.199   ip-192-168-43-199.ec2.internal   <none>           <none>
kube-system   aws-node-vwfw2             2/2     Running   0          7m35s   192.168.7.167    ip-192-168-7-167.ec2.internal    <none>           <none>
kube-system   coredns-586b798467-7m5v8   1/1     Running   0          11m     192.168.27.216   ip-192-168-7-167.ec2.internal    <none>           <none>
kube-system   coredns-586b798467-mpp5j   1/1     Running   0          11m     192.168.28.231   ip-192-168-7-167.ec2.internal    <none>           <none>
kube-system   kube-proxy-gl7v7           1/1     Running   0          7m25s   192.168.43.199   ip-192-168-43-199.ec2.internal   <none>           <none>
kube-system   kube-proxy-jhpvj           1/1     Running   0          7m35s   192.168.7.167    ip-192-168-7-167.ec2.internal    <none>           <none>
```

Visualizar o uso de recursos com o Kubernetes Metrics Server

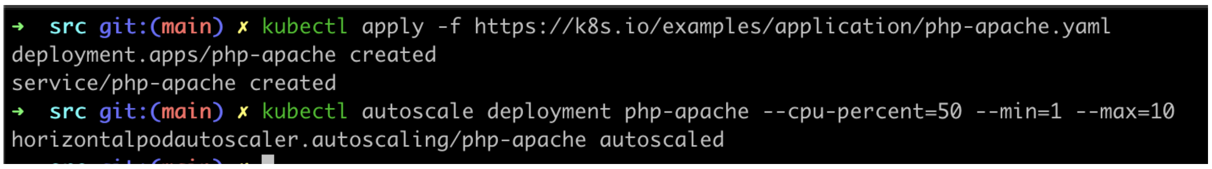
```
$  kubectl apply -f
https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/compon
ents.yaml
$  kubectl get deployment metrics-server -n kube-system
```



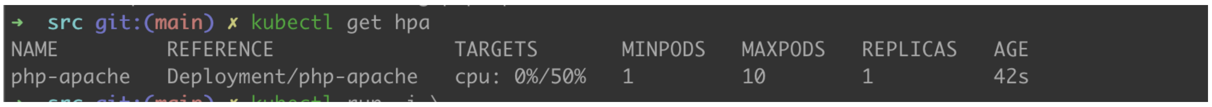
```
➜  src git:(main) ✗ kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
➜  src git:(main) ✗ kubectl get deployment metrics-server -n kube-system
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
metrics-server   1/1     1            1           53s
```

Configurar o Horizontal Pod Autoscaler

```
$  kubectl apply -f https://k8s.io/examples/application/php-apache.yaml
$  kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
$  kubectl get hpa
```



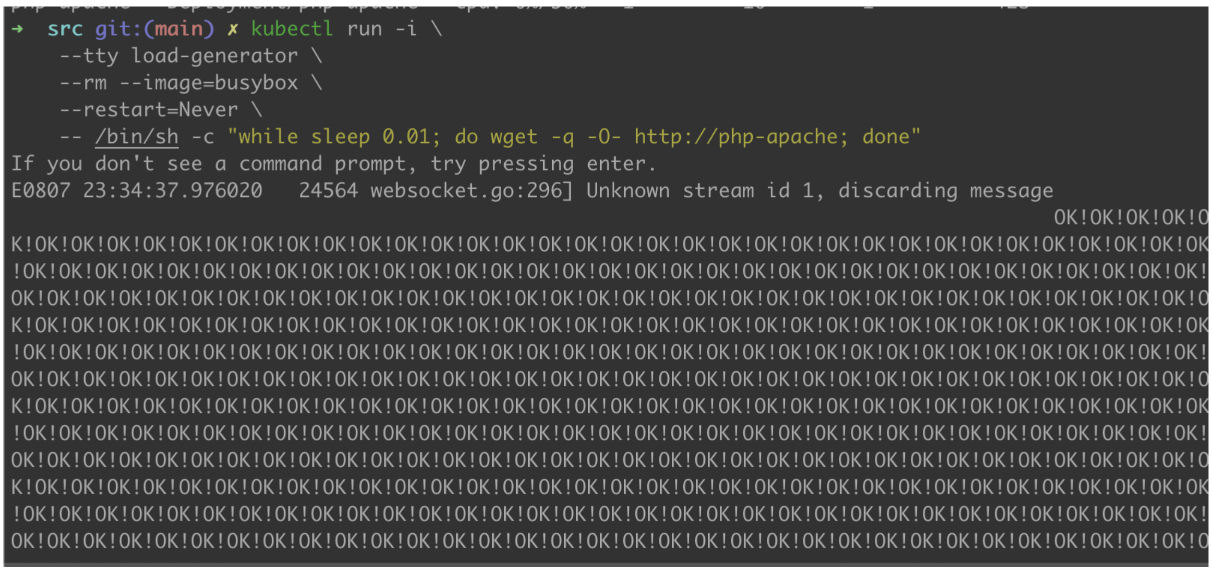
```
➜  src git:(main) ✗ kubectl apply -f https://k8s.io/examples/application/php-apache.yaml
deployment.apps/php-apache created
service/php-apache created
➜  src git:(main) ✗ kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
horizontalpodautoscaler.autoscaling/php-apache autoscaled
```



```
➜  src git:(main) ✗ kubectl get hpa
NAME         REFERENCE               TARGETS     MINPODS   MAXPODS   REPLICAS   AGE
php-apache   Deployment/php-apache   cpu: 0%/50%   1         10        1          42s
```

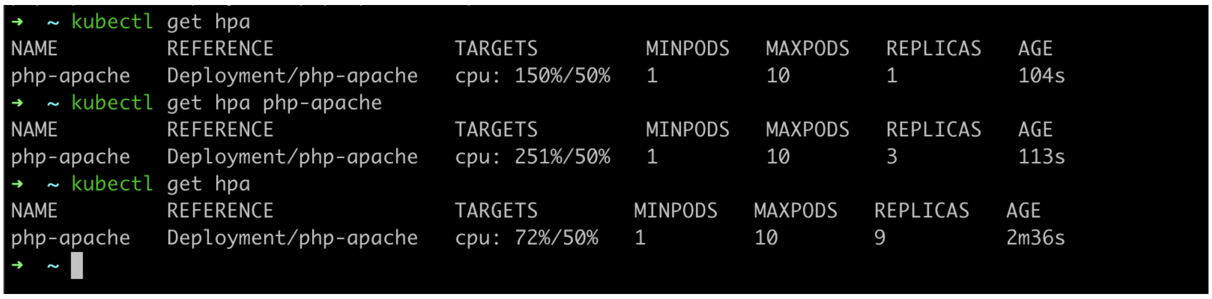
Teste do autoescalamento

```
$  kubectl run -i \
    --tty load-generator \
    --rm --image=busybox \
    --restart=Never \
    -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache; done"
```



```
➜  src git:(main) ✗ kubectl run -i \
    --tty load-generator \
    --rm --image=busybox \
    --restart=Never \
    -- /bin/sh -c "while sleep 0.01; do wget -q -O- http://php-apache; done"
If you don't see a command prompt, try pressing enter.
E0807 23:34:37.976020   24564 websocket.go:296] Unknown stream id 1, discarding message
                                                                                       OK!OK!OK!OK!O
K!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK
!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!
OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!O
K!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK
!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!
OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!O
K!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK
!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!
OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!O
K!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK
!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!
OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!OK!O
```

Após alguns segundos verificamos o número de réplicas feitas pelo HPA



```
➜  ~ kubectl get hpa
NAME         REFERENCE               TARGETS         MINPODS    MAXPODS    REPLICAS    AGE
php-apache   Deployment/php-apache   cpu: 150%/50%   1          10         1           104s
➜  ~ kubectl get hpa php-apache
NAME         REFERENCE               TARGETS         MINPODS    MAXPODS    REPLICAS    AGE
php-apache   Deployment/php-apache   cpu: 251%/50%   1          10         3           113s
➜  ~ kubectl get hpa
NAME         REFERENCE               TARGETS        MINPODS    MAXPODS    REPLICAS    AGE
php-apache   Deployment/php-apache   cpu: 72%/50%   1          10         9           2m36s
➜  ~ 
```