


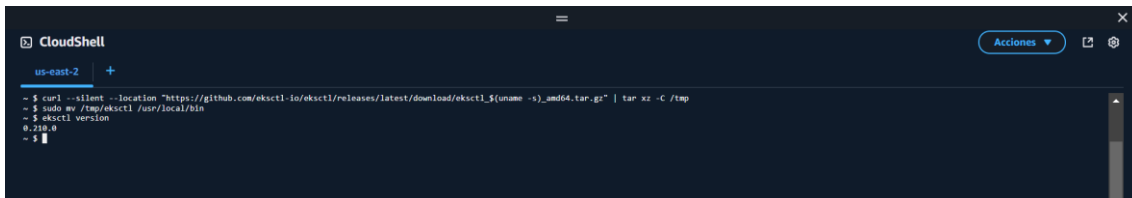
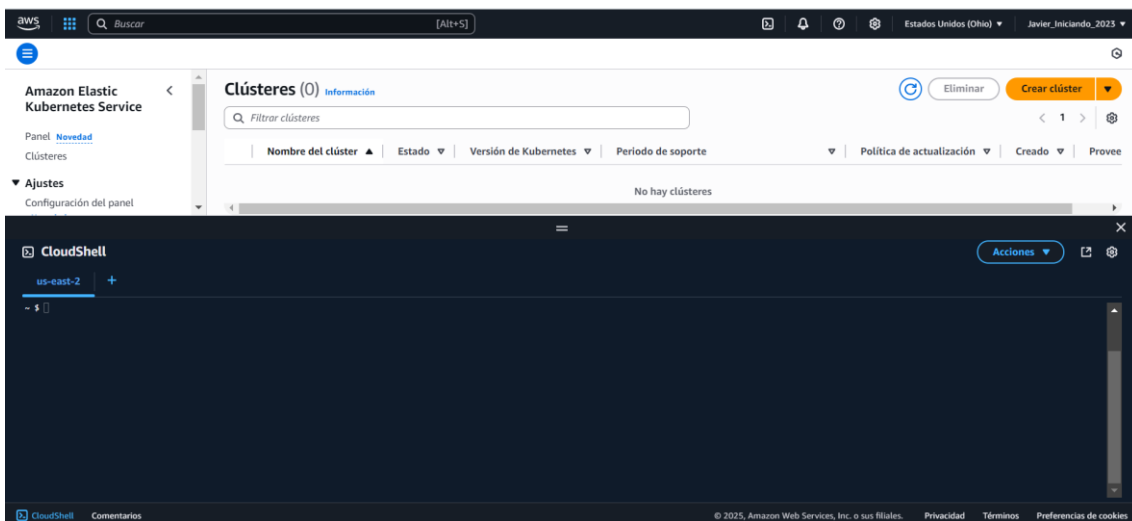
## Guía de Instalación en AWS EKS


 **Paso 1:** Ingresar al cloudShell de la consola de AWS e instalar el eksctl

```
```bash
```

```
curl --silent --location "https://github.com/eksctl-  
io/eksctl/releases/latest/download/eksctl_${uname -s}_amd64.tar.gz" | tar xz -C /tmp  
sudo mv /tmp/eksctl /usr/local/bin  
eksctl version
```

```
```
```



 **Paso 2:** Crear el clúster EKS (Puede tardar varios minutos en crear e iniciar)

```
```bash
```

```
eksctl create cluster --name myEKSCluster --region us-east-2 --zones us-east-2a,us-east-2b --  
without-nodegroup
```

```
```
```

```
CloudShell
us-east-2 +
$ curl --silent --location "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl_${uname -s}_amd64.tar.gz" | tar xz -C /tmp
$ sudo mv /tmp/eksctl /usr/local/bin
$ eksctl version
0.218.0
$ eksctl create cluster --name myEKSCluster --region us-east-2 --zones us-east-2a,us-east-2b --without-nodegroup
2025-06-28 19:18:24 [i] eksctl version 0.218.0
2025-06-28 19:18:24 [i] using region us-east-2
2025-06-28 19:18:24 [i] subnets for us-east-2a - public:102.108.0.0/19 private:102.108.04.0/19
2025-06-28 19:18:24 [i] subnets for us-east-2b - public:102.108.20.0/19 private:102.108.06.0/19
2025-06-28 19:18:24 [i] using kubernetes version 1.32
2025-06-28 19:18:24 [i] creating EKS cluster "myEKSCluster" in "us-east-2" region with:
2025-06-28 19:18:24 [i] If you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-2 --cluster=myEKSCluster'
2025-06-28 19:18:24 [i] Kubernetes API endpoint ARNs will use default of (publicAccess=true, privateAccess=false) for cluster "myEKSCluster" in "us-east-2"
2025-06-28 19:18:24 [i] Cloudwatch logging will not be enabled for cluster "myEKSCluster" in "us-east-2"
2025-06-28 19:18:24 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types=SPECIFY-VPC-LOG-TYPES=HERE (e.g. all) --region=us-east-2 --cluster=myEKSCluster'
2025-06-28 19:18:24 [i] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-06-28 19:18:24 [i]
2 sequential tasks:
1 create cluster control plane "myEKSCluster",
2 sequential sub-tasks:
1 task: create addons,
wait for control plane to become ready,
}
2025-06-28 19:18:24 [i] building cluster stack "eksctl-myEKSCluster-cluster"
2025-06-28 19:18:24 [i] deploying stack "eksctl-myEKSCluster-cluster"
2025-06-28 19:18:54 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-cluster"
```

```
CloudShell
us-east-2 +
2025-06-28 19:19:04 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-cluster"
2025-06-28 19:19:24 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-cluster"
2025-06-28 19:19:25 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-cluster"
2025-06-28 19:19:25 [i] Recommended policies were found for "vpc-cni" addon, but since OADK is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.podIdentityAssociations', and run 'eksctl update addon'
2025-06-28 19:19:25 [i]
2025-06-28 19:19:25 [i] successfully created addon: vpc-cni
2025-06-28 19:19:26 [i] creating addon: kube-proxy
2025-06-28 19:19:26 [i] successfully created addon: kube-proxy
2025-06-28 19:19:26 [i] creating addon: coredns
2025-06-28 19:19:26 [i] successfully created addon: coredns
2025-06-28 19:19:27 [i] creating addon: metrics-server
2025-06-28 19:19:27 [i] successfully created addon: metrics-server
2025-06-28 19:19:27 [i] waiting for the control plane to become ready
2025-06-28 19:19:28 [i] saved kubeconfig as "/home/cloudshell-user/.kube/config"
2025-06-28 19:19:28 [i] no tasks
2025-06-28 19:19:28 [i] All EKS cluster resources for "myEKSCluster" have been created
2025-06-28 19:19:28 [i] Kubectl command should work with "/home/cloudshell-user/.kube/config", try 'kubectl get nodes'
2025-06-28 19:19:28 [i] EKS cluster "myEKSCluster" in "us-east-2" region is ready
```

### Paso 3: Crear los nodos asociados al clúster EKS

```bash

```
eksctl create nodegroup \
--cluster myEKSCluster \
--region us-east-2 \
--name myNodeGroup \
--node-type t3.medium \
--nodes 2 \
--nodes-min 1 \
--nodes-max 3 \
--managed
```

```

```
CloudShell
us-east-2 +
2025-06-28 19:19:40 [i] successfully created addon: kube-proxy
2025-06-28 19:19:40 [i] creating addon: coredns
2025-06-28 19:19:40 [i] successfully created addon: coredns
2025-06-28 19:19:41 [i] creating addon: metrics-server
2025-06-28 19:19:41 [i] successfully created addon: metrics-server
2025-06-28 19:19:42 [i] waiting for the control plane to become ready
2025-06-28 19:19:42 [i] saved kubeconfig as "/home/cloudshell-user/.kube/config"
2025-06-28 19:19:43 [i] no tasks
2025-06-28 19:19:43 [i] All EKS cluster resources for "myEKSCluster" have been created
2025-06-28 19:19:43 [i] Kubectl command should work with "/home/cloudshell-user/.kube/config", try 'kubectl get nodes'
2025-06-28 19:19:43 [i] EKS cluster "myEKSCluster" in "us-east-2" region is ready
$ eksctl create nodegroup
> --cluster myEKSCluster \
> --region us-east-2 \
> --name myNodeGroup \
> --node-type t3.medium \
> --nodes 2 \
> --nodes-min 1 \
> --nodes-max 3 \
> --managed
```

```
CloudShell
us-east-2 +
2025-06-28 19:21:29 [i] 1 nodegroup (myNodegroup) was included (based on the include/exclude rules)
2025-06-28 19:21:29 [i] All create a CloudFormation stack for each of 1 managed nodegroup in cluster "myEKSCluster"
2025-06-28 19:21:29 [i] Is cluster compatibility, 1 task: { 1 task: { create managed nodegroup "myNodegroup" } }
2025-06-28 19:21:29 [i] checking cluster stack for missing resources
2025-06-28 19:21:29 [i] cluster stack has all required resources
2025-06-28 19:21:29 [i] building managed nodegroup stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] deploying stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] no tasks
2025-06-28 19:21:29 [i] created 0 nodegroup(s) in cluster "myEKSCluster"
2025-06-28 19:21:29 [i] nodegroup "myNodegroup" has 2 node(s)
2025-06-28 19:21:29 [i] node "ip-102-168-53-214.us-east-2.compute.internal" is ready
2025-06-28 19:21:29 [i] node "ip-102-168-9-199.us-east-2.compute.internal" is ready
2025-06-28 19:21:29 [i] waiting for at least 1 node(s) to become ready in "myNodegroup"
2025-06-28 19:21:29 [i] nodegroup "myNodegroup" has 2 node(s)
2025-06-28 19:21:29 [i] node "ip-102-168-53-214.us-east-2.compute.internal" is ready
2025-06-28 19:21:29 [i] node "ip-102-168-9-199.us-east-2.compute.internal" is ready
2025-06-28 19:21:29 [i] created 1 managed nodegroup(s) in cluster "myEKSCluster"
2025-06-28 19:21:29 [i] checking security group configuration for all nodegroups
2025-06-28 19:21:29 [i] all nodegroups have up-to-date CloudFormation templates
2025-06-28 19:21:29 [i]
```

#### Paso 4: Obtener las credenciales del clúster EKS

```
""bash
```

```
aws eks update-kubeconfig --name myEKSCluster --region us-east-2
```

```
""
```

```
CloudShell
us-east-2 +
2025-06-28 19:21:29 [i] fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "myNodegroup" } }
2025-06-28 19:21:29 [i] checking cluster stack for missing resources
2025-06-28 19:21:29 [i] cluster stack has all required resources
2025-06-28 19:21:29 [i] building managed nodegroup stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] deploying stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] waiting for CloudFormation stack "eksctl-myEKSCluster-nodegroup-myNodegroup"
2025-06-28 19:21:29 [i] no tasks
2025-06-28 19:21:29 [i] created 0 nodegroup(s) in cluster "myEKSCluster"
2025-06-28 19:21:29 [i] nodegroup "myNodegroup" has 2 node(s)
2025-06-28 19:21:29 [i] node "ip-102-168-53-214.us-east-2.compute.internal" is ready
2025-06-28 19:21:29 [i] node "ip-102-168-9-199.us-east-2.compute.internal" is ready
2025-06-28 19:21:29 [i] waiting for at least 1 node(s) to become ready in "myNodegroup"
2025-06-28 19:21:29 [i] nodegroup "myNodegroup" has 2 node(s)
2025-06-28 19:21:29 [i] node "ip-102-168-53-214.us-east-2.compute.internal" is ready
2025-06-28 19:21:29 [i] node "ip-102-168-9-199.us-east-2.compute.internal" is ready
2025-06-28 19:21:29 [i] created 1 managed nodegroup(s) in cluster "myEKSCluster"
2025-06-28 19:21:29 [i] checking security group configuration for all nodegroups
2025-06-28 19:21:29 [i] all nodegroups have up-to-date CloudFormation templates
- $ aws eks update-kubeconfig --name myEKSCluster --region us-east-2
Added new context aws:eks:us-east-2:747294369154:cluster/myEKSCluster to /home/cloudshell-user/.kube/config
- $
```

#### Paso 5: Verificar el estado de los nodos en el clúster

```
""bash
```

```
kubectl get nodes
```

```
""
```

```
CloudShell
us-east-2 +
2025-06-28 19:21:29 [i] checking security group configuration for all nodegroups
2025-06-28 19:21:29 [i] all nodegroups have up-to-date CloudFormation templates
- $ aws eks update-kubeconfig --name myEKSCluster --region us-east-2
Added new context aws:eks:us-east-2:747294369154:cluster/myEKSCluster to /home/cloudshell-user/.kube/config
- $ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-102-168-53-214.us-east-2.compute.internal Ready    cnode   3d28s v1.22.3-eks-473151a
ip-102-168-9-199.us-east-2.compute.internal Ready    cnode   3d27s v1.22.3-eks-473151a
- $
```

#### Paso 6: Instalar helm en el cluster

```
""bash
```

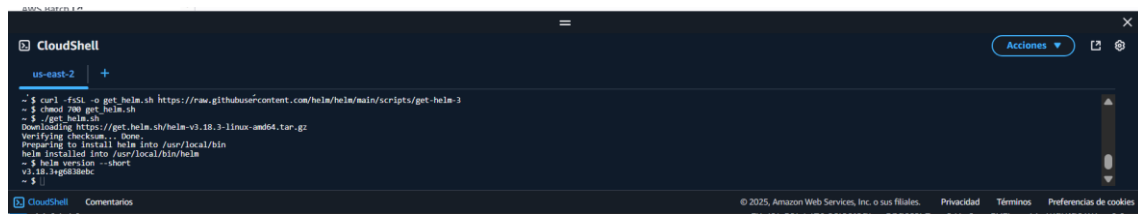
```
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
```

```
chmod 700 get_helm.sh
```

```
./get_helm.sh
```

helm version --short

'''



The screenshot shows a CloudShell terminal window with a dark theme. The terminal output is as follows:

```
us-east-2 +
~ $ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
~ $ chmod 700 get_helm.sh
~ $ ./get_helm.sh
Downloading https://get.helm.sh/helm-v3.18.3-linux-amd64.tar.gz
Verifying checksum: Done
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
~ $ helm version --short
v3.18.3+g6838ebc
~ $
```

The CloudShell interface includes a top bar with the 'CloudShell' logo, a hamburger menu, and an 'Acciones' button. The bottom status bar displays the copyright notice '© 2025, Amazon Web Services, Inc. o sus filiales.' along with links for 'Privacidad', 'Términos', and 'Preferencias de cookies'.