**tep 3: Launch and Configure Amazon EKS Worker Nodes**

Now that your VPC and Kubernetes control plane are created, you can launch and configure your worker nodes.

**Important**

Amazon EKS worker nodes are standard Amazon EC2 instances, and you are billed for them based on normal Amazon EC2 On-Demand Instance prices. For more information, see [Amazon EC2 Pricing](https://aws.amazon.com/ec2/pricing/on-demand/).

**To launch your worker nodes**

1. Wait for your cluster status to show as ACTIVE. If you launch your worker nodes before the cluster is active, the worker nodes will fail to register with the cluster and you will have to relaunch them.
2. Open the AWS CloudFormation console at [https://console.aws.amazon.com/cloudformation](https://console.aws.amazon.com/cloudformation/).
3. From the navigation bar, select a Region that supports Amazon EKS.

**Note**

Amazon EKS is available in the following Regions at this time:

* + **US West (Oregon)** (us-west-2)
  + **US East (N. Virginia)** (us-east-1)
  + **US East (Ohio)** (us-east-2)
  + **EU (Frankfurt)** (eu-central-1)
  + **EU (Stockholm)** (eu-north-1)
  + **EU (Ireland)** (eu-west-1)
  + **EU (London)** (eu-west-2)
  + **EU (Paris)** (eu-west-3)
  + **Asia Pacific (Tokyo)** (ap-northeast-1)
  + **Asia Pacific (Seoul)** (ap-northeast-2)
  + **Asia Pacific (Mumbai)** (ap-south-1)
  + **Asia Pacific (Singapore)** (ap-southeast-1)
  + **Asia Pacific (Sydney)** (ap-southeast-2)

1. Choose **Create stack**.
2. For **Choose a template**, select **Specify an Amazon S3 template URL**.
3. Paste the following URL into the text area and choose **Next**:

https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-02-11/amazon-eks-nodegroup.yaml

1. On the **Specify Details** page, fill out the following parameters accordingly, and choose **Next**.
   * **Stack name**: Choose a stack name for your AWS CloudFormation stack. For example, you can call it ***<cluster-name>*-worker-nodes**.
   * **ClusterName**: Enter the name that you used when you created your Amazon EKS cluster.

**Important**

This name must exactly match the name you used in [Step 1: Create Your Amazon EKS Cluster](https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html#eks-create-cluster); otherwise, your worker nodes cannot join the cluster.

* + **ClusterControlPlaneSecurityGroup**: Choose the **SecurityGroups** value from the AWS CloudFormation output that you generated with [Create your Amazon EKS Cluster VPC](https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html#vpc-create).
  + **NodeGroupName**: Enter a name for your node group. This name can be used later to identify the Auto Scaling node group that is created for your worker nodes.
  + **NodeAutoScalingGroupMinSize**: Enter the minimum number of nodes that your worker node Auto Scaling group can scale in to.
  + **NodeAutoScalingGroupDesiredCapacity**: Enter the desired number of nodes to scale to when your stack is created.
  + **NodeAutoScalingGroupMaxSize**: Enter the maximum number of nodes that your worker node Auto Scaling group can scale out to.
  + **NodeInstanceType**: Choose an instance type for your worker nodes.
  + **NodeImageId**: Enter the current Amazon EKS worker node AMI ID for your Region. The AMI IDs for the latest Amazon EKS-optimized AMI (with and without [GPU support](https://docs.aws.amazon.com/eks/latest/userguide/gpu-ami.html)) are shown in the following table.

**Note**

The Amazon EKS-optimized AMI with GPU support only supports P2 and P3 instance types. Be sure to specify these instance types in your worker node AWS CloudFormation template. Because this AMI includes third-party software that requires an end user license agreement (EULA), you must subscribe to the AMI in the AWS Marketplace and accept the EULA before you can use the AMI in your worker node groups. To subscribe to the AMI, visit [the AWS Marketplace](https://aws.amazon.com/marketplace/pp/B07GRHFXGM).

**Kubernetes version 1.11**

|  |  |  |
| --- | --- | --- |
| **Region** | **Amazon EKS-optimized AMI** | **with GPU support** |
| US West (Oregon) (us-west-2) | ami-0c28139856aaf9c3b | ami-06045aa686f46dd58 |
| US East (N. Virginia) (us-east-1) | ami-0eeeef929db40543c | ami-0558da965e2fc68b0 |
| US East (Ohio) (us-east-2) | ami-0484545fe7d3da96f | ami-0c3afad2ea917168e |
| EU (Frankfurt) (eu-central-1) | ami-032ed5525d4df2de3 | ami-0939712219b80b525 |
| EU (Stockholm) (eu-north-1) | ami-0154b2479ba20f8bb | ami-18bf3666 |
| EU (Ireland) (eu-west-1) | ami-098fb7e9b507904e7 | ami-014969e8d07b2fc9f |
| EU (London) (eu-west-2) | ami-0d69ab00cb41d6eda | ami-0bb14a7e038ad534c |
| EU (Paris) (eu-west-3) | ami-018ebb030cf6ae00b | ami-0a3db0dbd972b38f2 |
| Asia Pacific (Tokyo) (ap-northeast-1) | ami-07fdc9272ce5b0ce5 | ami-0880d3b662781d6d6 |
| Asia Pacific (Seoul) (ap-northeast-2) | ami-091e0e1906e653417 | ami-0c3db49d90afa0f1e |
| Asia Pacific (Mumbai) (ap-south-1) | ami-0b6f791fc54125a8a | ami-00b37b9a91efc5fff |
| Asia Pacific (Singapore) (ap-southeast-1) | ami-038d55c26bf01998f | ami-0c903ead334faa6a3 |
| Asia Pacific (Sydney) (ap-southeast-2) | ami-0e07b5081bb77d540 | ami-02d7e0f064bd7d8e0 |

**Kubernetes version 1.10**

|  |  |  |
| --- | --- | --- |
| **Region** | **Amazon EKS-optimized AMI** | **with GPU support** |
| US West (Oregon) (us-west-2) | ami-0e7ee8863c8536cce | ami-02e0b615d7749e016 |
| US East (N. Virginia) (us-east-1) | ami-09a7630ca9ee4ee22 | ami-00cce60e4c241de4c |
| US East (Ohio) (us-east-2) | ami-02a8a05e480e902e2 | ami-0bbfeb020c5ec10ee |
| EU (Frankfurt) (eu-central-1) | ami-0b8d223ce03e6fabc | ami-0c1746c6d5d61b4d3 |
| EU (Stockholm) (eu-north-1) | ami-09be5053dbb1a515d | ami-63aa231d |
| EU (Ireland) (eu-west-1) | ami-0103822d44fc52f97 | ami-08d23ed2de9320c90 |
| EU (London) (eu-west-2) | ami-017c4d847b606e125 | ami-0f136e808b9365a1c |
| EU (Paris) (eu-west-3) | ami-0c7fc5c0784b58207 | ami-0b6c4fac3cdcc191d |
| Asia Pacific (Tokyo) (ap-northeast-1) | ami-0e831f9f650f2f8ab | ami-061f5b653b1a98557 |
| Asia Pacific (Seoul) (ap-northeast-2) | ami-0378f1fac83cbf438 | ami-0a8159b97b9a7e078 |
| Asia Pacific (Mumbai) (ap-south-1) | ami-0ac369c3b2206d2ea | ami-03ba4c3cea82ce746 |
| Asia Pacific (Singapore) (ap-southeast-1) | ami-0fa3f3282eb89b795 | ami-02aa3e8ad27163456 |
| Asia Pacific (Sydney) (ap-southeast-2) | ami-01d0ab2e9506b8db0 | ami-0679fa5d74309eb79 |

**Note**

The Amazon EKS worker node AMI is based on Amazon Linux 2. You can track security or privacy events for Amazon Linux 2 at the [Amazon Linux Security Center](https://alas.aws.amazon.com/alas2.html) or subscribe to the associated [RSS feed](https://alas.aws.amazon.com/AL2/alas.rss). Security and privacy events include an overview of the issue, what packages are affected, and how to update your instances to correct the issue.

* + **KeyName**: Enter the name of an Amazon EC2 SSH key pair that you can use to connect using SSH into your worker nodes with after they launch. If you don't already have an Amazon EC2 keypair, you can create one in the AWS Management Console. For more information, see [Amazon EC2 Key Pairs](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html) in the *Amazon EC2 User Guide for Linux Instances*.

**Note**

If you do not provide a keypair here, the AWS CloudFormation stack creation fails.

* + **BootstrapArguments**: Specify any optional arguments to pass to the worker node bootstrap script, such as extra **kubelet** arguments. For more information, view the bootstrap script usage information at <https://github.com/awslabs/amazon-eks-ami/blob/master/files/bootstrap.sh>
  + **VpcId**: Enter the ID for the VPC that you created in [Create your Amazon EKS Cluster VPC](https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html#vpc-create).
  + **Subnets**: Choose the subnets that you created in [Create your Amazon EKS Cluster VPC](https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html#vpc-create).

1. On the **Options** page, you can choose to tag your stack resources. Choose **Next**.
2. On the **Review** page, review your information, acknowledge that the stack might create IAM resources, and then choose **Create**.
3. When your stack has finished creating, select it in the console and choose the **Outputs** tab.
4. Record the **NodeInstanceRole** for the node group that was created. You need this when you configure your Amazon EKS worker nodes.

**To enable worker nodes to join your cluster**

1. Download, edit, and apply the AWS authenticator configuration map:
   1. Download the configuration map.

**curl -O https://amazon-eks.s3-us-west-2.amazonaws.com/cloudformation/2019-02-11/aws-auth-cm.yaml**

* 1. Open the file with your favorite text editor. Replace the *<ARN of instance role (not instance profile)>*snippet with the **NodeInstanceRole** value that you recorded in the previous procedure, and save the file.

**Important**

Do not modify any other lines in this file.

apiVersion: v1

kind: ConfigMap

metadata:

name: aws-auth

namespace: kube-system

data:

mapRoles: |

- rolearn: <ARN of instance role (not instance profile)>

username: system:node:{{EC2PrivateDNSName}}

groups:

- system:bootstrappers

- system:nodes

* 1. Apply the configuration. This command may take a few minutes to finish.

**kubectl apply -f aws-auth-cm.yaml**

**Note**

If you receive the error "aws-iam-authenticator": executable file not found in $PATH, then your **kubectl** is not configured for Amazon EKS. For more information, see [Installing aws-iam-authenticator](https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html).

If you receive any other authorization or resource type errors, see [Unauthorized or Access Denied (kubectl)](https://docs.aws.amazon.com/eks/latest/userguide/troubleshooting.html#unauthorized) in the troubleshooting section.

1. Watch the status of your nodes and wait for them to reach the Ready status.

**kubectl get nodes --watch**

1. (GPU workers only) If you chose a P2 or P3 instance type and the Amazon EKS-optimized AMI with GPU support, you must apply the [NVIDIA device plugin for Kubernetes](https://github.com/NVIDIA/k8s-device-plugin) as a daemon set on your cluster with the following command.

**kubectl apply -f https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v1.11/nvidia-device-plugin.yml**

**Step 4: Launch a Guest Book Application**

In this section, you create a sample guest book application to test your new cluster.

**Note**

For more information about setting up the guest book example, see <https://github.com/kubernetes/examples/blob/master/guestbook-go/README.md> in the Kubernetes documentation.

**To create your guest book application**

1. Create the Redis master replication controller.

**kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-controller.json**

**Note**

If you receive the error "aws-iam-authenticator": executable file not found in $PATH, then your **kubectl** is not configured for Amazon EKS. For more information, see [Installing aws-iam-authenticator](https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html).

If you receive any other authorization or resource type errors, see [Unauthorized or Access Denied (kubectl)](https://docs.aws.amazon.com/eks/latest/userguide/troubleshooting.html#unauthorized) in the troubleshooting section.

Output:

replicationcontroller "redis-master" created

1. Create the Redis master service.

**kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-master-service.json**

Output:

service "redis-master" created

1. Create the Redis slave replication controller.

**kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-controller.json**

Output:

replicationcontroller "redis-slave" created

1. Create the Redis slave service.

**kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/redis-slave-service.json**

Output:

service "redis-slave" created

1. Create the guestbook replication controller.

**kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/guestbook-controller.json**

Output:

replicationcontroller "guestbook" created

1. Create the guestbook service.

**kubectl apply -f https://raw.githubusercontent.com/kubernetes/examples/master/guestbook-go/guestbook-service.json**

Output:

service "guestbook" created

1. Query the services in your cluster and wait until the **External IP** column for the guestbook service is populated.

**Note**

It may take several minutes before the IP address is available.

**kubectl get services -o wide**

1. After your external IP address is available, point a web browser to that address at port 3000 to view your guest book. For example, *http://a7a95c2b9e69711e7b1a3022fdcfdf2e-1985673473.us-west-2.elb.amazonaws.com:3000*

**Note**

It may take several minutes for DNS to propagate and for your guest book to show up.


                        Guest book
                    

**Important**

If you are unable to connect to the external IP address with your browser, be sure that your corporate firewall is not blocking non-standards ports, like 3000. You can try switching to a guest network to verify.

**Step 5: Cleaning Up Guest Book Objects**

When you are finished experimenting with your guest book application, you should clean up the resources that you created for it. The following command deletes all of the services and replication controllers for the guest book application:

**kubectl delete rc/redis-master rc/redis-slave rc/guestbook svc/redis-master svc/redis-slave svc/guestbook**

**Note**

If you receive the error "aws-iam-authenticator": executable file not found in $PATH, then your **kubectl** is not configured for Amazon EKS. For more information, see [Installing aws-iam-authenticator](https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html).

If you receive any other authorization or resource type errors, see [Unauthorized or Access Denied (kubectl)](https://docs.aws.amazon.com/eks/latest/userguide/troubleshooting.html#unauthorized) in the troubleshooting section.