Contents
Overview.........................................................................................................................2
   Portworx PX-Enterprise on Kubernetes on AWS.........................................................2
   Cost ...............................................................................................................................3
   Architecture ................................................................................................................4
   Planning the deployment ..............................................................................................6
      Specialized knowledge .............................................................................................6
      AWS account ............................................................................................................6
      Technical requirements ............................................................................................6
   Deployment options ....................................................................................................6
   Deployment steps .........................................................................................................7
      Step 1. Sign in to your AWS account .........................................................................7
      Step 2. Launch the Quick Start ................................................................................8
         Option 1: Parameters for deploying Portworx into a new VPC ...............................9
         Option 2: Parameters for deploying Portworx into an existing VPC .....................12
      Step 3. Test the deployment ......................................................................................15
      Step 4. Access Portworx from your laptop (optional). ..............................................17
   Best practices for using Portworx PX-Enterprise on AWS ......................................18
   Security .......................................................................................................................18
This Quick Start was created by Portworx, Inc. in collaboration with Amazon Web Services (AWS).

Quick Starts are automated reference deployments that use AWS CloudFormation templates to deploy key technologies on AWS, following AWS best practices.

Overview

This Quick Start reference deployment guide provides step-by-step instructions for deploying Portworx PX-Enterprise in Kubernetes on the AWS Cloud.

This Quick Start is for users who want to use AWS and its products and services, such as virtual private clouds (VPCs), Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Kubernetes Service (Amazon EKS), and AWS CloudFormation to launch Kubernetes and Portworx for their container environment.

This reference deployment uses the Amazon EKS Architecture Quick Start as a foundation to provide a fully managed, highly available, and certified Kubernetes-conformant control plane for Portworx PK-Enterprise.

It is meant to be low touch. If you want to customize the environment, however, you can use the underlying AWS CloudFormation templates directly.

Portworx PX-Enterprise on Kubernetes on AWS

Portworx is the cloud-native storage company that allows reduction of the cost and complexity of containerized application environments. With Portworx, you can manage databases or stateful services on Kubernetes.

When you use Portworx PX-Enterprise to run stateful services in containers, you get:
• **High performance:** Portworx PX-Enterprise typically delivers I/O within 3 percent of bare metal speeds for SQL and NoSQL databases, message queues, Content Management System (CMS), and CI/CD applications.

• **Application awareness:** All of the storage functionality provided by Portworx takes into account the inherent distributed nature of applications. It can also apply application-specific commands before key operations such as snapshots or migrations. The Quick Start will install STORK, an open-source Kubernetes scheduler-extender created by Portworx that enables application awareness in Amazon EKS.

• **Deep Kubernetes integration:** With STORK, Portworx PX-Enterprise can manage the entire stateful lifecycle of an application—from optimal scheduling decisions based on the proximity to data, to Kubernetes-driven backup, restore, and migration functionality.

• **Seamless migration:** Portworx can pair itself with other clusters across Availability Zones, Regions, and even different clouds. This allows the seamless migration of the entire state of an application to any paired Kubernetes cluster. By default, the Quick Start deploys across Availability Zones, and Portworx can easily move volumes across them. If you deploy the same Quick Start multiple times in different or same Regions, you can also connect these clusters.

• **Focus on automation and analytics:** The Quick Start includes the Portworx SDK so that engineers can use RESTful APIs to automatically determine how storage is allocated, consumed, encrypted, backed-up and managed across multiple clouds, including defining I/O levels, creating snapshots, and migrating and managing data based on their unique business needs.

Portworx enables data to be available across Availability Zones and enables failed pods to be rescheduled on a host in a different Availability Zone. This Quick Start deploys Portworx on an Amazon EKS cluster across Availability Zones where virtual volumes can then be provisioned.

**Cost**

You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using the Quick Start.

The AWS CloudFormation template for this Quick Start includes configuration parameters that you can customize. Some of these settings, such as instance type, will affect the cost of deployment. For cost estimates, see the pricing pages for each AWS service you will be using. Prices are subject to change.
Tip  After you deploy the Quick Start, we recommend that you enable the AWS Cost and Usage Report to track costs associated with the Quick Start. This report delivers billing metrics to an S3 bucket in your account. It provides cost estimates based on usage throughout each month, and finalizes the data at the end of the month. For more information about the report, see the AWS documentation.

Portworx PX-Enterprise is deployed as part of this Quick Start and is provided with a free 30-day license. After the trial is over, you can purchase a license to continue using the product. For more information on license types, what features are included with each type of license, and how to upgrade or transfer your license, visit the Portworx license documentation.

Architecture

Deploying this Quick Start for a new virtual private cloud (VPC) with default parameters builds the following Portworx PX-Enterprise environment in the AWS Cloud. The diagram shows two Availability Zones; the Quick Start deployment includes a third Availability Zone.
The Quick Start sets up the following:

- A highly available architecture that spans three Availability Zones.*
- A VPC configured with public and private subnets according to AWS best practices, to provide you with your own virtual network on AWS.*
- In the public subnets, managed NAT gateways to allow outbound internet access for resources in the private subnets.*
- In the public subnets, a Linux bastion host in an Auto Scaling group to allow inbound Secure Shell (SSH) access to EC2 instances in public and private subnets.*
- In the private subnets, an Amazon EKS cluster.
- Portworx, deployed as a DaemonSet in the Amazon EKS cluster to provide highly available virtual volumes to applications that are deployed in the Amazon EKS cluster.

* These requirements are specific to the AWS Cloud and may vary depending on the user's specific needs and configurations.
* The template that deploys the Quick Start into an existing VPC skips the tasks marked by asterisks and prompts you for your existing VPC configuration.

**Planning the deployment**

**Specialized knowledge**

Before you deploy this Quick Start, we recommend that you become familiar with the following AWS services. (If you are new to AWS, see [Getting Started with AWS](https://aws.amazon.com)).

- Amazon EC2
- Amazon Elastic Block Store (Amazon EBS)
- Amazon VPC
- AWS CloudFormation
- Amazon EKS
- Portworx PX-Enterprise

**AWS account**

If you don’t already have an AWS account, create one at [https://aws.amazon.com](https://aws.amazon.com) by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad.

Your AWS account is automatically signed up for all AWS services. You are charged only for the services you use.

**Technical requirements**

- [AWS account configuration](https://aws.amazon.com)
- Knowledge of Kubernetes
- Knowledge of how to use kubectl

**Deployment options**

This Quick Start provides two deployment options:

- **Deploy Portworx into a new VPC** (end-to-end deployment). This option builds a new AWS environment consisting of the VPC, subnets, NAT gateways, security groups, bastion hosts, and other infrastructure components, and then deploys Portworx PX-Enterprise into this new VPC in the Amazon EKS cluster.
• **Deploy Portworx into an existing VPC.** This option provisions Portworx PX-Enterprise in an Amazon EKS cluster in your existing AWS infrastructure.

The Quick Start provides separate templates for these options. It also lets you configure CIDR blocks, instance types, and Portworx settings, as discussed later in this guide.

**Deployment steps**

**Step 1. Sign in to your AWS account**

1. Sign in to your AWS account at [https://aws.amazon.com](https://aws.amazon.com) with an IAM user role that has the necessary permissions. For details, see [Planning the deployment](#) earlier in this guide.

2. Make sure that your AWS account is configured correctly, as discussed in the [Technical requirements](#) section.

3. Use the Region selector in the navigation bar to choose the AWS Region where you want to deploy Portworx PX-Enterprise on AWS.

   **Note** Amazon EKS isn’t currently supported in all AWS Regions. For a current list of supported Regions, see the [AWS Regions and Endpoints webpage](#).

4. Create a [key pair](#) in your preferred Region.

   ![Create Key Pair](image)

   **Figure 2:** Create a key pair by providing a name

5. If necessary, [request a service limit increase](#) for the Amazon EC2 instance type you want to use for your Amazon EKS nodes. You might need to do this if you already have an existing deployment that uses this instance type, and you think you might exceed the [default limit](#) with this deployment.
Step 2. Launch the Quick Start

**Note** You are responsible for the cost of the AWS services used while running this Quick Start reference deployment. There is no additional cost for using this Quick Start. For full details, see the pricing pages for each AWS service you will be using in this Quick Start. Prices are subject to change.

1. Choose one of the following options to launch the AWS CloudFormation template into your AWS account. For help choosing an option, see [deployment options](#) earlier in this guide.

```
Deploy Portworx into a
new VPC on AWS
```

```
Deploy Portworx into an
existing VPC on AWS
```

**Important** If you’re deploying Portworx PX-Enterprise into an existing VPC, make sure that your VPC has three private subnets in different Availability Zones for the database instances. These subnets require [NAT gateways or NAT instances](#) in their route tables, to allow the instances to download packages and software without exposing them to the internet. You will also need the domain name option configured in the DHCP options as explained in the [Amazon VPC documentation](#). You will be prompted for your VPC settings when you launch the Quick Start.

Each deployment takes 20-30 minutes to complete.

2. Check the Region that’s displayed in the upper-right corner of the navigation bar, and change it if necessary. This is where the network infrastructure for Portworx will be built. The template is launched in the US West (Oregon) Region by default.

3. On the Select Template page, keep the default setting for the template URL, and then choose Next.

4. On the Specify Details page, change the stack name if needed. Review the parameters for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary. When you finish reviewing and customizing the parameters, choose Next.
Figure 3: Provide Portworx settings

In the following tables, parameters are listed by category and described separately for the two deployment options:

- Parameters for deploying Portworx into a new VPC
- Parameters for deploying Portworx into an existing VPC

OPTION 1: PARAMETERS FOR DEPLOYING PORTWORX INTO A NEW VPC

View template

Network configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability Zones</td>
<td>Requires input</td>
<td>The list of Availability Zones to use for the subnets in the VPC. The Quick Start uses two Availability Zones from your list and preserves the logical order you specify.</td>
</tr>
<tr>
<td>VPC CIDR</td>
<td>10.0.0.0/16</td>
<td>The CIDR block for the VPC.</td>
</tr>
<tr>
<td>Private Subnet 1 CIDR</td>
<td>10.0.0.0/19</td>
<td>The CIDR block for the private subnet 1 located in Availability Zone 1.</td>
</tr>
<tr>
<td>Private Subnet 2 CIDR</td>
<td>10.0.32.0/19</td>
<td>The CIDR block for the private subnet 2 located in Availability Zone 2.</td>
</tr>
<tr>
<td>Private Subnet 3 CIDR</td>
<td>10.0.64.0/19</td>
<td>The CIDR block for the private subnet 3 located in Availability Zone 3.</td>
</tr>
</tbody>
</table>
### Amazon EC2 configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH key name</td>
<td>Requires input</td>
<td>The name of an existing public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred Region.</td>
</tr>
</tbody>
</table>

### Amazon EKS configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes instance type</td>
<td>t2.medium</td>
<td>The type of EC2 instance for the node instances.</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>3</td>
<td>The number of Amazon EKS node instances.</td>
</tr>
<tr>
<td>Node group name</td>
<td>Default</td>
<td>The name for the Amazon EKS node group.</td>
</tr>
<tr>
<td>Node volume size</td>
<td>20</td>
<td>The size for the node’s root EBS volumes.</td>
</tr>
</tbody>
</table>
### Additional EKS Admin ARNs

**Parameter label** (AdditionalEKSAdminArns)  
**Default**: Optional  
**Description**: The comma-separated list of IAM users/role Amazon Resource Names (ARNs) to be granted admin access to the EKS cluster.

### Kubernetes version

**Parameter label** (KubernetesVersion)  
**Default**: 1.11  
**Description**: The Kubernetes version of the Amazon EKS cluster.

---

### Portworx configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portworx version tag</td>
<td>v1.11.0</td>
<td>The tag for the Portworx container image version.</td>
</tr>
<tr>
<td>(PXKubernetesVersion)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version of Portworx X.Y.Z</td>
<td>2.0.1</td>
<td>The version of Portworx that is deployed by this Quick Start.</td>
</tr>
<tr>
<td>(PortworxVersion)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version of Lighthouse X.Y.Z</td>
<td>2.0.1</td>
<td>The version of the Portworx Lighthouse UI.</td>
</tr>
<tr>
<td>(LightHouseVersion)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Template string for partitioning</td>
<td>type=gp2, size=100</td>
<td>The template string used for partitioning, with the size in gigabytes. For more information, see <a href="https://docs.portworx.com/cloud-references/auto-disk-provisioning/aws/">https://docs.portworx.com/cloud-references/auto-disk-provisioning/aws/</a>.</td>
</tr>
<tr>
<td>(VolumeTemplateString)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### AWS Quick Start configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Start S3 bucket name</td>
<td>aws-quickstart</td>
<td>The S3 bucket you have created for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.</td>
</tr>
<tr>
<td>(QSS3BucketName)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Quick Start S3 key prefix (QSS3KeyPrefix)
- **Default**: quickstart-eks-portworx-px-enterprise/
- **Description**: The S3 key name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.

### Other parameters:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lambda zips bucket name (LambdaZipsBucketName)</td>
<td>Optional</td>
<td>The name of the bucket where the Lambda.zip files should be placed; if left blank, a bucket will be created.</td>
</tr>
</tbody>
</table>

---

**OPTION 2: PARAMETERS FOR DEPLOYING PORTWORX INTO AN EXISTING VPC**

**View template**

### Network configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allowed external access CIDR (RemoteAccessCIDR)</td>
<td>Requires input</td>
<td>The CIDR IP range that is permitted to access the instances. We recommend that you set this value to a trusted IP range.</td>
</tr>
<tr>
<td>Public subnet 1 ID (PublicSubnet1ID)</td>
<td>Requires input</td>
<td>The ID of the public subnet 1 in Availability Zone 1 (e.g., subnet-a0246dcd).</td>
</tr>
<tr>
<td>Public subnet 2 ID (PublicSubnet2ID)</td>
<td>Requires input</td>
<td>The ID of the public subnet in Availability Zone 2 in your existing VPC (e.g., subnet-a0246dcd).</td>
</tr>
<tr>
<td>Private Subnet 1 ID (PrivateSubnet1ID)</td>
<td>Requires input</td>
<td>The ID of the private subnet in Availability Zone 1 in your existing VPC (e.g., subnet-a0246dcd).</td>
</tr>
<tr>
<td>Private Subnet 2 ID (PrivateSubnet2ID)</td>
<td>Requires input</td>
<td>The ID of the private subnet in Availability Zone 2 in your existing VPC (e.g., subnet-a0246dcd).</td>
</tr>
<tr>
<td>Private Subnet 3 ID (PrivateSubnet3ID)</td>
<td>Requires input</td>
<td>The ID of the private subnet in Availability Zone 3 in your existing VPC (e.g., subnet-b58c3d67).</td>
</tr>
<tr>
<td>Parameter label (name)</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>VPC ID (VPCID)</strong></td>
<td><strong>Requires input</strong></td>
<td>The ID of your existing VPC (e.g., vpc-0343606e).</td>
</tr>
</tbody>
</table>

**Amazon EC2 configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SSH key name (KeyPairName)</strong></td>
<td><strong>Requires input</strong></td>
<td>The name of an existing public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred Region.</td>
</tr>
</tbody>
</table>

**Amazon EKS configuration:**

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nodes instance type (NodeInstanceType)</strong></td>
<td>t2.medium</td>
<td>The type of EC2 instance for the node instances.</td>
</tr>
<tr>
<td><strong>Number of nodes (NumberOfNodes)</strong></td>
<td>3</td>
<td>The number of Amazon EKS node instances.</td>
</tr>
<tr>
<td><strong>Node group name (NodeGroupName)</strong></td>
<td>Default</td>
<td>The name for the Amazon EKS node group.</td>
</tr>
<tr>
<td><strong>Node volume size (NodeVolumeSize)</strong></td>
<td>20</td>
<td>The size for the node's root EBS volumes.</td>
</tr>
<tr>
<td><strong>Additional EKS admin ARNs (AdditionalEKSAdminArns)</strong></td>
<td><strong>Optional</strong></td>
<td>The comma-separated list of IAM users/role Amazon Resource Names (ARNs) to be granted admin access to the EKS cluster.</td>
</tr>
<tr>
<td><strong>Kubernetes version (KubernetesVersion)</strong></td>
<td>1.11</td>
<td>The Kubernetes version of the Amazon EKS cluster.</td>
</tr>
</tbody>
</table>
### Portworx configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Portworx version tag (PXKubernetesVersion)</td>
<td>v1.11.0</td>
<td>The tag for the Portworx container image version.</td>
</tr>
<tr>
<td>Version of Portworx X.Y.Z (PortworxVersion)</td>
<td>2.0.1</td>
<td>The version of Portworx that is deployed by this Quick Start.</td>
</tr>
<tr>
<td>Version of Lighthouse X.Y.Z (LightHouseVersion)</td>
<td>2.0.1</td>
<td>The version of the Portworx Lighthouse UI.</td>
</tr>
</tbody>
</table>
| Template string for partitioning (VolumeTemplateString)     | type=gp2, size=100 | The template string used for partitioning, with the size in gigabytes. For more information, see [https://docs.portworx.com/cloud-references/auto-disk-provisioning/aws/](https://docs.portworx.com/cloud-references/auto-disk-provisioning/aws/).

### AWS Quick Start configuration:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Start S3 bucket name (QSS3BucketName)</td>
<td>aws-quickstart</td>
<td>The S3 bucket you have created for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. The bucket name can include numbers, lowercase letters, uppercase letters, and hyphens, but should not start or end with a hyphen.</td>
</tr>
<tr>
<td>Quick Start S3 key prefix (QSS3KeyPrefix)</td>
<td>quickstart-eks-portworx-px-enterprise/</td>
<td>The S3 key name prefix used to simulate a folder for your copy of Quick Start assets, if you decide to customize or extend the Quick Start for your own use. This prefix can include numbers, lowercase letters, uppercase letters, hyphens, and forward slashes.</td>
</tr>
</tbody>
</table>

### Other parameters:

<table>
<thead>
<tr>
<th>Parameter label (name)</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lambda zips bucket name (LambdaZipsBucketName)</td>
<td>Optional</td>
<td>The name of the bucket where the Lambda.zip files should be placed; if left blank, a bucket will be created.</td>
</tr>
</tbody>
</table>
5. On the **Options** page, you can **specify tags** (key-value pairs) for resources in your stack and **set advanced options**. When you're done, choose **Next**.

6. On the **Review** page, review and confirm the template settings. Under **Capabilities**, select the check box to acknowledge that the template will create IAM resources.

7. Choose **Create** to deploy the stack.

8. Monitor the status of the stack. When the status is **CREATE_COMPLETE**, the Portworx PX-Enterprise cluster is ready.

9. Use the URLs displayed in the **Outputs** tab for the stack to view the resources that were created.

```
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>KubeConfigPath</td>
<td>s3://ryan-test-0010-eksstream-115rb2v3iyjx/.kube/config</td>
</tr>
<tr>
<td>BastionIP</td>
<td>35.155.208.226</td>
</tr>
</tbody>
</table>
```

**Figure 4: Master stack outputs**

**Step 3. Test the deployment**

1. Connect to the bastion host by using **SSH** and your Private Key. The bastion host IP address is listed on the **Outputs** tab.

   ```
   $ ssh -i /path/to/keypair.pem ec2-user@<BastionIP>
   ```

2. Use **kubectl** to invoke **pxctl** (pixie-cuttle), the Portworx command line.

   ```
   $ kubectl get pods -o wide -n kube-system -l name=portworx
   $ PX_POD=$(kubectl get pods -l name=portworx -n kube-system -o jsonpath='{}items[0].metadata.name')
   $ kubectl exec $PX_POD -n kube-system -- /opt/pwx/bin/pxctl status
   ```

3. Use Portworx to list volumes and settings.

   ```
   $ kubectl exec $PX_POD -n kube-system -- /opt/pwx/bin/pxctl volume list
   ```
4. Use the `examples/` directory to deploy a database.

Using one the database templates available in `mysql/`, `cassandra/` or `postgres/`, go to the AWS CloudFormation console and launch the database onto the Quick Start Portworx cluster.

Accept the database template defaults and use `KubeConfigPath` and `KubeManifestLambdaArn` from the EKSStack nested stack output.

![KubeManifestConfig](image)

![PortworxDatabaseConfig](image)

**Figure 5: Settings for the cassandra/ template**

5. Test from the bastion host

```
$ kubectl get po -l app=cassandra
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>cassandra-0</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>cassandra-1</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>1m</td>
</tr>
<tr>
<td>cassandra-2</td>
<td>0/1</td>
<td>Running</td>
<td>0</td>
<td>33s</td>
</tr>
</tbody>
</table>
Step 4. Access Portworx from your laptop (optional)

1. From the bastion host, run the following command:

   ```bash
   kubectl edit -n kube-system configmap/aws-auth
   mapUsers: |
   - arn:aws:iam::<AccountID>:user/<User>
     username: <User>
     groups: 
     - system:masters
     - system:nodes
   mapAccounts: |
   - "<AccountID>"
   $ kubectl create clusterrolebinding <User>-cluster-admin-binding --
   clusterrole=cluster-admin --
   user=arn:aws:iam::<AccountID>:user/<User>
   ```

2. Next, from your laptop, run the following command:

   ```bash
   $ aws eks update-kubeconfig --name EKS-<UUID> --region <Region>
   Added new context arn:aws:eks:<Region>::<AccountID>::cluster/EKS-
   <UUID>
   $ kubectl config use-context arn:aws:eks:us-west-
   2:649513742363:cluster/EKS-j1oQjAxtMZR3
   ```

3. To access the Portworx UI.

   ```bash
   $ kubectl --namespace kube-system port-forward service/px-lighthouse
   8080:80
   Forwarding from 127.0.0.1:8080 -> 80
   Forwarding from [:1]:8080 -> 80
   ```

Best practices for using Portworx PX-Enterprise on AWS

Refer to Portworx’s detailed documentation on how to best run stateful services on AWS using Kubernetes, including how Auto-scaling groups work with Kubernetes.

Security

Use Portworx encrypted volumes for your databases to keep your data secure.

FAQ

Q. I encountered a CREATE_FAILED error when I launched the Quick Start.

A. If AWS CloudFormation fails to create the stack, we recommend that you relaunch the template with Rollback on failure set to No. (This setting is under Advanced in the AWS CloudFormation console, Options page.) With this setting, the stack’s state will be retained and the instance will be left running, so you can troubleshoot the issue. (Look at the log files in %ProgramFiles%\Amazon\EC2ConfigService and C:\cfn\log.)

Important When you set Rollback on failure to No, you will continue to incur AWS charges for this stack. Please make sure to delete the stack when you finish troubleshooting.

For additional information, see Troubleshooting AWS CloudFormation on the AWS website.
Q. I encountered a size limitation error when I deployed the AWS CloudFormation templates.

A. We recommend that you launch the Quick Start templates from the links in this guide or from another S3 bucket. If you deploy the templates from a local copy on your computer or from a non-S3 location, you might encounter template size limitations when you create the stack. For more information about AWS CloudFormation limits, see the [AWS documentation](https://aws.amazon.com/documentation/cloudformation/).

**Send us feedback**

To post feedback, submit feature ideas, or report bugs, use the [Issues](https://github.com/aws/Portworx-QuickStarts) section of the [GitHub repository](https://github.com/aws/Portworx-QuickStarts) for this Quick Start. If you’d like to submit code, please review the [Quick Start Contributor’s Guide](https://aws.amazon.com/about-aws/whats-new/quick-start-contributors-guide/).

**Additional resources**

**AWS services**

- [Amazon EC2](https://aws.amazon.com/ec2/)
- [Amazon EBS](https://aws.amazon.com/ebs/)
- [Amazon VPC](https://aws.amazon.com/vpc/)
- [Amazon EKS](https://aws.amazon.com/eks/)
- [AWS CloudFormation](https://aws.amazon.com/cloudformation/)

**Portworx documentation**

- [Portworx documentation](https://portworx.com/docs/)
- [Operating Portworx on Kubernetes](https://portworx.com/kubernetes/

**Quick Start reference deployments**

- [AWS Quick Start home page](https://aws.amazon.com/quickstart/)
- [Amazon EKS Architecture Quick Start](https://aws.amazon.com/eks/)

**Document revisions**

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
<th>In sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>February 2019</td>
<td>Initial publication</td>
<td>—</td>
</tr>
</tbody>
</table>