1.FSPVerify that AWS CLI version 1.16.308 or greater is installed on your system:

$ aws --version

Important: You must have Python version 2.7.9 or greater installed on your system. Otherwise, you receive an error.

Tip: Use package managers such as yum, apt-get, or homebrew for macOS to install the AWS CLI.

2.FSPCheck the current identity to verify that you're using the correct credentials that have permissions for the Amazon EKS cluster:

aws sts get-caller-identity

3.FSPCreate or update the kubeconfig file for your cluster:

aws eks --region region update-kubeconfig --name cluster\_name

Note: Replace region with your AWS Region. Replace cluster\_name with your cluster name.

By default, the configuration file is created at the kubeconfig path ($HOME/.kube/config) in your home directory or merged with an existing kubeconfig at that location. For Windows, the file is at %USERPROFILE%\.kube\config.

You can also specify another path by setting the [KUBECONFIG](https://kubernetes.io/docs/concepts/configuration/organize-cluster-access-kubeconfig/#the-kubeconfig-environment-variable) (from the Kubernetes website) environment variable, or with the following --kubeconfig option:

$ kubectl get pods --kubeconfig ./.kube/config

Note: For authentication when you run kubectl commands, you can specify an IAM role Amazon Resource Name (ARN) with the --role-arn option. Otherwise, the IAM entity in your default AWS CLI or AWS SDK credential chain is used. For more information, see [update-kubeconfig](https://awscli.amazonaws.com/v2/documentation/api/latest/reference/eks/update-kubeconfig.html) or complete step 6 in the "Create kubeconfig manually" section of [Creating or updating a kubeconfig file for an Amazon EKS cluster](https://docs.aws.amazon.com/eks/latest/userguide/create-kubeconfig.html).

4.FSPTest your configuration:

$ kubectl get svc

Example output:

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

svc/kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 1m