FleetIQ-EKS-Agones   
Modular Install Guide

This guide provides a four-step walkthrough on how to prepare for and run automated installment scripts for the FleetIQ-EKS-Agones solution. This walkthrough is particularly useful when you have existing workspaces or clusters that you want to leverage. Alternatively, follow the Quick Install Guide to do the automated deployment as a single step or use the Full Install Guide to complete each step manually.

Note: The installation process has been tested with new environments and clusters only. When using this process with existing workspaces and clusters, there might be configuration issues specific to your setup that might arise.

# PART 1: Create a workspace

The first step is to set up a standalone Cloud9 IDE workspace. This cloud-based IDE provides an isolated CLI environment from which you can create, configure, and control a cluster (recommended for initial sandbox tests). You can skip this step If you already have a workspace, but you must ensure that you have AWS CLI set up with administrator credentials that are configured to the environment you wish to work in. Verify that the environment variables $ACCOUNT\_ID and $REGION are configured for the account and region you want to deploy the solution in.

To get/set your account ID:

export ACCOUNT\_ID=$(aws sts get-caller-identity --output text --query Account)

To set your AWS region:

export AWS\_REGION=<region name e.g. us-east-1>

From the AWS Console, launch Cloud9 in the region you want to run the cluster in one of the following regions:

* + US East (Ohio)
  + US East (N. Virginia)
  + US West (Oregon)
  + Asia Pacific (Hong Kong)
  + Asia Pacific (Mumbai)
  + Asia Pacific (Seoul)
  + Asia Pacific (Singapore)
  + Asia Pacific (Sydney)
  + Asia Pacific (Tokyo)
  + Canada (Central)
  + Europe (Frankfurt)
  + Europe (Ireland)
  + Europe (London)
  + Europe (Paris)
  + Europe (Stockholm)
  + Middle East (Bahrain)
  + South America (São Paulo)
* Select Create environment
* Name it (e.g. EKS-FleetIQ-workspace), optionally provide a description and click Next step
* Select the following options:
  + Environment type: Create a new no-ingress EC2 instance for environment (access via Systems Manager)
  + Instance type: t3.small
  + Platform: Amazon Linux
  + Cost-saving setting: After one hour
  + Network settings: Keep default VPC/No preference or create/select a VPC and subnet based on your network environment
* Click Next step then Create environment
* Once the IDE is ready, customize the environment by closing the Welcome tab and lower work area, and opening a new terminal window in the main work area by clicking the green plus (+) sign.

**NOTE:** Please use the same terminal window throughout this guide to ensure environment variables are preserved between steps.

## **Add an IAM role to your workspace**

Create an administrator IAM role to manage permissions for all of the AWS components that are used in the FleetIQ-EKS-Agones solution. These permissions are required before you can complete the FleetIQ-EKS-Agones install process.

### Create a new IAM role

1. In the AWS Console, go to the IAM service. IAM is a global service, so you don’t need to specify a region.
2. In the sidebar, select **Roles** and click the **Create Role** button.
3. Select the following options:
   * **Select type of trusted entity**: AWS service.
   * **Choose a use case**: EC2.
4. Click **Next:Permissions**.
5. Select the permission policy **AdministratorAccess**.
6. Click **Next:Tags** to move forward, take the defaults, and click **Next:Review** to finalize the new role.
7. Enter the role name: “agonesfleetiq-admin” and add an optional role description.
8. Click **Create role**.

### Attach the IAM role to your Cloud9 EC2 instance

1. In the AWS Console, go to the EC2 service and select the region where you created your Cloud9 environment. (An EC2 instance was automatically created to run your Cloud9 environment).
2. In the **Resources** box, click **Running instances**. When you created your Cloud9 environment, an EC2 instance was automatically created for it. You should see this instance listed with a name that starts with “aws-cloud9…”.
3. Select your active Cloud9 instance (click the checkbox) and choose **Actions / Instance Settings / Modify IAM Role** (you may need to scroll the dropdown list).
4. From the **IAM Role** drop down list, choose the role “agonesfleetiq-admin” and select **Save**.

### Update IAM settings for your workspace

1. In the AWS Console, return to the Cloud9 service and open your IDE workspace. Be sure you’re working in the region where you created your workspace.
2. Click the gear icon (in top right corner), or open a new tab and choose **Open Preferences**.
3. Open **Project Settings: AWS Settings** and turn off AWS managed temporary credentials.
4. Open a terminal window and enter the following commands to finish updating your IAM settings.
   * Remove any existing credentials file:  
     rm -vf ${HOME}/.aws/credentials
   * Install jq:  
     sudo yum -y install jq
   * Configure the AWS CLI with your workspace region as default:  
     export ACCOUNT\_ID=$(aws sts get-caller-identity --output text --query Account)  
       
     export AWS\_REGION=$(curl -s 169.254.169.254/latest/dynamic/instance-identity/document | jq -r '.region')
   * Verify that AWS\_REGION is correctly set to your desired region:  
     test -n "$AWS\_REGION" && echo AWS\_REGION is "$AWS\_REGION" || echo AWS\_REGION is not set
   * Save these settings into bash\_profile:  
     echo "export ACCOUNT\_ID=${ACCOUNT\_ID}" | tee -a ~/.bash\_profile  
       
     echo "export AWS\_REGION=${AWS\_REGION}" | tee -a ~/.bash\_profile  
       
     aws configure set default.region ${AWS\_REGION}  
       
     aws configure get default.region
   * Use the [GetCallerIdentity](https://docs.aws.amazon.com/cli/latest/reference/sts/get-caller-identity.html) CLI command to validate that the Cloud9 IDE is using the correct IAM role:  
     aws sts get-caller-identity --query Arn | grep agonesfleetiq-admin -q && echo "IAM role valid" || echo "IAM role NOT valid"  
       
     If the IAM role is not valid, DO NOT PROCEED with cluster creation. Some possible reasons that your IAM role isn’t valid:
     + You named your IAM role differently. Modify the get-caller-identity command (between | grep and -q) with the correct name.
     + You had to reconnect to the instance. This may have re-applied the Managed Temporary Credentials. Restart the “Update IAM settings for your workspace“ at Step 1.

# Run the Part 1 Installation file

1. In the Cloud9 workspace main work area, click on the green plus (+) sign and create a **New File**.
2. Copy over the contents of the attached Part 1 install script and save the file as part1.sh. This script contains the Cloud9 tooling and configuration procedure.
3. In the Console tab, start the script:

. part1.sh

# PART 2: Create an EKS cluster

With a workspace set up, the next step is to create an EKS cluster. You can skip this step if you plan to work with an existing cluster. However, you must create an IAM policy and attach it to the existing node groups; this policy enables the pods, which handle status checks for the daemonset, can call the relevant GameLift APIs.

To create the IAM policy:

aws iam create-policy --policy-name <POLICYNAME> --policy-document '{"Version": "2012-10-17","Statement": [{"Sid": "VisualEditor0","Effect": "Allow","Action": ["gamelift:DescribeGameServerGroup","gamelift:DescribeGameServerInstances","gamelift:DescribeGameServer"],"Resource": "\*"}]}'

1. In the Cloud9 workspace main work area, click on the green plus (+) sign and create a **New File**.
2. Copy over the contents of the attached Part 2 install script and save the file as part2.sh. This script contains the EKS cluster creation procedure.
3. Review and modify the script variables as needed. Check the variable AVAILABILITYZONES to be sure that the listed AZs are available in your region.Some regions, such as Montreal and Tokyo do not have AZs A,B,C available (Canada is A,B,D and Tokyo is A,C,D).
4. In the Console tab, start the script:

. part2.sh

# PART 3: Configure your EKS cluster

In this step, configure the EKS cluster for the daemonset solution. This step is mandatory (you can refer to the Full Install Guide if you wish to customize the process). If you did not use the Part 2 install script to create the cluster, you must set the following variables before running the Part 3 script:

AWSREGION=<region your cluster is in>

EKSCLUSTERNAME=<name of your EKS cluster, default: agones>

NODEGROUP0NAME=<name of the nodegroup that will be used as a template for the FleetIQ-based nodegroup, default: ng-0>

* 1. In the Cloud9 workspace main work area, click on the green plus (+) sign and create a **New File**.
  2. Copy over the contents of the attached Part 3 install script and save the file as part3.sh. This script contains the EKS cluster configuration procedure.
  3. Review script variables and modify as needed. In particular, check the following variables:
* EBS volume size and type
* list of EC2 instance types to run for your game servers
* minimum and maximum autoscaling group instance count for your gameservers
  1. In the Console tab, start the script:

. part3.sh

# PART 4: Install Agones and set up an Agones fleet

In this final step, install Agones on the cluster and set up a sample fleet configuration using the open-source game Super Tux Cart. You can skip this step if you plan on running your own fleet configuration. However, you must specify the container image for your game server and apply the following configuration to the fleet for the daemonset solution:

nodeSelector:  
 role: game-servers  
tolerations:  
 - effect: NoExecute  
 key: agones.dev/gameservers  
 operator: Equal  
 value: 'true'  
 - effect: NoExecute  
 key: gamelift.status/active  
 operator: Equal  
 value: 'true'

* 1. In the Cloud9 workspace main work area, click on the green plus (+) sign and create a **New File**.
  2. Copy over the contents of the attached Part 4 install script and save the file as part4.sh. This script contains the Agones installation and sample fleet creation procedure.
  3. Review and modify the contents of the agonesfleetconfig.yaml script, including the following variables:
  + starting number of replicas
  + update strategy
  + container image
  + port allocation
  + startup commands

4. Ensure that you have the environment variable GSGNAME set (this will have been done as part of the Part 3 installation)

GSGNAME=<name of your FleetIQ GameServerGroup, default: agones-game-servers>

5. In the Console tab, start the script:

. part4.sh

# Testing Agones

Please refer to <https://agones.dev/site/docs/> for details on Agones.

Some commands you might want to run:

Check Agones Status:

kubectl describe --namespace agones-system pods

Check Agones Pod Status:

kubectl get pods --namespace agones-system

Get Fleets:

kubectl get fleets

Get Gameservers:

kubectl get gameservers

Get Gameserver details:

watch kubectl describe gameserver

Scale up Fleet:

kubectl scale fleet stk-fleet --replicas=100

watch kubectl get fleets

kubectl get nodes

Scale down Fleet:

kubectl scale fleet stk-fleet --replicas=1

watch kubectl get fleets

kubectl get nodes