

How To - ECS Jenkins Farm

Introduction -

As Novartis is using the ECS environment and we need to come up with the solution; how ECS Jenkins Farm can be created. Below mentioned example showing to create a ECS Jenkins Farm with our existing Jenkins Master which is deployed on "Devops Plateform".

1. ECS Cluster Creation -

Go to AWS Dashboard Search for ECS Click on Create Cluster then Select Cluster Templates as below.

Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

Networking only

Resources to be created:

- Cluster
- VPC (optional)
- Subnets (optional)

Powered by AWS Fargate

EC2 Linux + Networking

Resources to be created:

- Cluster
- VPC
- Subnets
- Auto Scaling group with Linux AMI

EC2 Windows + Networking

Resources to be created:

- Cluster
- VPC
- Subnets
- Auto Scaling group with Windows AMI

In this example, we will talk about ECS with EC2 Linux + Networking. Select "EC2 Linux + Networking" then click on "Next Step".

Configure cluster

Cluster name*



☐ Create an empty cluster

Instance configuration

Provisioning Model ☒ On-Demand Instance

With On-Demand Instances, you pay for compute capacity by the hour, with no long-term commitments or upfront payments.

☐ Spot

Amazon EC2 Spot Instances allow you to bid on spare Amazon EC2 computing capacity for up to 90% off the On-Demand price. [Learn more](#)

EC2 instance type*

m5ad.large



☐ Manually enter desired instance type

Number of instances*

1



EC2 Ami Id*

Amazon Linux 2 AMI [ami-08fa2...



EBS storage (GiB)*

22



Key pair

None - unable to SSH



You will not be able to SSH into your EC2 instances without a key pair. You can create a new key pair in the [EC2 console](#).

Cluster Name: Provide any meaning full name to this cluster

Provision Model: How you want to provision your EC2, so "On-Demand Instance" will be fine for this example.

EC2 Instance Type: Choose instance type, for this example I choose t2.medium.

Number of Instances: How many instances are required in this cluster.

EC2 AMI ID: Choose Ec2 AMI ID; there are two options Amazon Linux 1 and Amazon Linux 2. So choose the latest one Amazon Linux 2.

EBS Storage: EBS volume size, lets choose 30GB.

Key Pair: Choose the key pair to login into EC2 instances.

Networking

Configure the VPC for your container instances to use. A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You can choose an existing VPC, or create a new one with this wizard.

| | | |
|-------------------------------------|-------------------------------|----------|
| VPC | Create a new VPC ▾ | ↺ ⓘ |
| CIDR block | 10.0.0.0/16 | ⓘ |
| Subnet 1 | 10.0.0.0/24 | ✖ ⓘ |
| Subnet 2 | 10.0.1.0/24 | ✖ |
| + Add more subnets. | | |
| Security group | Create a new security gr... ▾ | ↺ ⓘ |
| Security group inbound rules | CIDR block ⓘ | |
| | 0.0.0.0/0 | |
| | Port range | Protocol |
| | 80 | tcp |

VPC: Need to choose VPC, however for this example "Novartis-devops-vpc" selected.

Subnets: Same subnet selected, which is shared with Jenkins Master.

Security Groups: Same SG selectd, which are selected for Jenkins Master.

Container instance IAM role

The Amazon ECS container agent makes calls to the Amazon ECS API actions on your behalf, so container instances that run the agent require the `ecsInstanceRole` IAM policy and role for the service to know that the agent belongs to you. If you do not have the `ecsInstanceRole` already, we can create one for you.

Container instance IAM role ecsInstanceRole ⓘ

Tags

| Key | Value |
|--------------------------------------|--|
| <input type="text" value="Add key"/> | <input type="text" value="Add value"/> |

CloudWatch Container Insights

CloudWatch Container Insights is a monitoring and troubleshooting solution for containerized applications and microservices. It collects, aggregates, and summarizes compute utilization such as CPU, memory, disk, and network; and diagnostic information such as container restart failures to help you isolate issues with your clusters and resolve them quickly. [Learn more](#)

CloudWatch Container Insights ☐ Enable Container Insights

*Required

[Cancel](#)

[Previous](#)

[Create](#)

Select IAM Role as "ecsInstanceRole". If cluster is being created first time then leave this field as default then it will automatically create the IAM Role as "ecsInstanceRole". Then click on Create button.

ECS status - 3 of 3 complete **myjenkins**

- ✓ **ECS cluster**
ECS Cluster myjenkins successfully created
- ✓ **ECS Instance IAM Policy**
IAM Policy for the role ecsInstanceRole successfully attached
- ✓ **CloudFormation Stack**
CloudFormation stack EC2ContainerService-myjenkins and its resources successfully created

Cluster Resources

| | |
|-----------------------------|--|
| Instance type | t2.micro |
| Desired number of instances | 1 |
| Key pair | ECS-Jenkins |
| ECS AMI ID | ami-0c65e6401a50512c5 |
| VPC | vpc-0649acbf2781cc87 |
| Subnets | subnet-0dac5463acfcfb234 |
| VPC Availability Zones | us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f |
| Security group | sg-0e4ed769ee611a28b |
| Launch configuration | EC2ContainerService-myjenkins-EcsInstanceLc-BN0QK98AN78U |
| Auto Scaling group | EC2ContainerService-myjenkins-EcsInstanceAsg-PS12MEA6M18Z |

Cluster is created successfully and all the details are mentioned here.

Cluster : JekinsCluster

Delete Cluster

Get a detailed view of the resources on your cluster.

Status **ACTIVE**

Registered container instances 1

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 0 Fargate, 0 EC2

Active service count 0 Fargate, 0 EC2

Draining service count 0 Fargate, 0 EC2

Services Tasks **ECS Instances** Metrics Scheduled Tasks Tags

Create Update Delete Actions ▾

Last updated on November 4, 2019 9:14:40 PM (0m ago) ↺ ⚙ ⓘ

Filter in this page Launch type ALL Service type ALL

| <input type="checkbox"/> | Service Name | Status | Service typ... | Task Definit... | Desired tas... | Running ta... | Launch typ... | Platform ve... |
|--------------------------|--------------|--------|----------------|-----------------|----------------|---------------|---------------|----------------|
| No results | | | | | | | | |

Services Tasks **ECS Instances** Metrics Scheduled Tasks Tags

Add additional ECS Instances using [Auto Scaling](#) or [Amazon EC2](#).

Actions ▾

Last updated on November 4, 2019 9:16:28 PM (0m ago) ↺ ⚙ ⓘ

Status: **ALL** ACTIVE DRAINING < 1-1 > Page size 50 ▾

Filter by attributes (click or press down arrow to view filter options)

| <input type="checkbox"/> | Container Instance | EC2 Instance | Availability Zo... | Agent Connec... | Status | Running tasks... | CPU available ... | Memory availa... | Age |
|--------------------------|----------------------------|-------------------|--------------------|-----------------|--------|------------------|-------------------|------------------|------|
| <input type="checkbox"/> | fe85e267-e29a-445b-a163... | i-0c7c571add37... | us-east-1a | true | ACTIVE | 0 | 2048 | 3945 | 1.32 |

You can see the Container and EC2 Instances created under this cluster.

2. Jenkins Configuration for ECS -

2.1 Jenkins Plugins -


Some plugins required to integrate ECS with the Jenkins Master; below mentioned plugin are required to install.

Installing Plugins/Upgrades


Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success


Amazon Web Services SDK

 Success

CloudBees AWS Credentials

 Success

Amazon EC2 Container Service plugin with autoscaling capabilities

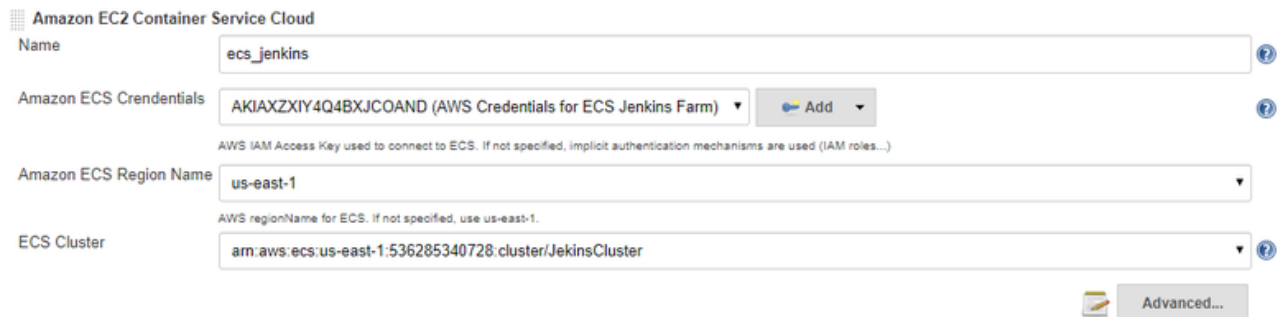
 Success

➡ [Go back to the top page](#)
(you can start using the installed plugins right away)

➡ ☐ Restart Jenkins when installation is complete and no jobs are running

2.2 Jenkins Configuration -

Go to Manage Jenkins > Configure System > Add a New Cloud > select "Amazon EC2 Container Service Cloud".



Amazon EC2 Container Service Cloud

Name:

Amazon ECS Credentials: Add

Amazon ECS Region Name:

ECS Cluster:

Advanced...

Name: Give any logical name this this configuration.

Amazon ECS Credentials: Provide the AWS "Access key ID" and "Secret access key" to authenticate.

Amazon ECS Region Name: Provide the same region name where ECS Cluster is created.

ECS Cluster: Once ECS Credentials and Region Name are entered correctly the It will start showing all the ECS Cluster here; just select the desired one.

If you are running jenkins master under an ELB, you need to add the tunnel configuration in the advanced section.

The Tunnel connection through option should have the elb URL followed by the JNLP port as shown below.

Amazon EC2 Container Service Cloud

Name

Amazon ECS Credentials [Add](#)

AWS IAM Access Key used to connect to ECS. If not specified, implicit authentication mechanisms are used (IAM roles...)

Amazon ECS Region Name

AWS regionName for ECS. If not specified, use us-east-1.

ECS Cluster

Tunnel connection through

Alternative Jenkins URL

If needed, the Jenkins URL can be overwritten with this property (e.g. to support other HTTP(S) endpoints due to reverse proxies or firewalling). By default the URL from the global Jenkins configuration is used.

ECS task creation timeout

Timeout (in second) for ECS task to be created, usefull if you use large docker slave image, because the host will take more time to pull the docker image

Now need to configure the ECS Slave Templates-

ECS slave templates

Label

Docker Image

Filesystem root

Memory

CPU units

[Advanced...](#)

[Delete](#)

Label: Mention label, so desired the jenkins job needs to run for this label.

Docker Images: Mention the Docker Image; which you want to invoke as a Jenkins Slave.

FileSystem Root: For Jenkins Slave it should be "/home/jenkins".

Memory: Mention memory here to provide for Slave. For this example, 2048 is mentioned. If Memory is not provided then It will give and error during Save.

CPU Limits: CPU Unit can be provided according to job and server's configuration.

3. Tasks Definition and Metrics in ECS-

Now run any Freestyle Job or Pipeline with label i.e "ecs", so it will trigger the job on ECS environment. You can see the same in Amazon ECS as below, it is triggering the job. Once Job is done then this task will be finished.

Services

Tasks

ECS Instances

Metrics

Scheduled Tasks

Tags

Run new Task

Stop

Stop All

Actions

Last updated on November 5, 2019 11:13:29 AM (0m ago)

Desired task status:

Running

Stopped

Filter in this page

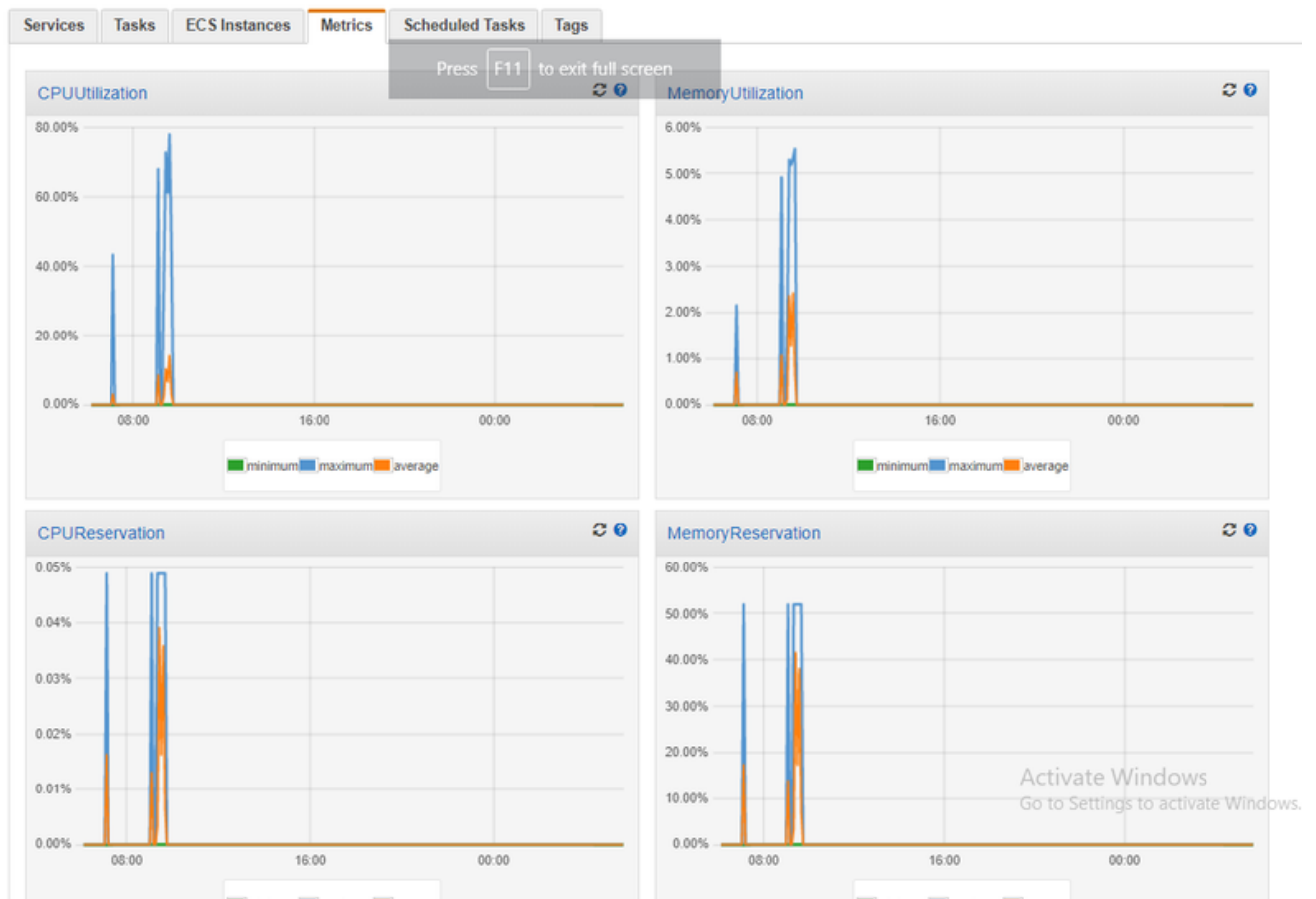
Launch type

ALL

< 1-1 > Page size 50

| <input type="checkbox"/> | Task | Task definitio... | Container inst... | Last status | Desired statu... | Started By | Group | Launch type | Platform versi... |
|--------------------------|------------------|-------------------|-------------------|-------------|------------------|------------|---------------------|-------------|-------------------|
| <input type="checkbox"/> | b2e6b16f-8de6... | jenkins-slave 6 | fe85e267-e29a... | PENDING | RUNNING | | family:jenkins-s... | EC2 | -- |

In Metrics tab, you can see how much resources have been taken for this job and other jobs-



Any Job execution will create the Task Definition under Amazon ECS as below-

Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use. [Learn more](#)

Create new Task Definition

Create new revision

Actions

Last updated on November 5, 2019 11:16:04 AM (12m ago)

Status: ACTIVE INACTIVE

Filter in this page

< 1-3 > Page size 50

| <input type="checkbox"/> | Task Definition | Latest revision status |
|--------------------------|-----------------|------------------------|
| <input type="checkbox"/> | jenkins-slave | ACTIVE |

If we click on jenkins-slave then it will show how many jobs and revisions are Active-

Task Definition Name : jenkins-slave

Select a revision for more details

[Create new revision](#)
[Actions](#)
Last updated on November 5, 2019 11:30:30 AM (0m ago)

Status: [Active](#) [Inactive](#)

< 1-6 > Page size 50

| <input type="checkbox"/> | Task Definition Name : Revision | Status |
|--------------------------|---------------------------------|--------|
| <input type="checkbox"/> | jenkins-slave:6 | Active |
| <input type="checkbox"/> | jenkins-slave:5 | Active |
| <input type="checkbox"/> | jenkins-slave:4 | Active |
| <input type="checkbox"/> | jenkins-slave:3 | Active |
| <input type="checkbox"/> | jenkins-slave:2 | Active |
| <input type="checkbox"/> | jenkins-slave:1 | Active |

4. Auto Scaling for ECS Jenkins Farm -

Auto Scaling Group is default created along with the ECS Cluster.

[Create Auto Scaling group](#)
[Actions](#)

< 1 to 1 of 1 Auto Scaling Groups >

| <input checked="" type="checkbox"/> | Name | Launch Configuration / | Instances | Desired | Min | Max | Availability Zones | Default Cooldown | Health Check Grac |
|-------------------------------------|-----------------|--------------------------|-----------|---------|-----|-----|--------------------|------------------|-------------------|
| <input checked="" type="checkbox"/> | EC2Container... | EC2ContainerService-J... | 1 | 1 | 1 | 4 | us-east-1a | 5 | 0 |

This Auto Scaling Group can be edited according to our requirement like how many instances are required (Desired, Min and Max) -



Edit details - EC2ContainerService-JekinsCluster-EcsInstanceAsg-1BYTQWEFYPM85

Launch Instances Using ⓘ

- ☐ Launch Template
☒ Launch Configuration

Launch Configuration ⓘ

EC2ContainerService-JekinsCluster-EcsInstanceLc-... ▾

Desired Capacity ⓘ

1

Min ⓘ

1

Max ⓘ

4

Availability Zone(s) ⓘ

us-east-1a ✕

Subnet(s) ⓘ

subnet-0dac5463acfcfb234(10.0.0.0/27) | us-east-1a ✕

Classic Load Balancers ⓘ

Target Groups ⓘ

Health Check Type ⓘ

EC2 ▾

Health Check Grace Period ⓘ

0

Instance Protection ⓘ

Termination Policies ⓘ

Default ✕

Suspended Processes ⓘ

Placement Groups ⓘ

Cancel

Save

Can also create the Scaling Policy with respect to our requirement -

Create Scaling policy

CancelCreate

Name:

Metric type:

Average CPU Utilization

Target value:

Instances need:

5

seconds to warm up after scaling

Disable scale-in:

☐

Create a simple scaling policy ⓘ

Create a scaling policy with steps ⓘ

5. References -

<https://devopscube.com/setup-ecs-cluster-as-build-slave-jenkins/>

<https://tech.ticketfly.com/our-journey-to-continuous-delivery-chapter-4-run-jenkins-infrastructure-on-aws-container-service-ef37e0304b95>