

# Deployment 7

We need to create a container with Jenkins, to build a Java app onto a container hosted by an EC2.

Creating the EC2 using the Ubuntu AMI on a public subnet:

- EC2 instances and launch instance
- Select UBUNTU AMI, t2 micro, Default VPC, Public Subnet

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

**Number of instances** ⓘ 1 [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ ☐ Request Spot instances

**Network** ⓘ vpc-e7d9a89a (default) [Create new VPC](#)

**Subnet** ⓘ subnet-94bc98f2 | Default in us-east-1a [Create new subnet](#)  
4090 IP Addresses available

**Auto-assign Public IP** ⓘ Use subnet setting (Enable)

**Placement group** ⓘ ☐ Add instance to placement group

**Capacity Reservation** ⓘ Open

**Domain join directory** ⓘ No directory [Create new directory](#)

**IAM role** ⓘ None [Create new IAM role](#)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

- Configure Security group 22, 8080
- Go to your created instance, Instance ID, Connect, SSH client and execute this ssh commands


EC2 Instance Connect

Session Manager

SSH client

EC2 Serial Console


Instance ID

 i-0cd54bb8aacd13415


1. Open an SSH client.

2. Locate your private key file. The key used to launch this instance is dockerec2.pem


3. Run this command, if necessary, to ensure your key is not publicly viewable.

 `chmod 400 dockerec2.pem`

4. Connect to your instance using its Public DNS:

 Command copied

8-185-246.compute-1.amazonaws.com

 `ssh -i "dockerec2.pem" ubuntu@ec2-44-198-185-246.compute-1.amazonaws.com`

- ssh into public ec2
- sudo apt-get update
- Sudo apt-get upgrade

Now we need install necessary packages Java, as well as Docker, installed on this EC2

<https://docs.docker.com/engine/install/ubuntu/>

(or we can follow this link

<https://dev.to/kamalhossain/how-to-run-docker-containers-in-aws-ec2-3bh0>)

Sudo apt-get update

sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"

sudo apt update

sudo apt-cache policy docker-ce

sudo apt install docker-ce

sudo systemctl status docker

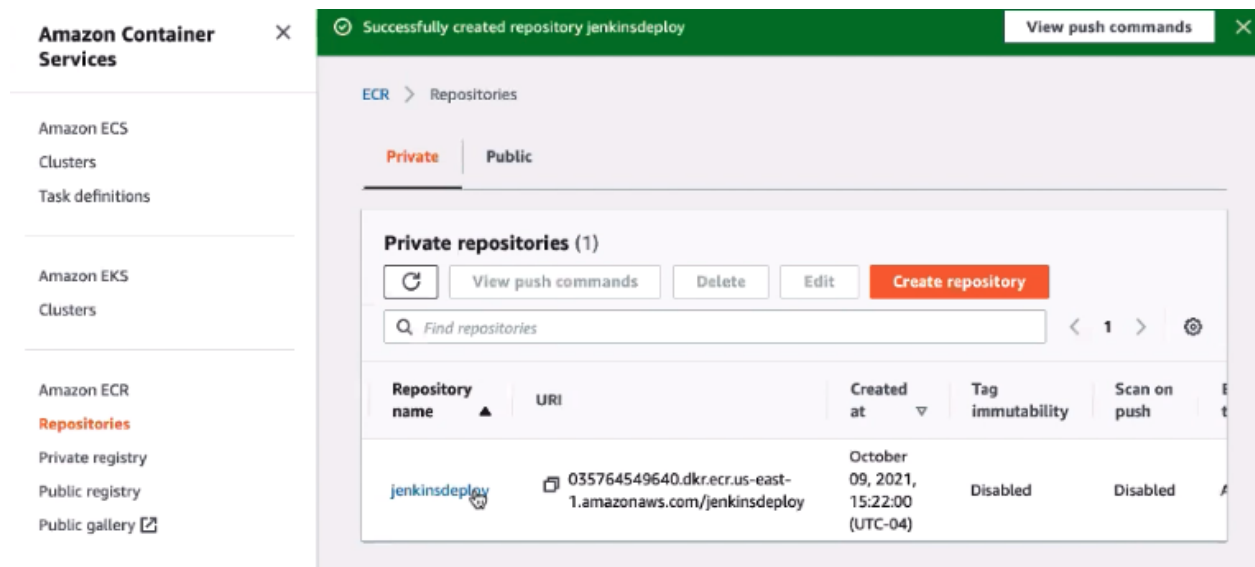
sudo usermod -aG docker ubuntu

sudo docker status

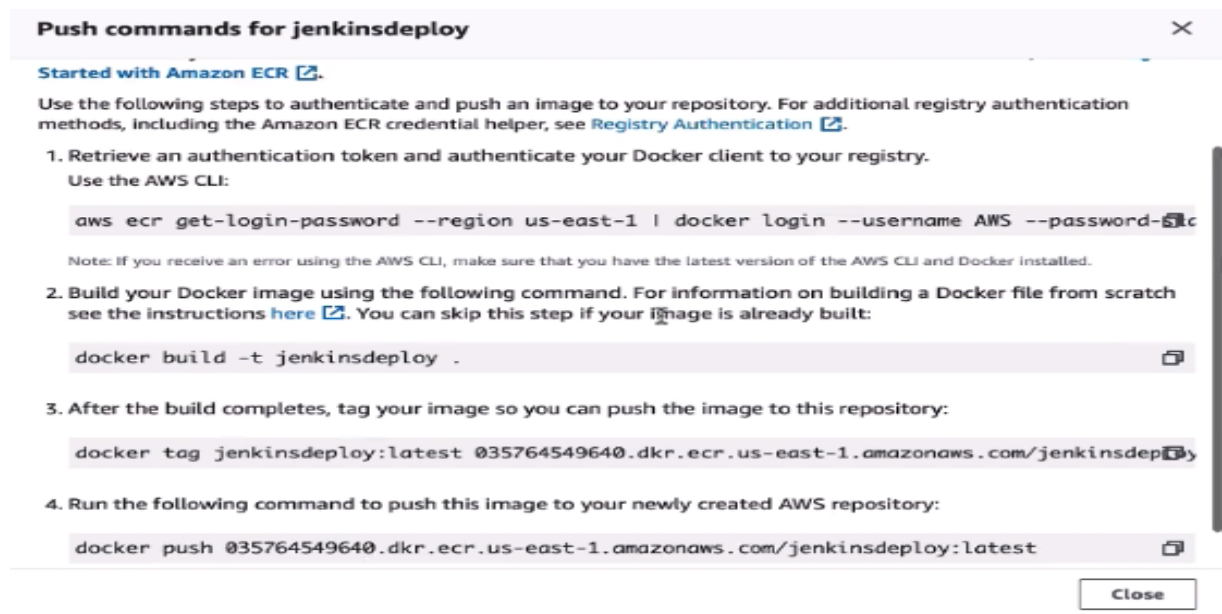
## Create the ECS Cluster

On the ECS dashboard (Amazon Container Services), create a new repository under ECR.

- Select Private (we making private repository so we can utilize for Jenkin image for ECS) , enter your repository name, click Create Repository.



## View push commands



We can run step 1, step 3 just change to jenkins/jenkins (docker tag jenkins/jenkins 03576549640.dkr.ecr.us-east-1.amazonaws.com/jenkins) step 4.

In your local terminal, get a Jenkins Docker image and push it to your ECR repository.

```
docker pull jenkins/jenkins
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS
--password-stdin public.ecr.aws/repo-tag
docker tag image-name:tag-name public.ecr.aws/repo-tag/repo-name:tag-name
docker push public.ecr.aws/repo-tag/repo-name:tag-name
```

## Github

Let's create GitHub repository for Jenkins following instruction with three files:

[https://github.com/DIrisova/DEPLOY07\\_ECS/blob/main/Dockerfile](https://github.com/DIrisova/DEPLOY07_ECS/blob/main/Dockerfile)

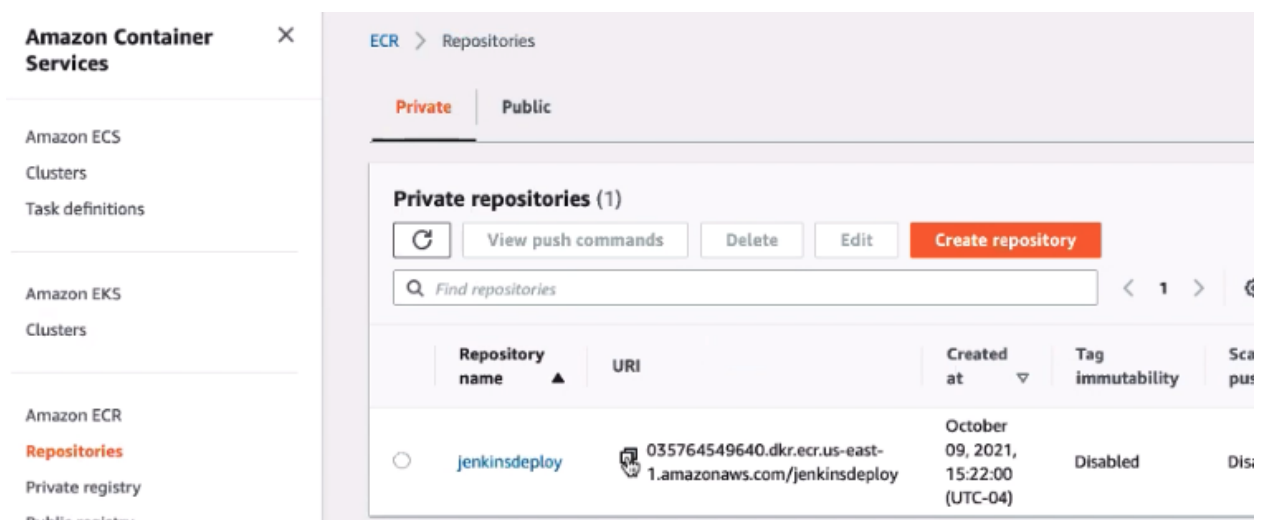
- The .jar file will hold the information to build our Java app.
- The Dockerfile will build an image of our Java app.
- The Jenkinsfile will run the Dockerfile and push our image to Dockerhub. Paste the code from the task pdf into the Jenkinsfile

```
pipeline {
  agent { label "your label" }
  environment {
    DOCKERHUB_CREDENTIALS = credentials('your username-dockerhub')
  }
  stages {
    stage ('Build') {
      steps {
        sh '''docker .....'''
      }
    }
    stage ('Login') {
      steps {
        sh '''echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u
$DOCKERHUB_CREDENTIALS_USR --password-stdin'''
      }
    }
    stage ('Push') {
      steps {
        sh '''docker .....'''
      }
    }
  }
}
```

- Drop the jar file into your repository
- Generate token from dockerhub
- Follow the steps provided by dockerhub when we generate your token..
- We should add the token as a password to our jenkins global credentials

Create a cluster.

- Select Networking Only, name the cluster, create the cluster.



- Next, create a New Task Definition. When creating the container use the image URI from our Jenkins image we pushed earlier to ECR earlier.
- Launch status, View Cluster, Task Definitions, Create new Task Definitions, FARGATE, Task definition name, Task memory we can select minimum or 1GB. Add container, for Image copy and past the image URI from our Jenkins image we pushed earlier to ECR earlier.

Amazon Container Services

Amazon ECS

Clusters

Task definitions

Amazon EKS

Clusters

Amazon ECR

Repositories

**Images**

Permissions

ECR > Repositories > jenkinsdeploy

jenkinsdeploy

View push commands Edit

Images (1) Refresh Delete Scan

Find images

Image tag

Pushed at

Size (MB)

Image URI

Digest

latest

October 09, 2021, 15:33:50 (UTC-04)

266.09

Copy URI

sha256:a9c3653...

Add container

Standard

Container name\*

jenkin

Container name is required.

Image\*

035764549640.dkr.ecr.us-east-1.amazonaws.com/jenkinsdeploy:latest

- Add
- View task definition
- Clusters, go to Deploy7

New ECS Experience

Tell us what you think

Amazon ECS

**Clusters**

Task Definitions

Account Settings

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Clusters

An Amazon ECS cluster is a regional group that receives a default cluster the first time you use it.

For more information, see the [ECS documentation](#).

Create Cluster Get Started

View list card

Deploy7 >

CloudWatch monitoring

Default Monitoring

FARGATE

- Tasks
- Run new Task

- Task Definitions
- Account Settings
- Amazon EKS
- Clusters
- Amazon ECR
- Repositories
- AWS Marketplace
- Discover software
- Subscriptions [↗](#)

Get a detailed view of the resources on your cluster.

Cluster ARN: `arn:aws:ecs:us-east-1:035764549640:cluster/Deploy7`

Status: **ACTIVE**

Registered container instances: 0

Pending tasks count: 0 Fargate, 0 EC2, 0 External

Running tasks count: 0 Fargate, 0 EC2, 0 External

Active service count: 0 Fargate, 0 EC2, 0 External

Draining service count: 0 Fargate, 0 EC2, 0 External

Services | **Tasks** | ECS Instances | Metrics | Scheduled Tasks | Tags | Capacity Providers

[Run new Task](#) | [Stop](#) | [Stop All](#) | [Actions](#) | Last updated on October 9, 2021 3:48:52 PM (0m ago) [↻](#)

Desired task status: **Running** | Stopped

Filter in this page | Launch type: ALL

<input type="checkbox"/>	Task	Task d...	Contal...	Last st...	Desire...	Started...	Started...	Group ...	Launch...	Platfor
No results										

- Launch type, FARGATE

Launch type ☒ FARGATE ☐ EC2 ☐ EXTERNAL [?](#)

Select VPC and Subnets (should be same subnet when we created EC2)

### VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Cluster VPC\* `vpc-e7d9a89a (172.31.0.0/16)` [?](#)

Subnets\* `subnet-94bc98f2 (172.31.0.0/20) - us-east-1a assign ipv6 on creation: Disabled` [?](#)

Security groups\* `taskfo-6226` [Edit](#) [?](#)

- Security group, Edit,

Configure security groups

A security group is a set of firewall rules that control the traffic for your task. On this page, you can add rules to allow specific traffic to reach your task, or you can choose to use an existing security group. [Learn more.](#)

Assigned security groups
☒ Create new security group
☐ Select existing security group

Security group name\*

Description

Inbound rules for security group

Type	Protocol	Port range	Source
Cus...	TCP	8080	Anywhere


- 
- Run Task
- So task running, click on Task, copy Public IP
- 

Amazon ECH

Repositories

AWS Marketplace

Discover software

Subscriptions 

Task definition [taskfordeploy7:1](#)

Group family:taskfordeploy7

Task role None

Last status **RUNNING**

Desired status RUNNING

Created at 2021-10-09 15:53:09 -0400

Started at 2021-10-09 15:53:45 -0400

## Network

Network mode awsvpc

ENI id [eni-02668cc513fed7431](#)

Subnet id subnet-94bc98f2

Private IP 172.31.4.55

Public IP [35.175.118.155](#)

Mac address 02:7f:19:cc:39:57

## Containers





- New Item, Enter an Item name, Multibranch Pipeline

These are installed add an agent hosted by our EC2 (use private IP).

Add credentials:

- We need to add Github login using the Personal Access Token as the password.
- We need to add Docker login using the Personal Access Token as the password.
- We need to add SSH login using the IP and .pem file for the agent EC2.
- Create a multi-branch pipeline that pulls from the repo you set up.
- Run it and check your Dockerhub to see if the Java app image has been uploaded.

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, a notification bell with 1 alert, the user name 'Dilobar', and a 'log out' button. The breadcrumb trail is 'Dashboard > Deployment7 > main'. The left sidebar contains links for 'Up', 'Status', 'Changes', 'Build Now' (highlighted), 'View Configuration', 'Full Stage View', and 'Pipeline Syntax'. Below this is the 'Build History' section with a 'trend' button and a search bar for builds. A list of builds is shown, with build #30 being the current one.

### Pipeline main

Full project name: Deploy7/main

Recent Changes

### Stage View

Average stage times: (Average full run time: ~30s)

	Declarative: Checkout SCM	Build	Login	Push
<b>#30</b> Oct 9, 2021, 11:24 PM 1 commit	540ms	21s	969ms	4s
<b>#29</b> Oct 9, 2021, 11:22 PM No Changes	456ms	1s failed	103ms failed	170ms failed
<b>#28</b> Oct 9, 2021, 11:18 PM No Changes	425ms			
<b>#27</b> Oct 9, 2021, 11:10 PM 2 commits	601ms			