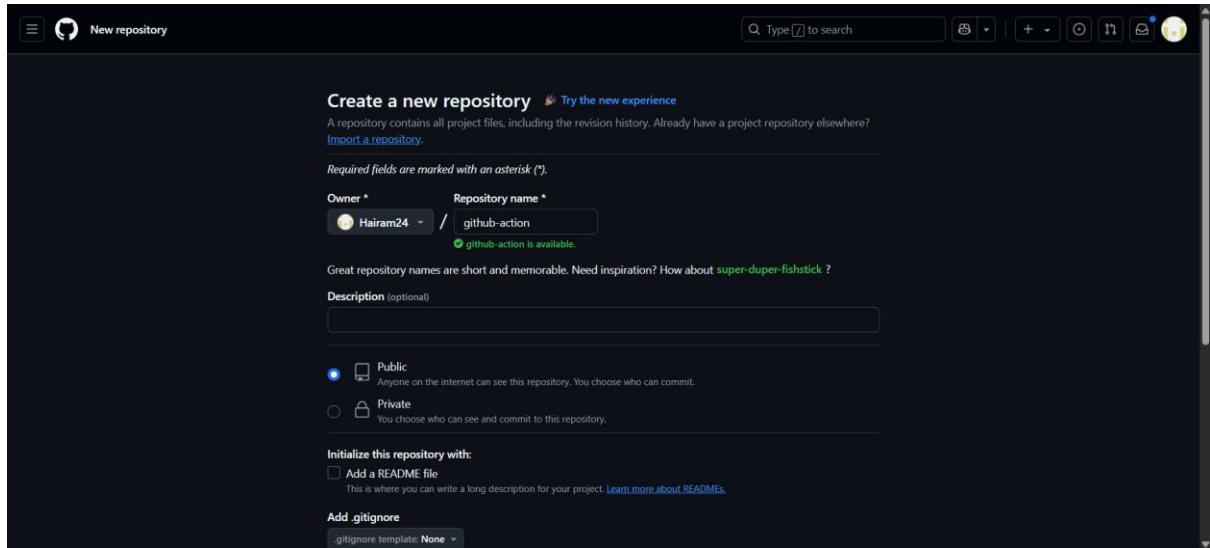
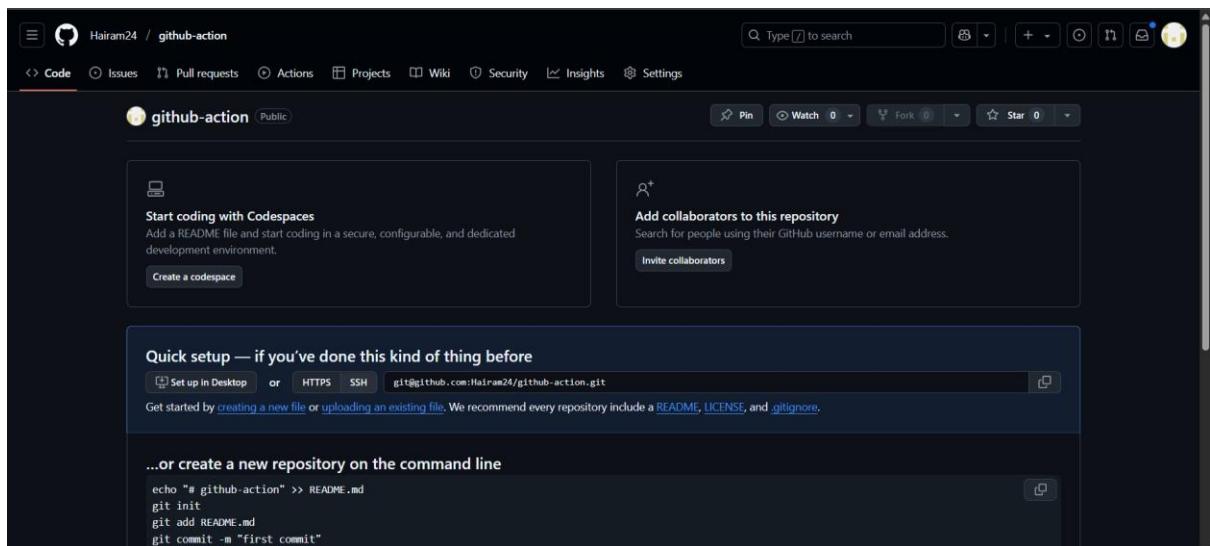


## 1) Easy

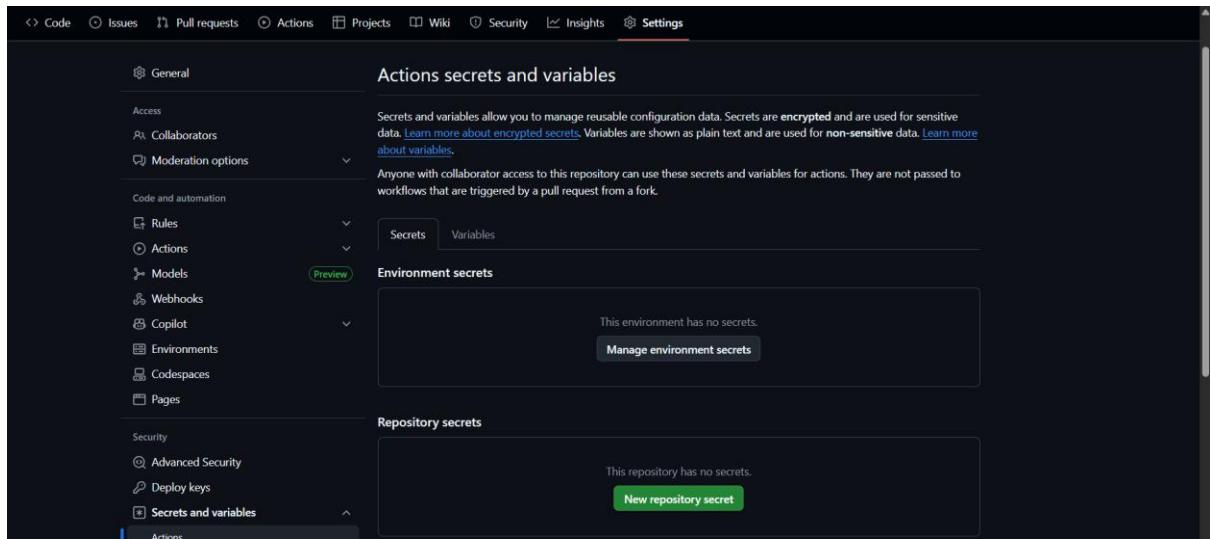
Using CI/CD tool GitHub actions deploy a static web site on AWS S3. And website should be expose publicly



New public repo is created.

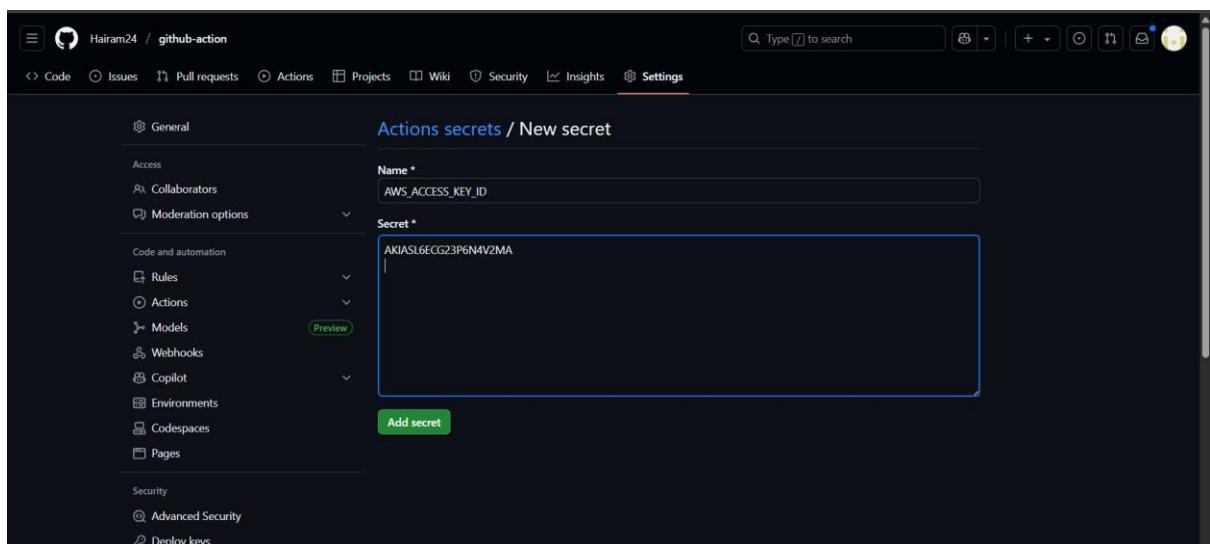


Repository successfully created.



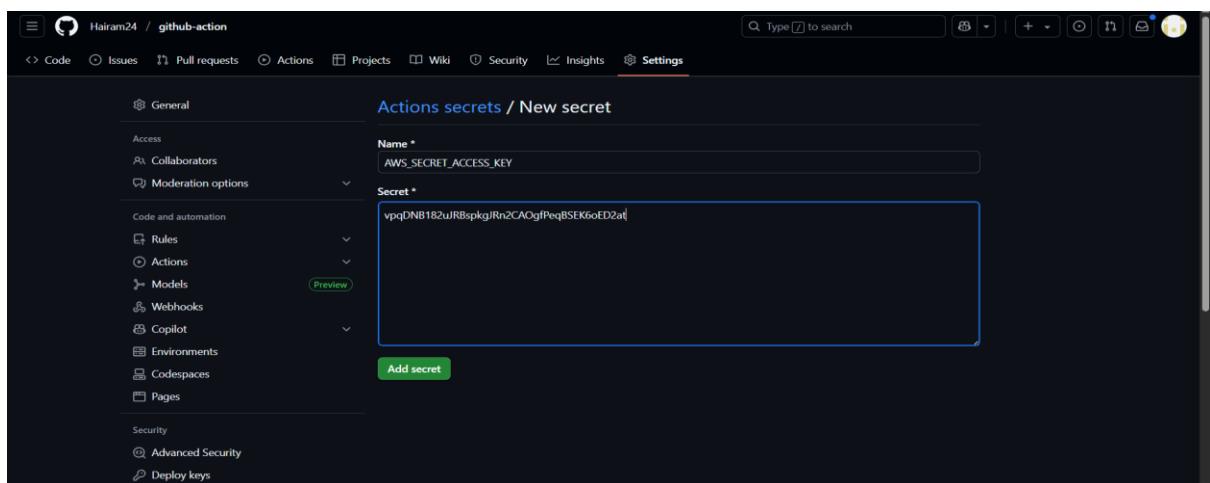
The screenshot shows the GitHub Actions settings page for a repository. The left sidebar contains sections for General, Access, Collaborators, Moderation options, Code and automation (with Rules, Actions, Models, Webhooks, Copilot, Environments, Codespaces, and Pages), Security (with Advanced Security, Deploy keys, and Secrets and variables), and Actions. The main content area is titled "Actions secrets and variables". It includes a section for "Environment secrets" which states "This environment has no secrets." and a "Manage environment secrets" button. Another section for "Repository secrets" states "This repository has no secrets." and a "New repository secret" button.

To add keys > secrets and variables> actions.



The screenshot shows the "Actions secrets / New secret" creation page. The left sidebar is identical to the previous screenshot. The main form has "Name \*" set to "AWS\_ACCESS\_KEY\_ID" and "Secret \*" containing the value "AKIASL6ECG23P6N4V2MA". A green "Add secret" button is at the bottom.

Added the aws\_acces\_key



The screenshot shows the "Actions secrets / New secret" creation page again. The left sidebar is identical. The main form now has "Name \*" set to "AWS\_SECRET\_ACCESS\_KEY" and "Secret \*" containing the value "vpgDNB182u/RBspkgjRn2CAOgIPeqBSEK6oED2aI". A green "Add secret" button is at the bottom.

Added the aws\_secret\_access\_key.

The screenshot shows the GitHub 'Moderation options' interface under the 'Code and automation' section. In the 'Repository secrets' section, two secrets are listed:

- AWS\_ACCESS\_KEY\_ID**: Last updated now, with edit and delete icons.
- AWS\_SECRET\_ACCESS\_KEY**: Last updated now, with edit and delete icons.

Both keys are added.

The screenshot shows the 'Create bucket' wizard in the Amazon S3 console. Under 'General configuration', the 'Bucket name' is set to 'hari-bucket-2003'. The 'General purpose' option is selected, with a note: 'Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.' The 'Directory' option is also available with its own note: 'Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.'

New S3 bucket is created .

The screenshot shows the 'Block Public Access settings for this bucket' step in the S3 creation wizard. The 'Block all public access' checkbox is checked, with a note: 'Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.' Below are four unchecked checkboxes:
 

- Block public access to buckets and objects granted through new access control lists (ACLs)**: 'S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.'
- Block public access to buckets and objects granted through any access control lists (ACLs)**: 'S3 will ignore ACLs that grant public access to buckets and objects.'
- Block public access to buckets and objects granted through new public bucket or access point policies**: 'S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.'
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**: 'S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.'

 A warning message at the bottom states: 'Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.' A checkbox at the bottom is checked: 'I acknowledge that the current settings might result in this bucket and the objects within becoming public.'

Block public access is disabled.

The screenshot shows the AWS S3 Buckets page. A green success message at the top states: "Successfully created bucket 'hari-bucket-2003'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, there's an account snapshot section with a link to "View Storage Lens dashboard". Under "General purpose buckets", there is one entry: "hari-bucket-2003" (US East (N. Virginia) us-east-1). The creation date is July 10, 2025, 14:14:49 (UTC+05:30). Action buttons include Copy ARN, Empty, Delete, and Create bucket.

S3 bucket is created.

The screenshot shows the "Edit bucket policy" page for the "hari-bucket-2003" bucket. The policy document is displayed in JSON format:

```
1 {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Sid": "PublicReadForGetBucketObjects",
6             "Effect": "Allow",
7             "Principal": "*",
8             "Action": "s3:GetObject",
9             "Resource": "arn:aws:s3:::hari-bucket-2003/*"
10        }
11    ]
12 }
```

On the right side, there are buttons for "Edit statement", "Select a statement", and "+ Add new statement".

New bucket policy is added.

The screenshot shows the "Buckets" page for the "hari-bucket-2003" bucket. A green success message at the top states: "Successfully edited bucket policy." Below this, the bucket details are shown again, including the newly added policy.

Bucket policy is added successfully.

The screenshot shows the 'Edit static website hosting' configuration page in the AWS S3 console. Under 'Static website hosting', 'Enable' is selected. Under 'Hosting type', 'Host a static website' is selected. A note at the bottom states: 'For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see Using Amazon S3 Block Public Access.' The 'Index document' field contains 'index.html'. The 'Error document - optional' field is empty.

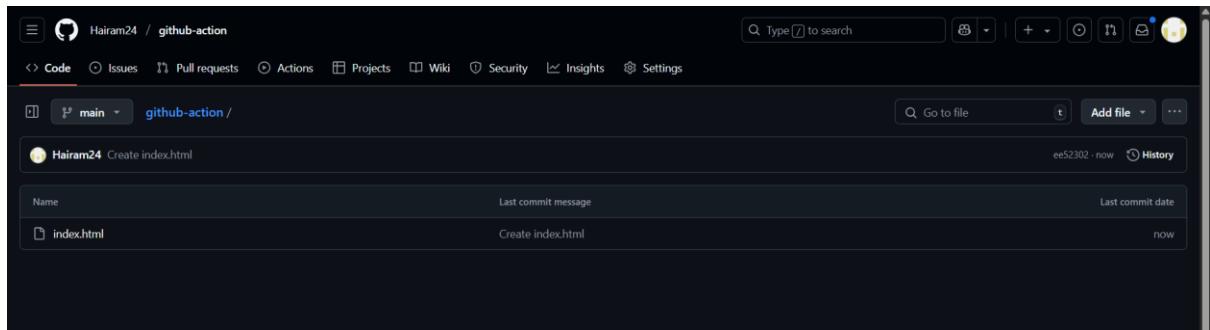
In s3 bucket , statis we hosting is enabled and index.html is added.

The screenshot shows a success message: 'Successfully edited static website hosting.' Under 'Requester pays', 'Disabled' is selected. The 'Static website hosting' section shows it is 'Enabled'. A note at the top of the section says: 'We recommend using AWS Amplify Hosting for static website hosting. Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps.' A 'Create Amplify app' button is available. The 'Bucket website endpoint' is listed as <http://hari-bucket-2003.s3-website-us-east-1.amazonaws.com>.

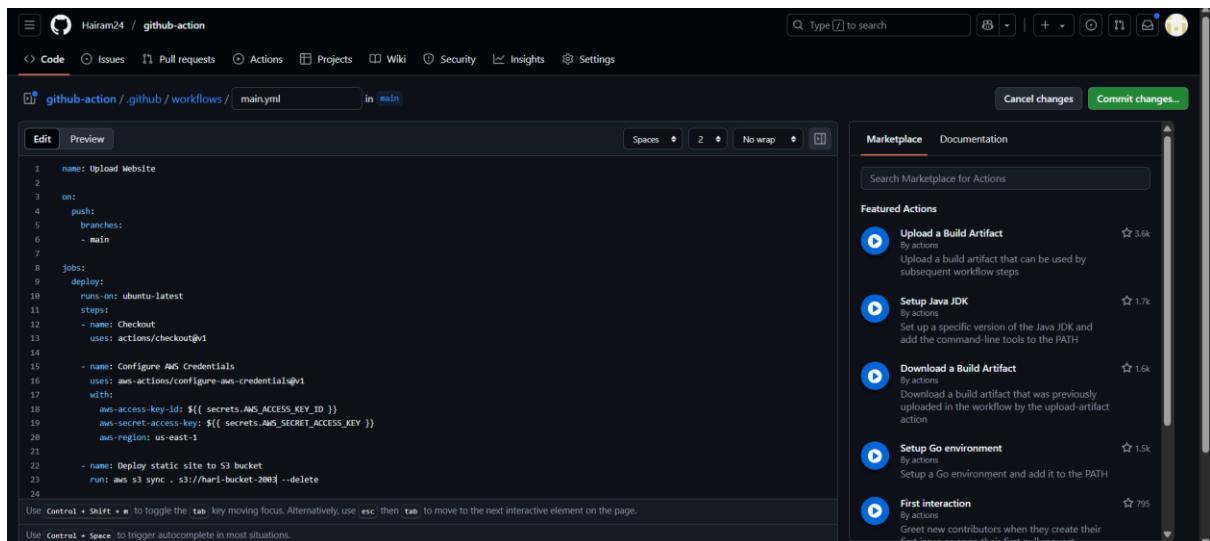
Static web hosting is enabled successfully.

The screenshot shows the GitHub Actions editor for the 'github-action' repository. The 'index.html' file is open in the main editor area. The content of the file is: 'This is my web server Hariram.' The editor has tabs for 'Edit' and 'Preview'. The preview shows the text 'This is my web server Hariram.' The GitHub navigation bar includes Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings.

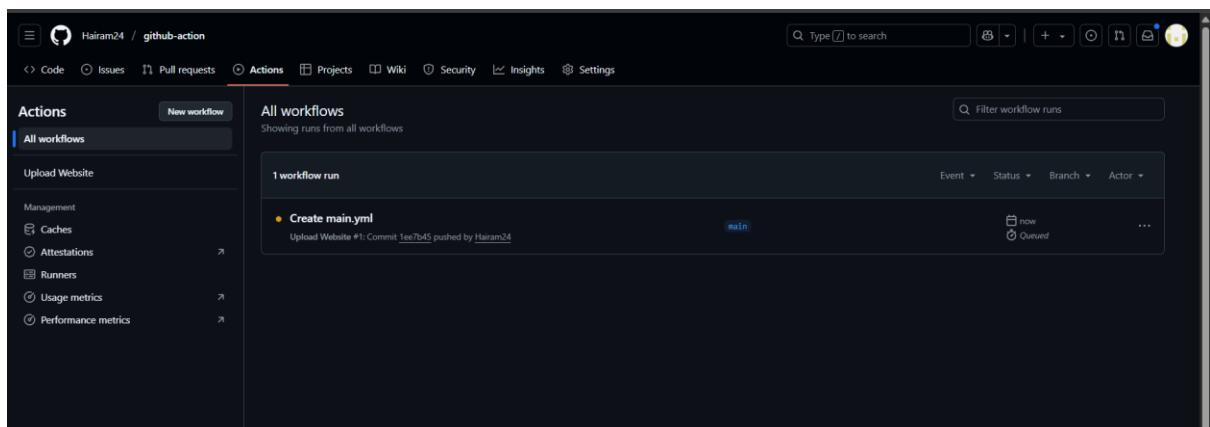
In the repo index.html file is created.



Index.html is added successfully.



Main.yaml is added with my bucket name.



Yaml file is getting executed in the actions.

The screenshot shows a GitHub Actions workflow summary for a repository named 'Create main.yml'. The workflow has one job named 'deploy' which was triggered via a push to the 'main' branch. The status of the job is 'Success' and it completed in 15 seconds. The workflow file is named 'main.yml' and contains a single step named 'deploy'. There is one warning listed under 'Annotations' regarding the use of the 'set-output' command.

Static website is deployed.

The screenshot shows a web browser window with three tabs. The active tab displays a static website from an S3 bucket at the URL 'hari-bucket-2003.s3-website-us-east-1.amazonaws.com'. The page content reads 'This is my web server Hariram.' Below the browser is a GitHub interface showing a repository named 'Create main.yml - Hairam24/github-action' with a commit message 'Create main.yml #1'.

To view the website past the static web hosting link in the browser.

Static website is hosted successfully using s3 bucket and allowed publically.

## 2) Medium

Deploy a web application in the kubernetes pod. And create a replica set. In any case load is going to increase on your replica set. increase the number of replica of the pods

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The 'Name and tags' section has 'eks-web' entered in the 'Name' field. The 'Application and OS Images (Amazon Machine Image)' section shows 'Amazon Linux' selected from a list of AMIs. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates 1 volume(s) - 8 GiB. The summary panel shows 1 instance and the software image as Amazon Linux 2023 AMI 2023.7.2... The 'Launch instance' button is visible at the bottom right.

The screenshot shows the 'Network settings' step of the 'Launch an instance' wizard. It includes fields for VPC (selected: 'vpc-0c1e9f84aa49d9d9e'), Subnet ('subnet-00608da34c56e569d'), and Auto-assign public IP ('Enable'). The 'Firewall (security group)' section shows 'Create security group' selected. The 'Common security groups' section lists 'launch-wizard-1 sg-0cd9a2b14bb27dd1a'. The summary panel remains the same as the previous step, showing 1 instance and the software image as Amazon Linux 2023 AMI 2023.7.2... The 'Launch instance' button is visible at the bottom right.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835505\OneDrive - LTIMindtree\Desktop> ssh -i "aws_Ltimindtree.pem" ec2-user@ec2-54-83-88-60.compute-1.amazonaws.com
The authenticity of host 'ec2-54-83-88-60.compute-1.amazonaws.com (54.83.88.60)' can't be established.
ED25519 key fingerprint is SHA256:ZxalDPqrxtXhrQ1yJv4Be8cw4nQOrg3KeVXGp30YMF8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-83-88-60.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

          _###_
         /####\
        /##|
       /#|   https://aws.amazon.com/linux/amazon-linux-2023
      /#|_>
     /##|/
    /##|/
   /##|/
  /##|/
 [ec2-user@ip-172-31-25-137 ~]$ sudo su -
[root@ip-172-31-25-137 ~]# hostnamectl set-hostname eks.example.com
[root@ip-172-31-25-137 ~]# bash
[root@eks ~]# yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[root@eks ~]# yum install unzip -y
Last metadata expiration check: 0:00:04 ago on Thu Jul 10 10:09:04 2025.
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@eks ~]# |
```

```
[root@eks ~]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
  % Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
          Dload  Upload   Total Spent   Left Speed
100 63.2M  100 63.2M    0     0  72.2M    0 --:--:-- --:--:-- 72.3M
Archive: awscliv2.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/README.md
  creating: aws/THIRD_PARTY_LICENSES
  creating: aws/dist/awscli/
  creating: aws/dist/docutils/
  creating: aws/dist/lib-dynload/
  inflating: aws/dist/aws
  inflating: aws/dist/aws_completer
  inflating: aws/dist/libpython3.13.so.1.0
  inflating: aws/dist/_awsctrl.abi3.so
  inflating: aws/dist/_ruamel_yaml.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/libz.so.1
  inflating: aws/dist/libbzma.so.5
  inflating: aws/dist/libbz2.so.1
  inflating: aws/dist/libffi.so.6
  inflating: aws/dist/libuuid.so.1
  inflating: aws/dist/libtinfo.so.5
  inflating: aws/dist/libreadline.so.6
  inflating: aws/dist/libsqlite3.so.0
  inflating: aws/dist/base_library.zip
  inflating: aws/dist/lib-dynload/_datetime.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_unicodedata.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_csv.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_statistics.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_contextvars.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_decimal.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_pickle.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_hashlib.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha3.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_blake2.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_md5.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha1.cpython-313-x86_64-linux-gnu.so
```

**hari info**

**Summary**

- ARN: arn:aws:iam::165083073206:user/hari
- Console access: Disabled
- Created: July 02, 2025, 11:12 (UTC+05:30)
- Last console sign-in: -
- Access key 1: AKIASL6ECG23P6N4V2MA - Active (Used today, 8 days old)
- Access key 2: Create access key

**Permissions** | Groups | **Tags** | Security credentials | Last Accessed

**Permissions policies (8)**

Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly
AmazonEC2ContainerRegistryFullAccess	AWS managed	Directly
AmazonECFullAccess	AWS managed	Directly
AmazonEKSServicePolicy	AWS managed	Directly
AmazonRoute53FullAccess	AWS managed	Directly

```
[root@eks ~]# aws configure
AWS Access Key ID [None]: AKIASL6ECG23P6N4V2MA
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: table
[root@eks ~]#
```

```
[root@ip-172-31-25-137- ~]# curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
[root@eks ~]# sudo mv /tmp/eksctl /usr/local/bin
[root@eks ~]# eksctl version
0.210.0
[root@eks ~]# curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
100 53.7M 100 53.7M 0 0 93.3M 0 --:--:-- --:--:-- 93.3M
[root@eks ~]# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[root@eks ~]# kubectl version --client
Client Version: v1.31.0
Kustomize Version: v5.4.2
[root@eks ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:EqASMcfc58+Setm4e4WqYz8a7PgjttkMYPDmGyt8/y2s root@eks.example.com
The key's randomart image is:
+--[RSA 3072]----+
|   o .o
|   . B= o
|... .*= S .
|... o=oo . .
|   . +t =o +
|   . .OE=o o
|   . +@XO.
+---[SHA256]----+
[root@eks ~]#
```

```
[root@eks ~]# eksctl create cluster --name my-cluster-hari --region us-east-1 --version 1.32 --vpc-public-subnets subnet-00608da34c56e569d,subnet-0541b7ff15bd92fd9 --without-nodegroup
2025-07-10 10:13:16 [ ] creating region us-east-1
2025-07-10 10:13:16 [ ] using existing VPC (vpc-0c1e9f84aa9fd9) and subnets (private:map[] public:map[us-east-1a:[subnet-00608da34c56e569d us-east-1a 172.31.16.0/20 0 } us-east-1b:[subnet-0541b7ff15bd92fd9 us-east-1b 172.31.32.0/20 0 }])
2025-07-10 10:13:16 [ ] custom VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configuration of VPC/subnets
2025-07-10 10:13:16 [ ] using Kubernetes version 1.32
2025-07-10 10:13:16 [ ] creating EKS cluster "my-cluster-hari" in "us-east-1" region with
2025-07-10 10:13:16 [ ]   if you encounter any issues, check CloudFormation console or try `eksctl utils describe-stacks --region=us-east-1 --cluster=my-cluster-hari`
2025-07-10 10:13:16 [ ]   API endpoint will use default of {publicAccess=true, privateAccess=false} for cluster "my-cluster-hari" in "us-east-1"
2025-07-10 10:13:16 [ ]   CloudWatch logging will not be enabled for cluster "my-cluster-hari" in "us-east-1"
2025-07-10 10:13:16 [ ]   you can enable it with `eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=my-cluster-hari`
2025-07-10 10:13:16 [ ]   default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-07-10 10:13:16 [ ]
2 sequential tasks:
  2 sequential tasks:
    1 task: { creates addons },
      wait for control plane to become ready,
    }
  }
2025-07-10 10:13:16 [ ] building cluster stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:13:16 [ ] deploying stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:13:16 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:14:16 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:15:16 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:16:16 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:17:17 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:18:17 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:19:17 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:20:17 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:21:17 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-cluster"
2025-07-10 10:21:18 [ ] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addonPodIdentityAssociations' and run `eksctl update addon`
2025-07-10 10:21:18 [ ] creating addon: vpc-cni
2025-07-10 10:21:18 [ ] successfully created addon: vpc-cni
2025-07-10 10:21:18 [ ] creating addon: kube-proxy
2025-07-10 10:21:19 [ ] successfully created addon: kube-proxy
2025-07-10 10:21:19 [ ] creating addon: coredns
2025-07-10 10:21:20 [ ] successfully created addon: coredns
2025-07-10 10:21:20 [ ] creating addon: metrics-server
2025-07-10 10:21:20 [ ] successfully created addon: metrics-server
2025-07-10 10:23:20 [ ] waiting for the control plane to become ready
2025-07-10 10:23:21 [ ] saved kubeconfig as "/root/.kube/config"
2025-07-10 10:23:21 [ ] no tasks
2025-07-10 10:23:21 [ ] all EKS cluster resources for "my-cluster-hari" have been created
2025-07-10 10:23:22 [ ] kubectl command should work with "/root/.kube/config", try "kubectl get nodes"
2025-07-10 10:23:22 [ ] EKS cluster "my-cluster-hari" in "us-east-1" region is ready
[root@eks ~]#

```

Cluster is created.

```
[root@eks ~]# eksctl create nodegroup --cluster my-cluster-hari --region us-east-1 --name my-node-group --node-ami-family Ubuntu2004 --node-type t2.small --subnet-ids subnet-0068da34c56e569d,subnet-0541b7ff15bd92fd9 --nodes 3 --nodes-min 2 --nodes-max 4 --ssh-access --ssh-public-key /root/.ssh/id_rsa.pub
2025-07-10 10:30:59 [ ] will use version 1.32 for new nodegroup(s) based on control plane version
2025-07-10 10:31:00 [ ] nodegroup "my-node-group" will use "ami-087657d77f5892006" [Ubuntu2004/1.32]
2025-07-10 10:31:00 [ ] using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-my-cluster-hari-nodegroup-my-node-group-ff:1b:22:be:47:fd:d4:d8:27:b9:44:e0:ff:88:67:03"
2025-07-10 10:31:01 [ ] nodegroup ("my-node-group") was intended (based on the include/exclude rules)
2025-07-10 10:31:01 [ ] will create a CloudFormation stack for each of 1 managed nodegroups in cluster "my-cluster-hari"
2025-07-10 10:31:01 [ ]
2 sequential tasks: { fix cluster compatibility, 1 task: { create managed nodegroup "my-node-group" } }
2025-07-10 10:31:01 [ ] checking cluster stack for missing resources
2025-07-10 10:31:01 [ ] cluster stack has no required resources
2025-07-10 10:31:01 [ ] building managed nodegroup stack "eksctl-my-cluster-hari-nodegroup-my-node-group"
2025-07-10 10:31:01 [ ] deploying stack "eksctl-my-cluster-hari-nodegroup-my-node-group"
2025-07-10 10:31:01 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-nodegroup-my-node-group"
2025-07-10 10:31:31 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-nodegroup-my-node-group"
2025-07-10 10:32:27 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-nodegroup-my-node-group"
2025-07-10 10:32:27 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-nodegroup-my-node-group"
2025-07-10 10:32:27 [ ] waiting for CloudFormation stack "eksctl-my-cluster-hari-nodegroup-my-node-group"
2025-07-10 10:35:16 [ ] no tasks
2025-07-10 10:35:16 [ ] created 0 nodegroup(s) in cluster "my-cluster-hari"
2025-07-10 10:35:16 [ ] nodegroup "my-node-group" has 3 node(s)
2025-07-10 10:35:16 [ ] node "ip-172-31-26-163.ec2.internal" is ready
2025-07-10 10:35:16 [ ] node "ip-172-31-29-97.ec2.internal" is ready
2025-07-10 10:35:16 [ ] node "ip-172-31-34-159.ec2.internal" is ready
2025-07-10 10:35:16 [ ] waiting for 1 node(s) to become ready in "my-node-group"
2025-07-10 10:35:16 [ ] nodegroup "my-node-group" has 3 node(s)
2025-07-10 10:35:16 [ ] node "ip-172-31-26-163.ec2.internal" is ready
2025-07-10 10:35:16 [ ] node "ip-172-31-29-97.ec2.internal" is ready
2025-07-10 10:35:16 [ ] node "ip-172-31-34-159.ec2.internal" is ready
2025-07-10 10:35:16 [ ] created 1 managed nodegroup(s) in cluster "my-cluster-hari"
2025-07-10 10:35:16 [ ] checking security group configuration for all nodegroups
2025-07-10 10:35:16 [ ] all nodegroups have up-to-date cloudformation templates
[root@eks ~]#

```

Node group is created.

```
apiVersion: autoscaling/v1
kind: HorizontalPodAutoscaler
metadata:
  name: k8s-autoscaler
spec:
  minReplicas: 10
  maxReplicas: 2
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: k8s-autoscaler
    targetCPUUtilizationPercentage: 10
replica.yaml" 12L, 256B

```

12, 36 All

Replica is automated based on load.

```
root@ip-172-31-25-137:~ + - x
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
|
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
22,1 All
```

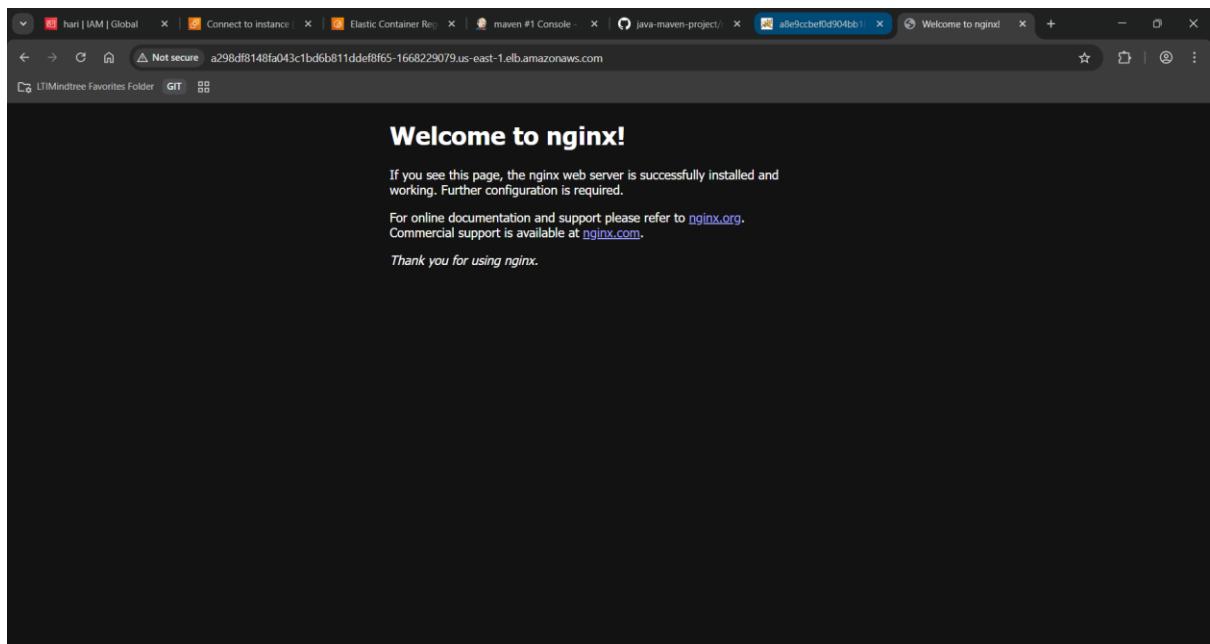
Nginx is deployed in yaml.

```
root@ip-172-31-25-137:~ % + - x
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

Service.yaml file is added.

```
[root@ip-172-31-25-137:~]# kubectl get replicaset
NAME             DESIRED   CURRENT   READY   AGE
nginx-deployment-96b9d695   3         3         3        44m
[root@eks ~]#
[root@ip-172-31-25-137:~]# vim nginx.yaml
[root@eks ~]# vim nginx-service.yaml
[root@eks ~]# kubectl apply -f nginx.yaml
deployment.apps/nginx-deployment created
[root@eks ~]# kubectl apply -f nginx-service.yaml
service/nginx-service created
[root@eks ~]# kubectl get nodes
NAME           STATUS   ROLES      AGE   VERSION
ip-172-31-26-163.ec2.internal   Ready    <none>    5m21s   v1.32.3
ip-172-31-29-97.ec2.internal   Ready    <none>    5m28s   v1.32.3
ip-172-31-34-159.ec2.internal Ready    <none>    5m12s   v1.32.3
[root@eks ~]# kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
nginx-deployment-96b9d695-b6276   1/1     Running   0          19s
nginx-deployment-96b9d695-h56g1   1/1     Running   0          19s
nginx-deployment-96b9d695-xstuj   1/1     Running   0          19s
[root@eks ~]# kubectl get svc
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP
kubernetes      ClusterIP   10.100.0.1      <none>
nginx-service   LoadBalancer 10.100.205.114  a298df8148fa043c1bd6b811ddef8f65-1668229079.us-east-1.elb.amazonaws.com
[root@eks ~]#
```

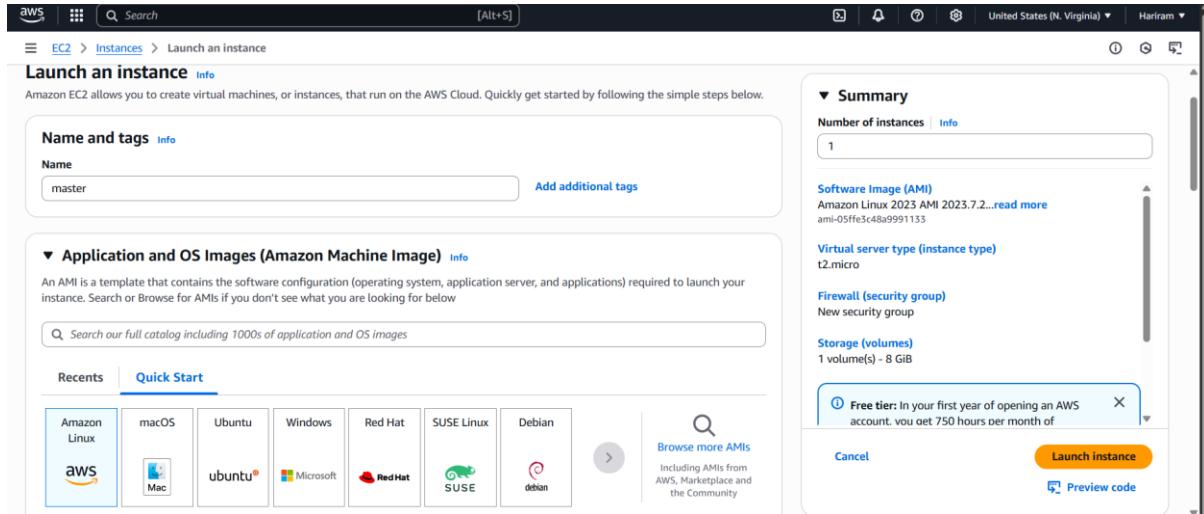
Pods are created successfully and external ip is taken.



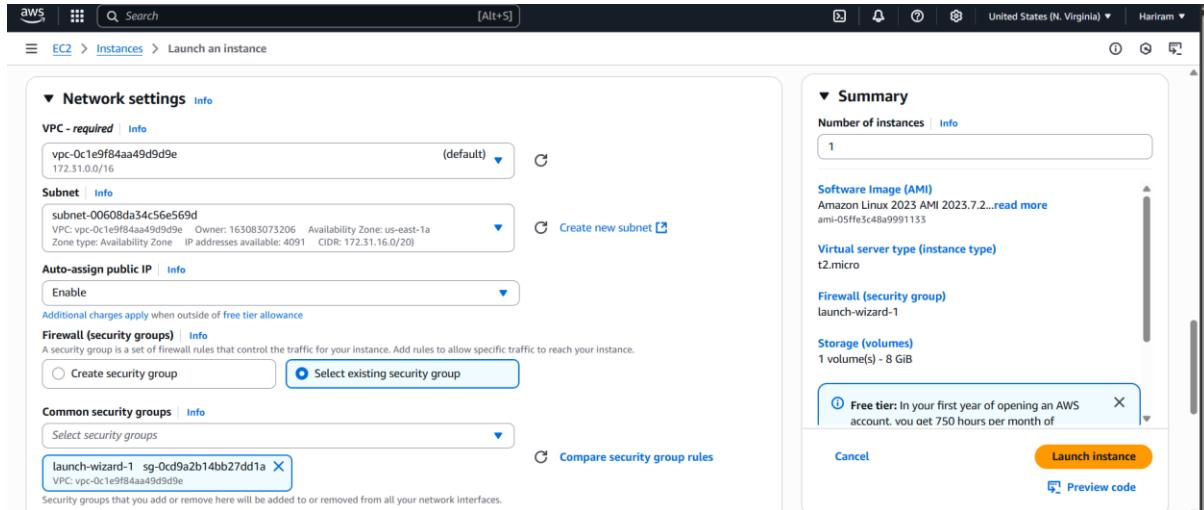
Nginx web app is deployed in the kubernetes pod and based on the load replica set will be dynamically added and deleted.

### 3) Easy-1

Using configuration management tool Ansible create multiple users and confirm on your manage host.



Master instance is created.



Existing key pair is selected and existing security group is selected.

**Name and tags**

Name: worker

**Software Image (AMI)**  
Amazon Linux 2023 AMI 2023.7.2...[read more](#)  
ami-05ffe3c48a9991133

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
New security group

**Storage (volumes)**  
1 volume(s) - 8 GiB

**Summary**

Number of instances: 1

**Launch instance**

Worker node instance is created.

**Network settings**

**VPC - required**

vpc-0c1e9f84aa49d9d9e (default)

**Subnet**

subnet-006080da34c56e569d

**Auto-assign public IP**

Enable

**Firewall (security groups)**

Select existing security group

**Common security groups**

Compare security group rules

**Summary**

Number of instances: 1

**Software Image (AMI)**  
Amazon Linux 2023 AMI 2023.7.2...[read more](#)  
ami-05ffe3c48a9991133

**Virtual server type (instance type)**  
t2.micro

**Firewall (security group)**  
launch-wizard-1

**Storage (volumes)**  
1 volume(s) - 8 GiB

**Free tier**: In your first year of opening an AWS account, you get 750 hours per month of

**Launch instance**

Existing key pair is selected and existing security group is selected.

**EC2**

**Instances**

**Instances (2)**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
master	i-0ef1120d2576eb634	Running	t2.micro	Initializing	<a href="#">View alarms +</a>	us-east-1a	ec2-54-9-
worker	i-06df4c767dbf299af	Pending	t2.micro	-	<a href="#">View alarms +</a>	us-east-1a	ec2-52-9-

**Select an instance**

Instances are created successfully.

```

root@ip-172-31-26-180:~ x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835505\OneDrive - LTIMindtree\Desktop> ssh -i "aws_Ltimindtree.pem" ec2-user@ec2-54-92-196-219.compute-1.amazonaws.com
The authenticity of host 'ec2-54-92-196-219.compute-1.amazonaws.com (54.92.196.219)' can't be established.
ED25519 key fingerprint is SHA256:Ime1yWan9T7EK3z25H6UMY2Q750hMw3QkHGKn8XnIvE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-92-196-219.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

  _#
  ~\_ #####_      Amazon Linux 2023
  ~~ \#####\
  ~~  \###|
  ~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
  ~~   V~' '-->
  ~~~   /_
  ~~_.-._/_/
  _/`_/_/
  _/m'_

[ec2-user@ip-172-31-26-180 ~]$ sudo su -
[root@ip-172-31-26-180 ~]# hostnamectl set-hostname master.example.com
[root@ip-172-31-26-180 ~]# bash
[root@master ~]# |

```

Master node is ready.

```

root@ip-172-31-24-212:~ x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835505\OneDrive - LTIMindtree\Desktop> ssh -i "aws_Ltimindtree.pem" ec2-user@ec2-52-90-231-203.compute-1.amazonaws.com
The authenticity of host 'ec2-52-90-231-203.compute-1.amazonaws.com (52.90.231.203)' can't be established.
ED25519 key fingerprint is SHA256:65F1ME7a6iAEBSX+WZ6kpBFN712pY1PpDyqwGeqTHDU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-52-90-231-203.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

  _#
  ~\_ #####_      Amazon Linux 2023
  ~~ \#####\
  ~~  \###|
  ~~   \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
  ~~   V~' '-->
  ~~~   /_
  ~~_.-._/_/
  _/`_/_/
  _/m'_

[ec2-user@ip-172-31-24-212 ~]$ sudo su -
[root@ip-172-31-24-212 ~]# hostnamectl set-hostname worker.example.com
[root@ip-172-31-24-212 ~]# bash
[root@worker ~]# |

```

Worker node is ready.

```

  _/_/_/
  _/m'/_/
[ec2-user@ip-172-31-26-180 ~]$ sudo su -
[root@ip-172-31-26-180 ~]# hostnamectl set-hostname master.example.com
[root@ip-172-31-26-180 ~]# bash
[root@master ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@master ~]# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 0a:ff:db:8b:ce:61 brd ff:ff:ff:ff:ff:ff
    altname eni-05841ed80f175a7f3
    altname device-number=0
    inet 172.31.26.180/20 metric 512 brd 172.31.31.255 scope global dynamic enX0
        valid_lft 3402sec preferred_lft 3402sec
    inet6 fe80::8ff:dbff:fe8b:ce61/64 scope link proto kernel ll
        valid_lft forever preferred_lft forever
[root@master ~]# hostname
master.example.com
[root@master ~]# |

```

Ip and hostname of master is taken.

```
[root@ip-172-31-24-212:~]# ~~~-.-. /  
[ec2-user@ip-172-31-24-212 ~]$ sudo su -  
[root@ip-172-31-24-212 ~]# hostnamectl set-hostname worker.example.com  
[root@ip-172-31-24-212 ~]# bash  
[root@worker ~]# passwd root  
Changing password for user root.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@worker ~]# ip a s  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000  
    link/ether 0a:ff:c2:10:e4:d5 brd ff:ff:ff:ff:ff:ff  
    altname eni-0b12981207e9b5601  
    altname device-number-0.0  
    inet 172.31.24.212/20 metric 512 brd 172.31.31.255 scope global dynamic enX0  
        valid_lft 3416sec preferred_lft 3416sec  
    inet6 fe80::8ff:c2ff:fe10:e4d5/64 scope link proto kernel_ll  
        valid_lft forever preferred_lft forever  
[root@worker ~]# hostname  
worker.example.com  
[root@worker ~]# |
```

Ip and hostname of worker is taken.

```
[root@ip-172-31-26-180:~]# ~~~-.-. /  
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1         localhost6 localhost6.localdomain6  
  
172.31.26.180 master.example.com  
172.31.24.212 worker.example.com|  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

```
[root@ip-172-31-26-180:~]# ~~~-.-. /  
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1         localhost6 localhost6.localdomain6  
  
172.31.26.180 master.example.com  
172.31.24.212 worker.example.com|  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

Both machine ip and hostnames are shared.

```
[root@master ~]# vim /etc/hosts
[root@master ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:EHYJPgQJJuGaCHr9GfDPoESkRktjNoQlhNryzJqq/fw root@master.example.com
The key's randomart image is:
+---[RSA 3072]---+
|+.o..o=...
|o+ o.+ o.
|+.o. o+
|o.o+ o o
|[BBo o+ S
|.=+. o *
|oo . o o
|+*. .
|B.+.o.E
+---[SHA256]---+
[root@master ~]# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQABAAAABgQDkgQ5wL78oUcIQvWZrTVg0B7zh0B0vhJtIe2J+NgAnCJgrb+/bi2SjRUss3ZKKCEPxzjo/mdTpJsD00n8zq8u1T526oVV/MDPZt1RXkjFMc9nU5Vhj3RZYlp3R+M2t6rTrFkUpHrZvLFPiGpz/cq/f/NM436nfUJtXkbh4s19jy/Wc3jqqM8fdcNAMVs5UmZK49fSGF04A7YjjBz2UDhkZwHTY/ue6SFnXHmN6IDN7RtxvaQtmTWKBWjB2c77vee41ySVQN904T1WPICW+bCZQcb/uWFmH0qBijGtRuaeVZ3p5UCo0Vbd2ibhhdfgLdj3/3ve044RGZ3VNbcsmpFnuap8U/sUP9fxZ8+y/X15WEQyDIJdRypViDg6auXbuGkwDcHNbp6xbDHHbSLhyYMT2sBbX2B879z3l1plJSAQLXxyciFaZLZzF9AxyxuvuiKx9mdo2/470nR3JutW61lUFVb810YsAJk0Pm250q2YYkEnllIyCv8= root@master.example.com
[root@master ~]# |
```

Ssh-keygen is done and pub key is copied.

```
+ x root@ip-172-31-24-212:~/ssh - + ~ no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDowCDZw93pIkZ5Zyr4FM0jcgHTQnJdrf8gR XkLMU5mZUBLAqJABCr5lsyzodN+N1iAx9Yee4zd3f+0meqX45ZxTXzDRNGT50m7fdB5Z4dw/IUJsRcSOICVPvJQwKb6Fcrm1Maa3yguwvt2pBWJcqgfV OrlyisqKOgwzTIUoHuDwCrLag8Lh+gXBZx5FRF4mpgLg67WRNP22dzE+hD5tIBe2DKQvSjU38FiiDcFRVm5x+Eka0VnqJBxVjqUspolzK/z3W25cVLqh0/N rBQrdn+0J8YNPtYzo2Np5ghhKV3ie4z0tCNY8Aa5ZWjFlptqmcIvg9Nzvdm25QDE/ aws_Ltimindtree

ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDkgQ5wL78oUcIQvWZrTvgh0B7zh0B0vbHJtIe2J+NgAnCJgrb+/bi2sJRUs3ZKKCEPxzjo/mdXTpJsD00n 8Zq8uIT5Z6oVV/MDPZTiRXKjfMC9nUSVhj3RZYlp3R+M2t6sTFkDuhpHeRzvLFPiGPz/cq/f/NUM36NFUJtXkb4hSL9jy/Wc3jqqM8fdcNAwv5zUmZK49fSGF T04A7YjjbIz2UDhkZwhTY/ue6SFNxHMN6IDN7RtxvaQtmtWKBBljB2c77vee41ySVQN904T1WPICW+bCZQcB/uWFmHoBijGtRuaeVZ3p5UCoObDz2ibih dgfLDji/3veU044RGZ23VbcsmpptFnuaP8u/sUP9fxZ8+v/X15WEQyDIJdrYpViDg6auxBuGkwDchNbpb6xbDHhbSLhyYMT2sBbx2B879z3L1lplJSAQLLxxyc ciFaLZzF9AxyxuvuiKx9mdo2/470nR3JutW61lUFVb810YsAjkOpM250q2YYkEnlliYcv8= root@master.example.com|
```

Pasted the pub key in worker node.

```
[root@ip-172-31-24-212 ~]# cd .ssh/  
[root@worker .ssh]# ll  
total 4  
-r--w----- 1 root root 563 Jul 10 08:55 authorized_keys  
[root@worker .ssh]# vim authorized_keys  
[root@worker .ssh]# |
```

Pasted successfully.

```
[root@master ~]# yum install ansible*
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Thu Jul 10 09:03:20 2025.
Dependencies resolved.
=====
Package           Architecture      Version        Repository      Size
=====
Installing:
ansible          noarch          8.3.0-1.amzn2023.0.1
ansible-core      x86_64         2.15.3-1.amzn2023.0.11
ansible-packaging noarch          1-11.amzn2023.0.1
ansible-packaging-tests noarch          1-11.amzn2023.0.1
ansible-srpm-macros noarch          1-11.amzn2023.0.1
=====
Installing dependencies:
ansible-test      x86_64         2.15.3-1.amzn2023.0.11
git-core          x86_64         2.47.1-1.amzn2023.0.3
python3-apipkg    noarch          1.5-12.amzn2023.0.2
python3-execnet   noarch          1-7.1-5.amzn2023.0.2
python3-iniconfig noarch          1.1.1-2.amzn2023.0.2
python3-packaging noarch          21.3-2.amzn2023.0.2
python3-pluggy    noarch          0.13.1-3.amzn2023.0.2
python3-py        noarch          1.10.0-2.amzn2023.0.2
python3-pyparsing noarch          2.4.7-6.amzn2023.0.2
python3-pytest    noarch          6.2.2-1.amzn2023.0.3
python3-pytest-forked noarch          1.3.0-2.amzn2023.0.2
python3-pytest-mock noarch          3.5.1-2.amzn2023.0.2
python3-pytest-xdist noarch          2.2.0-2.amzn2023.0.2
python3-toml     noarch          0.10.2-2.amzn2023.0.2
sshpss          x86_64         1.09-6.amzn2023.0.1
=====
Transaction Summary
=====
Install 20 Packages
=====
```

Ansible is installed in master node.

```
[root@master ~]# yum install ansible*
Verifying : ansible-core-2.15.3-1.amzn2023.0.11.x86_64
Verifying : ansible-packaging-1-11.amzn2023.0.1.noarch
Verifying : ansible-packaging-tests-1-11.amzn2023.0.1.noarch
Verifying : ansible-srpm-macros-1-11.amzn2023.0.1.noarch
Verifying : ansible-test-2.15.3-1.amzn2023.0.11.x86_64
Verifying : git-core-2.47.1-1.amzn2023.0.3.x86_64
Verifying : python3-apipkg-1.5-12.amzn2023.0.2.noarch
Verifying : python3-execnet-1.7.1-5.amzn2023.0.2.noarch
Verifying : python3-iniconfig-1.1.1-2.amzn2023.0.2.noarch
Verifying : python3-packaging-21.3-2.amzn2023.0.2.noarch
Verifying : python3-pluggy-0.13.1-3.amzn2023.0.2.noarch
Verifying : python3-py-1.10.0-2.amzn2023.0.2.noarch
Verifying : python3-pyparsing-2.4.7-6.amzn2023.0.2.noarch
Verifying : python3-pytest-6.2.2-1.amzn2023.0.3.noarch
Verifying : python3-pytest-forked-1.3.0-2.amzn2023.0.2.noarch
Verifying : python3-pytest-mock-3.5.1-2.amzn2023.0.2.noarch
Verifying : python3-pytest-xdist-2.2.0-2.amzn2023.0.2.noarch
Verifying : python3-toml-0.10.2-2.amzn2023.0.2.noarch
Verifying : sshpass-1.09-6.amzn2023.0.1.x86_64
=====
Installed:
ansible-8.3.0-1.amzn2023.0.1.noarch
ansible-packaging-tests-1-11.amzn2023.0.1.noarch
git-core-2.47.1-1.amzn2023.0.3.x86_64
python3-iniconfig-1.1.1-2.amzn2023.0.2.noarch
python3-py-1.10.0-2.amzn2023.0.2.noarch
python3-pytest-forked-1.3.0-2.amzn2023.0.2.noarch
python3-pytest-mock-3.5.1-2.amzn2023.0.2.noarch
python3-toml-0.10.2-2.amzn2023.0.2.noarch
sshpss-1.09-6.amzn2023.0.1.x86_64
=====
Complete!
[root@master ~]# ansible --version
ansible [core 2.15.3]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.9.23 (main, Jun 11 2025, 00:00:00) [GCC 11.5.0 20240719 (Red Hat 11.5.0-5)] (/usr/bin/python3.9)
  jinja version = 3.1.4
  libyaml = True
[root@master ~]#
```

Ansible is available in master node.

```
[root@master ~]# cd /etc/ansible
[root@master ansible]# ll
total 0
drwxr-xr-x. 2 root root 6 Mar 24 18:48 roles
[root@master ansible]# vim ansible.cfg
[root@master ansible]# ll
total 8
-rw-r--r--. 1 root root 6119 Jul 10 09:05 ansible.cfg
drwxr-xr-x. 2 root root 6 Mar 24 18:48 roles
[root@master ansible]#
```

Ansible.cfg file is added.

```

root@ip-172-31-26-180:/etc/ansible# vim user.yaml
---
- name: create multiple users
  hosts: all
  tasks:
    - name: create multiple users
      user:
        name: "{{ item }}"
        state: present
      loop:
        - hariram
        - dracula
        - ironman

```

Playbook to add multiple user is added . Users hariram,Dracula,ironman will be added.

```

[root@master ansible]# vim user.yaml
[root@master ansible]# ansible-playbook user.yaml

PLAY [create multiple users] ****
TASK [Gathering Facts] ****
The authenticity of host 'worker.example.com (172.31.24.212)' can't be established.
ED25519 key fingerprint is SHA256:65F1ME7a61AEBSX-WZ6kpBFN712pY1PpDyqwGeqTHDU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
[WARNING]: Platform linux on host worker.example.com is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [worker.example.com]

TASK [create multiple users] ****
changed: [worker.example.com] => (item=hariram)
changed: [worker.example.com] => (item=dracula)
changed: [worker.example.com] => (item=ironman)

PLAY RECAP ****
worker.example.com : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
[root@master ansible]#

```

Playbook is applied.

```

root@ip-172-31-24-212~:~# vim /etc/passwd
root@ip-172-31-24-212~:~#

```

In worker node, using vim/etc/passwd , user added can be confirmed.

```

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin/sbin/nologin
daemon:x:2:2:daemon:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/bin/shutdown
halt:x:7:0:halt:/sbin:/bin/halt
mail:x:8:12:mail:/var/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/usr/sbin/nologin
systemd-resolvd:x:999:999:systemd userspace DNS Killer:/usr/sbin/nologin
syslog:x:10:10:syslog:/var/log:/sbin/nologin
sshd:x:70:70:Privileged-separated SSH:/usr/share/empty:sshd:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
libstoragemgmt:x:997:997:daemon account for libstoragemgmt:/usr/sbin/nologin
systemd-coredump:x:996:996:systemd Core Dumper:/usr/sbin/nologin
systemd-timesync:x:995:995:systemd Time Synchronization:/usr/sbin/nologin
chrony:x:994:994:chrony system user:/var/lib/chrony:/sbin/nologin
ec2-instance-connect:x:993:993:/home/ec2-instance-connect:/sbin/nologin
stapunpriv:x:159:159:Systemtap unprivileged user:/var/lib/stapunpriv:/sbin/nologin
rpcuser:x:1000:1000:RPC User:/var/lib/nfs:/sbin/nologin
tcpdump:x:72:72::/sbin/nologin
ec2-user:x:1000:1000:EC2 Default User:/home/ec2-user:/bin/bash
hariram:x:1001:1001:/home/hariram:/bin/bash
dracula:x:1002:1002:/home/dracula:/bin/bash
ironman:x:1003:1003:/home/ironman:/bin/bash

```

Multiple users added successfully using loop concept.

## 4)Medium-1

Deploy in a Docker instance and create a Docker image, Store the Docker image in ECR, achieve this image on EKS cluster using the ECR image. Build a sample java web app using maven

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. In the 'Name and tags' section, a tag named 'jenkins' is added. The 'Application and OS Images (Amazon Machine Image)' section shows the selection of the 'Amazon Linux' AMI. The 'Summary' panel on the right indicates 1 instance will be launched, using the 'Amazon Linux 2023 AMI 2023.7.2...' image, a 't2.medium' instance type, and a new security group. A note about the free tier is visible.

Jenkins instance is created.

The screenshot shows the continuation of the 'Launch an instance' wizard. It includes sections for 'Instance type' (set to t2.medium), 'Key pair (login)' (set to 'aws\_ltiimindtree'), and 'Network settings'. The 'VPC - required' section shows a selected subnet ('subnet-00008da34c56e569d'). The 'Summary' panel on the right remains consistent with the previous step, showing 1 instance launching with the same configurations.

Existing key pair is selected . Existing security group is selected.

```
[root@ip-172-31-17-144 ~]# ssh -i "aws_ltmindtree.pem" ec2-user@ec2-54-234-228-125.compute-1.amazonaws.com
The authenticity of host 'ec2-54-234-228-125.compute-1.amazonaws.com (54.234.228.125)' can't be established.
ED25519 key fingerprint is SHA256:zRlI4NRR8+5JAF15zgOCRSqLMfwc/YP3+pFBJVJeJuW.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-234-228-125.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
`_\_ #####      Amazon Linux 2023
~~ \###\
~~ \###|
~~ #\#
~~ #\#|   https://aws.amazon.com/linux/amazon-linux-2023
~~ V, `-->
~~ /`_
~~ .-`/
~~ _/-`/
~~ _/`/
[ec2-user@ip-172-31-17-144 ~]$ sudo su -
[root@ip-172-31-17-144 ~]# hostnamectl set-hostname jenkins.example.com
[root@ip-172-31-17-144 ~]# bash
[root@jenkins ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@jenkins ~]# yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Thu Jul 10 09:23:27 2025.
No match for argument: -y
Error: No packages marked for upgrade.
[root@jenkins ~]# |
```

Jenkins is running successfully and necessary dependencies are installing.

```
[root@ip-172-31-17-144 ~]# curl https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-17-144 ~]$ sudo su -
[root@ip-172-31-17-144 ~]# hostnamectl set-hostname jenkins.example.com
[root@ip-172-31-17-144 ~]# bash
[root@jenkins ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@jenkins ~]# yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Thu Jul 10 09:23:27 2025.          137 kB/s | 17 kB     00:00
No match for argument: -y
Error: No packages marked for upgrade.
[root@jenkins ~]# wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2025-07-10 09:23:55 -- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.34.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.34.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.repo      100%[=====] 85 --.-KB/s   in 0s

2025-07-10 09:23:55 (5.83 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@jenkins ~]# rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@jenkins ~]# yum upgrade
Jenkins-stable
Dependencies resolved.
Nothing to do.
Complete!
[root@jenkins ~]#
```

Yum upgrade is done.

```
[root@jenkins ~]# yum install java-17-amazon-corretto -y
Last metadata expiration check: 0:00:48 ago on Thu Jul 10 09:24:19 2025.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
java-17-amazon-corretto x86_64        1:17.0.15+6-1.amzn2023.1      amazonlinux  213 k
Installing dependencies:
alsa-lib            x86_64        1.2.7.2-1.amzn2023.0.2      amazonlinux  504 k
cairo               x86_64        1.18.0-4.amzn2023.0.1      amazonlinux  718 k
dejavu-sans-fonts noarch        2.37-16.amzn2023.0.2      amazonlinux  1.3 M
dejavu-sans-mono-fonts noarch        2.37-16.amzn2023.0.2      amazonlinux  467 k
dejavu-serif-fonts noarch        2.37-16.amzn2023.0.2      amazonlinux  1.0 M
Fontconfig          x86_64        2.13.94-2.amzn2023.0.2      amazonlinux  273 k
Fonts-filesystem   noarch        1:2.0.5-12.amzn2023.0.2      amazonlinux  9.5 k
freetype            x86_64        2.13.2-5.amzn2023.0.1      amazonlinux  423 k
glibffi             x86_64        2.1.1-9.amzn2023.0.1      amazonlinux  49 k
google-noto-fonts-common noarch        28201206-2.amzn2023.0.2      amazonlinux  15 k
google-noto-sans-vf-fonts noarch        28201206-2.amzn2023.0.2      amazonlinux  492 k
graphite2          x86_64        1.3.14-7.amzn2023.0.2      amazonlinux  97 k
harfbuzz            x86_64        7.0.0-2.amzn2023.0.2      amazonlinux  873 k
java-17-amazon-corretto-headless x86_64        1:17.0.15+6-1.amzn2023.1      amazonlinux  91 M
javapackages-filesystem noarch        0.8.0-7.amzn2023.0.6      amazonlinux  32 k
langpacks-core-font-en noarch        3.0-21.amzn2023.0.4      amazonlinux  10 k
libICE              x86_64        1.1.1-3.amzn2023.0.1      amazonlinux  76 k
libSM               x86_64        1.2.4-3.amzn2023.0.1      amazonlinux  45 k
libX11              x86_64        1.8.10-2.amzn2023.0.1      amazonlinux  659 k
libX11-common       noarch        1.8.10-2.amzn2023.0.1      amazonlinux  147 k
libXau              x86_64        1.0.11-6.amzn2023.0.1      amazonlinux  33 k
libXext              x86_64        1.3.6-1.amzn2023.0.1      amazonlinux  42 k
libXi               x86_64        1.8.2-1.amzn2023.0.1      amazonlinux  42 k
libXinerama         x86_64        1.1.5-6.amzn2023.0.1      amazonlinux  16 k
libXrandr            x86_64        5.4-3.amzn2023.0.1      amazonlinux  29 k
libXrender            x86_64        0.9.11-6.amzn2023.0.1      amazonlinux  29 k
libXt               x86_64        1.3.0-3.amzn2023.0.1      amazonlinux  183 k
libXtst              x86_64        1.2.5-1.amzn2023.0.1      amazonlinux  22 k
libbrotli            x86_64        0.9.9-4.amzn2023.0.2      amazonlinux  315 k
libjpeg-turbo        x86_64        2.1.4-2.amzn2023.0.5      amazonlinux  190 k
libpng              x86_64        2.1.6.37-10.amzn2023.0.6     amazonlinux  128 k
libxml2             x86_64        1.17.0-1.amzn2023.0.1      amazonlinux  235 k
```

Java is installed.

```
[root@jenkins ~]# yum install jenkins -y
Last metadata expiration check: 0:01:15 ago on Thu Jul 10 09:24:19 2025.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
jenkins          noarch        2.504.3-1.1      jenkins      90 M
Transaction Summary
Install 1 Package
Total download size: 90 M
Installed size: 90 M
Downloading Packages:
jenkins-2.504.3-1.1.noarch.rpm                                         9.4 MB/s | 90 MB  00:09
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Running scriptlet: jenkins-2.504.3-1.1.noarch
Installing  : jenkins-2.504.3-1.1.noarch 1/1
Running scriptlet: jenkins-2.504.3-1.1.noarch
Verifying   : jenkins-2.504.3-1.1.noarch 1/1
Installed   : jenkins-2.504.3-1.1.noarch
Complete!
[root@jenkins ~] |
```

Jenkins is installed.

```
[root@jenkins ~]# systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@jenkins ~]# systemctl start jenkins
[root@jenkins ~]# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
    Active: active (running) since Thu 2025-07-10 09:26:27 UTC; 2s ago
      Main PID: 26271 (java)
        Tasks: 46 (limit: 4656)
       Memory: 769.7M
          CPU: 18.549s
         CGroup: /system.slice/jenkins.service
             └─26271 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Jul 10 09:26:23 jenkins.example.com jenkins[26271]: e502753074cf4495ae0f7309e0400bb
Jul 10 09:26:23 jenkins.example.com jenkins[26271]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Jul 10 09:26:23 jenkins.example.com jenkins[26271]: ****
Jul 10 09:26:23 jenkins.example.com jenkins[26271]: ****
Jul 10 09:26:23 jenkins.example.com jenkins[26271]: ****
Jul 10 09:26:23 jenkins.example.com jenkins[26271]: 2025-07-10 09:26:27.409+0000 [id=30]      INFO  jenkins.InitReactorRunner$1#onAttained: Complete>
Jul 10 09:26:27 jenkins.example.com jenkins[26271]: 2025-07-10 09:26:27.442+0000 [id=23]      INFO  hudson.lifecycle.Lifecycle#onReady: Jenkins is >
Jul 10 09:26:27 jenkins.example.com systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Jul 10 09:26:27 jenkins.example.com jenkins[26271]: 2025-07-10 09:26:27.729+0000 [id=49]      INFO  h.m.DownloadService$Downloadable#load: Obtained>
Jul 10 09:26:27 jenkins.example.com jenkins[26271]: 2025-07-10 09:26:27.731+0000 [id=49]      INFO  hudson.util.Retrier#start: Performed the action>
lines 1-20/20 (END)
```

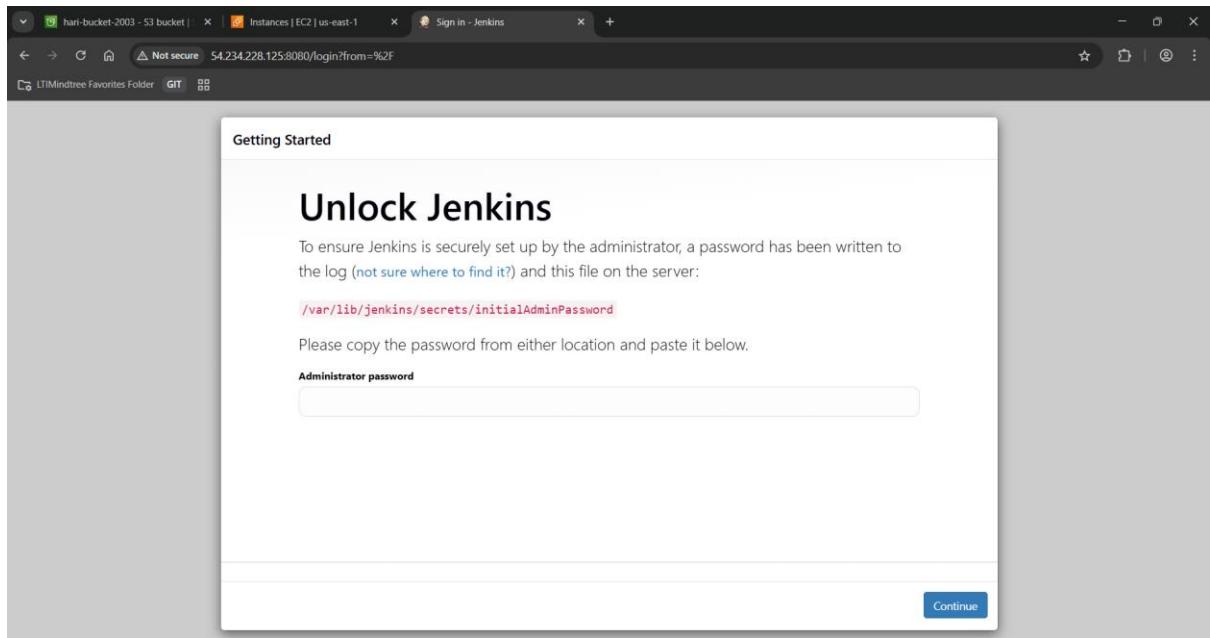
Jenkins is started and enabled. Jenkins status is active.

```
[root@jenkins ~]# yum install git -y
Last metadata expiration check: 0:02:32 ago on Thu Jul 10 09:24:19 2025.
Dependencies resolved.
=====
Package           Architecture      Version       Repository   Size
=====
Installing:
git              x86_64          2.47.1-1.amzn2023.0.3      amazonlinux  52 k
Installing dependencies:
git-core          x86_64          2.47.1-1.amzn2023.0.3      amazonlinux  4.5 M
git-core-doc      noarch          2.47.1-1.amzn2023.0.3      amazonlinux  2.8 M
perl-Error        noarch          1:0.17029-5.amzn2023.0.2    amazonlinux  41 k
perl-File-Find   noarch          1.37-477.amzn2023.0.7     amazonlinux  25 k
perl-Git          noarch          2.47.1-1.amzn2023.0.3      amazonlinux  40 k
perl-TermReadKey x86_64          2.38-9.amzn2023.0.2       amazonlinux  36 k
perl-lib          x86_64          0.65-477.amzn2023.0.7     amazonlinux  15 k
Transaction Summary
=====
Install 8 Packages
Total download size: 7.5 M
Installed size: 37 M
Downloading Packages:
(1/8): git-2.47.1-1.amzn2023.0.3.x86_64.rpm 1.5 MB/s | 52 kB  00:00
(2/8): git-core-doc-2.47.1-1.amzn2023.0.3.noarch.rpm 41 MB/s | 2.8 MB  00:00
(3/8): perl-Error-0.17029-5.amzn2023.0.2.noarch.rpm 1.1 MB/s | 41 kB  00:00
(4/8): perl-File-Find-1.37-477.amzn2023.0.7.noarch.rpm 1.2 MB/s | 25 kB  00:00
(5/8): git-core-2.47.1-1.amzn2023.0.3.x86_64.rpm 34 MB/s | 4.5 MB  00:00
(6/8): perl-Git-2.47.1-1.amzn2023.0.3.noarch.rpm 618 kB/s | 40 kB  00:00
(7/8): perl-TermReadKey-2.38-9.amzn2023.0.2.x86_64.rpm 786 kB/s | 36 kB  00:00
(8/8): perl-lib-0.65-477.amzn2023.0.7.x86_64.rpm 706 kB/s | 15 kB  00:00
=====
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : git-core-2.47.1-1.amzn2023.0.3.x86_64
=====
1/1
Installing : git-core-2.47.1-1.amzn2023.0.3.x86_64
1/8
```

Git is installed.

```
[root@jenkins ~]# yum install maven
Last metadata expiration check: 0:02:58 ago on Thu Jul 10 09:24:19 2025.
Dependencies resolved.
=====
Package           Architecture      Version       Repository   Size
=====
Installing:
maven             noarch          1:3.8.4-3.amzn2023.0.5      amazonlinux  18 k
Installing dependencies:
apache-commons-cli noarch          1.5.0-3.amzn2023.0.3      amazonlinux  76 k
apache-commons-codec noarch          1.15-6.amzn2023.0.3      amazonlinux  303 k
apache-commons-io   noarch          1:2.8.0-7.amzn2023.0.5    amazonlinux  283 k
apache-commons-lang3 noarch          3.12.0-7.amzn2023.0.3    amazonlinux  559 k
atInject          noarch          1.0.5-3.amzn2023.0.3      amazonlinux  23 k
cdi-api           noarch          2.0.2-6.amzn2023.0.3      amazonlinux  54 k
google-guice       noarch          4.2.3-8.amzn2023.0.6     amazonlinux  473 k
guava              noarch          31.0.1-3.amzn2023.0.6    amazonlinux  2.4 M
httpcomponents-client noarch          4.5.13-4.amzn2023.0.4    amazonlinux  657 k
httpcomponents-core noarch          4.4.13-6.amzn2023.0.3    amazonlinux  632 k
jakarta-annotations noarch          1.3.5-13.amzn2023.0.3    amazonlinux  46 k
jansi              x86_64          2.4.0-3.amzn2023.0.3      amazonlinux  113 k
java-17-amazon-corretto-devel x86_64          1:17.0.15+6-1.amzn2023.1 amazonlinux  142 k
jcl-over-slf4j       noarch          1.7.32-3.amzn2023.0.4    amazonlinux  25 k
jsoup              noarch          1.16.1-4.amzn2023.0.2    amazonlinux  433 k
jsr-385             noarch          3.0.2-5.amzn2023.0.4    amazonlinux  32 k
maven-amazon-corretto17 noarch          1:3.8.4-3.amzn2023.0.5    amazonlinux  9.4 k
maven-lib            noarch          1:3.8.4-3.amzn2023.0.5    amazonlinux  1.5 M
maven-resolver       noarch          1:1.7.3-3.amzn2023.0.4    amazonlinux  557 k
maven-shared-utils   noarch          3.3.4-4.amzn2023.0.3    amazonlinux  152 k
maven-wagon          noarch          3.4.2-6.amzn2023.0.4    amazonlinux  113 k
plexus-cipher        noarch          1.8-3.amzn2023.0.3      amazonlinux  27 k
plexus-classworlds  noarch          2.6.0-10.amzn2023.0.4    amazonlinux  61 k
plexus-containers-component-annotations noarch          2.1.0-9.amzn2023.0.4    amazonlinux  19 k
plexus-interpolation noarch          1.26-10.amzn2023.0.4    amazonlinux  80 k
plexus-sec-dispatcher noarch          2.0-3.amzn2023.0.3      amazonlinux  34 k
plexus-utils          noarch          3.3.0-9.amzn2023.0.4    amazonlinux  254 k
publicsuffix-list    noarch          20240212-61.amzn2023    amazonlinux  89 k
sisu                noarch          1:0.3.4-9.amzn2023.0.4    amazonlinux  510 k
slf4j               noarch          1.7.32-3.amzn2023.0.4    amazonlinux  70 k
=====
Transaction Summary
```

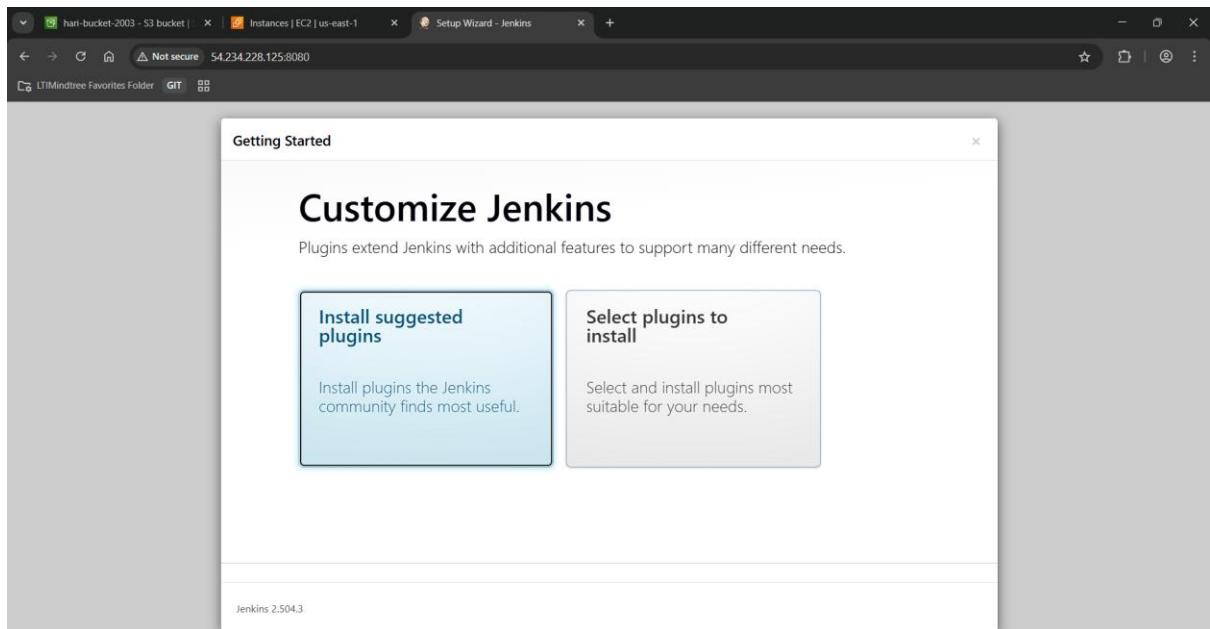
Maven is installed.



Jenkins dashboard is accessed using instances public ip:8080.

```
[root@jenkins ~]# mvn -v
Apache Maven 3.8.4 (Red Hat 3.8.4-3.amzn2023.0.5)
Maven home: /usr/share/maven
Java version: 17.0.15, vendor: Amazon.com Inc., runtime: /usr/lib/jvm/java-17-amazon-corretto.x86_64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.1.141-155.222.amzn2023.x86_64", arch: "amd64", family: "unix"
[root@jenkins ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
e502753074cf4495ae0f7309e0400bb
[root@jenkins ~]#
```

Password is taken from Jenkins terminal.



Suggested plugins are installing.

The screenshot shows the Jenkins Setup Wizard - Create First Admin User page. It has a form with fields for Username (admin), Password (redacted), Confirm password (redacted), Full name (Hariram), and E-mail address (hariramdevops@outlook.com). At the bottom, there are buttons for Jenkins 2.504.3, Skip and continue as admin, and Save and Continue.

Getting Started

## Create First Admin User

Username  
admin

Password  
.....

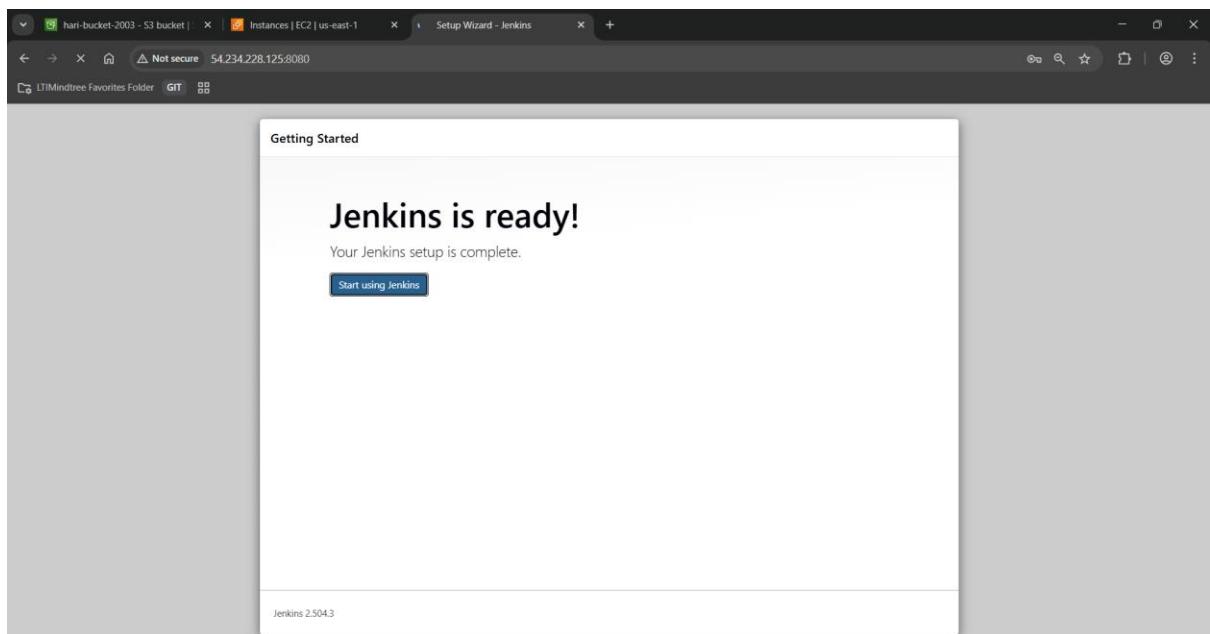
Confirm password  
.....

Full name  
Hariram

E-mail address  
hariramdevops@outlook.com

Jenkins 2.504.3      [Skip and continue as admin](#)      [Save and Continue](#)

Admin user is created.



Jenkins is ready to used.

The screenshot shows a GitHub repository page for 'java-maven-project'. The repository has 1 branch and 0 tags. It contains several files: server, webapp, Dockerfile, README.md, pom.xml, regapp-deploy.yml, and regapp-service.yml. The last commit was made by 'Haim24' 15 hours ago, updating index.jsp. The repository has 9 commits in total. On the right side, there is an 'About' section with a 'No description, website, or topics provided.' message. It also shows activity, releases (none), packages (none), and languages (Java 94.1%, Dockerfile 5.9%).

Existing repository is use where maven project is available.

The screenshot shows the Jenkins Security settings page. Under the 'API Token' section, there is a current token labeled 'test' with the value '1126d322bceeb68e34ee3951c30a00df4d'. A warning message says '⚠️ Copy this token now, because it cannot be recovered in the future.' Below this, there is a button to 'Add new Token'.

Token is generate in Jenkins to connect it with github.

The screenshot shows the GitHub Settings page for the 'java-maven-project' repository. In the left sidebar, under 'Webhooks', a new webhook is being added. The 'Payload URL' is set to 'http://54.234.228.125:8080/github-webhook/'. The 'Content type' is set to 'application/json'. The 'Secret' field contains the generated API token: '1126d322bceeb68e34ee3951c30a00df4d'. Under 'SSL verification', the 'Enable SSL verification' option is selected. In the 'Which events would you like to trigger this webhook?' section, the 'Just the push event.' option is selected.

Token is pasted in git webhook and Jenkins url is pasted.

The screenshot shows the GitHub repository settings page for 'java-maven-project'. Under the 'Webhooks' section, a new webhook has been added successfully. The URL is listed as 'http://54.234.228.125:8080/github... (push)'. A message indicates 'Last delivery was successful.'

Webhooks is added successfully.

The screenshot shows the Jenkins 'Manage Jenkins' > 'Plugins' page. The 'Available plugins' tab is selected. Two plugins are listed: 'Maven Integration' (version 3.26) and 'GitHub Integration' (version 0.7.2). Both are marked as 'Released' and have their last update dates listed.

Necessary plugins are added.

The screenshot shows the Jenkins 'Manage Jenkins' > 'Plugins' page. The 'Installed plugins' tab is selected. A large list of Jenkins components and their status is displayed, all marked as 'Success'. Components include 'SSH Build Agents', 'Matrix Authorization Strategy', 'PAM Authentication', 'LDAP', 'jsoup API', 'Email Extension', 'Mailer', 'Theme Manager', 'Dark Theme', 'Loading plugin extensions', 'Javadoc', 'Dev Tools Symbols API', 'JSch dependency', 'Maven Integration', 'GitHub Integration', 'Deploy to container', and 'Loading plugin extensions'.

Restarted after installation.

The screenshot shows the Jenkins 'Tools' configuration page under 'Manage Jenkins'. It is set to 'Use default maven global settings'. The 'JDK installations' section contains a single entry for 'java-17' with the path '/usr/lib/jvm/java-17-amazon-corretto.x86\_64'. There is an unchecked checkbox for 'Install automatically'. Below this is a 'Git installations' section which is currently empty. At the bottom are 'Save' and 'Apply' buttons.

Java path is added from the Jenkins terminal.

The screenshot shows the Jenkins 'Tools' configuration page under 'Manage Jenkins'. It is set to 'Use default maven global settings'. The 'Maven installations' section contains a single entry for 'maven' with the path '/usr/share/maven'. There is an unchecked checkbox for 'Install automatically'. Below this is a 'Add Maven' button. At the bottom are 'Save' and 'Apply' buttons.

Maven path is added from the Jenkins terminal.

The screenshot shows the Jenkins 'New Item' creation page. The item name is 'maven' and the item type is selected as 'Maven project'. Other options shown include 'Freetype project', 'Pipeline', 'Multi-configuration project', and 'Folder'. At the bottom is an 'OK' button.

New maven project is created.

Dashboard > maven > Configuration

**Configure**

- General
- Source Code Management**
- Triggers
- Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

**Git**

Repository URL: https://github.com/Hariram24/java-maven-project.git

Credentials: - none -

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any'): \*/main

Save Apply

Git http link is added and /master is changed to /main.

Dashboard > maven > #1 > Console Output

**Console Output**

Started by user Hariram  
Running as SYSTEM  
Building in workspace /var/lib/jenkins/workspace/maven  
The recommended git tool is: NONE  
No credentials specified  
Cloning the remote Git repository  
Cloning repository https://github.com/Hariram24/java-maven-project.git  
> git init /var/lib/jenkins/workspace/maven # timeout=10  
Fetching upstream changes from https://github.com/Hariram24/java-maven-project.git  
> git --version # timeout=10  
> git fetch --tags --force --progress -- https://github.com/Hariram24/java-maven-project.git +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
> git config remote.origin.url https://github.com/Hariram24/java-maven-project.git # timeout=10  
> git config --add remote.origin.fetch refs/heads/\*:refs/remotes/origin/\* # timeout=10  
Avoid second fetch  
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10  
Checking out Revision 4a7e106e0ff4beac3efaf8cd17609178a5e55027 (refs/remotes/origin/main)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f 4a7e106e0ff4beac3efaf8cd17609178a5e55027 # timeout=10  
Commit message: "Update index.jsp"  
First time build. Skipping changelog.  
Parsing POMs  
Discovered a new module com.example.maven-project:maven-project Maven Project  
Discovered a new module com.example.maven-project:server Server

Build is successful and war is created in the Jenkins.

Search [Alt+S]

EC2 > Instances > Launch an instance

**Launch an instance**

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags**

Name: docker

**Application and OS Images (Amazon Machine Image)**

Search our full catalog including 1000s of application and OS images

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, Debian

Quick Start AMIs: ubuntu, Microsoft, Red Hat, SUSE, Debian

**Summary**

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.7.2...read more  
ami-05ffe3c48a9991133

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of free compute usage.

Cancel Launch instance Preview code

Docker instance is created.

**Network settings**

VPC - required: Info  
vpc-0c1e9f84aa49d9d9e (default)

Subnet: Info  
subnet-00608da34c56e569d VPC: vpc-0c1e9f84aa49d9d9e Owner: 163083073206 Availability Zone: us-east-1a Zone type: Availability Zone IP addresses available: 4087 CIDR: 172.31.16.0/20

Auto-assign public IP: Info  
Enable Additional charges apply when outside of free tier allowance

Firewall (security groups): Info  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.  
Create security group Select existing security group

Common security groups: Info  
Select security groups  
launch-wizard-1 sg-0cd9a2b14bb27dd1a X VPC: vpc-0c1e9f84aa49d9d9e

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Advanced network configuration

**Summary**

Number of instances: Info  
1

Software Image (AMI)  
Amazon Linux 2023 AMI 2023.7.2...read more ami-05ffe3c48a9991133

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
launch-wizard-1

Storage (volumes)  
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of

Cancel Launch instance Preview code

Existing key pair is selected and existing security group is selected.

**Name and tags**

Name: eks Add additional tags

**Application and OS Images (Amazon Machine Image)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

Browse more AMIs Including AMIs from AWS, Marketplace and the Community

**Summary**

Number of instances: Info  
1

Software Image (AMI)  
Amazon Linux 2023 AMI 2023.7.2...read more ami-05ffe3c48a9991133

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of

Cancel Launch instance Preview code

Eks instance is created.

**Network settings**

VPC - required: Info  
vpc-0c1e9f84aa49d9d9e (default)

Subnet: Info  
subnet-00608da34c56e569d VPC: vpc-0c1e9f84aa49d9d9e Owner: 163083073206 Availability Zone: us-east-1a Zone type: Availability Zone IP addresses available: 4087 CIDR: 172.31.16.0/20

Auto-assign public IP: Info  
Enable Additional charges apply when outside of free tier allowance

Firewall (security groups): Info  
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.  
Create security group Select existing security group

Common security groups: Info  
Select security groups  
launch-wizard-1 sg-0cd9a2b14bb27dd1a X VPC: vpc-0c1e9f84aa49d9d9e

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Advanced network configuration

**Summary**

Number of instances: Info  
1

Software Image (AMI)  
Amazon Linux 2023 AMI 2023.7.2...read more ami-05ffe3c48a9991133

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
launch-wizard-1

Storage (volumes)  
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of

Cancel Launch instance Preview code

Existing key pair is selected and existing security group is selected.

**AWS Info**  
Allows EC2 Instances to call AWS services on your behalf.

**Summary**

Creation date	July 09, 2025, 23:00 (UTC+05:30)
Last activity	15 hours ago
ARN	arn:aws:iam::163083073206:role/aws
Maximum session duration	1 hour

**Permissions** | Trust relationships | Tags | Last Accessed | Revoke sessions

**Permissions policies (5) Info**  
You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AdministratorAccess	AWS managed - job function	2
AmazonEC2ContainerRegistryFullAccess	AWS managed	2
AmazonEC2FullAccess	AWS managed	2
AmazonEKSServicePolicy	AWS managed	2
IAMFullAccess	AWS managed	2

Role is created with necessary permissions.

**Modify IAM role Info**  
Attach an IAM role to your instance.

**Instance ID**  
i-05bf5dd9964a45aec (docker)

**IAM role**  
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

**Create new IAM role**

**Update IAM role**

Role Added to docker instance.

**Successfully attached aws to Instance i-05bf5dd9964a45aec**

**Instances (1/5) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
master	i-0ef1120d5276eb534	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-234-228-125.co...
worker	i-06df4c767dbf299af	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-234-228-125.co...
docker	i-05bf5dd9964a45aec	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-54-234-228-125.co...
jenkins	i-0d136b4e6d104173f	Running	t2.medium	2/2 checks passed	View alarms +	us-east-1a	ec2-54-234-228-125.co...
eks	i-0012437049b743394	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-98-81-161-144.co...

**Actions** | **Launch instances**

**eks** i-0012437049b743394 (eks)

Role added to eks instance.

```
[root@ip-172-31-28-224 ~]# root@ip-172-31-28-224 ~]# Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\l0835505\OneDrive - LTMindtree\Desktop> ssh -i "aws_Ltmindtree.pem" ec2-user@ec2-13-221-126-37.compute-1.amazonaws.com
The authenticity of host 'ec2-13-221-126-37.compute-1.amazonaws.com (13.221.126.37)' can't be established.
ED25519 key fingerprint is SHA256:zGqXIAdnCI1lb71WfHWPFPQQ+YU08agEoNA53MFQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-221-126-37.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

          _#
         /###_      Amazon Linux 2023
        /##\ \
       /##| \
      /#_--> https://aws.amazon.com/linux/amazon-linux-2023
     /#_\
    /_-->
   /_\
  /_/
 /_/
/_/`[ec2-user@ip-172-31-28-224 ~]$ sudo su
[root@ip-172-31-28-224 ec2-user]# sudo su -
Last login: Thu Jul 10 09:42:32 UTC 2025 on pts/1
[root@ip-172-31-28-224 ~]# hostnamectl set-hostname docker.example.com
[root@ip-172-31-28-224 ~]# bash
[root@docker ~]# passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@docker ~]#
```

Docker instance is accessible,

Docker is installed.

```
[root@docker ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:INC20KWhrPWEvx41/kQjJqHZxotT/y52xrkyOJSR0 root@docker.example.com
The key's randomart image is:
+---[RSA 3072]---+
|          .   |
|          .+  |
|         + o *oo |
|        o E 0.o . |
|       . / +S. o |
|      . @. . + |
|     = O++ o |
|    o =o* . |
|     +o+ . |
+---[SHA256]---+
[root@docker ~]# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQABAAAABQdQRVcXpZC8k703Ghy4z9MR4xZtvQpx52SpbV7M8XR0ua7d5sHnmWLyiiaU6ncml4ICWHT6KVgWORFrevG/I8IVYP8W8w60JfqzacxNvsWrmH9sVgcA69EfSFUdYtDChNqoZ0wAnHqLg6Gw2/13vSvryLq5IX0gq1K6b019IXkjXDTwZ1pMmRRAw56K91s1zcBAMGQ9D8rNrVopKudvFM6pk3E4fI4q0w6e13eqiVyyqHnrOlLnLzn32oqNt5w0Dk2x+r9803w/DR8LzA2x9GDL9547RGNzIk8yPeCpvuQvVMwPHOueZh96AH1972xboQ6k8pNjhqAvkhpmY-A6FLRU0+iuEwuV14RMwRFEYLH+jb5/CW1DHYL8uw0R8rjZI0Pym0maI2XcfxR2ziy4Faj5qB8RWf/3D2fx1w5AwA29DpjxjsDNA0fVF6fasAuIwyODXBjxUshfk5DlqnTXFwdCmho5tZJsGnCrzzwD0xCSdt7Qi8= root@docker.example.com
[root@docker ~]#
```

Ssh-keygen is done in docker.

```

root@ip-172-31-17-144:~/ssh ~ + 
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDowCDzW93pi1kZ5Yr4FMsojcgHTQnJdrf8RXkLMU5mZUBLaqJABCsR5LsyzdN+N1iaX9ye4zd3f+0meqx45zxtDRNKGt50m7fdb85Z4d w/IUjsrcsOICPvJQwKqB6Fcrn1Ma3yguwvt2pBWJcqgv0OrlyisqKOgwzIluoHuDwCrlag8Lh+gXLbz5FRF4mpg1g67wRNP22dzE+h5TiBe2DKQvSju38FiiocFRVm5x+Eka0VnqJbxVjqUspo1zKz 3W25cVLqh0/NzBQrdn+0JBYPcYZo2Np5ghhKV3ie4zotcNYAa5ZWJfptmcIvg9Nzvdm25QDE/ aws_Ltimindtree

ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQBgQDRVccXpZC8k703Ghy4z9MR4xZtvQpx52SpH7M8XR0ua7dssNmWLYii6AU6ncml4ICW6KvgWORFrevG/18IVYP8W8w60JfqzaczNvsWrmH9sVgcA69E SFUEDrydNgoz0wAnhMq16Gw2/13v5Vvrylq5IXBqgiok6be19IXkJXTDW1pMmRRAw56KYSlzcZBAMGQ8DNrvtVopkudvMopk3Ee4FI4q#we13eqivygVhnxOlmlzn3ZoqNt5w0Dk2x+n9803W/dr81 tc8:9GDLF9547RGmZ1p8yeCpvuiPVqMwPH0ueZH96kAH972xb0Qc6k8pGjihAvkhpm#+A0FLR0U+iUewu14RMWRFELH+jb5/CW1DHYLuw08RjzIBPym0ma12XcfXR2siy4Fa5j5qB8RFw/3D2fxiw/S AwA2fJ0pdjxjsDNak0FVF6FasAuIwy0DXBjxYUshFk5DLqnTFWdCmho5tZJsgnCrzzw2D0xCVdStT7Qix8= root@docker.example.com
~
~
~
~
~
~
~
```

Docker pub key is added to the Jenkins.

```

root@jenkins ~]# cd .ssh/
[root@jenkins ~]# vim authorized_keys
[root@jenkins ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:GuFdS7MnQjUsM+A0a3qXdbMEVdaKoM5K0cqZD05ktU root@jenkins.example.com
The key's randomart image is:
+---[RSA 3072]----+
|          o   |
|         . o =  |
|        . o + +  |
|       . o= .   |
|      +|=E .. S  |
|     +BBo **.   |
|    +oo..o+*B   |
|   + o. o *     |
|  . o. o       |
+---[SHA256]----+
[root@jenkins ~]# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQBgQDZHTskf8naZ0x1cmxFg3MHxC/UuoG+4rP2G3/ila5fpj2VjVUEtXX0eTiRDsylNoKUMaydktKOYHh18jJ9Y/T+mQ62jPEj3TUNiOut+h3ngqHGeIpvt3T dvt5znzvYB/yFqJV5tx6Ulwmcn8jUwYTdjFuubBTG9yJfcy0Ef04f7MMOK6PFJe+bvyXm3YRvF944ZQ9uE+krsqvMdoMxXGKwTOqjlg5oAD9vlfD5+E1R/b3DugaytIMSePJF5feZPYwpf1BDN iVE34iHd0o8w1sRja8RJ2fZPTxcvg7GCoM1rs/0d8jEPY0T5AWXYlx1lxNjLm0073D8yUsz4HR2K+cfojF8N9GMYNoZt0HoeaM1R63/RDZhFkMbIipD8RTinUd8XiXSG1KXVrRBy5nVcdssWQ3f8 t6q1k0aceFjNvbhp0/6/R2vuaYYq6hSMFlx1Rw51jh20g8tlAvgLPlg06wNRfjntHilvJZpzVjFRaNu3ow8= root@jenkins.example.com
[root@jenkins ~]#

```

ssh-keygen is done in Jenkins.

```

root@ip-172-31-28-224:~/ssh ~ + 
no-port-forwarding,no-agent-forwarding,no-X11-forwarding,command="echo 'Please login as the user \"ec2-user\" rather than the user \"root\".';echo;sleep 10;exit 142" ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQBgQDZHTskf8naZ0x1cmxFg3MHxC/UuoG+4rP2G3/ila5fpj2VjVUEtXX0eTiRDsylNoKUMaydktKOYHh18jJ9Y/T+mQ62jPEj3TUNiOut+h3ngqHGeIpvt3T dvt5znzvYB/yFqJV5tx6Ulwmcn8jUwYTdjFuubBTG9yJfcy0Ef04f7MMOK6PFJe+bvyXm3YRvF944ZQ9uE+krsqvMdoMxXGKwTOqjlg5oAD9vlfD5+E1R/b3DugaytIMSePJF5feZPYwpf1BDN iVE34iHd0o8w1sRja8RJ2fZPTxcvg7GCoM1rs/0d8jEPY0T5AWXYlx1lxNjLm0073D8yUsz4HR2K+cfojF8N9GMYNoZt0HoeaM1R63/RDZhFkMbIipD8RTinUd8XiXSG1KXVrRBy5nVcdssWQ3f8 t6q1k0aceFjNvbhp0/6/R2vuaYYq6hSMFlx1Rw51jh20g8tlAvgLPlg06wNRfjntHilvJZpzVjFRaNu3ow8= root@jenkins.example.com
~
~
~
~
~
~
```

Jenkins pub key is added to the docker.

```

root@ip-172-31-28-224~: ~ + ~
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_keys2
# but this is overridden so installations will only check .ssh/authorized_keys
AuthorizedKeysFile      .ssh/authorized_keys

#AuthorizedPrincipalsFile none

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# Explicitly disable PasswordAuthentication. By presetting it, we
# avoid the cloud-init set_passwords module modifying sshd_config and
# restarting sshd in the default instance launch configuration.
PasswordAuthentication yes
PermitEmptyPasswords no

# Change to no to disable s/key passwords
#KbdInteractiveAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no
#KerberosUseKuserok yes

# GSSAPI options
#GSSAPIAuthentication no
-- INSERT --

```

65,27 41%

Sshd config is done in docker with password auth and permit login enabled.

```

root@ip-172-31-28-224~: ~ + ~
[root@docker ~]# vim /etc/ssh/sshd_config
[root@docker ~]# systemctl restart sshd
[root@docker ~]# systemctl enable sshd
[root@docker ~]# |

```

Sshd is restarted and enabled.

```

root@jenkins ~]# rsync -avh /var/lib/jenkins/workspace/maven* root@172.31.28.224:/opt
The authenticity of host '172.31.28.224 (172.31.28.224)' can't be established.
ED25519 key fingerprint is SHA256:z6gXIAoN+C11b71WcfHwPPFQQ+YUUBagEoNAd53Mfq.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.31.28.224' (ED25519) to the list of known hosts.
sending incremental file list
maven/
maven/Dockerfile
maven/README.md
maven/pom.xml
maven/regapp-deploy.yml
maven/regapp-service.yml
maven/.git/
maven/.git/FETCH_HEAD
maven/.git/HEAD
maven/.git/config
maven/.git/description
maven/.git/index
maven/.git/branches/
maven/.git/hooks/
maven/.git/hooks/applypatch-msg.sample
maven/.git/hooks/commit-msg.sample
maven/.git/hooks/fsmonitor-watchman.sample
maven/.git/hooks/post-update.sample
maven/.git/hooks/pre-applypatch.sample
maven/.git/hooks/pre-commit.sample
maven/.git/hooks/pre-merge-commit.sample
maven/.git/hooks/pre-push.sample
maven/.git/hooks/pre-rebase.sample
maven/.git/hooks/pre-receive.sample
maven/.git/hooks/prepare-commit-msg.sample
maven/.git/hooks/push-to-checkout.sample
maven/.git/hooks/sendemail-validate.sample
maven/.git/hooks/update.sample
maven/.git/info/
maven/.git/info/exclude
maven/.git/logs/
maven/.git/logs/HEAD
maven/.git/logs/refs/
maven/.git/logs/refs/remotes/

```

Using rsync cmd build maven project with war file is copied from Jenkins to the docker /opt folder.

```
[root@docker ~]# cd /opt
[root@docker opt]# cd maven/
[root@docker maven]# ll
total 24
-rw-r--r--. 1 992 docker 143 Jul 10 09:36 Dockerfile
-rw-r--r--. 1 992 docker 29 Jul 10 09:36 README.md
-rw-r--r--. 1 992 docker 6333 Jul 10 09:36 pom.xml
-rw-r--r--. 1 992 docker 488 Jul 10 09:36 regapp-deploy.yml
-rw-r--r--. 1 992 docker 196 Jul 10 09:36 regapp-service.yml
drwxr-xr-x. 4 992 docker 46 Jul 10 09:36 server
drwxr-xr-x. 4 992 docker 46 Jul 10 09:36 webapp
[root@docker maven]# |
```

Files successfully copied to /opt of docker.

**General settings**

**Repository name**  
Enter a concise name. Repositories support namespaces, which you can use to group similar repositories.  
163083073206.dkr.ecr.us-east-1.amazonaws.com/ **my-ecr-hari**  
11 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, and special characters \_-/-.

**Image tag mutability**  
Choose the tag mutability setting.  
 **Mutable**  
Image tags can be overwritten.  
 **Immutable**  
Image tags can't be overwritten.

**Encryption settings** Info  
⚠ The encryption settings for a repository can't be changed once the repository is created.

**Encryption configuration**  
By default, repositories use the industry standard Advanced Encryption Standard (AES) encryption. You can optionally choose to use a key stored in the AWS Key Management Service (KMS) to encrypt the images in your repository.  
 **AES-256**

ECR is created.

Repository name	URI	Created at	Tag immutability	Encryption type
my-ecr-hari	163083073206.dkr.ecr.us-east-1.amazonaws.com/my-ecr-hari	July 10, 2025, 15:21:08 (UTC+05:00)	Mutable	AES-256

Ecr created successfully.

**Push commands for my-ecr-hari**

**macOS / Linux** **Windows**

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see [Getting Started with Amazon ECR](#).

Use the following steps to authenticate your Docker client to your registry. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:  
`aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 163083073206.dkr.ecr.us-east-1.amazonaws.com`  
Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.
2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:  
`docker build -t my-ecr-hari .`
3. After the build completes, tag your image so you can push the image to this repository:  
`docker tag my-ecr-hari:latest 163083073206.dkr.ecr.us-east-1.amazonaws.com/my-ecr-hari:latest`
4. Run the following command to push this image to your newly created AWS repository:  
`docker push 163083073206.dkr.ecr.us-east-1.amazonaws.com/my-ecr-hari:latest`

Push commands are taken from ecr.

```
[root@docker maven]# aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 163083073206.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@docker maven]# docker build -t my-ecr-hari .
ERROR: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
[root@docker maven]# systemctl restart docker
[root@docker maven]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@docker maven]# docker build -t my-ecr-hari .
[+] Building 13.6s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 235B
=> [internal] load metadata for docker.io/library/tomcat:latest
=> => transferring context: 2B
[1/3] FROM docker.io/library/tomcat:latest@sha256:5cfc7100fef1f6f7a07c527524cdc99cd2c8af171a93e34c1c3eb513bd42e93e
=> => resolve docker.io/library/tomcat:latest@sha256:5cfc7100fef1f6f7a07c527524cdc99cd2c8af171a93e34c1c3eb513bd42e93e
=> sha256:3a76363fd5907ff81ec9a71788495f31da0900c1a5679c7a968863240f5057c2.72kB / 2.72kB
=> sha256:2a526:577e9a4b5bae3696d17cefa293cd68a8cf1f9e5e279f7c6eeef1fc67481ccca2 22.96MB / 22.96MB
=> sha256:2b89a0565c11c5b581a4c1cb68597b9e128fb0f7f199cb5240fd42c2c7e56d15 157.65MB / 157.65MB
=> sha256:5cfc7100fef1f6f7a07c527524cd9c2dcha711a3e34c1c3e513bd42e93e 6.64kB / 6.64kB
=> sha256:fc45b7d52d563e34f95f6721a899e8023775c00cb3867d974827467c4 159B / 159B
=> sha256:3ae9dfad7a3a525bc34083d04e4eb6cf0804bb4eafec955c44dabdea8f14 159B / 159B
=> sha256:4ab6fa3d88a83c661740f09a789f19cc5a3cea13e17c94d7a483136ab23ad42 2.28kB / 2.28kB
=> sha256:4f4fb780e15461cefa2571a0edb9a0dc1e0cd657748046d75e58dc38e8ac1 32B / 32B
=> sha256:c1bc010d93f431a8a3396c709b77ef840075209924ad1ca0a3581911f4b9ab15 17.92MB / 17.92MB
=> extracting sha256:b08e2ff4391ef70ca7936ba731d1f21a75febd86edc403cd1514a099615888
=> extracting sha256:557e9a4b5bae3696d17cefa293cd68a8cf1f9e5e279f7c6eeef1fc67481ccca2
=> extracting sha256:2b00a0b65c11c5b581a4c1cb68597b9e128fb0f7f199cb5240fd42c2c7e56d15
=> extracting sha256:fc45b7d52d563e34f95f6721a899e8023775c00cb3867d974827467c4
=> extracting sha256:3ae9dfad7a3a525bc34083d04e4eb6cf0804bb4eafec955c44dabdea8f14
=> extracting sha256:4f4fb780e15461cefa2571a0edb9a0dc1e0cd657748046d75e58dc38e8ac1
=> => extracting sha256:1bc010d93f431a8a3396c709b77ef8400b7520924ad1ca0a3581911f4b9ab15
=> [internal] load build context
```

Push cmd is pasted and successfully logged in.

```
[root@ip-172-31-28-224/opt/]# + + +
=> sha256:b08e2ff4391ef70ca7936ba731d1f21a75febd86edc403cd1514a099615888 29.72MB / 29.72MB
=> sha256:557e9a4b5bae3696d17cefa293cd68a8cf1f9e5e279f7c6eeef1fc67481ccca2 22.96MB / 22.96MB
=> sha256:2b00a0b65c11c5b581a4c1cb68597b9e128fb0f7f199cb5240fd42c2c7e56d15 157.65MB / 157.65MB
=> sha256:fc45b7d52d563e34f95f6721a899e8023775c00cb3867d974827467c4 6.64kB / 6.64kB
=> sha256:3ae9dfad7a3a525bc34083d04e4eb6cf0804bb4eafec955c44dabdea8f14 159B / 159B
=> sha256:4ab6fa3d88a83c661740f09a789f19cc5a3cea13e17c94d7a483136ab23ad42 2.28kB / 2.28kB
=> sha256:4f4fb780e15461cefa2571a0edb9a0dc1e0cd657748046d75e58dc38e8ac1 32B / 32B
=> sha256:c1bc010d93f431a8a3396c709b77ef840075209924ad1ca0a3581911f4b9ab15 17.92MB / 17.92MB
=> extracting sha256:b08e2ff4391ef70ca7936ba731d1f21a75febd86edc403cd1514a099615888
=> extracting sha256:557e9a4b5bae3696d17cefa293cd68a8cf1f9e5e279f7c6eeef1fc67481ccca2
=> extracting sha256:2b00a0b65c11c5b581a4c1cb68597b9e128fb0f7f199cb5240fd42c2c7e56d15
=> extracting sha256:fc45b7d52d563e34f95f6721a899e8023775c00cb3867d974827467c4
=> extracting sha256:3ae9dfad7a3a525bc34083d04e4eb6cf0804bb4eafec955c44dabdea8f14
=> extracting sha256:4ab6fa3d88a83c661740f09a789f19cc5a3cea13e17c94d7a483136ab23ad42
=> extracting sha256:4f4fb780e15461cefa2571a0edb9a0dc1e0cd657748046d75e58dc38e8ac1
=> => extracting sha256:1bc010d93f431a8a3396c709b77ef8400b7520924ad1ca0a3581911f4b9ab15
=> [internal] load build context
=> transferring context: 2.66kB
[2/3] RUN cp -r /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps
=> [3/3] COPY /webapp/target/*.war /usr/local/tomcat/webapps
=> exporting to image
=> => exporting layers
=> => writing image sha256:c75aaad37b8080elee520a7b8ad489b705616a#bbc8cb2b994lc35b6a2b22a87ab
=> => naming to docker.io/library/my-ecr-hari
WARNING: current commit information was not captured by the build: git was not found in the system: exec: "git": executable file not found in $PATH
[root@docker maven]# docker tag my-ecr-hari:latest 163083073206.dkr.ecr.us-east-1.amazonaws.com/my-ecr-hari:latest
[root@docker maven]# docker push 163083073206.dkr.ecr.us-east-1.amazonaws.com/my-ecr-hari:latest
The push refers to repository [163083073206.dkr.ecr.us-east-1.amazonaws.com/my-ecr-hari]
769617597e20: Pushed
5a46c54d5ebe: Pushed
5f79bf18a086: Pushed
a5b50fa46aee: Pushed
6ac6adfa9259: Pushed
9574addb8357: Pushed
37d126a69996: Pushed
78635f3af26b: Pushed
4d8ccb8462b9: Pushed
45a01f98e78c: Pushed
latest: digest: sha256:c75aaad37b8080elee520a7b8ad489b705616a#bbc8cb2b994lc35b6a2b22a87ab size: 2620
[root@docker maven]#
```

Image is created and pushed into the ECR.

The screenshot shows the AWS Amazon Elastic Container Registry (Amazon ECR) interface. On the left, there's a sidebar with navigation links for 'Private registry' (Repositories, Summary, Images, Permissions, Lifecycle Policy, Repository tags, Features & Settings) and 'Public registry' (Repositories, Settings). Below that is a section for 'ECR public gallery' with links to 'Amazon ECS' and 'Amazon EKS'. The main area is titled 'Images (1)' and lists a single image entry:

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
latest	Image	July 10, 2025, 15:23:34 (UTC+0:5)	231.42	<a href="#">Copy URI</a>	<a href="#">sha256:3179e68c9c77f4f...</a>	-

Latest image is available.

```
[root@ip-172-31-31-243-] ~ + | x
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\10835505\OneDrive - LTIMindtree\Desktop> ssh -i "aws_Ltimindtree.pem" ec2-user@ec2-54-235-14-118.compute-1.amazonaws.com
The authenticity of host 'ec2-54-235-14-118.compute-1.amazonaws.com (54.235.14.118)' can't be established.
ED25519 key fingerprint is SHA256:4RjukB9ClxwgbzMEWzLGSRMDrekUtgJqfrLPvjE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-235-14-118.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_ _ _ _ _ Amazon Linux 2023
~~~ \####\_
~~~ \|##|
~~~  '/----> https://aws.amazon.com/linux/amazon-linux-2023
~~~   \_/
~~~    /_
~~~     /_
~~~      /_
~~~       /_
~~~        /_
~~~         /_
~~~          /_
~~~           /_
~~~            /_
~~~             /_
~~~              /_
~~~               /_
~~~                /_
~~~                 /_
~~~                  /_
~~~                   /_
~~~                     /_
~~~                       /_
~~~                         /_
~~~                           /_
~~~                             /_
~~~                               /_
~~~                                 /_
~~~                                   /_
~~~                                     /_
~~~                                       /_
~~~                                         /_
~~~                                           /_
~~~                                             /_
~~~                                               /_
~~~                                                 /_
~~~                                                   /_
~~~                                                     /_
~~~                                                       /_
~~~                                                         /_
~~~                                                           /_
~~~                                                             /_
~~~                                                               /_
~~~                                                                 /_
~~~                                                               bash
[ec2-user@ip-172-31-31-243 ~]$ sudo su -
[root@ip-172-31-31-243 ~]# hostnamectl set-hostname eks.example.com
[root@ip-172-31-31-243 ~]# bash
[root@eks ~]# yum update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[root@eks ~]# curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
[root@eks ~]# sudo mv /tmp/eksctl /usr/local/bin
[root@eks ~]# eksctl version
0.210.0
[root@eks ~]# curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
% total % Received % Xferd Average Speed Time Time Current
% total 100 53.7M 0 0 116M 0 ---:---:---:--- 115M
[root@eks ~]# sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[root@eks ~]#
```

Eks instance is accessible. Necessary dependencies are installed.

```
[root@eks ~]# eksctl create cluster --name my-cluster-hariram --region us-east-1 --version 1.32 --vpc-public-subnets subnet-00608da34c56e569d,subnet-0541b7ff15bd92fd9 --without-nodegroup
2025-07-10 10:01:36 [ ] eksctl version 1.32
2025-07-10 10:01:36 [ ] using region us-east-1
2025-07-10 10:01:37 [ ] using existing VPC (vpc-0c1e9f84aa9d9d9e) and subnets (private:map[] public:map[us-east-1a:[subnet-00608da34c56e569d us-east-1a 172.31.16.0/20 0 } us-east-1b:[subnet-0541b7ff15bd92fd9 0 }])
2025-07-10 10:01:37 [!] custom VPC/subnets will be used; if resulting cluster doesn't function as expected, make sure to review the configuration of VPC/subnets
2025-07-10 10:01:37 [ ] using Kubernetes version 1.32
2025-07-10 10:01:37 [ ] creating EKS cluster "my-cluster-hariram" in "us-east-1" region with
2025-07-10 10:01:37 [ ]   if you encounter any issues, check CloudFormation console or try `eksctl utils describe-stacks --region=us-east-1 --cluster=my-cluster-hariram`
2025-07-10 10:01:37 [ ]   API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "my-cluster-hariram" in "us-east-1"
2025-07-10 10:01:37 [ ]   CloudWatch logging will not be enabled for cluster "my-cluster-hariram" in "us-east-1"
2025-07-10 10:01:37 [ ]   you can enable it with `eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=my-cluster-hariram`
2025-07-10 10:01:37 [ ]   default addons kube-proxy, coredns, metrics-server, vpc-cni were not specified, will install them as EKS addons
2025-07-10 10:01:37 [ ] 2 sequential tasks: [ create cluster control plane "my-cluster-hariram",
2025-07-10 10:01:37 [ ]   1 task: [ create addons ],
2025-07-10 10:01:37 [ ]     wait for control plane to become ready,
2025-07-10 10:01:37 [ ]
2025-07-10 10:01:37 [ ]   building cluster stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:01:37 [ ]   deploying stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:02:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:02:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:03:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:04:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:05:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:06:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:07:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:08:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:09:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:10:37 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-cluster"
2025-07-10 10:10:38 [ ]   successfully created addon: kube-proxy
2025-07-10 10:10:38 [ ]   successfully created addon: coredns
2025-07-10 10:10:39 [ ]   successfully created addon: metrics-server
2025-07-10 10:10:40 [ ]   successfully created addon: metrics-server
2025-07-10 10:10:40 [ ]   successfully created addon: metrics-server
2025-07-10 10:10:40 [ ]   recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended
2025-07-10 10:10:40 [ ]   way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentity'
2025-07-10 10:10:40 [ ]   run `eksctl update-addon` to update the addon
2025-07-10 10:10:40 [ ]   creating addon: vpc-cni
2025-07-10 10:10:41 [ ]   successfully created addon: vpc-cni
2025-07-10 10:12:41 [ ]   waiting for the control plane to become ready
2025-07-10 10:12:42 [ ]   saved kubeconfig as "/root/.kube/config"
2025-07-10 10:12:42 [ ]   no tasks
2025-07-10 10:12:42 [ ]   all EKS cluster resources for "my-cluster-hariram" have been created
2025-07-10 10:12:43 [ ]   kubectl command should work with "/root/.kube/config", try `kubectl get nodes`
2025-07-10 10:12:43 [ ]   EKS cluster "my-cluster-hariram" in "us-east-1" region is ready
```

Cluster is created.

```
[root@eks ~]# eksctl create nodegroup --cluster my-cluster-hariram --region us-east-1 --name my-node-group --node-ami-family Ubuntu2204 --node-type t2.small --subnet-ids subnet-00608da34c56e569d,subnet-0541b7ff15bd92fd9 --nodes 3 --nodes-min 2 --nodes-max 4 --ssh-access --ssh-public-key /root/.ssh/id_rsa.pub
2025-07-10 10:13:56 [ ] will use version 1.32 for new nodegroup(s) based on control plane version
2025-07-10 10:13:56 [ ] nodegroup "my-node-group" will use "ami-087657d77f5892006" [Ubuntu2204/1.32]
2025-07-10 10:13:57 [ ] using SSH public key "/root/.ssh/id_rsa.pub" as "eksctl-my-cluster-hariram-nodegroup-my-node-group-4d:48:28:15:5e:b2:bc:37:f4:cc:93:86:f3:83:9b:a2"
2025-07-10 10:13:57 [ ] 1 nodegroup (my-node-group) was included (based on the include/exclude rules)
2025-07-10 10:13:57 [ ] will create a CloudFormation stack for each of 1 managed nodegroups in cluster "my-cluster-hariram"
2025-07-10 10:13:57 [ ] 2 sequential tasks: [ fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "my-node-group" } }
2025-07-10 10:13:57 [ ]   checking cluster stack for missing resources
2025-07-10 10:13:57 [ ]   cluster stack has all required resources
2025-07-10 10:13:57 [ ]   building managed nodegroup stack "eksctl-my-cluster-hariram-nodegroup-my-node-group"
2025-07-10 10:13:57 [ ]   deploying stack "eksctl-my-cluster-hariram-nodegroup-my-node-group"
2025-07-10 10:13:58 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-nodegroup-my-node-group"
2025-07-10 10:14:28 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-nodegroup-my-node-group"
2025-07-10 10:15:06 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-nodegroup-my-node-group"
2025-07-10 10:16:58 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-nodegroup-my-node-group"
2025-07-10 10:17:33 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-nodegroup-my-node-group"
2025-07-10 10:18:54 [ ]   waiting for CloudFormation stack "eksctl-my-cluster-hariram-nodegroup-my-node-group"
2025-07-10 10:18:54 [ ]   no tasks
2025-07-10 10:18:54 [ ]   created 0 nodegroup(s) in cluster "my-cluster-hariram"
2025-07-10 10:18:55 [ ]   nodegroup "my-node-group" has 3 node(s)
2025-07-10 10:18:55 [ ]   node "ip-172-31-16-173.ec2.internal" is ready
2025-07-10 10:18:55 [ ]   node "ip-172-31-22-95.ec2.internal" is ready
2025-07-10 10:18:55 [ ]   node "ip-172-31-43-226.ec2.internal" is ready
2025-07-10 10:18:55 [ ]   waiting for at least 2 node(s) to become ready in "my-node-group"
2025-07-10 10:18:55 [ ]   nodegroup "my-node-group" has 3 node(s)
2025-07-10 10:18:55 [ ]   node "ip-172-31-16-173.ec2.internal" is ready
2025-07-10 10:18:55 [ ]   node "ip-172-31-22-95.ec2.internal" is ready
2025-07-10 10:18:55 [ ]   node "ip-172-31-43-226.ec2.internal" is ready
2025-07-10 10:18:55 [ ]   created 1 managed nodegroup(s) in cluster "my-cluster-hariram"
2025-07-10 10:18:55 [ ]   checking security group configuration for all nodegroups
2025-07-10 10:18:55 [ ]   all nodegroups have up-to-date cloudformation templates
[root@eks ~]#
```

Node group is created.

The screenshot shows the AWS ECR console with the following navigation path: AWS > Amazon ECR > Private registry > Repositories > my-ecr-hari > sha256:3179e68c9c77f4f5eb16a9165e9597a2b4a6c29e8f7df7db240f76d8e0b... . The main panel displays the 'Image' details for the repository 'my-ecr-hari'. It includes sections for 'Image tags' (with a tooltip 'Repository URI copied' over the latest tag), 'General information' (Artifact type: Image, Repository: my-ecr-hari, Pushed at: July 10, 2025, 15:23:34 (UTC+05.5)), and 'Last recorded pull time' ( - ). On the left sidebar, there are sections for 'Private registry' (Repositories, Summary, Images, Permissions, Lifecycle Policy, Repository tags) and 'Public registry' (Repositories, Settings). At the bottom, there are links for 'ECR public gallery', 'Amazon ECS', and 'Amazon EKS'.

URI of the latest image is copied.

```
root@ip-172-31-243-: ~ % cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: regapp-deployment
  labels:
    app: regapp
spec:
  replicas: 2
  selector:
    matchLabels:
      app: regapp
  template:
    metadata:
      labels:
        app: regapp
    spec:
      containers:
        - name: regapp
          image: 163083073206.dkr.ecr.us-east-1.amazonaws.com/my-ecr-hari:latest
          imagePullPolicy: Always
          ports:
            - containerPort: 8080
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 1
      maxUnavailable: 1
```

Deployment.yaml file is added with latest URI of the image.

```
root@ip-172-31-31-243~ + ~
apiVersion: v1
kind: Service
metadata:
  name: regapp-service
  labels:
    app: regapp
spec:
  selector:
    app: regapp
  ports:
    - port: 8080
      targetPort: 8080
  type: LoadBalancer
~
```

Service.yaml file is added with port enabled.

```
[root@eks ~]# vi deployment.yaml
[root@eks ~]# vi service.yaml
[root@eks ~]# kubectl apply -f deployment.yaml
deployment.apps/regapp-deployment created
[root@eks ~]# kubectl apply -f service.yaml
service/regapp-service created
[root@eks ~]# kubectl get nodes
NAME           STATUS   ROLES   AGE     VERSION
ip-172-31-16-173.ec2.internal   Ready   <none>  8m24s  v1.32.3
ip-172-31-22-95.ec2.internal   Ready   <none>  8m21s  v1.32.3
ip-172-31-43-226.ec2.internal Ready   <none>  8m25s  v1.32.3
[root@eks ~]# kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
regapp-deployment-55b56cbbf6-6956d  1/1     Running   0          35s
regapp-deployment-55b56cbbf6-hmvzq  1/1     Running   0          35s
[root@eks ~]# kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP   10.100.0.1   <none>       443/TCP       17m
regapp-service   LoadBalancer  10.100.149.72  a8e9ccbef0d904bb18618cc34999010d-1752834079.us-east-1.elb.amazonaws.com  8080:30954/TCP  33s
[root@eks ~]#
```

Both files are applied and pods, nodes are in running state. Using svc external ip is taken



Hi this is HARIRAM - 10835505

Please fill in this form to create an account.

Enter Name	<input type="text"/>
Enter mobile	<input type="text"/>
Enter Email	<input type="text"/>
Password	<input type="password"/>
Repeat Password	<input type="password"/>

By creating an account you agree to our [Terms & Privacy](#).

Already have an account? [Sign in](#).

**Thank You, Happy Learning**

**See You Again**

Finally maven project is deployed in live . External\_ip:8080/webapp to access the webpage.