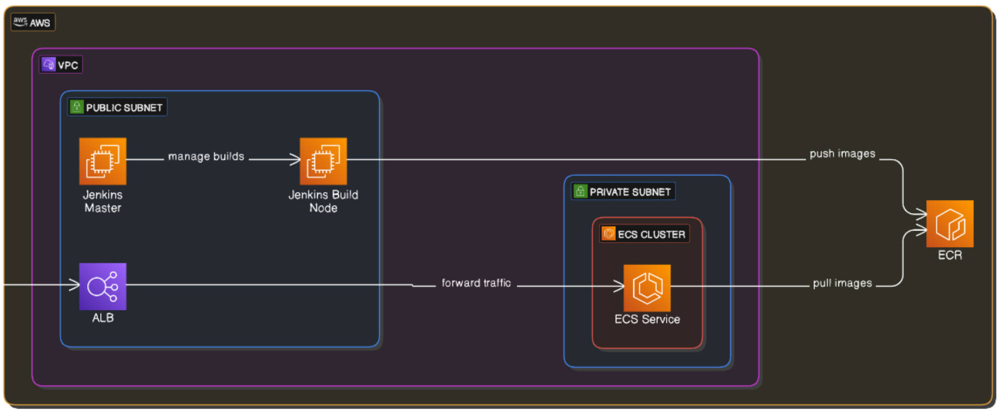
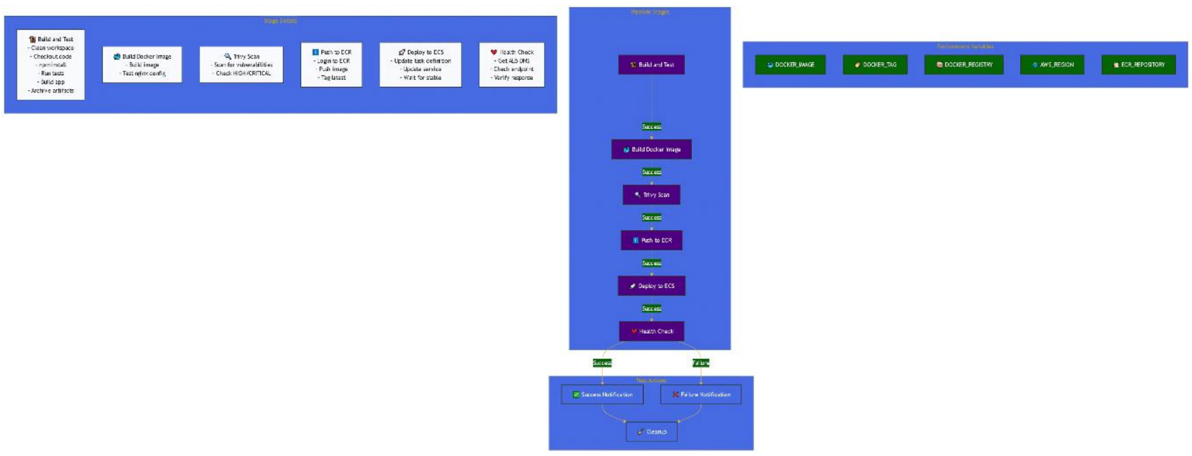
Jenkins CI/CD Pipeline for AWS ECS Deployment





1. Create AWS Key Pair

aws ec2 create-key-pair --key-name jenkins-node-key --query 'KeyMaterial' --output text > jenkins-node-key.pem

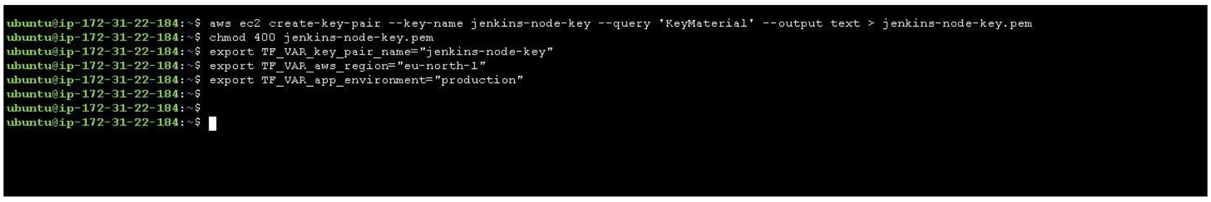
chmod 400 jenkins-node-key.pem

2. Set Required Environment Variables

export TF_VAR_key_pair_name="jenkins-node-key"

export TF_VAR_aws_region="eu-north-1"

export TF_VAR_app_environment="production"



```
ubuntu@ip-172-31-22-184:~$ aws ec2 create-key-pair --key-name jenkins-node-key --query 'KeyMaterial' --output text > jenkins-node-key.pem
ubuntu@ip-172-31-22-184:~$ chmod 400 jenkins-node-key.pem
ubuntu@ip-172-31-22-184:~$ export TF_VAR_key_pair_name="jenkins-node-key"
ubuntu@ip-172-31-22-184:~$ export TF_VAR_aws_region="eu-north-1"
ubuntu@ip-172-31-22-184:~$ export TF_VAR_app_environment="production"
ubuntu@ip-172-31-22-184:~$
ubuntu@ip-172-31-22-184:~$
ubuntu@ip-172-31-22-184:~$
```

3. Clone the project

git clone https://github.com/sandeepkalathil/Jenkins-ECS-Project.git

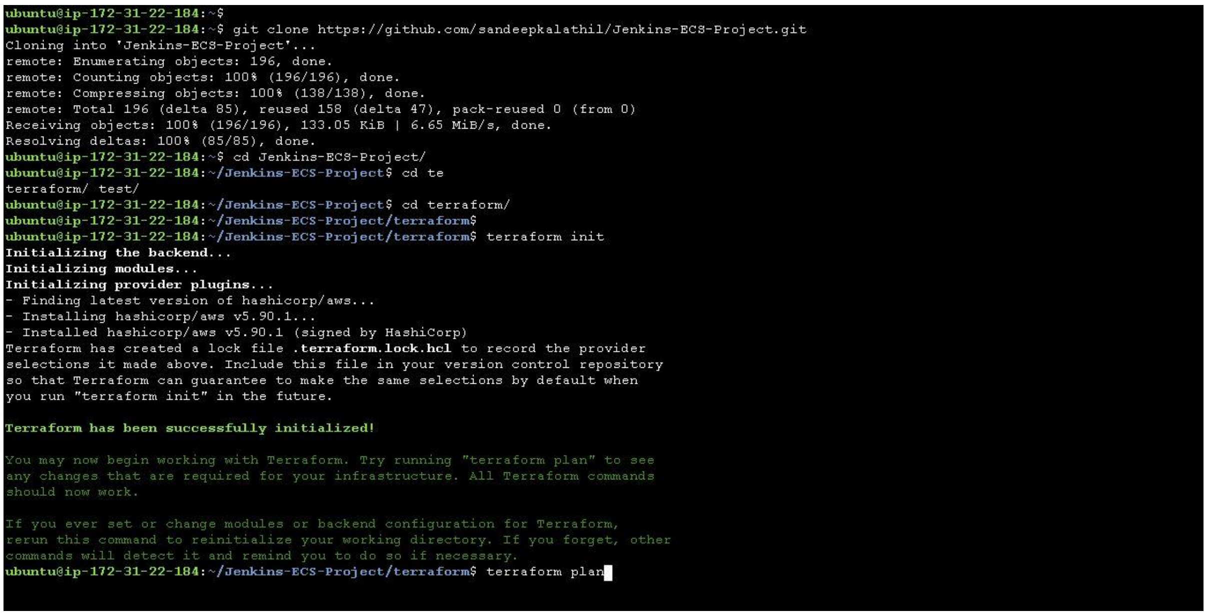
cd Jenkins-ECS-Project/

cd terraform

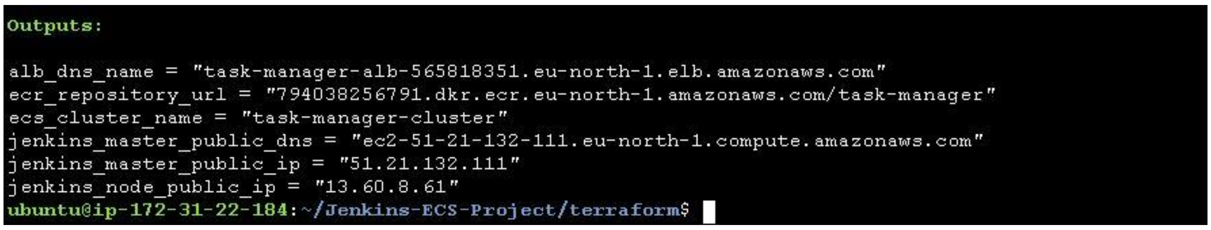
terraform init

terraform plan

terraform apply



```
ubuntu@ip-172-31-22-184:~$
ubuntu@ip-172-31-22-184:~$ git clone https://github.com/sandeepkalathil/Jenkins-ECS-Project.git
Cloning into 'Jenkins-ECS-Project'...
remote: Enumerating objects: 196, done.
remote: Counting objects: 100% (196/196), done.
remote: Compressing objects: 100% (138/138), done.
remote: Total 196 (delta 85), reused 158 (delta 47), pack-reused 0 (from 0)
Receiving objects: 100% (196/196), 133.05 KiB | 6.65 MiB/s, done.
Resolving deltas: 100% (85/85), done.
ubuntu@ip-172-31-22-184:~$ cd Jenkins-ECS-Project/
ubuntu@ip-172-31-22-184:~/Jenkins-ECS-Project$ cd te
terraform/ test/
ubuntu@ip-172-31-22-184:~/Jenkins-ECS-Project$ cd terraform/
ubuntu@ip-172-31-22-184:~/Jenkins-ECS-Project/terraform$
ubuntu@ip-172-31-22-184:~/Jenkins-ECS-Project/terraform$ terraform init
Initializing the backend...
Initializing modules...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.90.1...
- Installed hashicorp/aws v5.90.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-22-184:~/Jenkins-ECS-Project/terraform$ terraform plan
```



```
Outputs:

alb_dns_name = "task-manager-alb-565818351.eu-north-1.elb.amazonaws.com"
ecr_repository_url = "794038256791.dkr.ecr.eu-north-1.amazonaws.com/task-manager"
ecs_cluster_name = "task-manager-cluster"
jenkins_master_public_dns = "ec2-51-21-132-111.eu-north-1.compute.amazonaws.com"
jenkins_master_public_ip = "51.21.132.111"
jenkins_node_public_ip = "13.60.8.61"
ubuntu@ip-172-31-22-184:~/Jenkins-ECS-Project/terraform$
```

Key Components of the Infrastructure

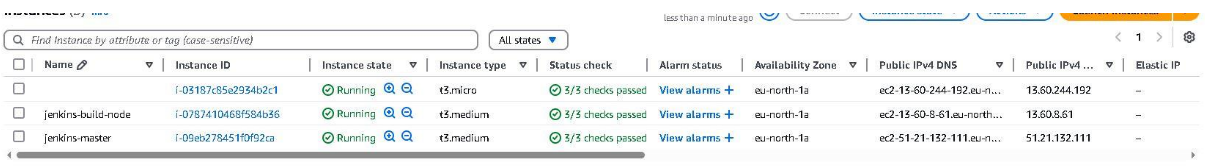
This Terraform script deploys a Jenkins master and build node on AWS EC2, along with an ECS-based web application. Below are the key components:

1. **Networking (VPC Module)**:
   - Creates a VPC with public and private subnets.
   - Configures a single NAT gateway for outbound internet access from private subnets.
2. **Jenkins Master and Node EC2 Instances**:
   - Installs Jenkins, Docker, and AWS CLI on the master node.
   - The build node is configured with Docker and Java for running builds.
   - Security groups allow SSH and Jenkins access (though the wide-open ingress rules should be restricted for production).
3. **ECR for Container Storage**:
   - Creates an Elastic Container Registry (ECR) for storing Docker images.
4. **ECS Cluster and Fargate Task Definition**:
   - Defines an ECS cluster and Fargate-based service.
   - Deploys the containerized application with an ALB for load balancing.
   - Manages IAM roles and policies for task execution and logging.
5. **Security and IAM Roles**:
   - Separate security groups for ALB and ECS tasks.
   - IAM role for ECS task execution with access to ECR and CloudWatch logs.

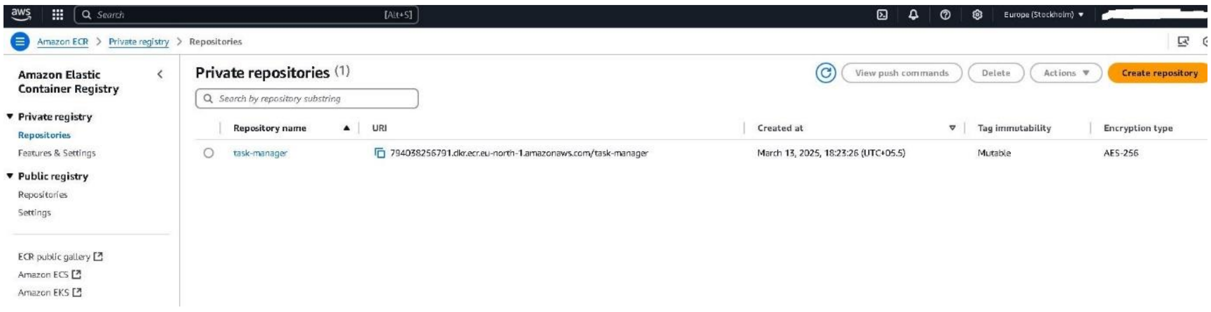
Suggestions for Improvement:

1. **Security Hardening**:
   - Restrict the security group ingress rules to specific IP ranges instead of 0.0.0.0/0.
   - Use AWS Secrets Manager or SSM Parameter Store for sensitive credentials.
2. **Scalability and Monitoring**:
   - Add Auto Scaling policies for ECS services.
   - Integrate CloudWatch alarms for monitoring.

3. **CI/CD Integration**:
   - Set up a Jenkins pipeline to build and push Docker images to ECR.
   - Automate ECS service updates with Blue-Green or Canary deployments.



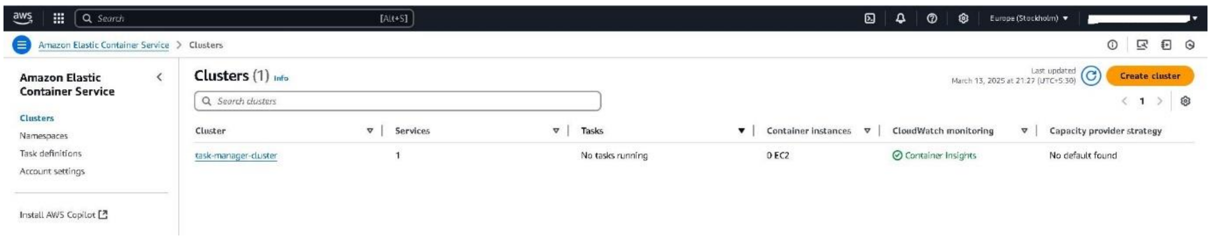
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
	i-03187c85e2934b2c1	Running	t3.micro	3/3 checks passed	View alarms +	eu-north-1a	ec2-13-60-244-192.eu-n...	13.60.244.192	–
jenkins-build-node	i-0787410468f584b36	Running	t3.medium	3/3 checks passed	View alarms +	eu-north-1a	ec2-13-60-8-61.eu-north...	13.60.8.61	–
jenkins-master	i-09eb278451f0f92ca	Running	t3.medium	3/3 checks passed	View alarms +	eu-north-1a	ec2-51-21-132-111.eu-n...	51.21.132.111	–



[redacted]

Private repositories (1)

Repository name	URI	Created at	Tag immutability	Encryption type
task-manager	794038256791.dkr.ecr.eu-north-1.amazonaws.com/task-manager	March 13, 2025, 18:23:26 (UTC+05.5)	Mutable	AES-256



[redacted]

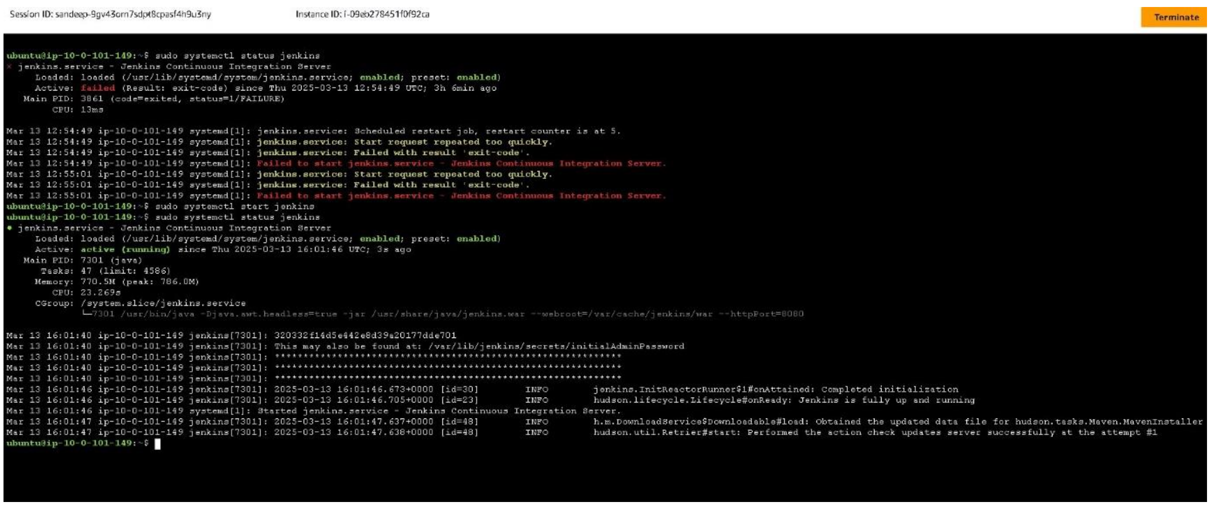
Clusters (1)

Last updated March 13, 2025 at 21:27 (UTC+5:30)

Cluster	Services	Tasks	Container instances	CloudWatch monitoring	Capacity provider strategy
task-manager-cluster	1	No tasks running	0 EC2	Container Insights	No default found

Jenkins Setup

On the Jenkins master server make sure that the Jenkins service is running, else start the service.


Session ID: sandeep-9gv43orn7sdpt8cpasf4h9u3ny
Instance ID: i-09eb278451f0f92ca
Terminate

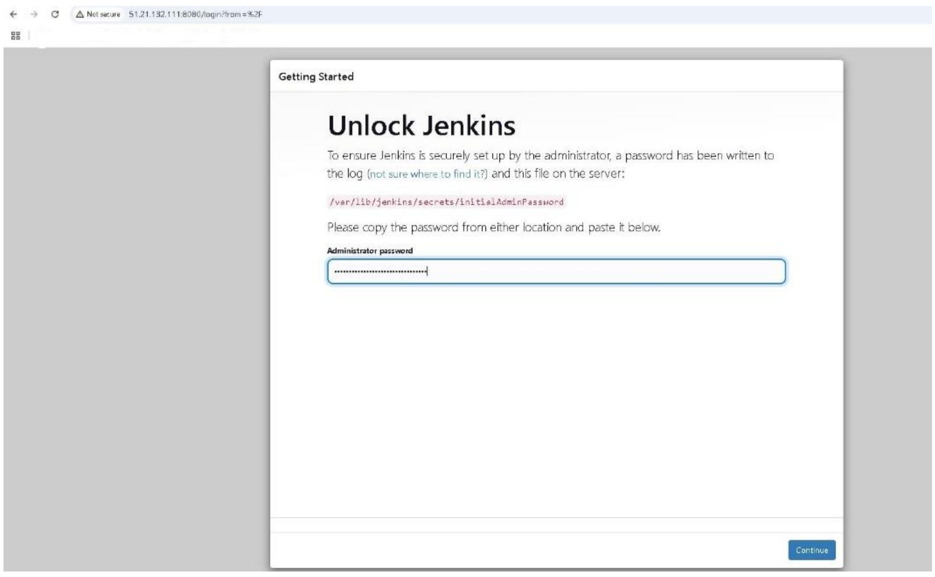
```
ubuntu@ip-10-0-101-149:~$ sudo systemctl status jenkins
× jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: failed (Result: exit-code) since Thu 2025-03-13 12:54:49 UTC; 3h 6min ago
   Main PID: 3861 (code=exited, status=1/FAILURE)
        CPU: 13ms

Mar 13 12:54:49 ip-10-0-101-149 systemd[1]: jenkins.service: Scheduled restart job, restart counter is at 5.
Mar 13 12:54:49 ip-10-0-101-149 systemd[1]: jenkins.service: Start request repeated too quickly.
Mar 13 12:54:49 ip-10-0-101-149 systemd[1]: jenkins.service: Failed with result 'exit-code'.
Mar 13 12:54:49 ip-10-0-101-149 systemd[1]: Failed to start jenkins.service - Jenkins Continuous Integration Server.
Mar 13 12:55:01 ip-10-0-101-149 systemd[1]: jenkins.service: Start request repeated too quickly.
Mar 13 12:55:01 ip-10-0-101-149 systemd[1]: jenkins.service: Failed with result 'exit-code'.
Mar 13 12:55:01 ip-10-0-101-149 systemd[1]: Failed to start jenkins.service - Jenkins Continuous Integration Server.
ubuntu@ip-10-0-101-149:~$ sudo systemctl start jenkins
ubuntu@ip-10-0-101-149:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-03-13 16:01:46 UTC; 3s ago
   Main PID: 7301 (java)
      Tasks: 47 (limit: 4586)
     Memory: 770.5M (peak: 786.0M)
        CPU: 23.269s
     CGroup: /system.slice/jenkins.service
             └─7301 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Mar 13 16:01:40 ip-10-0-101-149 jenkins[7301]: 320332f14d5e442e8d39a20177dde701
Mar 13 16:01:40 ip-10-0-101-149 jenkins[7301]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 13 16:01:40 ip-10-0-101-149 jenkins[7301]: *************************************************************
Mar 13 16:01:40 ip-10-0-101-149 jenkins[7301]: *************************************************************
Mar 13 16:01:40 ip-10-0-101-149 jenkins[7301]: *************************************************************
Mar 13 16:01:46 ip-10-0-101-149 jenkins[7301]: 2025-03-13 16:01:46.673+0000 [id=30]        INFO        jenkins.InitReactorRunner$1#onAttained: Completed initialization
Mar 13 16:01:46 ip-10-0-101-149 jenkins[7301]: 2025-03-13 16:01:46.705+0000 [id=23]        INFO        hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
Mar 13 16:01:46 ip-10-0-101-149 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Mar 13 16:01:47 ip-10-0-101-149 jenkins[7301]: 2025-03-13 16:01:47.637+0000 [id=48]        INFO        h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
Mar 13 16:01:47 ip-10-0-101-149 jenkins[7301]: 2025-03-13 16:01:47.638+0000 [id=48]        INFO        hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1
ubuntu@ip-10-0-101-149:~$
```

1. Access Jenkins Master

- Get Jenkins master public IP from Terraform output
- Access Jenkins UI: http://<jenkins_master_public_ip>:8080
- Get initial admin password:

ssh -i jenkins-node-key.pem ubuntu@<jenkins_master_public_ip>

sudo cat /var/lib/jenkins/secrets/initialAdminPassword



Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

/var/lib/jenkins/secrets/initialAdminPassword

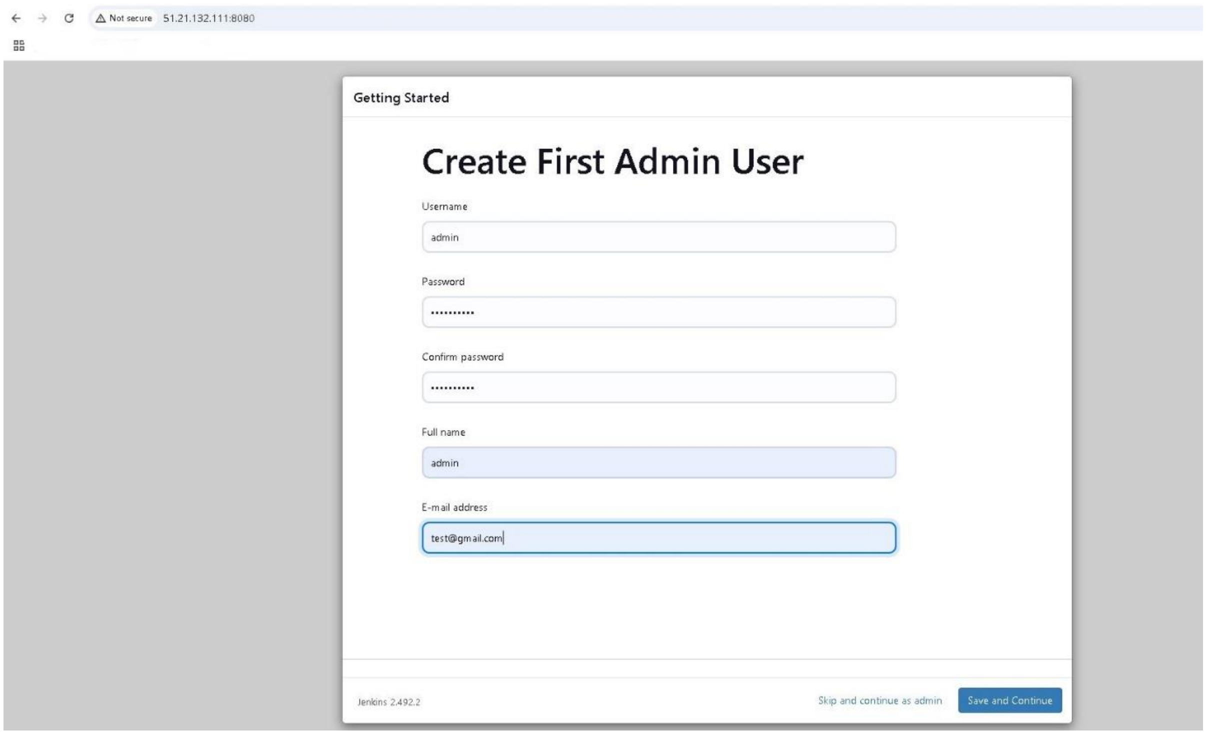
Please copy the password from either location and paste it below.

Administrator password

································

2. Configure Jenkins Master

1. Install suggested plugins
2. Create admin user
3. Install additional plugins:
    - Docker Pipeline
    - AWS Credentials
    - Blue Ocean



Create First Admin User

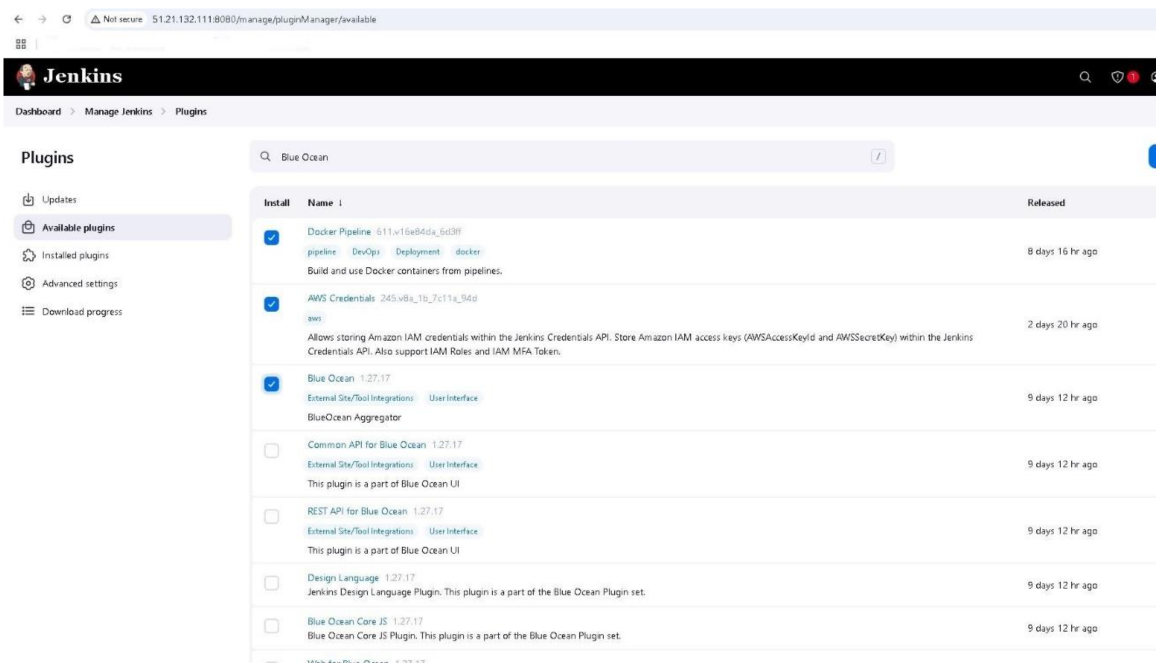
Username: admin

Password: ••••••••••

Confirm password: ••••••••••

Full name: admin

E-mail address: test@gmail.com



Install	Name ↓	Released
✓	Docker Pipeline 611.v16e84da_6d3ff pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	8 days 16 hr ago
✓	AWS Credentials 245.v8a_1b_7c11a_94d aws Allows storing Amazon IAM credentials within the Jenkins Credentials API. Store Amazon IAM access keys (AWSAccessKeyId and AWSSecretKey) within the Jenkins Credentials API. Also support IAM Roles and IAM MFA Token.	2 days 20 hr ago
✓	Blue Ocean 1.27.17 External Site/Tool Integrations User Interface BlueOcean Aggregator	9 days 12 hr ago
	Common API for Blue Ocean 1.27.17 External Site/Tool Integrations User Interface This plugin is a part of Blue Ocean UI	9 days 12 hr ago
	REST API for Blue Ocean 1.27.17 External Site/Tool Integrations User Interface This plugin is a part of Blue Ocean UI	9 days 12 hr ago
	Design Language 1.27.17 Jenkins Design Language Plugin. This plugin is a part of the Blue Ocean Plugin set.	9 days 12 hr ago
	Blue Ocean Core JS 1.27.17 Blue Ocean Core JS Plugin. This plugin is a part of the Blue Ocean Plugin set.	9 days 12 hr ago

Docker Pipeline Plugin

- **Purpose**: Allows Jenkins to interact with Docker containers directly within the pipeline.
- **Usage**: It enables building, running, and managing Docker containers and images as part of the CI/CD pipeline. This is especially useful for creating isolated build environments, running tests in containers, and deploying containerized applications to platforms like AWS ECS or Kubernetes.
- **Example Scenario**: Building a Docker image from a Jenkins pipeline and pushing it to AWS ECR.

---

2. AWS Credentials Plugin

- **Purpose**: Provides a secure way to manage AWS credentials within Jenkins.
- **Usage**: Allows Jenkins to access AWS resources such as S3, EC2, and ECS by securely storing and managing IAM user credentials or access tokens. It integrates with AWS CLI and SDKs, enabling actions like uploading artifacts to S3 or deploying infrastructure with Terraform.
- **Example Scenario**: Deploying a Docker container to AWS ECS or updating CloudFormation stacks from a Jenkins pipeline.

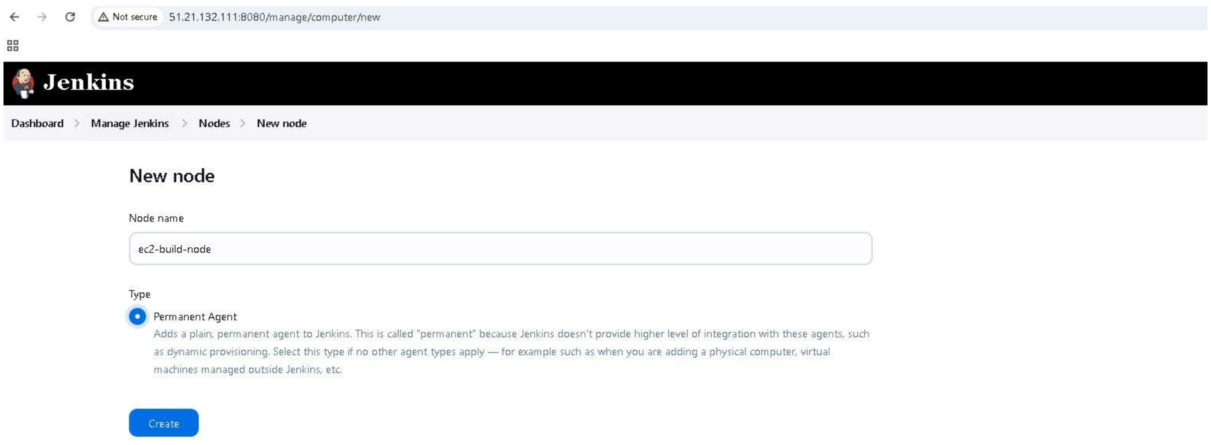
---

3. Blue Ocean Plugin

- **Purpose**: Provides a modern, user-friendly interface for Jenkins pipelines.
- **Usage**: Offers visual pipeline editing and monitoring, making it easier to understand and manage complex CI/CD workflows. It also supports pipeline visualization, parallel execution, and real-time status updates.
- **Example Scenario**: Viewing the stages of a CI/CD pipeline for a microservices deployment and debugging any failed steps visually.

3. Configure Jenkins Build Node

1. Go to Manage Jenkins → Manage Nodes
2. Add new node:
   - Name: ec2-build-node
   - Permanent Agent: Yes
   - Remote root directory: /home/ubuntu/jenkins-agent
   - Labels: ec2-build-node
   - Launch method: Launch agent via SSH
   - Host: <EC2_INSTANCE_PUBLIC_IP> (from Terraform output)
   - Credentials: Add SSH with private key
   - Host Key Verification Strategy: Non verifying



Not secure 51.21.132.111:8080/manage/computer/new

Jenkins

Dashboard > Manage Jenkins > Nodes > New node

New node

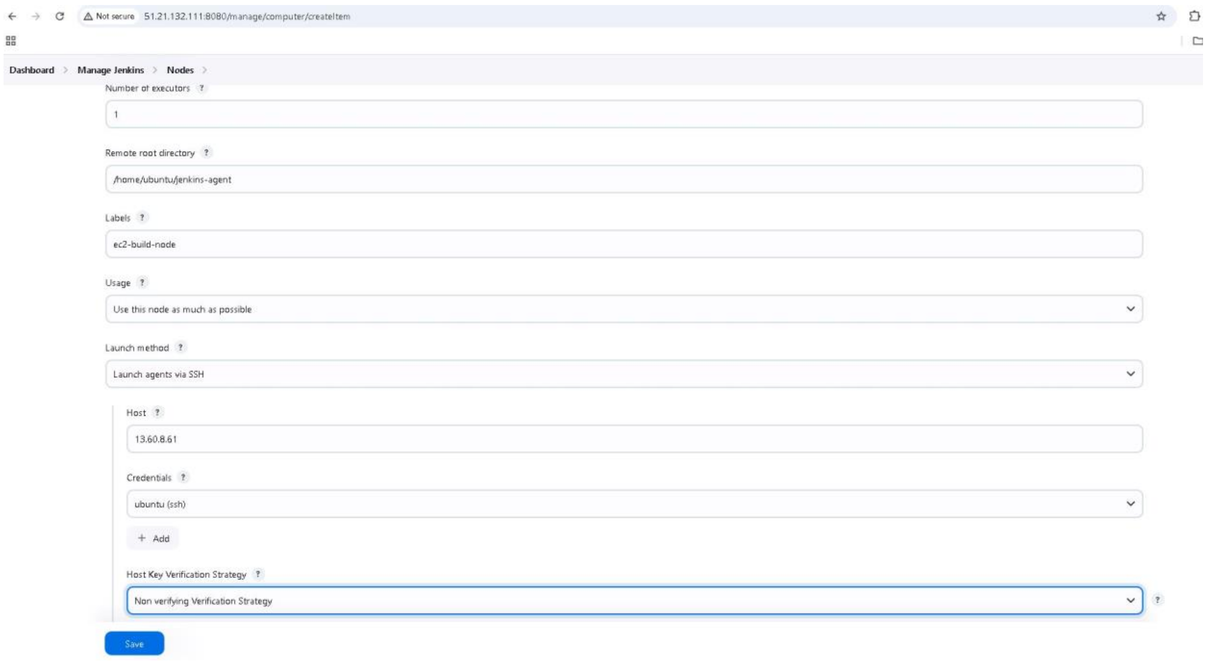
Node name

ec2-build-node

Type

[✓ Permanent Agent]
Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create



Not secure 51.21.132.111:8080/manage/computer/createItem

Dashboard > Manage Jenkins > Nodes >

Number of executors: 1

Remote root directory: /home/ubuntu/jenkins-agent

Labels: ec2-build-node

Usage: Use this node as much as possible

Launch method: Launch agents via SSH

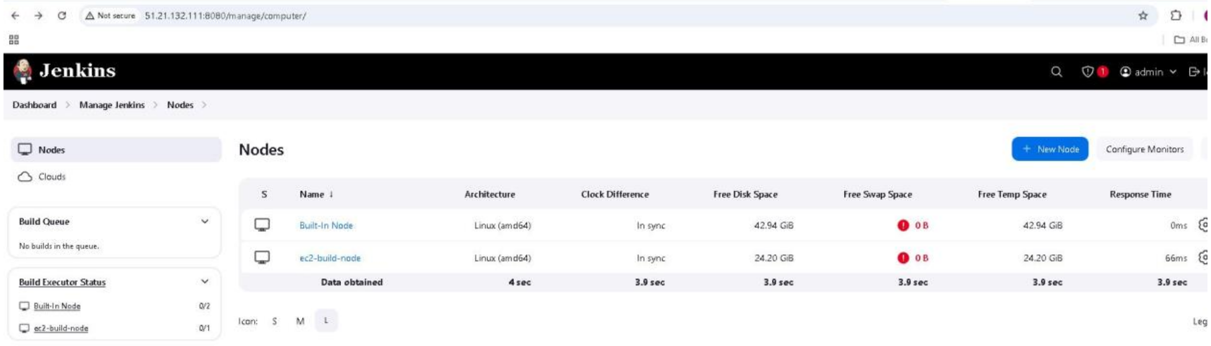
Host: 13.60.8.61

Credentials: ubuntu (ssh)

+ Add

Host Key Verification Strategy: Non verifying Verification Strategy

Save



Not secure 51.21.132.111:8080/manage/computer/

Jenkins

Dashboard > Manage Jenkins > Nodes >

Nodes

Clouds

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node	0/2
ec2-build-node	0/1

Nodes

+ New Node Configure Monitors

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	42.94 GiB	❗ 0 B	42.94 GiB	0ms
	ec2-build-node	Linux (amd64)	In sync	24.20 GiB	❗ 0 B	24.20 GiB	66ms
	Data obtained	4 sec	3.9 sec	3.9 sec	3.9 sec	3.9 sec	3.9 sec

Icon: S M L

4. Configure Jenkins Credentials

1. AWS Credentials:

   - Kind: AWS Credentials
   - ID: aws-credentials
   - Description: AWS Credentials
   - Access Key ID: Your AWS access key
   - Secret Access Key: Your AWS secret key

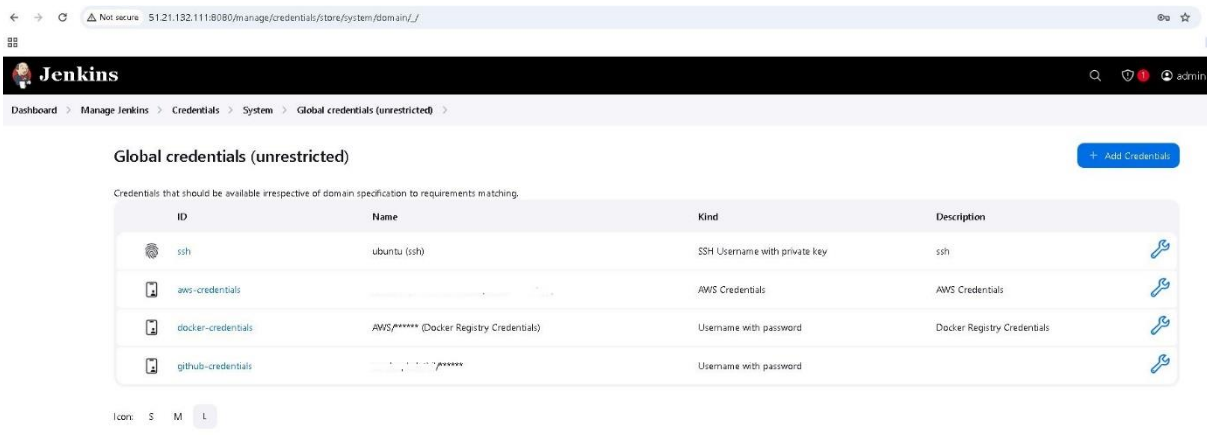
2. Docker Registry:

   - Kind: Username with password
   - ID: docker-credentials
   - Description: Docker Registry Credentials
   - Username: AWS
   - Password: (Use AWS CLI get-login-password output)

Use command to generate password : aws ecr get-login-password --region eu-north-1

3. GitHub:

   - Kind: Username with password
   - ID: github-credentials
   - Add your GitHub credentials



Not secure 51.21.132.111:8080/manage/credentials/store/system/domain/_/

Jenkins

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

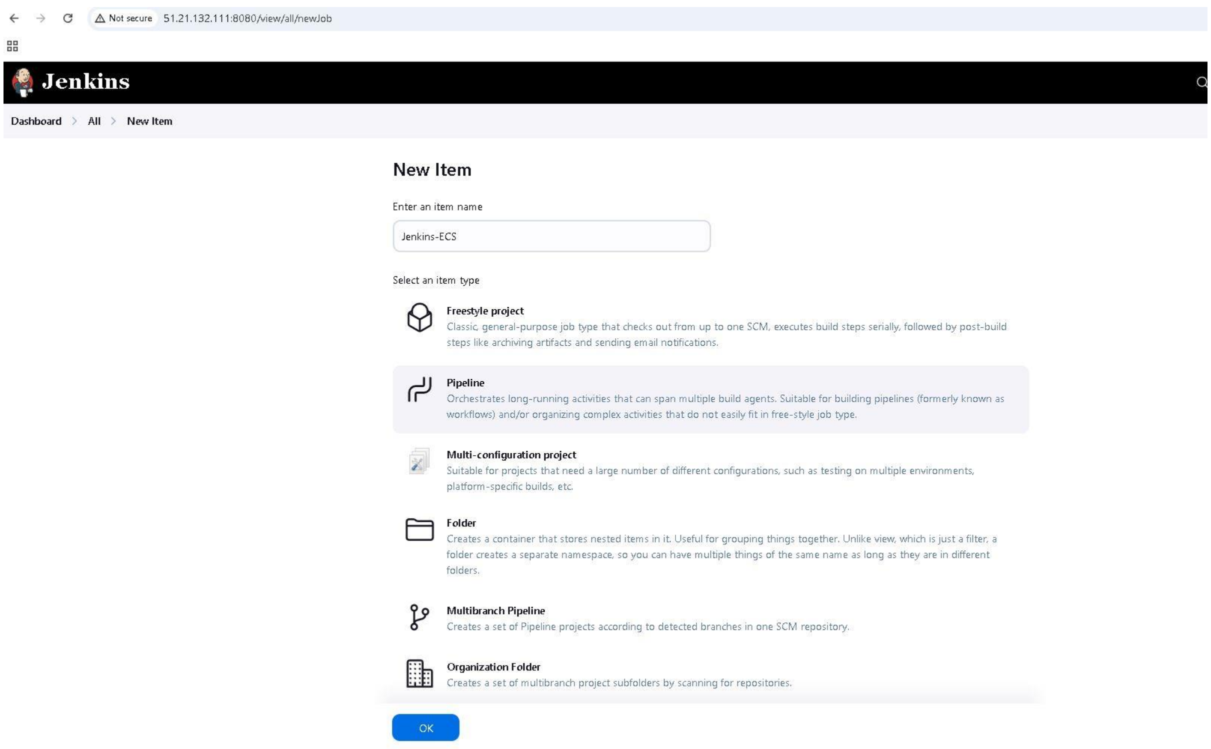
Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
ssh	ubuntu (ssh)	SSH Username with private key	ssh
aws-credentials	[redacted]	AWS Credentials	AWS Credentials
docker-credentials	AWS/****** (Docker Registry Credentials)	Username with password	Docker Registry Credentials
github-credentials	[redacted]/******	Username with password	

Pipeline Setup

1. Create Jenkins Pipeline

1. New Item → Pipeline
2. Configure Pipeline:
    - Definition: Pipeline script from SCM
    - SCM: Git
    - Repository URL: Your repository URL
    - Credentials: github-credentials (in case of Private Repo)
    - Branch Specifier: */main
    - Script Path: Jenkinsfile


Not secure 51.21.132.111:8080/view/all/newJob

Jenkins

Dashboard > All > New Item

New Item

Enter an item name

Jenkins-ECS

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

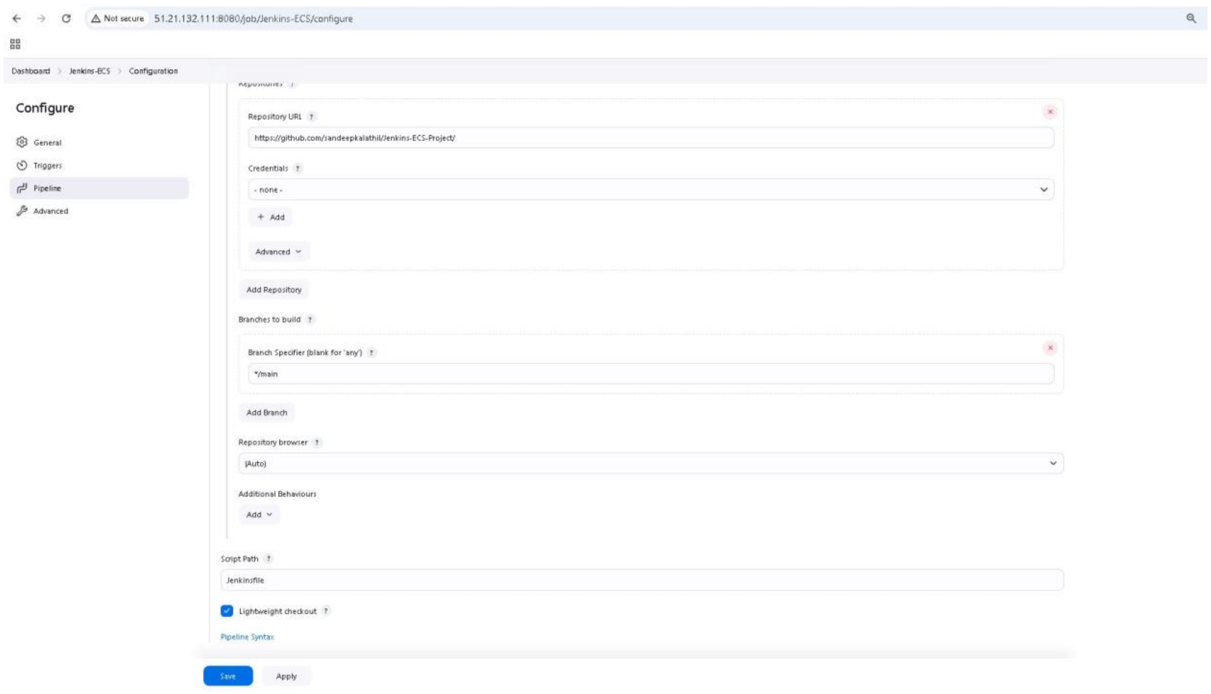
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

OK


Not secure 51.21.132.111:8080/job/Jenkins-ECS/configure

Dashboard > Jenkins-ECS > Configuration

Configure

General
Triggers
Pipeline
Advanced

Repository URL: https://github.com/sandeepkalathil/Jenkins-ECS-Project/

Credentials: - none -

Add

Advanced

Add Repository

Branches to build

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

Add Branch

Repository browser: (Auto)

Additional Behaviours

Add

Script Path: Jenkinsfile

[✓] Lightweight checkout

Pipeline Syntax

Save Apply

2. Pipeline Stages

1. Build and Test

- Runs in Docker container on build node

- NPM install and build
- Unit tests
- Static code analysis

2. Docker Image Creation

- Builds Docker image
- Tests image configuration

3. Security Scan (Trivy)

- Scans for vulnerabilities

4. Push to ECR

- Authenticates with ECR
- Pushes image with versioning

5. Deployment

- Updates ECS service
- Performs health checks

Jenkins Pipeline Overview

This pipeline is designed to build, test, scan, and deploy a Docker-based application to AWS ECS. It leverages Docker containers for build isolation, performs security scanning with Trivy, and pushes the Docker image to AWS ECR. Finally, it updates the ECS service and performs a health check on the deployed service.

Stages Breakdown

Stage 1: Build and Test

- Runs on an ec2-build-node with a custom workspace.
- Steps:
    1. Clean workspace.
    2. Checkout code from the SCM (e.g., Git).
    3. Run the build in a Docker container (node:18-alpine) with limited memory and CPU.
    4. Install dependencies (npm ci), run tests, lint, and build the app.

5. Archive the build artifacts (dist.tar.gz).

Stage 2: Build Docker Image

- Builds the Docker image with --no-cache to ensure a fresh build.
- Runs a basic check (nginx -t) inside the container to validate the image.

Stage 3: Trivy Security Scan

- Runs a Trivy container to scan for vulnerabilities in the Docker image.
- Scans for HIGH and CRITICAL severity vulnerabilities and exits with a non-zero code on failure.

Stage 4: Push to ECR

- Logs in to AWS ECR and pushes the Docker image with retry logic.
- If the branch is main, tags the image as latest and pushes it.

Stage 5: Deploy to ECS

- Updates the ALB health check settings.
- Fetches the current ECS task definition and modifies the image reference and port mapping using jq.
- Registers a new task definition and forces a new deployment in the ECS service.
- Waits for the service to stabilize.

Stage 6: Health Check

- Retrieves the ALB DNS name.
- Performs periodic health checks by sending requests to the service endpoint.

---

3. Post Actions

- **Success**: Prints a success message.
- **Failure**: Prints a failure message.

- **Always**: Cleans up Docker images, removes unused containers, and clears the workspace.

---

4. Error Handling and Retry Logic

- Handles exceptions for each stage and sets the build status to FAILURE.
- Retries critical steps like ECR login and image push to handle transient errors.

---

5. AWS CLI Commands Used

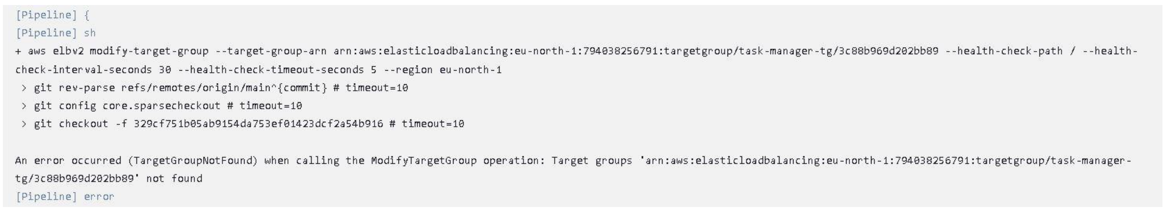
- aws elbv2 modify-target-group: Updates ALB health check.
- aws ecs describe-task-definition: Fetches current task definition.
- aws ecs register-task-definition: Registers a new task definition.
- aws ecs update-service: Deploys the new task definition to ECS.
- aws ecs wait services-stable: Waits for the service to stabilize.

---

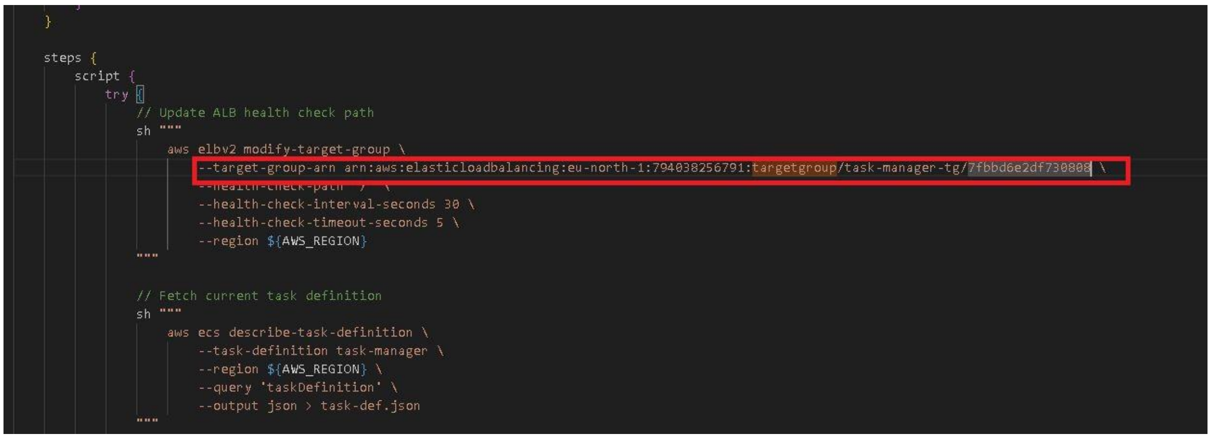
6. Docker and Trivy Integration

- Docker is used for build isolation and running tests.
- Trivy is used to scan Docker images for security vulnerabilities.

Before proceeding with the build update the “Jenkinsfile” with the Target group ARN as shown below.

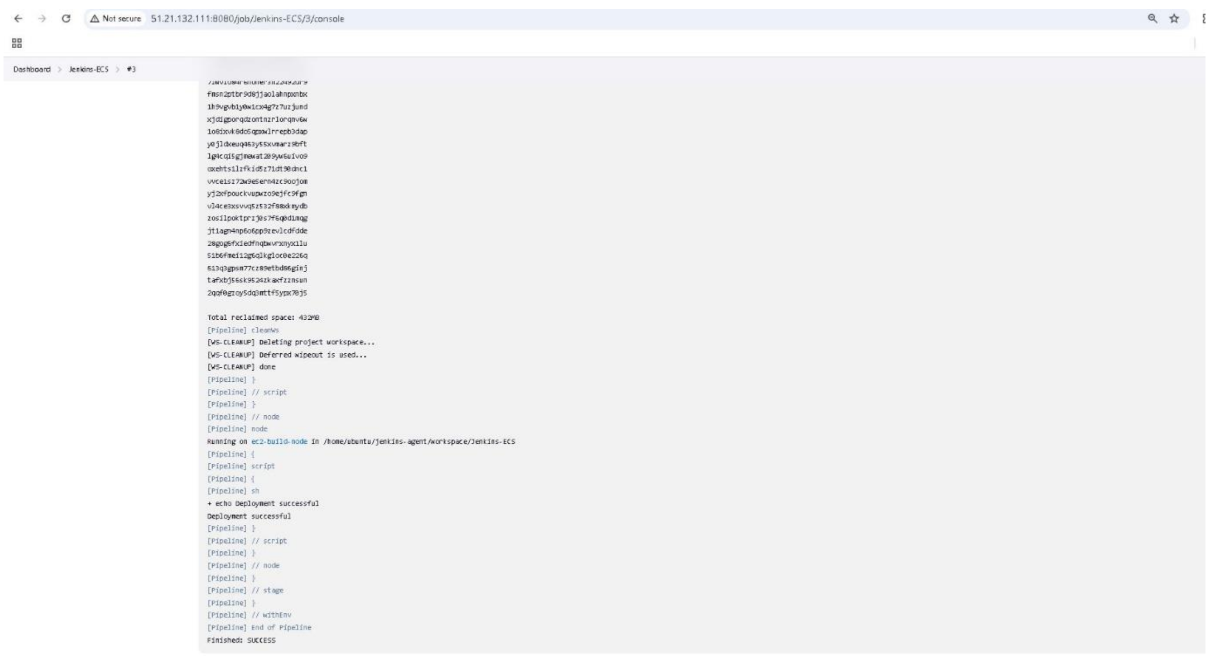


```
[Pipeline] {
[Pipeline] sh
+ aws elbv2 modify-target-group --target-group-arn arn:aws:elasticloadbalancing:eu-north-1:794038256791:targetgroup/task-manager-tg/3c88b969d202bb89 --health-check-path / --health-
check-interval-seconds 30 --health-check-timeout-seconds 5 --region eu-north-1
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 329cf751b05ab9154da753ef01423dcf2a54b916 # timeout=10

An error occurred (TargetGroupNotFound) when calling the ModifyTargetGroup operation: Target groups 'arn:aws:elasticloadbalancing:eu-north-1:794038256791:targetgroup/task-manager-
tg/3c88b969d202bb89' not found
[Pipeline] error
```



```
    }

    steps {
        script {
            try {
                // Update ALB health check path
                sh """
                    aws elbv2 modify-target-group \
                        --target-group-arn arn:aws:elasticloadbalancing:eu-north-1:794038256791:targetgroup/task-manager-tg/7fbbd6e2df730808 \
                        --health-check-path "/" \
                        --health-check-interval-seconds 30 \
                        --health-check-timeout-seconds 5 \
                        --region ${AWS_REGION}
                """

                // Fetch current task definition
                sh """
                    aws ecs describe-task-definition \
                        --task-definition task-manager \
                        --region ${AWS_REGION} \
                        --query 'taskDefinition' \
                        --output json > task-def.json
                """
```

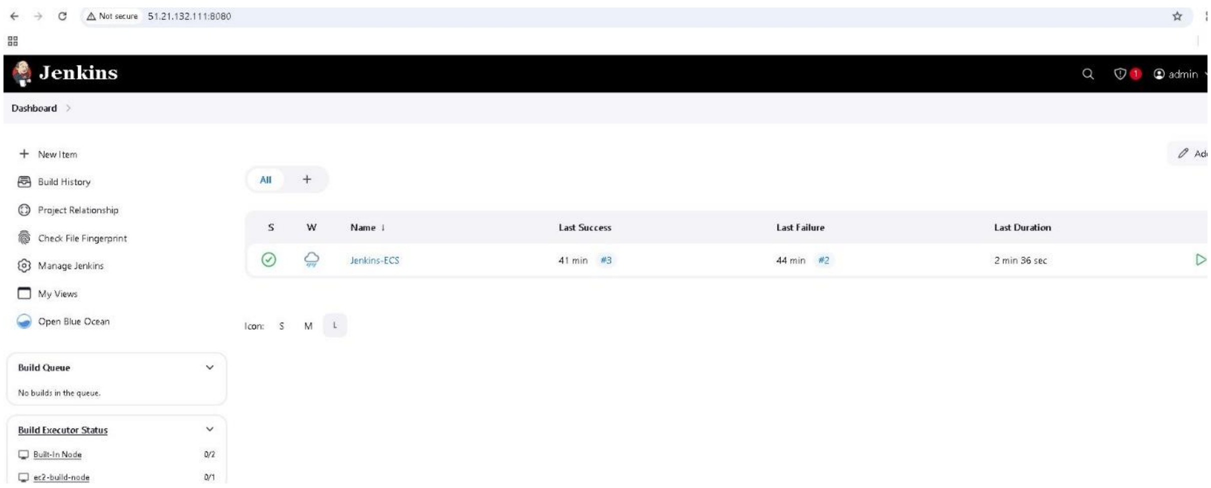
Once the build is complete the Jenkins build console will show Success message.



Not secure 51.21.132.111:8080/job/Jenkins-ECS/3/console

Dashboard > Jenkins-ECS > #3

```
fmsn2ptbr9d8jjaolahnpxnbx
1h9vgvb1y0w1cx4g7z7uzjund
xjdigporqdzontnzrlorqnv6w
1o8dxvk8do5qgowlrrepb3dap
y0jldxeuq463y55xvmarz9bft
1g4cqi5gjmawat209yw6uivo9
oxehts1lzfkid5z71dt90dnc1
vvce1sz72w9e5ern4zc9oojom
yj2xfpouckvupwzo9ejfc9fgn
vl4ce3xsvvq5z532f88xkmydb
zosilpoktprzj0s7f6q0dimqg
jt1agn4np6o6pp9zevlcdfdde
28gog6fxiedfnqbwvrxnyx1lu
51b6fmei12g6qlkgloc0e226q
613q3gpsm77cz89etbd86ginj
tafxbj56sk9524zkaxfzznsun
2qqf0gzoy5dq3mttf5ypx70j5

Total reclaimed space: 432MB
[Pipeline] cleanWs
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // node
[Pipeline] node
Running on ec2-build-node in /home/ubuntu/jenkins-agent/workspace/Jenkins-ECS
[Pipeline] {
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ echo Deployment successful
Deployment successful
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // node
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Not secure 51.21.132.111:8080

Jenkins

Dashboard >

- New Item
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Open Blue Ocean

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		Jenkins-ECS	41 min #3	44 min #2	2 min 36 sec

Icon: S M L

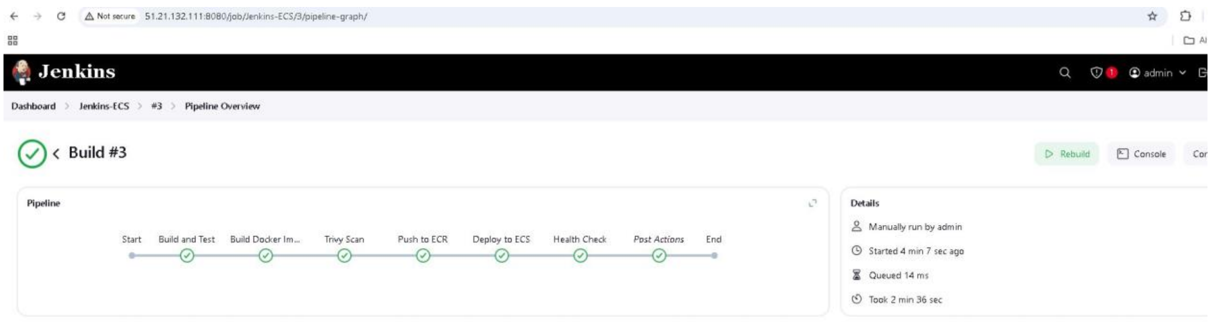
Build Queue

No builds in the queue.

Build Executor Status

Built-In Node	0/2
ec2-build-node	0/1

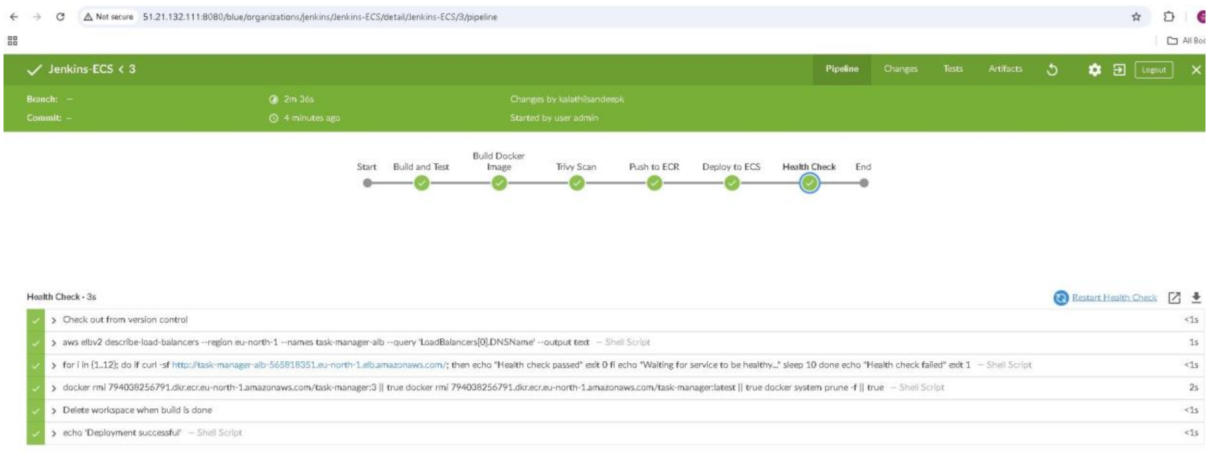
The image shows various steps used in the build and its status.


Build #3

Pipeline: Start → Build and Test → Build Docker Im... → Trivy Scan → Push to ECR → Deploy to ECS → Health Check → Post Actions → End

Details
- Manually run by admin
- Started 4 min 7 sec ago
- Queued 14 ms
- Took 2 min 36 sec

The below image is from Blue Ocean plugin interface. This also shows the Status in the pipeline.


Jenkins-ECS < 3

Branch: —   Commit: —   2m 36s   4 minutes ago   Changes by kalathilsandeepk   Started by user admin

Health Check - 3s

```
> Check out from version control                                                                 <1s
> aws elbv2 describe-load-balancers --region eu-north-1 --names task-manager-alb --query 'LoadBalancers[0].DNSName' --output text  — Shell Script   1s
> for i in {1..12}; do if curl -sf http://task-manager-alb-565818351.eu-north-1.elb.amazonaws.com/; then echo "Health check passed" exit 0 fi echo "Waiting for service to be healthy..." sleep 10 done echo "Health check failed" exit 1  — Shell Script   <1s
> docker rmi 794038256791.dkr.ecr.eu-north-1.amazonaws.com/task-manager:3 || true docker rmi 794038256791.dkr.ecr.eu-north-1.amazonaws.com/task-manager:latest || true docker system prune -f || true  — Shell Script   2s
> Delete workspace when build is done                                                            <1s
> echo 'Deployment successful'  — Shell Script                                                   <1s
```

Verify that the newly built Docker images are present in ECR.

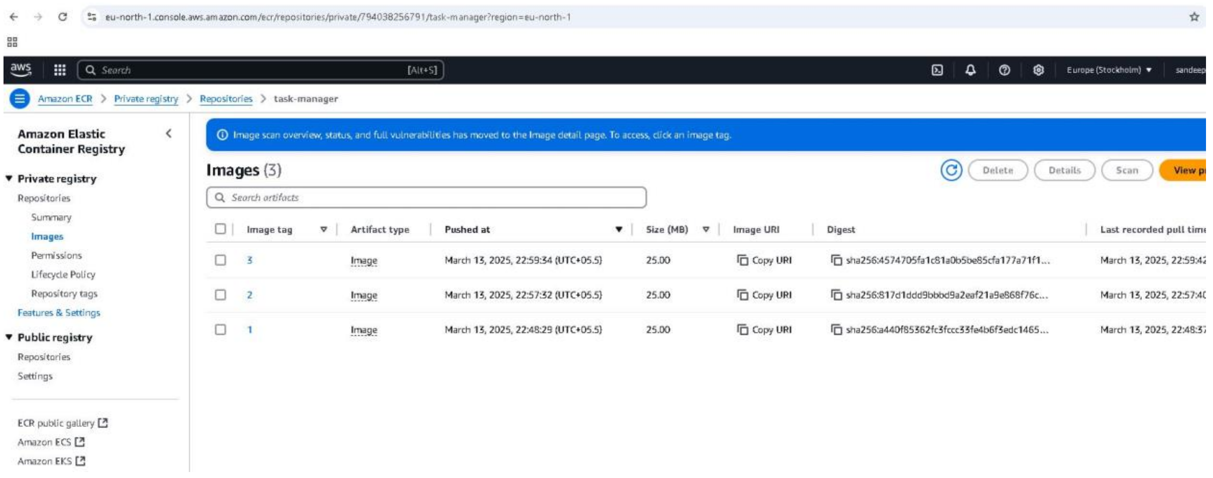
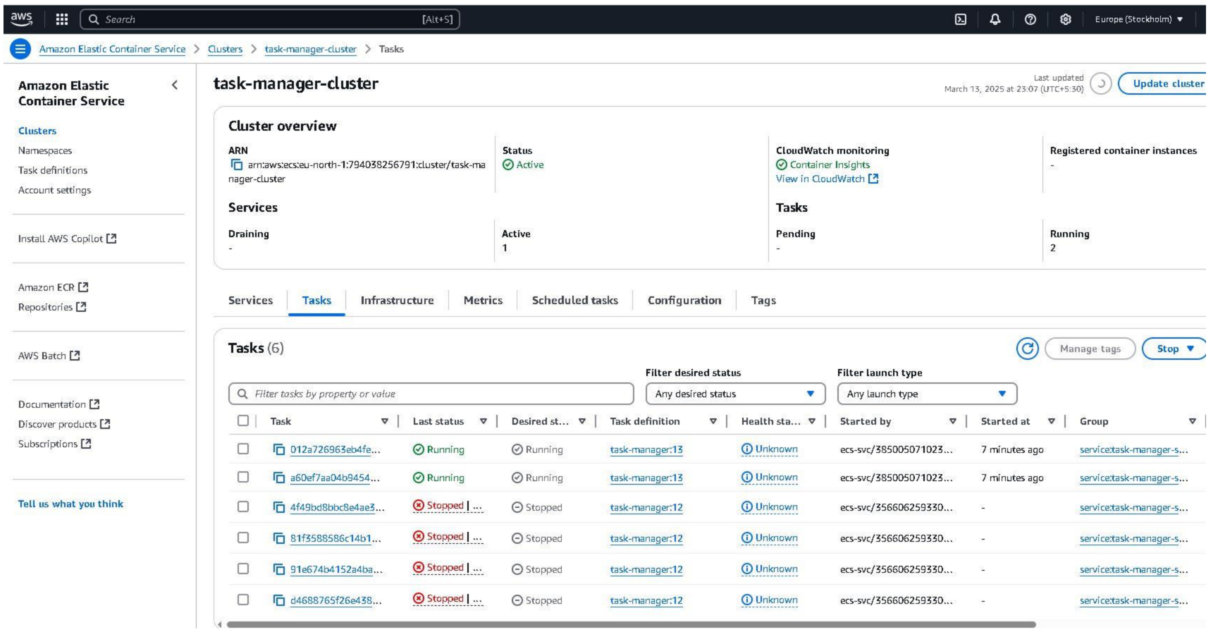


Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.

Images (3)

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
3	Image	March 13, 2025, 22:59:34 (UTC+05.5)	25.00	Copy URI	sha256:4574705fa1c81a0b5be85cfa177a71f1...	March 13, 2025, 22:59:4
2	Image	March 13, 2025, 22:57:32 (UTC+05.5)	25.00	Copy URI	sha256:817d1ddd9bbbd9a2eaf21a9e868f76c...	March 13, 2025, 22:57:4
1	Image	March 13, 2025, 22:48:29 (UTC+05.5)	25.00	Copy URI	sha256:a440f85362fc3fccc33fe4b6f3edc1465...	March 13, 2025, 22:48:3

Check the ECS console to ensure tasks are running and services are active.



Amazon Elastic Container Service > Clusters > task-manager-cluster > Tasks

task-manager-cluster

Last updated March 13, 2025 at 23:07 (UTC+5:30)

Cluster overview

ARN	Status	CloudWatch monitoring	Registered container instances
arn:aws:ecs:eu-north-1:794038256791:cluster/task-manager-cluster	Active	Container Insights View in CloudWatch	-

Services

Draining	Active
-	1

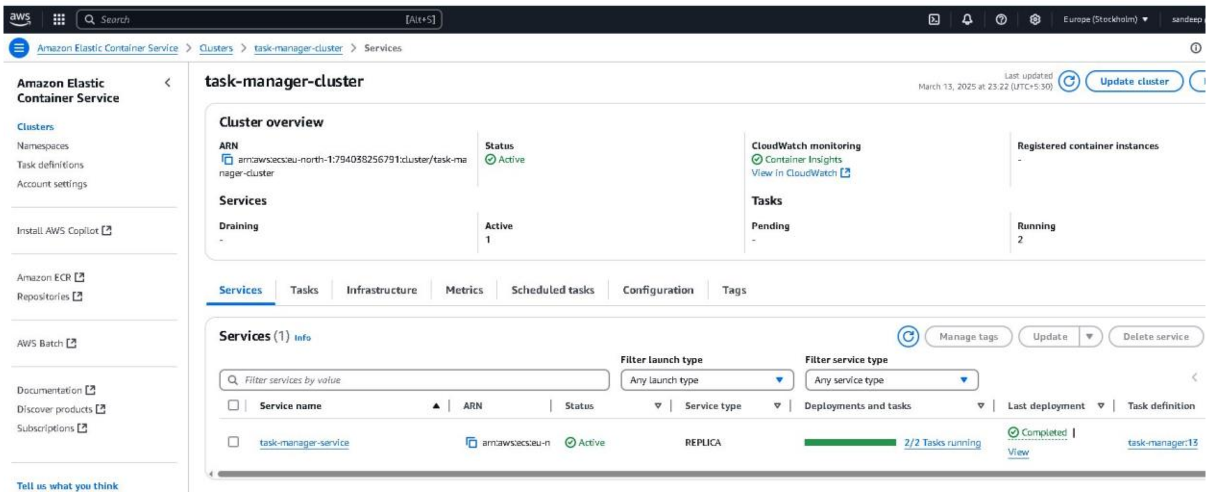
Tasks

Pending	Running
-	2

Tasks (6)

Filter desired status: Any desired status
Filter launch type: Any launch type

Task	Last status	Desired st...	Task definition	Health sta...	Started by	Started at	Group
012a726963eb4fe...	Running	Running	task-manager:13	Unknown	ecs-svc/385005071023...	7 minutes ago	service:task-manager-s...
a60ef7aa04b9454...	Running	Running	task-manager:13	Unknown	ecs-svc/385005071023...	7 minutes ago	service:task-manager-s...
4f49bd8bbc8e4ae3...	Stopped | ...	Stopped	task-manager:12	Unknown	ecs-svc/356606259330...	-	service:task-manager-s...
81f3588586c14b1...	Stopped | ...	Stopped	task-manager:12	Unknown	ecs-svc/356606259330...	-	service:task-manager-s...
91e674b4152a4ba...	Stopped | ...	Stopped	task-manager:12	Unknown	ecs-svc/356606259330...	-	service:task-manager-s...
d4688765f26e438...	Stopped | ...	Stopped	task-manager:12	Unknown	ecs-svc/356606259330...	-	service:task-manager-s...



Amazon Elastic Container Service > Clusters > task-manager-cluster > Services

task-manager-cluster

Last updated March 13, 2025 at 23:22 (UTC+5:30)

Cluster overview

ARN	Status	CloudWatch monitoring	Registered container instances
arn:aws:ecs:eu-north-1:794038256791:cluster/task-manager-cluster	Active	Container Insights View in CloudWatch	-

Services

Draining	Active
-	1

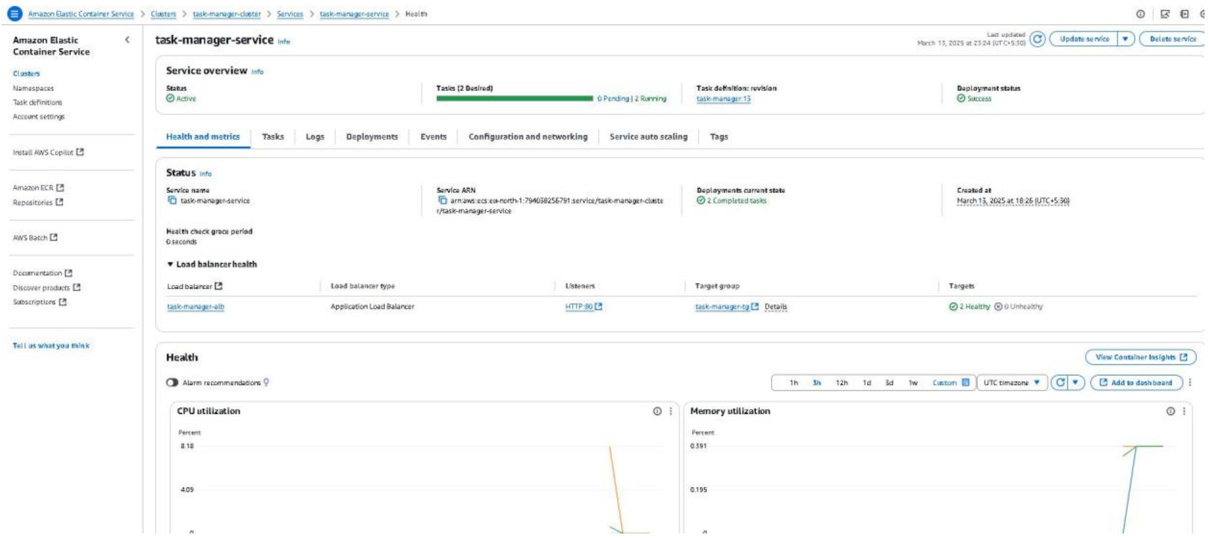
Tasks

Pending	Running
-	2

Services (1)

Filter launch type: Any launch type
Filter service type: Any service type

Service name	ARN	Status	Service type	Deployments and tasks	Last deployment	Task definition
task-manager-service	arn:aws:ecs:eu-n	Active	REPLICA	2/2 Tasks running	Completed | View	task-manager:13



Amazon Elastic Container Service > Clusters > task-manager-cluster > Services > task-manager-service > Health

Amazon Elastic Container Service

Clusters
Namespaces
Task definitions
Account settings

Install AWS Copilot
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think

task-manager-service Info

Last updated March 13, 2025 at 23:24 (UTC+5:30) | Update service | Delete service

Service overview Info

Status	Tasks (2 Desired)	Task definition: revision	Deployment status
Active	0 Pending | 2 Running	task-manager:13	Success

Health and metrics | Tasks | Logs | Deployments | Events | Configuration and networking | Service auto scaling | Tags

Status Info

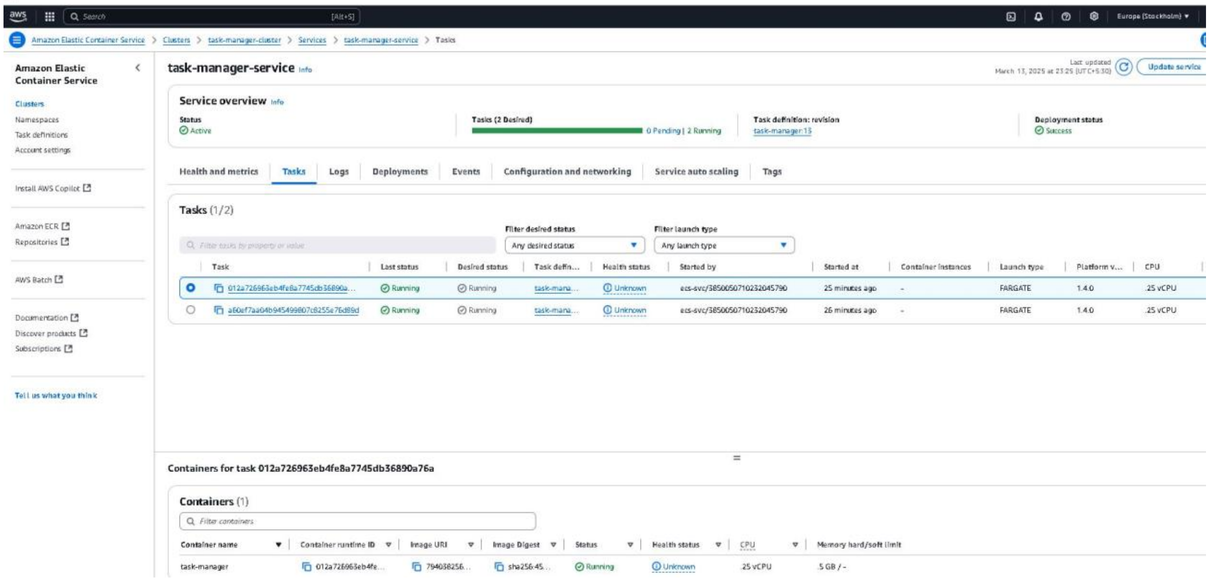
Service name	Service ARN	Deployments current state	Created at
task-manager-service	arn:aws:ecs:eu-north-1:794038256791:service/task-manager-cluster/task-manager-service	2 Completed tasks	March 13, 2025 at 18:26 (UTC+5:30)

Health check grace period
0 seconds

▼ Load balancer health

Load balancer	Load balancer type	Listeners	Target group	Targets
task-manager-alb	Application Load Balancer	HTTP:80	task-manager-tg Details	2 Healthy 0 Unhealthy

Health



Search [Alt+S] Europe (Stockholm)

Amazon Elastic Container Service > Clusters > task-manager-cluster > Services > task-manager-service > Tasks

task-manager-service Info

Last updated March 13, 2025 at 23:25 (UTC+5:30) | Update service

Service overview Info

Status	Tasks (2 Desired)	Task definition: revision	Deployment status
Active	0 Pending | 2 Running	task-manager:13	Success

Health and metrics | Tasks | Logs | Deployments | Events | Configuration and networking | Service auto scaling | Tags

Tasks (1/2)

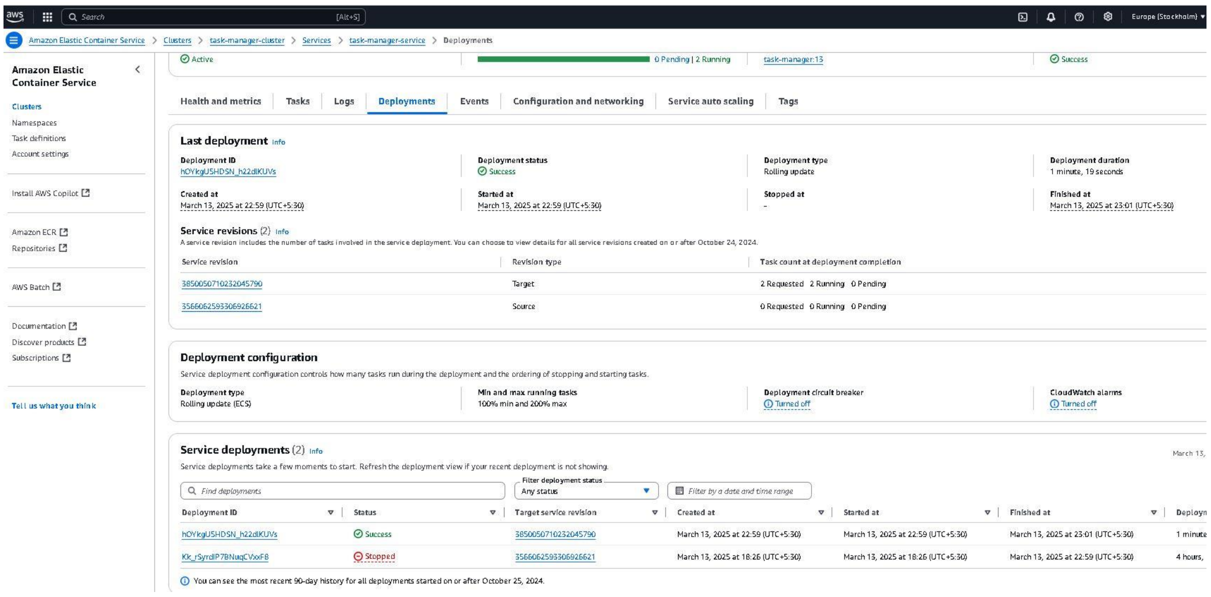
Filter desired status: Any desired status
Filter launch type: Any launch type

	Task	Last status	Desired status	Task defin...	Health status	Started by	Started at	Container instances	Launch type	Platform v...	CPU
●	012a726963eb4fe8a7745db36890a...	Running	Running	task-mana...	Unknown	ecs-svc/3850050710232045790	25 minutes ago	-	FARGATE	1.4.0	.25 vCPU
○	a60ef7aa04b945499807c8255e76d89d	Running	Running	task-mana...	Unknown	ecs-svc/3850050710232045790	26 minutes ago	-	FARGATE	1.4.0	.25 vCPU

Containers for task 012a726963eb4fe8a7745db36890a76a

Containers (1)

Container name	Container runtime ID	Image URI	Image Digest	Status	Health status	CPU	Memory hard/soft limit
task-manager	012a726963eb4fe...	794038256...	sha256:45...	Running	Unknown	.25 vCPU	.5 GB / -



Search [Alt+S] Europe (Stockholm)

Amazon Elastic Container Service > Clusters > task-manager-cluster > Services > task-manager-service > Deployments

Active | 0 Pending | 2 Running | task-manager:13 | Success

Health and metrics | Tasks | Logs | Deployments | Events | Configuration and networking | Service auto scaling | Tags

Last deployment Info

Deployment ID	Deployment status	Deployment type	Deployment duration
hOYkgU5HDSN_h22dlKUVs	Success	Rolling update	1 minute, 19 seconds

Created at	Started at	Stopped at	Finished at
March 13, 2025 at 22:59 (UTC+5:30)	March 13, 2025 at 22:59 (UTC+5:30)	-	March 13, 2025 at 23:01 (UTC+5:30)

Service revisions (2) Info

A service revision includes the number of tasks involved in the service deployment. You can choose to view details for all service revisions created on or after October 24, 2024.

Service revision	Revision type	Task count at deployment completion
3850050710232045790	Target	2 Requested 2 Running 0 Pending
3566062593306926621	Source	0 Requested 0 Running 0 Pending

Deployment configuration

Service deployment configuration controls how many tasks run during the deployment and the ordering of stopping and starting tasks.

Deployment type	Min and max running tasks	Deployment circuit breaker	CloudWatch alarms
Rolling update (ECS)	100% min and 200% max	Turned off	Turned off

Service deployments (2) Info

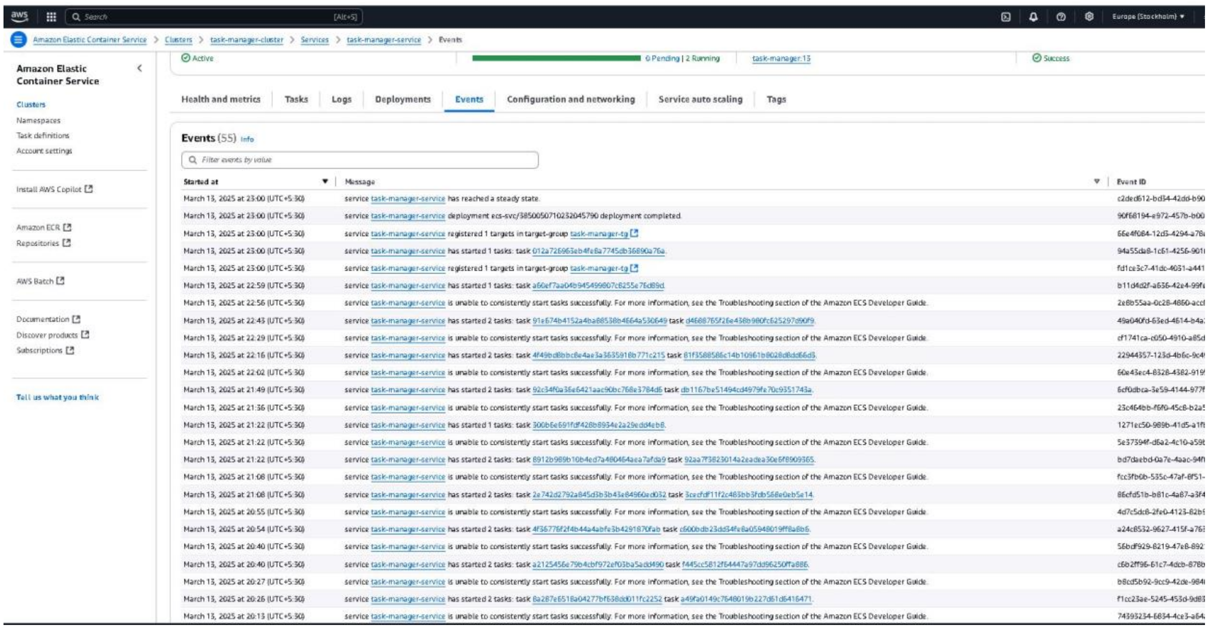
March 13,

Service deployments take a few moments to start. Refresh the deployment view if your recent deployment is not showing.

Find deployments | Filter deployment status: Any status | Filter by a date and time range

Deployment ID	Status	Target service revision	Created at	Started at	Finished at	Deploym
hOYkgU5HDSN_h22dlKUVs	Success	3850050710232045790	March 13, 2025 at 22:59 (UTC+5:30)	March 13, 2025 at 22:59 (UTC+5:30)	March 13, 2025 at 23:01 (UTC+5:30)	1 minute
Kk_rSyrdIP7BNuqCVxF8	Stopped	3566062593306926621	March 13, 2025 at 18:26 (UTC+5:30)	March 13, 2025 at 18:26 (UTC+5:30)	March 13, 2025 at 22:59 (UTC+5:30)	4 hours,

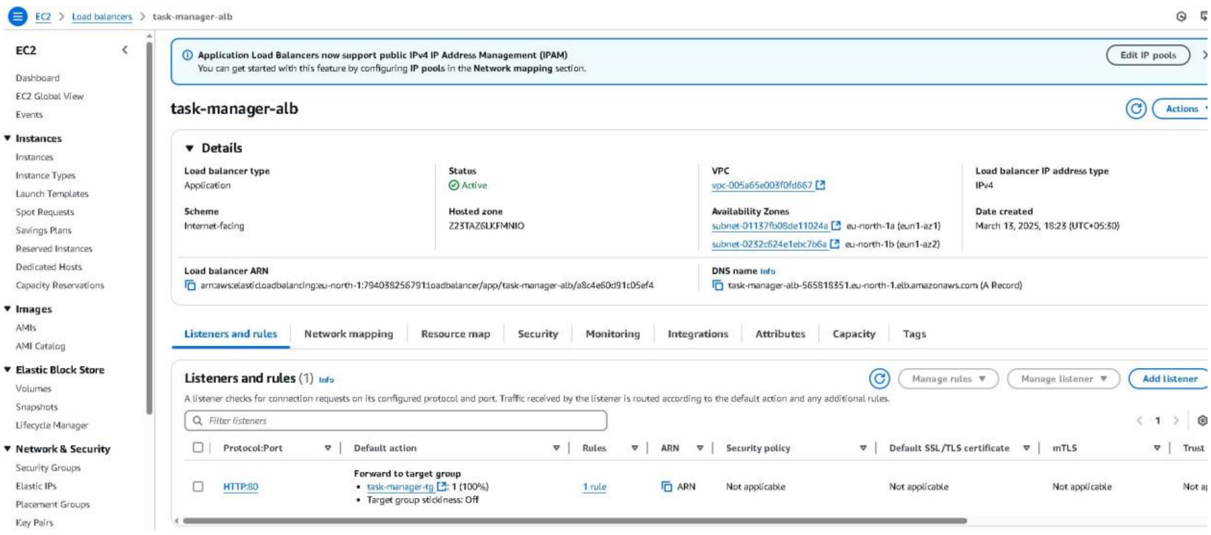
You can see the most recent 90-day history for all deployments started on or after October 25, 2024.



Events (55) Info

Started at	Message	Event ID
March 13, 2025 at 23:00 (UTC+5:30)	service task-manager-service has reached a steady state.	c2ded612-bd34-42dd-b90
March 13, 2025 at 23:00 (UTC+5:30)	service task-manager-service deployment ecs-svc/3850050710232045790 deployment completed.	90f68194-e972-457b-b00
March 13, 2025 at 23:00 (UTC+5:30)	service task-manager-service registered 1 targets in target-group task-manager-tg	66e4f084-12d3-4294-a78
March 13, 2025 at 23:00 (UTC+5:30)	service task-manager-service has started 1 tasks: task 012a726963eb4fe8a7745db36890a76a.	94a55da8-1c61-4256-901
March 13, 2025 at 23:00 (UTC+5:30)	service task-manager-service registered 1 targets in target-group task-manager-tg	fd1ce3c7-41dc-4031-a441
March 13, 2025 at 22:59 (UTC+5:30)	service task-manager-service has started 1 tasks: task a60ef7aa04b945499807c8255e76d89d.	b11d4d2f-a636-42e4-99f
March 13, 2025 at 22:56 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	2e8b55aa-0c28-4860-acc
March 13, 2025 at 22:43 (UTC+5:30)	service task-manager-service has started 2 tasks: task 91e674b4152a4ba88538b4664a530649 task d4688765f26e438b980fc625297d90f9.	49a040fd-63ed-4614-b4a
March 13, 2025 at 22:29 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	cf1741ca-c050-4910-a85d
March 13, 2025 at 22:16 (UTC+5:30)	service task-manager-service has started 2 tasks: task 4f49bd8bbc8e4ae3a3635918b771c215 task 81f3588586c14b10961b8028d8dd66d8.	22944357-123d-4b6c-9c4
March 13, 2025 at 22:02 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	60e43ec4-8328-4382-919
March 13, 2025 at 21:49 (UTC+5:30)	service task-manager-service has started 2 tasks: task 92c34f0a36e6421aac90bc768e3784d6 task db1167be51494cd4979fe70c9351743a.	6cf0dbca-3e59-4144-977f
March 13, 2025 at 21:36 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	23c464bb-f6f0-45c8-b2a5
March 13, 2025 at 21:22 (UTC+5:30)	service task-manager-service has started 1 tasks: task 300b6e691fdf428b8934e2a29edd4eb8.	1271ec50-989b-41d5-a1f
March 13, 2025 at 21:22 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	5e37394f-d6a2-4c10-a59
March 13, 2025 at 21:22 (UTC+5:30)	service task-manager-service has started 2 tasks: task 8912b989b10b4ed7a480464aea7afda9 task 92aa7f3823014a2eadea30e6f8909365.	bd7daebd-0a7e-4aac-94f
March 13, 2025 at 21:08 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	fcc3fb0b-535c-47af-8f51-
March 13, 2025 at 21:08 (UTC+5:30)	service task-manager-service has started 2 tasks: task 2e742d2792a845d3b3b43e84960ed032 task 3cecfdf11f2c483bb3fdb568e0eb5e14.	86cfd51b-b81c-4a87-a3f4
March 13, 2025 at 20:55 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	4d7c5dc8-2fe0-4123-82b
March 13, 2025 at 20:54 (UTC+5:30)	service task-manager-service has started 2 tasks: task 4f36776f2f4b44a4abfe3b4291870fab task c600bdb23dd34fe8a05948019ff8a8b6.	a24c8532-9627-415f-a76
March 13, 2025 at 20:40 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	56bdf929-8219-47e8-892
March 13, 2025 at 20:40 (UTC+5:30)	service task-manager-service has started 2 tasks: task a2125456e79b4cbf972ef03ba5add490 task f445cc5812f64447a97dd96250ffa886.	c6b2ff96-61c7-4dcb-878b
March 13, 2025 at 20:27 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	b8cd5b92-9cc9-42de-984
March 13, 2025 at 20:26 (UTC+5:30)	service task-manager-service has started 2 tasks: task 8a287e6518a04277bf638dd011fc2252 task a49fa0149c7648019b227d61d6416471.	f1cc23ae-5245-453d-9d8
March 13, 2025 at 20:13 (UTC+5:30)	service task-manager-service is unable to consistently start tasks successfully. For more information, see the Troubleshooting section of the Amazon ECS Developer Guide.	74393234-6834-4ce3-a64

Use the ALB DNS name to access the deployed website in the browser.



Application Load Balancers now support public IPv4 IP Address Management (IPAM)
You can get started with this feature by configuring **IP pools** in the **Network mapping** section.

task-manager-alb

Details

Load balancer type Application	**Status** Active	**VPC** vpc-005a65e003f0fd667	**Load balancer IP address type** IPv4
Scheme Internet-facing	**Hosted zone** Z23TAZ6LKFMNIO	**Availability Zones** subnet-01137fb08de11024a eu-north-1a (eun1-az1); subnet-0232c624e1ebc7b6a eu-north-1b (eun1-az2)	**Date created** March 13, 2025, 18:23 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:eu-north-1:794038256791:loadbalancer/app/task-manager-alb/a8c4e60d91c05ef4

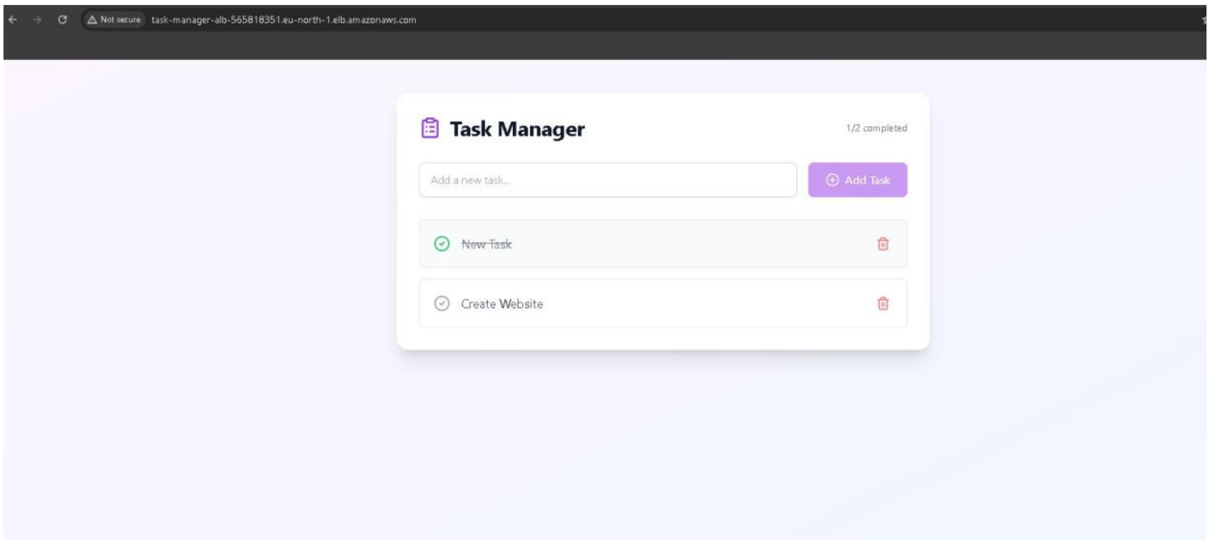
DNS name Info
task-manager-alb-565818351.eu-north-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust
HTTP:80	Forward to target group • task-manager-tg: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable	Not a

Confirm that the website loads successfully and functions as expected.


Not secure task-manager-alb-565818351.eu-north-1.elb.amazonaws.com

Task Manager

1/2 completed

~~New Task~~

Create Website