

Task 1

Setup infrastructure(infra) using Terraform (tf)

Sub task -1

1. Ensure AWS CLI is installed and configured with full access in your Linux machine.

Below steps are used to install AWS CLI on ec2 instance

Installing AWS CLI:

```
ubuntu@ip-172-31-23-243:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1019 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
```

```
ubuntu@ip-172-31-23-243:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received % Xferd  Average Speed   Time     Time   Current
          Dload  Upload   Total Spent    Left  Speed
100  63.5M  100  63.5M    0      0  343M      0 --:--:-- --:--:--:--:-- 343M
```

```
ubuntu@ip-172-31-23-243:~$ unzip awscliv2.zip
Archive: awscliv2.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/install
  inflating: aws/README.md
  inflating: aws/THIRD_PARTY_LICENSES
  creating: aws/dist/awscli/
```

```
ubuntu@ip-172-31-23-243:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
ubuntu@ip-172-31-23-243:~$ aws --version
aws-cli/2.27.56 Python/3.13.4 Linux/6.8.0-1029-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-23-243:~$ |
```

2. Initialize a bucket in s3 for the backend state store using Terraform. This bucket will be used later in this project to use the state files of Terraform.

Amazon S3 bucket is created in aws:

The screenshot shows the AWS S3 Buckets page. At the top, a green banner displays a success message: "Successfully created bucket 'shivu-upgrad-s3-bucket' To upload files and folders, or to configure additional bucket settings, choose View details." Below the banner, the heading "General purpose buckets (1) Info" is visible. A toolbar with buttons for "Copy ARN", "Empty", "Delete", and "Create bucket" is present. A sub-header states: "Buckets are containers for data stored in S3." A search bar with placeholder text "Find buckets by name" and a pagination indicator "< 1 >" are also shown. A table lists the single bucket:

Name	AWS Region	Creation date
shivu-upgrad-s3-bucket	US East (N. Virginia) us-east-1	July 22, 2025, 18:47:55 (UTC+05:30)

Sub task – 2

Create the following:

- AWS VPC,
- 1 IGW,
- 1 NAT-GW in AZ-a,
- Allow 0.0.0.0/0
- Subnets (2 private, 1 each in AZ-a &b),
- Route Tables for both subnets

Choose a /16 CIDR for VPC and /24 CIDRs for other subnets

Installing Terraform on ec2 machine:

Updating the packages

```
ubuntu@ip-172-31-23-243:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
```

Upgrading the packages

```
ubuntu@ip-172-31-23-243:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Installing the required packages

```
ubuntu@ip-172-31-23-243:~$ sudo apt install gnupg software-properties-common -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.4.4-2ubuntu17.3).
```

Download the HashiCorp GPG key for Terraform

```
ubuntu@ip-172-31-23-243:~$ wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
--2025-07-22 13:23:48-- https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 18.160.10.45, 18.160.10.69, 18.160.10.126, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|18.160.10.45|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9K) [binary/octet-stream]
```

Add the HashiCorp APT Repository

```
ubuntu@ip-172-31-23-243:~$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main"
tc/apt/sources.list.d/hashicorp.list
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
ubuntu@ip-172-31-23-243:~$ |
```

Update Ubuntu Package List

```
ubuntu@ip-172-31-23-243:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 0 B in 0s (0 B/s)
```

Install Terraform via APT

```
ubuntu@ip-172-31-23-243:~$ sudo apt install terraform -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
```

Provider.tf file (attached it in zip file)

Provider.tf file to define provider details (aws) and region used

Terraform init

Initializing the terraform

```
ubuntu@ip-172-31-23-243:~$ terraform init
Initializing the backend...

Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.

Initializing provider plugins...
- Finding hashicorp/aws versions matching "6.4.0"...
- Installing hashicorp/aws v6.4.0...
- Installed hashicorp/aws v6.4.0 (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-172-31-23-243:~$ |
```

Terraform.tfstate file saved in S3 bucket

Below screenshot showing terraform.tfstate file saved in aws s3 bucket (backend S3 details are mentioned in provider.tf file)

The screenshot shows the AWS S3 console interface. At the top, there are tabs for Objects, Metadata, Properties, Permissions, Metrics, Management, and Access Points. The Objects tab is selected. Below the tabs, the heading 'Objects (1)' is displayed. There are several action buttons: Copy S3 URI, Copy URL, Download, Open, Delete, Create folder, and Upload. A search bar labeled 'Find objects by prefix' is present. To the right of the search bar are navigation arrows and a settings icon. A table lists the single object: 'terraform.tfstate' is a tfstate file, last modified on July 22, 2025, at 19:53:04 (UTC+05:30), with a size of 56.1 KB and a storage class of Standard.

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	terraform.tfstate	tfstate	July 22, 2025, 19:53:04 (UTC+05:30)	56.1 KB	Standard

Vpc.tf file (attached it in zip file)

Creating the vpc, 1 IGW, 1 NAT-GW, 2 sub nets and route tables using vpc.tf file through terraform

Below steps shows how it is executed

Terraform.apply

```
ubuntu@ip-172-31-23-243:~$ terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:

# aws_eip.eip will be created
+ resource "aws_eip" "eip" {
  + allocation_id      = (known after apply)
  + arn                = (known after apply)
  + association_id    = (known after apply)
  + carrier_ip         = (known after apply)
  + customer_owned_ip = (known after apply)
  + domain             = "vpc"
  + id                 = (known after apply)
  + instance            = (known after apply)
  + ipam_pool_id       = (known after apply)
  + network_border_group = (known after apply)
  + network_interface   = (known after apply)
  + private_dns         = (known after apply)
  + private_ip          = (known after apply)
  + ptr_record          = (known after apply)
  + public_dns           = (known after apply)
  + public_ip            = (known after apply)
  + public_ipv4_pool     = (known after apply)
  + region              = "us-east-1"
  + tags_all            = (known after apply)
```

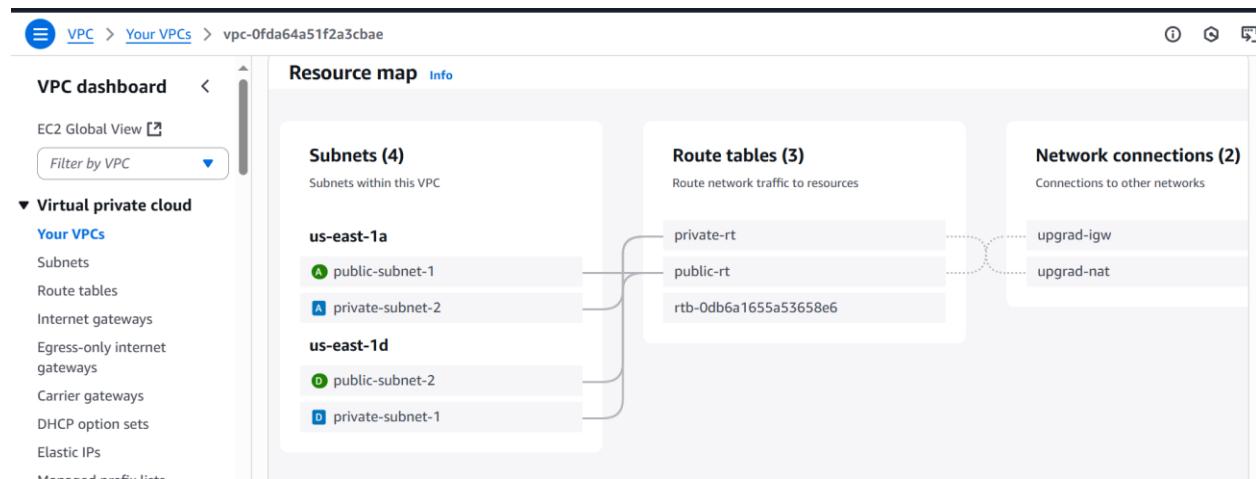
```

aws_vpc.this: Creating...
aws_eip.eip: Creating...
aws_eip.eip: Creation complete after 1s [id=eipalloc-0c7f823bbf3ab3123]
aws_vpc.this: Creation complete after 1s [id=vpc-01909e63e7aaaf9f64]
aws_internet_gateway.igw: Creating...
aws_subnet.public2: Creating...
aws_subnet.public1: Creating...
aws_subnet.private1: Creating...
aws_subnet.private2: Creating...
aws_internet_gateway.igw: Creation complete after 1s [id=igw-039541c2ae58551ed]
aws_route_table.public: Creating...
aws_subnet.public2: Creation complete after 1s [id=subnet-0f91e818a06134c68]
aws_subnet.public1: Creation complete after 1s [id=subnet-0c1e965ddd7d88ef7]
aws_nat_gateway.nat: Creating...
aws_route_table.public: Creation complete after 1s [id=rtb-0edd326d3006ed2df]
aws_route_table_association.public2: Creating...
aws_route_table_association.public1: Creating...
aws_route_table_association.public1: Creation complete after 0s [id=rtbassoc-063c339fcde3f822d]
aws_subnet.private1: Creation complete after 3s [id=subnet-0ed71cd26d5e3ab80]
aws_route_table_association.public2: Creation complete after 1s [id=rtbassoc-095c01f5df4deb4ce]
aws_subnet.private2: Creation complete after 4s [id=subnet-0b3e9412567c1d25c]
aws_nat_gateway.nat: Still creating... [00m10s elapsed]
aws_nat_gateway.nat: Still creating... [00m20s elapsed]
aws_nat_gateway.nat: Still creating... [00m30s elapsed]
aws_nat_gateway.nat: Still creating... [00m40s elapsed]
aws_nat_gateway.nat: Still creating... [00m50s elapsed]
aws_nat_gateway.nat: Still creating... [01m00s elapsed]
aws_nat_gateway.nat: Still creating... [01m10s elapsed]
aws_nat_gateway.nat: Still creating... [01m20s elapsed]
aws_nat_gateway.nat: Still creating... [01m30s elapsed]
aws_nat_gateway.nat: Still creating... [01m40s elapsed]
aws_nat_gateway.nat: Still creating... [01m50s elapsed]
aws_nat_gateway.nat: Still creating... [02m00s elapsed]
aws_nat_gateway.nat: Creation complete after 2m4s [id=nat-0eabfc8897c335a2a]
aws_route_table.private: Creating...
aws_route_table.private: Creation complete after 1s [id=rtb-011aa37dfcf14dff5]
aws_route_table_association.private2: Creating...
aws_route_table_association.private1: Creating...
aws_route_table_association.private2: Creation complete after 0s [id=rtbassoc-07d21464bbf53774a]
aws_route_table_association.private1: Creation complete after 0s [id=rtbassoc-0e948e9ce13902062]

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-23-243:~$ |

```

VPC Resource map structure in AWS console after creating it through terraform vpc.tf (VPC architecture diagram to show networking as mentioned in subtask bonus)



Sub task -3

Create EC2 instances (bastion, Jenkins, app) & ALB. As part of this subtask, you are required to create security groups for them.

Create Security Groups(SGs) for the following resources:

- Bastion host SG - Allow self ip to ssh to ‘bastion’ instance and allow all egress.
- Private Instances SG - Allow all incoming traffic from within VPC and all egress.
- Public Web SG - Allow incoming to port 80 from self IP and all egress.

All terraform .tf files used to create aws resources

```
ubuntu@ip-172-31-23-243:~$ ls
alb.tf  aws  awscliV2.zip  ec2.tf  key.tf  provider.tf  security_groups.tf  upgrad-key.pem  vpc.tf
ubuntu@ip-172-31-23-243:~$ |
```

Security groups, key -pair and ec2 machines were created. (attached .tf files in zip)

Creating public and private security groups using **security_groups.tf** file

Ssh key pairs are created using **key.tf** file

Creating public bastion instance, Jenkins instance, app instance using **ec2.tf** file

```
tls_private_key.ssh_key: Creating...
aws_security_group.public_sg: Creating...
aws_security_group.private_sg: Creating...
aws_route_table.private: Modifying... [id=rtb-011aa37dfcf14dff5]
aws_route_table.private: Modifications complete after 0s [id=rtb-011aa37dfcf14dff5]
tls_private_key.ssh_key: Creation complete after 1s [id=1a26e5355cc246effc794b9e670b71063bbdafda]
aws_key_pair.generated_key: Creating...
aws_key_pair.generated_key: Creation complete after 0s [id=upgrad-key]
aws_security_group.private_sg: Creation complete after 3s [id=sg-01e8200ae1b088156]
aws_instance.app: Creating...
aws_instance.jenkins: Creating...
aws_security_group.public_sg: Creation complete after 3s [id=sg-0d769bb92f4ada3bc]
aws_instance.public: Creating...
aws_instance.jenkins: Still creating... [00m10s elapsed]
aws_instance.app: Still creating... [00m10s elapsed]
aws_instance.public: Still creating... [00m10s elapsed]
aws_instance.app: Still creating... [00m20s elapsed]
aws_instance.jenkins: Still creating... [00m20s elapsed]
aws_instance.public: Still creating... [00m20s elapsed]
aws_instance.jenkins: Still creating... [00m30s elapsed]
aws_instance.app: Still creating... [00m30s elapsed]
aws_instance.public: Still creating... [00m30s elapsed]
aws_instance.public: Creation complete after 32s [id=i-0578948af8485628e]
aws_instance.jenkins: Creation complete after 32s [id=i-0094b427344c0a7c0]
aws_instance.app: Still creating... [00m40s elapsed]
aws_instance.app: Creation complete after 42s [id=i-0e43dd1645c01f07a]
```

ALB created using **alb.tf** file (file is attached in Zip)

```
aws_lb_target_group.public_tg: Creating...
aws_security_group.alb_sg: Creating...
aws_route_table.private: Modifying... [id=rtb-011aa37dfcf14dff5]
aws_route_table.private: Modifications complete after 0s [id=rtb-011aa37dfcf14dff5]
aws_lb_target_group.public_tg: Creation complete after 0s [id=arn:aws:elasticloadbalancing:us-east-1:237458753027:targetgroup/public-tg/d78f9853af93251a]
aws_security_group.public_sg: Modifications complete after 0s [id=sg-0d769bb92f4ada3bc]
aws_lb_target_group.attachment.public_instance_attachment: Creating...
aws_lb_target_group.attachment.public_instance_attachment: Creation complete after 1s [id=arn:aws:elasticloadbalancing:us-east-1:237458753027:targetgroup/public-tg/d78f9853af93251a-20250722153545639000000001]
aws_security_group.alb_sg: Creation complete after 2s [id=sg-032fab3d31d54421f]
aws_lb_app_alb: Creating...
aws_lb_app_alb: Still creating... [00m00s elapsed]
aws_lb_app_alb: Still creating... [00m20s elapsed]
aws_lb_app_alb: Still creating... [00m30s elapsed]
aws_lb_app_alb: Still creating... [00m40s elapsed]
aws_lb_app_alb: Still creating... [00m50s elapsed]
aws_lb_app_alb: Still creating... [01m00s elapsed]
aws_lb_app_alb: Still creating... [01m10s elapsed]
aws_lb_app_alb: Still creating... [01m20s elapsed]
aws_lb_app_alb: Still creating... [01m30s elapsed]
aws_lb_app_alb: Still creating... [01m40s elapsed]
aws_lb_app_alb: Still creating... [01m50s elapsed]
aws_lb_app_alb: Still creating... [02m00s elapsed]
aws_lb_app_alb: Still creating... [02m10s elapsed]
aws_lb_app_alb: Still creating... [02m20s elapsed]
aws_lb_app_alb: Still creating... [02m30s elapsed]
aws_lb_app_alb: Still creating... [02m40s elapsed]
aws_lb_app_alb: Still creating... [02m50s elapsed]
aws_lb_app_alb: Creation complete after 2m52s [id=arn:aws:elasticloadbalancing:us-east-1:237458753027:loadbalancer/app/upgrad-alb/273a9da90289cca4]
aws_lb_listener.http_listener: Creating...
aws_lb_listener.http_listener: Creation complete after 0s [id=arn:aws:elasticloadbalancing:us-east-1:237458753027:listener/app/upgrad-alb/273a9da90289cca4/af3659ecf0cbe3c9]
Apply complete! Resources: 5 added, 2 changed, 0 destroyed.
```

Public bastion instance:

Instance summary for i-0578948af8485628e (public-instance) [Info](#)

Connect	Instance state ▾	Actions ▾
Updated less than a minute ago		
Instance ID i-0578948af8485628e	Public IPv4 address 34.207.120.232 open address	Private IPv4 addresses 10.100.1.157
IPv6 address –	Instance state Running	Public DNS –
Hostname type IP name: ip-10-100-1-157.ec2.internal	Private IP DNS name (IPv4 only) ip-10-100-1-157.ec2.internal	

App instance:

Instance summary for i-0e43dd1645c01f07a (app-instance) [Info](#)

Connect	Instance state ▾	Actions ▾
Updated less than a minute ago		
Instance ID i-0e43dd1645c01f07a	Public IPv4 address –	Private IPv4 addresses 10.100.3.135
IPv6 address –	Instance state Running	Public DNS –
Hostname type IP name: ip-10-100-3-135.ec2.internal	Private IP DNS name (IPv4 only) ip-10-100-3-135.ec2.internal	

Jenkins instance

Instance summary for i-0094b427344c0a7c0 (jenkins-instance) [Info](#)

[C](#) [Connect](#) [Instance state](#) [Actions](#)

Updated less than a minute ago

Instance ID

[i-0094b427344c0a7c0](#)

Public IPv4 address

-

Private IPv4 addresses

[10.100.4.216](#)

IPv6 address

-

Instance state

[Running](#)

Public DNS

-

Hostname type

IP name: ip-10-100-4-216.ec2.internal

Private IP DNS name (IPv4 only)

[ip-10-100-4-216.ec2.internal](#)

Public security group

sg-0d769bb92f4ada3bc - public-sg

[Actions](#)

Details

Security group name
[public-sg](#)

Security group ID
[sg-0d769bb92f4ada3bc](#)

Description
[Allow SSH from specific IP](#)

VPC ID
[vpc-01909e63e7aaaf9f64](#)

Owner
[237458753027](#)

Inbound rules count
2 Permission entries

Outbound rules count
1 Permission entry

[Inbound rules](#)

[Outbound rules](#)

[Sharing - new](#)

[VPC associations - new](#)

[Tags](#)

Inbound rules (2)

[C](#)

[Manage tags](#)

[Edit inbound rules](#)

< 1 > | [⚙️](#)

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol
<input type="checkbox"/>	-	sgr-0740521a930a8003d	IPv4	SSH	TCP

Private Security group

sg-01e8200ae1b088156 - private-sg

Actions ▾

Details	
Security group name private-sg	Security group ID sg-01e8200ae1b088156
Owner 237458753027	Description Allow all traffic within VPC
	VPC ID vpc-01909e63e7aaf9f64 [2]
	Inbound rules count 1 Permission entry
	Outbound rules count 1 Permission entry

Inbound rules | **Outbound rules** | **Sharing - new** | **VPC associations - new** | **Tags**

Inbound rules (1)

Search		Manage tags	Edit inbound rules	
Name	Security group rule ID	IP version	Type	Protocol
-	sgr-0bcde244a09c28b9d	IPv4	All traffic	All

Application load balancer:

EC2 > Load balancers > upgrad-alb

upgrad-alb

Actions ▾

▼ Details	
Load balancer type Application	Status Active
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K
	VPC vpc-01909e63e7aaf9f64 [2]
	Availability Zones subnet-0f91e818a06134c68 [2] us-east-1c (use1-az6) subnet-0c1e965ddd7d88ef7 [2] us-east-1e (use1-az3)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:237458753027:loadbalancer/app/upgrad-alb/273a9da90289cca4	Date created July 22, 2025, 21:05 (UTC+05:30)
	DNS name Info upgrad-alb-276182885.us-east-1.elb.amazonaws.com (A Record)

Ec2 instances with IP details:

Public bastion server: 4.207.120.232 /10.100.1.157 (private IP)

Jenkin sever IP (Private): 10.100.4.216

App server IP (Private): 10.100.3.135

Refer above IP addresses whenever mentioned in below screenshots mentioning bastion/Jenkins/app server during operations.

Public security group is open to ssh from only my system IP only

The screenshot shows the AWS EC2 console under the 'Security Groups' section. A public security group named 'public-sg' is selected. The 'Inbound rules' tab is active, displaying five rules. One rule allows SSH (TCP port 22) from the specific IP 167.103.54.90/32. Other rules allow HTTP (TCP port 80) and HTTPS (TCP port 443) from 0.0.0.0/0.

Type	Protocol	Port range	Source
IPv4	HTTP	80	0.0.0.0/0
IPv4	SSH	22	167.103.54.90/32
IPv4	TCP	443	0.0.0.0/0
IPv4	TCP	443	0.0.0.0/0

Private security group is open for only vpc CIDR range and port 8080 for jenkins

The screenshot shows the AWS EC2 console under the 'Security Groups' section. A private security group named 'private-sg' is selected. The 'Inbound rules' tab is active, displaying two rules. One rule allows all traffic (All) from the VPC CIDR range 10.100.0.0/16. Another rule allows Custom TCP (port 8080) from the Jenkins instance's IP address, sg-032fab3d31d54421f...

Type	Protocol	Port range	Source	Description
All traffic	All	All	10.100.0.0/16	-
Custom TCP	TCP	8080	sg-032fab3d31d54421f...	-

Below are the connections check from servers:

Able to connect from my system to public bastion machine (public IP - 34.207.120.232/Private IP - 10.100.1.157):

```
ASIAAPACIFIC+nds@HPE-5CG31205CV MINGW64 ~/Downloads
$ ssh -i "upgrad-key.pem" ubuntu@34.207.120.232
The authenticity of host '34.207.120.232 (34.207.120.232)' can't be established.
ED25519 key fingerprint is SHA256:PGydPyLsHsoeSUOX34THx2QfaG2V0H10sq21kXBsYyM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.207.120.232' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue Jul 22 14:55:17 UTC 2025

System load: 0.0          Processes: 96
Usage of /: 20.5% of 7.57GB  Users logged in: 0
Memory usage: 10%          IPv4 address for eth0: 10.100.1.157
Swap usage: 0%
```

Able to connect from the bastion machine (10.100.1.157) to App server (10.100.3.135):

```
ubuntu@ip-10-100-1-157:~$ ssh -i "upgrad-key.pem" ubuntu@10.100.3.135
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Tue Jul 22 14:58:43 UTC 2025

System load: 0.0          Processes: 95
Usage of /: 20.5% of 7.57GB  Users logged in: 0
Memory usage: 10%          IPv4 address for eth0: 10.100.3.135
Swap usage: 0%
```

Able to connect from the bastion machine (10.100.1.157) to Jenkins server (10.100.4.216)

```
ubuntu@ip-10-100-1-157:~$ ssh -i "upgrad-key.pem" ubuntu@10.100.4.216
The authenticity of host '10.100.4.216 (10.100.4.216)' can't be established.
ED25519 key fingerprint is SHA256:LsRqfszspnq0HZRZ4FSdvgWGLINHCTfbT94HsizG2rE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.100.4.216' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Tue Jul 22 15:00:54 UTC 2025

System load: 0.0          Processes:      96
Usage of /:   20.5% of 7.57GB  Users logged in:  0
Memory usage: 10%          IPv4 address for eth0: 10.100.4.216
Swap usage:   0%
```

List of all .tf files

```
ubuntu@ip-172-31-23-243:~$ ls -l
total 65088
-rw-rw-r-- 1 ubuntu ubuntu    1841 Jul 22 15:41 alb.tf
drwxr-xr-x 3 ubuntu ubuntu    4096 Jul 21 18:11 aws
-rw-rw-r-- 1 ubuntu ubuntu 66617248 Jul 22 13:12 awscliv2.zip
-rw-rw-r-- 1 ubuntu ubuntu    1104 Jul 22 15:18 ec2.tf
-rw-rw-r-- 1 ubuntu ubuntu     315 Jul 22 15:18 key.tf
-rw-rw-r-- 1 ubuntu ubuntu     351 Jul 22 15:14 provider.tf
-rw-rw-r-- 1 ubuntu ubuntu     870 Jul 22 15:17 security_groups.tf
-r----- 1 ubuntu ubuntu    3243 Jul 22 14:40 upgrad-key.pem
-rw-rw-r-- 1 ubuntu ubuntu    1984 Jul 22 15:15 vpc.tf
ubuntu@ip-172-31-23-243:~$ \|
```

Task – 2

Setup Config Management for hosts using Ansible & CI pipeline using Jenkins

Sub task -1

Setting up Config Management for hosts using Ansible

Installing Ansible on Bastion server (10.100.1.157)

```
ubuntu@ip-10-100-1-157:~$ sudo apt-add-repository ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts for your applications—automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/
```

Updating the packages:

```
Reading package lists... Done
ubuntu@ip-10-100-1-157:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:5 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
242 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Install ansible:

```
ubuntu@ip-10-100-1-157:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python3-jmespath python3-kerberos python3-nacl python3-ntlm-auth pytl
  python3-resolvelib python3-winrm python3-xmlltodict sshpass
  .
```

Inventory file:

```
ubuntu@ip-10-100-1-157:~$ cat inventory
10.100.3.135 ansible_user=ubuntu ansible_ssh_private_key_file="/home/ubuntu/upgrad-key.pem"
10.100.4.216 ansible_user=ubuntu ansible_ssh_private_key_file="/home/ubuntu/upgrad-key.pem"
ubuntu@ip-10-100-1-157:~$ |
```

Able to connect to both app server (10.100.3.135) and Jenkins server (10.100.4.216) through ansible

```
ubuntu@ip-10-100-1-157:~$ ansible all -m ping -i inventory
[WARNING]: Platform Linux on host 10.100.4.216 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
10.100.4.216 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform Linux on host 10.100.3.135 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
10.100.3.135 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.10"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-10-100-1-157:~|
```

Refer Playbook.yaml from zip file attached

Docker getting installed through ansible on app server (10.100.3.135) and Jenkin server (10.100.4.216)

```
ubuntu@ip-10-100-1-157:~$ vi playbook.yaml
ubuntu@ip-10-100-1-157:~$ ansible-playbook playbook.yaml -i inventory
PLAY [Install Docker] ****
TASK [Gathering Facts] ****
[WARNING]: Platform Linux on host 10.100.4.216 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [10.100.4.216]
[WARNING]: Platform Linux on host 10.100.3.135 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [10.100.3.135]

TASK [Install aptitude] ****
changed: [10.100.4.216]
changed: [10.100.3.135]

TASK [Install required system packages] ****
changed: [10.100.4.216]
changed: [10.100.3.135]

TASK [Add Docker GPG apt Key] ****
changed: [10.100.4.216]
changed: [10.100.3.135]

TASK [Add Docker Repository] ****
changed: [10.100.4.216]
changed: [10.100.3.135]

TASK [Update apt and install docker-ce] ****
changed: [10.100.4.216]
changed: [10.100.3.135]

TASK [Ensure Docker service is started and enabled] ****
ok: [10.100.4.216]
ok: [10.100.3.135]

PLAY RECAP ****
10.100.3.135      : ok=7    changed=5    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0
10.100.4.216      : ok=7    changed=5    unreachable=0   failed=0    skipped=0   rescued=0   ignored=0
```

Proof that docker is installed on app server (10.100.3.135)

```
ubuntu@ip-10-100-3-135:~$ docker --version
Docker version 28.1.1, build 4eba377
ubuntu@ip-10-100-3-135:~|
```

Proof that docker is installed on jenkins server (10.100.4.216)

```
ubuntu@ip-10-100-4-216:~$ docker --version
Docker version 28.1.1, build 4eba377
ubuntu@ip-10-100-4-216:~$ |
```

Connected to Jenkins server (10.100.4.216) from local without storing .pem file on bastion server using below command in screenshot

```
ASIAPACIFIC+nds@HPE-5CG31205CV MINGW64 ~/Downloads
$ ssh -o "IdentitiesOnly=yes" \
    -o "ProxyCommand=ssh -i upgrad-key.pem -o IdentitiesOnly=yes -W %h:%p ubuntu@34.207.120.232" \
    -i upgrad-key.pem \
    ubuntu@10.100.4.216
The authenticity of host '10.100.4.216 (<no hostip for proxy command>)' can't be established.
ED25519 key fingerprint is SHA256:LsRqfszspnq0HZRZ4FsdvgWGLINHCTfbT94HsizG2rE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.100.4.216' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Wed Jul 23 04:44:13 UTC 2025
```

Sub task 2 and 3

Installing Java on Jenkins server (10.100.4.216)

```
sudo refresher
ubuntu@ip-10-100-4-216:~$ sudo apt install openjdk-21-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsamixer alsound at-spi2-core ca-certificates-java dconf
  gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme java
  libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi
  libdatriel libdconf1 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libevoxv
```

Java successfully installed

```
NO VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-100-4-216:~$ java -version
openjdk version "21.0.7" 2025-04-15
OpenJDK Runtime Environment (build 21.0.7+6-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 21.0.7+6-Ubuntu-0ubuntu122.04, mixed mode, sharing)
ubuntu@ip-10-100-4-216:~$ |
```

Installing Jenkins now

```
ubuntu@ip-10-100-4-216:~$ sudo apt install jenkins -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 224 not upgraded.
Need to get 92.5 MB of archives.
After this operation, 95.2 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 net-tools amd64 1
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.504.3 [92.3 MB]
```

Jenkins successfully installed and running

```
NO VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-100-4-216:~$ sudo systemctl status jenkins
● Jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-07-23 04:49:34 UTC; 18s ago
     Main PID: 18146 (java)
        Tasks: 42 (limit: 2329)
       Memory: 368.2M
          CPU: 17.576s
         CGroup: /system.slice/jenkins.service
                   └─18146 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080
```

Application load balancer created using terraform

upgrad-alb

Actions ▾

▼ Details

Load balancer type Application	Status Active	VPC vpc-01909e63e7aaf9f64	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7X7K	Availability Zones subnet-0f91e818a06134c68 us-east-1c (use1-az6) subnet-0c1e965ddd7d88ef7 us-east-1e (use1-az3)	Date created July 22, 2025, 21:05 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:237458753027:loadbalancer/app/upgrad-alb/273a9da90289cca4	DNS name Info upgrad-alb-276182885.us-east-1.elb.amazonaws.com (A Record)		

Listening on port 80 and forwarding to target group having jenkins server

< [Listeners and rules](#) Network mapping Resource map Security Monitoring Integrations >

Listeners and rules (1) [Info](#) [Manage rules](#) [Manage listener](#) [Add listener](#)

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
<input type="checkbox"/> HTTP:80	Forward to target group <ul style="list-style-type: none">jenkins-tg: 1 (100%)Target group stickiness: Off	1 rule	ARN	Not applicable

Load balancer security group is opened to listen from port 80

sg-032fab3d31d54421f - alb-sg

Details

Security group name alb-sg	Security group ID sg-032fab3d31d54421f	Description Allow HTTP access from internet	VPC ID vpc-01909e63e7aaaf9f64 [2]
Owner 237458753027	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules | **Outbound rules** | **Sharing - new** | **VPC associations - new** | **Tags**

Inbound rules (1)

Protocol	Port range	Source	Description
TCP	80	0.0.0.0/0	Allow HTTP from all

Target group Jenkins is having instance Jenkins server, port 8080 is opened to listen from port

jenkins-tg

Details

Target type Instance	Protocol : Port HTTP: 8080	Protocol version HTTP1	VPC vpc-01909e63e7aaaf9f64 [2]
IP address type IPv4	Load balancer upgrad-alb [2]		

Registered targets (1) Info

Anomaly mitigation: Not applicable

Deregister Register targets

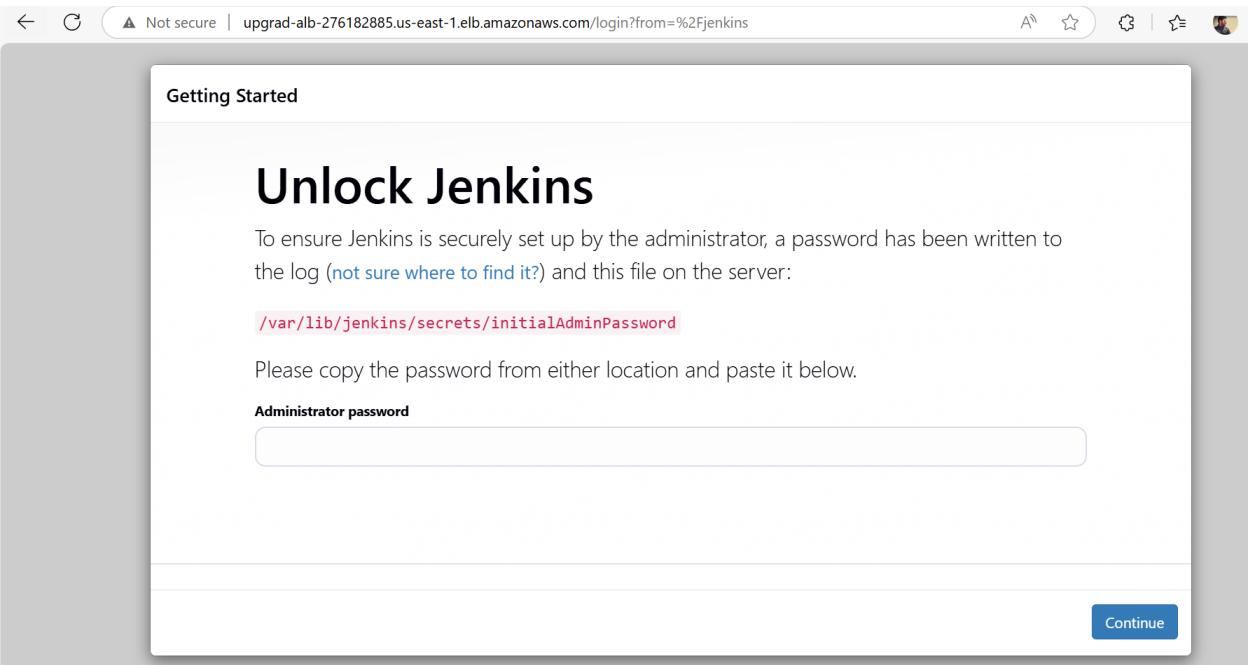
Filter targets

Instance ID	Name	Port	Zone
i-0094b427344c0a7c0	jenkins-instance	8080	us-east-1c (use1-az6)

Able to access Jenkins from web which is installed on jenkin server through load balancer dns url

DNS name [Info](#)

 [upgrad-alb-276182885.us-east-1.elb.amazonaws.com \(A Record\)](#)



The screenshot shows a browser window with the URL upgrad-alb-276182885.us-east-1.elb.amazonaws.com/login?from=%2Fjenkins. The page title is "Getting Started". The main heading is "Unlock Jenkins". A text block explains that a password has been written to the log ([not sure where to find it?](#)) and provides the file path `/var/lib/jenkins/secrets/initialAdminPassword`. Below this, instructions say to copy the password from either location and paste it into the "Administrator password" input field. A "Continue" button is visible at the bottom right.

Installing all suggested plugins

Getting Started

The screenshot shows the Jenkins 'Getting Started' page. On the left, there's a sidebar titled 'Folders' with sections for 'Timestamper', 'Pipeline', 'Git', and 'LDAP'. To the right is a table with columns for 'Folders', 'OWASP Markup Formatter', 'Build timeout', and 'Credentials Binding'. A tooltip for 'Credentials Binding' lists various Jenkins API dependencies:

- ** Plain Credentials
- ** Variant
- ** SSH Credentials
- Credentials Binding**
- ** SCM API
- ** Pipeline: API
- ** commons-lang3 v3.x Jenkins API
- Timestamper**
- ** Caffeine API
- ** Script Security
- ** JavaBeans Activation Framework (JAF) API
- ** JAXB
- ** SnakeYAML API
- ** JSON API
- ** Jackson 2 API
- ** commons-text API
- ** - required dependency

Created user in my name

Getting Started

The screenshot shows the Jenkins 'Create User' configuration page. It includes fields for 'Password' (containing '.....'), 'Confirm password' (containing '.....'), 'Full name' (containing 'Shiva kumara N D'), and 'E-mail address' (containing 'shivakumarand95@gmail.com'). At the bottom, there are links for 'Jenkins 2.504.3', 'Skip and continue as admin', and a blue 'Save and Continue' button.

Getting Started

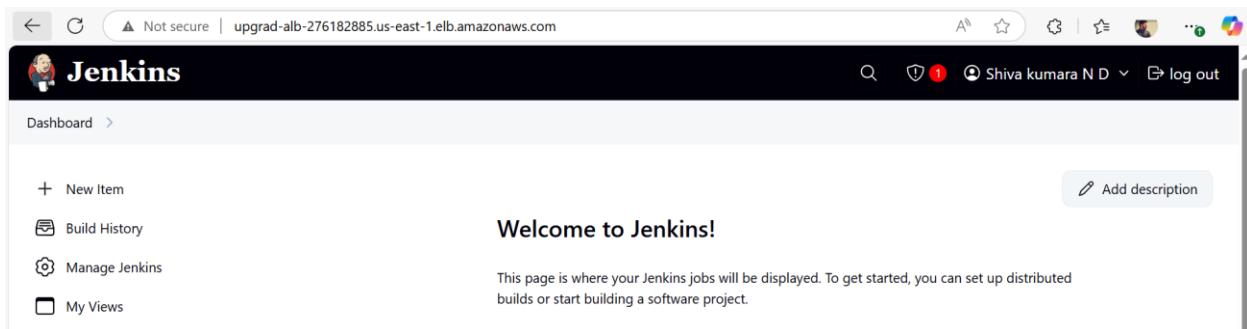
Instance Configuration

Jenkins URL:

http://upgrad-alb-276182885.us-east-1.elb.amazonaws.com/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper

Jenkins is ready to use



A screenshot of a web browser displaying the Jenkins dashboard. The title bar shows the URL: http://upgrad-alb-276182885.us-east-1.elb.amazonaws.com. The main content area has a dark header with the Jenkins logo and the word "Jenkins". Below the header, there's a "Dashboard" link and a "New Item" button. On the left, there are links for "Build History", "Manage Jenkins", and "My Views". The central area displays the "Welcome to Jenkins!" message, which reads: "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." There are also "Add description" and "log out" buttons.

Made changes in `/etc/default/jenkins` file to add `--prefix=/jenkins`

```
# --httpListenAddress=$HTTP_HOST (default 0.0.0.0)
# --httpPort=$HTTP_PORT (default 8080; disable with -1)
# --httpsPort=$HTTP_PORT
# --argumentsRealm.passwd.$ADMIN_USER=[password]
# --argumentsRealm.roles.$ADMIN_USER=admin
# --webroot=/.jenkins/war
# --prefix=$PREFIX

JENKINS_ARGS="--webroot=/var/cache/$NAME/war --httpPort=$HTTP_PORT --prefix=/jenkins"
"/etc/default/jenkins" [readonly] 80L, 2849B
```

Sub task 4

Creating ECR repository

The screenshot shows the Amazon ECR console under the 'Private registry' section. A success message 'Successfully created node-app' is displayed. The 'Private repositories' list shows one entry: 'node-app' with URI '237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app'. The table includes columns for Repository name, URI, Created at, Tag immutability, and Encryption type.

IAM role is created for ECR access

The screenshot shows the AWS IAM console under the 'Access management' section. A new role 'EC2ECRAccessRole' is being configured. It has two attached managed policies: 'AmazonEC2ContainerRegistryFullAccess' and 'AmazonS3ReadOnlyAccess'. The 'Permissions policies' section shows these two policies listed.

IAM role is assigned to App and Jenkins server

The screenshot shows the AWS EC2 instance details page for instance 'i-0e43dd1645c01f07a'. Under the 'Modify IAM role' section, the 'Instance ID' is listed as 'i-0e43dd1645c01f07a (app-instance)'. The 'IAM role' dropdown is set to 'EC2ECRAccessRole'. A button to 'Create new IAM role' is also visible.

The screenshot shows the 'Modify IAM role' page in the AWS Management Console. The top navigation bar includes 'EC2 > Instances > i-0094b427344c0a7c0 > Modify IAM role'. The main section is titled 'Modify IAM role' with a 'Info' link. A sub-instruction 'Attach an IAM role to your instance.' is present. The 'Instance ID' field shows 'i-0094b427344c0a7c0 (jenkins-instance)'. The 'IAM role' dropdown menu contains 'EC2ECRAccessRole' and a 'Create new IAM role' button with a blue circular icon.

Generating SSH keygen on jenkin server (10.100.4.216)

```
ubuntu@ip-10-100-4-216:~$ ssh-keygen -t rsa -b 4096 -f ~/.ssh/upgrad_task_key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/upgrad_task_key
Your public key has been saved in /home/ubuntu/.ssh/upgrad_task_key.pub
The key fingerprint is:
SHA256:a7DZRKZqZz2aUTid0YA9FD8g8y/hF01XWoJCx4ZBagU ubuntu@ip-10-100-4-216
The key's randomart image is:
+---[RSA 4096]---+
|   E==+o.o.o |
|   B =o++. +  |
| . O *o. .    |
| o @ =        |
|   B S         |
|   . % o       |
| o * *         |
| . o = .       |
|   o           |
+---[SHA256]---+
ubuntu@ip-10-100-4-216:~$ |
```

Able to connect from Jenkins server (10.100.4.216) to app server (10.100.3.135)

```
ubuntu@ip-10-100-4-216:~$ ssh -i ~/.ssh/upgrad_task_key ubuntu@10.100.3.135
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Wed Jul 23 08:40:47 UTC 2025

 System load: 0.080078125      Processes:          100
 Usage of /: 49.1% of 7.57GB   Users logged in:    1
 Memory usage: 21%            IPv4 address for docker0: 172.17.0.1
 Swap usage: 0%                IPv4 address for eth0: 10.100.3.135
```

Able to connect from jenkin server (10.100.4.216) to ECR repository

```
ubuntu@ip-10-100-4-216:~$ aws ecr get-login-password --region us-east-1 | \
docker login --username AWS --password-stdin 237458753027.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
ubuntu@ip-10-100-4-216:~$ |
```

Able to connect from App server (10.100.3.135) to ECR repository

```
ubuntu@ip-10-100-3-135:~$ aws ecr get-login-password --region us-east-1 | \
docker login --username AWS --password-stdin 237458753027.dkr.ecr.us-east-1.amazonaws.com
WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
ubuntu@ip-10-100-3-135:~$ |
```

Node app on app server (10.100.3.135).

```
Login Succeeded
ubuntu@ip-10-100-3-135:~/node-app$ ls -l
total 12
-rw-rw-r-- 1 ubuntu ubuntu 258 Jul 23 09:04 dockerfile
-rw-rw-r-- 1 ubuntu ubuntu 248 Jul 23 09:03 index.js
-rw-rw-r-- 1 ubuntu ubuntu 169 Jul 23 09:03 package.json
ubuntu@ip-10-100-3-135:~/node-app$ |
```

Docker image of node app is built on app server (10.100.3.135)

```
ubuntu@ip-10-100-3-135:~/node-app$ docker build -t node-app .
[+] Building 36.2s (10/10) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 297B
=> [internal] load metadata for docker.io/library/node:18
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:18@sha256:c6ae79e38498325db67193d391e6ec1d224d96c693a8a4d943498556716d37
=> => resolve docker.io/library/node:18@sha256:c6ae79e38498325db67193d391e6ec1d224d96c693a8a4d943498556716d37
=> sha256:c6ae79e38498325db67193d391e6ec1d224d96c693a8a4d943498556716d3783 6.41kB / 6.41kB
=> sha256:3e6b9d1a95114e19f12262a4e8a59ad1d1a10ca7b82108adcf0605a200294964 48.49MB / 48.49MB
=> sha256:37927ed901b1b2608b72796c6881bf645480268eca4ac9a37b9219e050bb4d84 24.02MB / 24.02MB
=> sha256:79b2f47ad4443652b9b5cc81a95ede249fd976310efdbe159f29638783778c0 64.40MB / 64.40MB
=> sha256:eb29363371ee2859fad6a3c5af88d4abc6ff7d399addb13b7de3c1f11bdee6b9 2.49kB / 2.49kB
=> sha256:b50082bc3670d0396b2d90e4b0e5bb10265ba5d0ee16bf40f9a505f7045ee563 6.39kB / 6.39kB
=> sha256:e23f099911d692f62b851cf49a1e93294288a115f5cd2d014180e4d3684d34ab 211.36MB / 211.36MB
=> => extracting sha256:3e6b9d1a95114e19f12262a4e8a59ad1d1a10ca7b82108adcf0605a200294964
=> sha256:cda7f44f2bddcc4bb7514474024b3f3705de0ddb6355a33be5ac9a37b9219e050bb4d84 3.32kB / 3.32kB
=> sha256:c6b30c3f16966552af10ac00521f60355b1fcfd46ac1c20b1038587e28583ce7 45.68MB / 45.68MB
=> sha256:3697be50c98b9d071df4637eid3491d00e7b9f3a732768c876d82309b3c5a145 1.25MB / 1.25MB
=> sha256:461077a2fb7fe40d34a37d6a1958c4d16772d0d77f572ec50a1fdc41a3754d 446B / 446B
=> => extracting sha256:37927ed901b1b2608b72796c6881bf645480268eca4ac9a37b9219e050bb4d84
=> => extracting sha256:79b2f47ad4443652b9b5cc81a95ede249fd976310efdbe159f29638783778c0
=> => extracting sha256:e23f099911d692f62b851cf49a1e93294288a115f5cd2d014180e4d3684d34ab
=> => extracting sha256:cda7f44f2bddcc4bb7514474024b3f3705de0ddb6355a33be5ac9a37b9219e050bb4d84
```

```
ubuntu@ip-10-100-3-135:~/node-app$ docker run -d -p 3000:3000 node-app
e5114ef5021d55ccdc3901b66a3c1c6bd18859f57a282511db82c2502e8a8031
```

Pushing image to the ECR from app server (10.100.3.135)

```
ubuntu@ip-10-100-3-135:~/node-app$ docker tag node-app:latest 237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app:latest
ubuntu@ip-10-100-3-135:~/node-app$ docker push 237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app:latest
The push refers to repository [237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app]
50f4a08f3021: Pushed
fb02fcf523ba: Pushed
4afcfb3e85e2: Pushed
9b16fd7e710d: Pushed
d2a991bcab4d: Pushed
b624aa2d5ea2: Pushed
d399c9dc306f: Pushed
84f9fa179c1b: Pushed
ce84ba212e49: Pushed
e4dc8cd9ecc8: Pushed
6428cc293366: Pushed
2f7436e79a0b: Pushed
latest: digest: sha256:a64615fff40e145241ea7404eb20ab1ffc4b760febe0c0230370573e2c71a8fa size: 2835
ubuntu@ip-10-100-3-135:~/node-app$
```

Images (1)						
C Delete Details Scan View push commands						
Search artifacts						
<input type="checkbox"/>	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
						Last recorded pull time
<input type="checkbox"/>	latest	Image	July 23, 2025, 14:39:56 (UTC+05.5)	398.64	Copy URI	sha256:a64615ff...

Pulling image from ECR to Jenkins server (10.100.4.216)

```
ubuntu@ip-10-100-4-216:~$ docker pull 237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app:latest
latest: Pulling from node-app
5c9256e0f3ca: Pull complete
37927ed901b1: Pull complete
79b2f47ad444: Pull complete
e23f099911d6: Pull complete
cda7f44f2bdd: Pull complete
c6b30c3f1696: Pull complete
3697be50c98b: Pull complete
461077a72fb7: Pull complete
2a484a399430: Pull complete
736b8c7eac4c: Pull complete
0c209d5c4be9: Pull complete
cd908557d16c: Pull complete
Digest: sha256:a64615fff40e145241ea7404eb20ab1ffc4b760febe0c0230370573e2c71a8fa
Status: Downloaded newer image for 237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app:latest
237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app:latest
ubuntu@ip-10-100-4-216:~$ |
```

Jenkins job is able to ssh into the app host (10.100.3.135) and run simple commands.

The screenshot shows the Jenkins interface for a build named "Test-SSH-To-App" (Build #2). The "Console Output" tab is selected. The output window displays the following log entries:

```
Started by user Shiva kumara N D
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Test-SSH-To-App
[Test-SSH-To-App] $ /bin/sh -xe /tmp/jenkins13205258763659490567.sh
+ ssh -i /var/lib/jenkins/.ssh/upgrad_task_key -o StrictHostKeyChecking=no ubuntu@10.100.3.135 echo Hello
from Jenkins && hostname && docker ps
Hello from Jenkins
ip-10-100-3-135
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
e5114ef5021d node-app "docker-entrypoint.s..." 19 hours ago Up 19 hours 0.0.0.0:3000->3000/tcp,
[::]:3000->3000/tcp funny_moore
Finished: SUCCESS
```

Task 3

Dockerize sample Node.js app & write Jenkinsfile

Created simple node application with 1 API defined and stored in in private github repository

The screenshot shows a GitHub repository page for 'node-demo-app'. The repository belongs to user 'shivu9900'. The commit history is displayed, showing two commits:

Name	Last commit message	Last commit date
index.js	Create index.js	now
package.json	Create package.json	now

Docker file setup

```
ubuntu@ip-10-100-1-15: ~
# Use official Node.js base image
FROM node:18-alpine

# Set working directory
WORKDIR /app

# Copy package files and install dependencies
COPY package*.json ./
RUN npm install

# Copy the rest of the app
COPY .

# Expose app port
EXPOSE 3000

# Run the app
CMD ["npm", "start"]
```

Pushing dockerfile to github

```
ubuntu@ip-10-100-1-157:~/node-demo-app$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ubuntu/node-demo-app/.git/
ubuntu@ip-10-100-1-157:~/node-demo-app$ git remote add origin https://github.com/shivu9900/node-demo-app.git
ubuntu@ip-10-100-1-157:~/node-demo-app$ git add .
ubuntu@ip-10-100-1-157:~/node-demo-app$ git commit -m "Initial commit with Node app and Dockerfile"
```

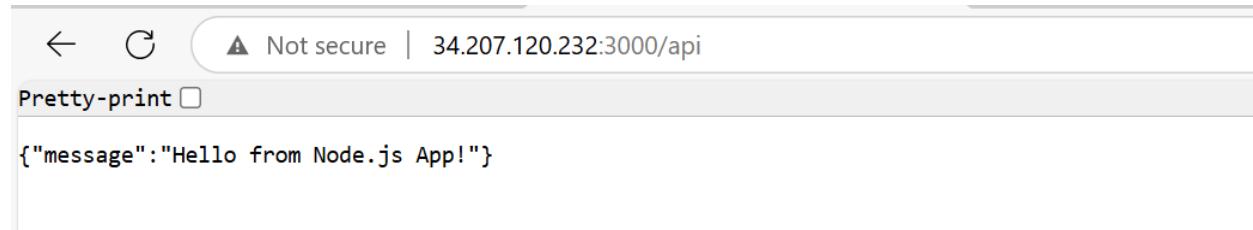
Docker build locally to test

```
ubuntu@ip-10-100-1-157:~/node-demo-app$ sudo docker build -t node-demo-app:latest .
[+] Building 4.2s (6/9)
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 151B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
=> => sha256:929b04d7c782f04f615cf785488fed452b6569f87c73ff666ad53a7554f0006 1.72kB / 1.72kB
=> => sha256:ee77c6cd7c1886ecc802ad6cedef3a8eclea27d1fb96162bf03dd3710839b8da 6.18kB / 6.18kB
=> => sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4feea161b2b5a3d0e 40.01MB / 40.01MB
=> => sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3 1.26MB / 1.26MB
=> => sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e 7.67kB / 7.67kB
=> => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e9c2d5c870
=> => sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549 446B / 446B
=> => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e9c2d5c870 3.64MB / 3.64MB
=> => extracting sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4feea161b2b5a3d0e
=> => extracting sha256:1e5a4c89cee5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3
```

Docker run

```
ubuntu@ip-10-100-1-157:~/node-demo-app$ sudo docker run -d -p 3000:3000 node-demo-app
4d66722ed15b78ede9a5aa73a3e7818dca7d178c159d3d09d5c394c5af44234c
```

Deployed application API is able to call from browser



Subtask -2

stage 1

Git clone node app and dockerfile into app server (10.100.3.135)

```
ubuntu@ip-10-100-3-135:~$ git clone https://github.com/shivu9900/node-demo-app.git
Cloning into 'node-demo-app'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), done.
ubuntu@ip-10-100-3-135:~$ cd node-demo-app
ubuntu@ip-10-100-3-135:~/node-demo-app$ |
```

Docker build on app server (10.100.3.135)

```
ubuntu@ip-10-100-3-135:~/node-demo-app$ docker build -t node-demo-app:latest .
[+] Building 11.8s (10/10) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 150B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
=> => resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
=> => sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4fe161b2b5a3d0e 40.01MB / 40.01MB
=> => sha256:1e5a4c89ce5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3 1.26MB / 1.26MB
=> => sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e 7.67KB / 7.67KB
=> => sha256:929b04d7c782f04f615cf785488fed452b6569f87c73ff666ad53a7554f0006 1.72KB / 1.72KB
=> => sha256:ee77c6cd7c1886ecc802ad6cdef3a8ec1ea27dfb96162bf03dd3710839b8da 6.18KB / 6.18KB
=> => sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB
=> => extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870
=> => sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549 446B / 446B
=> => extracting sha256:dd71dde834b5c203d162902e6b8994cb2309ae049a0eabc4fe161b2b5a3d0e
=> => extracting sha256:1e5a4c89ce5c0826c540ab06d4b6b491c96eda01837f430bd47f0d26702d6e3
=> => extracting sha256:25ff2da83641908f65c3a74d80409d6b1b62ccfaab220b9ea70b80df5a2e0549
=> [internal] load build context
=> => transferring context: 32.48kB
=> [2/5] WORKDIR /app
=> [3/5] COPY package*.json .
=> [4/5] RUN npm install
=> [5/5] COPY .
=> exporting to image
=> => exporting layers
=> => writing image sha256:8e3fdceed722a1ab17244387f659798906b4c97063306de094077bf3a76f2d77
=> => naming to docker.io/library/node-demo-app:latest
```

Connecting to ECR to push the docker image

```
=> => naming to docker.io/library/node-demo-app:latest
ubuntu@ip-10-100-3-135:~/node-demo-app$ aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 237458753027.dkr.ecr.us-east-1.amazonaws.com
          U,US
WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
ubuntu@ip-10-100-3-135:~/node-demo-app$ |
```

Pushed the docker image to ECR registry

```
ubuntu@ip-10-100-3-135:~/node-demo-app$ docker push 237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app:latest
The push refers to repository [237458753027.dkr.ecr.us-east-1.amazonaws.com/node-app]
1920cb59048e: Pushed
125da8c23e8f: Pushed
592c390b2f1c: Pushed
9d0460705ad7: Pushed
82140d9a70a7: Pushed
f3b40b0cdb1c: Pushed
0b1f26057bd0: Pushed
08000c18d16d: Pushed
latest: digest: sha256:c1570ce0218187159fe2245b0db5cb389ddc219b7422f5183d0124383cbb1e63 size: 1991
ubuntu@ip-10-100-3-135:~/node-demo-app$ |
```

Image on ECR registry in aws

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

The screenshot shows the AWS ECR Images page. At the top, there is a blue header bar with the message: "ⓘ Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag." Below the header, there is a search bar labeled "Search artifacts". The main area is titled "Images (2)". There are several buttons at the top right: "Delete", "Details", "Scan", and "View push commands". A table below lists the images. The columns are: "Image tag", "Artifact type", "Pushed at", "Size (MB)", "Image URI", "Digest", and "Last recorded pull time". The first row shows an image tagged "latest" which was pushed on July 24, 2025, at 11:38:47 (UTC+05.5). The size is 47.28 MB, and the digest is sha256:c1570ce0... . The last recorded pull time is listed as "-". There are "Copy URI" and "Delete" buttons next to the image URI.

<input type="checkbox"/>	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
<input type="checkbox"/>	latest	Image	July 24, 2025, 11:38:47 (UTC+05.5)	47.28	Copy URI	sha256:c1570ce0...	-

Stage 2

Jenkins build is successful to pull docker image from git and build it in server and push image to ECR

The screenshot shows a GitHub repository page for 'node-demo-app'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below the navigation bar, the repository name 'node-demo-app' is displayed, along with a 'Public' badge. There are buttons for Pin and Watch. The main content area shows a commit history with 7 commits. The latest commit is by 'shivu9900' titled 'Update Jenkinsfile' with a timestamp of 'a58e001 · 17 minutes ago'. Below the commit list are file details for Jenkinsfile, dockerfile, index.js, and package.json, each with their respective creation or update times. A 'README' file is also listed.

The screenshot shows the Jenkins Pipeline configuration screen for a job named 'NodeApp-ECR-Build'. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main panel is titled 'Configure' and shows the SCM section set to 'Git'. Under 'Repositories', there is one entry with a 'Repository URL' of 'https://github.com/shivu9900/node-demo-app.git' and 'Credentials' set to '- none -'. There is also a '+' button to add more repositories.

The screenshot shows the Jenkins interface for a build named "NodeApp-ECR-Build" (Build #5). On the left, there's a sidebar with various build-related links. The main area is titled "Console Output" and displays the build log. The log output is as follows:

```

Started by user Shiva kumara N D
Obtained Jenkinsfile from git https://github.com/shivu9900/node-demo-app.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/NodeApp-ECR-Build
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/NodeApp-ECR-Build/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/shivu9900/node-demo-app.git # timeout=10
Fetching upstream changes from https://github.com/shivu9900/node-demo-app.git

```



```

2cdc1aa23f05: Pushed
b1a5766fefc5: Pushed
a3b60e9741f6: Pushed
latest: digest: sha256:d716d0d8e3a6ecf32a1e5e99b478dd448ead37851c4d8ca5b50a5b41ccbf00d6 size: 1991
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

The first image in the list is pushed from Jenkins job

The screenshot shows the Amazon ECR console for the "node-app" repository. The left sidebar has sections for "Private registry" (Repositories, Images, Permissions, Lifecycle Policy, Repository tags, Features & Settings) and "Public registry" (Repositories, Settings). The main area is titled "Images (3)" and shows the following table:

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
latest	Image	July 24, 2025, 12:08:13 (UTC+05.5)	47.29	Copy URI	sha256:d716d0d8e3a6ecf32a1e5e99b478dd448ead37851c4d8ca5b50a5b41ccbf00d6...	-
-	Image	July 24, 2025, 11:38:47 (UTC+05.5)	47.28	Copy URI	sha256:c1570ce0...	-

Sub task 3

Jenkin file with app server details to deploy node application on app server through Jenkins pipeline build

Jenkinsfile_app jenkins file attached in the zip is used here

Able to connect from staging server to app server (10.100.3.135)

```
ubuntu@ip-10-100-4-216:~$ ssh -i ~/.ssh/upgrad_task_key ubuntu@10.100.3.135
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1012-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Jul 23 08:40:47 UTC 2025

System load: 0.080078125      Processes:                 100
Usage of /:   49.1% of 7.57GB  Users logged in:        1
Memory usage: 21%              IPv4 address for docker0: 172.17.0.1
Swap usage:   0%                IPv4 address for eth0:   10.100.3.135
```

Dashboard > NodeApp-ECR-Build > #9

Status
Console Output
Download
Copy
View as plain text

Started by user Shiva kumara N D
 Obtained Jenkinsfile from git <https://github.com/shivu9900/node-demo-app.git>
 [Pipeline] Start of Pipeline
 [Pipeline] node
 Running on Jenkins in /var/lib/jenkins/workspace/NodeApp-ECR-Build
 [Pipeline] {
 [Pipeline] stage
 [Pipeline] { (Declarative: Checkout SCM)
 [Pipeline] checkout
 Selected Git installation does not exist. Using Default
 The recommended git tool is: NONE
 No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/NodeApp-ECR-Build/.git # timeout=10
 Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/shivu9900/node-demo-app.git # timeout=10
 Fetching upstream changes from https://github.com/shivu9900/node-demo-app.git
 > git --version # timeout=10

- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#9'
- Timings
- Git Build Data
- Pipeline Overview
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces

Latest images pushed to ECR repository

Amazon ECR > Private registry > Repositories > node-app

Images (7)						
	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
<input type="checkbox"/>	latest	Image	July 24, 2025, 12:46:30 (UTC+05.5)	47.30	Copy URI	sha256:fcddb906...
<input type="checkbox"/>	-	Image	July 24, 2025, 12:42:33 (UTC+05.5)	47.30	Copy URI	sha256:010bba1...
July 24.						

App files and docker files in app server which deployed through Jenkins build

```
ubuntu@ip-10-100-3-135:~$ cd node-app
ubuntu@ip-10-100-3-135:~/node-app$ ls -l
total 12
-rw-rw-r-- 1 ubuntu ubuntu 258 Jul 23 09:04 dockerfile
-rw-rw-r-- 1 ubuntu ubuntu 248 Jul 23 09:03 index.js
-rw-rw-r-- 1 ubuntu ubuntu 169 Jul 23 09:03 package.json
ubuntu@ip-10-100-3-135:~/node-app$ |
```

Subtask bonus:

Calculate the total hourly/monthly cost of the complete infrastructure.

Component	Hourly	Monthly
EC2 (3 x t2.small)	\$0.069	\$49.68
NAT Gateway	\$0.045	~\$33.75
ALB	~\$0.03	~\$22.00
ECR (First 500 MB is free)	Free	\$0.00
S3 (First 5 GB is free)	Free	\$0.00
VPC / IGW / SGs	Free	\$0.00
Total	~\$0.144/hour	~\$105.43/month