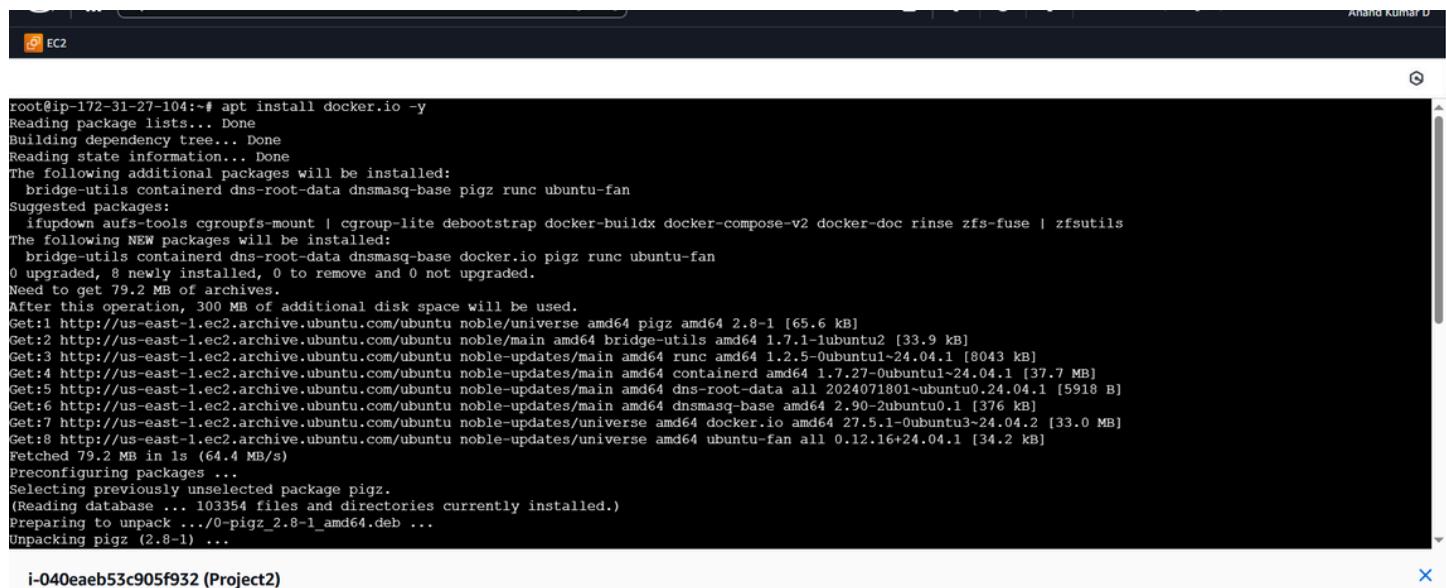


Project: TrendStore

1.Docker

1.Installing Docker:



```
root@ip-172-31-27-104:~# apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 0 not upgraded.
Need to get 79.2 MB of archives.
After this operation, 300 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.2.5-0ubuntu1-24.04.1 [8043 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.27-0ubuntu1-24.04.1 [37.7 MB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801~ubuntu0.24.04.1 [5918 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 27.5.1-0ubuntu3~24.04.2 [33.0 MB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 79.2 MB in 1s (64.4 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 103354 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...

i-040eaeb53c905f932 (Project2)
```

2.Docker Installed:

```
root@ip-172-31-27-104:~# docker --version
Docker version 27.5.1, build 27.5.1-0ubuntu3~24.04.2
root@ip-172-31-27-104:~# █
```

3.Application Files:

 EC2

```
root@ip-172-31-27-104:~/project# ls
Dockerfile  dist
root@ip-172-31-27-104:~/project# █
```

4.Dockerfile Code:

 EC2

```
GNU nano 7.2
FROM nginx:alpine
COPY dist/ /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

5.Building Docker Image:

```

root@ip-172-31-27-104:~/project# docker build -t trendstore .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 18.36MB
Step 1/4 : FROM nginx:alpine
alpine: Pulling from library/nginx
9824c27679d3: Pull complete
6bc572a340ec: Pull complete
403e3f251637: Pull complete
9adfbbae99cb7: Pull complete
7a8a46741e18: Pull complete
c9ebe2ff2d2c: Pull complete
a992fbc61ecc: Pull complete
cb1ff4086f82: Pull complete
Digest: sha256:42a516af16b852e33b7682d5ef8acbd5d13fe08fecadc7ed98605ba5e3b26ab8
Status: Downloaded newer image for nginx:alpine
--> 4a86014ec699
Step 2/4 : COPY dist/ /usr/share/nginx/html
--> a768bdac2f4e
Step 3/4 : EXPOSE 80
--> Running in e5e6abe2236d
--> Removed intermediate container e5e6abe2236d
--> 69b3dd07149b
Step 4/4 : CMD ["nginx", "-g", "daemon off;"]
--> Running in 11a703921061

```

i-040eaeb53c905f932 (Project2)

6.Docker Container Created:

```

root@ip-172-31-27-104:~/project# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
trendstore      latest   e425b3eb7be2  57 seconds ago  61.8MB
nginx           alpine   4a86014ec699  2 weeks ago   52.5MB
root@ip-172-31-27-104:~/project#
root@ip-172-31-27-104:~/project# docker run -itd -p "3000:80" --name trendstore trendstore
6978905bcaa52d74580fac66720dc97bc40495ed02ba8372f1d4778a1c368cbc
root@ip-172-31-27-104:~/project# █

```

7.Enabling Port 3000 To Access the Application:

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-04ac34e9616d92e62	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-044ef18e1def8ad91	HTTPS	TCP	443	Custom	0.0.0.0/0
sgr-0044d548b88fd762d	SSH	TCP	22	Custom	0.0.0.0/0
-	Custom TCP	TCP	3000	Anywhere	0.0.0.0/0
					TrendStore Application

Add rule

8. Accessing the Application in Browser - Output 1:

The screenshot shows the Trendify homepage. At the top, there is a navigation bar with links for HOME, COLLECTION, ABOUT, and CONTACT. To the right of the navigation bar are icons for search, heart, bag, and user profile. Below the navigation bar, there is a section titled "OUR BEST SELLERS" with the heading "Latest Arrivals". A "SHOP NOW" button is located below the heading. To the right of this text area is a large, prominent image of a woman with blonde hair, wearing a black top, set against a pink background. Below this main image, there is a horizontal line with the text "LATEST COLLECTIONS" centered under it.

9. Application Output 2:

The screenshot shows a product page for a men's slim fit relaxed denim jacket. At the top, there is a navigation bar with links for HOME, COLLECTION, ABOUT, and CONTACT. To the right of the navigation bar are icons for search, heart, bag, and user profile. On the left side of the page, there is a large image of a man wearing a yellow quilted jacket. To the left of this main image is a smaller thumbnail image of the same jacket. To the right of the main image, the product title "Men Slim Fit Relaxed Denim Jacket" is displayed, followed by a rating of "★★★★★ 0 (0)". The price "\$230" is prominently displayed. Below the price, a product description states: "A lightweight, usually knitted, pullover shirt, close-fitting and with a round neckline and short sleeves, worn as an undershirt or outer garment." There is a "Select Size" section with size options S, M, L, and XL. A "ADD TO CART" button is located below the size options. At the bottom of the page, there is a footer with the text: "Guaranteed 100% Authentic – Shop with Confidence! Enjoy Cash on Delivery – Pay at Your Doorstep! Hassle-Free Returns & Exchanges – 10 Days, No Questions Asked!"

10. Application Output 3:

The screenshot shows a web browser window with the URL <https://54.226.41.207:3000/about>. The page has a header with the 'trendify' logo and navigation links for HOME, COLLECTION, ABOUT, and CONTACT. A search bar and user icons are also present. The main content area features a heading 'ABOUT US —' above a photograph of a fashion haul including jeans, a cable-knit sweater, boots, and a small bag. To the right of the photo is a welcome message from Trendify, followed by a paragraph about prioritizing customer satisfaction. Below these are sections for 'Our Mission' and 'Our Vision'.

Welcome to Trendify, where style meets quality. Our mission is to bring you the latest fashion trends and must-have items, all curated with an eye for quality and design. We believe that everyone deserves to express themselves through fashion, and we're here to make that easier and more enjoyable. Our collections are carefully selected to offer you a range of options that cater to every taste and occasion.

At Trendify, we prioritize your satisfaction. From the moment you browse our site to the day your order arrives, we are dedicated to providing a seamless shopping experience. Our team is always on the lookout for the latest trends, ensuring that you have access to the freshest styles as soon as they hit the runway. Thank you for choosing Trendify. We're excited to be a part of your style journey.

Our Mission

At Trendify, our mission is to empower you to express your unique style with high-quality, on-trend fashion. We strive to make fashion accessible to all, offering diverse products that inspire confidence.

Our Vision

At Trendify, our vision is to be a global fashion leader, known for cutting-edge style and quality. We aim to inspire confidence and creativity, making Trendify the go-to choice for individual

2.Terraform

1.Installing Terraform:

A terminal window titled 'EC2' showing the command-line process for installing Terraform. It includes commands to download the HashiCorp GPG key, update the package list, and install Terraform itself.

```
root@ip-172-31-27-104:~# wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(grep -oP '(?=<=UBUNTU_CODENAME=).*/etc/os-release || lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt update && sudo apt install terraform
```

:

2.Terraform Installed:

A terminal window showing the output of the 'terraform -v' command, indicating Terraform version 1.13.1 is installed on a Linux AMD64 system.

```
root@ip-172-31-27-104:~# terraform -v
Terraform v1.13.1
on linux_amd64
root@ip-172-31-27-104:~#
```

3.AWS CLI Installed:

```
root@ip-172-31-27-104:~# aws --version
aws-cli/2.28.21 Python/3.13.7 Linux/6.14.0-1011-aws exe/x86_64.ubuntu.24
root@ip-172-31-27-104:~# █
```

4. Enabling Programmatic Access To the VM:

```
root@ip-172-31-27-104:~# aws configure
AWS Access Key ID [None]: AKIA3O204QWYMS4TPDMO
AWS Secret Access Key [None]: RRcLepKWPOTJZp8mHDCT8D7IAPTB4YPuz5HMQ3TO
Default region name [None]: us-east-1
Default output format [None]: json
root@ip-172-31-27-104:~# █
```

5. Main.tf File First part:

```
provider "aws" {
  region = "us-east-1"
}
#For VPC
resource "aws_vpc" "my_vpc" {
  cidr_block = "10.0.0.0/16"
  tags = { Name = "myapp-vpc" }
}
#subnet
resource "aws_subnet" "public_subnet" {
  vpc_id          = aws_vpc.my_vpc.id
  cidr_block     = "10.0.1.0/24"
  map_public_ip_on_launch = true
  availability_zone = "us-east-1a"
  tags = { Name = "myapp-public-subnet" }
}
#internet_Gateway
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.my_vpc.id
  tags = { Name = "myapp-igw" }
}
#Route Table
resource "aws_route_table" "public_rt" {
  vpc_id = aws_vpc.my_vpc.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }
  tags = { Name = "myapp-public-rt" }
}

resource "aws_route_table_association" "a" {
  subnet_id   = aws_subnet.public_subnet.id
  route_table_id = aws_route_table.public_rt.id
}
#security_Group
resource "aws_security_group" "jenkins_sg" {
  name        = "jenkins-sg"
  description = "Allow SSH, HTTP, Jenkins"
  vpc_id      = aws_vpc.my_vpc.id
```

6. Main.tf Second Part:

```

ingress {
  from_port   = 22
  to_port     = 22
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}

ingress {
  from_port   = 8080
  to_port     = 8080
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}

ingress {
  from_port   = 80
  to_port     = 80
  protocol    = "tcp"
  cidr_blocks = ["0.0.0.0/0"]
}

egress {
  from_port   = 0
  to_port     = 0
  protocol    = "-1"
  cidr_blocks = ["0.0.0.0/0"]
}

tags = { Name = "jenkins-sg" }
}
#Instance For Jenkins Server
resource "aws_instance" "jenkins" {
  ami           = "ami-0360c520857e3138f"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.public_subnet.id
  vpc_security_group_ids = [aws_security_group.jenkins_sg.id]
  key_name      = "Project2"

  tags = {
    Name = "jenkins-server"
  }
}

```

7.Variables.tf File:

```

GNU nano 7.2
variable "region" {
  default = "us-east-1"
}
variables.tf

```

8.Output.tf File:

```

GNU nano 7.2
output "jenkins_public_ip" {
  value = aws_instance.jenkins.public_ip
}
outputs.tf *

```

9.Terraform Files:

```
root@ip-172-31-27-104:~/terraform-infra# ls  
main.tf  outputs.tf  variables.tf  
root@ip-172-31-27-104:~/terraform-infra#
```

10.Initialized Terraform:

```
root@ip-172-31-27-104:~/terraform-infra# terraform init  
Initializing the backend...  
Initializing provider plugins...  
- Finding latest version of hashicorp/aws...  
- Installing hashicorp/aws v6.11.0...  
- Installed hashicorp/aws v6.11.0 (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.

```
root@ip-172-31-27-104:~/terraform-infra#
```

11.Running Terraform:

```
root@ip-172-31-27-104:~/terraform-infra# 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_instance.jenkins will be created  
+ resource "aws_instance" "jenkins" {  
    + ami = "ami-0360c520857e3130f"  
    + arn = (known after apply)  
    + associate_public_ip_address = (known after apply)  
    + availability_zone = (known after apply)  
    + disable_api_stop = (known after apply)  
    + disable_api_termination = (known after apply)  
    + ebs_optimized = (known after apply)  
    + enable_primary_ipv6 = (known after apply)  
    + force_destroy = false  
    + get_password_data = false  
    + host_id = (known after apply)  
    + host_resource_group_arn = (known after apply)  
    + iam_instance_profile = (known after apply)  
    + id = (known after apply)  
    + instance_initiated_shutdown_behavior = (known after apply)  
    + instance_lifecycle = (known after apply)  
    + instance_state = (known after apply)  
    + instance_type = "t2.micro"
```

12.Infrastructure Created:

```
aws_vpc.my_vpc: Creating...
aws_vpc.my_vpc: Creation complete after 2s [id=vpc-0526393953097c16f]
aws_security_group.jenkins_sg: Creating...
aws_subnet.public_subnet: Creating...
aws_internet_gateway.igw: Creating...
aws_internet_gateway.igw: Creation complete after 0s [id=igw-0c9d89297ecdafe82]
aws_route_table.public_rt: Creating...
aws_route_table.public_rt: Creation complete after 1s [id=rtb-00291e56da88ed90c]
aws_security_group.jenkins_sg: Creation complete after 2s [id=sg-036247dbf881e809c]
aws_subnet.public_subnet: Still creating... [00m10s elapsed]
aws_subnet.public_subnet: Creation complete after 11s [id=subnet-0cf71bcb5643156e2]
aws_instance.jenkins: Creating...
aws_route_table_association.a: Creating...
aws_route_table_association.a: Creation complete after 0s [id=rtbassoc-007e0e90d23b4257d]
aws_instance.jenkins: Still creating... [00m10s elapsed]
aws_instance.jenkins: Still creating... [00m20s elapsed]
aws_instance.jenkins: Still creating... [00m30s elapsed]
aws_instance.jenkins: Creation complete after 31s [id=i-0044a3fa32cb9ebe3]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
```

Outputs:

```
jenkins_public_ip = "3.231.146.7"
```

```
root@ip-172-31-27-104:~/terraform-infra#
```

i-040eaeb53c905f932 (Project2)

13.Jenkins Server Created Using Terraform:

Instances (3) Info		Last updated less than a minute ago	Connect	Instance state ▾	Actions ▾	Launch instances		
<input type="text"/> Find Instance by attribute or tag (case-sensitive)				All states ▾				
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	jenkins-server	i-0044a3fa32cb9ebe3	Running Q Q	t2.micro	2/2 checks passed View alarms +	View alarms +	us-east-1a	-
<input type="checkbox"/>	test	i-03321cf1133f75169	Stopped Q Q	t2.micro	-	View alarms +	us-east-1c	-
<input type="checkbox"/>	Project2	i-040eaeb53c905f932	Running Q Q	t2.micro	2/2 checks passed View alarms +	View alarms +	us-east-1c	ec2-54-2-

Select an instance



14.VPC Created:

Your VPCs (2) Info		Last updated less than a minute ago	Actions ▾	Create VPC		
<input type="text"/> Find VPCs by attribute or tag						
	Name	VPC ID	State	Block Public... ▾	IPv4 CIDR	IPv6 CIDR
<input type="checkbox"/>	myapp-vpc	vpc-0526393953097c16f	Available Q	Off Q	10.0.0.0/16	-
<input type="checkbox"/>	-	vpc-01f5ce5f3796adcd4	Available Q	Off Q	172.31.0.0/16	-

3.DockerHub

1.Creating DockerHub Repository:

The screenshot shows the Docker Hub interface for creating a new repository. At the top, there's a blue header bar with the 'Hub' logo, a search bar, and various navigation icons. Below the header, the URL 'Repositories / Create' is visible, along with a message indicating 'Using 1 of 1 private repositories.' A 'Get more' link is also present. The main form is titled 'Create repository' and contains a 'Repository Name *' field where 'project2' is typed. To the right, there's a section titled 'Pushing images' with instructions on how to push a new image using the CLI. Below the repository name field is a 'Short description' input area containing placeholder text about identifying the repository. Under 'Visibility', there are two options: 'Public' (selected) and 'Private'. The 'Public' option includes a note that it appears in Docker Hub search results. The 'Private' option is described as being only visible to the user. At the bottom of the form are 'Cancel' and 'Create' buttons.

2.Logged into the DockerHub:

```
root@ip-172-31-27-104:~# docker login

USING WEB-BASED LOGIN
To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: JXVJ-NSCQ
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
root@ip-172-31-27-104:~#
```

3.Docker Image Pushed to DockerHub:

```
root@ip-172-31-27-104:~# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
trendstore      latest    e425b3eb7be2   4 hours ago   61.8MB
nginx          alpine    4a86014ec699   2 weeks ago   52.5MB
root@ip-172-31-27-104:~# docker tag trendstore:latest anand20003/project2:latest
root@ip-172-31-27-104:~# docker push anand20003/project2:latest
The push refers to repository [docker.io/anand20003/project2]
8f25d744f156: Pushed
f9985d3fc94d: Mounted from library/nginx
d208138be39d: Mounted from library/nginx
a2b76470e8f1: Mounted from library/nginx
917b2c97271e: Mounted from library/nginx
16ca725632e5: Mounted from library/nginx
7978a9c91f72: Mounted from library/nginx
b6ff0212304e: Mounted from library/nginx
418dccbd85a: Mounted from library/nginx
latest: digest: sha256:751549364c2bcf6b8502bed7584f3007532f5bfbb0e97ef82ec045f1bc2bf7ef size: 2200
root@ip-172-31-27-104:~# █
```

4.Kubernetes

1.EKSctl and Kubectl Installed:

```
root@ip-172-31-27-104:~# eksctl version && kubectl version --client
0.214.0
Client Version: v1.34.0
Kustomize Version: v5.7.1
root@ip-172-31-27-104:~# █
```

2.Creating EKS Cluster:

```
root@ip-172-31-27-104:~# eksctl create cluster --name myapp-cluster --region us-east-1 --nodegroup-name my-nodes --node-type t3.medium --nodes 1 --nodes-min 1 --nodes-max 3 --managed
2025-08-31 14:31:26 [i]  eksctl version 0.214.0
2025-08-31 14:31:26 [i]  using region us-east-1
2025-08-31 14:31:26 [i]  setting availability zones to (us-east-1c us-east-1b)
2025-08-31 14:31:26 [i]  subnets for us-east-1c - public:192.168.0.0/19 private:192.168.64.0/19
2025-08-31 14:31:26 [i]  subnets for us-east-1b - public:192.168.32.0/19 private:192.168.96.0/19
2025-08-31 14:31:26 [i]  nodegroup "my-nodes" will use "" [AmazonLinux2023/1.32]
2025-08-31 14:31:26 [i]  using Kubernetes version 1.32
2025-08-31 14:31:26 [i]  creating EKS cluster "myapp-cluster" in "us-east-1" region with managed nodes
2025-08-31 14:31:26 [i]  will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-08-31 14:31:26 [i]  if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=myapp-cluster'
2025-08-31 14:31:26 [i]  Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "myapp-cluster" in "us-east-1"
2025-08-31 14:31:26 [i]  CloudWatch logging will not be enabled for cluster "myapp-cluster" in "us-east-1"
2025-08-31 14:31:26 [i]  you can enable it with 'eksctl utils update-cluster-logging --enable-types=(SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)) --region=us-east-1 --cluster=myapp-cluster'
2025-08-31 14:31:26 [i]  default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-08-31 14:31:26 [i]  2 sequential tasks: ( create cluster control plane "myapp-cluster",
  2 sequential sub-tasks:
    2 sequential sub-tasks:
      1 task: ( create addons ),
      wait for control plane to become ready,
    ),
    create managed nodegroup "my-nodes",
  )
}
```

3.Cluster Created:

```
root@ip-172-31-27-104:~# kubectl get nodes
NAME                  STATUS   ROLES      AGE     VERSION
ip-192-168-39-88.ec2.internal   Ready    <none>   33m    v1.32.7-eks-3abbecl
root@ip-172-31-27-104:~# eksctl get cluster
NAME        REGION      EKSCTL CREATED
myapp-cluster us-east-1  True
root@ip-172-31-27-104:~#
```

4.Deployment.yaml file:

```
GNU nano 7.2                                     deployment.yaml *
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp-deployment
  labels:
    app: myapp
spec:
  replicas: 2
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
        - name: myapp
          image: anand20003/project2:latest
          ports:
            - containerPort: 80
```

5.Service.yaml File:

```
GNU nano 7.2                                     service.yaml *
apiVersion: v1
kind: Service
metadata:
  name: myapp-service
spec:
  type: LoadBalancer
  selector:
    app: myapp
  ports:
    - port: 80
      targetPort: 80
```

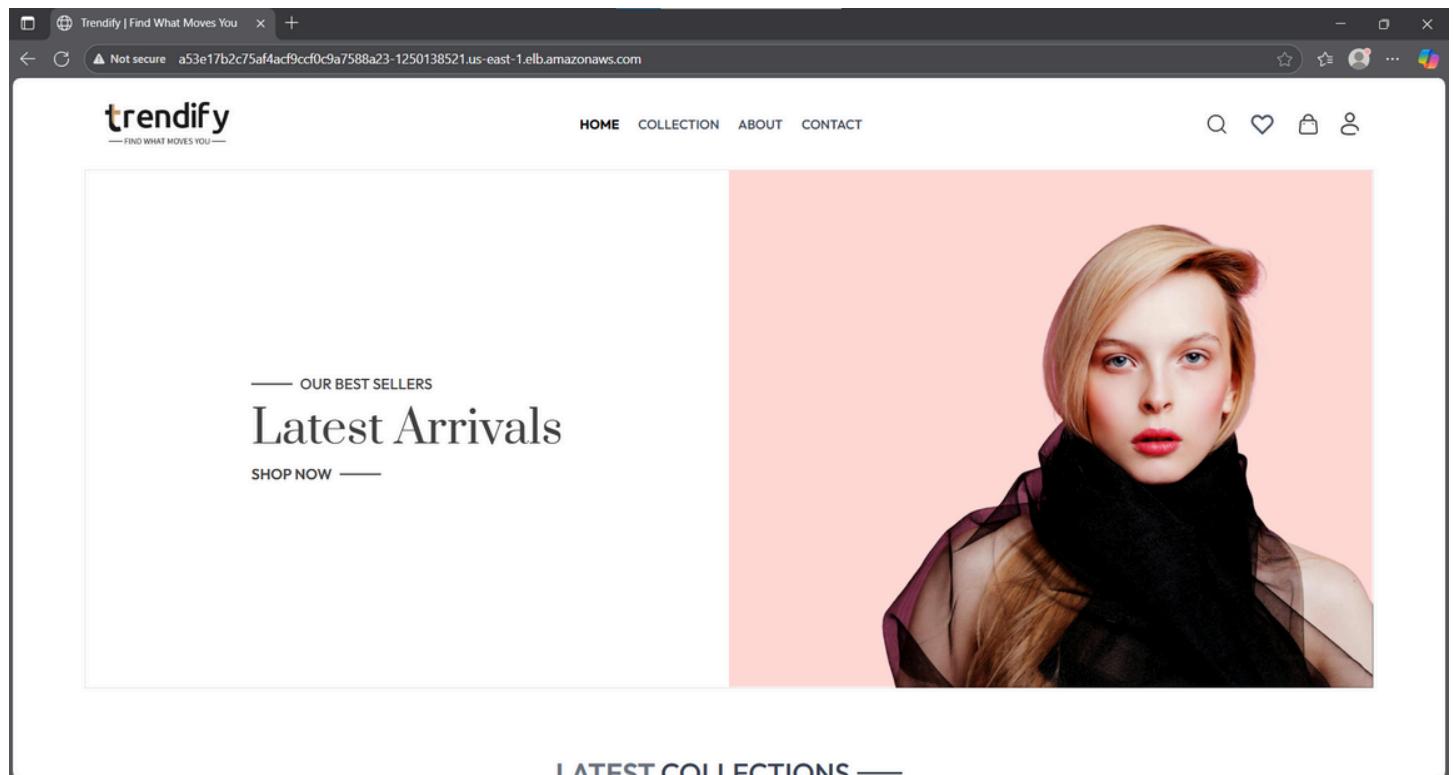
6.Deployment,Service and Pods Created:

```

root@ip-172-31-27-104:~# kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
myapp-deployment   2/2     2          2          99s
root@ip-172-31-27-104:~#
root@ip-172-31-27-104:~# kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
myapp-deployment-77cff48886-h2hv4   1/1     Running   0          117s
myapp-deployment-77cff48886-ltf2j   1/1     Running   0          117s
root@ip-172-31-27-104:~#
root@ip-172-31-27-104:~# kubectl get svc
NAME            TYPE      CLUSTER-IP   EXTERNAL-IP
kubernetes      ClusterIP  10.100.0.1   <none>
myapp-service   LoadBalancer  10.100.40.144  a53e17b2c75af4acf9ccf0c9a7588a23-1250138521.us-east-1.elb.amazonaws.com
root@ip-172-31-27-104:~# █

```

7.Output 1:



8.Output 2:

The screenshot shows a product page for a men's round-neck t-shirt. At the top, the Trendify logo and navigation links (HOME, COLLECTION, ABOUT, CONTACT) are visible. A search bar and user icons are on the right. The main image features a male model wearing a black t-shirt with a Puma logo graphic. To the left is a smaller inset image of the same t-shirt. To the right, the product title "Men Round Neck Pure Cotton T-shirt" is displayed with a 5-star rating. The price is listed as \$110. A description notes it's a lightweight, close-fitting pullover shirt. Below the price are size selection buttons for S, M, and XXL, followed by an "ADD TO CART" button. A promotional banner at the bottom offers 100% authenticity, cash on delivery, and hassle-free returns.

8.Output 3:

The screenshot shows the "ABOUT US" page of the Trendify website. The header includes the Trendify logo and navigation links. The main section is titled "ABOUT US —". On the left is a photograph of various fashion items: blue jeans, a white cable-knit sweater, a white mug with "TRENDIFY" written on it, a small brown bag, and two brown boots. To the right, the "ABOUT US" section begins with a paragraph about the company's mission to bring the latest fashion trends to customers. It then discusses the team's dedication to providing a seamless shopping experience and staying updated with runway trends. Below this is a "Our Mission" section, which states the company's goal of empowering customers through high-quality, on-trend fashion. The "Our Vision" section follows, stating the company's aim to be a global leader known for cutting-edge style and quality, inspiring confidence and creativity.

10.Deleting EKS Cluster:

```
root@ip-172-31-27-104:~# eksctl delete cluster --name myapp-cluster --region us-east-1
2025-08-31 16:08:05 [i] deleting EKS cluster "myapp-cluster"
2025-08-31 16:08:05 [i] will drain 0 unmanaged nodegroup(s) in cluster "myapp-cluster"
2025-08-31 16:08:05 [i] starting parallel draining, max in-flight of 1
2025-08-31 16:08:06 [i] deleted 0 Fargate profile(s)
2025-08-31 16:08:06 [✓] kubeconfig has been updated
2025-08-31 16:08:06 [i] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2025-08-31 16:08:07 [i] 2 sequential tasks: { delete nodegroup "my-nodes", delete cluster control plane "myapp-cluster" [async]
}
2025-08-31 16:08:07 [i] will delete stack "eksctl-myapp-cluster-nodegroup-my-nodes"
2025-08-31 16:08:07 [i] waiting for stack "eksctl-myapp-cluster-nodegroup-my-nodes" to get deleted
2025-08-31 16:08:07 [i] waiting for CloudFormation stack "eksctl-myapp-cluster-nodegroup-my-nodes"
```

5.Version Control

1.Creating GitHub Repository:

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

1 General

Owner * Repository name *

Anand-kumar-git / Project2

Project2 is available.

Great repository names are short and memorable. How about [miniature-succotash](#)?

Description

0 / 350 characters

2 Configuration

Choose visibility *

Choose who can see and commit to this repository

Public

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

Off

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

2..dockerignorefile:

```
◆ .dockerignore
1 node_modules
2 .git
3 .gitignore
4 Dockerfile
5 *.md
6 *.log
7
```

3..gitignore file:

```
◆ .gitignore
1 node_modules/
2 dist/
3 .env
4 *.log
5
```

6.Jenkins

1.Docker, EKSctl, Kubectl, AWScli Installed in Jenkins Server:

```
root@ip-10-0-1-223:~# apt docker --version && aws --version && eksctl version && kubectl version --client
apt 2.8.3 (amd64)
aws-cli/2.28.21 Python/3.13.7 Linux/6.14.0-1011-aws exe/x86_64.ubuntu.24
0.214.0
Client Version: v1.34.0
Kustomize Version: v5.7.1
root@ip-10-0-1-223:~#
```

2.Jenkins Installed:

```
root@ip-10-0-1-223:~# systemctl status jenkins
● jenkins.service - Jenkins continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
  Active: active (running) since Sun 2025-08-31 16:48:13 UTC; 13min ago
    Main PID: 11114 (java)
      Tasks: 38 (limit: 1121)
     Memory: 297.1M (peak: 330.7M)
        CPU: 19.453s
       CGroup: /system.slice/jenkins.service
               └─11114 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Aug 31 16:48:05 ip-10-0-1-223 jenkins[11114]: 468f40bd43f741508d50f12bdedb1e86
Aug 31 16:48:05 ip-10-0-1-223 jenkins[11114]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Aug 31 16:48:05 ip-10-0-1-223 jenkins[11114]: ****
Aug 31 16:48:05 ip-10-0-1-223 jenkins[11114]: 2025-08-31 16:48:13.056+0000 [id=31]      INFO      jenkins.InitReactorRunner$1#onAttained: Completed initialization
Aug 31 16:48:13 ip-10-0-1-223 jenkins[11114]: 2025-08-31 16:48:13.089+0000 [id=23]      INFO      hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
Aug 31 16:48:13 ip-10-0-1-223 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Aug 31 16:48:13 ip-10-0-1-223 jenkins[11114]: 2025-08-31 16:48:13.401+0000 [id=47]      INFO      h.m.DownloadService$Downloadable#load: Obtained the updated data
Aug 31 16:48:13 ip-10-0-1-223 jenkins[11114]: 2025-08-31 16:48:13.402+0000 [id=47]      INFO      hudson.util.Retriger#start: Performed the action check updates se>
lines 1-20/20 (END)
```

3.Enabling Programmatic Access To Jenkins Server:

```
root@ip-10-0-1-223:~# aws configure
AWS Access Key ID [None]: AKIA3O2O4QWYFPSJ4R5Y
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
root@ip-10-0-1-223:~#
```

4.Adding Jenkins to Docker Group:

```
root@ip-10-0-1-223:~# usermod -aG docker jenkins
root@ip-10-0-1-223:~#
```

5.Creating EKS Cluster in Jenkins-server:

```

root@ip-10-0-1-223:~# eksctl create cluster --name my-cluster --region us-east-1 --nodegroup-name my-nodes --node-type t3.medium --nodes 1 --nodes-min 1 --nodes-max 3 --managed

2025-09-01 05:28:02 [i] eksctl version 0.214.0
2025-09-01 05:28:02 [i] using region us-east-1
2025-09-01 05:28:02 [i] setting availability zones to [us-east-1c us-east-1a]
2025-09-01 05:28:02 [i] subnets for us-east-1c - public:192.168.0.0/19 private:192.168.64.0/19
2025-09-01 05:28:02 [i] subnets for us-east-1a - public:192.168.32.0/19 private:192.168.96.0/19
2025-09-01 05:28:02 [i] nodegroup "my-nodes" will use "" (AmazonLinux2023/1.32)
2025-09-01 05:28:02 [i] using Kubernetes version 1.32
2025-09-01 05:28:02 [i] creating EKS cluster "my-cluster" in "us-east-1" region with managed nodes
2025-09-01 05:28:02 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-09-01 05:28:02 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=my-cluster'
2025-09-01 05:28:02 [i] Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "my-cluster" in "us-east-1"
2025-09-01 05:28:02 [i] CloudWatch logging will not be enabled for cluster "my-cluster" in "us-east-1"
2025-09-01 05:28:02 [i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE} (e.g. all) --region=us-east-1 --cluster=my-cluster'
2025-09-01 05:28:02 [i] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-09-01 05:28:02 [i]

2 sequential tasks: { create cluster control plane "my-cluster",
  2 sequential sub-tasks: {
    2 sequential sub-tasks: {
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "my-nodes",
  }
}

```

6.Creating User In Jenkins:

Getting Started

Create First Admin User

Username
Anand

Password
.....

Confirm password
.....

Full name
Anand

E-mail address
.....

Jenkins 2.525

Skip and continue as admin

Save and Continue

7.Installing Required Plugins in Jenkins:

Jenkins / Manage Jenkins / Plugins

Plugins

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

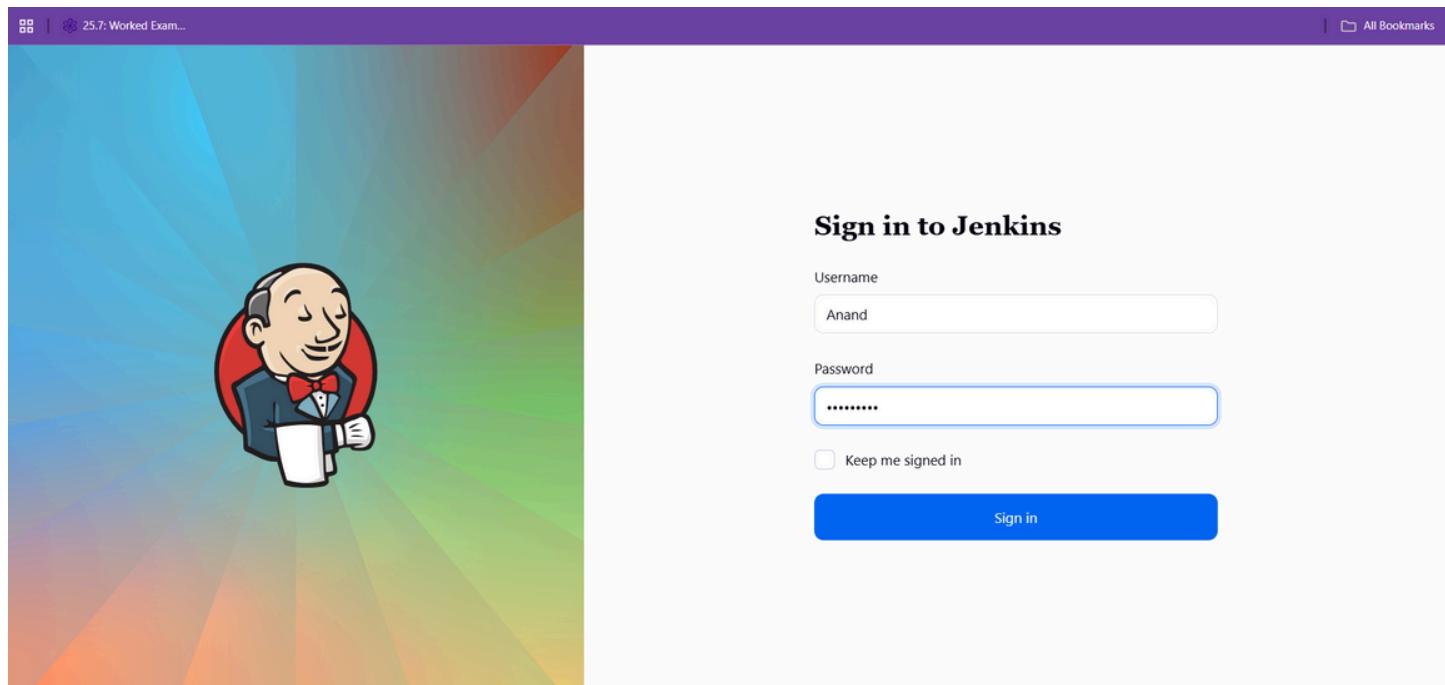
Plugin	Status
SSH Build Agents	Success
Matrix Authorization Strategy	Success
LDAP	Success
jsoup API	Success
Email Extension	Success
Mailer	Success
Theme Manager	Success
Dark Theme	Success
Loading plugin extensions	Success
Kubernetes Client API	Success
Authentication Tokens API	Success
Kubernetes Credentials	Success
Kubernetes	Success
Loading plugin extensions	Success

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

→ Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.525

8.Jenkins Login Page:



9.Adding Dockerhub Credential in Jenkins:

25.7: Worked Exam... All Bookmarks

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... /

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

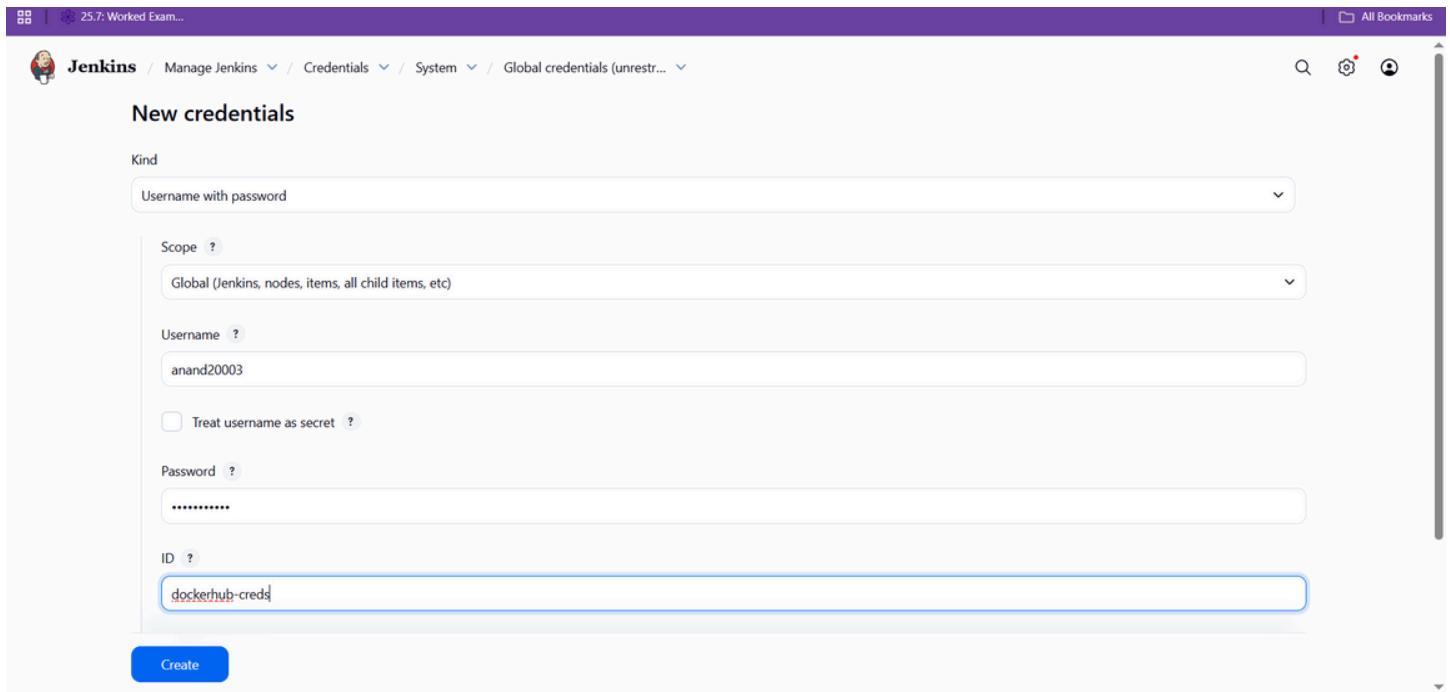
Username: anand20003

Treat username as secret

Password:

ID: dockerhub-creds

Create



10.Adding AWS Configure in Jenkins:

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... / AKIA3O2O4QWYFPSJ4R5Y /

Update credentials

Update Delete Move

Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: aws-creds

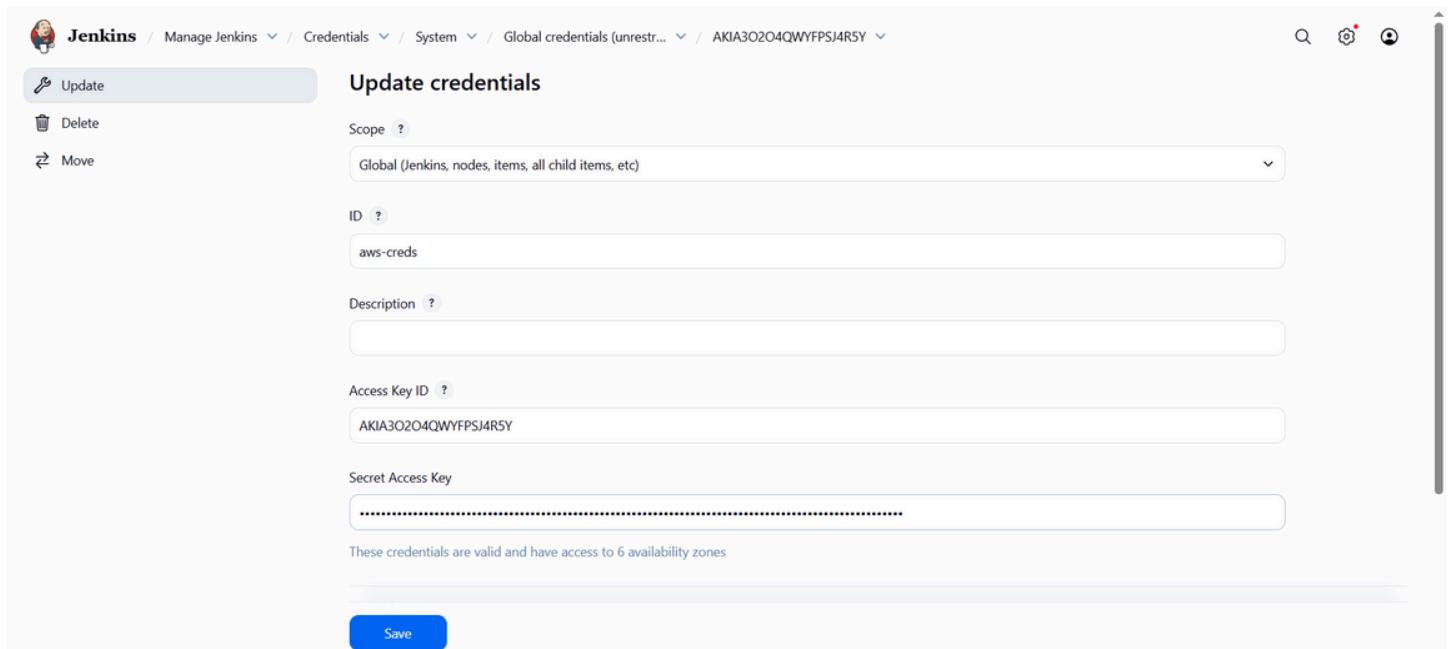
Description:

Access Key ID: AKIA3O2O4QWYFPSJ4R5Y

Secret Access Key:

These credentials are valid and have access to 6 availability zones

Save



11.Adding GitHub Webhook:

The screenshot shows the GitHub settings interface for a repository named 'Project2'. On the left, there's a sidebar with sections like General, Access, Collaborators, and Code and automation (Branches, Tags, Rules, Actions, Models, Webhooks). The 'Webhooks' section is currently selected. The main area is titled 'Webhooks / Add webhook' and contains fields for 'Payload URL' (set to 'http://3.231.146.7:8080/github-webhook/'), 'Content type' (set to 'application/json'), and a 'Secret' field. There's also a 'SSL verification' section with options for 'Enable SSL verification' (selected) and 'Disable (not recommended)'. At the bottom, it asks 'Which events would you like to trigger this webhook?' with the option 'Just the push event.' selected.

12. Enabling GitHub Webhook in Jenkins:

The screenshot shows the Jenkins 'GitHub Server' configuration page. It has fields for 'Name' (empty), 'API URL' (set to 'https://api.github.com'), and 'Credentials' (set to '- none -'). There's a checkbox for 'Manage hooks' which is checked. At the bottom, there are 'Save' and 'Apply' buttons, and a 'Test connection' button on the right.

13. Creating Jenkins Pipeline Job:



New Item

Enter an item name

Trend_Store

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.

OK

14.Pipeline Configure:

**Configure**

General

Triggers

Pipeline

Advanced

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?**Repository URL ?**

https://github.com/Anand-kumar-git/Project2.git

Credentials ?

- none -

Save**Apply**

15.Jenkinsfile First Part:

```

1 Jenkinsfile M X ! deployment.yaml ! service.yaml Dockerfile .gitignore .dockerignore
2 Jenkinsfile
3
4 pipeline {
5     agent any
6
7     environment {
8         DOCKER_REPO = "anand20003/project2"
9         AWS_REGION = "us-east-1"
10        IMAGE_TAG = "${env.BUILD_NUMBER}"
11    }
12
13    stages {
14        stage('Checkout Code') {
15            steps {
16                git branch: 'main', url: 'https://github.com/Anand-kumar-git/Project2.git'
17            }
18        }
19
20        stage('Build Docker Image') {
21            steps {
22                sh """
23                    echo "Building Docker image..."
24                    docker build -t ${DOCKER_REPO}:${IMAGE_TAG} .
25                    docker tag ${DOCKER_REPO}:${IMAGE_TAG} ${DOCKER_REPO}:latest
26                """
27            }
28        }
29
30        stage('Push Docker Image') {
31            steps {
32                withCredentials([usernamePassword(
33                    credentialsId: 'dockerhub-creds',
34                    usernameVariable: 'DOCKER_USER',
35                    passwordVariable: 'DOCKER_PASS'
36                )]) {
37                    sh """
38                        echo "$DOCKER_PASS" | docker login -u "$DOCKER_USER" --password-stdin
39                        docker push ${DOCKER_REPO}:${IMAGE_TAG}
40                        docker push ${DOCKER_REPO}:latest
41                    """
42                }
43            }
44        }
45    }
46
47    post {
48        success {
49            echo "Deployment successful!"
50        }
51        failure {
52            echo "Deployment failed. Check logs in Jenkins!"
53        }
54    }
55 }
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

```

16.Jenkinsfile Second Part:

```

39
40
41
42
43 stage('Deploy to EKS') {
44     steps {
45         withCredentials([$class: 'AmazonWebServicesCredentialsBinding',
46                         | credentialsId: 'aws-creds']) {
47             sh """
48                 echo "Updating kubeconfig..."
49                 aws eks update-kubeconfig --region ${AWS_REGION} --name my-cluster
50
51                 echo "Deploying to Kubernetes..."
52                 kubectl set image deployment/myapp-deployment myapp=${DOCKER_REPO}:${IMAGE_TAG} || true
53
54                 # Apply Kubernetes manifests from k8s folder
55                 kubectl apply -f k8s/deployment.yaml
56                 kubectl apply -f k8s/service.yaml
57             """
58         }
59     }
60 }
61
62
63 post {
64     success {
65         echo "Deployment successful!"
66     }
67     failure {
68         echo "Deployment failed. Check logs in Jenkins!"
69     }
70 }
71

```

17.Dockerfile:

```
🐳 Dockerfile
1  FROM nginx:alpine
2
3  COPY dist/ /usr/share/nginx/html
4
5  EXPOSE 80
6
7  CMD ["nginx", "-g", "daemon off;"]
```

18.Pushing to the Github:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

ADMIN@DESKTOP-0E234HV MINGW64 ~/Project2 (main)
● $ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 286 bytes | 143.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Anand-kumar-git/Project2.git
  ae31b8d..b81d1a1  main -> main

ADMIN@DESKTOP-0E234HV MINGW64 ~/Project2 (main)
$ 
○
ADMIN@DESKTOP-0E234HV MINGW64 ~/Project2 (main)
○ $ 
```

19.Pipeline Build:

Jenkins / Trend_Store

Status

</> Changes

Build Now

Configure

Delete Pipeline

Stages

Rename

Pipeline Syntax

GitHub Hook Log

Trend_Store

Permalinks

- Last build (#1), 45 sec ago
- Last stable build (#1), 45 sec ago
- Last successful build (#1), 45 sec ago
- Last completed build (#1), 45 sec ago

Builds

Filter

Today

#1 6:43 PM

REST API Jenkins 2.525

20. Deployment, Service, Pods are Created through Pipeline:

```
root@ip-10-0-1-223:~# kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
myapp-deployment  2/2     2           2          6h14m
root@ip-10-0-1-223:~#
root@ip-10-0-1-223:~# kubectl get svc
NAME        TYPE      CLUSTER-IP      EXTERNAL-IP
kubernetes  ClusterIP  10.100.0.1    <none>
myapp-service  LoadBalancer  10.100.252.14  ad09e73d70149400ba2ff85ec4b4477c-607758845.us-east-1.elb.amazonaws.com  80:31108/TCP  6h14m
root@ip-10-0-1-223:~# kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
myapp-deployment-68bd9cbd4c-8bqw4  1/1     Running   0          12m
myapp-deployment-68bd9cbd4c-rnmm5  1/1     Running   0          12m
root@ip-10-0-1-223:~#
```

21. Output 1:

Trendify | Find What Moves You

Not secure ad09e73d70149400ba2ff85ec4b4477c-60775845.us-east-1.elb.amazonaws.com

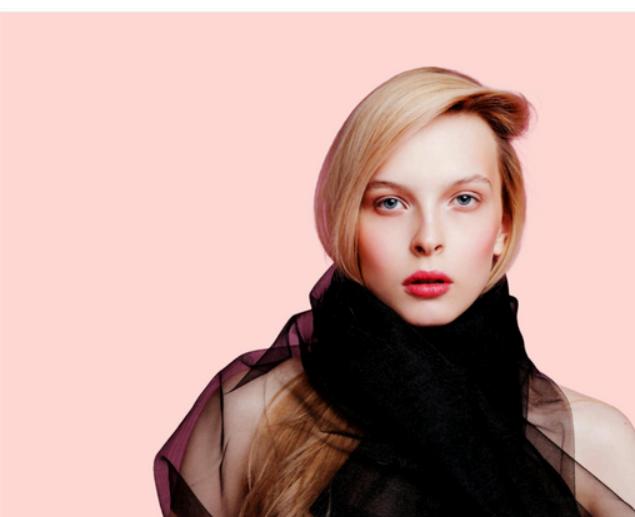
trendify FIND WHAT MOVES YOU

HOME COLLECTION ABOUT CONTACT

OUR BEST SELLERS

Latest Arrivals

SHOP NOW



LATEST COLLECTIONS

22.Output 2:

Trendify | Find What Moves You

Not secure ad09e73d70149400ba2ff85ec4b4477c-60775845.us-east-1.elb.amazonaws.com/product/683413d7245102641d71aeaf

trendify FIND WHAT MOVES YOU

HOME COLLECTION ABOUT CONTACT



Men Slim Fit Relaxed Denim Jacket

★★★★★ 0 (0)

\$230

A lightweight, usually knitted, pullover shirt, close-fitting and with a round neckline and short sleeves, worn as an undershirt or outer garment.

Select Size

S M L XL

ADD TO CART

Guaranteed 100% Authentic – Shop with Confidence!
Enjoy Cash on Delivery – Pay at Your Doorstep!
Hassle-Free Returns & Exchanges – 10 Days, No Questions Asked!

23.Output 3:

Welcome to Trendify, where style meets quality. Our mission is to bring you the latest fashion trends and must-have items, all curated with an eye for quality and design. We believe that everyone deserves to express themselves through fashion, and we're here to make that easier and more enjoyable. Our collections are carefully selected to offer you a range of options that cater to every taste and occasion.

At Trendify, we prioritize your satisfaction. From the moment you browse our site to the day your order arrives, we are dedicated to providing a seamless shopping experience. Our team is always on the lookout for the latest trends, ensuring that you have access to the freshest styles as soon as they hit the runway. Thank you for choosing Trendify. We're excited to be a part of your style journey.

Our Mission

At Trendify, our mission is to empower you to express your unique style with high-quality, on-trend fashion. We strive to make fashion accessible to all, offering diverse products that inspire confidence.

Our Vision

At Trendify, our vision is to be a global fashion leader, known for cutting-edge style and quality. We aim to inspire confidence and creativity, making Trendify the go-to choice for individual

24.Output 4:

danandkumar2003@gmail.com

.....

Forgot your password? [Create a new account](#)

[Sign In](#)

COMPANY

[Home](#)
[About Us](#)
[Delivery](#)
[Privacy & Policy](#)

GET IN TOUCH

+11-558-669-447
contact.trendify@info.com

7.Monitoring

1. Installing Helm:

```
root@ip-172-31-19-78:~# curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
% Total    % Received % Xferd  Average Speed   Time     Time   Current
                                         Dload  Upload Total Spent   Left  Speed
100 11913  100 11913    0     0  331k      0  --:--:-- --:--:-- --:--:--  332k
Downloaded https://get.helm.sh/helm-v3.18.6-linux-amd64.tar.gz
Verifying checksum... Done.
Preparing to install helm into /usr/local/bin
helm installed into /usr/local/bin/helm
root@ip-172-31-19-78:~#
```

2. Installing Prometheus:

```
root@ip-172-31-19-78:~# helm install prometheus prometheus-community/kube-prometheus-stack -n monitoring --create-namespace
```

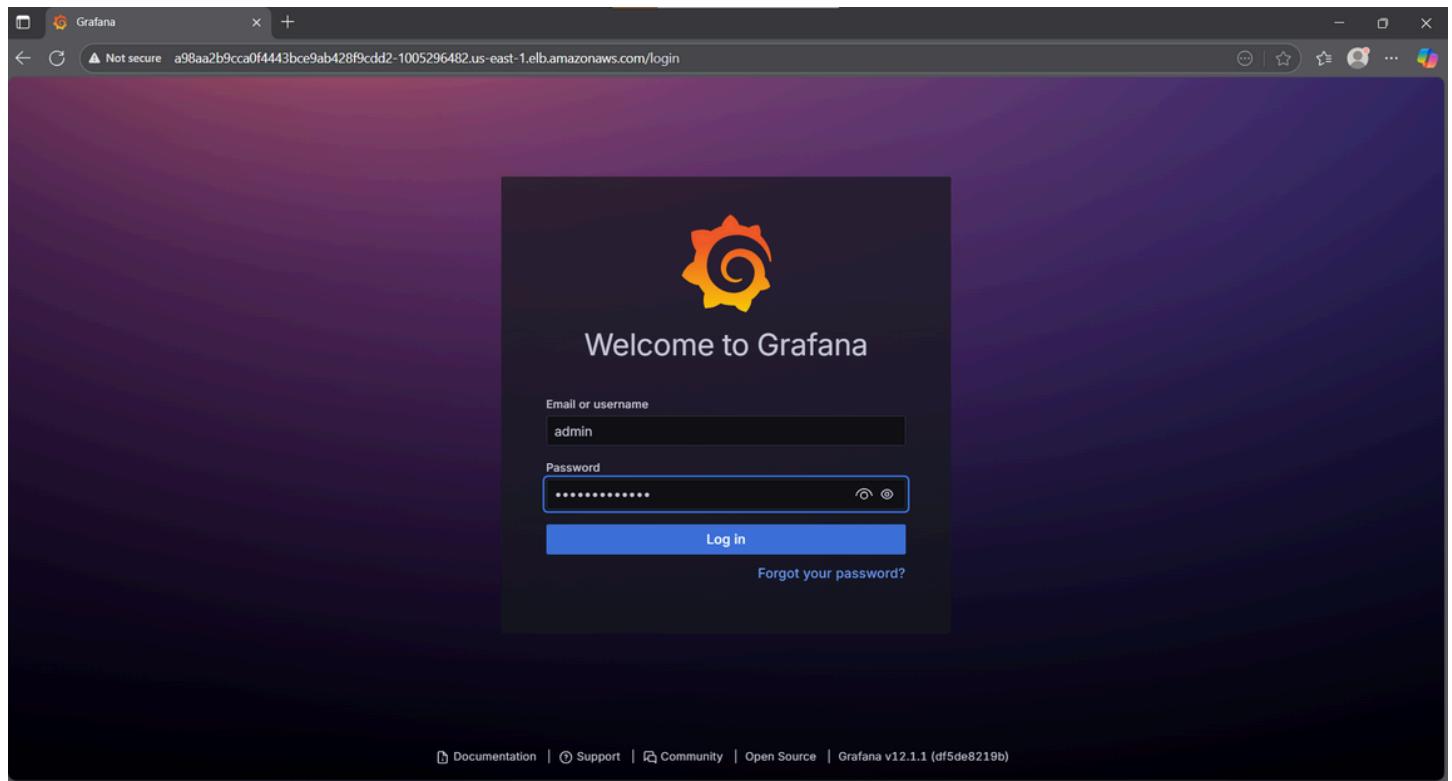
3. Creating Monitoring Namespace:

```
root@ip-10-0-1-223:~# kubectl create namespace monitoring
helm install monitoring prometheus-community/kube-prometheus-stack -n monitoring
namespace/monitoring created
NAME: monitoring
LAST DEPLOYED: Mon Sep  1 20:08:22 2025
NAMESPACE: monitoring
STATUS: deployed
REVISION: 1
NOTES:
kube-prometheus-stack has been installed. Check its status by running:
  kubectl --namespace monitoring get pods -l "release=monitoring"

Get Grafana 'admin' user password by running:
  kubectl --namespace monitoring get secrets monitoring-grafana -o jsonpath=".data.admin-password" | base64 -d ; echo

Access Grafana local instance:
  export POD_NAME=$(kubectl --namespace monitoring get pod -l "app.kubernetes.io/name=grafana,app.kubernetes.io/instance=monitoring" -oname)
  kubectl --namespace monitoring port-forward $POD_NAME 3000
Visit https://github.com/prometheus-operator/kube-prometheus for instructions on how to create & configure Alertmanager and Prometheus instances using the Operator.
root@ip-10-0-1-223:~#
```

4. Login To Grafana in Browser:



5.Adding Prometheus Datasource in Grafana:

A screenshot of a web browser showing the "Data sources" configuration page in Grafana. The title bar says "prometheus-1 - Data sources". The address bar shows a URL starting with "a98aa2b9cca0f4443bce9ab428f9cdd2-1005296482.us-east-1.elb.amazonaws.com/connections/datasources/edit/aewsoj6y3oirkc". The left sidebar has a "Data sources" section highlighted with an orange bar. The main content area shows a "Connection" section with a "Prometheus server URL" input field containing "http://monitoring-kube-prometheus-prometheus.mon". Below it is an "Authentication" section with a dropdown menu set to "No Authentication".

6.Importing Dashboard:

Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from [Grafana.com](#)

Published by rfmoz

Updated on 2025-06-13 01:36:17

Options

Name

Node Exporter Full

Folder

Dashboards

Unique identifier (UID)

The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

rYdddIPWk

[Change uid](#)

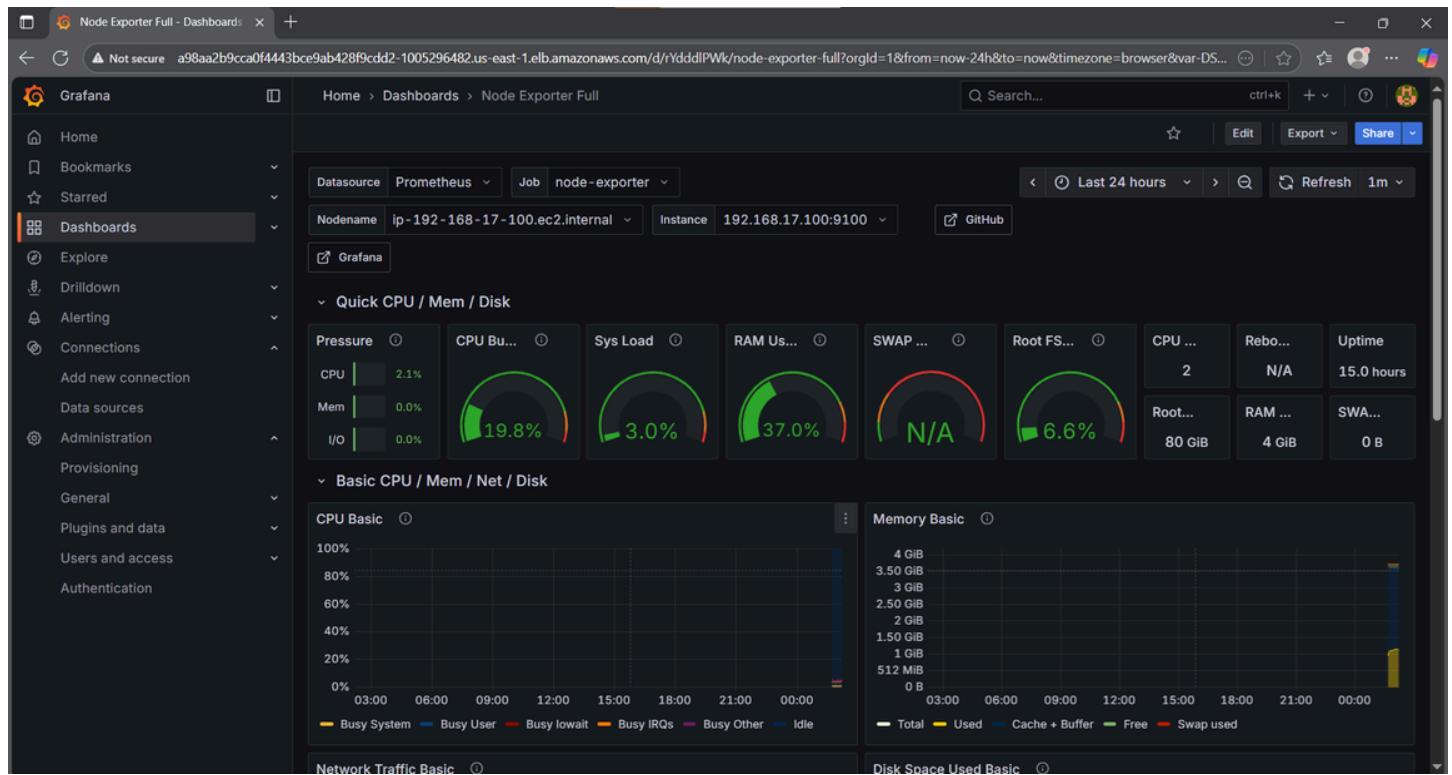
prometheus

 Prometheus

[Import](#)

[Cancel](#)

7. Monitoring Dashboard:



8.CPU,Memory Metrics:

