

Name: Prashanth Raghavendra Rao

Task: Mini Project – Trend Store

Date: 30/08/2025

Task Description:

Clone the below mentioned repository and deploy the application (Run application in port 3000).

Repo URL : <https://github.com/Vennilavan12/Trend.git>

NPM LOCAL RUN ON PORT 3000

```
dist - node ~/nvm/versions/node/v21.6.1/bin/serve — 197x64
(base) prashantr@Prashanths-MacBook-Pro-2 TrendApp %
(base) prashantr@Prashanths-MacBook-Pro-2 TrendApp %
(base) prashantr@Prashanths-MacBook-Pro-2 TrendApp % git clone git@github.com:Vennilavan12/Trend.git
Cloning into 'Trend'...
Enter passphrase for key '/Users/prashantr/.ssh/id_ed25519':
remote: Enumerating objects: 77, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 77 (delta 0), reused 0 (delta 0), pack-reused 76 (from 1)
Receiving objects: 100% (77/77), 8.58 MiB | 4.21 MiB/s, done.
Resolving deltas: 100% (1/1), done.
(base) prashantr@Prashanths-MacBook-Pro-2 TrendApp % ls
Trend
(base) prashantr@Prashanths-MacBook-Pro-2 TrendApp % cd Trend
(base) prashantr@Prashanths-MacBook-Pro-2 Trend % ls -l
total 0
drwxr-xr-x 5 prashantr staff 160 Aug 30 12:27:27
(base) prashantr@Prashanths-MacBook-Pro-2 Trend % ls
dist
(base) prashantr@Prashanths-MacBook-Pro-2 Trend %
(base) prashantr@Prashanths-MacBook-Pro-2 Trend % cd dist
(base) prashantr@Prashanths-MacBook-Pro-2 dist % npm install -g serve
changed 88 packages in 3s

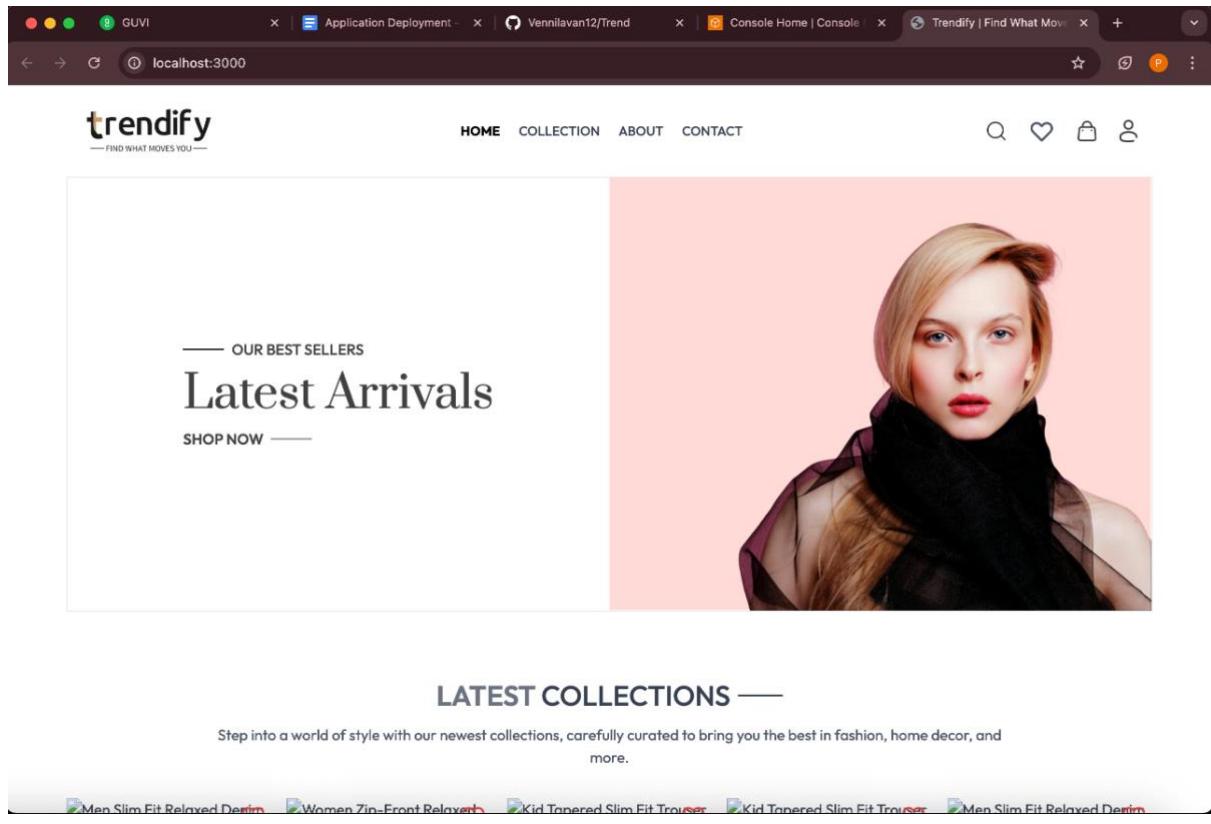
24 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New major version of npm available! 10.4.0 → 11.5.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.5.2
npm notice Run `npm install -g npm@11.5.2` to update!
npm notice
(base) prashantr@Prashanths-MacBook-Pro-2 dist % serve

Serving!
- Local:  http://localhost:3000
- Network: http://192.168.68.102:3000

Copied local address to clipboard!
```

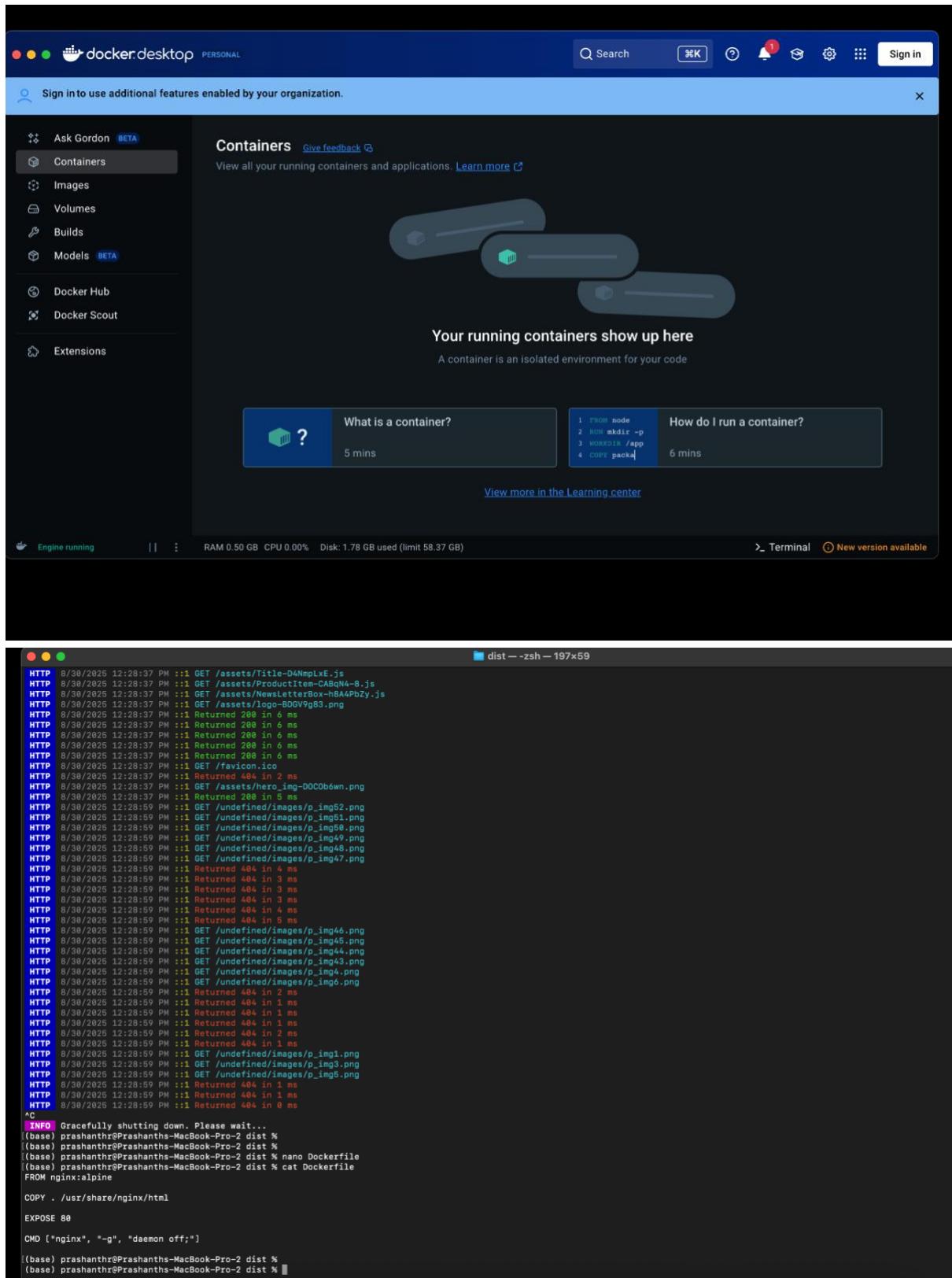


```
HTTP 8/30/2025 12:28:37 PM ::1 GET /
HTTP 8/30/2025 12:28:37 PM ::1 Returned 200 in 14 ms
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/index-BjDoubly.js
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/index-BG2g6lW.css
HTTP 8/30/2025 12:28:37 PM ::1 Returned 200 in 2 ms
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/icon-404.png
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/Home-BjlxAQU1.js
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/Title-D4Nmqlxe.js
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/ProductItem-CABqN4-8.js
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/NewsLetterBox-n8AA4PbZy.js
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/logo-BDGVgB83.png
HTTP 8/30/2025 12:28:37 PM ::1 Returned 200 in 6 ms
HTTP 8/30/2025 12:28:37 PM ::1 Returned 200 in 6 ms
HTTP 8/30/2025 12:28:37 PM ::1 Returned 200 in 6 ms
HTTP 8/30/2025 12:28:37 PM ::1 Returned 200 in 6 ms
HTTP 8/30/2025 12:28:37 PM ::1 GET /favicon.ico
HTTP 8/30/2025 12:28:37 PM ::1 Returned 404 in 2 ms
HTTP 8/30/2025 12:28:37 PM ::1 GET /assets/hero_img-DOOCObwn.png
HTTP 8/30/2025 12:28:37 PM ::1 Returned 200 in 5 ms
```



Docker:

- Dockerize the application by creating Dockerfile
- Build an application and check output using docker image.

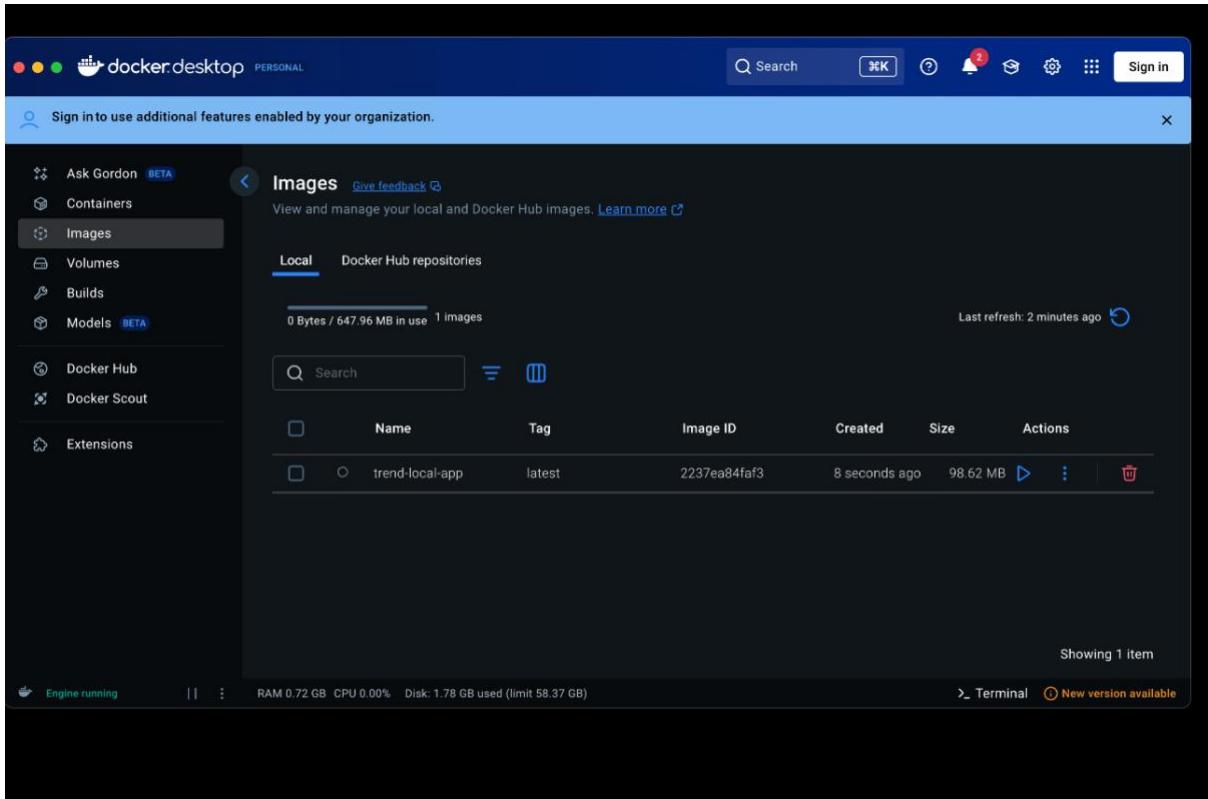


The screenshot shows the Docker Desktop application running on a Mac. The main interface has a dark theme with blue highlights. On the left, a sidebar lists various sections: Ask Gordon (Beta), Containers (selected), Images, Volumes, Builds, Models (Beta), Docker Hub, Docker Scout, and Extensions. A central panel titled "Containers" displays a placeholder message: "Your running containers show up here" with the subtext "A container is an isolated environment for your code". Below this are two cards: "What is a container?" (5 mins) and "How do I run a container?" (6 mins). At the bottom of the sidebar, status information is shown: "Engine running", "RAM 0.50 GB CPU 0.00%", "Disk: 1.78 GB used (limit 58.37 GB)", and "Terminal" with a note "New version available". The main content area is a terminal window titled "dist -- zsh -- 197x59" showing a log of HTTP requests and Docker commands. The log includes entries like "HTTP 8/38/2025 12:28:37 PM ::1 GET /assets>Title-D4NmpxE.js", "HTTP 8/38/2025 12:28:37 PM ::1 GET /assets/ProductItem-CABeN4-B.js", and "HTTP 8/38/2025 12:28:37 PM ::1 GET /assets/NewsletterBox-hdA4PbZy.js". It also shows Docker commands such as "FROM node", "RUN mkdir -p", "WORKDIR /app", and "COPY packa". The terminal ends with "INFO Gracefully shutting down. Please wait..." and a series of "Returned 404" errors followed by "COPY . /usr/share/nginx/html", "EXPOSE 80", and "CMD ["nginx", "-g", "daemon off;"]". The Dockerfile at the bottom is shown as:

```

FROM nginx:alpine
COPY . /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]

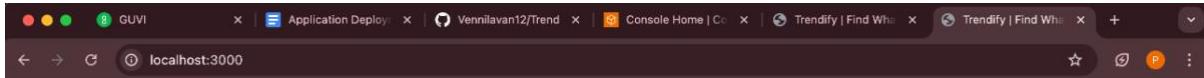
```



```

(base) prashanthr@Prashanths-MacBook-Pro-2 dist %
(base) prashanthr@Prashanths-MacBook-Pro-2 dist % docker images
REPOSITORY          TAG      IMAGE ID      CREATED             SIZE
trend-local-app    latest   2237ea84faf3   26 minutes ago   98.6MB
(base) prashanthr@Prashanths-MacBook-Pro-2 dist %
(base) prashanthr@Prashanths-MacBook-Pro-2 dist % docker ps
CONTAINER ID        IMAGE     COMMAND       CREATED          STATUS    PORTS     NAMES
(base) prashanthr@Prashanths-MacBook-Pro-2 dist %
(base) prashanthr@Prashanths-MacBook-Pro-2 dist % docker run -d -p 3000:80 trend-local-app
7466fcfe045ad7cfab8400b7218e41ab66c0ff5f07f788azc851e25b8ed8d140
(base) prashanthr@Prashanths-MacBook-Pro-2 dist %
(base) prashanthr@Prashanths-MacBook-Pro-2 dist % docker ps
CONTAINER ID        IMAGE     COMMAND       CREATED          STATUS    PORTS     NAMES
7466fcfe045ad7cfab8400b7218e41ab66c0ff5f07f788azc851e25b8ed8d140
(base) prashanthr@Prashanths-MacBook-Pro-2 dist %
(base) prashanthr@Prashanths-MacBook-Pro-2 dist %

```



trendify
FIND WHAT MOVES YOU

HOME COLLECTION ABOUT CONTACT

OUR BEST SELLERS

Latest Arrivals

SHOP NOW

LATEST COLLECTIONS —

Step into a world of style with our newest collections, carefully curated to bring you the best in fashion, home decor, and more.

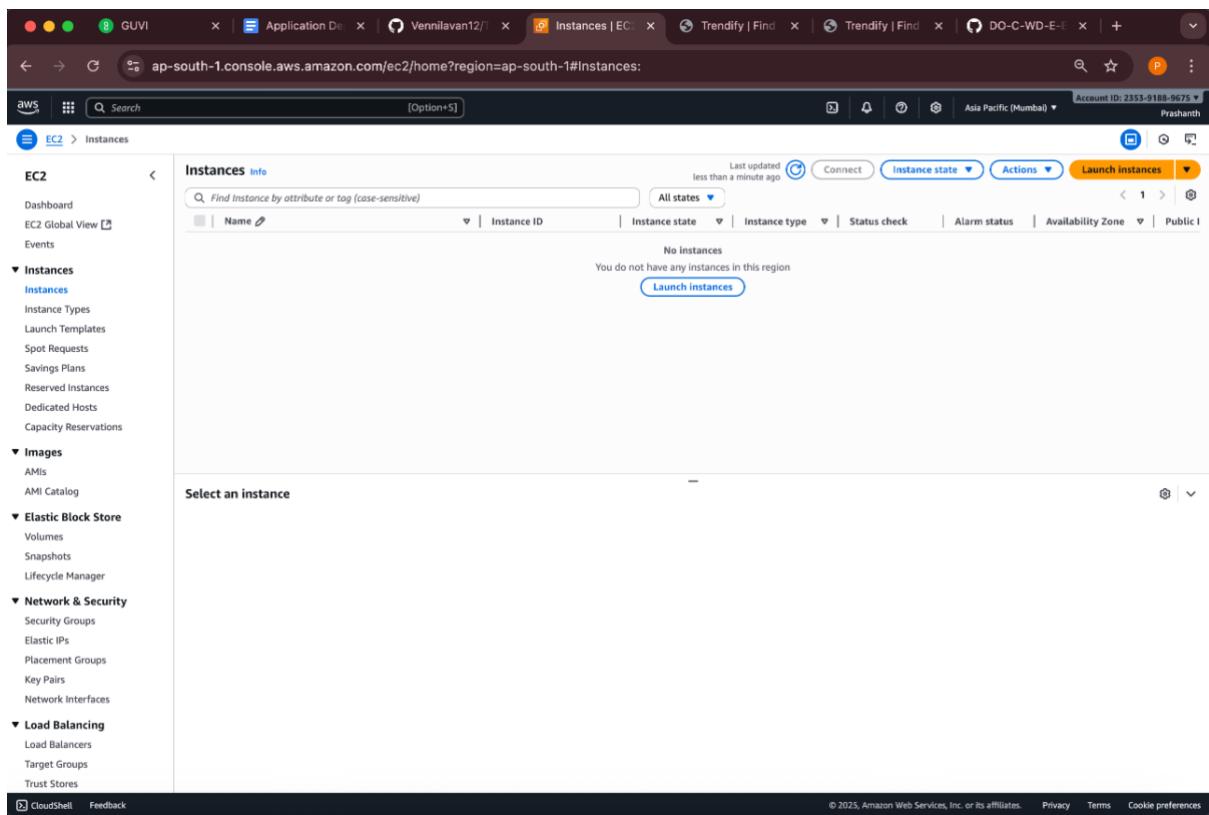


=====

Terraform:

- Define infrastructure in main.tf to create VPC, IAM, EC2 with Jenkins, etc.
- Use terraform command to provision infrastructure.

BEFORE TERRAFORM



TERRAFORM INSTALL (main.tf)

```
# -----
# Security Group
# -----
resource "aws_security_group" "jenkins_sg" {
  name      = "jenkins-sp"
  description = "Allow SSH and Jenkins"
  vpc_id    = aws_vpc.main_vpc.id

  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"]
  }

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

# -----
# IAM Role & Instance Profile
# -----
resource "aws_iam_role" "ec2_role" {
  name = "ec2-jenkins-role"

  assume_role_policy = jsonencode({
    Version = "2012-10-17",
    Statement = [
      Action     = "sts:AssumeRole"
      Effect    = "Allow"
      Principal = { Service = "ec2.amazonaws.com" }
    ]
  })
}

resource "aws_iam_role_policy_attachment" "ec2_attach" {
  role      = aws_iam_role.ec2_role.name
  policy_arn = "arn:aws:iam::aws:policy/AdministratorAccess"
}

resource "aws_iam_instance_profile" "ec2_profile" {
  name = "ec2-jenkins-profile"
  role = aws_iam_role.ec2_role.name
}

# -----
# Provider
# -----
provider "aws" {
  region = var.region
}

# -----
# VPC + Subnet + IGW + Route
# -----
resource "aws_vpc" "main_vpc" {
  cidr_block = "10.0.0.0/16"
  tags       = { Name = "MainVPC" }
}

resource "aws_subnet" "public_subnet" {
  vpc_id        = aws_vpc.main_vpc.id
  cidr_block   = "10.0.1.0/24"
  availability_zone = "ap-south-1a"
  map_public_ip_on_launch = true
  tags         = { Name = "PublicSubnet" }
}

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main_vpc.id
  tags   = { Name = "MainIGW" }
}

resource "aws_route_table" "public_rt" {
  vpc_id = aws_vpc.main_vpc.id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.igw.id
  }

  tags = { Name = "PublicRouteTable" }
}

resource "aws_route_table_association" "public_assoc" {
  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-sp"
  description = "Allow SSH and Jenkins"
  vpc_id    = aws_vpc.main_vpc.id

  ingress {
```

```
role = aws_iam_role.ec2_role.name

}

# -----
# EC2 Instance with Jenkins
#
resource "aws_instance" "jenkins_server" {
  ami                      = "ami-02d26659fd82cf299" # Ubuntu CANONICAL (ap-south-1)
  instance_type             = var.instance_type
  subnet_id                 = aws_subnet.public_subnet.id
  key_name                  = var.key_name
  vpc_security_group_ids   = [aws_security_group.jenkins_sg.id]
  iam_instance_profile      = aws_iam_instance_profile.ec2_profile.name

  user_data = <<<EOF
  #!/bin/bash
  # Step 1: Update system
  sudo apt update -y && sudo apt upgrade -y

  # Step 2: Install Java (OpenJDK 17)
  sudo apt install -y openjdk-17-jdk

  # Step 3: Add Jenkins repo & key
  curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \
    /usr/share/keyrings/jenkins-keyring.asc > /dev/null
  echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian binary/" | sudo tee \
    /etc/apt/sources.list.d/jenkins.list > /dev/null

  # Step 4: Install Jenkins
  sudo apt update
  sudo apt install -y jenkins

  # Step 5: Start & enable Jenkins
  sudo systemctl start jenkins
  sudo systemctl enable jenkins
  EOF

  tags = { Name = "JenkinsServer" }
}

variable "region" {
  description = "AWS region"
  default     = "ap-south-1"
}

variable "instance_type" {
  description = "EC2 instance type"
  default     = "t2.micro"
}

variable "key_name" {
  description = "PrashanthEC2"
  type        = string
}
```

```
(base) prashantr@Prashanth-MacBook-Pro-2 terraform %
(base) prashantr@Prashanth-MacBook-Pro-2 terraform % nano variables.tf
(base) prashantr@Prashanth-MacBook-Pro-2 terraform % cat variables.tf
variable "region" {
  description = "AWS region"
  default     = "ap-south-1"
}

variable "instance_type" {
  description = "EC2 instance type"
  default     = "t2.micro"
}

variable "key_name" {
  description = "PrashanthEC2"
  type        = string
}

(base) prashantr@Prashanth-MacBook-Pro-2 terraform %
(base) prashantr@Prashanth-MacBook-Pro-2 terraform % nano outputs.tf
(base) prashantr@Prashanth-MacBook-Pro-2 terraform % cat outputs.tf
output "jenkins_public_ip" {
  description = "Public IP of Jenkins server"
  value       = aws_instance.jenkins_server.public_ip
}

output "jenkins_url" {
  description = "Jenkins Web UI URL"
  value       = "http://${aws_instance.jenkins_server.public_ip}:8080"
}

(base) prashantr@Prashanth-MacBook-Pro-2 terraform %
(base) prashantr@Prashanth-MacBook-Pro-2 terraform % terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.11.0

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.
(base) prashantr@Prashanth-MacBook-Pro-2 terraform %
(base) prashantr@Prashanth-MacBook-Pro-2 terraform % terraform validate
Success! The configuration is valid.

(base) prashantr@Prashanth-MacBook-Pro-2 terraform %
(base) prashantr@Prashanth-MacBook-Pro-2 terraform % terraform plan
var.key_name
  PrashanthEC2

Enter a value:
```

TERRAFORM PLAN

```
  + inline_policy (known after apply)
}

# aws_iam_role_attachment.ec2_attach will be created
+ resource "aws_iam_role_policy_attachment" "ec2_attach" {
  + id          = (known after apply)
  + policy_arn = "arn:aws:iam::aws:policy/AdministratorAccess"
  + role        = "ec2-jenkins-role"
}

# aws_instance.jenkins_server will be created
+ resource "aws_instance" "jenkins_server" {
  + ami           = "ami-02d26659fd82cf299"
  + arn          = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone      = (known after apply)
  + disable_eni_stretch    = (known after apply)
  + disable_ei_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 = "ecc2-jenkins-profile"
  + 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"
  + ipv6_address_count   = (known after apply)
  + ipv6_addresses        = (known after apply)
  + key_name              = (known after apply)
  + monitoring            = (known after apply)
  + outpost_arn           = (known after apply)
  + password_data         = (known after apply)
  + placement_group       = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns           = (known after apply)
  + private_ip             = (known after apply)
  + public_dns             = (known after apply)
  + public_ip              = (known after apply)
  + region                = "ap-south-1"
  + secondary_private_ips = (known after apply)
  + security_groups        = (known after apply)
  + source_dest_check     = true
  + spot_instance_request_id = (known after apply)
  + subnet_id              = (known after apply)
  + tags
    + "Name" = "JenkinsServer"
  + tags_all               = {
    + "Name" = "JenkinsServer"
  }
  + tenancy                = (known after apply)
  + user_data              = <<-EOT
    #!/bin/bash
}

Success! The configuration is valid.

(base) prashanth@Prashanth-MacBook-Pro-2:~/Desktop$ terraform validate
Success! The configuration is valid.

(base) prashanth@Prashanth-MacBook-Pro-2:~/Desktop$ terraform plan
var.key_name
PrashanthEC2

Enter a value:

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_iam_instance_profile.ec2_profile will be created
+ resource "aws_iam_instance_profile" "ec2_profile" {
  + arn          = (known after apply)
  + create_date  = (known after apply)
  + id          = (known after apply)
  + name        = "ecc2-jenkins-profile"
  + name_prefix  = (known after apply)
  + path         = "/"
  + role        = "ec2-jenkins-role"
  + tags_all    = (known after apply)
  + unique_id   = (known after apply)
}

# aws_iam_role.ec2_role will be created
+ resource "aws_iam_role" "ec2_role" {
  + arn          = (known after apply)
  + assume_role_policy = jsonencode(
    {
      + Statement = [
        + {
          + Action    = "sts:AssumeRole"
          + Effect   = "Allow"
          + Principal = {
            + Service = "ec2.amazonaws.com"
          }
        ],
        + Version  = "2012-10-17"
      ]
    }
  )
  + create_date  = (known after apply)
  + force_detach_policies = false
  + id          = (known after apply)
  + managed_policy_arns = (known after apply)
  + max_session_duration = 3600
  + name        = "ec2-jenkins-role"
  + name_prefix  = (known after apply)
  + path         = "/"
  + tags_all    = (known after apply)
  + unique_id   = (known after apply)
}

+ inline_policy (known after apply)
```

```
 terraform --zsh - 197x59
+ tags = {
+   + "Name" = "JenkinsServer"
+ }
+ tags_all = {
+   + "Name" = "JenkinsServer"
+ }
+ tenancy = (known after apply)
+ user_data = <<-EOT
# /bin/bash
# Step 1: Update system
sudo apt update -y && sudo apt upgrade -y
# Step 2: Install Java (OpenJDK 17)
sudo apt install -y openjdk-17-jdk
# Step 3: Add Jenkins repo & key
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian binary/" | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
# Step 4: Install Jenkins
sudo apt update
sudo apt install -y jenkins
# Step 5: Start & enable Jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
EOT
+ user_data_base64 = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ primary_network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
```

```
 terraform --zsh - 197x59
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

# aws_internet_gateway.igw will be created
resource "aws_internet_gateway" "igw" {
+ arn = (known after apply)
+ id = (known after apply)
+ owner_id = (known after apply)
+ region = "ap-south-1"
+ tags = {
+   + "Name" = "MainIGW"
}
+ tags_all = {
+   + "Name" = "MainIGW"
}
+ vpc_id = (known after apply)
}

# aws_route_table.public_rt will be created
resource "aws_route_table" "public_rt" {
+ arn = (known after apply)
+ id = (known after apply)
+ owner_id = (known after apply)
+ propagating_vgws = (known after apply)
+ region = "ap-south-1"
+ route = [
+   {
+     + cidr_block = "0.0.0.0/0"
+     + gateway_id = (known after apply)
+     # (all unchanged attributes hidden)
+   },
]
+ tags = {
+   + "Name" = "PublicRouteTable"
}
+ tags_all = {
+   + "Name" = "PublicRouteTable"
}
+ vpc_id = (known after apply)
}

# aws_route_table_association.public_assoc will be created
resource "aws_route_table_association" "public_assoc" {
+ id = (known after apply)
+ region = "ap-south-1"
+ route_table_id = (known after apply)
+ subnet_id = (known after apply)
}

# aws_security_group.jenkins_sg will be created
resource "aws_security_group" "jenkins_sg" {
+ arn = (known after apply)
+ description = "Allow SSH and Jenkins"
+ egress = [
+   {
+     + cidr_blocks = [
+       "0.0.0.0/0",
]
```

```

# aws_security_group.jenkins_sg will be created
+ resource "aws_security_group" "jenkins_sg" {
  + arn           = (known after apply)
  + description   = "Allow SSH and Jenkins"
  + egress        = [
    + {
      + cidr_blocks = [
        + "0.0.0.0/0",
      ]
      + from_port   = 0
      + ipv6_cidr_blocks = []
      + prefix_list_ids = []
      + protocol    = "tcp"
      + self         = false
      + to_port     = 0
      # (1 unchanged attribute hidden)
    },
  ]
  + id           = (known after apply)
  + ingress      = [
    + {
      + cidr_blocks = [
        + "0.0.0.0/0",
      ]
      + from_port   = 22
      + ipv6_cidr_blocks = []
      + prefix_list_ids = []
      + protocol    = "tcp"
      + security_groups = []
      + self         = false
      + to_port     = 22
      # (1 unchanged attribute hidden)
    },
    + {
      + cidr_blocks = [
        + "0.0.0.0/0",
      ]
      + from_port   = 8080
      + ipv6_cidr_blocks = []
      + prefix_list_ids = []
      + protocol    = "tcp"
      + security_groups = []
      + self         = false
      + to_port     = 8080
      # (1 unchanged attribute hidden)
    },
  ],
  + name         = "jenkins-sg"
  + name_prefix  = (known after apply)
  + owner_id     = (known after apply)
  + region       = "ap-south-1"
  + revoke_rules_on_delete = false
  + tags_all    = (known after apply)
  + vpc_id       = (known after apply)
}

# aws_subnet.public_subnet will be created
+ resource "aws_subnet" "public_subnet" {
  + arn           = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone   = "ap-south-1a"
  + availability_zone_id = (known after apply)
  + cidr_block     = "10.0.1.0/24"
  + enable_dns64   = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id             = (known after apply)
  + ipv4_cidr_block_association_id = (known after apply)
  + ipv4_native    = false
  + map_public_ip_on_launch = true
  + owner_id       = (known after apply)
  + private_dns_hostname_type_on_launch = (known after apply)
  + region         = "ap-south-1"
  + tags           = {
    + "Name" = "PublicSubnet"
  }
  + tags_all      = {
    + "Name" = "PublicSubnet"
  }
  + vpc_id         = (known after apply)
}

# aws_vpc.main_vpc will be created
+ resource "aws_vpc" "main_vpc" {
  + arn           = (known after apply)
  + cidr_block    = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id = (known after apply)
  + enable_dns_hostnames = (known after apply)
  + enable_dns_support = true
  + enable_network_address_usage_metrics = (known after apply)
  + id             = (known after apply)
  + instance_tenancy = "default"
  + ipv4_association_id = (known after apply)
  + ipv4_cidr_block = (known after apply)
  + ipv4_cidr_block_association_id = (known after apply)
  + main_route_table_id = (known after apply)
  + owner_id       = (known after apply)
  + region         = "ap-south-1"
  + tags           = {
    + "Name" = "MainVPC"
  }
  + tags_all      = {
    + "Name" = "MainVPC"
  }
}

Plan: 10 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ jenkins_public_ip = (known after apply)
+ jenkins_url      = (known after apply)

```

TERRAFORM APPLY

```
● ● ● terraform -- zsh - 197x59
Plan: 10 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ jenkins_public_ip = (known after apply)
+ jenkins_url      = (known after apply)

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
(base) prashantr@Prashanth-MacBook-Pro-2:~/Desktop$ (base) prashantr@Prashanth-MacBook-Pro-2:~/Desktop$ terraform apply -var="key_name=PrashanthEC2" -auto-approve

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_iam_instance_profile.ec2_profile will be created
+ resource "aws_iam_instance_profile" "ec2_profile" {
  + arn      = (known after apply)
  + create_date = (known after apply)
  + id       = (known after apply)
  + name     = "ec2-jenkins-profile"
  + name_prefix = (known after apply)
  + path     = "/"
  + role     = "ec2-jenkins-role"
  + tags_all = (known after apply)
  + unique_id = (known after apply)
}

# aws_iam_role.ec2_role will be created
+ resource "aws_iam_role" "ec2_role" {
  + arn      = (known after apply)
  + assume_role_policy = jsonencode(
    {
      + Statement = [
        + {
          + Action      = "sts:AssumeRole"
          + Effect     = "Allow"
          + Principal  = {
              + Service = "ec2.amazonaws.com"
            }
        },
        + Version   = "2012-10-17"
      ]
    }
  )
  + create_date      = (known after apply)
  + force_detach_policies = false
  + id       = (known after apply)
  + managed_policy_arns = (known after apply)
  + max_session_duration = 3600
  + name     = "ec2-jenkins-role"
  + name_prefix = (known after apply)
  + path     = "/"
  + tags_all = (known after apply)
  + unique_id = (known after apply)
  + inline_policy (known after apply)
}

# aws_iam_role_policy_attachment.ec2_attach will be created
+ resource "aws_iam_role_policy_attachment" "ec2_attach" {
  + id      = (known after apply)
  + policy_arn = "arn:aws:iam::aws:policy/AdministratorAccess"
  + role     = "ec2-jenkins-role"
}

# aws_instance.jenkins_server will be created
+ resource "aws_instance" "jenkins_server" {
  + ami      = "ami-02d26659fd82cf299"
  + 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    = "ec2-jenkins-profile"
  + 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"
  + ipv4_address_count     = (known after apply)
  + ipv6_addresses          = (known after apply)
  + key_name               = "PrashanthEC2"
  + monitoring             = (known after apply)
  + outpost_arn             = (known after apply)
  + password_data           = (known after apply)
  + placement_group         = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns             = (known after apply)
  + private_ip              = (known after apply)
  + public_dns              = (known after apply)
  + public_ip               = (known after apply)
  + region                 = "ap-south-1"
  + secondary_private_ips  = (known after apply)
  + security_groups         = (known after apply)
  + source_dest_check       = true
  + spot_instance_request_id = (known after apply)
  + subnet_id               = (known after apply)
  + tags                   = {
      + "Name" = "JenkinsServer"
    }
  + tags_all               = {
      + "Name" = "JenkinsServer"
    }
  + tenancy                = (known after apply)
  + user_data               = <<EOT
    #!/bin/bash
    # Step 1: Update system
    sudo apt update -y && sudo apt upgrade -y
  
```

```

  + tenancy           = (known after apply)
  + user_data         = <<EOA
    #!/bin/bash
    # Step 1: Update system
    sudo apt update -y && sudo apt upgrade -y

    # Step 2: Install Java (OpenJDK 17)
    sudo apt install -y openjdk-17-jdk

    # Step 3: Add Jenkins repo & key
    curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null

    echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
      https://pkg.jenkins.io/debian/binary/ | sudo tee \
      /etc/apt/sources.list.d/jenkins.list > /dev/null

    # Step 4: Install Jenkins
    sudo apt update
    sudo apt install -y jenkins

    # Step 5: Start & enable Jenkins
    sudo systemctl start jenkins
    sudo systemctl enable jenkins
  EOT
  + user_data_base64      = (known after apply)
  + user_data_replace_on_change = false
  + vpc_security_group_ids = (known after apply)

  + capacity_reservation_specification (known after apply)
  + cpu_options (known after apply)
  + ebs_block_device (known after apply)
  + enclave_options (known after apply)
  + ephemeral_block_device (known after apply)
  + instance_market_options (known after apply)
  + maintenance_options (known after apply)
  + metadata_options (known after apply)
  + network_interface (known after apply)
  + primary_network_interface (known after apply)
  + private_dns_name_options (known after apply)
  + root_block_device (known after apply)
}

# aws_internet_gateway.igw will be created
resource "aws_internet_gateway" "igw" {
  + arn      = (known after apply)
  + id       = (known after apply)
  + owner_id = (known after apply)
}

# aws_route_table.public_rt will be created
resource "aws_route_table" "public_rt" {
  + arn      = (known after apply)
  + id       = (known after apply)
  + owner_id = (known after apply)
  + propagating_vgws = (known after apply)
  + region   = "ap-south-1"
  + route   = [
    + {
      + cidr_block          = "0.0.0.0/0"
      + gateway_id          = (known after apply)
      + # (11 unchanged attributes hidden)
    },
  ],
  + tags      = {
    + "Name" = "PublicRouteTable"
  }
  + tags_all  = {
    + "Name" = "PublicRouteTable"
  }
  + vpc_id    = (known after apply)
}

# aws_route_table_association.public_assoc will be created
resource "aws_route_table_association" "public_assoc" {
  + id       = (known after apply)
  + region   = "ap-south-1"
  + route_table_id = (known after apply)
  + subnet_id = (known after apply)
}

# aws_security_group.jenkins_sg will be created
resource "aws_security_group" "jenkins_sg" {
  + arn      = (known after apply)
  + description = "Allow SSH and Jenkins"
  + egress   = [
    + {
      + cidr_blocks = [
        + "0.0.0.0/0",
      ]
      + from_port   = 8
      + ipv6_cidr_blocks = []
      + prefix_list_ids = []
    }
  ]
}

```

```

  + from_port          = 0
  + ipv6_cidr_blocks = []
  + prefix_list_ids  = []
  + protocol          = "1"
  + security_groups   = []
  + self               = false
  + to_port            = 0
  # (1 unchanged attribute hidden)
},
]
+ id                  = (known after apply)
+ ingress           =
+ {
  + cidr_blocks     = [
    ]
  + from_port        = 22
  + ipv6_cidr_blocks = []
  + prefix_list_ids  = []
  + protocol          = "tcp"
  + security_groups   = []
  + self               = false
  + to_port            = 22
  # (1 unchanged attribute hidden)
},
+ {
  + cidr_blocks     = [
    ]
  + from_port        = 8080
  + ipv6_cidr_blocks = []
  + prefix_list_ids  = []
  + protocol          = "tcp"
  + security_groups   = []
  + self               = false
  + to_port            = 8080
  # (1 unchanged attribute hidden)
},
]
+ name                = "jenkins-sg"
+ name_prefix         = (known after apply)
+ owner_id             = (known after apply)
+ region              = "ap-south-1"
+ revoke_rules_on_delete = false
+ tags_all             = (known after apply)
+ vpc_id               = (known after apply)
}

# aws_subnet.public_subnet will be created
resource "aws_subnet" "public_subnet" {
  + arn          = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone      = "ap-south-1a"
  + availability_zone_id   = (known after apply)
  + cidr_block           = "10.0.1.0/24"
  + enable_dns64          = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id                   = (known after apply)
}


```

SUCCESSFUL JENKINS RUN USING TERRAFORM

```

  # aws_vpc.main_vpc will be created
resource "aws_vpc" "main_vpc" {
  + arn          = (known after apply)
  + cidr_block           = "10.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dhcp_options_id      = (known after apply)
  + enable_dns_hostnames = (known after apply)
  + enable_dns_support    = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                   = (known after apply)
  + instance_tenancy      = "default"
  + ipv6_association_id   = (known after apply)
  + ipv6_cidr_block       = (known after apply)
  + main_route_table_id   = (known after apply)
  + owner_id             = (known after apply)
  + region              = "ap-south-1"
  + tags
    + "Name" = "MainVPC"
  }
+ tags_all             = {
    + "Name" = "MainVPC"
  }

Plan: 10 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ jenkins_public_ip = (known after apply)
+ jenkins_url      = (known after apply)
aws_iam_role.ec2_role: Creating...
aws_vpc.main_vpc: Creation complete after 1s [id=vpc-0ad33d0cd729f001]
aws_internet_gateway.igw: Creating...
aws_subnet.public_subnet: Creating...
aws_security_group.jenkins_sg: Creating...
aws_iam_role.ec2_role: Creation complete after 2s [id=ec2-jenkins-role]
aws_iam_role_policy_attachment.ec2_attach: Creating...
aws_iam_instance_profile.ec2_profile: Creating...
aws_internet_gateway.igw: Creation complete after 1s [id=igw-08841e00c4d721d4c]
aws_route_table.public_rt: Creating...
aws_iam_role_policy_attachment.ec2_attach: Creation complete after 1s [id=ec2-jenkins-role/arn:aws:iam::aws:policy/AdministratorAccess]
aws_route_table.public_rt: Creation complete after 1s [id=rth-0d81f11024c1a5fd]
aws_security_group.jenkins_sg: Creation complete after 3s [id=sg-06ce9e72bcabfb1d6]
aws_iam_instance_profile.ec2_profile: Creation complete after 7s [id=ec2-jenkins-profile]
aws_subnet.public_subnet: Still creating... [00m10s elapsed]
aws_subnet.public_subnet: Creation complete after 11s [id=subnet-059a9394e183080df5]
aws_route_table_association.public_assoc: Creating...
aws_instance_jenkins_server: Creating...
aws_route_table_association.public_assoc: Creation complete after 1s [id=rtbassoc-03a164e64c08f6c63]
aws_instance_jenkins_server: Still creating... [00m10s elapsed]
aws_instance_jenkins_server: Still creating... [00m20s elapsed]
aws_instance_jenkins_server: Still creating... [00m30s elapsed]
aws_instance_jenkins_server: Creation complete after 32s [id=i-01e85f107d6f2f6bb]

Apply complete! Resources: 10 added, 0 changed, 0 destroyed.


```

```

  terraform --zsh -- 197x59

  + default_security_group_id      = (known after apply)
  + dhcp_options_id               = (known after apply)
  + enable_dns_hostnames          = (known after apply)
  + enable_dns_support             = true
  + enable_network_address_usage_metrics = (known after apply)
  + id                            = (known after apply)
  + instance_tenancy              = "default"
  + ipv4_association_id           = (known after apply)
  + ipv4_cidr_block                = (known after apply)
  + ipv6_cidr_block_network_border_group = (known after apply)
  + main_route_table_id            = (known after apply)
  + owner_id                       = (known after apply)
  + region                          = "ap-south-1"
  + tags                           = {
    + "Name" = "MainVPC"
  }
  + tags_all                        = {
    + "Name" = "MainVPC"
  }

Plan: 10 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + jenkins_public_ip = (known after apply)
  + jenkins_url       = (known after apply)
aws_vpc.main_vpc: Creating...
aws_iam_role.ec2_role: Creating...
aws_iam_role_policy_attachment.ec2_attach: Creating...
aws_iam_instance_profile.ec2_profile: Creating...
aws_internet_gateway.igw: Creation complete after 1s [id=igw-08841e00c4d721a4c]
aws_route_table.public_rt: Creating...
aws_route_table_attachment.ec2_attach: Creation complete after 1s [id=ec2-0ad33d0dc729f001]
aws_internet_gateway.igw: Creating...
aws_subnet.public_subnet: Creating...
aws_security_group.jenkins_sg: Creating...
aws_iam_role.ec2_role: Creation complete after 2s [id=ec2-jenkins-role]
aws_iam_role_policy_attachment.ec2_attach: Creating...
aws_iam_instance_profile.ec2_profile: Creating...
aws_internet_gateway.igw: Creation complete after 1s [id=igw-08841e00c4d721a4c]
aws_route_table.public_rt: Creating...
aws_route_table_attachment.ec2_attach: Creation complete after 1s [id=ec2-0db1f11024c1a5fd]
aws_security_group.jenkins_sg: Creation complete after 3s [id=sg-06ce9e72bcabfb1d6]
aws_iam_instance_profile.ec2_profile: Creation complete after 7s [id=ec2-jenkins-profile]
aws_subnet.public_subnet: Still creating... [0m10s elapsed]
aws_subnet_public_subnet: Creation complete after 11s [id=subnet-059a9394e18308df5]
aws_instance_association_public: Creating...
aws_instance_jenkins_server: Creating...
aws_route_table_association.public_assoc: Creation complete after 1s [id=rtbassoc-03a164e64c08f6c63]
aws_instance_jenkins_server: Still creating... [0m10s elapsed]
aws_instance_jenkins_server: Still creating... [0m20s elapsed]
aws_instance_jenkins_server: Still creating... [0m30s elapsed]
aws_instance_jenkins_server: Creation complete after 32s [id=i-01e05f107d6f2f6bb]

Apply complete! Resources: 10 added, 0 changed, 0 destroyed.

Outputs:

jenkins_public_ip = "13.233.168.10"
jenkins_url = "http://13.233.168.10:8080"
(base) prashanth@Prashanth-MacBook-Pro-2:~/terraform % 
(base) prashanth@Prashanth-MacBook-Pro-2:~/terraform % 

```

AFTER TERRAFORM

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar navigation includes links for GUVI, Application Dev, Vennilavan12, Instances (selected), Trendify | Find, Trendify | Find, DO-C-WD-E-E, and CloudShell Feedback. The top right corner displays the Account ID: 2353-0188-9675, Region: Asia Pacific (Mumbai), and User: Prashanth.

The main content area is titled "Instances (1) Info". It lists one instance named "JenkinsServer" with the following details:

- Instance ID: i-01e05f107d6f2f6bb
- Instance state: Running
- Instance type: t2.micro
- Status check: Initializing
- Last updated: less than a minute ago
- Actions: Connect, Instance state, Actions, Launch instances

Below the table, there is a section titled "Select an instance" which is currently empty.

The bottom of the page includes a footer with links for © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Instance summary for i-01e05f107d6f2f6bb (JenkinsServer) [Info](#)

Updated less than a minute ago

Instance ID	i-01e05f107d6f2f6bb	Public IPv4 address	10.0.1.129
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-10-0-1-129.ap-south-1.compute.internal	Private IP DNS name (IPv4 only)	ip-10-0-1-129.ap-south-1.compute.internal
Answer private resource DNS name	-	Instance type	t2.micro
Auto-assigned IP address	13.233.168.10 [Public IP]	VPC ID	vpc-0ad35d0cd729f001 (MainVPC)
IAM Role	ec2-jenkins-role	Subnet ID	subnet-059a9394e18308df5 (PublicSubnet)
IMDv2	Required	Instance ARN	arn:aws:ec2:ap-south-1:235391889675:instance/i-01e05f107d6f2f6bb
Operator	-	Platform details	Linux/UNIX
Details	Status and alarms	Monitoring	Termination protection
Instance details Info		disabled	Disabled
AMI ID	ami-02d26659fd82cf299	AMI name	ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20250821
AMI name	ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20250821	Launch time	Sat Aug 30 2025 23:30:09 GMT+0530 (India Standard Time) (5 minutes)
Stop protection	Disabled	Instance auto-recovery	Default
Instance reboot migration	Default (On)	Lifecycle	normal

[CloudShell](#) [Feedback](#)

Security details

IAM Role	Owner ID	Launch time
ec2-jenkins-role	235391889675	Sat Aug 30 2025 23:30:09 GMT+0530 (India Standard Time)
Security groups		
sg-06ce9e72bcabfb1d6 (jenkins-sg)		

Inbound rules

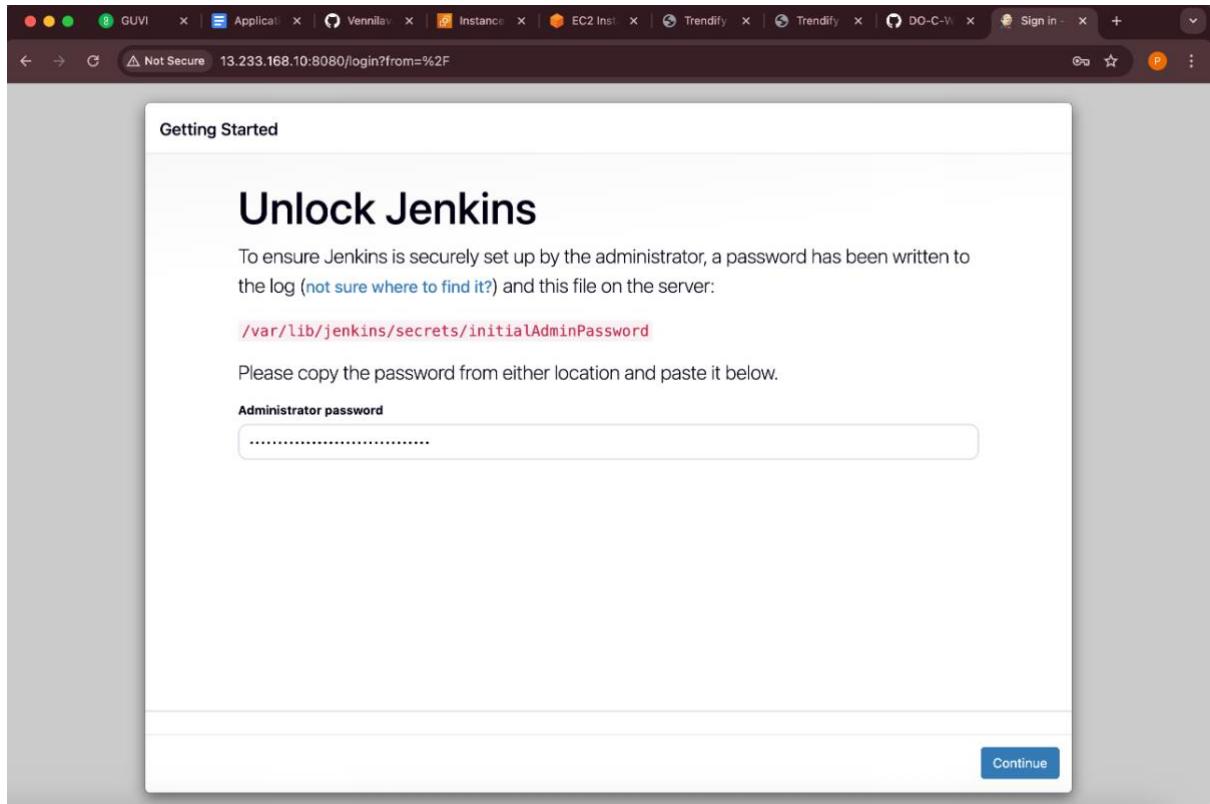
Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-01c5737d245fd2e80	22	TCP	0.0.0.0/0	jenkins-sg	-
-	sgr-0bf16fd79d5725b66	8080	TCP	0.0.0.0/0	jenkins-sg	-

Outbound rules

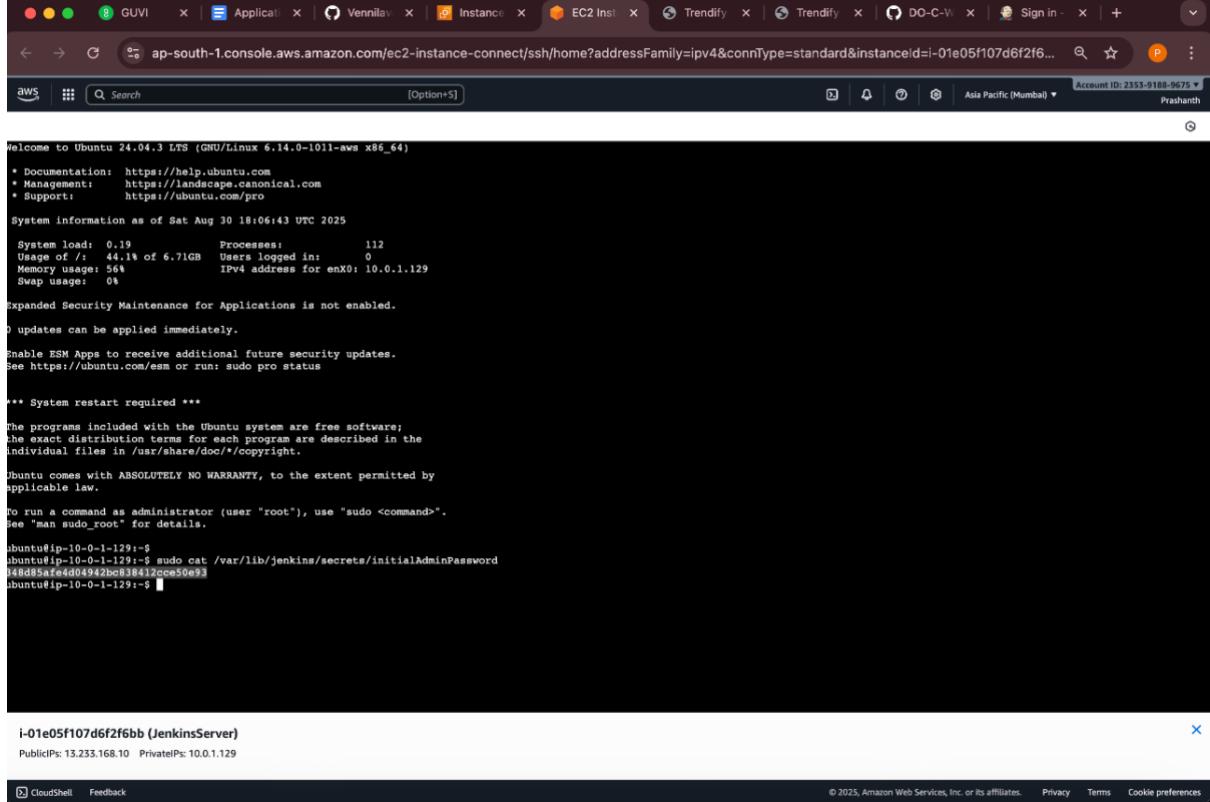
Name	Security group rule ID	Port range	Protocol	Destination	Security groups	Description
-	sgr-086a4af2b508d11da	All	All	0.0.0.0/0	jenkins-sg	-

[CloudShell](#) [Feedback](#)

JENKINS SETUP



The screenshot shows a web browser window with the URL `13.233.168.10:8080/login?from=%2F`. The page title is "Getting Started" and the main heading is "Unlock Jenkins". A note says: "To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server: `/var/lib/jenkins/secrets/initialAdminPassword`". Below this, it says "Please copy the password from either location and paste it below." There is a text input field labeled "Administrator password" containing several dots. At the bottom right is a blue "Continue" button.



The screenshot shows an AWS CloudShell terminal window. The terminal prompt is `i-01e05f107d6f2f6bb (JenkinsServer)`. The terminal output shows the Ubuntu 24.04 LTS welcome screen followed by a command-line session:

```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1011-aws x86_64)

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

System information as of Sat Aug 30 18:06:43 UTC 2025

 System Load: 0.19      Processes:          112
 Usage of /: 44.1% of 6.71GB  Users logged in: 0
 Memory usage: 56%
 Swap usage:  0%
IPv4 address for enX0: 10.0.1.129

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

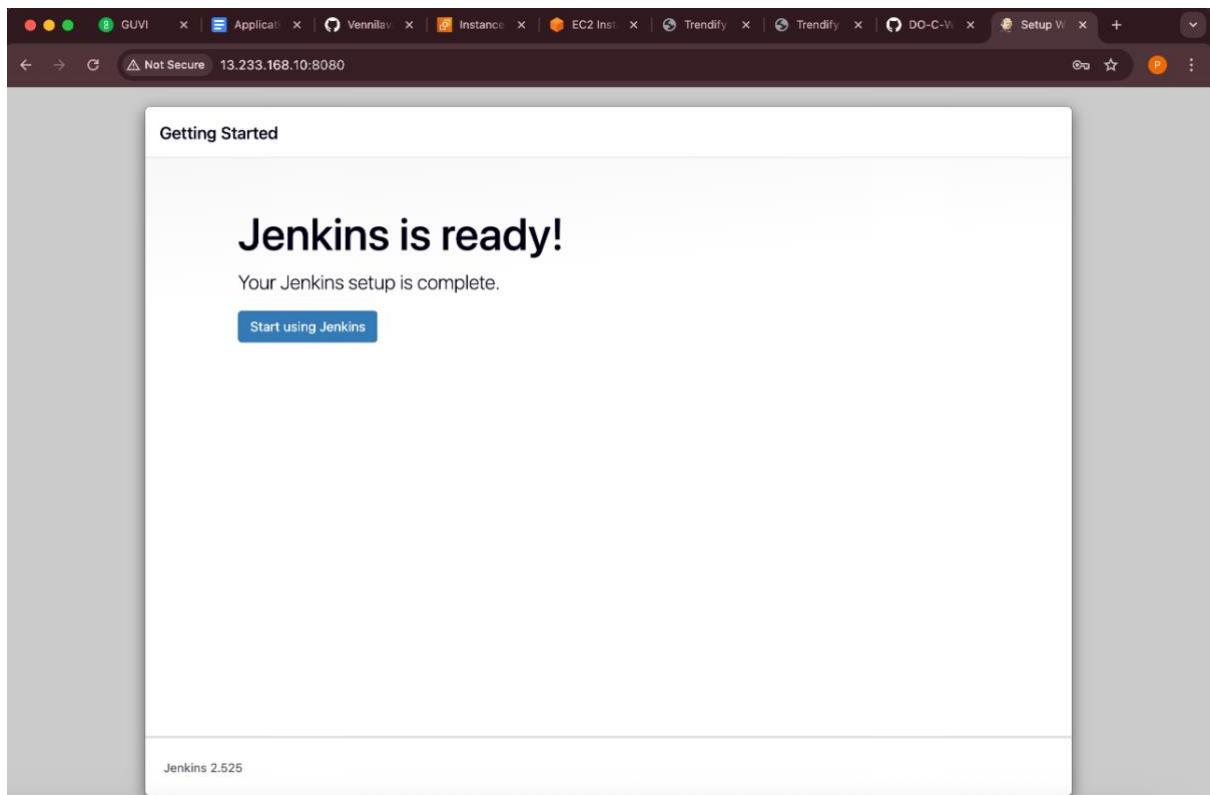
*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

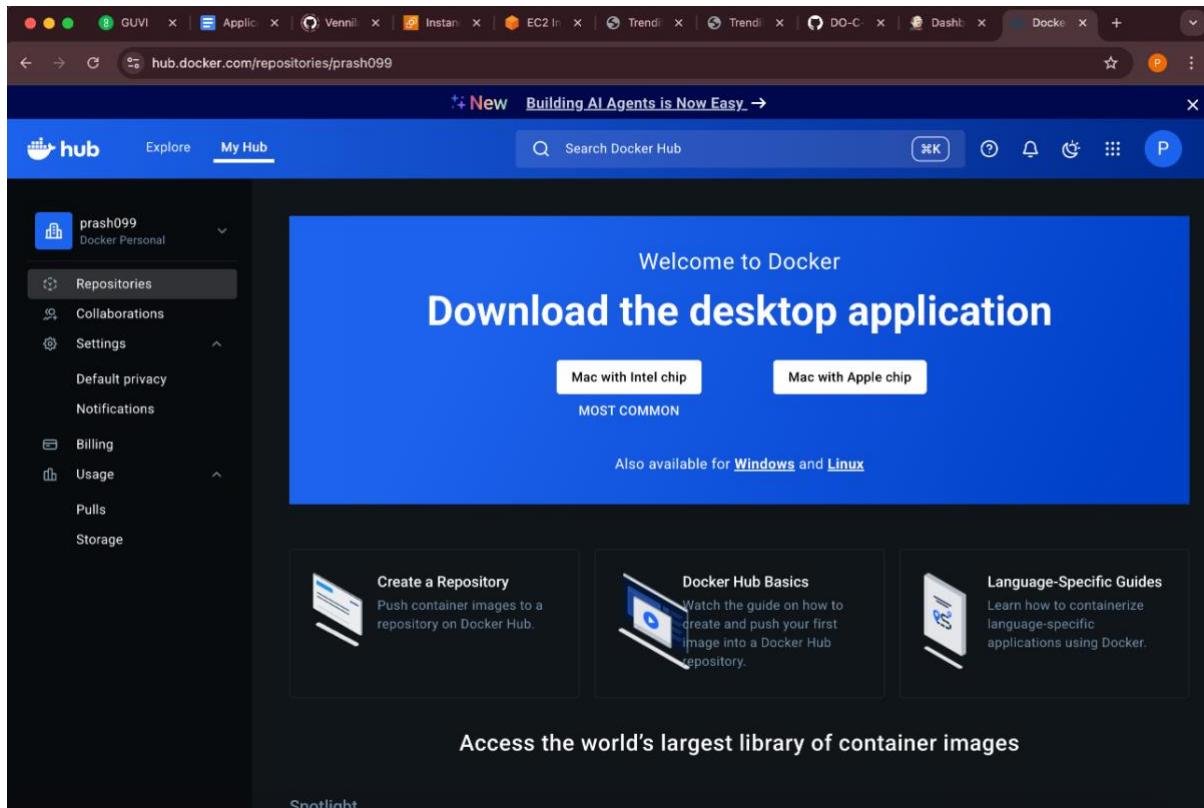
ubuntu@ip-10-0-1-129:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
i48d85afe4d04942bc038412cce50e93
ubuntu@ip-10-0-1-129:~$
```



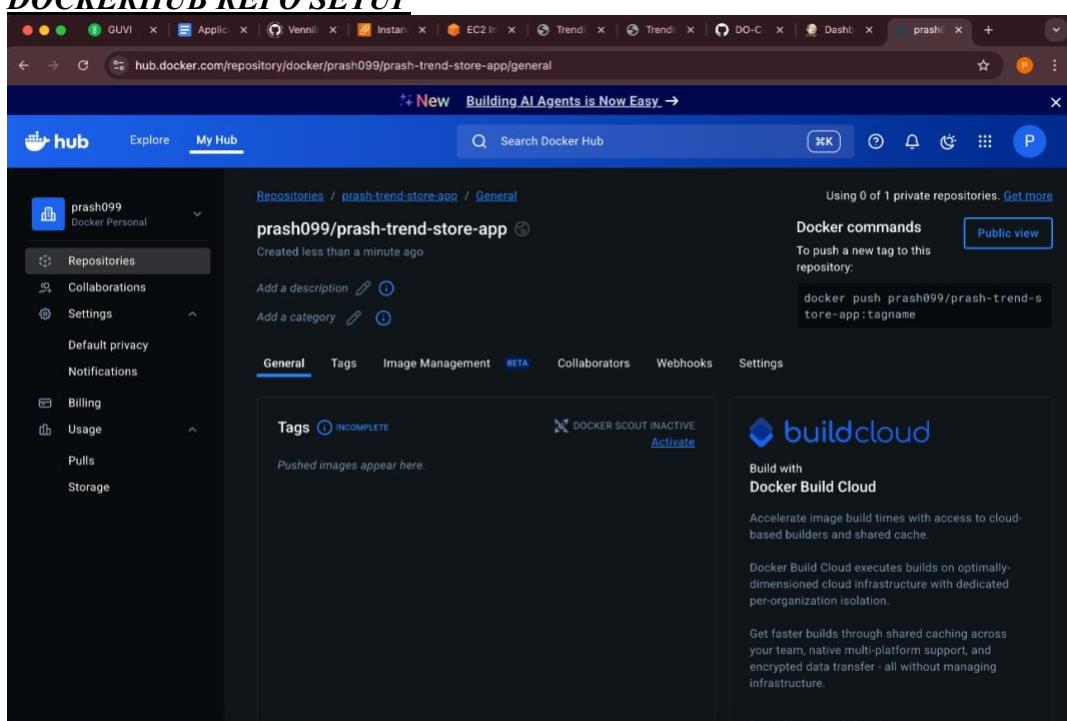
The screenshot shows the Jenkins dashboard. At the top left, there's a Jenkins icon and the word 'Jenkins'. To the right are links for '+ New Item', 'Build History', and 'Add description'. In the center, a large heading says 'Welcome to Jenkins!'. Below it, a message says 'This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.' A button labeled 'Start building your software project' is present. On the left, there's a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing '0/2'. On the right, there's a 'Create a job' button with a '+' sign. Below that, a section titled 'Set up a distributed build' includes 'Set up an agent' (with a monitor icon), 'Configure a cloud' (with a cloud icon), and a link 'Learn more about distributed builds' with a help icon. At the bottom right, it says 'REST API' and 'Jenkins 2.525'.

DockerHub:

- Create a DockerHub repository.



DOCKERHUB REPO SETUP



Kubernetes:

- Setup Kubernetes in AWS EKS and Confirm EKS cluster is running.

```
=====
(base) prashanth@Prashanth-MacBook-Pro-2 Trend %
(base) prashanth@Prashanth-MacBook-Pro-2 Trend % ls -ltr
total 69848
drwxr-xr-x  6 prashanthr  staff   192 Aug 30 12:44 dist
-rw-r--r--Q  1 prashanthr  staff 35755117 Aug 30 23:52 eksctl_Linux_amd64.tar.gz
-rw-r--r--  1 prashanthr  staff 1575 Aug 30 23:57 Jenkinsfile
drwxr-xr-x  4 prashanthr  staff  128 Aug 30 23:58 k8s
drwxr-xr-x  9 prashanthr  staff  288 Aug 31 02:18 terraform
(base) prashanth@Prashanth-MacBook-Pro-2 Trend %
(base) prashanth@Prashanth-MacBook-Pro-2 Trend % eksctl create cluster --name my-eks-cluster --version 1.28 --region ap-south-1 --nodegroup-name standard-workers --node-type t3.medium --nodes 2 --nodes-min 1 --nodes-max 3 --managed
2025-08-31 02:26:41 [i] eksctl version 0.210.8
2025-08-31 02:26:41 [i] using region ap-south-1
2025-08-31 02:26:41 [i] Amazon EKS will no longer publish EKS-optimized Amazon Linux 2 (AL2) AMIs after November 26th, 2025. Additionally, Kubernetes version 1.32 is the last version for which Amazon EKS will release AL2 AMIs. From version 1.33 onwards, Amazon EKS will continue to release AL2023 and Bottlerocket based AMIs. The default AMI family when creating clusters and nodegroups in Eksctl will be changed to AL2023 in the future.
2025-08-31 02:26:42 [i] setting availability zones to ap-south-1a, ap-south-1b, ap-south-1c
2025-08-31 02:26:42 [i] subnets for ap-south-1b - public:192.168.0.0/19 private:192.168.96.0/19
2025-08-31 02:26:42 [i] subnets for ap-south-1a - public:192.168.32.0/19 private:192.168.128.0/19
2025-08-31 02:26:42 [i] subnets for ap-south-1c - public:192.168.64.0/19 private:192.168.160.0/19
2025-08-31 02:26:42 [i] nodegroup "standard-workers" will use "" [AmazonLinux2/1.28]
2025-08-31 02:26:42 [i] using Kubernetes version 1.28
2025-08-31 02:26:42 [i] creating EKS cluster "my-eks-cluster" in "ap-south-1" region with managed nodes
2025-08-31 02:26:42 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2025-08-31 02:26:42 [i] if you encounter any issues, check CloudFormation console or try `eksctl utils describe-stacks --region=ap-south-1 --cluster=my-eks-cluster`
2025-08-31 02:26:42 [i] Kubernetes API endpoint access will use default of (publicAccess=true, privateAccess=false) for cluster "my-eks-cluster" in "ap-south-1"
2025-08-31 02:26:42 [i] CloudWatch logging will not be enabled for cluster "my-eks-cluster" in "ap-south-1"
2025-08-31 02:26:42 [i] you can enable it with `eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-south-1 --cluster=my-eks-cluster`
2025-08-31 02:26:42 [i] default addons vpc-cni, kube-proxy, coredns, metrics-server were not specified, will install them as EKS addons
2025-08-31 02:26:42 [i] 
2 sequential tasks:
  2 sequential sub-tasks:
    2 sequential sub-tasks:
      1 task: { create addons },
      wait for control plane to become ready,
    },
    create managed nodegroup "standard-workers",
  }
)
2025-08-31 02:26:42 [i] building cluster stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:26:42 [i] deploying stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:27:12 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:27:42 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:28:12 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:28:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:29:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:30:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:31:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:32:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:33:44 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:34:44 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:34:45 [i] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run `eksctl update addon`
2025-08-31 02:34:45 [i] creating addon: vpc-cni
2025-08-31 02:34:46 [i] successfully created addon: vpc-cni
2025-08-31 02:34:46 [i] creating addon: kube-proxy
2025-08-31 02:34:46 [i] successfully created addon: kube-proxy
2025-08-31 02:34:47 [i] creating addon: coredns
```

```

Trend --zsh-- 176x59

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

2025-08-31 02:26:42 [i] building cluster stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:26:42 [i] deploying stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:27:12 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:27:42 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:28:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:29:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:30:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:31:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:32:43 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:33:44 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:34:44 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-cluster"
2025-08-31 02:34:45 [i] recommended policies were found for "vpc-cni" addon, but since OIDC is disabled on the cluster, eksctl cannot configure the requested permissions; the recommended way to provide IAM permissions for "vpc-cni" addon is via pod identity associations; after addon creation is completed, add all recommended policies to the config file, under 'addon.PodIdentityAssociations', and run 'eksctl update addon'
2025-08-31 02:34:45 [i] creating addon: vpc-cni
2025-08-31 02:34:46 [i] successfully created addon: vpc-cni
2025-08-31 02:34:46 [i] creating addon: kube-proxy
2025-08-31 02:34:47 [i] successfully created addon: kube-proxy
2025-08-31 02:34:47 [i] creating addon: coredns
2025-08-31 02:34:47 [i] successfully created addon: coredns
2025-08-31 02:34:48 [i] creating addon: metrics-server
2025-08-31 02:34:48 [i] successfully created addon: metrics-server
2025-08-31 02:36:49 [i] building managed nodegroup stack "eksctl-my-eks-cluster-nodegroup-standard-workers"
2025-08-31 02:36:49 [i] deploying stack "eksctl-my-eks-cluster-nodegroup-standard-workers"
2025-08-31 02:37:19 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-nodegroup-standard-workers"
2025-08-31 02:37:19 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-nodegroup-standard-workers"
2025-08-31 02:37:56 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-nodegroup-standard-workers"
2025-08-31 02:38:35 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-nodegroup-standard-workers"
2025-08-31 02:40:23 [i] waiting for CloudFormation stack "eksctl-my-eks-cluster-nodegroup-standard-workers"
2025-08-31 02:40:23 [i] waiting for the control plane to become ready
2025-08-31 02:40:23 [v] saved kubeconfig as "/Users/prashanthr/.kube/config"
2025-08-31 02:40:23 [i] no tasks
2025-08-31 02:40:23 [i] all EKS cluster resources for "my-eks-cluster" have been created
2025-08-31 02:40:23 [i] nodegroup "standard-workers" has 2 node(s)
2025-08-31 02:40:23 [i] node "ip-192-168-48-194.ap-south-1.compute.internal" is ready
2025-08-31 02:40:23 [i] node "ip-192-168-69-61.ap-south-1.compute.internal" is ready
2025-08-31 02:40:24 [i] nodegroup "standard-workers" is ready in "standard-workers"
2025-08-31 02:40:24 [i] nodegroup "standard-workers" has 2 node(s)
2025-08-31 02:40:24 [i] node "ip-192-168-48-194.ap-south-1.compute.internal" is ready
2025-08-31 02:40:24 [i] node "ip-192-168-69-61.ap-south-1.compute.internal" is ready
2025-08-31 02:40:24 [v] created 1 managed nodegroup(s) in cluster "my-eks-cluster"
2025-08-31 02:40:24 [v] kubectl command should work with "/Users/prashanthr/.kube/config", try 'kubectl get nodes'
2025-08-31 02:40:24 [v] EKS cluster "my-eks-cluster" in "ap-south-1" region is ready
(base) prashanth@Prashanth-MacBook-Pro-2 Trend % eksctl get cluster --region ap-south-1
NAME      REGION   EKSCLOUD CREATED
my-eks-cluster  ap-south-1  True
(base) prashanth@Prashanth-MacBook-Pro-2 Trend %
(base) prashanth@Prashanth-MacBook-Pro-2 Trend % kubectl get nodes
NAME           STATUS   ROLES   AGE     VERSION
ip-192-168-48-194.ap-south-1.compute.internal  Ready   <none>  4m53s  v1.28.15-eks-3abbe1
ip-192-168-69-61.ap-south-1.compute.internal  Ready   <none>  4m49s  v1.28.15-eks-3abbe1
(base) prashanth@Prashanth-MacBook-Pro-2 Trend %

```

The screenshot shows the AWS CloudWatch Metrics interface. A log stream titled 'eksctl-my-eks-cluster-nodegroup-standard-workers' is displayed, with a single log entry from August 31, 2025, at 02:40:23. The log message indicates that the 'standard-workers' node group has 2 nodes, both of which are ready.

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-48-194.ap-south-1.compute.internal	Ready	<none>	4m53s	v1.28.15-eks-3abbe1
ip-192-168-69-61.ap-south-1.compute.internal	Ready	<none>	4m49s	v1.28.15-eks-3abbe1

- Write deployment and service YAML files.

```

TrendStoreOps -- zsh - 176x59

c67554ad9fa1: Pushed
fbba4caac0b7: Pushed
6156cb6ffff: Pushed
6e174226ea69: Pushed
66ce170f7dd8: Pushed
0bc2f07fbf03: Pushed
latest: digest: sha256:2237ea84faf38ead11a9b9a2d1b6f6ad161671150f191540525f775fa74f91d1 size: 856
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % cd k8s
(base) prashanth@Prashanths-MacBook-Pro-2 k8s % nano deployment.yaml
(base) prashanth@Prashanths-MacBook-Pro-2 k8s % cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: trend-deployment
  labels:
    app: trend
spec:
  replicas: 2
  selector:
    matchLabels:
      app: trend
  template:
    metadata:
      labels:
        app: trend
    spec:
      containers:
        - name: trend-containernano deployment.yaml
          image: prash099/prash-trend-store-app:latest
          ports:
            - containerPort: 80
(base) prashanth@Prashanths-MacBook-Pro-2 k8s %
(base) prashanth@Prashanths-MacBook-Pro-2 k8s % nano service.yaml
(base) prashanth@Prashanths-MacBook-Pro-2 k8s % cat service.yaml
apiVersion: v1
kind: Service
metadata:
  name: trend-service
spec:
  selector:
    app: trend
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      type: LoadBalancer
(base) prashanth@Prashanths-MacBook-Pro-2 k8s % cd ..
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % nano Jenkinsfile
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % nano .gitignore
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % cat .gitignore
cat: .git: Is a directory
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % cat .gitignore
# Node.js / frontend build files (optional)
node_modules/
#dist/

```

DOCKER HUB PUSH

```

Trend -- zsh - 176x59

2025-08-31 02:48:23 [+] nodegroup "standard-workers" has 2 nodes(s)
2025-08-31 02:48:23 [+] node "ip-192-168-48-194.ap-south-1.compute.internal" is ready
2025-08-31 02:48:23 [+] node "ip-192-168-49-61.ap-south-1.compute.internal" is ready
2025-08-31 02:48:23 [+] waiting for at least 1 node(s) to become ready in "standard-workers"
2025-08-31 02:48:24 [+] nodegroup "standard-workers" has 2 node(s)
2025-08-31 02:48:24 [+] node "ip-192-168-48-194.ap-south-1.compute.internal" is ready
2025-08-31 02:48:24 [+] node "ip-192-168-49-61.ap-south-1.compute.internal" is ready
2025-08-31 02:48:24 [+] created 1 managed nodegroup(s) in cluster "my-eks-cluster"
2025-08-31 02:48:24 [+] kubectl command should work with "/Users/prashanth/.kube/config", try 'kubectl get nodes'
2025-08-31 02:48:24 [+] EKS cluster "my-eks-cluster" in "ap-south-1" region is ready
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % eksctl get cluster --region ap-south-1
NAME          REGION          EKSCTL CREATED
my-eks-cluster  ap-south-1  True
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % kubectl get nodes
NAME          STATUS   ROLES   AGE     VERSION
ip-192-168-48-194.ap-south-1.compute.internal  Ready   <none>  4m53s  v1.28.15-eks-3abbe1
ip-192-168-49-61.ap-south-1.compute.internal  Ready   <none>  4m49s  v1.28.15-eks-3abbe1
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % kubectl cluster-info
Kubernetes control plane is running at https://:8443
CoreDNS is running at https://:8443
CoreDNS is running at https://:53
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % docker login
USING WEB-BASED LOGIN
Info: To sign in with credentials on the command line, use 'docker login -u <username>'

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

Waiting for authentication in the browser...
Login Succeeded
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
trend-local-app    latest   2237ea84faf3  14 hours ago  98.6MB
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % docker tag trend-local-app:latest prash099/prash-trend-store-app:latest
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % docker push prash099/prash-trend-store-app:latest
The push refers to repository [docker.io/prash099/prash-trend-store-app]
49f3b06c840f: Pushed
6c2c01fdb094: Pushed
04ba7957fd92: Pushed
021cb5923c0e: Pushed
c67554ad9fa1: Pushed
fbba4caac0b7: Pushed
6156cb6ffff: Pushed
6e174226ea69: Pushed
66ce170f7dd8: Pushed
0bc2f07fbf03: Pushed
latest: digest: sha256:2237ea84faf38ead11a9b9a2d1b6f6ad161671150f191540525f775fa74f91d1 size: 856
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %

```

hub.docker.com/repository/docker/prash099/prash-trend-store-app/general

New Building AI Agents is Now Easy.

hub Explore My Hub

Repositories / prash-trend-store-app / General

prash099/prash-trend-store-app

Last pushed less than a minute ago · Repository size: 30.5 MB

Add a description Add a category

General Tags Image Management BETA Collaborators Webhooks Settings

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest	Image		less than 1 day	less than a minute

See all

Using 0 of 1 private repositories. [Get more](#)

Docker commands

To push a new tag to this repository:

```
docker push prash099/prash-trend-store-app:tagname
```

Public view

buildcloud

Build with Docker Build Cloud

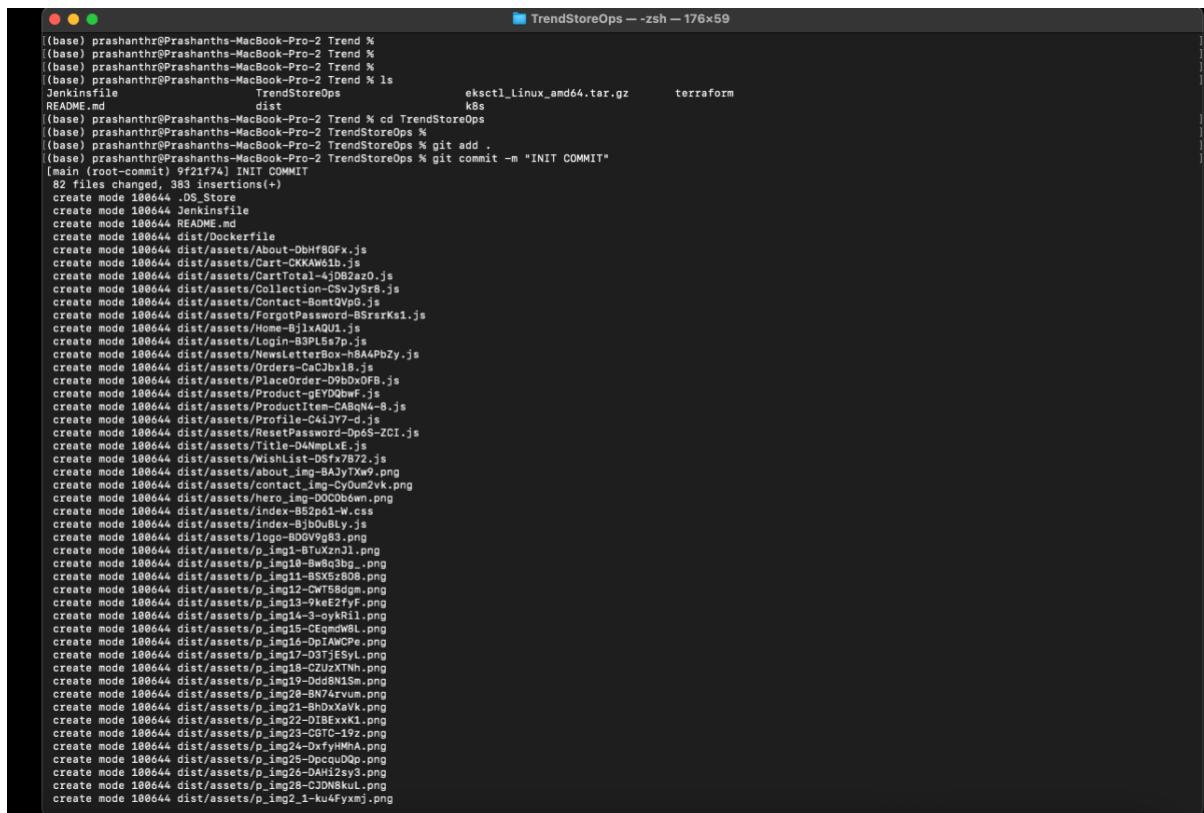
Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all without managing infrastructure.

Version Control:

- Push the codebase to a Git provider (GitHub).
- Add gitignore and dockerignore files and use CLI commands to push code.



```
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend %
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % ls
Jenkinsfile          TrendStoreOps      eksctl_Linux_amd64.tar.gz      terraform
Jenkinsfile.md       TrendStoreOps      k8s
(base) prashanth@Prashanths-MacBook-Pro-2 Trend % cd TrendStoreOps
(base) prashanth@Prashanths-MacBook-Pro-2 TrendStoreOps %
(base) prashanth@Prashanths-MacBook-Pro-2 TrendStoreOps % git add .
(base) prashanth@Prashanths-MacBook-Pro-2 TrendStoreOps % git commit -m "INIT COMMIT"
[main (root-commit) 9f21f74] INIT COMMIT
 82 files changed, 383 insertions(+)
 create mode 100644 .DS_Store
 create mode 100644 Jenkinsfile
 create mode 100644 README.md
 create mode 100644 dist/Dockerfile
 create mode 100644 dist/assets/about-AboutHf8Gfx.js
 create mode 100644 dist/assets/Cart-CKKAw6ib.js
 create mode 100644 dist/assets/CartTotal-4jDB2zv0.js
 create mode 100644 dist/assets/Contact-8oxtVp0.js
 create mode 100644 dist/assets/ForgotPassword-8SpxrKs1.js
 create mode 100644 dist/assets/Home-BjlxAQU1.js
 create mode 100644 dist/assets/Login-B3PL5a7p7.js
 create mode 100644 dist/assets/NewsletterBox-h8A4PbZy.js
 create mode 100644 dist/assets/Orders-CatCbx18.js
 create mode 100644 dist/assets/PlaceOrder-D9DxOfB.js
 create mode 100644 dist/assets/Product-gEVQDwf.js
 create mode 100644 dist/assets/ProductItem-CABoN4-8.js
 create mode 100644 dist/assets/Profile-C4iJY7-d.js
 create mode 100644 dist/assets/ResetPassword-Dp65-ZCI.js
 create mode 100644 dist/assets/Title-D4Wmplxf.js
 create mode 100644 dist/assets/Wishlist-DSIwL.js
 create mode 100644 dist/assets/contact-1qAjTAv2.png
 create mode 100644 dist/assets/contact-img-SyOunvk.png
 create mode 100644 dist/assets/hero-img-DOCCObwn.png
 create mode 100644 dist/assets/index-B52p4l-W.css
 create mode 100644 dist/assets/index-BjbOuUly.js
 create mode 100644 dist/assets/logo-BDGVgB3.js
 create mode 100644 dist/assets/p_img1-BtUxznJl.png
 create mode 100644 dist/assets/p_img10-Bw8G3bg_.png
 create mode 100644 dist/assets/p_img11-BSX5zB0B.png
 create mode 100644 dist/assets/p_img12-CWT58dgm.png
 create mode 100644 dist/assets/p_img13-9keE2fYf.png
 create mode 100644 dist/assets/p_img14-3-oykRil.png
 create mode 100644 dist/assets/p_img15-CEqqWvL.png
 create mode 100644 dist/assets/p_img16-DeIAWCe.png
 create mode 100644 dist/assets/p_img17-DgJzXnL.png
 create mode 100644 dist/assets/p_img18-1q1B-2ZUXXTNv.png
 create mode 100644 dist/assets/p_img19-0ddBN1Ss.png
 create mode 100644 dist/assets/p_img20-BN7Arvum.png
 create mode 100644 dist/assets/p_img21-BnDxXvK.png
 create mode 100644 dist/assets/p_img22-0IBExxX1.png
 create mode 100644 dist/assets/p_img23-CGTO-4z2.png
 create mode 100644 dist/assets/p_img24-DxfyHMhA.png
 create mode 100644 dist/assets/p_img25-DpcquQq.png
 create mode 100644 dist/assets/p_img26-DAH1zsy3.png
 create mode 100644 dist/assets/p_img28-CJDN8Kul.png
 create mode 100644 dist/assets/p_img2_1-ku4Fyxmj.png
```

```
  TrendStoreOps -- zsh -- 176x59
create mode 100644 dist/assets/p_img18-CIUuXTNh.png
create mode 100644 dist/assets/p_img19-ODdBN1Sm.png
create mode 100644 dist/assets/p_img20-BN74rvun.png
create mode 100644 dist/assets/p_img21-BHDXxAvK.png
create mode 100644 dist/assets/p_img22-D1BEExX1.png
create mode 100644 dist/assets/p_img23-CGTC-19z.png
create mode 100644 dist/assets/p_img24-DxfyHNhA.png
create mode 100644 dist/assets/p_img25-DqfquDq.png
create mode 100644 dist/assets/p_img26-DAH12sy3.png
create mode 100644 dist/assets/p_img28-CJDN8kul.png
create mode 100644 dist/assets/p_img1-1kuufYxmj.png
create mode 100644 dist/assets/p_img2-2-qkFhpUu.png
create mode 100644 dist/assets/p_img2-CeauMm.png
create mode 100644 dist/assets/p_img3-4-0nGIA9V-.png
create mode 100644 dist/assets/p_img3-Bmpg8mc.png
create mode 100644 dist/assets/p_img31-CWAriNUF.png
create mode 100644 dist/assets/p_img34-BXNSk9Lc.png
create mode 100644 dist/assets/p_img35-DHYZ5dxs.png
create mode 100644 dist/assets/p_img36-B11Ky1Pb.png
create mode 100644 dist/assets/p_img39-CbowV562.png
create mode 100644 dist/assets/p_img4-CYIVcKUm.png
create mode 100644 dist/assets/p_img40-CSBZ2iZB.png
create mode 100644 dist/assets/p_img41-IMY9mpgt.png
create mode 100644 dist/assets/p_img43-CXXL7sau.png
create mode 100644 dist/assets/p_img44-CwJd9P6l.png
create mode 100644 dist/assets/p_img45-D6kVi7rp.png
create mode 100644 dist/assets/p_img46-D9wAqPh.png
create mode 100644 dist/assets/p_img47-DG-7V-.png
create mode 100644 dist/assets/p_img48-DRWHR8.png
create mode 100644 dist/assets/p_img49-OphJrG1.png
create mode 100644 dist/assets/p_imgs-BMWJX1_.png
create mode 100644 dist/assets/p_imgs-Co-7CHi5.png
create mode 100644 dist/assets/p_img51-FRQODDGA.png
create mode 100644 dist/assets/p_img52-CY-YbGLu.png
create mode 100644 dist/assets/p_img6-B2VXR-Iw.png
create mode 100644 dist/assets/p_img7-Ds_MiPCY.png
create mode 100644 dist/assets/p_img8-BeUAw8.png
create mode 100644 dist/assets/p_img9-0mR8Ahyl.png
create mode 100644 dist/assets/razorpay_logo-DrY6yMwi.png
create mode 100644 dist/index.html
create mode 100644 dist/vite.svg
create mode 100644 k8s/deployment.yaml
create mode 100644 k8s/service.yaml
create mode 100644 terraform/.tfstate
create mode 100644 terraform/outputs.tf
create mode 100644 terraform/variables.tf
(base) prashanth@Prashanth-MacBook-Pro-2 TrendStoreOps % git push
Enter passphrase for key '/Users/prashanth/.ssh/id_ed25519':
Enumerating objects: 88, done.
Counting objects: 100% (88/88), done.
Delta compression using up to 10 threads
Compressing objects: 100% (87/87), done.
Writing objects: 100% (88/88), 8.59 MiB | 2.91 MiB/s, done.
Total 88 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:Prash099/TrendStoreOps.git
 * [new branch]      main -> main
(base) prashanth@Prashanth-MacBook-Pro-2 TrendStoreOps %
(base) prashanth@Prashanth-MacBook-Pro-2 TrendStoreOps %
```

github.com/Prash099/TrendStoreOps/tree/main

Prash099 / TrendStoreOps

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Prash099 Update Jenkinsfile

1 Branch 0 Tags

Go to file Add file Code

19e3459 · 13 minutes ago 22 Commits

dist INIT COMMIT 2 hours ago

k8s Update deployment.yaml 1 hour ago

terraform INIT COMMIT 2 hours ago

Dockerfile Update Dockerfile 1 hour ago

Jenkinsfile Update Jenkinsfile 16 minutes ago

README.md INIT COMMIT 2 hours ago

build.sh Update build.sh 1 hour ago

deploy.sh Update deploy.sh 1 hour ago

docker-compose.yml Create docker-compose.yml 1 hour ago

README

TrendStoreOps

About

No description, website, or topics provided.

Readme Activity 0 stars 0 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

Languages

HCL 85.8% Shell 12.6% Dockerfile 1.8%

Suggested workflows

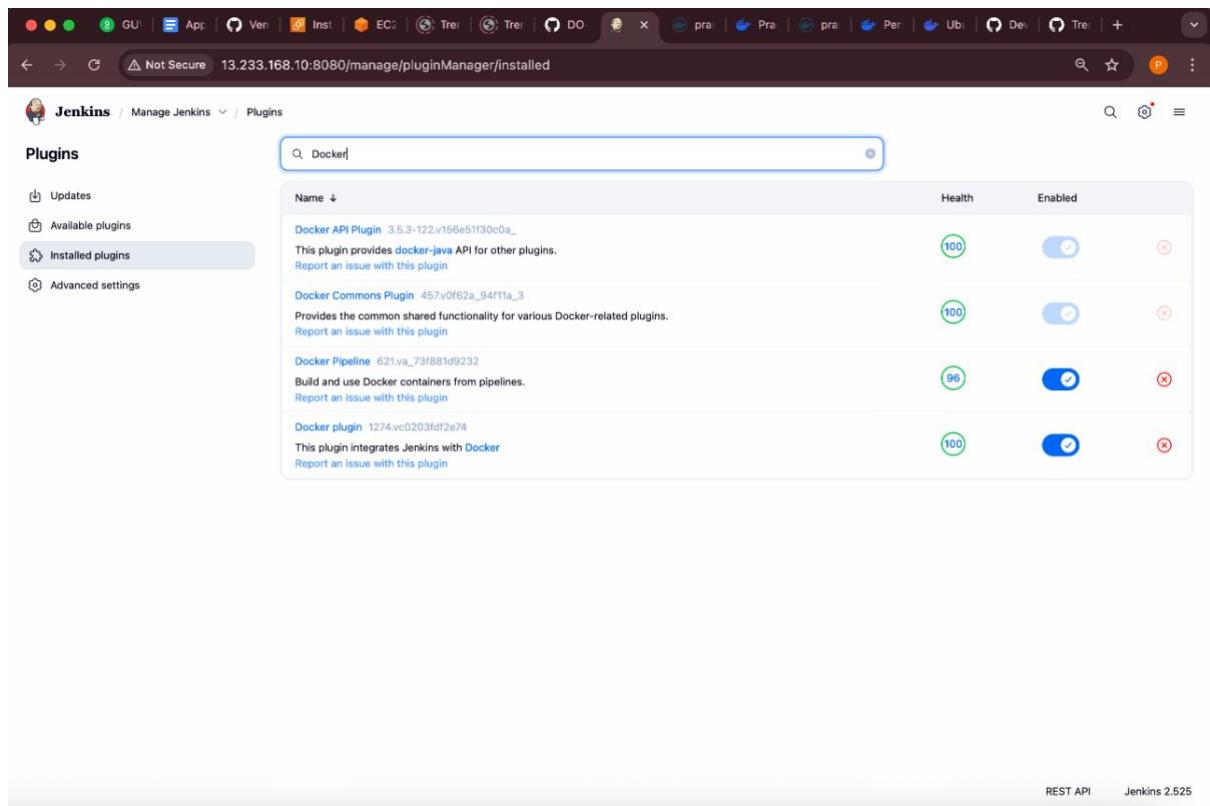
Based on your tech stack

Publish Docker Container Configure

Build, test and push Docker image to GitHub Packages.

Jenkins:

- Install Jenkins and necessary plugins (Docker, Git, Kubernetes, Pipeline) for build, push & deploy applications.



The screenshot shows the Jenkins plugin manager interface. The URL in the browser is `13.233.168.10:8080/manage/pluginManager/installed`. The search bar at the top contains the text "Docker". On the left, there is a sidebar with options: Updates, Available plugins, **Installed plugins**, and Advanced settings. The main area displays a table of installed plugins related to Docker:

Name	Health	Enabled
Docker API Plugin	100	Enabled
Docker Commons Plugin	100	Enabled
Docker Pipeline	96	Enabled
Docker plugin	100	Enabled

At the bottom right of the page, there are links for REST API and Jenkins 2.525.

Not Secure 13.233.168.10:8080/manage/pluginManager/installed

Jenkins / Manage Jenkins / Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings

GitHub

Name	Health	Enabled
Generic Webhook Trigger Plugin 2.4.1	100	✓
GitHub API Plugin 1.321-488.v9b_c0da_9533fb	100	✓
GitHub Authentication plugin 651.v135e939e8b_60	100	✓
GitHub Branch Source Plugin 1844.v4a_9883d49126	100	✓
GitHub Custom Notification Context SCM Behaviour 45.vBef831029589	94	✓
GitHub Integration Plugin 0.7.2	87	✓
GitHub plugin 1.44.0	97	✓
Pipeline implementation for Blue Ocean 1.27.21	97	✓
Pipeline: GitHub 2.8-162.382498405fdc	91	✓

Not Secure 13.233.168.10:8080/manage/pluginManager/installed

Jenkins / Manage Jenkins / Plugins

Plugins

Updates Available plugins Installed plugins Advanced settings

Kube[

Name	Health	Enabled
Kubernetes :: Pipeline :: DevOps Steps 1.6	55	✓
Kubernetes CLI Plugin 1.364.vadef8cb8b823	100	✓
Kubernetes Client API Plugin 7.3.1-256.v788a_0b_78711a	96	✓
Kubernetes Credentials Plugin 206.vde31a_b_0f71a_c	100	✓
Kubernetes Credentials Provider 1.290.v656494531260	94	✓
Kubernetes plugin 4369.va_9a_89327dd35	100	✓

Not Secure 13.233.168.10:8080/manage/pluginManager/installed

Plugins

Updates Available plugins Installed plugins Advanced settings

Search: Pipeline

Name	Health	Enabled
Build Pipeline Plugin 2.0.2	52	<input checked="" type="checkbox"/>
Checks API plugin 373.vfe7645102093	98	<input checked="" type="checkbox"/>
Docker Pipeline 621.va_73f881d9232	96	<input checked="" type="checkbox"/>
Git Pipeline for Blue Ocean 1.27.21	97	<input checked="" type="checkbox"/>
Kubernetes :: Pipeline Steps 1.6	55	<input checked="" type="checkbox"/>
Pipeline 608.v67378e9d3db_1	100	<input checked="" type="checkbox"/>
Pipeline Graph Analysis Plugin 241.vc3d48fb_b_2582	100	<input checked="" type="checkbox"/>

Warning: The currently installed plugin version may not be safe to use. Please review the following security notices:
• [Stored XSS vulnerability](#)

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.
Report an issue with this plugin

This plugin defines an API for Jenkins to publish checks to SCM platforms.
Report an issue with this plugin

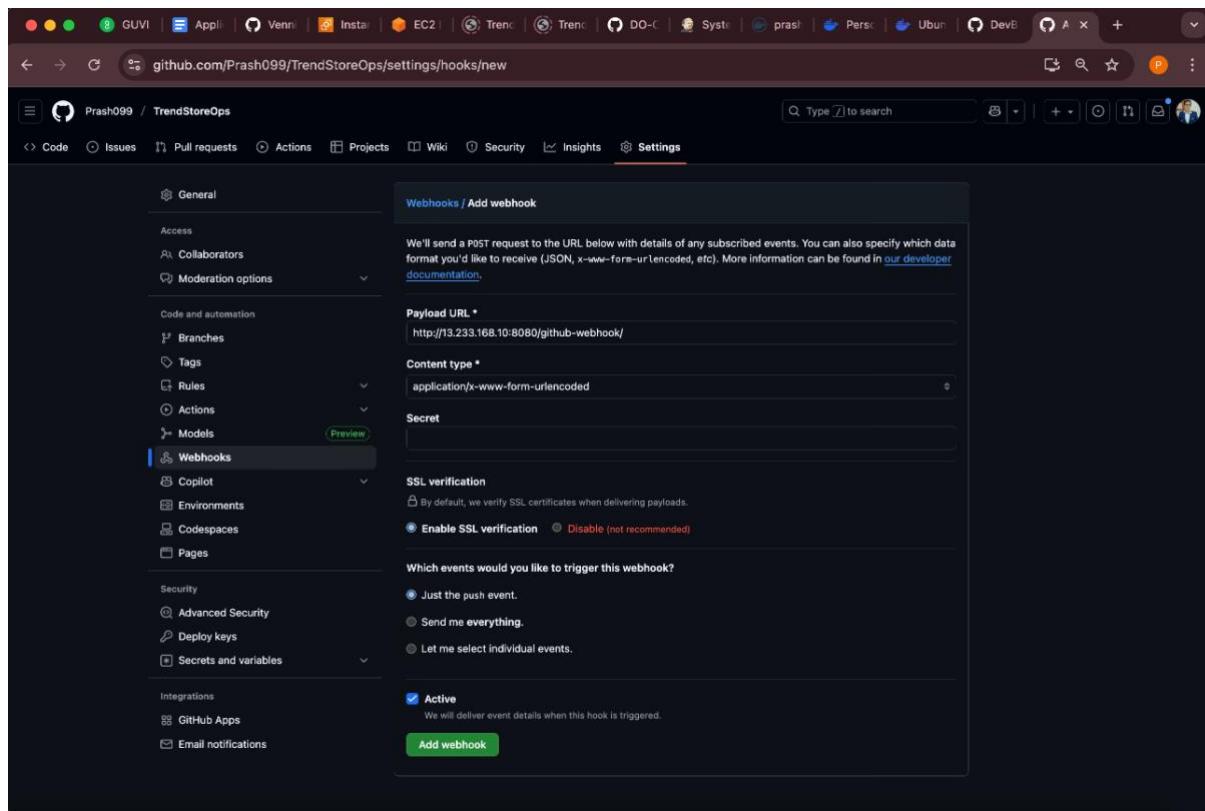
Build and use Docker containers from pipelines.
Report an issue with this plugin

Report an issue with this plugin

A suite of plugins that lets you orchestrate automation, simple or complex. See [Pipeline as Code](#) with Jenkins for more details.
Report an issue with this plugin

Provides a REST API to access pipeline and pipeline run data.
Report an issue with this plugin

- Setup Github and jenkins integration using github webhook build trigger for auto build for every commit.



The screenshot shows the GitHub settings page for the repository 'Prash099/TrendStoreOps'. The user is on the 'Webhooks' tab. A success message states: 'Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at <https://docs.github.com/webhooks/ping-event>'. Below this, there is a list of webhooks with one entry: 'http://13.233.168.10:8080/github... (push)'. There are 'Edit' and 'Delete' buttons next to it. The sidebar on the left includes sections for General, Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Rules, Actions, Models), Webhooks (selected), Copilot, Environments, Codespaces, Pages, Security (Advanced Security, Deploy keys, Secrets and variables), Integrations (GitHub Apps, Email notifications), and Settings.

JENKINS PIPELINE SETUP

The screenshot shows the Jenkins 'New Item' setup page. The title is 'New Item'. It asks for an item name, which is 'TrendStoreApp'. It then asks to select an item type. The 'Pipeline' option is highlighted with a blue border. Other options shown are 'Freestyle project', 'Multi-configuration project', 'Folder', and 'Multibranch Pipeline'. At the bottom is a blue 'OK' button.

Not Secure 13.233.168.10:8080/job/TrendStoreApp/configure

Jenkins / TrendStoreApp / Configuration

Configure General

Description: TrendStoreAppFrontend

Enabled: Enabled

General Options:

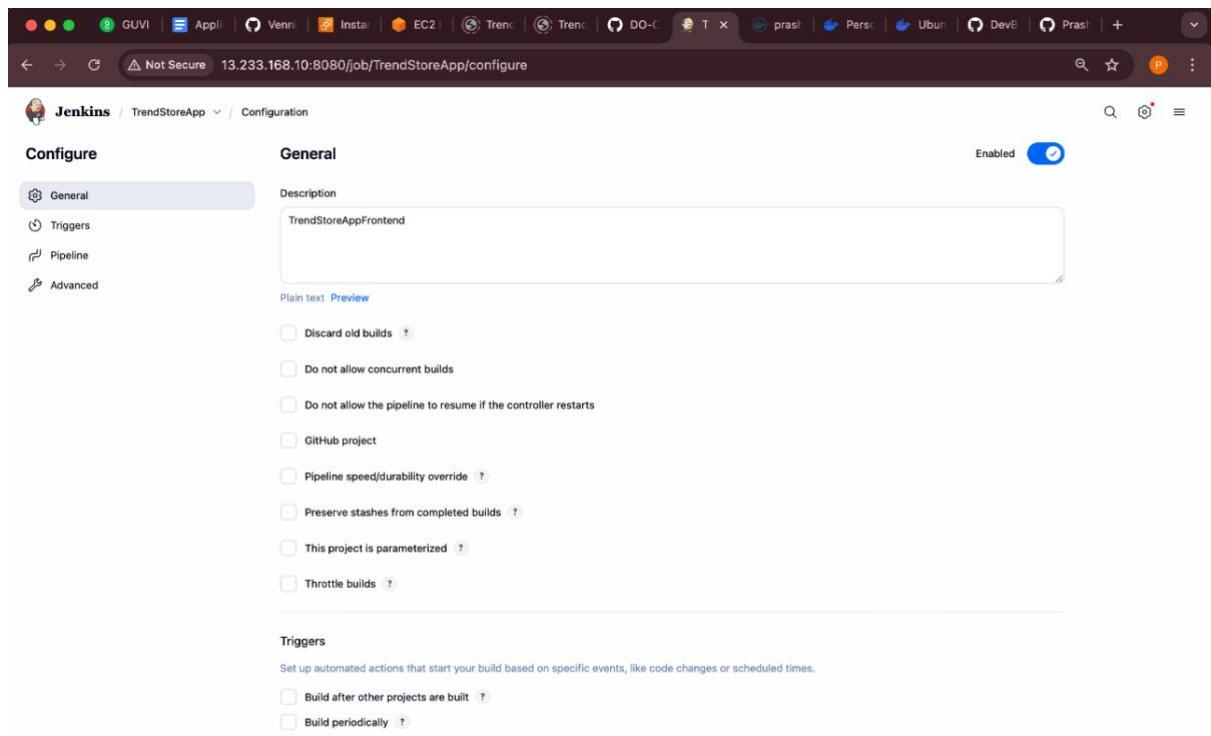
- Discard old builds
- Do not allow concurrent builds
- Do not allow the pipeline to resume if the controller restarts
- GitHub project
- Pipeline speed/durability override
- Preserve stashes from completed builds
- This project is parameterized
- Throttle builds

Triggers:

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- Build after other projects are built
- Build periodically
- Generic Webhook Trigger

Save Apply



Not Secure 13.233.168.10:8080/job/TrendStoreApp/configure

Jenkins / TrendStoreApp / Configuration

Configure Triggers

General Options:

- Generic Webhook trigger
- GitHub Branches
- GitHub Pull Requests
- GitHub hook trigger for GITScm polling
- Poll SCM

Pipeline:

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

Definition: Pipeline script from SCM

SCM: Git

Repositories:

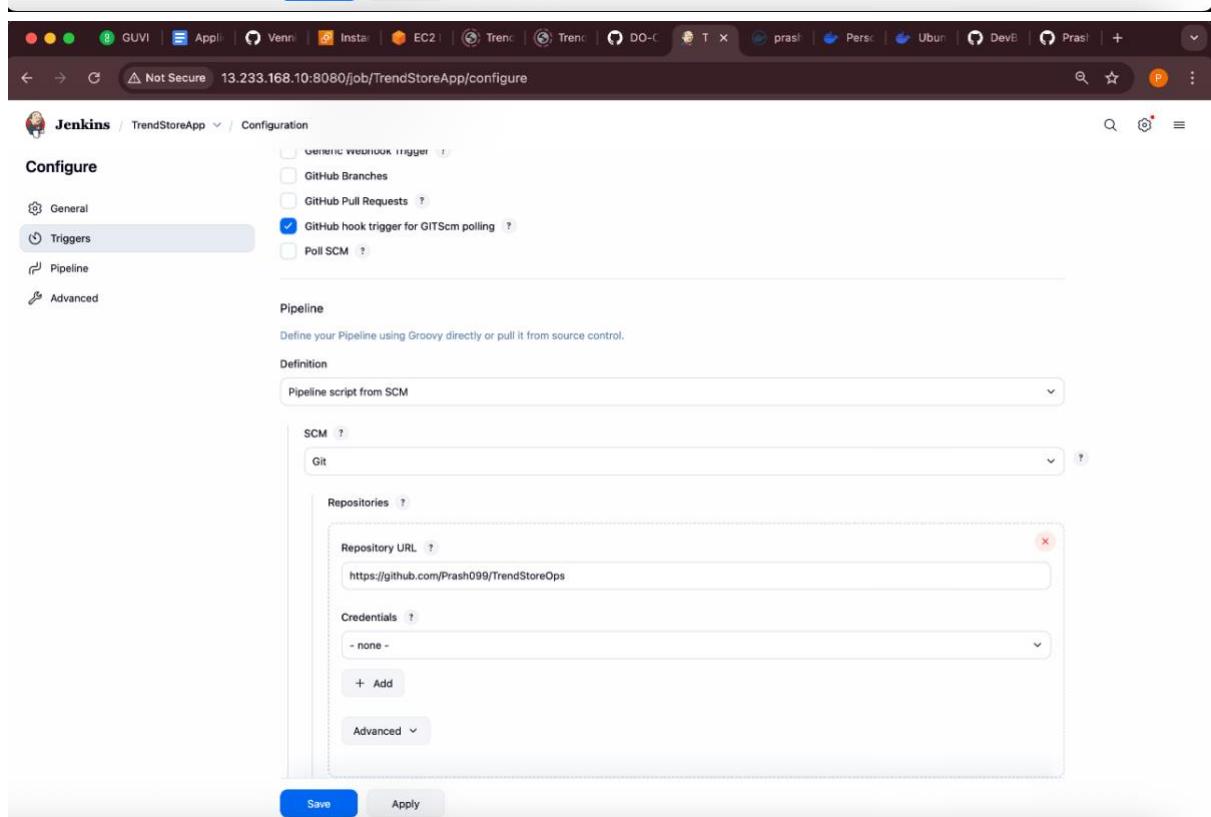
Repository URL: https://github.com/Prash099/TrendStoreOps

Credentials: - none -

+ Add

Advanced

Save Apply



Not Secure 13.233.168.10:8080/job/TrendStoreApp/configure

Jenkins / TrendStoreApp / Configuration

Configure

General Triggers Pipeline Advanced

+ Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

+ Add Branch

Repository browser ?

(Auto)

Additional Behaviours

+ Add

Script Path ?

Jenkinsfile

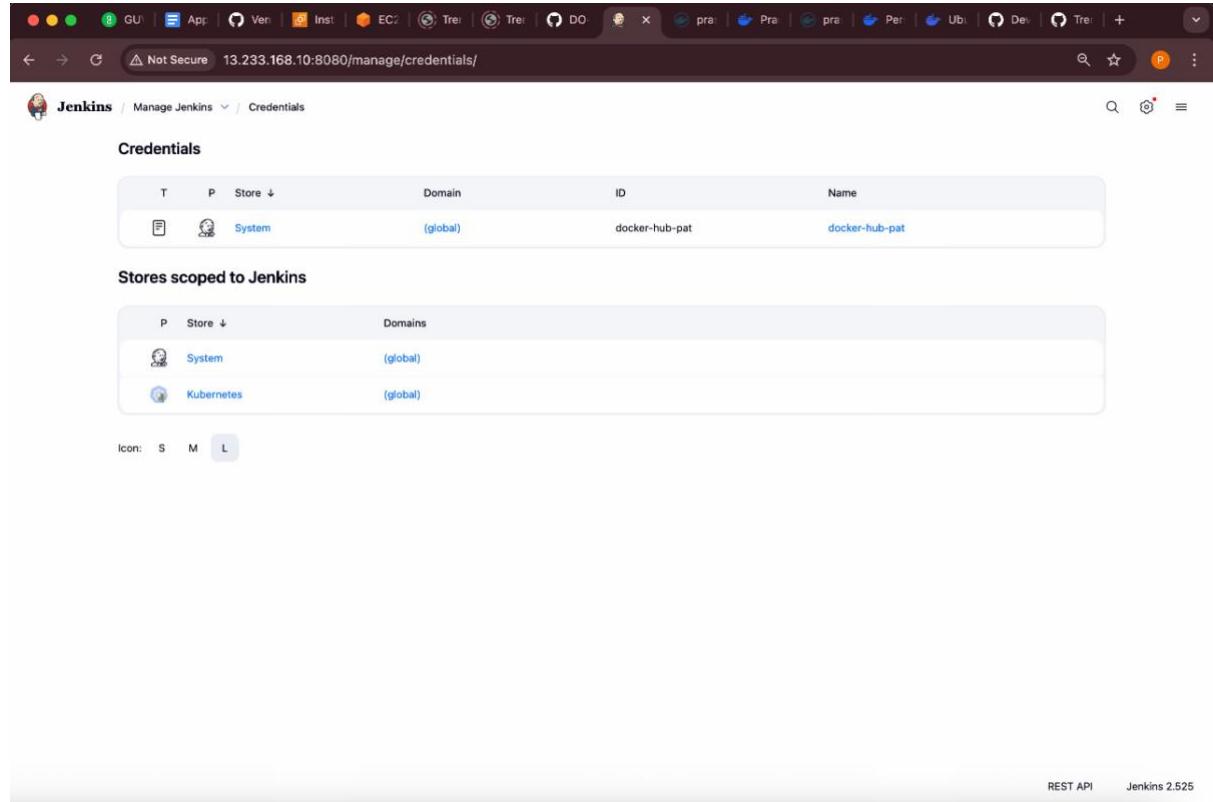
Lightweight checkout ?

Save Apply

This screenshot shows the Jenkins Pipeline configuration page for the 'TrendStoreApp' job. The 'Pipeline' tab is active. In the 'Branches to build' section, the branch specifier is set to '*/main'. The 'Repository browser' is set to '(Auto)'. Under 'Additional Behaviours', the 'Lightweight checkout' option is checked. The 'Script Path' is set to 'Jenkinsfile'. At the bottom, there are 'Save' and 'Apply' buttons.

- Create a declarative pipeline script and pipeline project to build, push & deploy using CI-CD.

SETTING-UP-JENKINS DOCKER PAT



The screenshot shows the Jenkins 'Credentials' management interface. At the top, the URL is 13.233.168.10:8080/manage/credentials/. The page title is 'Jenkins / Manage Jenkins / Credentials'. Below the title, there's a table titled 'Credentials' with columns: T, P, Store, Domain, ID, and Name. One entry is visible: a System credential with domain '(global)', ID 'docker-hub-pat', and name 'docker-hub-pat'. Below the table, there's a section titled 'Stores scoped to Jenkins' with a table showing two stores: 'System' and 'Kubernetes', both with domain '(global)'. At the bottom left, there are icons for S, M, and L. On the right side, there are links for 'REST API' and 'Jenkins 2.525'.

JENKINS PIPELINE

The screenshot shows the Jenkins 'New Item' creation dialog. The title bar says 'Not Secure 13.233.168.10:8080/newJob'. The main area is titled 'New Item' and has a sub-section 'Enter an item name' with the value 'TrendStoreApp'. Below it, 'Select an item type' is set to 'Pipeline'. Other options shown include 'Freestyle project', 'Multi-configuration project', 'Folder', and 'Multibranch Pipeline'. A large blue 'OK' button is at the bottom.

The screenshot shows the Jenkins 'Configure' dialog for the 'TrendStoreApp' job. The left sidebar has tabs for 'General', 'Triggers', 'Pipeline', and 'Advanced', with 'General' selected. The 'General' tab includes a 'Description' field containing 'TrendStoreAppFrontend', an 'Enabled' toggle switch (which is checked), and several configuration options like 'Discard old builds', 'Do not allow concurrent builds', etc. The 'Triggers' tab lists triggers like 'Build after other projects are built', 'Build periodically', and 'Generic Webhook Trigger'. At the bottom are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins job configuration page for 'TrendStoreApp'. The 'Triggers' section is selected. Under 'Generic Webhook trigger', there are several options: 'GitHub Branches', 'GitHub Pull Requests', 'GitHub hook trigger for GITScm polling' (which is checked), and 'Poll SCM'. Below the triggers, the 'Pipeline' section is shown with the definition 'Pipeline script from SCM'. The 'SCM' dropdown is set to 'Git'. The 'Repositories' section shows a single repository with the URL <https://github.com/Prash099/TrendStoreOps>. There are buttons for 'Save' and 'Apply' at the bottom.

The screenshot shows the Jenkins job configuration page for 'TrendStoreApp'. The 'Pipeline' section is selected. The 'Definition' dropdown is set to 'Pipeline script from SCM'. The 'SCM' dropdown is set to 'Git'. The 'Repositories' section shows a single repository with the URL <https://github.com/Prash099/TrendStoreOps>. The 'Branches to build' section has a 'Branch Specifier (blank for 'any')' set to */main. The 'Repository browser' dropdown is set to '(Auto)'. Under 'Additional Behaviours', there is a 'Script Path' set to 'Jenkinsfile' and a checked checkbox for 'Lightweight checkout'. There are buttons for 'Save' and 'Apply' at the bottom.

- Deploy using kubectl via Jenkins.

```
pipeline {  
    agent any  
  
    environment {  
        DOCKERHUB_CREDENTIALS = credentials('docker-hub-pat')  
        KUBECONFIG = '/var/lib/jenkins/.kube/config'  
        IMAGE_NAME = 'prash099/prash-trend-store-app'  
        IMAGE_TAG = 'latest'  
    }  
  
    stages {  
        stage('Clone Repository') {  
            steps {  
                git branch: 'main', url: 'https://github.com/Prash099/TrendStoreOps.git'  
            }  
        }  
  
        stage('Checkout') {  
            steps {  
                checkout scm  
            }  
        }  
  
        stage('Build Docker Image') {  
            steps {  
                // Build Docker Image steps  
            }  
        }  
    }  
}
```

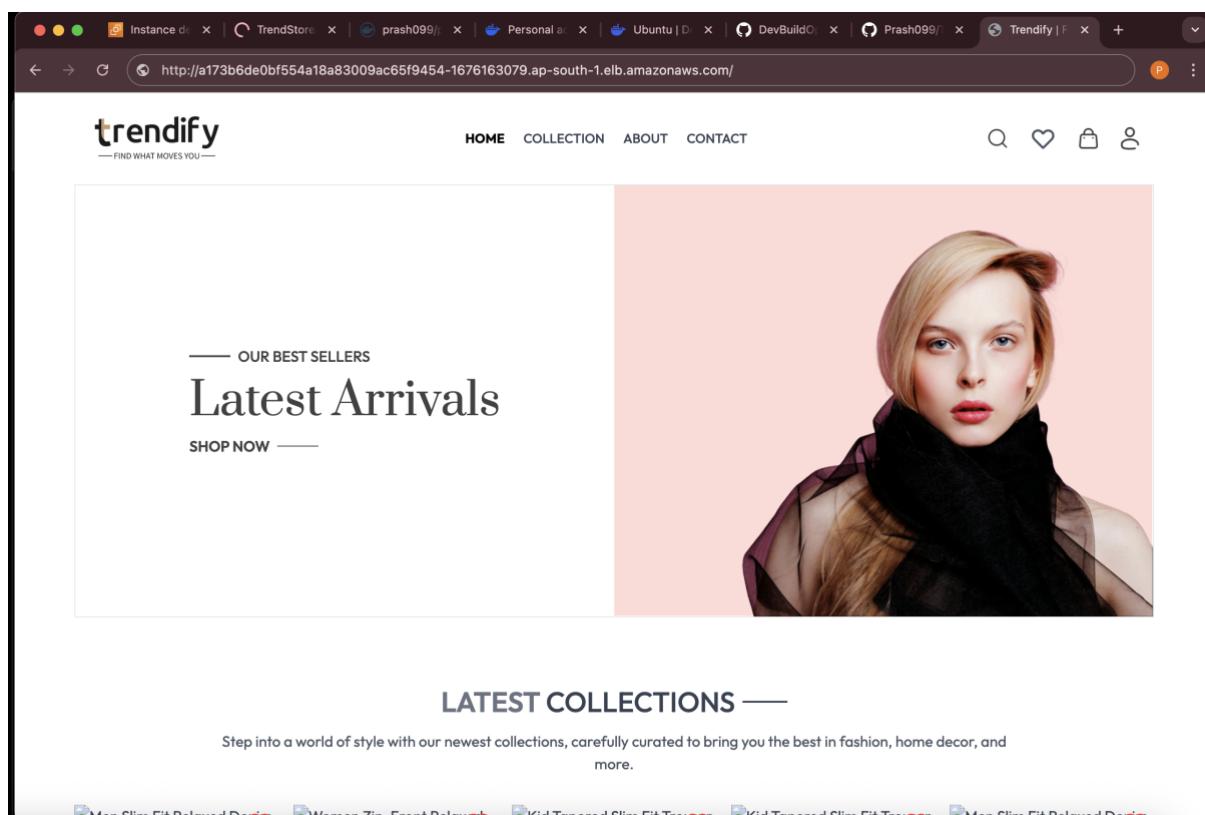
```
        echo "Building Docker image..."
        sh 'chmod +x build.sh'
        sh './build.sh'
    }

}

stage('Deploy') {
    steps{
        script{
            echo "Deploying dev branch..."
            sh 'chmod +x deploy.sh'
            sh './deploy.sh'
        }
    }
}

stage('Deploy to EKS') {
    steps{
        // Apply both deployment and service manifests
        script{
            sh 'kubectl get nodes'
            echo "kubectl get nodes"
        }
        sh "kubectl apply -f k8s/deployment.yaml"
        sh "kubectl apply -f k8s/service.yaml"
    }
}
```

```
stage('Verify Deployment') {  
    steps {  
        sh 'kubectl get pods'  
        sh 'kubectl get svc'  
    }  
}  
}  
}
```



=====

=====