

Trend App Deployment - Screenshots Document

1. Git Cloning

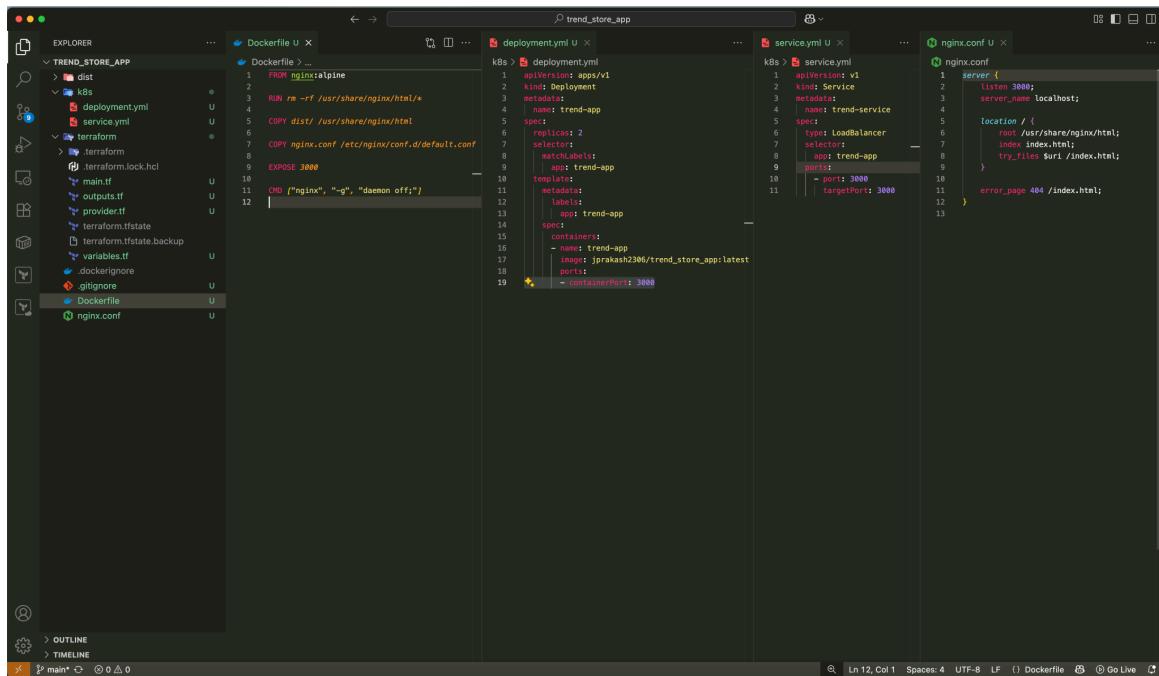
1a) Git cloning the react application build output folder



```
prakashMac ~ % git clone https://github.com/VennilaV12/Trend.git trend_store_app
Cloning into 'trend_store_app...'...
remote: Enumerating objects: 1086, done.
remote: Counting objects: 1086 (1/1), done.
remote: Total 77 (delta 0), reused 0 (delta 0), pack-reused 76 (from 1)
Receiving objects: 100% (77/77) 8.88 MiB | 7.54 MiB/s, done.
Resolving deltas: 100% (76/76), done.
prakashMac ~ % cd trend_store_app
prakashMac trend_store_app % code .
```

2. Docker Build and Push

2a) Adding Config files to project



The screenshot shows a code editor interface with four tabs open:

- Dockerfile**:

```
FROM nginx:alpine
RUN rm -rf /usr/share/nginx/html/*
COPY dist /usr/share/nginx/html/
EXPOSE 3000
CMD ["nginx", "-g", "daemon off;"]
```
- deployment.yaml**:

```
apiVersion: v1
kind: Deployment
metadata:
  name: trend-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: trend-app
  template:
    metadata:
      labels:
        app: trend-app
    spec:
      containers:
        - name: trend-app
          image: jprakash2306/trend_store_app:latest
          ports:
            - containerPort: 3000
```
- service.yaml**:

```
apiVersion: v1
kind: Service
metadata:
  name: trend-service
spec:
  selector:
    app: trend-app
  ports:
    - port: 3000
      targetPort: 3000
  type: LoadBalancer
```
- nginx.conf**:

```
server {
  listen 3000;
  server_name localhost;
  location / {
    root /usr/share/nginx/html;
    index index.html;
    try_files $uri /index.html;
  }
  error_page 404 /index.html;
}
```

2b) Build docker image using Dockerfile and running on local machine

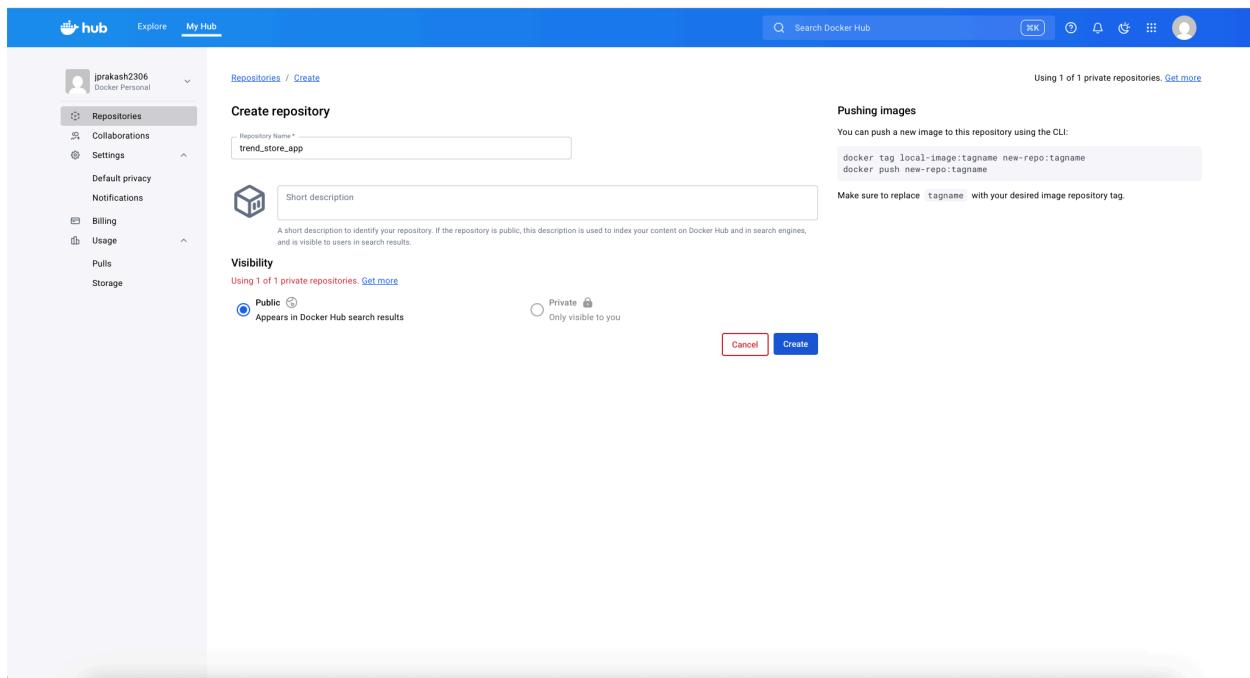
```
prakash@mac: trend_store_app % docker build -t trend-app:latest .
[+] 1/1 [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/nginx:alpine
[+] 1/1 [internal] load .dockerignore
=> [internal] transfer context: 2B
[+] 1/1 FROM docker.io/library/nginx:alpine@sha256:a24516af16b852e33b7682d5ef8acbd5d13fe08fecadc7ed98685ba5e3b26ab8
=> [internal] resolve docker.io/library/nginx:alpine@sha256:a24516af16b852e33b7682d5ef8acbd5d13fe08fecadc7ed98685ba5e3b26ab8
[+] 1/1 [internal] load build context: 3.9KB
=> CACHED [2/4] RUN rm -rf /usr/share/nginx/html/*
=> CACHED [3/4] COPY dist/ /usr/share/nginx/html
=> CACHED [4/4] COPY ./etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf
[+] 1/1 exporting to image
=> 1/1 exporting layers
=> 1/1 exporting manifest sha256:dc1c15db1c99d8d89978fb9e29437d088c40ceb64d399d568749411f3838
=> 1/1 exporting config sha256:8cd9712aa3eb8e02a7a2fa75bb8661e1e2d20f77335e7aecd4b1a5b09896acc
[+] 1/1 exporting manifest sha256:6571da2731d8a2607ff4d11a5a540972214988794d0943788797c33b2a85
=> 1/1 naming to docker.io/library/trend-app:latest
=> 1/1 unpacking to docker.io/library/trend-app:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/v8yua1k1ccphghrex97k80p6
prakash@mac: trend_store_app % docker run -p 3000:3000 trend-app:latest
/docker-entrypoint.sh: /docker-entrypoint.sh is empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for a shell in /bin/sh, /usr/bin/sh, /bin/bash, /usr/bin/bash
/docker-entrypoint.sh: Launching /docker-entrypoint.sh/18-listen-on-ipv4-by-default.sh
18-listen-on-ipv4-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
18-listen-on-ipv4-by-default.sh: info: /etc/nginx/conf.d/default.conf differs from the packaged version
/docker-entrypoint.sh: /etc/nginx/conf.d/default.conf is up-to-date. No action required.
/docker-entrypoint.sh: Launching /docker-entrypoint.sh/20-exists-subst-on-tslates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.sh/38-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration completed, ready for start up
2025/08/25 18:53:52 [notice] 1#1: using "epoll" event method
2025/08/25 18:53:52 [notice] 1#1: nginx/1.29.1
2025/08/25 18:53:52 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0)
2025/08/25 18:53:52 [notice] 1#1: OS: Linux 6.19.14-linuskit
2025/08/25 18:53:52 [notice] 1#1: getrlimit(R_NOFILE): 1048576:1048576
2025/08/25 18:53:52 [notice] 1#1: start worker process
2025/08/25 18:53:52 [notice] 1#1: start worker process 29
2025/08/25 18:53:52 [notice] 1#1: start worker process 30
2025/08/25 18:53:52 [notice] 1#1: start worker process 31
2025/08/25 18:53:52 [notice] 1#1: start worker process 32
2025/08/25 18:53:52 [notice] 1#1: start worker process 33
2025/08/25 18:53:52 [notice] 1#1: start worker process 34
2025/08/25 18:53:52 [notice] 1#1: start worker process 35
2025/08/25 18:53:52 [notice] 1#1: start worker process 36
2025/08/25 18:53:52 [notice] 1#1: start worker process 37
2025/08/25 18:53:52 [notice] 1#1: start worker process 38
2025/08/25 18:53:52 [notice] 1#1: start worker process 39
2025/08/25 18:53:52 [notice] 1#1: start worker process 40
2025/08/25 18:53:52 [notice] 1#1: start worker process 41
2025/08/25 18:53:52 [notice] 1#1: start worker process 42
2025/08/25 18:53:52 [notice] 1#1: start worker process 43
2025/08/25 18:53:52 [notice] 1#1: start worker process 44
```

2c) Running on local machine on port 3000

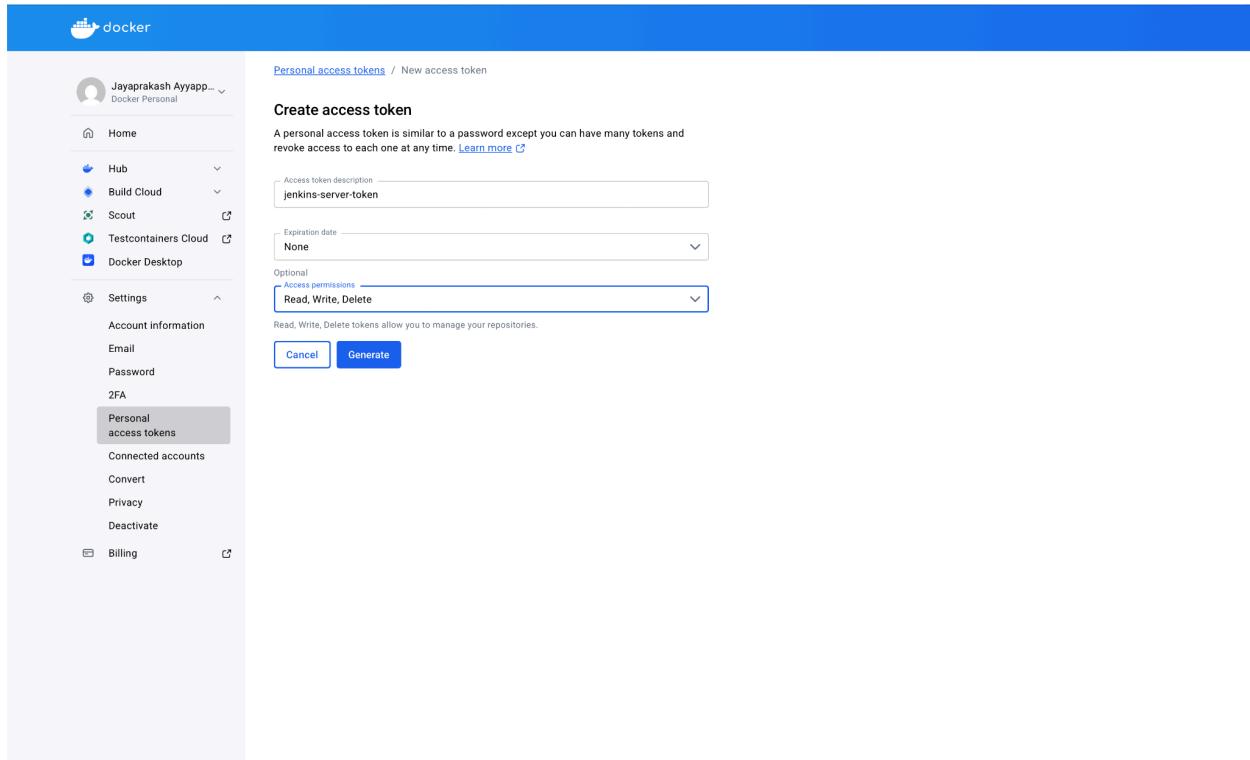
The screenshot shows a web browser window with the URL `localhost:3000`. The page features a header with the **trendify** logo and navigation links for HOME, COLLECTION, ABOUT, and CONTACT. Below the header, there's a section titled "OUR BEST SELLERS" with a "Latest Arrivals" heading and a "SHOP NOW" button. To the right, there's a large image of a woman with blonde hair wearing a black sheer scarf. At the bottom of the page, there's a "LATEST COLLECTIONS" section with a sub-headline "Step into a world of style with our newest collections, carefully curated to bring you the best in fashion, home decor, and more." and a note "Loading latest collections...".

2d) Create new DockerHub Repo



The screenshot shows the 'Create repository' page on DockerHub. On the left, there's a sidebar with user information (jprakash2306) and navigation links like Repositories, Collaborations, Settings, Default privacy, Notifications, Billing, Usage, Pulls, and Storage. The main area has a 'Create repository' form. It includes fields for 'Repository Name' (trend_store_app), 'Short description' (A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.), 'Visibility' (set to 'Public'), and 'Pushing images' instructions. At the bottom are 'Cancel' and 'Create' buttons.

2e) Create new AccessToken to Push the image



The screenshot shows the 'Create access token' page on Docker. The left sidebar includes 'Home', 'Hub', 'Build Cloud', 'Scout', 'Testcontainers Cloud', 'Docker Desktop', 'Settings' (with 'Personal access tokens' selected), 'Account information', 'Email', 'Password', '2FA', 'Connected accounts', 'Convert', 'Privacy', 'Deactivate', and 'Billing'. The main form for creating a token has fields for 'Access token description' (jenkins-server-token), 'Expiration date' (None), 'Optional Access permissions' (Read, Write, Delete), and a note about managing repositories. At the bottom are 'Cancel' and 'Generate' buttons.

3. Terraform

3a) Create new SSh key for new EC2 jenkins-server

The screenshot shows the AWS CloudShell interface with the following details:

- Region:** Asia Pacific (Mumbai)
- Account ID:** 8281-7133-2440
- Key Pair:** Jenkins-server
- Key Pair Type:** RSA
- Private Key File Format:** .pem (selected)
- Tags:** Jenkins-server
- Create Key Pair:** Button at the bottom right.

3b) Adding permission to ssh key

```
prakash@mac ~ % mv Downloads/jenkins-server.pem ~/.ssh
prakash@mac ~ % cd .ssh
prakash@mac .ssh % ls
bitbucket      config          github_key.pub      id_ed25519.pub      known_hosts
bitbucket.pub   github_key     id_ed25519        jenkins-server.pem  known_hosts.old
prakash@mac .ssh % chmod 400 jenkins-server.pem
prakash@mac .ssh %
```

3c) terraform init command

```
prakash@mac trend_store_app % cd terraform
prakash@mac terraform % terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.10.0...
- Installed hashicorp/aws v6.10.0 (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.
prakash@mac terraform % terraform validate
Success! The configuration is valid.

prakash@mac terraform %
```

3d) terraform validate command

```
prakash@mac trend_store_app % cd terraform
prakash@mac terraform % terraform validate
      )
  + vpc_id          = (known after apply)

# aws_subnet.trend_subnet will be created
+ resource "aws_subnet" "trend_subnet" {
    + arn                = (known after apply)
    + assign_ipv6_address_on_creation   = false
    + availability_address            = (known after apply)
    + availability_zone_id          = (known after apply)
    + cidr_block                   = "19.0.1.0/24"
    + ipv6_association_id           = (known after apply)
    + enable_resource_name_dns_a_record_on_launch = false
    + enable_resource_name_dns_aaaa_record_on_launch = false
    + id                            = (known after apply)
    + ipv6_cidr_block_association_id = (known after apply)
    + ipv6_native                  = true
    + map_public_ip_on_launch       = (known after apply)
    + owner_id                     = (known after apply)
    + private_dns_hostname_type_on_launch = (known after apply)
    + region                       = "ap-south-1"
    + tags                         = {
        + "Name" = "trend-subnet"
    }
    + tags_all                    = {
        + "Name" = "trend-subnet"
    }
  + vpc_id          = (known after apply)

# aws_vpc.trend_vpc will be created
+ resource "aws_vpc" "trend_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_dngv6_addresses = true
    + enable_dns_support     = true
    + enable_network_address_usage_metrics = (known after apply)
    + id                  = (known after apply)
    + instance_tenancy      = "mult"
    + ipv6_association_id   = (known after apply)
    + ipv6_cidr_block        = (known after apply)
    + ipv6_cidr_block_network_border_group = (known after apply)
    + ipv6_route_table_id    = (known after apply)
    + owner_id              = (known after apply)
    + region                = "ap-south-1"
    + tags                  = {
        + "Name" = "trend-vpc"
    }
    + tags_all              = {
        + "Name" = "trend-vpc"
    }
  }

Plan: 8 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + ec2_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.
prakash@mac terraform %
```

3e) terraform apply auto-approve command

```
terraform --zsh -- 254x67
+ tag_all = {
  + "Name" = "trend-subnet"
}
+ vpc_id = (known after apply)

# aws_vpc.trend_vpc will be created
resource "aws_vpc" "trend_vpc" {
  + arn = (known after apply)
  + cidr_block = "19.0.0.0/16"
  + default_network_acl_id = (known after apply)
  + default_route_table_id = (known after apply)
  + default_security_group_id = (known after apply)
  + dns_support = (known after apply)
  + enable_dns_hostnames = true
  + enable_dns_support = true
  + enable_network_address_usage_metrics = (known after apply)
  + enable路由表 = (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" = "trend-vpc"
  }
  + tags_all = {
    + "Name" = "trend-vpc"
  }
}

Plant: 0 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ ec2_public_ip = (known after apply)
+ Jenkins_url = (known after apply)
aws_vpc.trend_vpc: Creating...
aws_vpc.trend_vpc: Creation complete after 12s [id=vpc-0e7648a983801d1f2a]
aws_internet_gateway.trend_igw: Creating...
aws_internet_gateway.trend_igw: Creation complete after 1s [id=igw-9726b751bacd4d3be]
aws_subnet.trend_subnet: Creating...
aws_subnet.trend_subnet: Creation complete after 1s [id=subnet-094ec1f5d5ef7b56c]
aws_subnet.trend_subnet: Creating...
aws_security_group.trend_sg: Creating...
aws_internet_gateway.trend_igw: Creation complete after 1s [id=igw-9726b751bacd4d3be]
aws_subnet.trend_subnet: Creation complete after 1s [id=subnet-094ec1f5d5ef7b56c]
aws_route.default_route: Creating...
aws_route.default_route: Creation complete after 1s [id=rtb-0a9ec1f565fcbb5dc1000289494]
aws_security_group.trend_sg: Creation complete after 4s [id=sg-0bf735ccce1a09d0eb]
aws_subnet.trend_subnet: Still creating... [last check: 1m ago]
aws_subnet.trend_subnet: Creation complete after 12s [id=subnet-07201a24be74a4a36]
aws_route_table_association.a: Creating...
aws_instance.jenkins_server: Creating...
aws_instance.jenkins_server: Creation complete after 9s [id=rta-basssoc-0f69a1cb52cd21f2e]
aws_instance.jenkins_server: Still creating... [0m16ms elapsed]
aws_instance.jenkins_server: Still creating... [0m26ms elapsed]
aws_instance.jenkins_server: Still creating... [0m36ms elapsed]
aws_instance.jenkins_server: Creation complete after 33s [id=i-0a2e01d28e208619c6]

Apply complete! Resources: 8 added, 0 changed, 0 destroyed.

Outputs:
ec2_public_ip = "13.233.71.23"
jenkins_url = "http://13.233.71.23:8080"
prakash@mac terraform %
```

3f) EC2 created successfully via Terraform

The screenshot shows the AWS CloudWatch Metrics Insights interface. A query is being run against the 'AWS Lambda Metrics' dataset over the last hour. The results are displayed in a table with columns for 'Time' (UTC), 'Function Name', 'Metric Name', 'Value', and 'Unit'. The table shows data for various AWS Lambda functions, including 'lambda@v2', 'lambda-trend', 'lambda-trend-1', 'lambda-trend-2', 'lambda-trend-3', 'lambda-trend-4', 'lambda-trend-5', 'lambda-trend-6', 'lambda-trend-7', and 'lambda-trend-8', with values ranging from 0 to 1000.

Time (UTC)	Function Name	Metric Name	Value	Unit
2023-09-20T10:00:00Z	lambda@v2	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend-1	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend-2	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend-3	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend-4	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend-5	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend-6	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend-7	ApproximateInvokeCount	1000	Count
2023-09-20T10:00:00Z	lambda-trend-8	ApproximateInvokeCount	1000	Count

3g) EC2 created and logged via ssh key

```
[prakash@mac ~]$ ssh -i "jenkins-server.pem" ubuntu@ec2-3-6-40-243.ap-south-1.compute.amazonaws.com
[prakash@mac ~]$ .ssh --ubuntu@ip-10-0-1-96:-- ssh -i jenkins-server.pem ubuntu@ec2-3-6-40-243.ap-south-1.compute.amazonaws.com -- 254x67
The authenticity of host 'ec2-3-6-40-243.ap-south-1.compute.amazonaws.com (3.6.40.243)' can't be established.
ED25519 key fingerprint is SHA256:y8skJjr0RxDZ0prbrz9liucAPXwRh0vVkd3zNwoD2M.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-6-40-243.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1035-aws x86_64)

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

System information as of Mon Aug 25 20:08:35 UTC 2025

 System load: 0.75          Processes:           108
 Usage of /:  42.8% of 7.57GB   Users logged in:     0
 Memory usage: 49%           IPv4 address for eth0: 10.0.1.96
 Swap usage:  0%

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

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

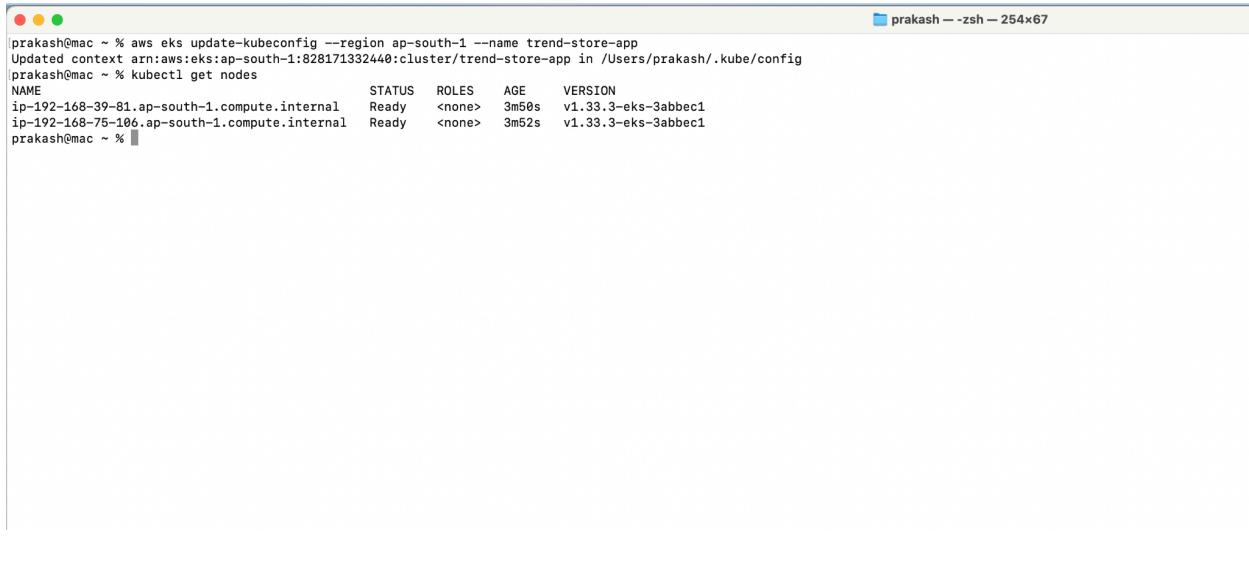
Last login: Mon Aug 25 20:07:07 2025 from 13.233.177.3
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-1-96:~$
```

4. EKS setup and Config

4a) New cluster trend-store-app created

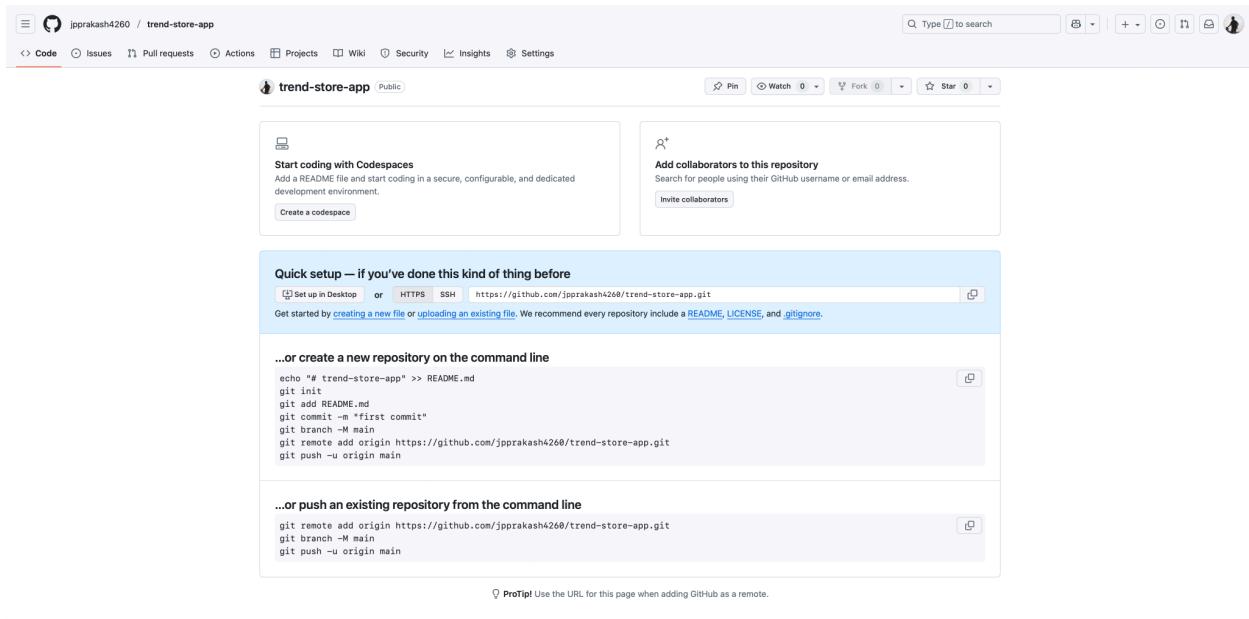
4b) eks setup verification and get pods commands



```
prakash@mac ~ % aws eks update-kubeconfig --region ap-south-1 --name trend-store-app
Updated context arn:aws:eks:ap-south-1:828171332440:cluster/trend-store-app in /Users/prakash/.kube/config
prakash@mac ~ % kubectl get nodes
NAME           STATUS  ROLES   AGE    VERSION
ip-192-168-39-81.ap-south-1.compute.internal  Ready   <none>  3m50s  v1.33.3-eks-3abbec1
ip-192-168-75-106.ap-south-1.compute.internal  Ready   <none>  3m52s  v1.33.3-eks-3abbec1
prakash@mac ~ %
```

5. GitHub setup

5a) New GitHub Repo



The screenshot shows the GitHub repository creation interface for a new repository named "trend-store-app". The repository is set to be public. The page provides instructions for quick setup via desktop, HTTPS, or SSH, and includes sections for creating a codespace and adding collaborators. It also shows command-line examples for initializing a new repository or pushing an existing one.

Quick setup — if you've done this kind of thing before

...or create a new repository on the command line

```
echo "# trend-store-app" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/jpprakash4260/trend-store-app.git
git push -u origin main
```

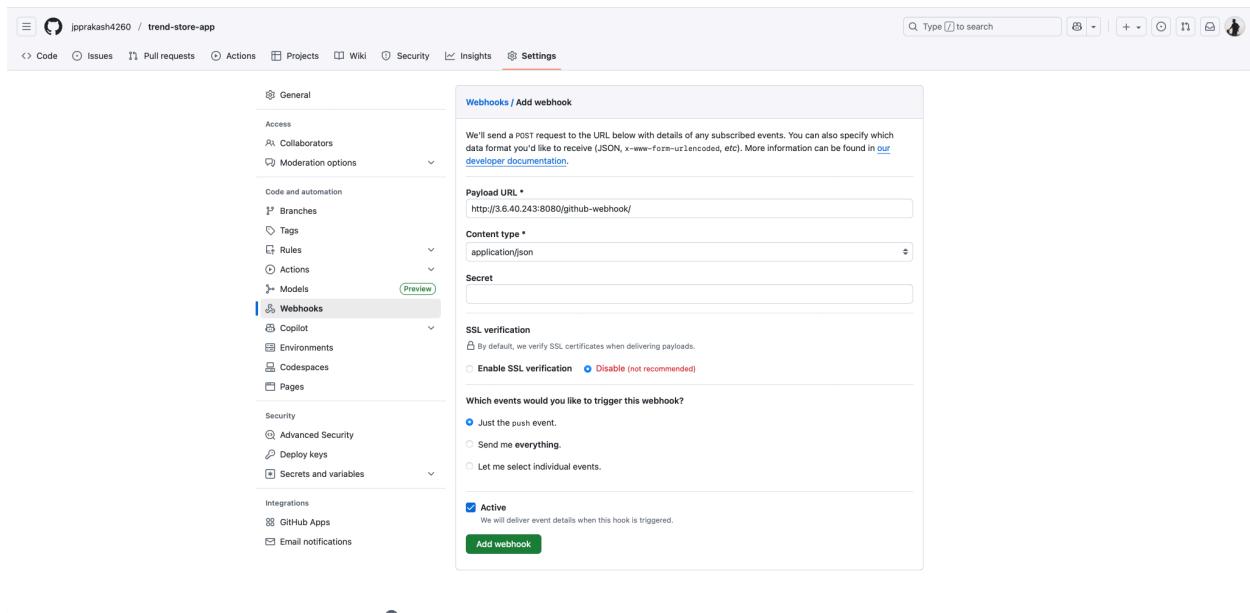
...or push an existing repository from the command line

```
git remote add origin https://github.com/jpprakash4260/trend-store-app.git
git branch -M main
git push -u origin main
```

5b) Git commit and pushing commands

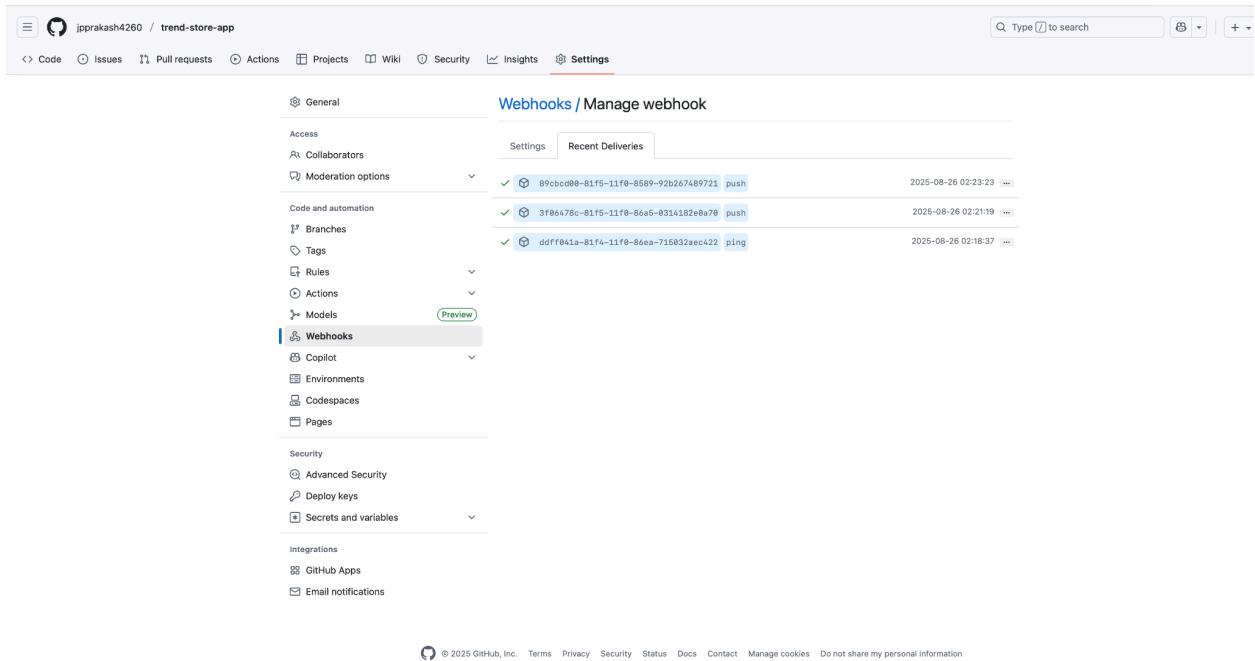
```
prakash@mac ~ % cd trend_store_app
prakash@mac trend_store_app % git init
Initialized empty Git repository in /Users/prakash/trend_store_app/.git/
prakash@mac trend_store_app % git remote add origin git@github.com:jpprakash4260/trend-store-app.git
prakash@mac trend_store_app % git add .
prakash@mac trend_store_app % git commit -am '1st commit'
[main (root-commit) 0b3175e] 1st commit
 9 files changed, 277 insertions(+)
create mode 100644 .gitignore
create mode 100644 Dockerfile
create mode 100644 k8s/deployment.yml
create mode 100644 k8s/service.yaml
create mode 100644 nginx.conf
create mode 100644 terraform/main.tf
create mode 100644 terraform/outputs.tf
create mode 100644 terraform/provider.tf
create mode 100644 terraform/variables.tf
prakash@mac trend_store_app % git branch -M main
prakash@mac trend_store_app % git push -u origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 2.93 KiB | 2.93 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:jpprakash4260/trend-store-app.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
prakash@mac trend_store_app %
```

5c) Webhook Configuration to trigger Jenkins job



The screenshot shows the GitHub settings interface for a repository named 'trend-store-app'. The left sidebar has a 'Webhooks' section selected under 'General' settings. The main right panel is titled 'Webhooks / Add webhook'. It contains fields for 'Payload URL' (set to 'http://3.6.40.243:8080/github-webhook/'), 'Content type' (set to 'application/json'), and a 'Secret' field. Below these, there's a 'SSL verification' section with a note about verifying SSL certificates and a radio button for 'Disable (not recommended)'. Under 'Which events would you like to trigger this webhook?', there are three options: 'Just the push event.' (selected), 'Send me everything.', and 'Let me select individual events.'. At the bottom, a checkbox for 'Active' is checked, and a 'Add webhook' button is visible.

5d) delivery verification



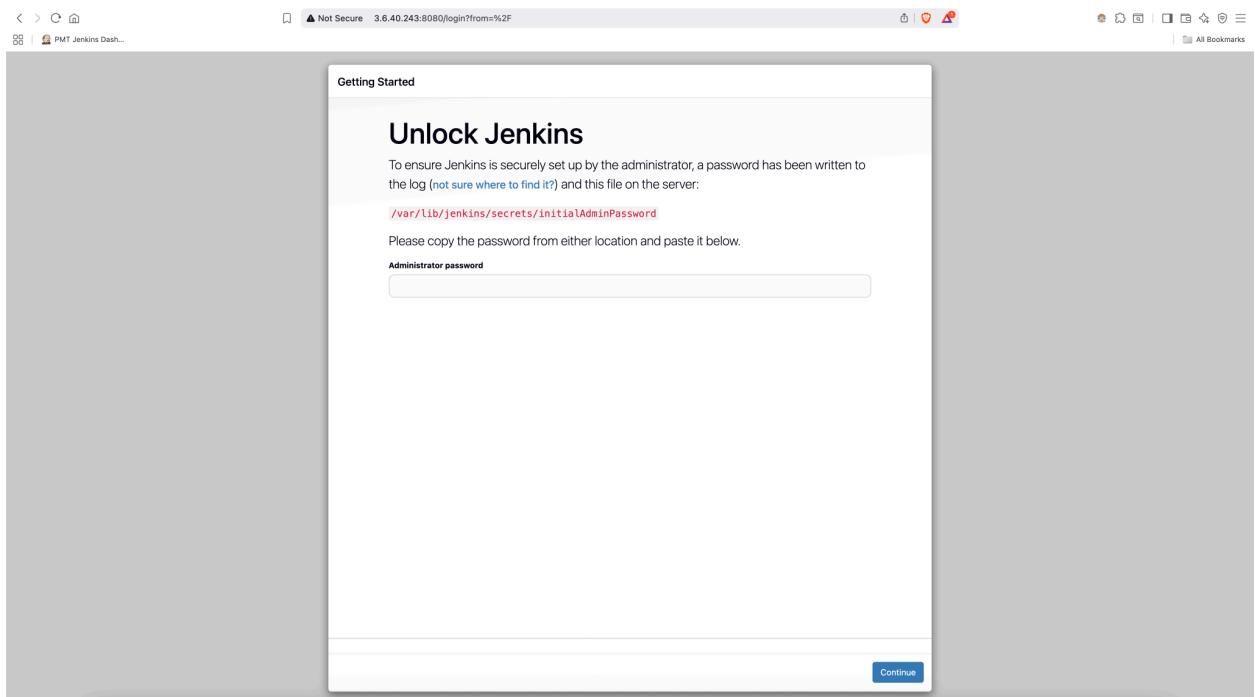
The screenshot shows the GitHub Settings page for the repository 'trend-store-app'. The 'Webhooks' tab is selected under the 'Webhooks' section of the sidebar. The 'Recent Deliveries' tab is active, displaying three recent webhook deliveries:

User	Event	Timestamp
89cbcdd0-81f5-11f0-8589-92b267489721	push	2025-08-26 02:23:23
3f86478c-81f5-11f0-86a5-0314182e0a70	push	2025-08-26 02:21:19
ddff041a-81f4-11f0-86ea-715832aec422	ping	2025-08-26 02:18:37

The sidebar also 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, and Email notifications.

6. Jenkins Setup and Config

6a) jenkins works on default 8080 port



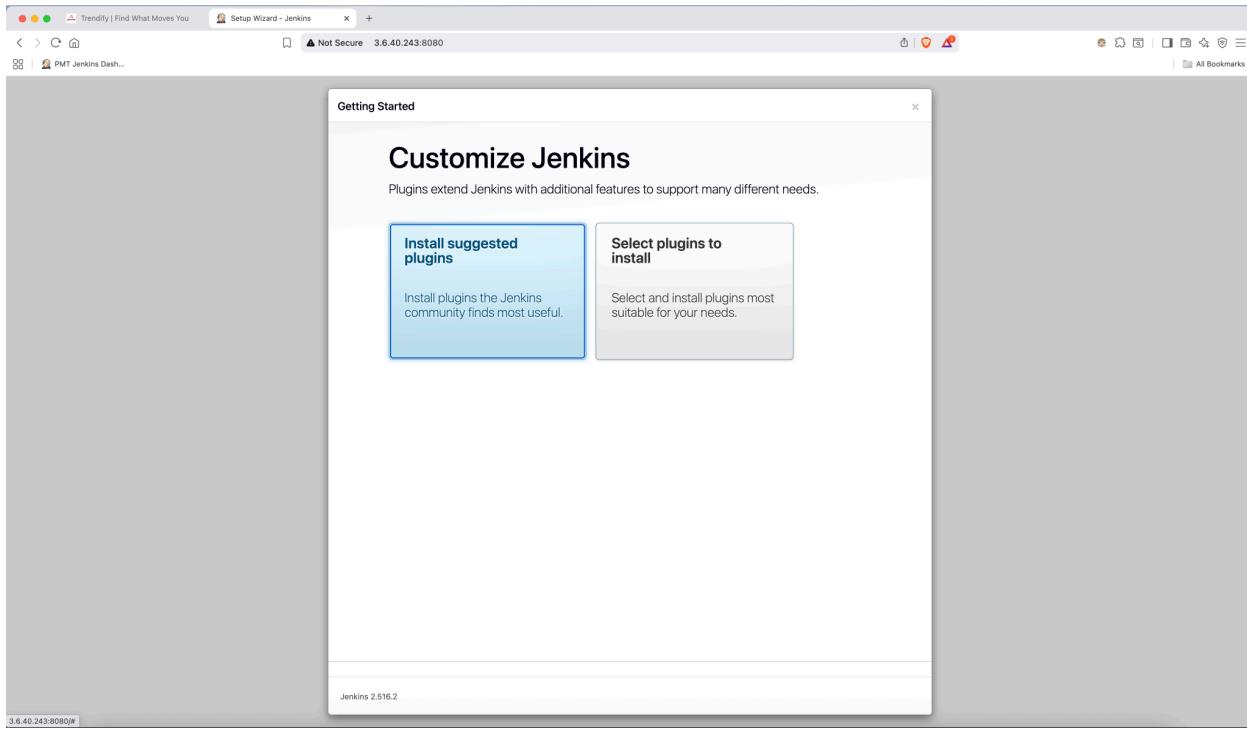
The screenshot shows a browser window displaying the Jenkins 'Unlock Jenkins' setup step. The URL is `3.6.40.243:8080/login?from=%2F`. The page title is 'Getting Started' and the main heading is 'Unlock Jenkins'. It instructs the user to ensure Jenkins is securely set up by the administrator, with a password written to the log file `/var/lib/jenkins/secrets/initialAdminPassword`. A text input field is provided for pasting the password, labeled 'Administrator password'. A 'Continue' button is at the bottom right.

6b) get initial password



```
.ssh — ubuntu@ip-10-0-1-96: ~ — ssh -i jenkins-se
[ubuntu@ip-10-0-1-96:~$ cat /var/lib/jenkins/secrets/initialAdminPassword
cat: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
[ubuntu@ip-10-0-1-96:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
01a3d9b095a84ec4af235dc8614b821a
[ubuntu@ip-10-0-1-96:~$
```

6c) Jenkins Initial Plugin setup



6d) Create New Admin User

The screenshot shows the Jenkins Setup Wizard - Create First Admin User page. It has fields for Username (Admin), Password (.....), Confirm password (.....), Full name (Jayaprakash), and E-mail address (jayaprakash4260@gmail.com). At the bottom, there are buttons for Jenkins 2.516.2, Skip and continue as admin, and Save and Continue.

6e) Installing Essential Plugins like Git, Docker, kubernetes, stage view

The screenshot shows the Jenkins Manage Jenkins - Plugins page. The sidebar includes options for Plugins, Updates, Available plugins, Installed plugins, Advanced settings, and Download progress. The main area displays a list of installed and available plugins with their status (e.g., success, pending). A note at the bottom says "you can start using the installed plugins right away".

Plugin	Status
EDDSA API	Success
Trilead API	Success
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
Cloud Statistics	Success
Authentication Tokens API	Success
Docker Commons	Success
Apache HttpComponents Client 5.x API	Success
Commons Compress API	Success
Docker API	Success
Docker	Success
Docker Pipeline	Success
SSH server	Success
Git server	Success
Kubernetes Client API	Success
Kubernetes Credentials	Success
Kubernetes	Pending
Kubernetes CLI	Pending
Loading plugin extensions	Pending
Restarting Jenkins	Pending

6f) Adding Docker Credentials to Jenkins Credentials Globally

The screenshot shows the Jenkins 'New credentials' configuration page. The 'Kind' dropdown is set to 'Username with password'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. The 'Username' field contains 'jprakash2306'. The 'Password' field contains a masked password. The 'ID' field is 'jprakash2306-DockerHub-credentials'. The 'Description' field is 'DockerHub credential'. A blue border highlights the 'Description' field. At the bottom is a 'Create' button.

6g) Creating New Pipeline Job

The screenshot shows the Jenkins 'New Item' configuration page. The 'Enter an item name' field contains 'Trend-Store-App'. The 'Select an item type' section shows several options: 'Freestyle project' (classic job type), 'Pipeline' (selected and highlighted with a black border), 'Multi-configuration project' (for testing multiple configurations), 'Folder' (for grouping items), 'Multibranch Pipeline' (for creating branches), and 'Organization Folder' (for scanning repositories). The 'OK' button is at the bottom.

6h) Adding Jenkinsfile on project root folder

```
prakash@mac trend-store-app % git add .
prakash@mac trend-store-app % git commit -am 'adding Jenkinsfile'
[main dc84d6] adding Jenkinsfile
 1 file changed, 37 insertions(+)
的伟大 mode
prakash@mac trend-store-app % git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 696 bytes | 696.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1)
To github.com:jprakash4260/trend-store-app.git
 8bfec3..dc84d6 main --> main
prakash@mac trend-store-app % [ ]
```

6i) Enabling Github Hook Trigger for Gitscm polling

The screenshot shows the Jenkins Pipeline configuration page for the 'Trend-Store-App' job. The 'General' section is active, displaying the job name 'Trend-Store-App Pipeline Job'. The 'Triggers' section is expanded, showing several options: 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project' (which is checked), 'Pipeline speed/durability override', 'Preserve stashes from completed builds', 'This project is parameterised', and 'Throttle builds'. The 'Triggers' section also includes a note about setting up automated actions based on events like code changes or scheduled times, with options for 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling' (which is checked), 'Poll SCM', and 'Trigger builds remotely (e.g., from scripts)'. At the bottom, there are 'Save' and 'Apply' buttons.

6j) get Jenkinsfile from project repo

The screenshot shows the Jenkins Pipeline configuration page for a job named 'Trend-Store-App'. The 'Pipeline' tab is selected under the 'Configure' section. The 'SCM' section is set to 'Git' and has a 'Repository URL' of 'https://github.com/jprakash4260/trend-store-app.git'. The 'Branches to build' section specifies 'Branch Specifier (blank for 'any')' as '*/main'. The 'Script Path' is set to 'Jenkinsfile'. The 'Lightweight checkout' checkbox is checked. At the bottom, there are 'Save' and 'Apply' buttons.

6k) installing kubectl for apply and rollout from Jenkins Pipeline

```
ubuntu@ip-10-0-1-96:~$ curl -LO "https://dl.k8s.io/release/v1.30.0/bin/linux/amd64/kubectl"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload   Total Spent  Left  Speed
100  138  100  138     0      0  404      0 --:--:--:--:--:-- 403
100 49.0M  100 49.0M     0      0 47.3M      0 0:00:01 0:00:01 --:--:-- 123M
ubuntu@ip-10-0-1-96:~$ chmod +x kubectl
ubuntu@ip-10-0-1-96:~$ sudo mv kubectl /usr/local/bin/
ubuntu@ip-10-0-1-96:~$ kubectl version --client
Client Version: v1.30.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
ubuntu@ip-10-0-1-96:~$
```

6l) AWS configure on EC2

```
ubuntu@ip-10-0-1-96:~$ sudo apt-get install -y awscli
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
awscli is already the newest version (1.22.34-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-10-0-1-96:~$ aws --version
aws-cli/2.28.17 Python/3.13.7 Linux/6.8.0-1035-aws exe/x86_64.ubuntu.22
ubuntu@ip-10-0-1-96:~$ aws eks update-kubeconfig --region ap-south-1 --name trend-store-app
Updated context arn:aws:eks:ap-south-1:1828171332440:cluster/trend-store-app in /home/ubuntu/.kube/config
ubuntu@ip-10-0-1-96:~$ kubectl get nodes
NAME                      STATUS   ROLES      AGE   VERSION
ip-192-168-39-81.ap-south-1.compute.internal   Ready    <none>   84m   v1.33.3-eks-3abbec1
ip-192-168-75-106.ap-south-1.compute.internal   Ready    <none>   84m   v1.33.3-eks-3abbec1
ubuntu@ip-10-0-1-96:~$
```

6m) Jenkins Pipeline Job Triggered Automatically

The screenshot shows the Jenkins Pipeline Job 'Trend-Store-App' interface. On the left, there's a sidebar with various Jenkins management options like Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, Pipeline Syntax, and GitHub Hook Log. Below this is a list of recent builds: #7 21:34, #6 21:32, #5 21:27, #4 20:56, #3 20:53, #2 20:51, and #1 20:44.

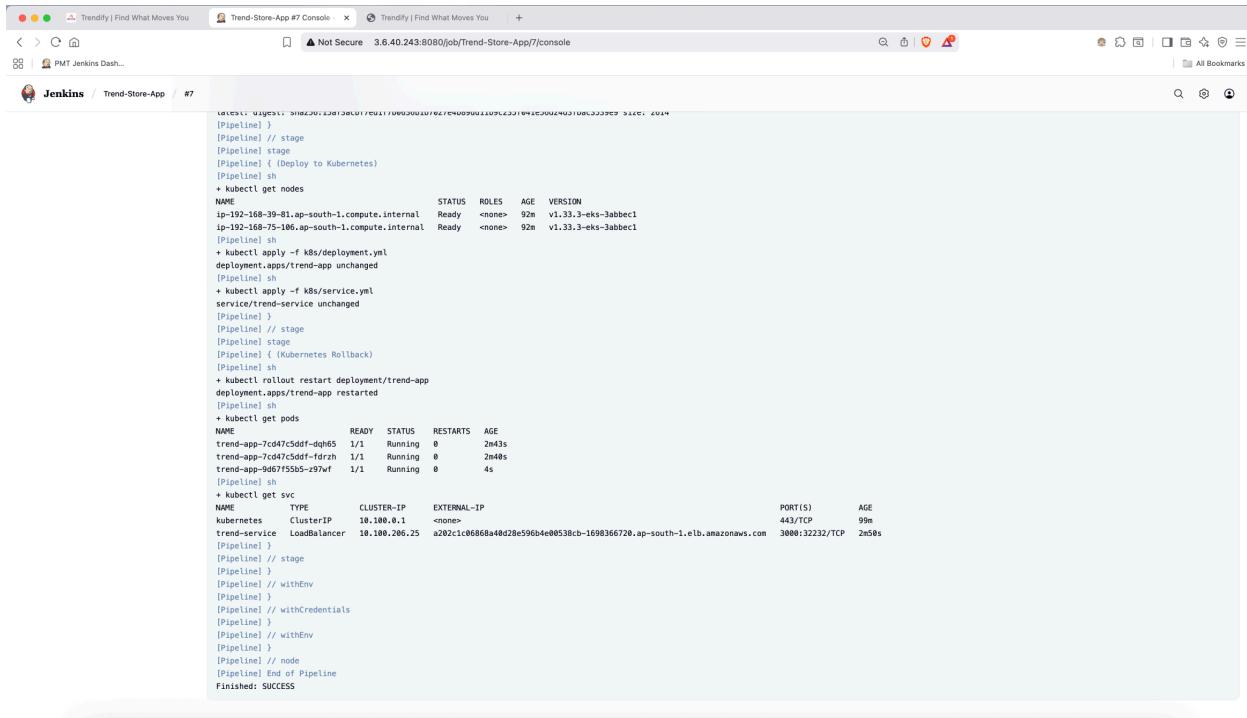
The main area is titled 'Stage View' and displays a grid of pipeline stages. The columns are labeled: Declarative: Checkout SCM, Checkout, Build Docker Image, Push to DockerHub, Deploy to Kubernetes, and Kubernetes Rollback. Each stage row contains a timestamp, commit details, and duration for each step. Some steps are marked as failed (e.g., #5 build). The last successful build (#7) is shown at the bottom.

	Declarative: Checkout SCM	Checkout	Build Docker Image	Push to DockerHub	Deploy to Kubernetes	Kubernetes Rollback	
#7 Aug 26 03:04	2 commits	1s	982ms	3s	9s	11s	9s
#6 Aug 26 02:57	No changes	795ms	778ms	4s	9s	8s	7s
#5 Aug 26 02:57	No changes	714ms	717ms	4s	9s	908ms failed	85ms failed
#4 Aug 26 02:25	2 commits	2s	996ms	10s	21s	417ms failed	97ms failed
#3 Aug 26 02:23	1 commit	1s	921ms	4s failed	136ms failed	128ms failed	119ms failed
#2 Aug 26 02:21	1 commit	1s	1s	2s failed	295ms failed	146ms failed	140ms failed
#1 Aug 26 02:14	No changes	1s	309ms failed	166ms failed	157ms failed	140ms failed	

Permalinks

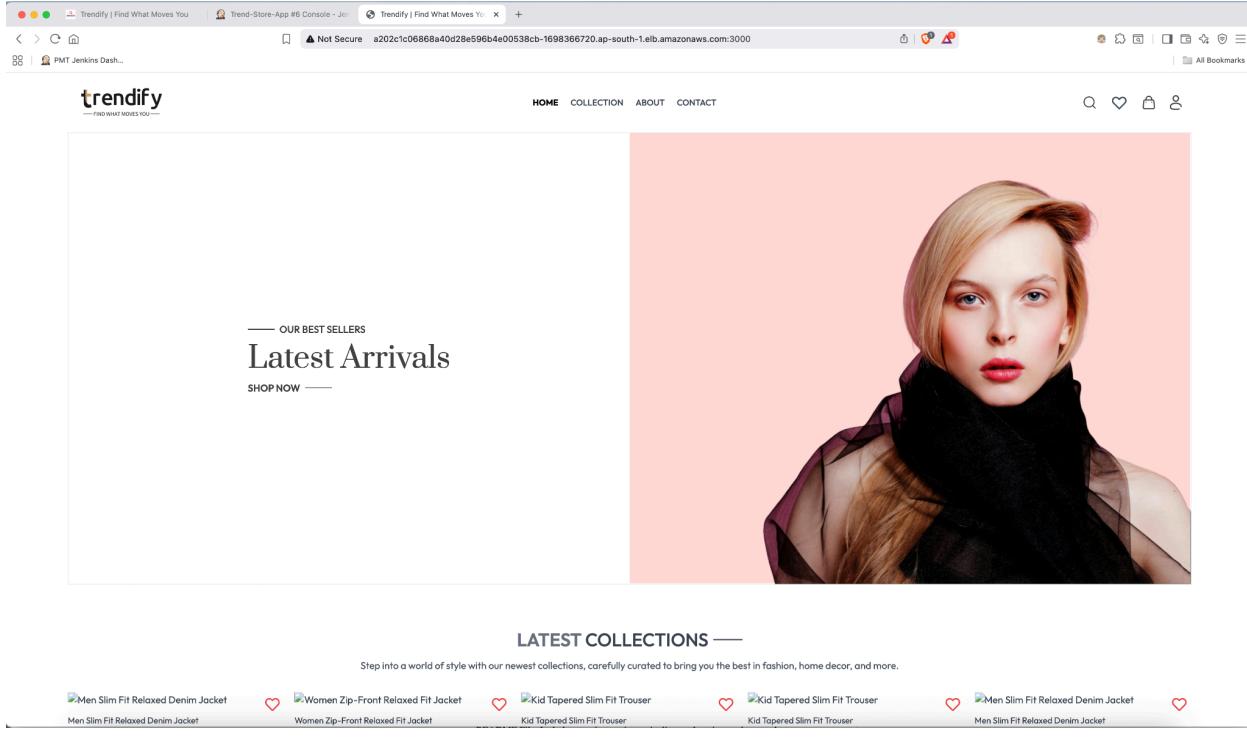
- Last build (#7), 7 hr 50 min ago
- Last stable build (#7), 7 hr 50 min ago
- Last successful build (#7), 7 hr 50 min ago
- Last failed build (#6), 7 hr 58 min ago
- Last unsuccessful build (#5), 7 hr 58 min ago
- Last completed build (#7), 7 hr 50 min ago

6n) Jenkins Pipeline Deployed App successfully



```
[Pipeline] 
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to Kubernetes)
[Pipeline] sh
+ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-192-168-39-81.ap-south-1.compute.internal Ready <none> 92m v1.33.3-eks-3abbe1
ip-192-168-75-186.ap-south-1.compute.internal Ready <none> 92m v1.33.3-eks-3abbe1
[Pipeline]
+ kubectl apply -f k8s/deployment.yaml
deployment.apps/trend-app unchanged
[Pipeline] sh
+ kubectl apply -f k8s/service.yaml
service/trend-service unchanged
[Pipeline]
[Pipeline] stage
[Pipeline] { (Kubernetes Rollback)
[Pipeline]
+ kubectl rollout restart deployment/trend-app
deployment.apps/trend-app restarted
[Pipeline] sh
+ kubectl get pods
NAME READY STATUS RESTARTS AGE
trend-app-7cd47c5ddef-dqhb65 1/1 Running 0 2m43s
trend-app-7cd47c5ddef-fdrzh 1/1 Running 0 2m48s
trend-app-9d67f75b5-z97wf 1/1 Running 0 4s
[Pipeline] sh
+ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.100.0.1 <none> 443/TCP 99m
trend-service LoadBalancer 10.100.206.25 a202c1c06868a40d28e596b4e00538cb-1698366720.ap-south-1.elb.amazonaws.com 3000:32322/TCP 2650s
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

6o) App Running via EKS-cluster on 3000 port



The screenshot shows a web browser displaying the Trendify website. The URL in the address bar is `a202c1c06868a40d28e596b4e00538cb-1698366720.ap-south-1.elb.amazonaws.com:3000`. The page content includes:

- The Trendify logo and tagline "Find What Moves You".
- A navigation bar with links for HOME, COLLECTION, ABOUT, and CONTACT.
- A large image of a woman with blonde hair wearing a black turtleneck.
- A section titled "OUR BEST SELLERS" with a "Latest Arrivals" heading and a "SHOP NOW" button.
- A "LATEST COLLECTIONS" section featuring five product cards: "Men Slim Fit Relaxed Denim Jacket", "Women Zip-Front Relaxed Fit Jacket", "Kid Tapered Slim Fit Trouser", "Kid Tapered Slim Fit Trouser", and "Men Slim Fit Relaxed Denim Jacket". Each card includes a small image, a red heart icon, and a link to the product page.
- At the bottom, there is a decorative horizontal line with a dashed center.

7. Monitoring

7a) Installing Helm and Prometheus on new Namespace called monitoring

7b) Prometheus configurations

```
ubntulip-10-0-1-96-8$ kubectl get nodes
NAME          STATUS   ROLES    AGE     VERSION
ip-192-168-98-1   ready   <none>   7h32m   v1.33.3-eks-3abbec1
ip-192-168-75-196   ready   <none>   7h32m   v1.33.3-eks-3abbec1
ip-192-168-75-196-7   ready   <none>   7h32m   v1.33.3-eks-3abbec1

Error from server (AlreadyExists): namespaces "monitoring" already exists
ubuntu@ip-10-0-1-96:~$ cat > k8s-monitoring-values-monitoring.yaml <<'EOF'
# k8s-monitoring-stack overrides (soft, minimal) ---

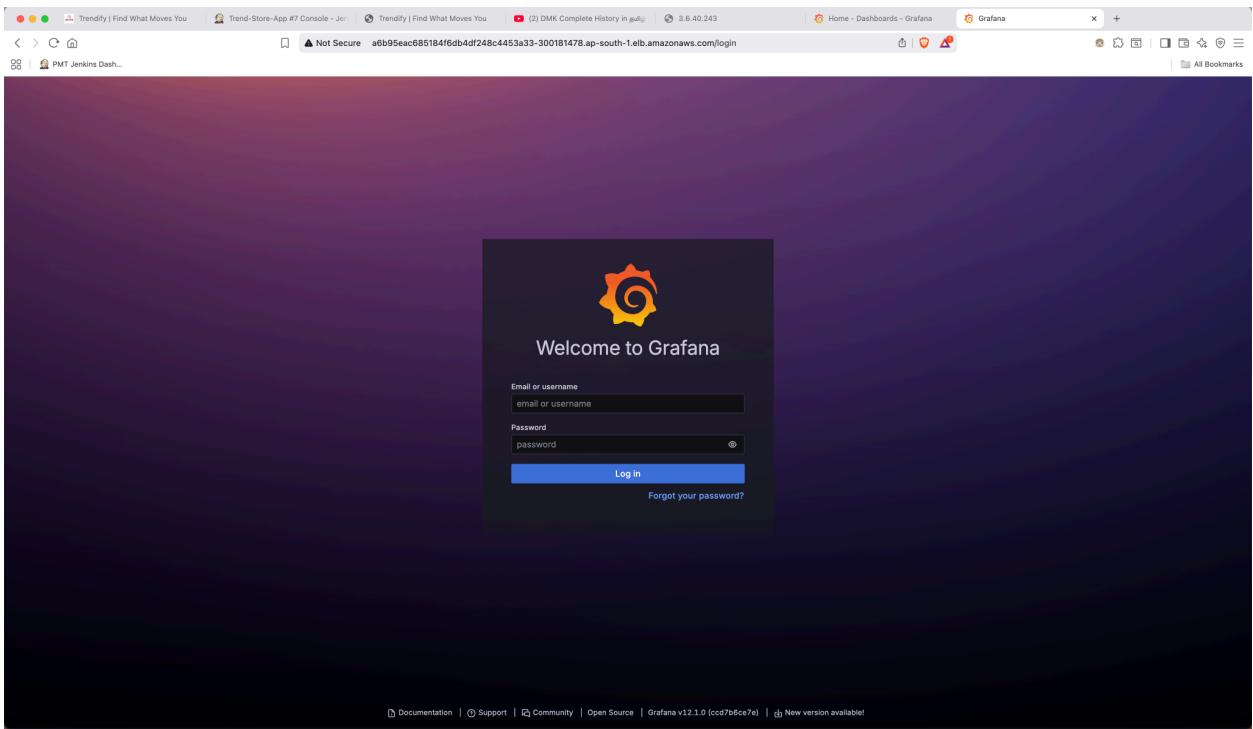
grafana:
  adminPassword: "Admin123"      # change for prod
  service:
    type: LoadBalancer
  defaultClusterEnabled: true

prometheus:
  service:
    type: ClusterIP           # generally keep internal
  persistentVolume:
    retention: 3d              # keep 15 days of metrics (adjust)
    scrapeInterval: 30s         # default scrape interval

alertmanager:
  config:
    route:
      receiver: "null"
    receivers:
      - name: "null"
  # kube-state-metrics and node-exporter are enabled by default
EOF
ubuntu@ip-10-0-1-96:~$ helm upgrade --install monitoring prometheus-community/kube-prometheus-stack \
--namespace monitoring \
--values ./k8s-monitoring-values-monitoring.yaml
Release "monitoring" does not exist. Installing it now.

NAME: monitoring
LAST DEPLOYED: Tue Aug 26 03:35:57 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 "telesemonoring"
```

7c) Grafana Dashboard for EKS-cluster



7d) trend-service.yml file configuration

```
GNU nano 6.2
apiVersion: v1
kind: Service
metadata:
  name: trend-service
  namespace: default
  labels:
    app: trend-app
spec:
  selector:
    app: trend-app
  ports:
  - name: http
    port: 80
    targetPort: 80
```

7e) alertmanager.yml file configuration

```
GNU nano 6.2
alertmanager:
  config:
    global:
      smtp_smarthost: 'smtp.gmail.com:587' # Your SMTP server
      smtp_from: 'jayaprakash42601@gmail.com'
      smtp_auth_username: 'jayaprakash42601@gmail.com'
      smtp_auth_password: 'qfjxbhxdrtqverm' # Use an App Password for Gmail
      smtp_require_tls: true

    route:
      receiver: 'email-receiver'
      group_wait: 30s
      group_interval: 5m
      repeat_interval: 3h

  receivers:
    - name: 'email-receiver'
      email_configs:
        - to: 'jayaprakash4260@gmail.com' # Primary Email Id
```

7f) Grafana Dashboard

The screenshot shows the Grafana interface with the sidebar menu open. The 'Dashboards' option is selected. The main area displays a list of available dashboards, each represented by a thumbnail icon and a title. To the right of the dashboard list, there is a 'Tags' section where specific tags are highlighted in colored boxes. The tags shown are 'alertmanager-mixin' (blue), 'coredns-dns' (orange), and 'etcd-mixin' (red). The rest of the tags listed are 'kubernetes-minikube' (purple).

Dashboard Title	Tags
Alertmanager / Overview	alertmanager-mixin
CoreDNS	coredns-dns
etcd	etcd-mixin
Grafana Overview	kubernetes-minikube
Kubernetes / API server	kubernetes-minikube
Kubernetes / Compute Resources / Multi-Cluster	kubernetes-minikube
Kubernetes / Compute Resources / Cluster	kubernetes-minikube
Kubernetes / Compute Resources / Namespace (Pods)	kubernetes-minikube
Kubernetes / Compute Resources / Namespace (Workloads)	kubernetes-minikube
Kubernetes / Compute Resources / Node (Pods)	kubernetes-minikube
Kubernetes / Compute Resources / Pod	kubernetes-minikube
Kubernetes / Compute Resources / Workload	kubernetes-minikube
Kubernetes / Controller Manager	kubernetes-minikube
Kubernetes / Kubelet	kubernetes-minikube
Kubernetes / Networking / Cluster	kubernetes-minikube
Kubernetes / Networking / Namespace (Pods)	kubernetes-minikube
Kubernetes / Networking / Namespace (Workload)	kubernetes-minikube
Kubernetes / Networking / Pod	kubernetes-minikube
Kubernetes / Networking / Workload	kubernetes-minikube
Kubernetes / Persistent Volumes	kubernetes-minikube
Kubernetes / Proxy	kubernetes-minikube

7g) Cluster's Dashboard View

