BANKING/FINANCE CAPSTONE PROJECT

Setting up Repositories
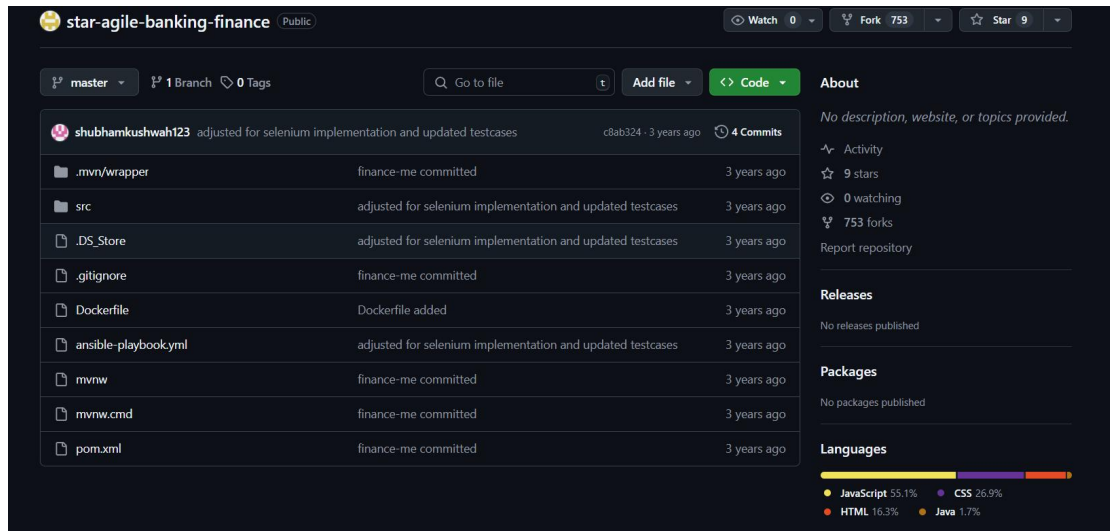
Forked the required files from Staragile github

Made a local repo

```
1   pipeline{
2       agent any
3       stages{
4           stage('checkout the code from github'){
5               steps{
6                   git url: 'https://github.com/athul-anilkumar/star-agile-banking-finance/'
7                   echo 'github url checkout'
8               }
9           }
10          stage('codecompile '){
11              steps{
12                  echo 'starting compiling'
13                  sh 'mvn compile'
14              }
15          }
16          stage('codetesting'){
17              steps{
18                  sh 'mvn test'
19              }
20          }
21          stage('qa'){
22              steps{
23                  sh 'mvn checkstyle:checkstyle'
24              }
25          }
26          stage('package'){
27              steps{
28                  sh 'mvn package'
29              }
30          }
31          stage('run dockerfile'){
32              steps{
33                  sh 'docker build -t myimg .'
34              }
35          }
36          stage('port expose'){
37              steps{
38                  sh 'docker run -dt -p 8091:8091 --name c000 myimg'
39              }
40          }
41      }
42  }
```

Added Jenkins file to Repo

Updated remote repository by using git push

Setting Up Terraform


```
terraform > credentials.tf > provider "aws"
  1   provider "aws"{
  2     region = "ap-south-1"
  3     access_key = "AKI        "
  4     secret_key = "PptDe      "
  5   }
```

Created a credentials.tf file.region used is Mumbai.


```
ec2.tf > resource "aws_instance" "res1" > tags
  1   resource "aws_instance" "res1" {
  2
  3
  4     ami           = "ami-00bb6a80f01f03502"
  5     instance_type = "t3.micro"
  6
  7     tags = {
  8       Name = "grafana"
  9     }
 10   }
 11   resource "aws_instance" "res1" {
 12     ami           = "ami-00bb6a80f01f03502"
 13     instance_type = "t3.medium"
 14
 15     tags = {
 16       Name = "jenkins"
 17     }
 18   }
```

Created a ec2.tf file to create 1 t2.micro instance and 1 t2.medium in Mumbai.


```
C:\Users\Athul\Desktop\Banking Capstone project\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 v5.84.0

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
```

Initiated terraform using terraform init in CMD terminal.


```
C:\Users\Athul\Desktop\Banking Capstone project\terraform>terraform validate
Success! The configuration is valid.


C:\Users\Athul\Desktop\Banking Capstone project\terraform>terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.res1 will be created
  + resource "aws_instance" "res1" {
      + ami                                  = "ami-00bb6a80f01f03502"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (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)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t3.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
```

Used "terraform apply " to start the instances.

Connected to The jenkins machine

Installed jenkins using shell script.

Installed docker in Jenkins machine

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	↻ Credentials Binding
↻ Timestamper	↻ Workspace Cleanup	↻ Ant	↻ Gradle
↻ Pipeline	↻ GitHub Branch Source	↻ Pipeline: GitHub Groovy Libraries	↻ Pipeline Graph View
↻ Git	↻ SSH Build Agents	↻ Matrix Authorization Strategy	○ PAM Authentication
○ LDAP	↻ Email Extension	○ Mailer	○ Dark Theme

```
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** Token Macro
Build Timeout
```

```
** - required dependency
```

Connected to jenkins machine using publicip:8080 logged in and installed default plugins

Jenkins   Search (CTRL+K)   🛡 ❶ 👤 admin ⌄   ↪ log out

Dashboard > Manage Jenkins > Plugins

Plugins

🔍 docker

⬇ Install ⌄ ↻

	Install	Name ↓	Released
	☑	**Docker** 1.7.0 Cloud Providers Cluster Management docker This plugin integrates Jenkins with Docker	3 mo 4 days ago
	☐	**Docker Commons** 445.v6b_646c962a_94 Library plugins (for use by other plugins) docker Provides the common shared functionality for various Docker-related plugins.	2 mo 10 days ago
	☐	**Docker Pipeline** 580.vc0c340686b_54 pipeline DevOps Deployment docker Build and use Docker containers from pipelines.	8 mo 0 days ago
	☐	**Docker API** 3.4.1-96.v77147a_de67f8 Library plugins (for use by other plugins) docker This plugin provides docker-java API for other plugins.	1 mo 4 days ago

Plugins sidebar:
- ⬆ Updates
- 🛍 Available plugins
- ⟳ Installed plugins
- ⚙ Advanced settings
- ≡ Download progress

Downloaded docker plugin for jenkins

Created new "bank finance" pipeline.

Gave Jenkins user docker privilages

Global credentials (unrestricted)

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

	ID	Name	Kind	Description	
	43b1fd6f-7d74-47b8-8c76-cfade65f52de	Theertha23/******	Username with password		🔧

Icon: S   M   L

Build after other projects are built ?
Build periodically ?
GitHub hook trigger for GITScm polling ?
☑ Poll SCM ?
Schedule ?

```
H * * * *
```

Would last have run at Friday, January 17, 2025 at 4:06:07 PM Coordinated Universal Time; would next run at Friday, January 17, 2025 at 5:06:07 PM Coordinated Universal Time.

Ignore post-commit hooks ?

Trigger builds remotely (e.g., from scripts) ?

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/athul-anilkumar/star-agile-banking-finance

Credentials ?

- none - ▾

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ? ✕

*/master

Save   Apply

Pipeline configured, added git SCM hook as well as a timed build every 1 hour
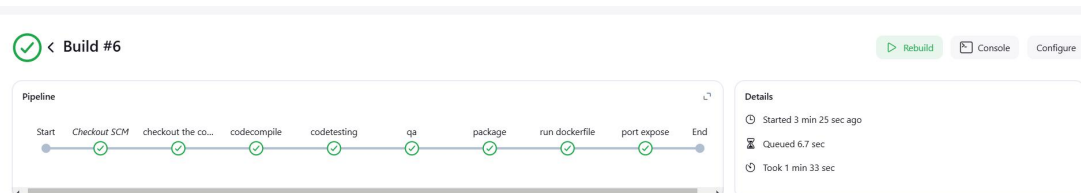
Build completed successfully

Website is running successfully

Setup of monitoring

Installed grafana on a separate server

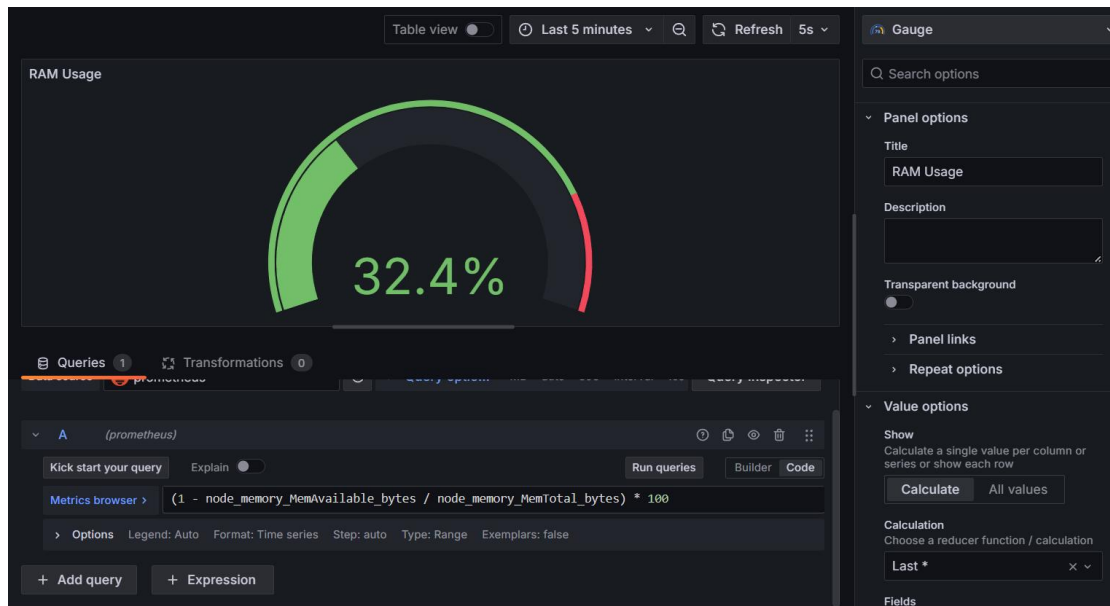Installed Prometheus and node_exporter on the jenkins server using shell script and edited prometheus.yml file
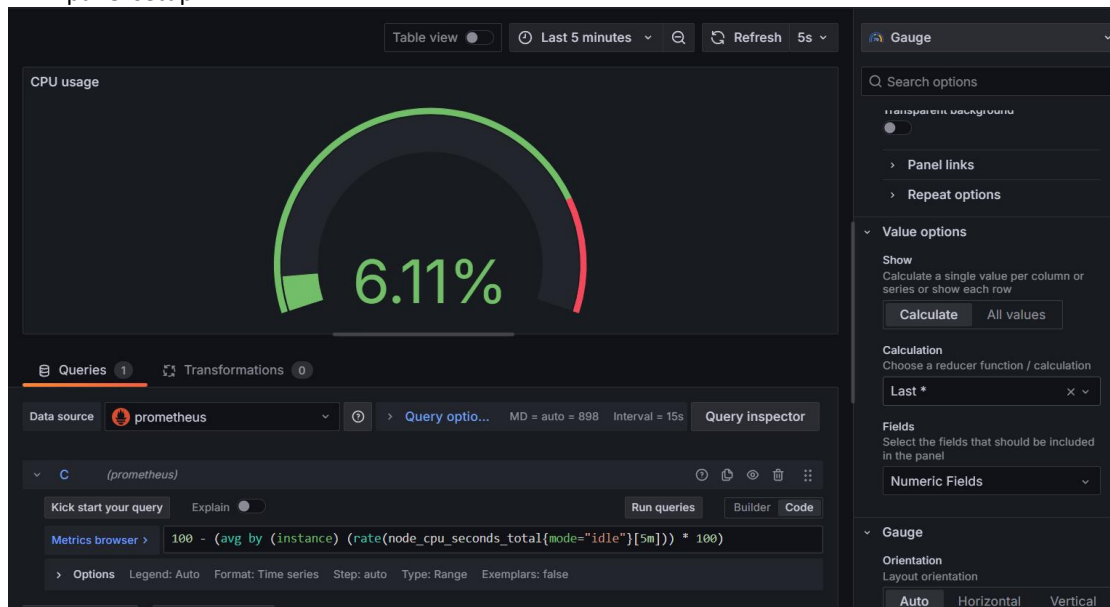
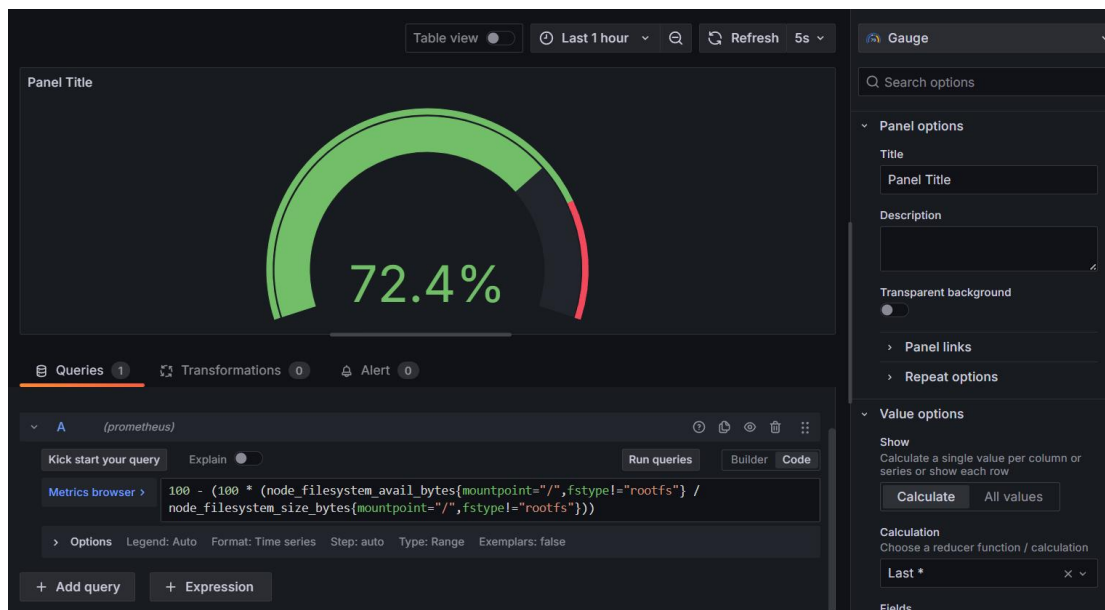Started prometheus using "./prometheus"
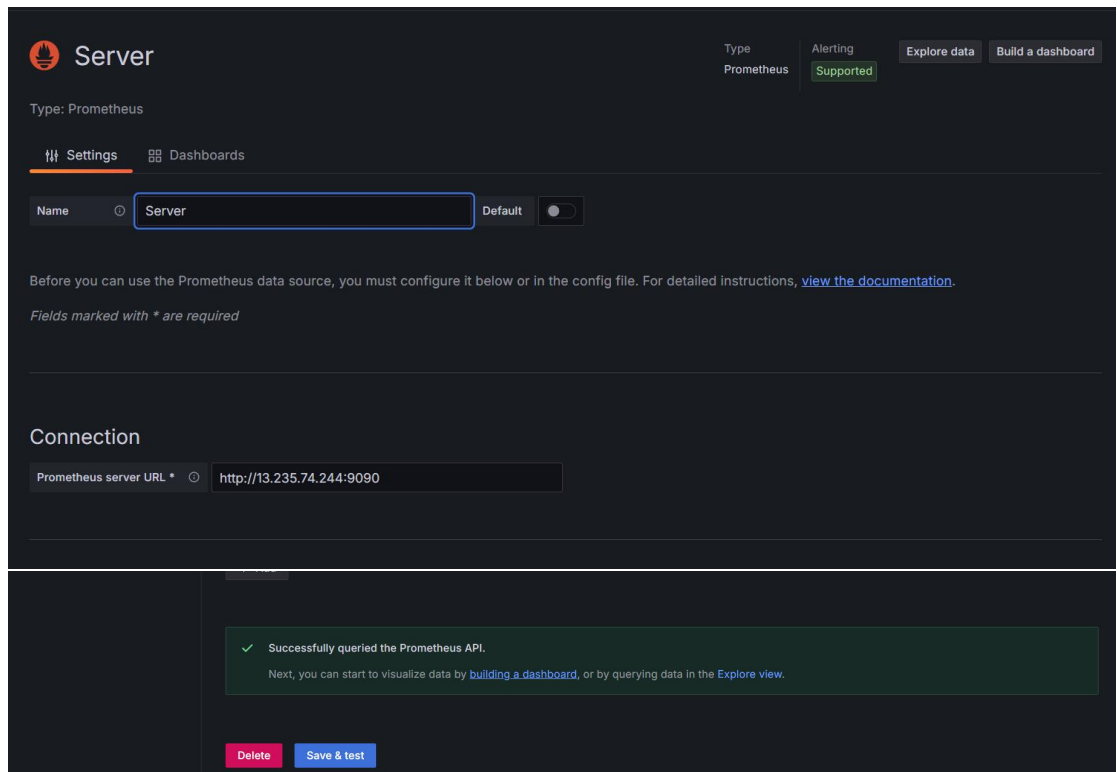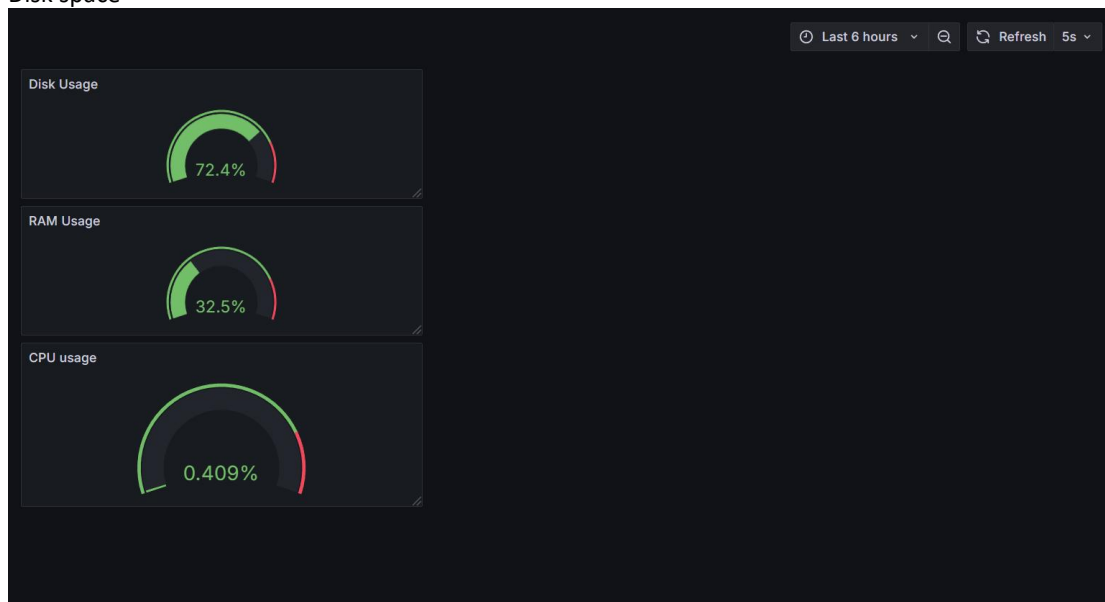

Setting up Grafana using prometheus datapoint

RAM panel setup

CPU panel setup

Disk space

Final dashboard setup