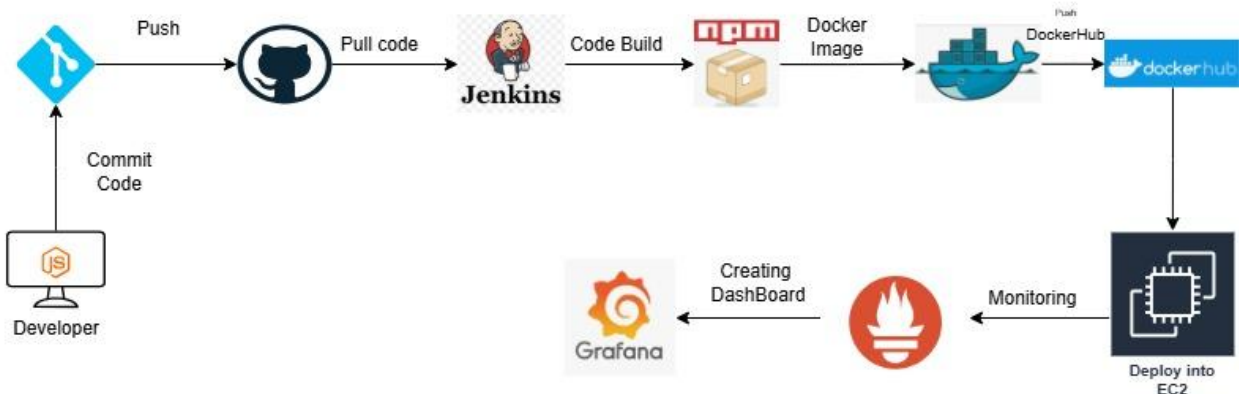


NODE.JS DEPLOYMENT

PROJECT FLOW:



1. Set Up a Node.js Application

Create the Node.js application:

In your project directory, initialize a new Node.js project:

- 1) `npm init -y`
- 2) Installing require Dependencies `npm install express`

Project Structure:

```
EXPLORER  JS server.js  package.json  package-lock.json  Dockerfile
NODEDEPLOYMENT
  Dockerfile
  package-lock.json
  package.json
  README.md
  JS server.js

JS server.js
1  const express = require('express');
2  const app = express();
3  const port = 5000;
4
5  app.get('/', (req, res) => {
6    res.send('Hello, World!');
7  });
8
9  app.listen(port, () => {
10    console.log(`App listening at http://localhost:${port}`);
11  });
12
```

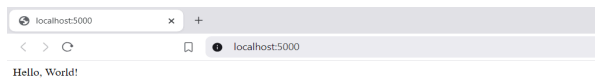
2. Containerize the Application

Creating Docker file for Node.JS Application

Docker File:

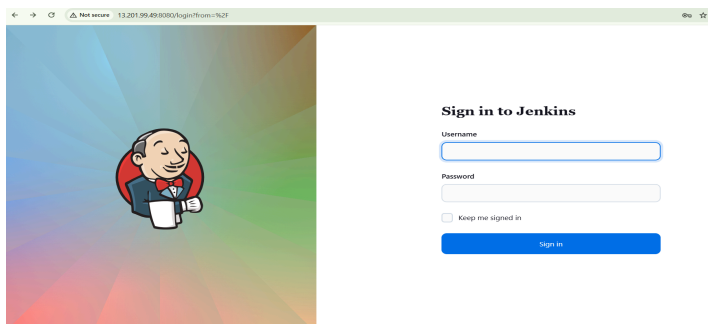
```
Dockerfile > ...  
You, 3 hours ago | 1 author (You)  
1 FROM node:alpine  
2 WORKDIR /app  
3 COPY . .  
4 RUN npm install  
5 COPY package.json .  
6 EXPOSE 5000 You, 3 hours ago * Unco  
7 CMD [ "npm" ,"start" ]
```

Deploying Application Locally:



3. CI/CD Pipeline with Jenkins

Creating Ec2 and installing Jenkins in the ec2



Setting Jenkins Worker to avoid the load on Master

```
ubuntu@ip-172-31-0-48:~$ java -jar agent.jar -url http://13.201.99.49:8080/ -secret lee4cde8fc475f1cda0f1c75c518568da0b40a119fb121be09f299872alb092e -name "Worker-1"
webSocket -workDir "/home/ubuntu"
Jan 09, 2025 7:03:36 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/remoting as a remoting work directory
Jan 09, 2025 7:03:36 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ubuntu/remoting
Jan 09, 2025 7:03:36 AM hudson.remoting.Launcher createEngine
INFO: Setting up agent: Worker-1
Jan 09, 2025 7:03:36 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3261.v9c670a4748a_9
Jan 09, 2025 7:03:36 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ubuntu/remoting as a remoting work directory
Jan 09, 2025 7:03:36 AM hudson.remoting.Launcher$CuiListener status
INFO: WebSocket connection open
Jan 09, 2025 7:03:36 AM hudson.remoting.Launcher$CuiListener status
INFO: Connected
```

Dashboard > Nodes > Worker-1

Status

Delete Agent

Configure

Build History

Load Statistics

Script Console

Log

System Information

Disconnect

Agent Worker-1

worker-1

Agent is connected.

Monitoring Data ▾

Labels

Slave-01

Projects tied to Worker-1

None

Build Executor Status

0/1 ▾

Edit description

Mark this node temporarily offline

ⓘ

Creating Pipeline for Code Build and Docker image build and pushing image into Docker Hub.

```
pipeline {
    agent any

    stages {
        stage('Code CheckOut') {
            steps {
                git branch: 'main', url: 'https://github.com/mohangopi08/NodeDeployment.git'
            }
        }
        stage('Code Build') {
            steps {
                echo 'Build Artificat'
                sh 'npm install'
            }
        }
        stage('Code Test') {
            steps {
                echo 'Run Test Cases'
                sh 'npm test -- --passWithNoTests'
            }
        }
        stage('Build Image') {
            steps {
                echo 'Building Docker Image'
                sh 'sudo docker build -t swargamgopi/nodeapp:${BUILD_NUMBER} . '
            }
        }
        stage('Docker Login') {
            steps {
                withCredentials([string(credentialsId: 'dockerhub', variable: 'dockerhub')]) {
                    sh 'sudo docker login -u swargamgopi -p ${dockerhub}'
                    echo 'login into DockerHub'
                }
            }
        }
        stage('Push To DockerHub') {
            steps {
                sh 'sudo docker push swargamgopi/nodeapp:${BUILD_NUMBER}'
                echo 'Pushed to Docker Hub'
            }
        }
    }
}
```

Build is Success for creating docker images and pushed into docker hub

✓ Node-Deployment

Stage View

		Code CheckOut	Code Build	Code Test	Build Image	Docker Login	Push To DockerHub
Average stage times: (Average full run time: ~42s)		879ms	2s	2s	12s	2s	12s
#27	Jan 09 12:43 No Changes	879ms	2s	2s	12s	2s	12s

Permalinks

- Last build (#27), 2 hr 45 min ago
- Last stable build (#27), 2 hr 45 min ago
- Last successful build (#27), 2 hr 45 min ago
- Last completed build (#27), 2 hr 45 min ago

Code is pushed into the jenkins Worker workspace

```
ubuntu@ip-172-31-0-48:~/workspace/Node-Deployment$ ls -l
total 180
-rw-rw-r-- 1 ubuntu ubuntu 110 Jan 9 07:04 Dockerfile
-rw-rw-r-- 1 ubuntu ubuntu 17 Jan 8 10:09 README.md
drwxrwxr-x 261 ubuntu ubuntu 12288 Jan 9 07:13 node_modules
-rw-rw-r-- 1 ubuntu ubuntu 153131 Jan 9 07:13 package-lock.json
-rw-rw-r-- 1 ubuntu ubuntu 292 Jan 8 10:32 package.json
-rw-rw-r-- 1 ubuntu ubuntu 231 Jan 9 07:04 server.js
ubuntu@ip-172-31-0-48:~/workspace/Node-Deployment$
```

Checking the images is created in Worker Work-space or not

```
ubuntu@ip-172-31-0-48:~/workspace/Node-Deployment$ sudo docker images
REPOSITORY          TAG          IMAGE ID        CREATED         SIZE
swargamgopi/nodeapp 27           933d4abf0357   3 hours ago    188MB
```

Docker Repository is Created in Docker Hub

Name ↑	Last Pushed ↑	Contains	Visibility	Scout
swargamgopi/nodeapp	about 3 hours ago	IMAGE	Public	Inactive

4. AWS Deployment

Deploying into AWS EC2

To Deploy into Ec2 first we SSH into the EC2 by using SSH Agent plugin and set up ec2 Private key in global credentials.

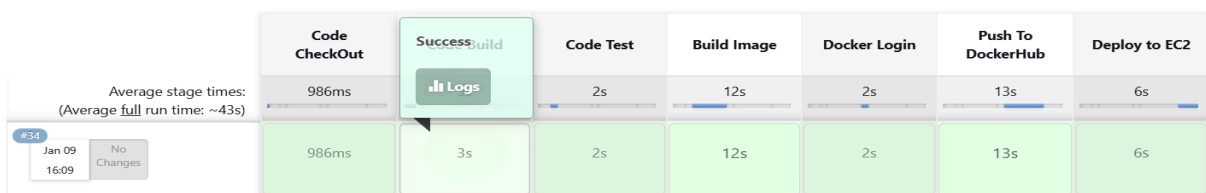
```
stage('Deploy to EC2') {
  steps {
    script {
      withCredentials([sshUserPrivateKey(credentialsId: 'ec2-ssh-key', keyFileVariable: 'SSH_KEY_PATH')]) {
        sh """
        ssh -o StrictHostKeyChecking=no -i ${SSH_KEY_PATH} ubuntu@13.203.161.234 \
        'sudo docker pull swargamgopi/nodeapp:${BUILD_NUMBER} && sudo docker run -d -p 5000:5000 swargamgopi/nodeapp:${BUILD_NUMBER}'
        """
      }
      echo 'Deployment to EC2 successful'
    }
  }
}
```

Ec2 Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	dockerhub	dockerhub
		System	(global)	ec2-ssh-key	ubuntu (ec2-ssh-key)

Build Successful for Deployment

Stage View



Permalinks

Checking Application is Deployed or not

```
ubuntu@ip-172-31-6-42:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
ed4c74643cea   swargangopi/nodeapp:34  "docker-entrypoint.s..."  4 minutes ago  Up 4 minutes  0.0.0.0:5000->5000/tcp, :::5000->5000/tcp  affectionate_albattani
```

Accessing Deployed Application



A screenshot of a web browser window. The address bar shows a green 'Not secure' warning and the URL '13.203.161.234:5000'. The page content displays 'Hello, World!' in a simple black font.

5. Monitoring and Logging

Setting Prometheus and Grafana By using Docker

```
ubuntu@ip-172-31-6-42:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
ed4c74643cea   swargangopi/nodeapp:34  "docker-entrypoint.s..."  8 minutes ago  Up 8 minutes  0.0.0.0:5000->5000/tcp, :::5000->5000/tcp  affectionate_albattani
1f7feeed629c   grafana/grafana  "/run.sh"               3 hours ago   Up 3 hours    0.0.0.0:3000->3000/tcp, :::3000->3000/tcp  grafana
43a896250cc4   prom/prometheus  "/bin/prometheus --c..."  3 hours ago   Up 3 hours    0.0.0.0:9090->9090/tcp, :::9090->9090/tcp  admiring_kilby
ubuntu@ip-172-31-6-42:~$
```

Setting Docker as a target for Prometheus

```
etc/prometheus $ cat prometheus.yml
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

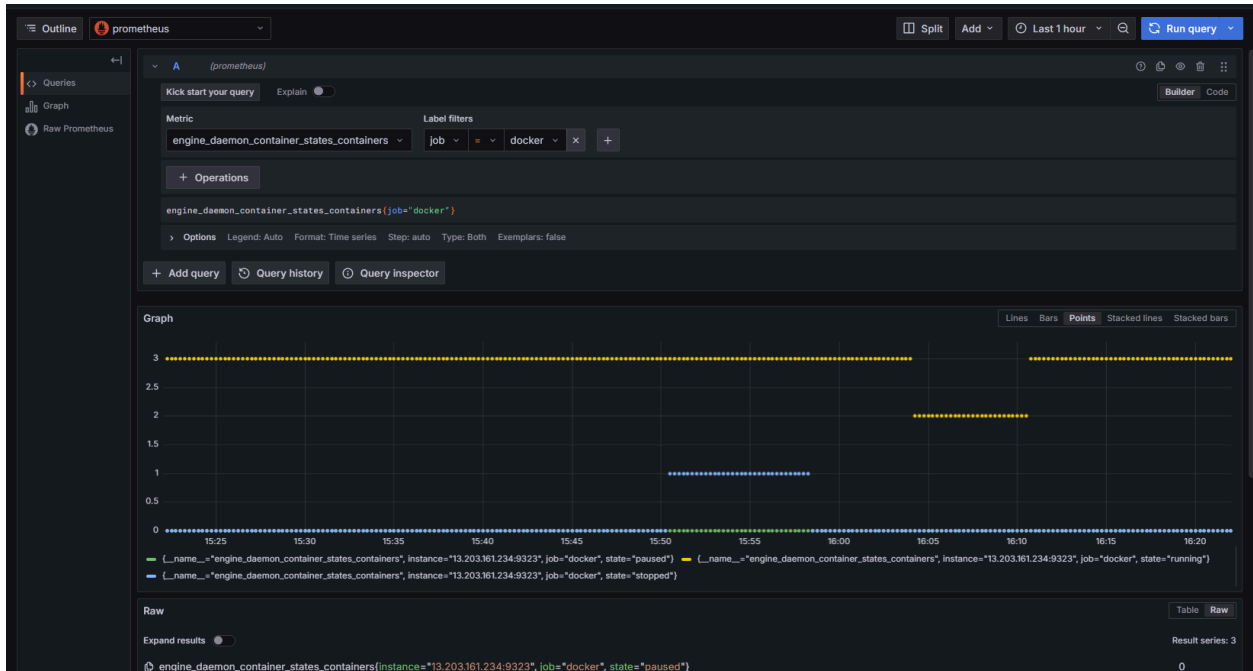
# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

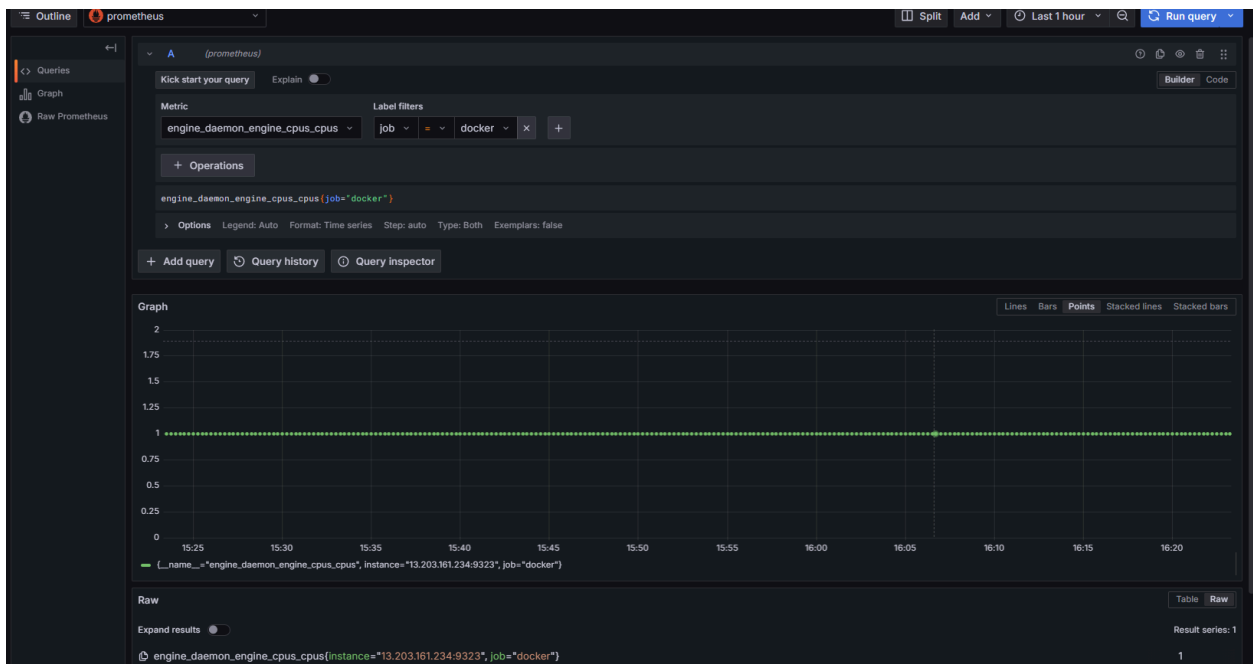
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
      - targets: ["localhost:9090"]

  - job_name: 'docker'
    scrape_interval: 5s
    static_configs:
      - targets: ['13.203.161.234:9323']
```

Monitoring containers By using Grafana



Monitoring Container CPU Metrics by using Grafana



Monitoring Metrics Of application by using Prometheus

```
# HELP builder_builds_failed_total Number of failed image builds
# TYPE builder_builds_failed_total counter
builder_builds_failed_total{reason="build_canceled"} 0
builder_builds_failed_total{reason="build_target_not_reachable_error"} 0
builder_builds_failed_total{reason="command_not_supported_error"} 0
builder_builds_failed_total{reason="dockerfile_empty_error"} 0
builder_builds_failed_total{reason="dockerfile_syntax_error"} 0
builder_builds_failed_total{reason="error_processing_commands_error"} 0
builder_builds_failed_total{reason="missing_onbuild_arguments_error"} 0
builder_builds_failed_total{reason="unknown_instruction_error"} 0
# HELP builder_builds_triggered_total Number of triggered image builds
# TYPE builder_builds_triggered_total counter
builder_builds_triggered_total 0
# HELP engine_daemon_container_actions_seconds The number of seconds it takes to process each container action
# TYPE engine_daemon_container_actions_seconds histogram
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="2.5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="10"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="+Inf"} 1
engine_daemon_container_actions_seconds_sum{action="changes"} 0
engine_daemon_container_actions_seconds_count{action="changes"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="2.5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="10"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="+Inf"} 1
engine_daemon_container_actions_seconds_sum{action="commit"} 0
engine_daemon_container_actions_seconds_count{action="commit"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.1"} 2
engine_daemon_container_actions_seconds_bucket{action="create",le="0.25"} 3
engine_daemon_container_actions_seconds_bucket{action="create",le="0.5"} 3
engine_daemon_container_actions_seconds_bucket{action="create",le="1"} 3
engine_daemon_container_actions_seconds_bucket{action="create",le="2.5"} 3
```

Application Access:- 13.203.161.234:500

Metrics Of application :- 13.203.161.234:9323/metrics

Monitoring :- http://13.203.161.234:3000/