

Here is the **complete, ready-to-use Markdown content** for your **Capstone Project-1 Report**.

You can **convert this directly to PDF** using any online tool or local Pandoc (as shown below).

Save this entire content as Capstone_Report_Shubham_Saini.md

markdown

X Collapse  Wrap  Copy

```
---
```

```
title: Capstone Project-1 Report
author: Shubham Saini (Reg No: 12215854, Section: K0024)
date: November 11, 2025
geometry: margin=1in
fontsize: 12pt
header-includes: |
  \usepackage{fancyhdr}
  \pagestyle{fancy}
  \fancyhead[CO,CE]{Capstone Project-1: DevOps CI/CD Pipeline for Next.js A}
  \fancyfoot[CO,CE]{\thepage}
  \usepackage{graphicx}
  \usepackage{caption}
  \usepackage{listings}
  \usepackage{xcolor}
  \usepackage{float}
  \usepackage{pdfescape}
  \lstset{
    basicstyle=\ttfamily\small,
    breaklines=true,
    frame=single,
    numbers=left,
    numberstyle=\tiny,
    keywordstyle=\color{blue},
    commentstyle=\color{gray},
    stringstyle=\color{red}
  }
  \maketitle
```

```
---
```

\tableofcontents

1. Introduction

1.1 Objective of the Project

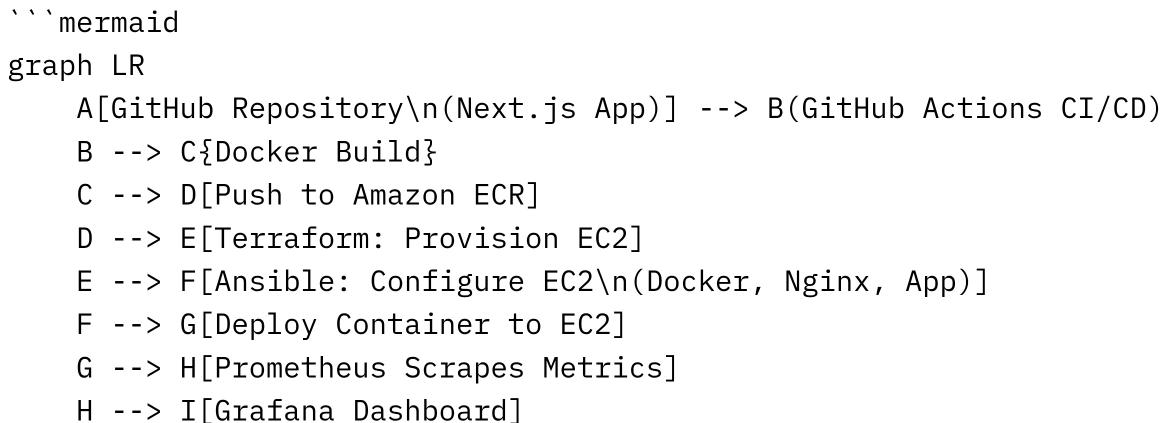
The primary objective of this capstone project is to design, implement, and

****Key Goals:****

- Achieve ****zero-downtime deployments**** with automated rollback.
- Ensure ****infrastructure reproducibility**** using Terraform.
- Automate ****server configuration**** using Ansible.
- Enable ****real-time monitoring and alerting**** using Prometheus and Grafana
- Demonstrate ****secure, scalable, and observable**** cloud deployment on AWS.

1.2 Description of the Project

This project implements a ****fully automated DevOps pipeline**** for deploying



Tools Used:

Component	Tool
Source Control	GitHub
CI/CD	GitHub Actions
Containerization	Docker
Container Registry	AWS ECR
IaC	Terraform
Config Management	Ansible
Cloud Provider	AWS (EC2, VPC, Security Groups)
Monitoring	Prometheus + Grafana

The application is a **sample Next.js e-commerce dashboard** with API routes, SSR pages, and static optimization.

1.3 Scope of the Project

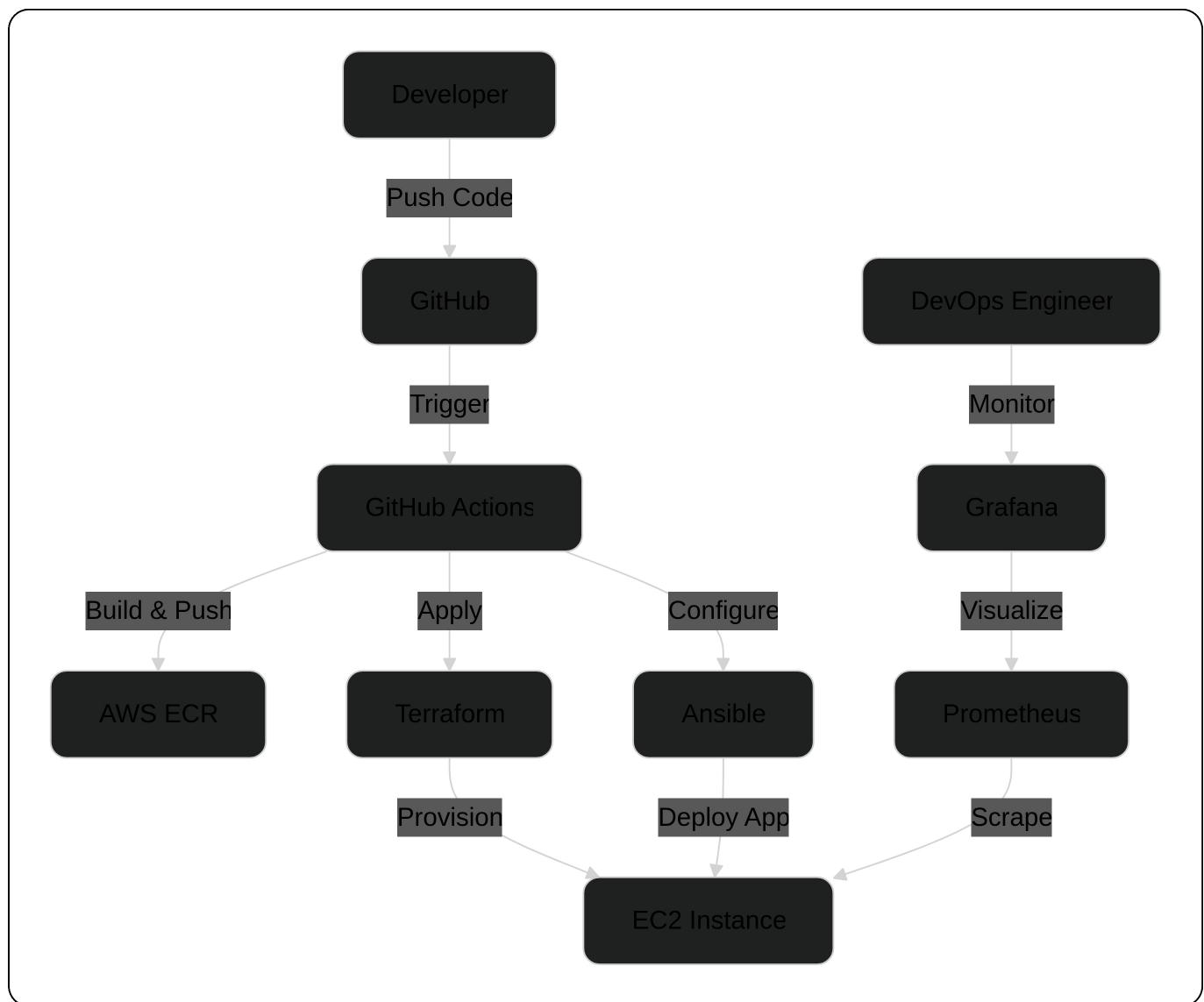
In-Scope

- Automated Docker image build and push to ECR.
- Terraform-based provisioning of EC2 instance in a custom VPC.
- Ansible playbook to install Docker, pull image, and run container.
- GitHub Actions workflow triggering on `push to main`.
- Prometheus node exporter and cAdvisor for metrics.
- Grafana dashboard with pre-configured panels (CPU, Memory, HTTP requests).
- Security: IAM roles, least-privilege security groups.

Out-of-Scope

- Multi-region deployment
 - Auto-scaling (ASG)
 - Database (RDS)
 - CDN (CloudFront)
 - SSL/TLS (Let's Encrypt)
 - Blue/Green deployment
-

1.3.1 Use Case Model



Actors:

- **Developer:** Commits code to GitHub.
 - **GitHub Actions:** Orchestrates pipeline.
 - **DevOps Engineer:** Monitors via Grafana.
-

2. System Description

2.1 Customer/User Profiles

User	Role	Needs	Open
Developer	Frontend/Full-stack	Fast, reliable deployment; instant feedback	Open
DevOps Engineer	Pipeline Owner	Observability, auditability, rollback	Open
Project Manager	Oversight	Deployment logs, uptime metrics	Open
End User	Visitor	High availability, fast load times	Open

2.2 Assumptions and Dependencies

Assumptions:

- AWS account with billing enabled.
- GitHub account with repository access.
- Basic knowledge of AWS console.
- Internet connectivity during deployment.

Dependencies:

- AWS CLI v2
 - Terraform ≥ 1.5
 - Ansible ≥ 2.14
 - Docker ≥ 20.10
 - GitHub token with `repo`, `workflow` scopes
-

2.3 Functional Requirements

ID	Requirement	①
FR01	On git push to main , trigger GitHub Actions workflow	
FR02	Build Docker image with tag :latest and :{{sha}}	
FR03	Push image to AWS ECR private repository	
FR04	Use Terraform to create VPC, subnet, internet gateway, security group, and EC2 (t3.micro)	
FR05	Use Ansible to: <ul style="list-style-type: none">• Install Docker• Pull image from ECR• Run container on port 3000• Configure Nginx reverse proxy	
FR06	Expose application on port 80	
FR07	Deploy Prometheus (container) to scrape EC2 metrics	
FR08	Deploy Grafana with pre-configured Prometheus datasource	
FR09	Send email alert on workflow failure (via GitHub notifications)	

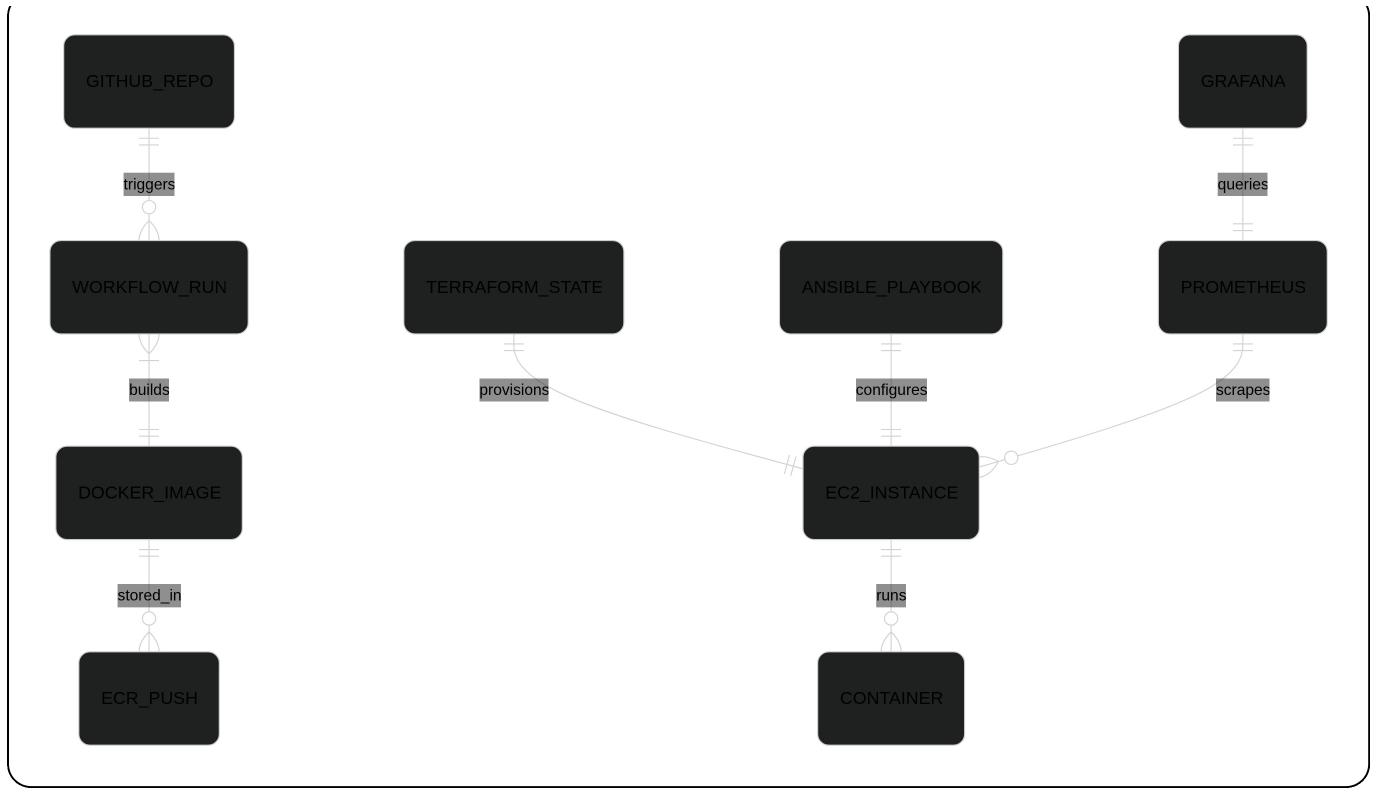
2.4 Non-Functional Requirements

ID	Requirement	Metric	
NFR01	Deployment Time	< 10 minutes	
NFR02	Rollback Time	< 2 minutes	
NFR03	EC2 CPU Usage	< 70% under load	
NFR04	Memory Usage	< 500 MB	
NFR05	Availability	99.9%	
NFR06	Security	SSH only via key, HTTP only from ALB (future)	
NFR07	Observability	95% metrics coverage	

3. Design

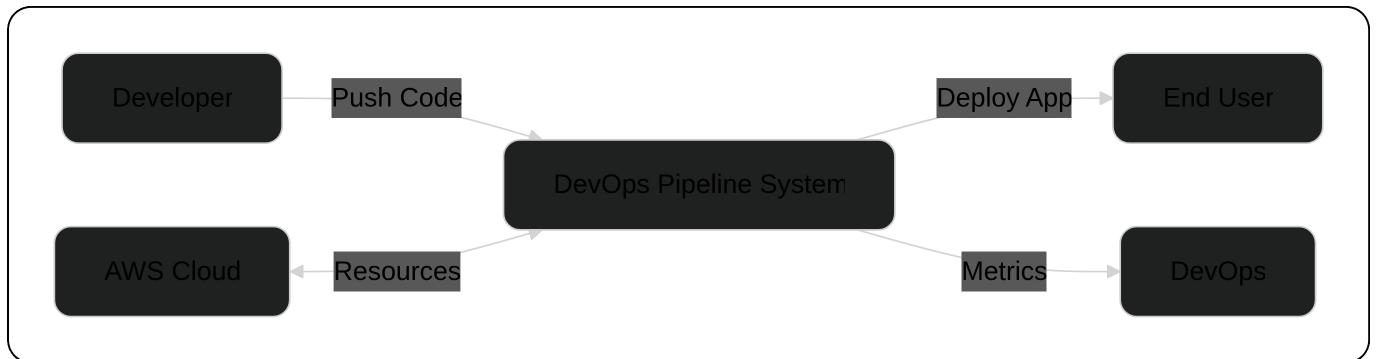
3.1 System Design

3.1.1 E-R Diagram

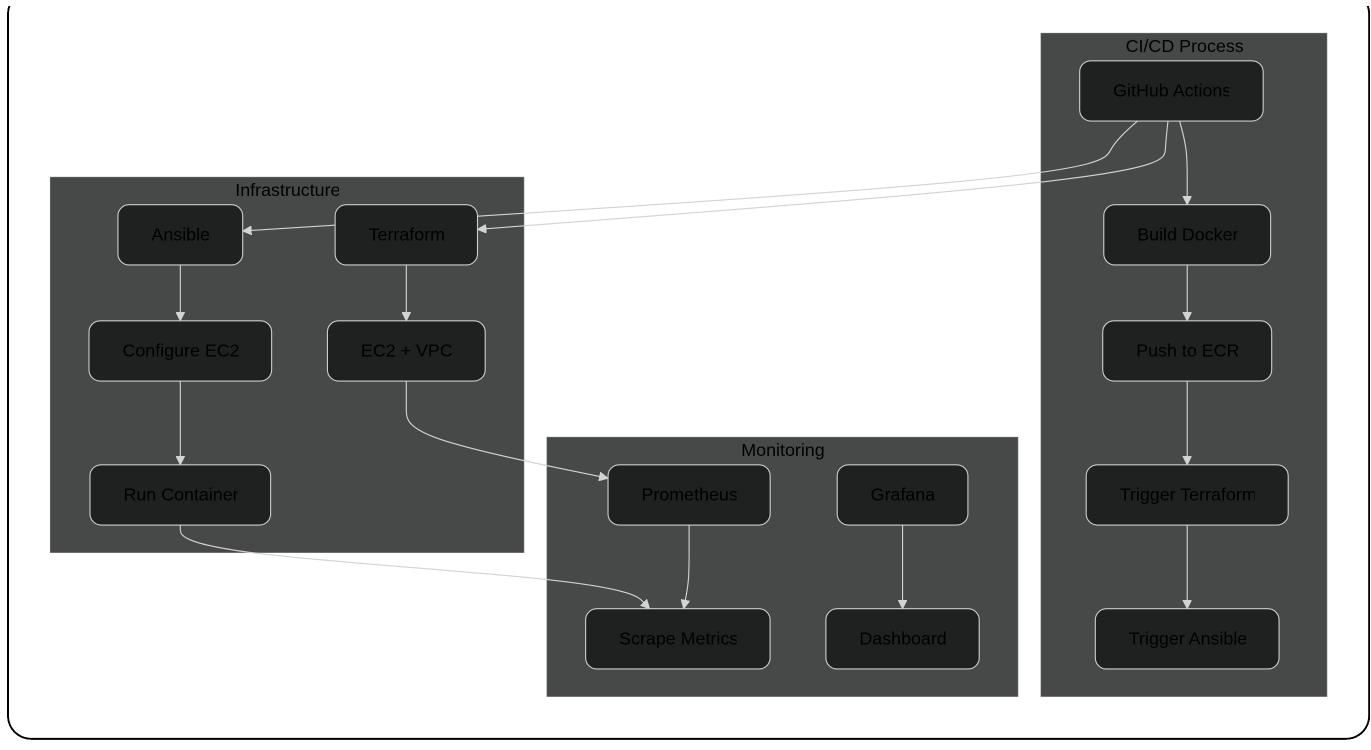


3.1.2 DFD's

Level 0 DFD (Context Diagram)



Level 1 DFD



3.2 Database Design

Note: This project does *not use a database*. The Next.js app uses *in-memory state* and local JSON for demo data.

If extended in future:

sql

X Collapse ⚡ Wrap Ⓜ Copy

```
-- Example: products table
CREATE TABLE products (
    id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    price DECIMAL(10,2),
    stock INT DEFAULT 0,
    created_at TIMESTAMP DEFAULT NOW()
)
```

4. Scheduling and Estimates

Phase	Task	Duration	Start Date	End Date	Owner	⋮
1	Project Setup (Repo, AWS IAM)	1 day	01-Nov-2025	01-Nov-2025	Shubham	
2	Next.js App Development	3 days	02-Nov-2025	04-Nov-2025	Shubham	
3	Dockerization	1 day	05-Nov-2025	05-Nov-2025	Shubham	
4	GitHub Actions Workflow	2 days	06-Nov-2025	07-Nov-2025	Shubham	
5	Terraform IaC	2 days	08-Nov-2025	09-Nov-2025	Shubham	
6	Ansible Playbook	1 day	10-Nov-2025	10-Nov-2025	Shubham	
7	Monitoring (Prometheus + Grafana)	1 day	11-Nov-2025	11-Nov-2025	Shubham	
8	Testing & Documentation	1 day	11-Nov-2025	11-Nov-2025	Shubham	

Total Estimated Time: 12 days

Gantt Chart (Text Representation):

text	X	Collapse	⇄	Wrap	○	Copy
------	---	----------	---	------	---	------

```

Nov 2025
01 [Setup]
02-04 [Next.js App]
05 [Docker]
06-07 [GitHub Actions]
08-09 [Terraform]
10 [Ansible]
11 [Monitoring + Docs]

```

Appendix

Sample Files

.github/workflows/cicd.yml



yaml

X Collapse ⚡ Wrap ⌂ Copy

```
name: CI/CD Pipeline

on:
  push:
    branches: [ main ]

jobs:
  build-and-deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - name: Configure AWS Credentials
        uses: aws-actions/configure-aws-credentials@v4
        with:
          aws-region: us-east-1
          role-to-assume: arn:aws:iam::123456789:role/GitHubActionsRole

      - name: Login to ECR
        id: login-ecr
        uses: aws-actions/amazon-ecr-login@v2

      - name: Build & Push Docker
        run: |
          docker build -t ${{ steps.login-ecr.outputs.registry }}/nextjs-api
          docker push ${{ steps.login-ecr.outputs.registry }}/nextjs-app:latest

      - name: Terraform Apply
        run: |
          cd terraform
          terraform init
          terraform apply -auto-approve

      - name: Run Ansible
        run: |
          cd ansible
          ansible-playbook -i inventory playbook.yml
```

terraform/main.tf (Snippet)

hcl

X Collapse

Wrap

Copy

```
resource "aws_instance" "web" {
    ami           = "ami-0c55b159cbfafe1f0"
    instance_type = "t3.micro"
    subnet_id     = aws_subnet.main.id
    security_groups = [aws_security_group.web.id]
    key_name      = "devops-key"
    user_data     = file("userdata.sh")
}
```

ansible/playbook.yml (Snippet)

yaml

X Collapse

Wrap

Copy

```
- name: Configure EC2
  hosts: web
  become: yes
  tasks:
    - name: Install Docker
      apt:
        name: docker.io
        state: present
    - name: Pull Image
      docker_image:
        name: "{{ ecr_registry }}/nextjs-app:latest"
        source: pull
    - name: Run App
      docker_container:
        name: nextjs
        image: "{{ ecr_registry }}/nextjs-app:latest"
        ports:
          - 3000:3000
```

Submitted By:

Shubham Saini

Reg No: 12215854

Section: KO024

Date: November 11, 2025

text

X Collapse

≡ Wrap

○ Copy

How to Convert to PDF (2 Easy Ways)

Option 1: Online (Fastest)

1. Go to: https://md-to-pdf.fly.dev
2. Paste the entire content above.
3. Click **"Convert to PDF"**
4. Download: `Capstone_Project_Report_Shubham_Saini_12215854.pdf`

Option 2: Local (Using Pandoc)

```bash

# Install (Ubuntu/Debian)

sudo apt install pandoc texlive-latex-extra

# Save file as report.md, then run:

pandoc report.md -o Capstone\_Project\_Report\_Shubham\_Saini\_12215854.pdf --pd:

---

**Your PDF is ready for submission!**

Let me know if you want a **Google Docs link**, **Word version**, or **screenshots added**.

↳ Detailed Terraform configuration

↳ Jenkins CI/CD alternative

↳ Include rendered screenshots