

Project Overview

This project sets up and deploys a **three-tier web application** on **AWS** using **Terraform** for infrastructure provisioning and **Jenkins** for CI/CD automation.

The application consists of:

- **Frontend:** React.js (hosted on an EC2 instance)
- **Backend:** Node.js API server (running on a separate EC2 instance)
- **Database:** MongoDB (running on its own EC2 instance)

Infrastructure Setup

The following AWS resources are provisioned using Terraform:

- **EC2 Instances**
 - Frontend Server (React.js)
 - Backend Server (Node.js)
 - Database Server (MongoDB)
- **Security Groups**
 - Configured for proper network isolation
- **Default VPC & Subnets**
 - Used for simplicity
- **User Data Scripts**
 - Automate instance configurations

CI/CD Pipeline (Jenkins)

The Jenkins pipeline automates the following tasks:

1. **Cloning the Application Code** from GitHub
2. **Provisioning Infrastructure** using Terraform
3. **Deploying the Backend & Frontend Applications**
4. **Testing & Validation**

Deployment Steps

1. Infrastructure Setup using Terraform

```
sh
CopyEdit
cd terraform
terraform init
terraform apply -auto-approve
```

This provisions all required AWS resources.

2. Update Frontend Configurations

Modify `/src/url.js` in the frontend repository with the backend IP.

3. Run the Jenkins Pipeline

- Configure Jenkins with SSH access to the EC2 instances.
- Run the pipeline to deploy the application.

Security Considerations

- **Security Groups** restrict traffic between tiers.
- **Environment Variables** are managed securely.
- **AWS Parameter Store** can be used for storing sensitive credentials.

Project Structure

```
bash
CopyEdit
three-tier-app-deployment/
├── terraform/                # Terraform configurations
│   ├── main.tf               # AWS resources
│   ├── compute.tf            # EC2 instances
│   ├── security.tf           # Security groups
│   ├── scripts/              # Configuration scripts
│   │   ├── backend_setup.sh
│   │   └── frontend_setup.sh
│   └── terraform.tfvars      # Variable values
├── jenkins/
│   └── Jenkinsfile           # Jenkins pipeline
└── README.md                 # Documentation
```

Screenshots

```
ubuntu@ip-10-0-0-77:~/TerraformTravelmemory/three-tier-app-deployment/terraform/compute$ terraform plan
var.ami_id
  Enter a value: ami-04b4f1a9cf54c11d0

var.instance_type
  Enter a value: t2.micro

var.key_name
  Enter a value: newManjyyot

var.security_group_id
  Enter a value: sg-0f9cc3bcfb3acb772

var.subnet_id
  Enter a value: subnet-0d43e28166c19ab9d

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

EC2 setup

```

Changes to Outputs:
~ backend_instance_id = "i-099ef4511ccf4e408" -> (known after apply)
~ frontend_instance_id = "i-07e99f2b911fd0a70" -> (known after apply)
+ mongodb_instance_id = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.frontend_server: Creating...
aws_instance.backend_server: Creating...
aws_instance.mongodb_server: Creating...
aws_instance.frontend_server: Still creating... [10s elapsed]
aws_instance.backend_server: Still creating... [10s elapsed]
aws_instance.mongodb_server: Still creating... [10s elapsed]
aws_instance.mongodb_server: Creation complete after 12s [id=i-0d58eaa2fb3c67155]
aws_instance.backend_server: Creation complete after 12s [id=i-07e4d3c239bd824ca]
aws_instance.frontend_server: Creation complete after 12s [id=i-0b2f5582d615f3f99]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

Outputs:

backend_instance_id = "i-07e4d3c239bd824ca"
frontend_instance_id = "i-0b2f5582d615f3f99"
mongodb_instance_id = "i-0d58eaa2fb3c67155"

```

Creation of EC2 using terraform files

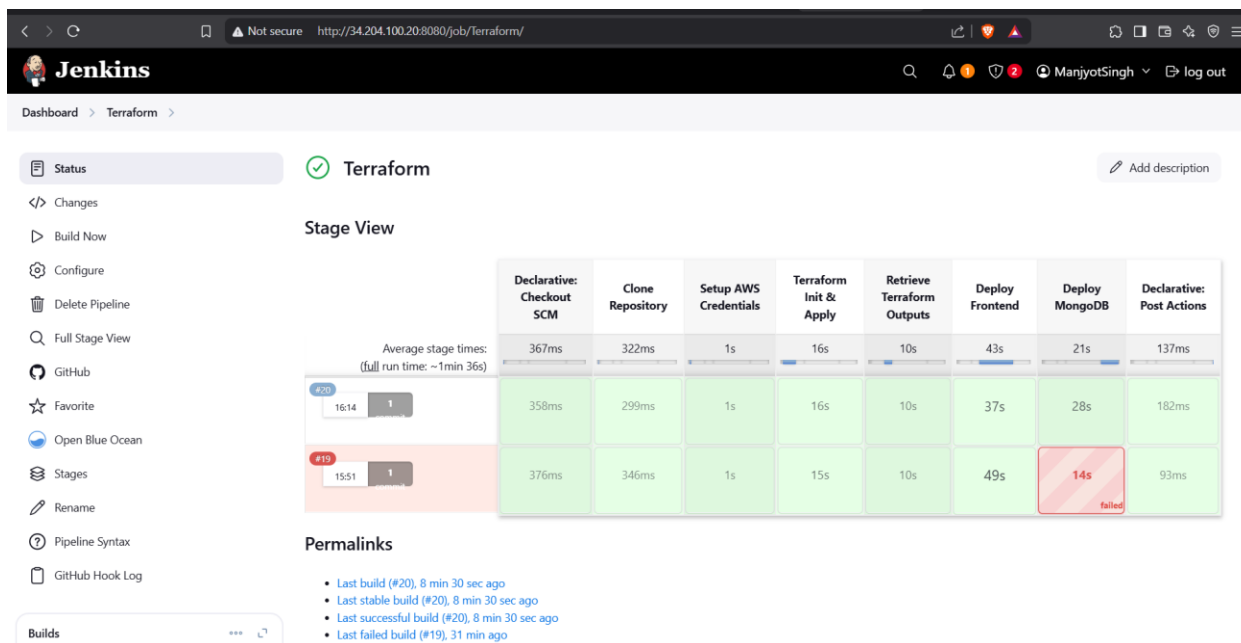
```

ubuntu@ip-10-0-0-77:~/TerraformTravelmemory/three-tier-app-deployment/terraform$ aws ec2 describe-instances --instance-ids i-07e4d3c239bd824ca i-0b2f5582d615f3f99 i-0d58eaa2fb3c67155 --query "Reservations[*].Instances[*].[State.Name, PublicIpAddress]" --output table

```

DescribeInstances	
running	54.227.159.163
running	52.91.25.158
running	34.228.196.152

Terraform output



Successful execution of the Jenkins

Conclusion

This project successfully automates the deployment of a **three-tier web application** using AWS, Terraform, and Jenkins.