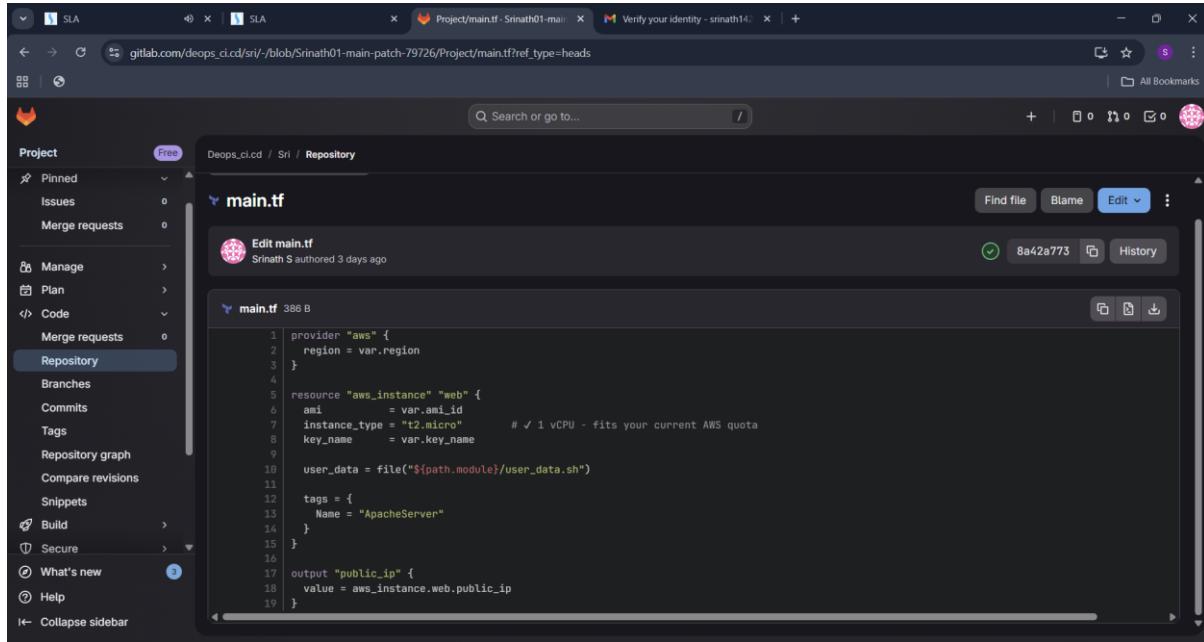


TASK-17

Gitlab pipeline for Infrastructure Terraform

Create Terraform main file:



The screenshot shows a GitLab interface for a project named "Deops_ci_cd". The sidebar on the left is collapsed. The main area displays the "main.tf" file under the "Repository" tab. The code in the file is as follows:

```
provider "aws" {
  region = var.region
}

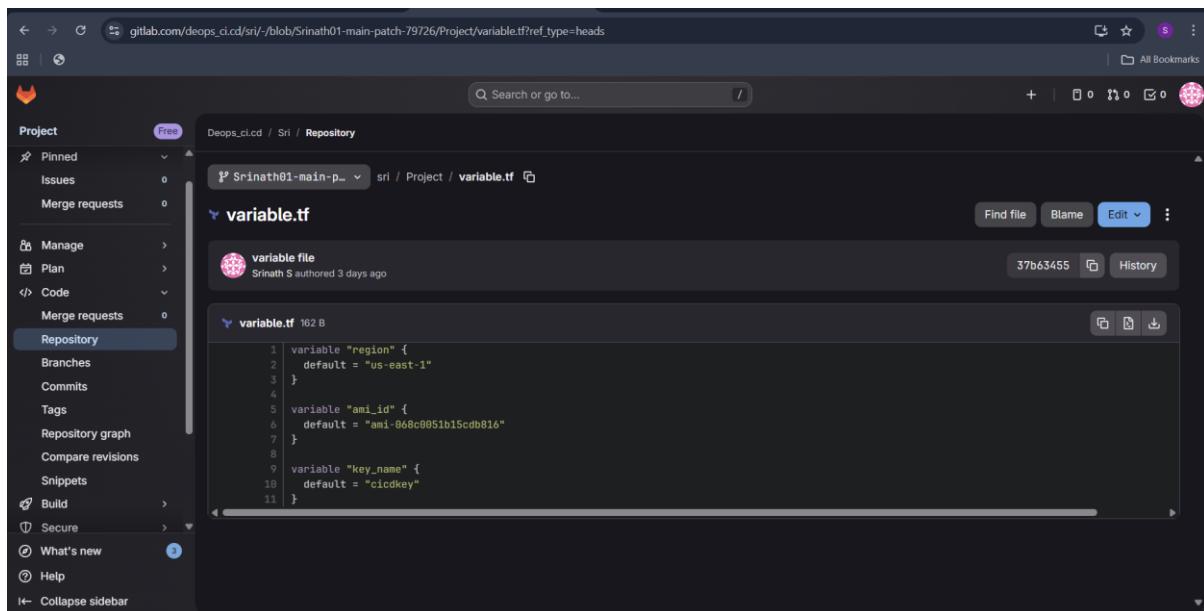
resource "aws_instance" "web" {
  ami           = var.ami_id
  instance_type = "t2.micro"      # ✓ 1 vCPU - fits your current AWS quota
  key_name      = var.key_name

  user_data = file("${path.module}/user_data.sh")

  tags = {
    Name = "ApacheServer"
  }
}

output "public_ip" {
  value = aws_instance.web.public_ip
}
```

Create Variable File:



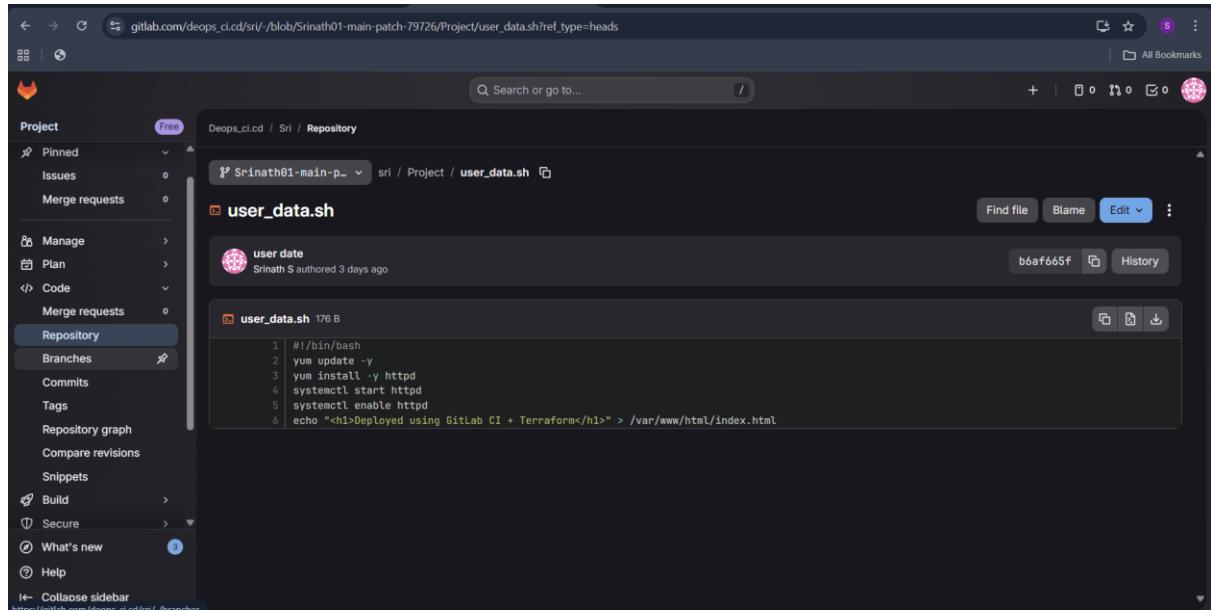
The screenshot shows a GitLab interface for a project named "Srinath01-main-p...". The sidebar on the left is collapsed. The main area displays the "variable.tf" file under the "Project / Repository" tab. The code in the file is as follows:

```
variable "region" {
  default = "us-east-1"
}

variable "ami_id" {
  default = "ami-068c0051b15cdb816"
}

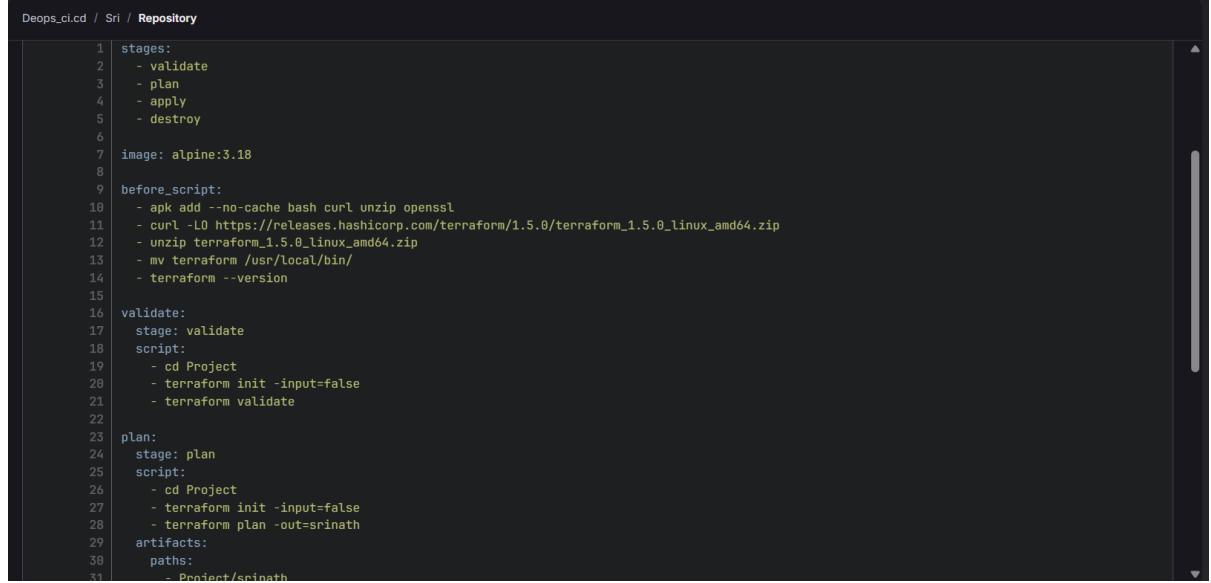
variable "key_name" {
  default = "cidkkey"
}
```

Create User_date:



```
#!/bin/bash
```

Write a yml file as .gitlab-ci.yml:



```
stages:
- validate
- plan
- apply
- destroy

image: alpine:3.18

before_script:
- apk add --no-cache bash curl unzip openssl
- curl -L https://releases.hashicorp.com/terraform/1.5.0/terraform_1.5.0_linux_amd64.zip
- unzip terraform_1.5.0_linux_amd64.zip
- mv terraform /usr/local/bin/
- terraform --version

validate:
stage: validate
script:
- cd Project
- terraform init -input=false
- terraform validate

plan:
stage: plan
script:
- cd Project
- terraform init -input=false
- terraform plan -out=srinath
artifacts:
paths:
- Project/srinath
```

Code:

stages:

- validate
- plan
- apply

- destroy

image: alpine:3.18

before_script:

- apk add --no-cache bash curl unzip openssl

- curl -LO

https://releases.hashicorp.com/terraform/1.5.0/terraform_1.5.0_linux_amd64.zip

- unzip terraform_1.5.0_linux_amd64.zip

- mv terraform /usr/local/bin/

- terraform --version

validate:

stage: validate

script:

- cd Project

- terraform init -input=false

- terraform validate

plan:

stage: plan

script:

- cd Project

- terraform init -input=false

- terraform plan -out=srinath

artifacts:

paths:

- Project/srinath

apply:

 stage: apply

 when: manual

 dependencies:

 - plan

 script:

 - cd Project

 - terraform init -input=false

 - terraform apply -auto-approve srinath

 artifacts:

 paths:

 - Project/terraform.tfstate

 - Project/terraform.tfstate.backup

 expire_in: 1 hour

destroy:

 stage: destroy

 when: manual

 dependencies:

 - apply

Pipe Line:

The screenshot shows a GitLab pipeline interface. On the left, there's a sidebar with various project management options like Pinned issues, Merge requests, Manage, Plan, Code, Build, Pipelines, Jobs, Pipeline editor, Pipeline schedules, Artifacts, Secure, Deploy, Operate, Monitor, What's new, and Help. The Pipelines section is currently selected. The main area is titled "Edit .gitlab-ci.yml" and shows a pipeline run for commit 213beb31. It indicates a "Passed" status, created 2 days ago by Srinath S., finished 1 minute ago. The pipeline has 8 jobs, 6.06 minutes total, and was queued for 2 seconds. The pipeline stages shown are validate, plan, apply, and destroy, each with a green checkmark indicating success.

Instance Created

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, and AMI Catalog. The main content area displays a table titled "Instances (1) Info" with one row. The row details an instance named "ApacheServer" with Instance ID i-0225b7b37cccd5611d, Instance state Running, Instance type t2.micro, Status check Initializing, Alarm status View alarms +, Availability Zone us-east-1b, and Public IPv4 ec2-50-19-1. There is also a "Select an instance" dropdown below the table.

Instanced Deleted:

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main content area displays a table titled "Instances (1) Info". The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. One row is present, labeled "ApacheServer" with Instance ID "i-0225b7b57cccd5611d", Instance state "Terminated", Instance type "t2.micro", and Availability Zone "us-east-1b". Below the table, a modal window titled "Select an instance" is open, listing the same instance "ApacheServer". The bottom of the screen shows standard AWS navigation links: CloudShell, Feedback, Console Mobile App, and footer links: © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
ApacheServer	i-0225b7b57cccd5611d	Terminated	t2.micro	-	View alarms +	us-east-1b	-

Select an instance