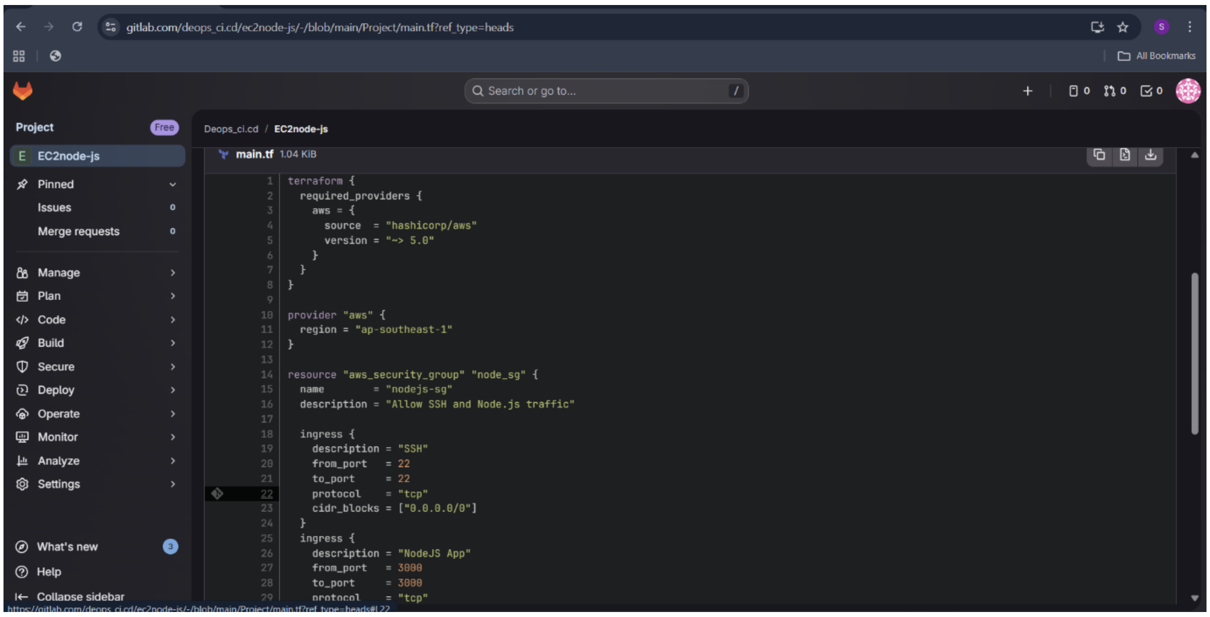
Task-18

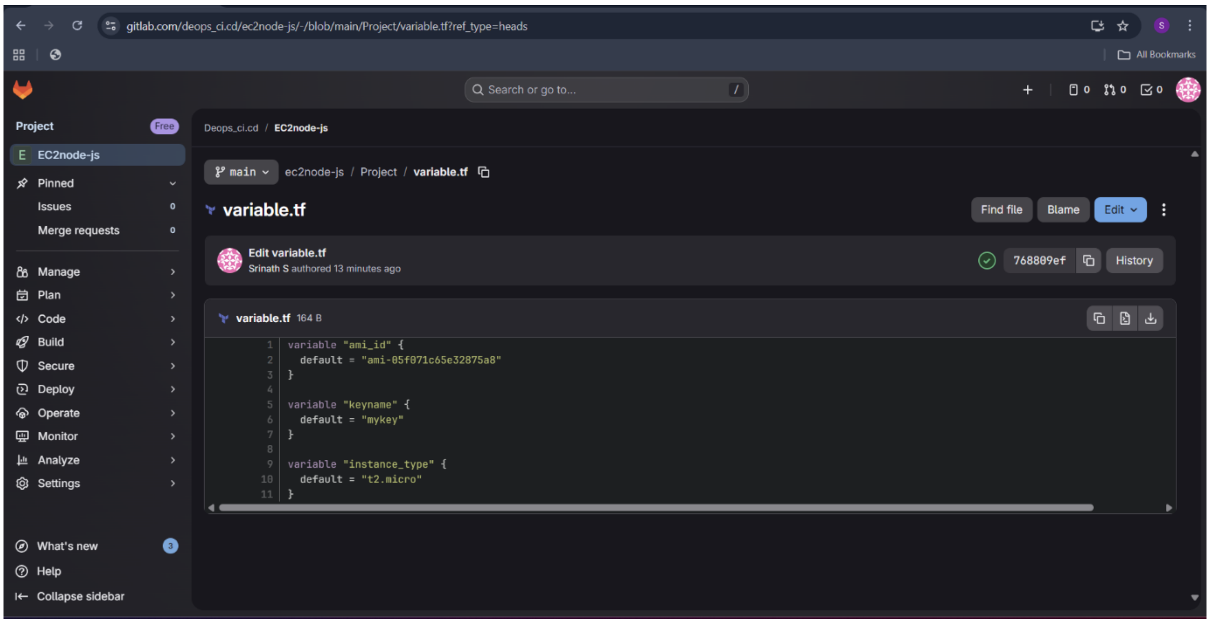
Deploy the Node js hands on - in the EC2

Create main.tf in Gitlab:



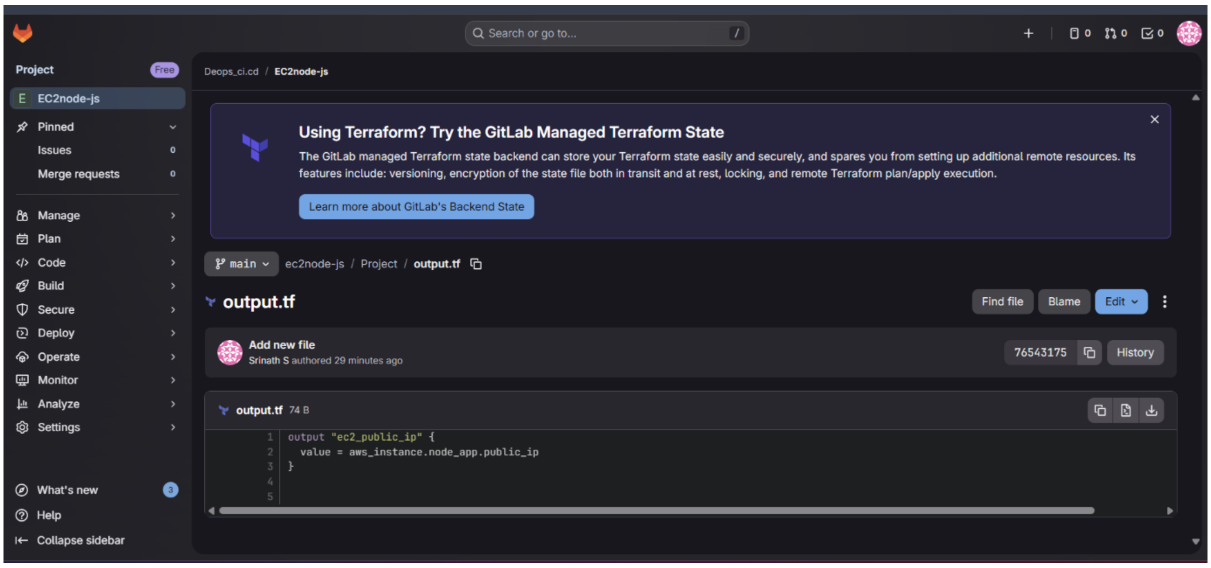
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region = "ap-southeast-1"
}

resource "aws_security_group" "node_sg" {
  name        = "nodejs-sg"
  description = "Allow SSH and Node.js traffic"

  ingress {
    description = "SSH"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  ingress {
    description = "NodeJS App"
    from_port   = 3000
    to_port     = 3000
    protocol    = "tcp"
```

Create Variable.tf file:



```
variable "ami_id" {
  default = "ami-05f071c65e32875a8"
}

variable "keyname" {
  default = "mykey"
}

variable "instance_type" {
  default = "t2.micro"
}
```

Create output.tf file:


Using Terraform? Try the GitLab Managed Terraform State

The GitLab managed Terraform state backend can store your Terraform state easily and securely, and spares you from setting up additional remote resources. Its features include: versioning, encryption of the state file both in transit and at rest, locking, and remote Terraform plan/apply execution.

Deops_ci.cd / EC2node-js

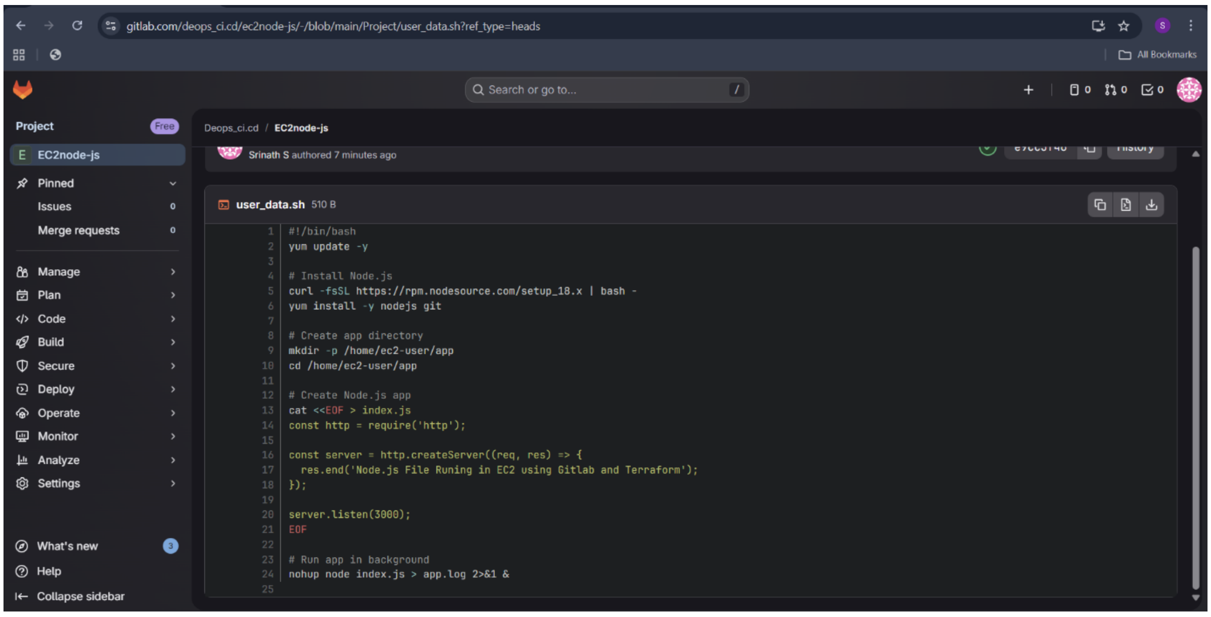
main ec2node-js / Project / output.tf

Add new file
Srinath S authored 29 minutes ago 76543175

output.tf 74 B

```
output "ec2_public_ip" {
  value = aws_instance.node_app.public_ip
}
```

Create user data File:


gitlab.com/deops_ci.cd/ec2node-js/-/blob/main/Project/user_data.sh?ref_type=heads

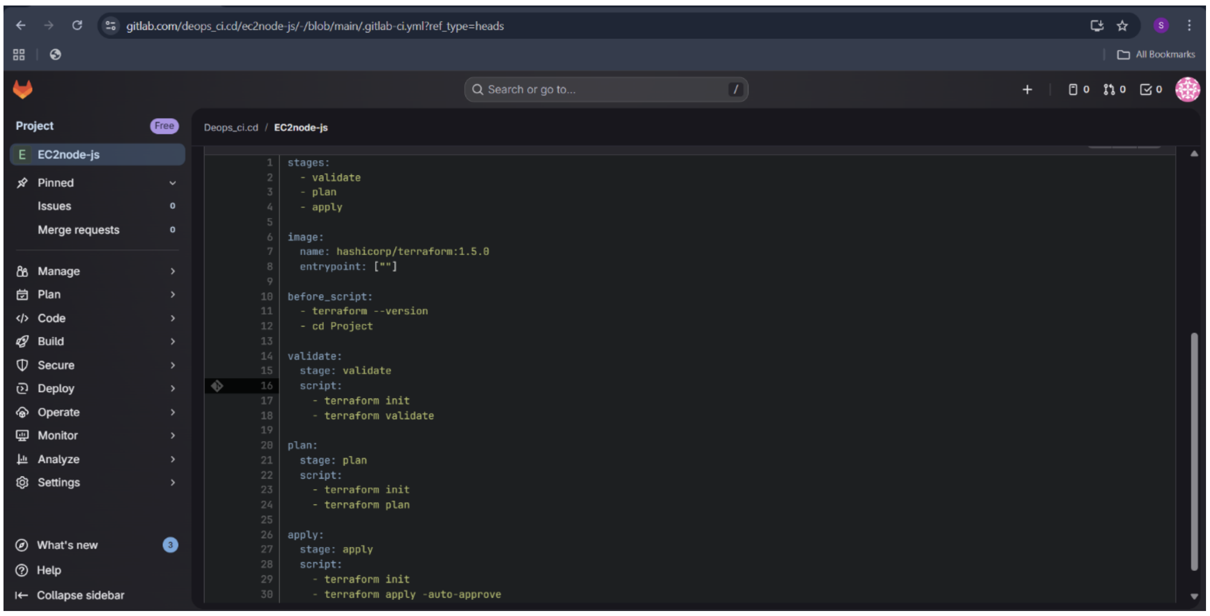
Deops_ci.cd / EC2node-js

Srinath S authored 7 minutes ago

user_data.sh 510 B

```
#!/bin/bash
yum update -y

# Install Node.js
curl -fsSL https://rpm.nodesource.com/setup_18.x | bash -
yum install -y nodejs git

# Create app directory
mkdir -p /home/ec2-user/app
cd /home/ec2-user/app

# Create Node.js app
cat <<EOF > index.js
const http = require('http');

const server = http.createServer((req, res) => {
  res.end('Node.js File Runing in EC2 using Gitlab and Terraform');
});

server.listen(3000);
EOF

# Run app in background
nohup node index.js > app.log 2>&1 &
```

Create .gitlab-ci.yml file:

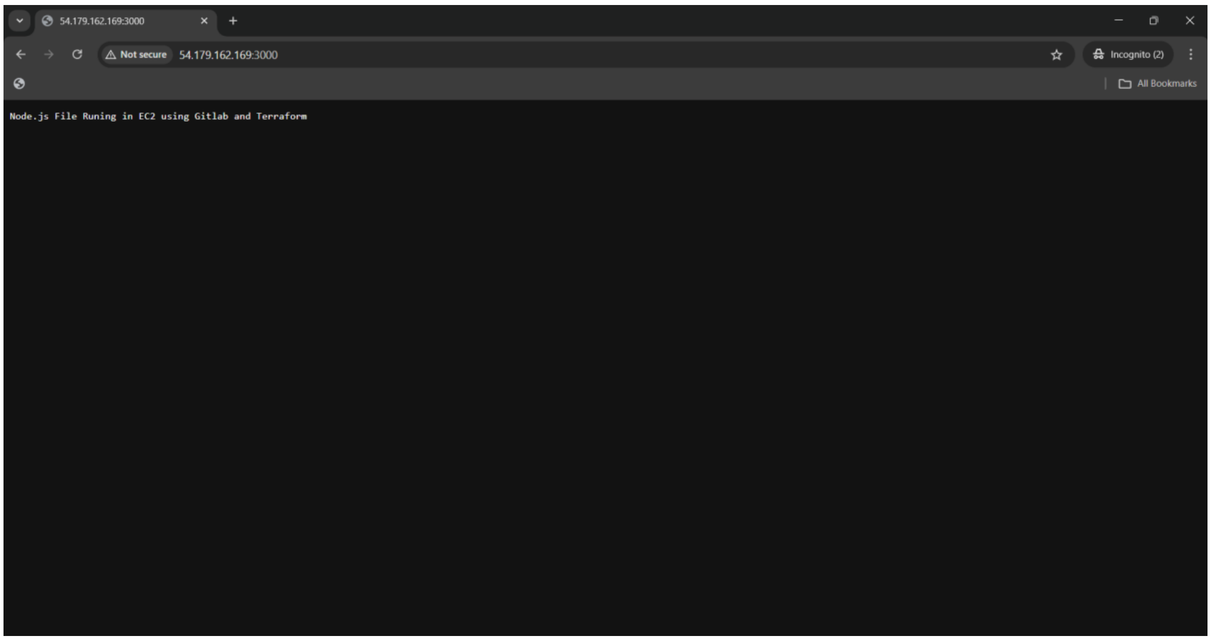


```
stages:
  - validate
  - plan
  - apply

image:
  name: hashicorp/terraform:1.5.0
  entrypoint: [""]

before_script:
  - terraform --version
  - cd Project

validate:
  stage: validate
  script:
    - terraform init
    - terraform validate

plan:
  stage: plan
  script:
    - terraform init
    - terraform plan

apply:
  stage: apply
  script:
    - terraform init
    - terraform apply -auto-approve
```

Code:

```
stages:
  - validate
  - plan
  - apply

image:
  name: hashicorp/terraform:1.5.0
  entrypoint: [""]

before_script:
  - terraform --version
  - cd Project

validate:
  stage: validate
  script:
    - terraform init
    - terraform validate

plan:
  stage: plan
  script:
    - terraform init
    - terraform plan

apply:
  stage: apply
  script:
    - terraform init
    - terraform apply -auto-approve
```

Output:



```
Node.js File Runing in EC2 using Gitlab and Terraform
```