**Creating an EC2 using Terraform Automation in AWS**

This guide helps you automate an EC2 instance in AWS using Terraform Automation.

**Step1:** Navigate to AWS Dashboard.

-sign in to AWS console.

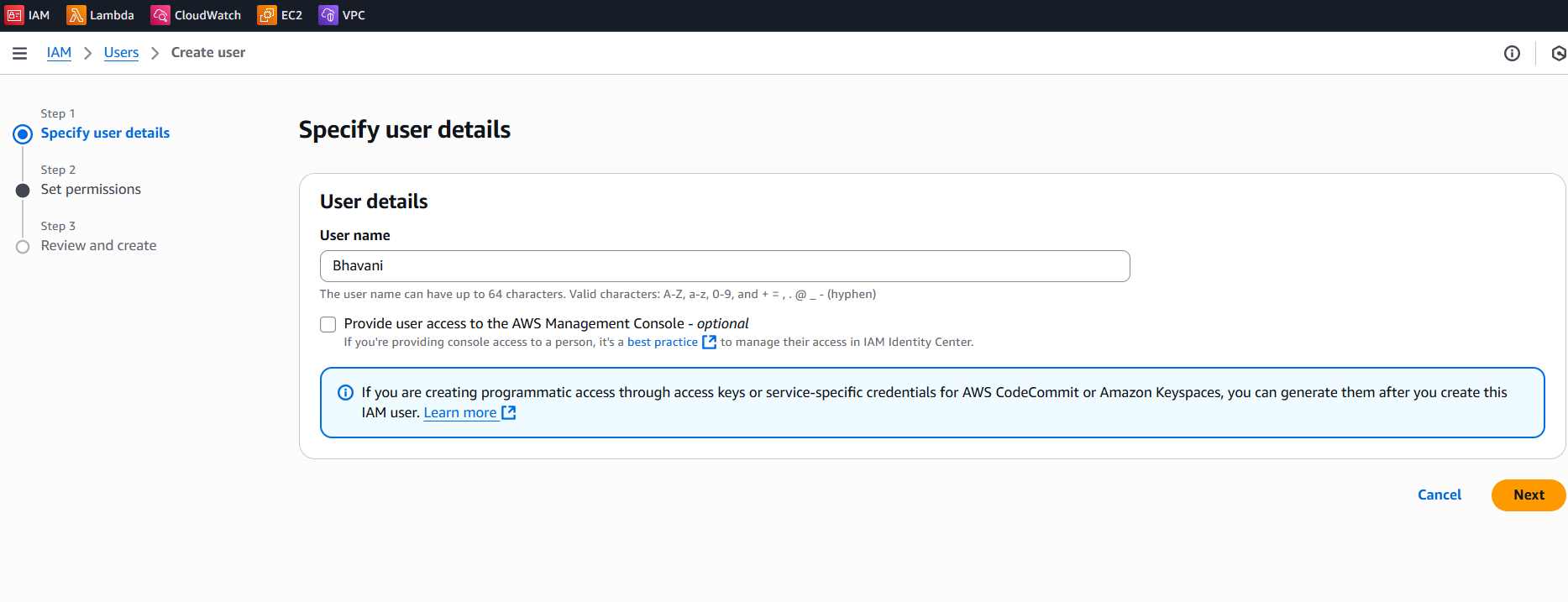
-Goto IAM Dashboard.

**Step2:** Create a user in the IAM dashboard, In the Specify user details

-Add Name ->Click on next set permissions

->Review and create.

**Step 3:** In security credentials, Create new Access Key and password.



A screenshot of a computer

AI-generated content may be incorrect.

**You also need this prerequisite to automate this infrastructure.**

**Prerequisites**

* The [Terraform CLI](https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli) (1.2.0+) installed.
* The [AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html) installed.
* [AWS account](https://aws.amazon.com/free) and [associated credentials](https://docs.aws.amazon.com/general/latest/gr/aws-sec-cred-types.html) that allow you to create resources.

To use your IAM credentials to authenticate the Terraform AWS provider, set the AWS\_ACCESS\_KEY\_ID environment variable.

export AWS\_ACCESS\_KEY\_ID=

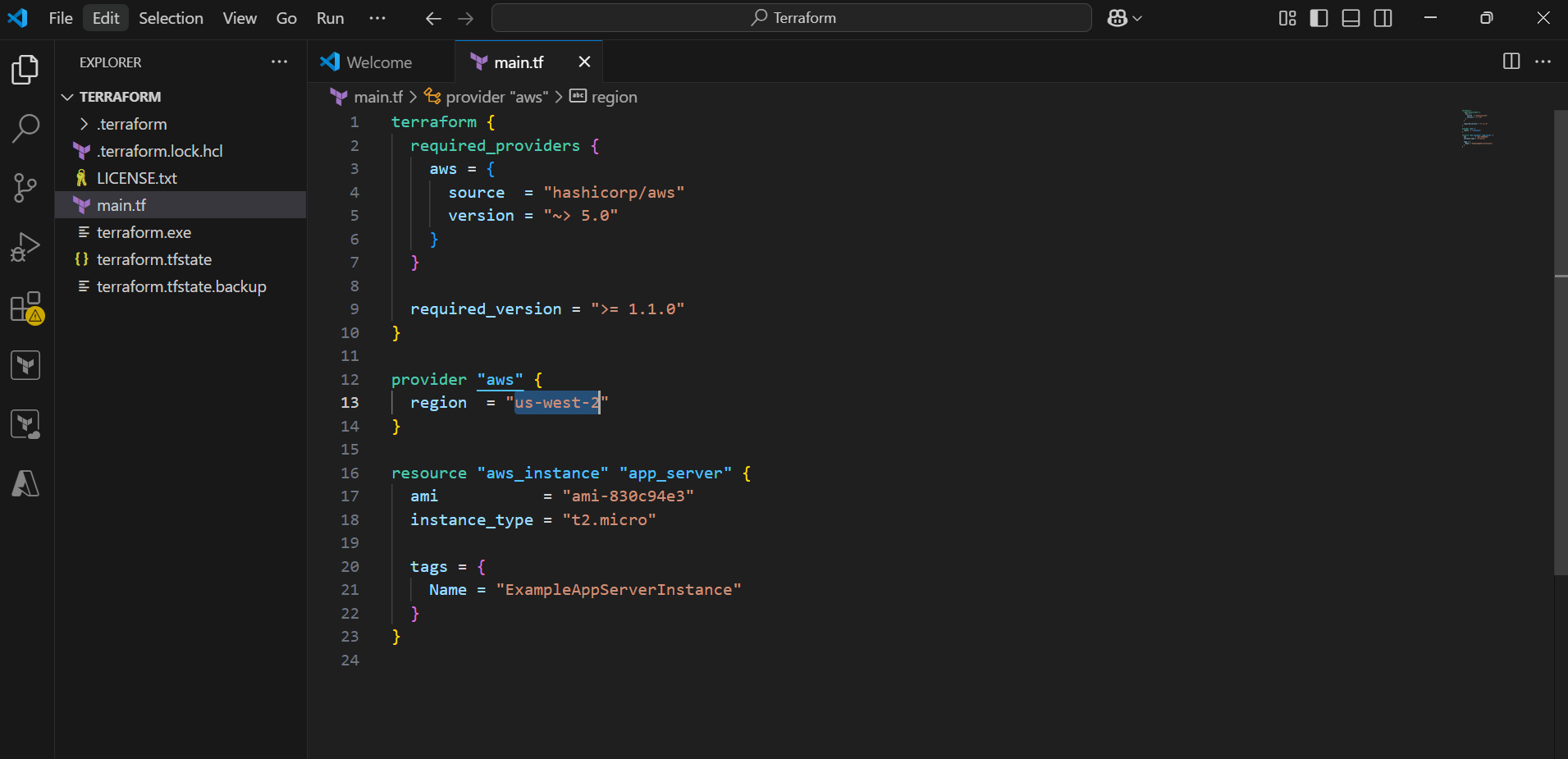
export AWS\_SECRET\_ACCESS\_KEY=

Now in Visual studio code, create a file using terraform once the terraform and terraform CLI are installed.

For installation ,go to [Install Terraform | Terraform | HashiCorp Developer](https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli) guide .

Step1: create main.tf in vs code.

Step2: write this configuration in main.tf file.



terraform {

required\_providers {

aws = {

source = "hashicorp/aws"

version = "~> 4.16"

}

}

required\_version = ">= 1.2.0"

}

provider "aws" {

region = "us-west-2"

}

resource "aws\_instance" "app\_server" {

ami = "ami-830c94e3"

instance\_type = "t2. micro"

tags = {

Name = "ExampleAppServerInstance"

}

}

To run this configuration in AWS console, open terraform CLI and run these commands

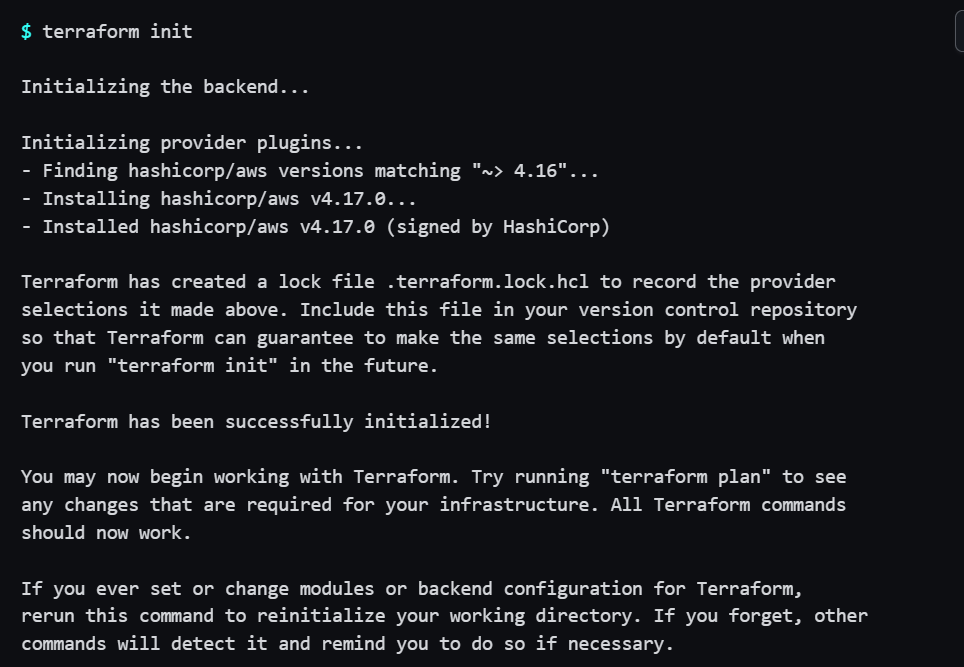
* **terraform init**
* **terraform fmt**
* **terraform plan**
* **terraform apply**
* **terraform destroy**

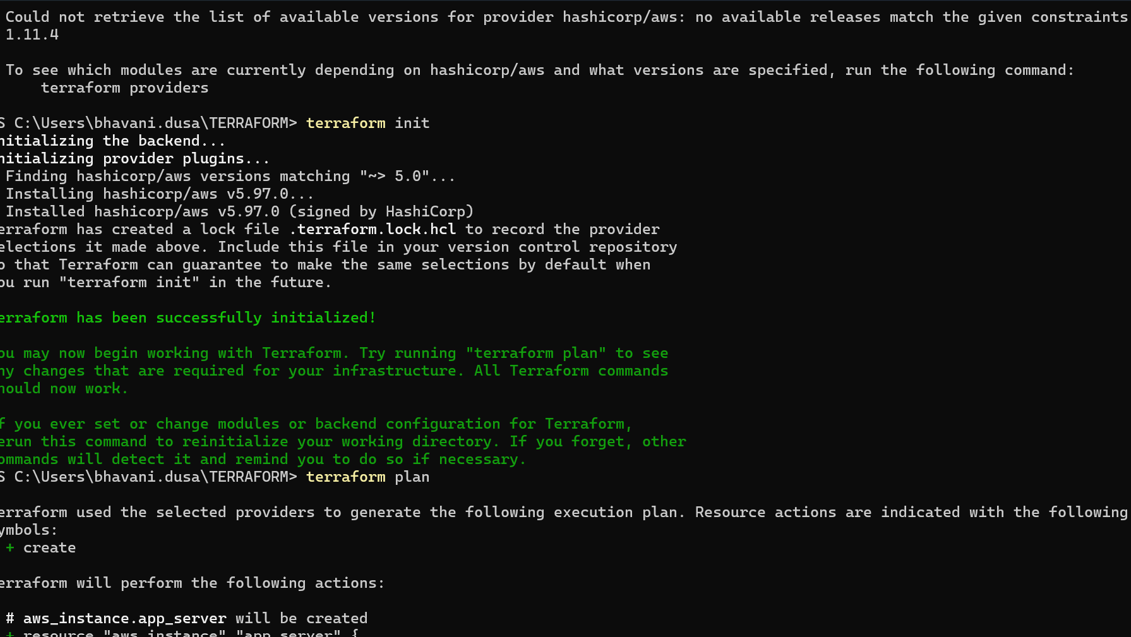
terraform init to initialize the terraform

terraform plan- shows the planning of infra which is about to deploy in cloud

terraform apply will start deploying your infrastructure in Cloud environment

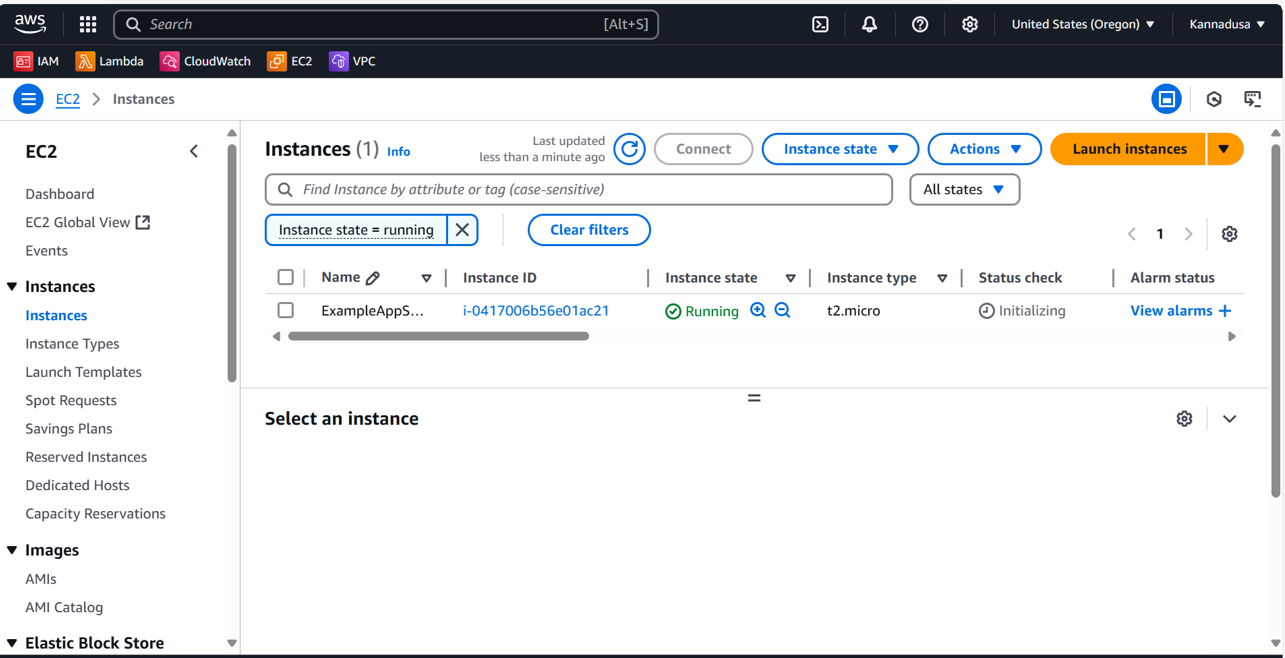
terraform destroy -will destroy infrastructure.





A screenshot of a computer program

AI-generated content may be incorrect.



Then you will the terraform will create the EC2 instance in the given region in terraform file through automation.