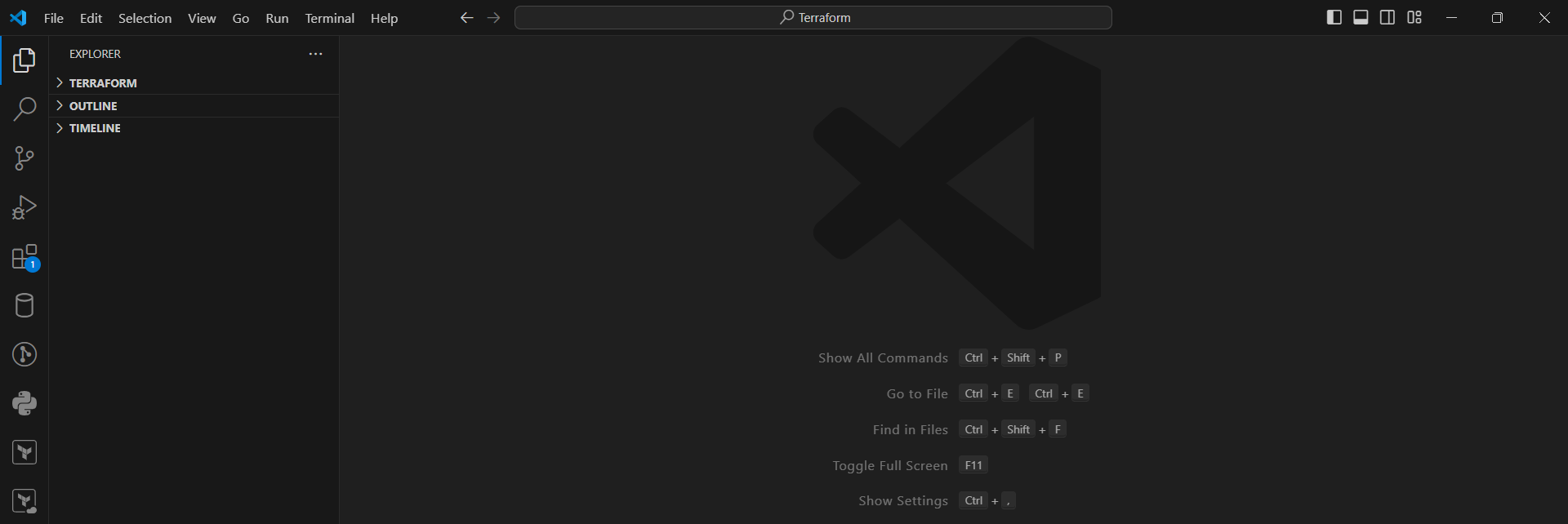
**VPC:** A virtual private cloud (VPC) is a virtual network dedicated to your Alibaba Cloud account. You have full control over your VPC. For example, you can specify the CIDR block and configure route tables and gateways for your VPC. You can also deploy Alibaba Cloud resources, such as Elastic Compute Service (ECS) instances, ApsaraDB RDS instances, and Server Load Balancer (SLB) instances in your VPC.

You can connect your VPC to on-premises networks over Express Connect circuits to create a custom network environment. This lets you easily migrate your applications to the cloud and extend the capabilities of your on-premises environment.

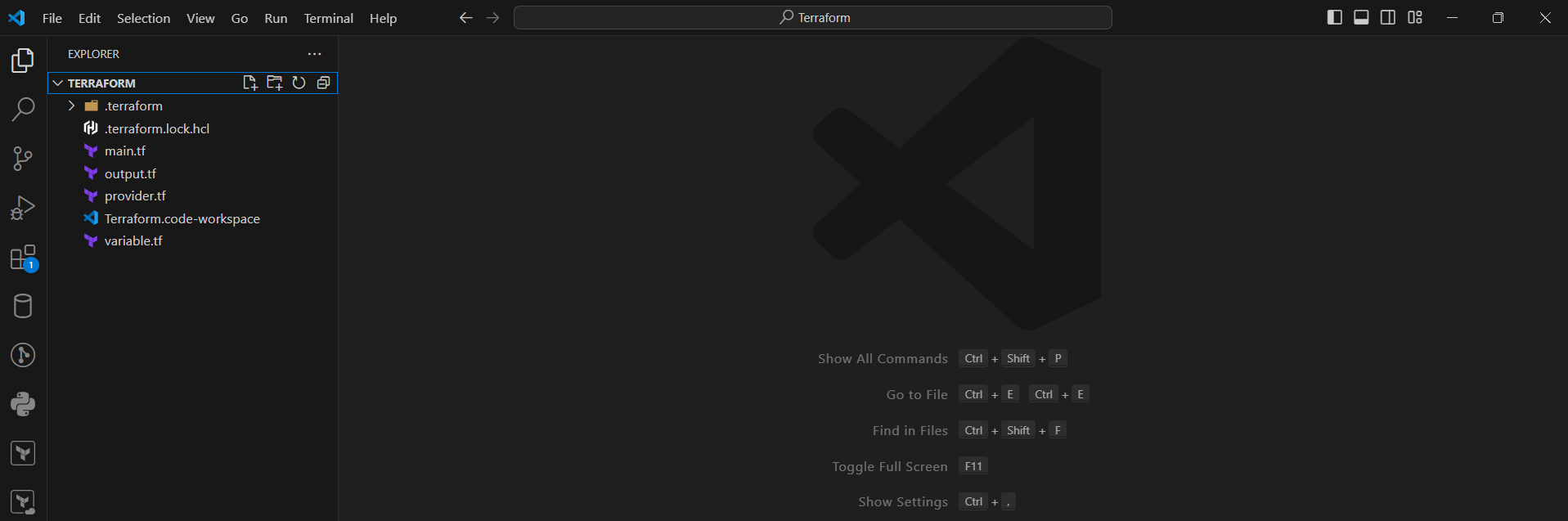
**Terraform:** Define cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

Download Terraform using the link below: <https://www.terraform.io/>

* Create a folder locally and add it to the workspace i.e: Terraform
* Open VS Code, open the folder ‘Terraform’



* Create files in the ‘Terraform’ folder as below:
* ‘main.tf’
* ‘provider.tf’
* ‘output.tf’
* ‘variables.tf’



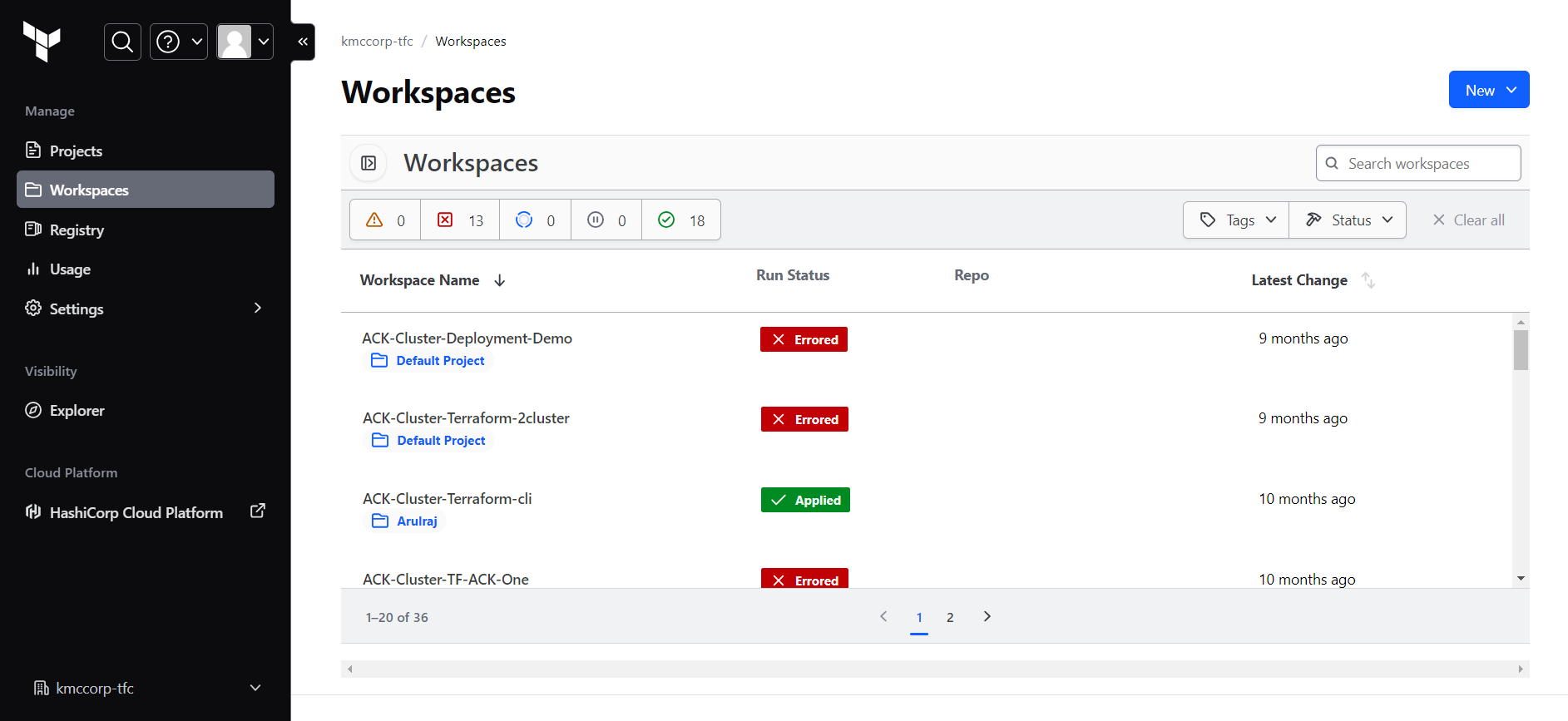
Go to the Terraform Cloud Provider website and get the provider code:

<https://developer.hashicorp.com/terraform/language/providers/configuration?utm_source=tf_registry&utm_content=sidebar>

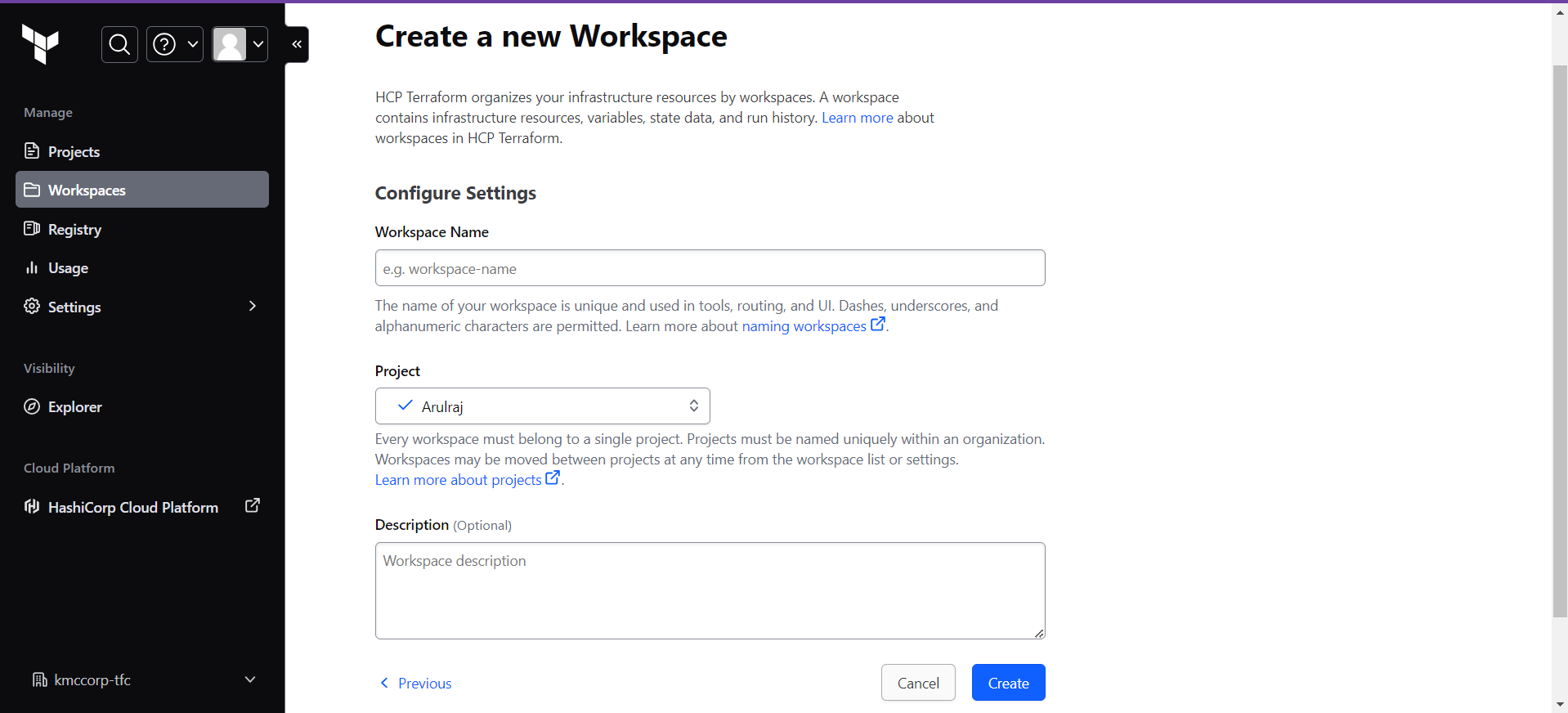
Open the Terraform Cloud using the below link and login with the credentials:

<https://app.terraform.io/app/kmccorp-tfc/workspaces>

* Create a workspace under any selected project:



* There are three different kinds of workflows. They are
* Version Control Workflow: Trigger runs based on changes to configuration in repositories.
* CLI–Driven Workflow: Trigger runs in a workspace using the Terraform CLI.
* API–Driven Workflow: Trigger runs using the Terraform Cloud API.
* We go with CLI-Driven Workflow as we run commands in CLI. Configure the settings as below:
* Workspace name**: chandra-alicoud**



* It provides an example code as below which we copy in the ‘main.tf’ file

terraform {

  cloud {

    organization = "kmccorp-tfc"

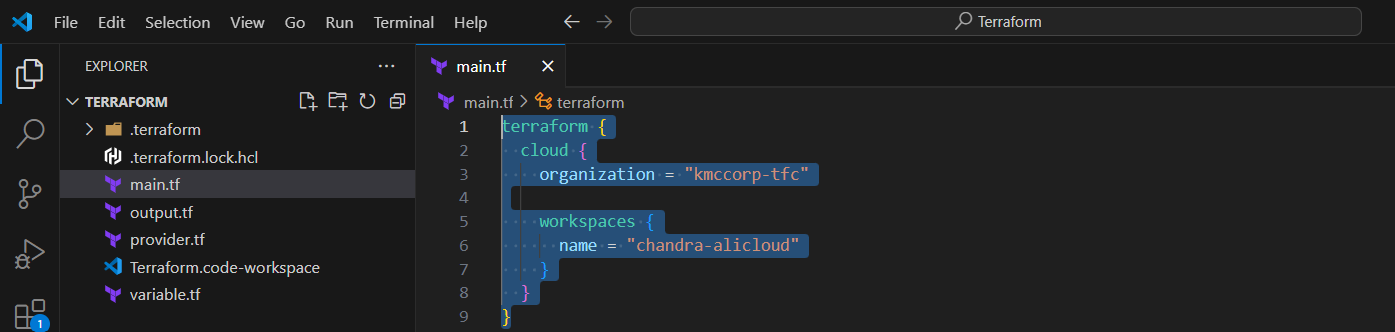
    workspaces {

      name = "chandra-alicloud"

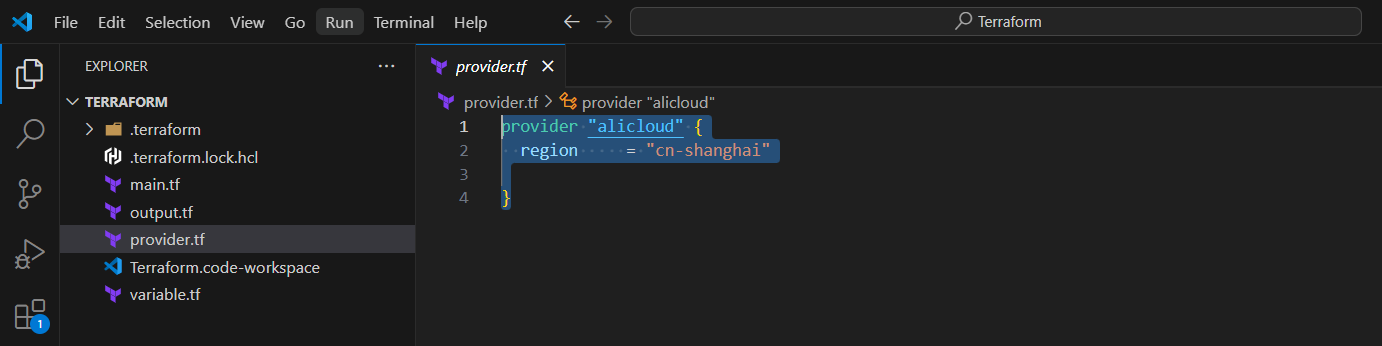
    }

  }

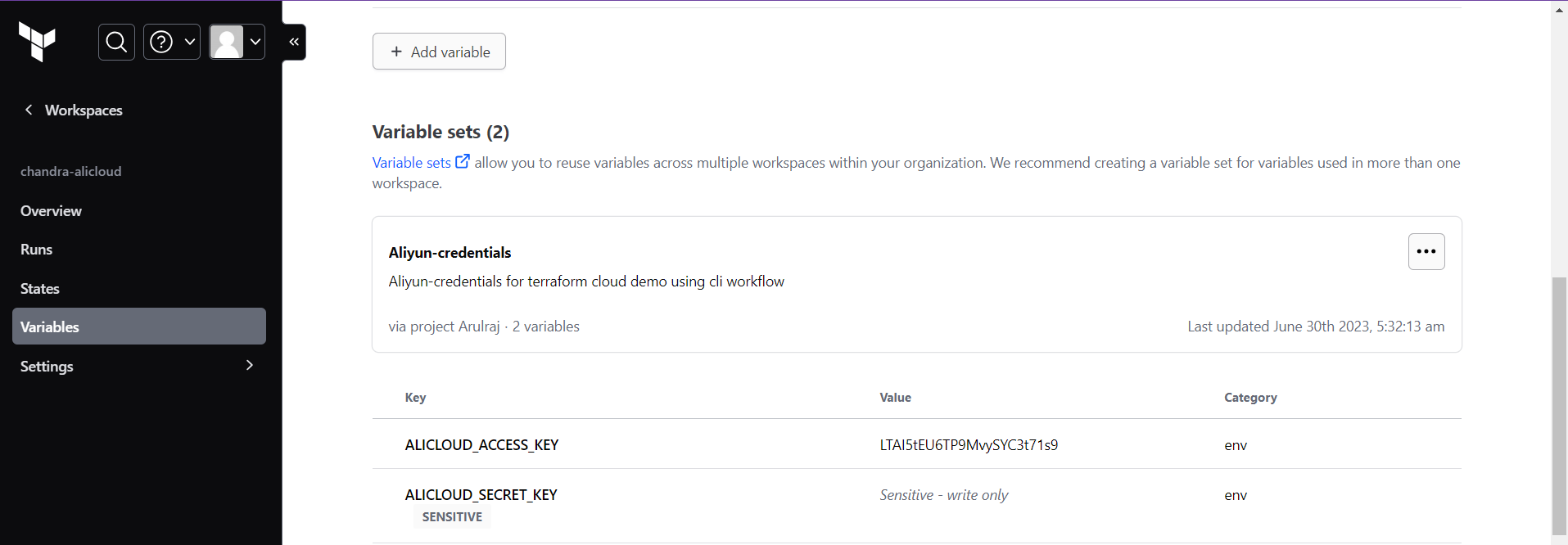
}



* Now we need a provider. We have to find the Alicloud provider from the registry.

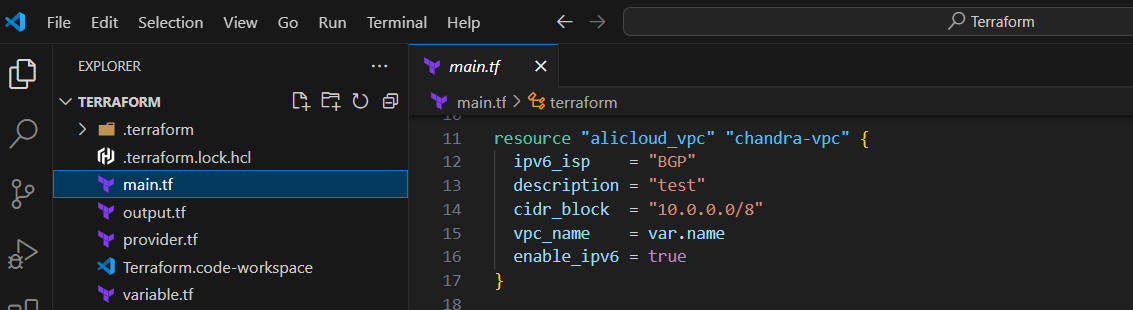


* As we have created the workspace under ARUL’s project we don’t have to explicitly assign ACCESS\_KEY and SECRET\_KEY. It will be automatically assigned to the workspace as it is already under an existing project.

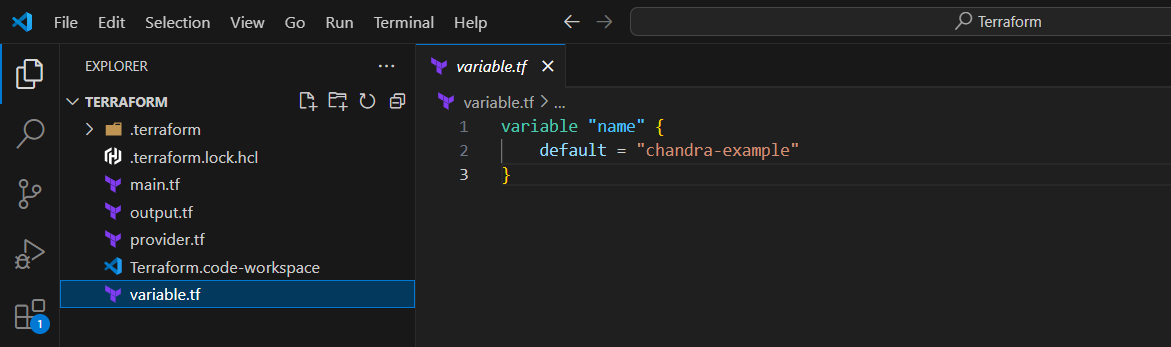


* If we are creating a new project, we have to create an ACCESS\_KEY and SECRET\_KEY explicitly.
* There are two types of workspace variables which are:
* Terraform Variable: These variables should match the description in your configuration. It needs to be accessed with ‘var.’
* Environment Variable: These variables are available in the Terraform runtime environment.
* Now we have to find the Resource. We can access it from the Terraform Alicloud VPC resource website using the link:

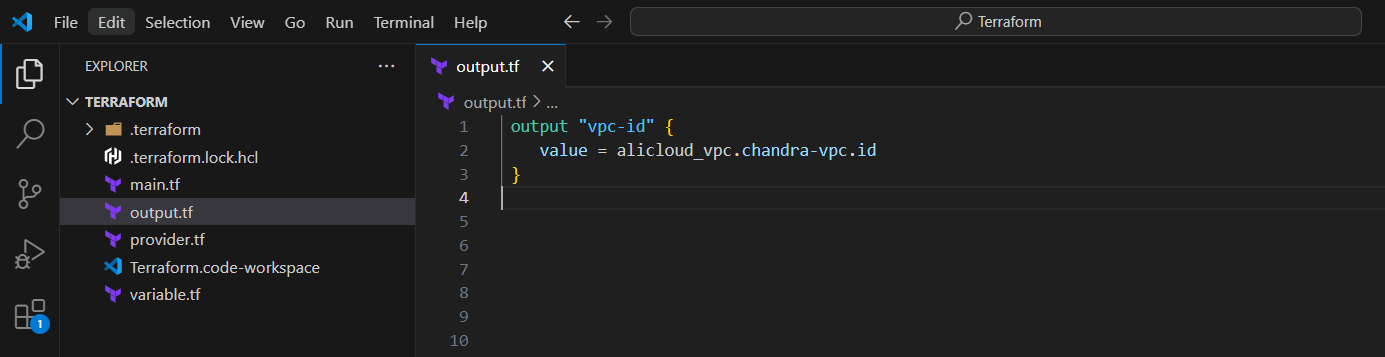
<https://registry.terraform.io/providers/aliyun/alicloud/latest/docs/resources/vpc.html>



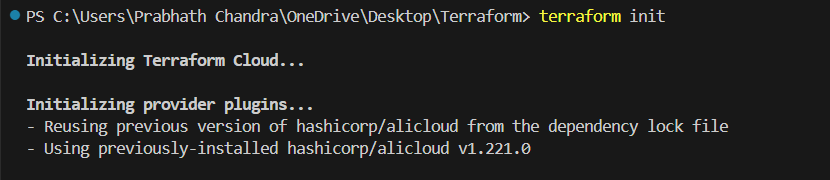
* We have to create a variable now as we don’t have one. And declare the variable.

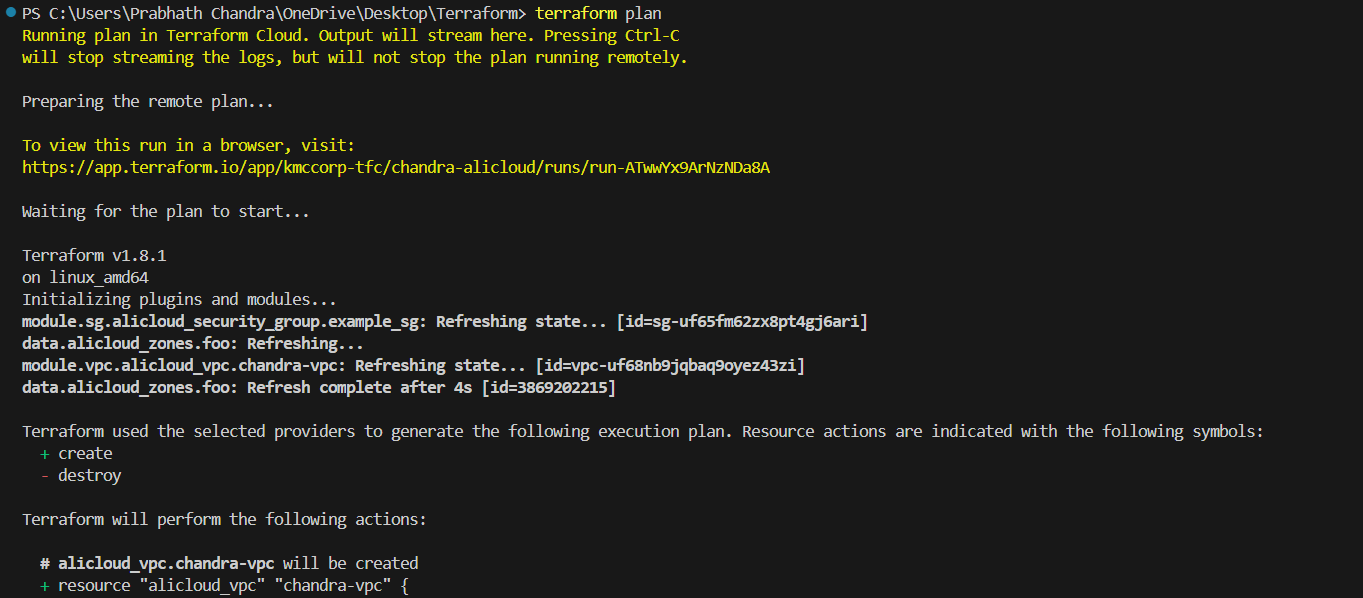


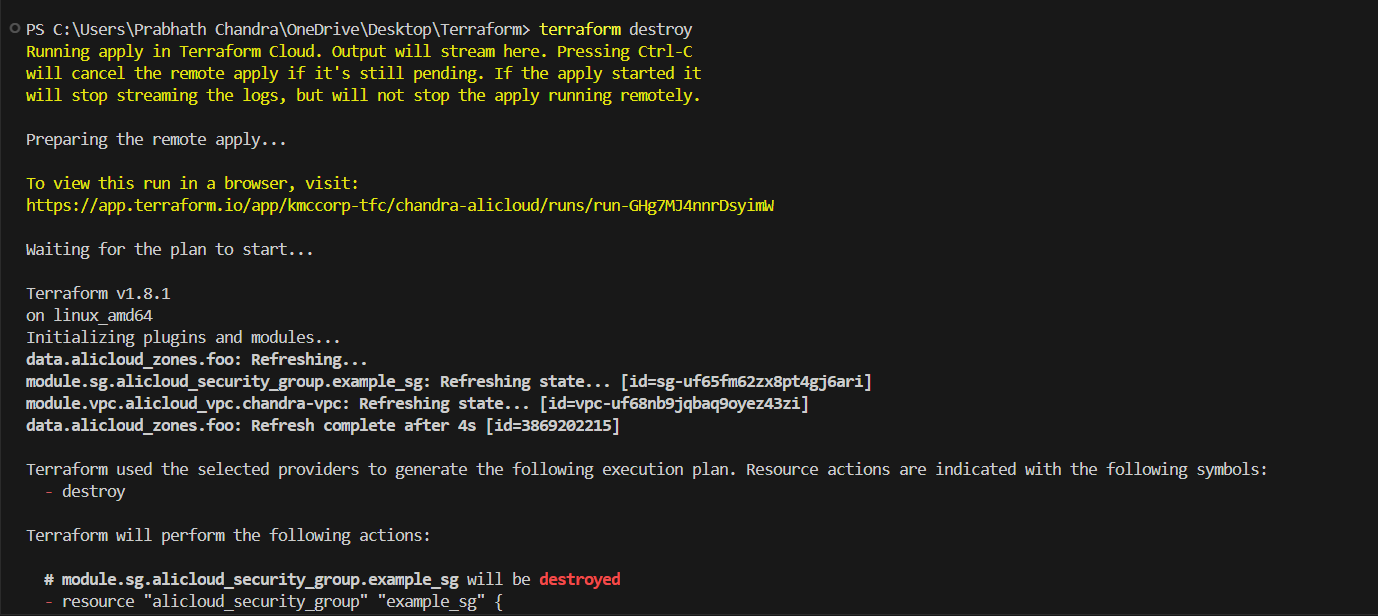
* We have to now create the output.

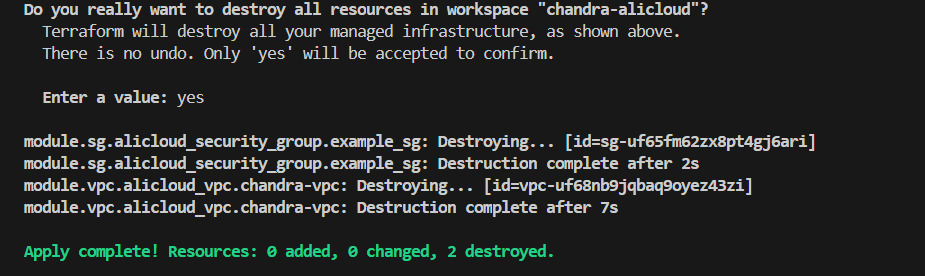


* As we have given all the required values, we now have to run three commands in the terminal which are
* Terraform init: initializes a working directory downloads the necessary provider plugins and modules and sets up the backend for storing your infrastructure's state
* Terraform plan: creates a dry-run, determining what actions are necessary to achieve the desired state defined in the Terraform configuration files
* Terraform apply: applies the changes required to reach the desired state of the configuration, thereby creating, modifying, or deleting the infrastructure resources as necessary.



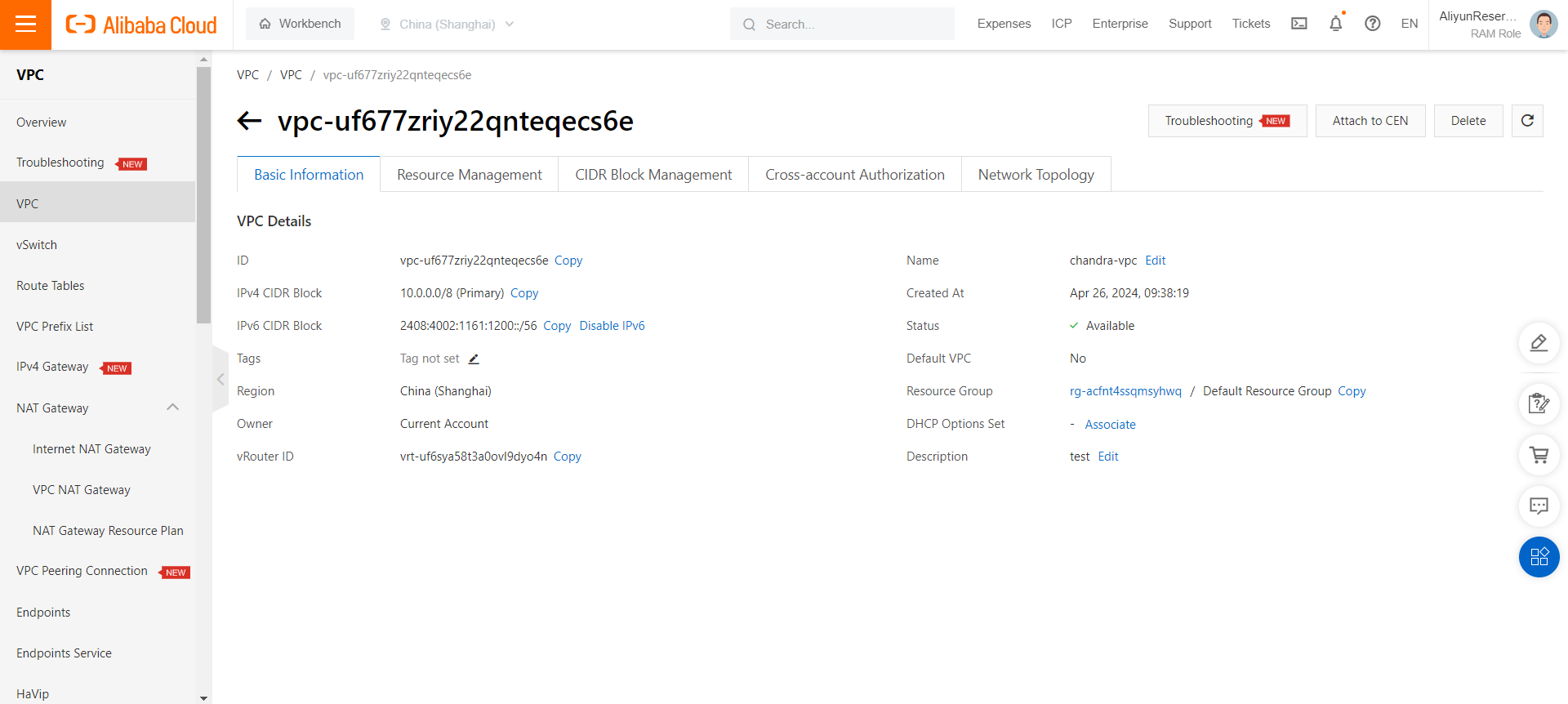






* After the ‘apply’ command to make sure we can go to the Ali cloud console to verify through the VPC ID

https://vpc.console.aliyun.com/vpc/cn-shanghai/vpcs/vpc-uf677zriy22qnteqecs6e?type=base



**CREATING A vSWITCH :**

A vSwitch is the basic networking component within an Alibaba Cloud Virtual Private Cloud (VPC). It's a logically isolated network segment within a VPC.

Subnets: You create subnets within a vSwitch to further organize your cloud resources.

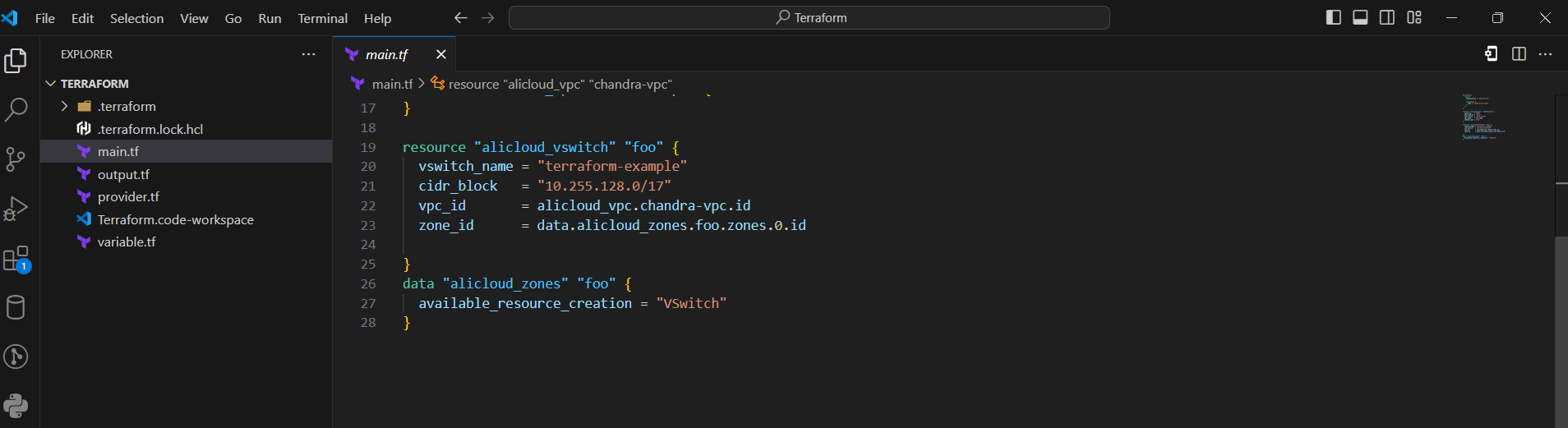
Resource Connectivity: Instances (virtual machines), databases, and other cloud services you launch within a VPC need to reside in a vSwitch to communicate with each other.

Prerequisites:

* An active Alibaba Cloud account.
* A VPC (Virtual Private Cloud) has already been created within your desired region.
* You can use to the existing [vpc module](https://registry.terraform.io/modules/alibaba/vpc/alicloud) to create a VPC and several VSwitches with one click.

Create a Terraform Configuration File:

* Create a file with a .tf extension (e.g., main.tf), In our case, we already have created a ‘main.tf’ file during the VPC creation
* Add the following basic Terraform configuration inside the file:
* Here in the below code, we have used the already existing data source.



Replace Placeholders:

* Replace "your\_access\_key" and "your\_secret\_key" with your actual Alibaba Cloud API credentials.
* Replace "your\_vpc\_id" with the ID of the VPC in which you want to create the vSwitch. You can find this in the Alibaba Cloud console.

Initialize Terraform:

* Open a terminal in the directory with your configuration file.
* Run terraform init to download the Alicloud provider plugin.

Plan Changes:

* Run terraform plan to preview the changes Terraform will make.

Apply Changes:

* Run terraform apply to create the vSwitch. Type yes when prompted to confirm.

After the successful creation of vSwitch, you can check it with the assigned Vpc-id on the Alibaba Cloud Console > VPC > Resource Management

