**Terraform with AWS**

**Install Terraform**

You can install Terraform from the official HashiCorp website: <https://developer.hashicorp.com/terraform/tutorials/aws-get-started/install-cli>or [https://developer.hashicorp.com/terraform/install?product\_intent=terraform#windows](https://developer.hashicorp.com/terraform/install?product_intent=terraform%23windows)

**Setup Environment Variable**

After installing Terraform, set up your environment variable.

**Install AWS CLI**

Install AWS CLI from the official AWS website: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

**Confirm Installation**

Verify your installation by running the following commands:

* terraform --version
* aws --version

**Setup AWS IAM User**

After installing AWS CLI, set up an AWS IAM user with the required permissions. Copy your AWS access key ID and AWS secret access key.

**Configure AWS**

Run the following command to configure AWS:

* **aws configure** :- This command will ask for your AWS access key ID, AWS secret access key, and AWS region.

**Create Terraform File**

Create a directory for your Terraform file using the following command:

* **mkdir terraform\_setup**

Create a file called **main.tf** using the following command**:**

* **touch main.tf**

Open **main.tf** in your text editor and paste the following configuration:



**TERRAFORM BLOCK:-**

Terraform {} this block contains terraform settings, including the required providers Terraform will use to provision your infrastructure. For each provider, the source attribute defines an optional hostname, a namespace, and the provider type. Terraform installs providers from the Terraform Registry by default.

We can also set a version constraint for each provider defined in the required\_providers block. The version attribute is optional, But If we can not specify a provider version, Terraform will automatically download the most recent version during initialization.

**PROVIDERS BLOCK:-**

The provider block configures the specified provider, in this case **aws**. A provider is a plugin that Terraform uses to create and manage your resources.

We can use multiple provider blocks in your Terraform configuration to manage resources from different providers. We can even use different providers together. For example, we could pass the IP address of your AWS EC2 instance to a monitoring resource from DataDog.

**RESOURCES BLOCK:-**

Use resource blocks to define components of your infrastructure. A resource might be a physical or virtual component such as an EC2 instance, or it can be a logical resource.

Resource blocks have two strings before the block: the resource type and the resource name. In this example, the resource type is aws\_instance and the name is app\_server.

**Terraform Commands**

Here are the Terraform commands to run your Terraform file:

**Build**

* terraform init: Initialize the Terraform directory.
* terraform fmt: Format your configuration.
* terraform validate: Validate your configuration.

**Create Infrastructure**

* terraform plan: Show your infrastructure in detail.
* terraform apply: Apply your Terraform file to create your infrastructure.

**Inspect Infrastructure State**

* terraform show: Show your infrastructure state.
* terraform state list: List the resources in your project's state.