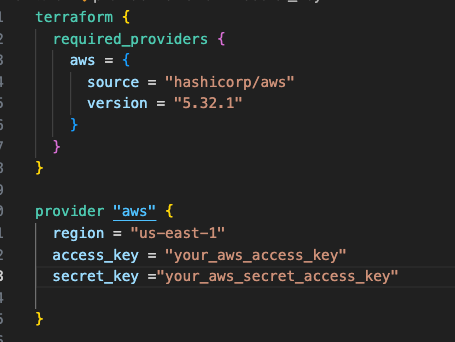
**What is Terraform and how to connect Terraform to your AWS account**

n the world of computers and cloud infrastructure, Terraform is used by developers and system administrators to create and manage their digital landscapes & infrastructure (servers, networks, databases) without the hassle of doing everything manually. It’s a powerful tool that helps make managing and scaling complex systems much more manageable and, dare I say, magical! Terraform is similar to Lego blocks, where pieces are used to create whatever you want in the digital space from the bottom up.

Now let’s dig into how to connect Terraform to your AWS account. After you finish writing your infrastructure code (provisioning resources and other necessary blocks) Terraform will need to communicate with AWS to make these resources. There are 3 different ways you can do this

1. Hardcoding Access Key into your terraform code (not recommended approach) — In your Terraform file, you can simply add your AWS access key and AWS secret access keythat Terraform can use when connecting with AWS. Like the example below under the “Provider” block



This is almost never a good idea. Certainly, opting for this method provides a quick but notably insecure means of connecting to AWS. Anybody looking at your code will be able to view the access key to your AWS account. This approach is particularly detrimental when your Terraform file is committed to version control systems like git, posing a significant security risk.

A much safer approach is to use Environmental Variables

2. Using environmental variables — You can create environmental variables for your access key and secret access key. Environmental variables are similar to sticky notes that are not written directly into your code but information that your code will have access to. Terraform CLI is designed to detect the presence of environmental variables like your access key. You can create the environmental variables in your current terminal session by executing the following command -

export AWS\_ACCESS\_KEY\_ID=your\_aws\_access\_key  
export AWS\_SECRET\_ACCESS\_KEY=your\_aws\_secret\_access\_key

Now you can delete your hard coded access key information from your terraform file (as shows in the first method) and terraform will automatically detect the environmental variable to connect to AWS.

3. The last approach is to store your credentials in a file in your local system. Terraform provider can be configured to use this file using the shared\_credential\_file attribute:



Just make sure not to commit this file in version control so other people can see your credentials.