**Class Notes - Day 2**

**Terraform Documentation Links**

• Terraform Official Documentation: https://developer.hashicorp.com/terraform

• Terraform Registry: https://registry.terraform.io/

**Understanding How Terraform Communicates with AWS**

**How does Terraform send requests to AWS?**

Terraform uses providers to interact with cloud platforms. For AWS, it uses the aws provider plugin which defines how Terraform communicates with the AWS API.

**How does Terraform authenticate requests?**

Terraform authenticates with AWS using credentials configured in your environment. This can be done through:  
- aws configure command (sets access key, secret key, region, etc.)  
- Environment variables (e.g., AWS\_ACCESS\_KEY\_ID, AWS\_SECRET\_ACCESS\_KEY)  
- IAM role (for EC2 or other AWS services)

**How do we log in to our AWS account from Terraform?**

Use the following command to configure credentials:  
aws configure  
Then provide your Access Key, Secret Key, and default Region.

**Terraform Workflow**

1. terraform init  
 - Downloads required provider plugins (e.g., AWS)  
 - Initializes the working directory  
 - Creates .terraform folder and terraform.lock.hcl file  
  
2. terraform plan  
 - Shows the execution plan: what Terraform will add, change, or destroy  
 - Generates terraform.tfstate file to store infrastructure state and metadata  
  
3. terraform apply  
 - Applies the proposed changes to your cloud infrastructure

**Terraform Installation**

1. Follow the official installation guide: https://developer.hashicorp.com/terraform/downloads  
2. Install HashiCorp Terraform extension in VS Code for enhanced support

**Practical Exercise**

**Step 1: Create Terraform Files**

Inside your working directory, create:  
- provider.tf — to define the AWS provider and region  
- main.tf — to define the actual AWS resources

**Step 2: Configure AWS CLI**

Run:  
aws configure  
Provide the AWS credentials and default region. The EC2 AMI will be fetched based on the region configured here.

**Step 3: Define Region in provider.tf**

To specify a region explicitly:  
  
provider "aws" {  
 region = "us-east-1"  
}

**Terraform Files Explained**

- .terraform/: Directory created by terraform init, holds provider plugins  
- terraform.lock.hcl: Lock file that records the exact provider version  
- terraform.tfstate: Tracks the current state of your infrastructure

**Provider Versioning**

Terraform automatically fetches the latest provider version unless specified.  
To pin a specific version in provider.tf:  
  
provider "aws" {  
 region = "us-east-1"  
 version = "~> 5.69.0"  
}  
  
If you want to change the version, manually update provider.tf and run:  
terraform init -upgrade  
  
This updates the provider and modifies the lock file.

Important Notes:  
- If provider.tf is missing the version, Terraform uses the version specified in terraform.lock.hcl  
- To downgrade from a newer version (e.g., 6.2.0 to 5.69.0), update provider.tf and reinitialize

**Best Practices**

- After changing provider.tf, always run terraform init again  
- Add unnecessary/auto-generated files (e.g., .terraform/, terraform.lock.hcl) to .gitignore at the root level  
- Keep provider.tf and main.tf separate for better organization