Terraform State Remote Configuration

HashiCorp Terraform

State Backend Config

- To transition from a local state to a remote state, must configure a remote backend
- Remote backends are configured on top level terraform block
- Use the backend keyword, provide the type of backend
  - In example, we are using s3
- Provide configuration to the backend
  - Required configuration values depend on the type of backend being used
  - In example of using S3, we provide values for bucket name, the key to the state file, and region where bucket is created

```
terraform {
  backend "s3" {
    bucket = "mybucket"
    key    = "path/to/my/key"
    region = "us-east-1"
  }
}
```

State Backend Config

⚠ A Note on AWS Authentication ⚠

- For several configurations, such as the Provider and state Backend, we have to authenticate to AWS
  - Providers authenticate to AWS for making API calls to read and create infrastructure
  - States authenticate to AWS for reading, and writing into remote state from S3
- There are many different ways to authenticate, but three are most common:
  - Profiles - directly specify which AWS profile to use
  - Assumed Roles - directly specify which AWS role to assume
  - Provide the AWS Access Key & AWS Secret Key as environment variables
    - Terraform will automatically look for these environment variables
    - DO NOT DO THIS!
    - Not safe for production, not dependable, generally using exposed access & secret keys are a no no
- Recommendation:
  - Use Assumed Roles when running Terraform in automated workflow
  - Use Profiles for local work

State Backend Config

```
provider "aws" {
  region     = "us-west-2"
  access_key = "my-access-key"
  secret_key = "my-secret-key"
}
```

```
backend "s3" {
  bucket  = "udemy-terraform-bootcamp"
  key     = "terraform_state_course/terraf
  region  = "us-west-2"
  profile = "terraform_bootcamp"
}
```

HashiCorp Terraform

State Backend Config

- Before transitioning to remote state, S3 bucket must be created beforehand
  - Generally okay to manually create this bucket
  - Good practice to enable versioning and server-side encryption
  - Make sure the bucket is private
- Possible to create S3 bucket in Terraform with local state - then migrate statefile to remote state
- Whenever backend is first configured or changed
  - Must re-run terraform init
  - Why? Look back on notes and try to figure it out