Step-01: Introduction

- Understand Terraform Backends
- Understand about Remote State Storage and its advantages
- This state is stored by default in a local file named `terraform.tfstate` , but it can also be stored remotely, which works better in a team environment.
- Create Azure Storage Account to store `terraform.tfstate` file and enable backend configurations in terraform settings block

Step-02: Create Azure Storage Account

Step-02-01: Create Resource Group

- Go to Resource Groups -> Add
- **Resource Group:** terraform-storage-rg
- **Region:** East US
- Click on **Review + Create**
- Click on **Create**

Step-02-02: Create Azure Storage Account

- Go to Storage Accounts -> Add
- **Resource Group:** terraform-storage-rg
- **Storage Account Name:** terraformstate201 (THIS NAME SHOULD BE UNIQUE ACROSS AZURE CLOUD)
- **Region:** East US
- **Performance:** Standard
- **Redundancy:** Geo-Redundant Storage (GRS)
- In `Data Protection` , check the option `Enable versioning for blobs`
- REST ALL leave to defaults
- Click on **Review + Create**
- Click on **Create**

Step-02-03: Create Container in Azure Storage Account

- Go to Storage Account -> `terraformstate201` -> Containers -> `+Container`
- **Name:** tfstatefiles
- **Public Access Level:** Private (no anonymous access)
- Click on **Create**

Step-03: Terraform Backend Configuration

- **Reference Sub-folder:** terraform-manifests
- Terraform Backend as Azure Storage Account
- Add the below listed Terraform backend block in `Terrafrom Settings` block in `c1-versions.tf`

```
# Terraform State Storage to Azure Storage Container
  backend "azurerm" {
    resource_group_name   = "terraform-storage-rg"
    storage_account_name  = "terraformstate201"
    container_name        = "tfstatefiles"
    key                   = "terraform.tfstate"
  }
```

Step-04: Review Terraform Configuration Files

1. c1-versions.tf
2. c2-variables.tf
3. c3-locals.tf
4. c4-resource-group.tf
5. c5-virtual-network.tf
6. c6-linux-virtual-machine.tf
7. c7-outputs.tf
8. terraform.tfvars

Step-05: Test with Remote State Storage Backend

```
# Initialize Terraform
terraform init
Observation:
1. Review below message
2. Verify the Azure Storage Account and you should see terraform.tfstate file created
## Sample CLI Output
Initializing the backend...
Successfully configured the backend "azurerm"! Terraform will automatically
use this backend unless the backend configuration changes.

# Validate Terraform configuration files
terraform validate

# Review the terraform plan
terraform plan
Observation:
1. Acquiring state lock. This may take a few moments...

# Create Resources
terraform apply -auto-approve

# Verify Azure Storage Account for terraform.tfstate file
Observation:
1. Finally at this point you should see the terraform.tfstate file in Azure Storage Account.

# Access Application
http://<Public-IP>
```

Step-05: Storage Account Container Versioning Test

- Update in `c3-locals.tf`
- Uncomment Demo tag

```
common_tags = {
  Service = local.service_name
  Owner   = local.owner
  Tag = "demo-tag1"  # Uncomment during step-05
}
```

- Execute Terraform Commands

```
# Review the terraform plan
terraform plan

# Create Resources
terraform apply -auto-approve
```

```
# Verify terraform.tfstate file in Azure Storage Account
Observation:
1. New version of terraform.tfstate file will be created
2. Understand about Terraform State Locking
3. terraform.tfsate file should be in "leased" state which means no one can apply changes using terraform to Azure Reso
4. Once the changes are completed "terraform apply", Lease State should be in "Available" state.
```

Step-06: Destroy Resources

- Destroy Resources and Verify Storage Account `terraform.tfsate` file Versioning

```
# Destroy Resources
terraform destroy -auto-approve

# Delete Files
rm -rf .terraform*

# c3-locals.tf - Comment demo tag  seamless demo
  common_tags = {
    Service = local.service_name
    Owner   = local.owner
    #Tag = "demo-tag1"
  }
```

References

- Terraform Backends
- Terraform State Storage
- Terraform State Locking
- Remote Backends - Enhanced