Agenda and Training Details

1 July 2025

Course Content – Day 1 & 2

Terraform Fundamentals & AWS Provisioning Basics

- 🧠 **Concepts**
- Introduction to DevOps & IaC
- What is Terraform and how it works?
- Declarative vs Imperative IaC tools
- Terraform CLI, workflow, providers, and state

Course Content – Day 1 & 2

Terraform Fundamentals & AWS Provisioning Basics

- 🔨 **Hands-On**
- Install Terraform CLI
- Setup AWS CLI & credentials
- Create first Terraform project (S3 Bucket or EC2 instance)
- Use AWS provider
- Write and apply basic .tf files
- Understand Terraform plan/apply/destroy
- Understand resource blocks and meta arguments (count, depends_on)
- Cleanup with destroy

Course Content – Day 3 & 4

Variables, Outputs, Modules, and State Management

- 🧠 **Concepts**
- Input variables, local variables, outputs
- Data sources (e.g., AMI lookup, EC2 tags)
- Remote state overview and state file best practices
- Terraform backends (S3 + DynamoDB for locking)
- Reusability with modules

Course Content – Day 3 & 4

Variables, Outputs, Modules, and State Management

- 🔨 **Hands-On**
- Use variables with tfvars
- Output key resource values (e.g., public IP of EC2)
- Use terraform_remote_state
- Configure S3 as a remote backend (with DynamoDB locking)
- Break project into modules (VPC, EC2, security group)
- Reference output from modules
- Simple module registry usage

Course Content – Day 5 & 6

AWS Resource Management, Monitoring & Update Automation

- 🧠 **Concepts**
- Infrastructure updates (patches, package upgrades)
- Basic monitoring with CloudWatch via Terraform
- AWS EC2 lifecycle (stop, update, reboot, terminate)
- Terraform provisioners (basic usage)
- Templating with templatefile function
- Managing dependencies

Course Content – Day 5 & 6

AWS Resource Management, Monitoring & Update Automation

- 🔨 **Hands-On**
- Create a complete 3-tier architecture (VPC + EC2 + RDS)
- Install and update packages via remote-exec provisioner
- Use user_data to automate package updates (Amazon Linux)
- Enable CloudWatch monitoring using aws_cloudwatch_log_group, metric_alarm
- Write alert for high CPU on EC2
- Use depends_on and ignore_changes effectively

Course Content – Day 7 & 8

Best Practices, Collaboration, Testing & Final Project

- 🧠 **Concepts**
- Best practices for production-like Terraform code
- Using Terraform with Git (version control, remote modules)
- Secure variables using environment variables
- Team collaboration tips: workspaces, state isolation
- Testing basics: validate, terraform fmt, plan check
- Overview of CI/CD integration

Course Content – Day 7 & 8

Best Practices, Collaboration, Testing & Final Project

- 🔨 **Hands-On**
- Use terraform fmt, validate, and plan in Git workflow
- Create separate workspaces (dev, stage, prod)
- Final project:
  - Provision a VPC + public/private subnets
  - Launch EC2 instance with updates enabled
  - Configure monitoring & alarms
  - Reuse modules
  - Use terraform.tfvars and output key values

Attendance

During the training

- At any time during the training, please interrupt if:
  - My speed is slow or fast
  - Any topic is to be repeated again
  - You feel a need of a break

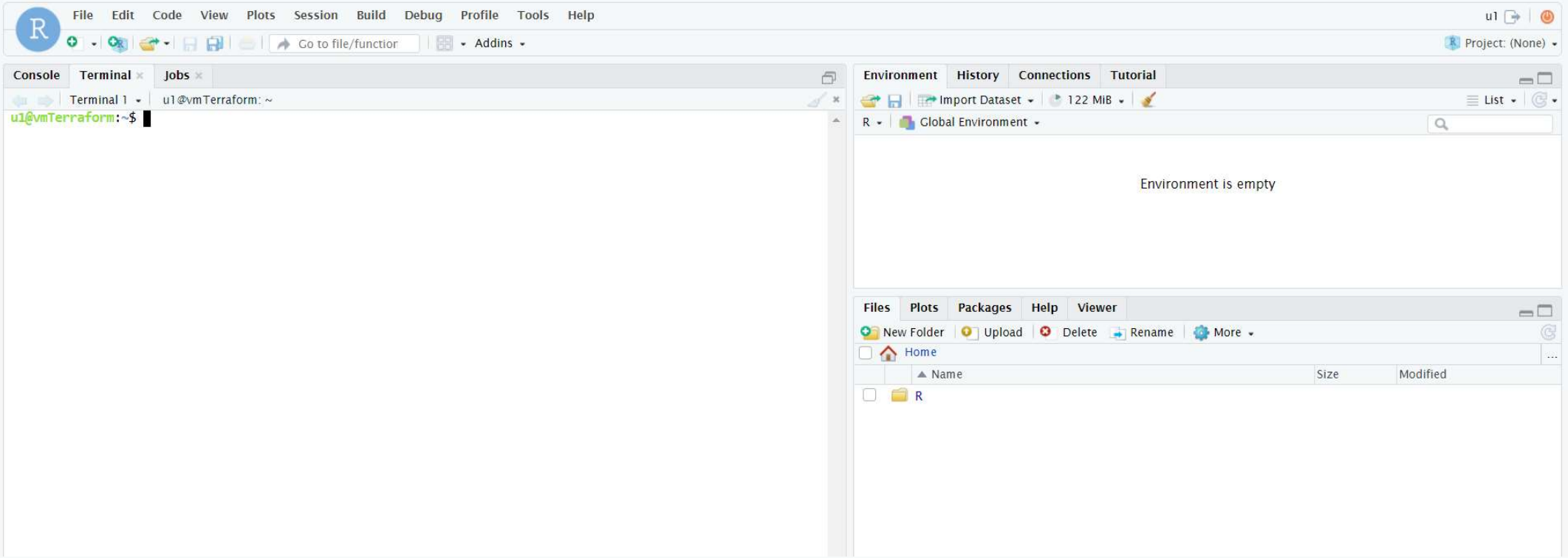
Tea Break Time

- ?

Lab Environment

- AWS Cloud Access
  - Refer in Chat
- In Browser Console
  - http://ec2-35-159-175-24.eu-central-1.compute.amazonaws.com:8787/

Lab Environment



Thank You