Terraform Cloud Automation

This playbook describes the sequential steps required to define and provision cloud infrastructure using HashiCorp's Terraform. It guides through setting up Terraform, writing the configuration, initializing and applying the configuration to automate cloud deployments.

Step 1: **Installation**

Install Terraform by downloading the appropriate package for your operating system from the Terraform website and follow the installation instructions.

Step 2: **Configuration**

Define your cloud infrastructure in configuration files with an extension `.tf`. Use Terraform's declarative language to specify your infrastructure as code.

Step 3: **Initialization**

Initialize your Terraform workspace using the command `terraform init`, which will install the necessary plugins and prepare your configuration for execution.

Step 4: **Planning**

Run `terraform plan` to create an execution plan. Terraform will determine what actions are necessary to achieve the desired state specified in the configuration files.

Step 5: **Applying**

Execute the plan by running `terraform apply`. Terraform will provision the infrastructure as described, giving you the opportunity to review the plan before confirming the action.

Step 6: **Monitoring**

After applying, you should monitor the state of your infrastructure. Terraform provides commands like `terraform state` to help manage and understand the current state.

Step 7: **Updating**

To update your infrastructure, modify your configuration files and repeat the planning and applying steps to realize the changes.

Step 8: **Destruction**

When you no longer need the infrastructure, you can remove it using `terraform destroy`, which will terminate the resources defined in your configuration files.

General Notes

Version Control

It's recommended to use version control to manage your Terraform configuration files. This helps in tracking changes and collaborating with others.

Sensitive Data

Be cautious with sensitive data in your configurations. Use variables and secrets management to protect credentials and other sensitive information.

Modules

Terraform modules can be used to create reusable components for your infrastructure, helping to keep your configurations DRY (Don’t Repeat Yourself).

State Management

Understand Terraform state management, especially when working in a team. Remote state backends like Terraform Cloud can help with state locking and sharing.