

Terraform: A Beginner's Guide – Summary

1. What is Infrastructure as Code (IaC)?

IaC allows managing and provisioning computing infrastructure through machine-readable definition files instead of manual hardware configuration.

2. What is Terraform?

Terraform is an open-source IaC tool by HashiCorp to define cloud and on-premises resources using a declarative language.

3. Installing Terraform

- Windows: Download binary, add to PATH, verify with ``terraform -v``
- macOS: Install via Homebrew (``brew install terraform``), verify with ``terraform -v``
- Linux: Install via package manager, verify with ``terraform -v``

4. Creating a Terraform Project

- Create a project directory
- Create ``main.tf`` file
- Define provider and resources

5. Basic Terraform Commands

- ``terraform init``: Initialize directory
- ``terraform plan``: Show execution plan
- ``terraform apply``: Apply changes
- ``terraform show``: Show current state
- ``terraform destroy``: Destroy infrastructure

6. Working with Variables and Outputs

- Variables: Define in ``variables.tf``, referenced in ``main.tf``
- Outputs: Define in ``outputs.tf`` to display infrastructure info

7. Modules

Modules group resources for reuse and better organization.

8. Managing Environments

- Folders: Organize configuration files
- Workspaces: Manage multiple versions in a single configuration

9. Terraform Project Structure

- ``main.tf``: Main configuration
- ``variables.tf``: Variables
- ``outputs.tf``: Outputs
- ``terraform.tfvars``: Variable values

10. AWS Examples (Short)

- ****Create IAM User****:

```
resource "aws_iam_user" "user1" {  
  name = "user1"  
}
```

- ****Create EC2 Instance****:

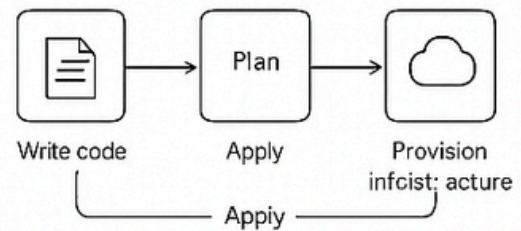
```
resource "aws_instance" "example" {  
  ami          = "ami-12345678"  
  instance_type = "t2.micro"  
}
```

Terraform: A Beginner's Guide – Summary



1 What is Infrastructure as Code (IAC)?

IaC is a practice that allows you to manage and provision computing infrastructure through machine readable definition files, rather than physical hardware configuration or interactive configuration tools.



3 Installing Terraform

Windows:

- Download the Terraform binary from the official website.
- Add the binary to your PATH
- Verify installation by running `terraform -v` in command prompt.

macOS:

- Install Homebrew if not already installed.
- Run `brew install terraform` to install `terraform -v` in the terminal

Linux:

- Download the appropriate package for your distribution.
- Install terraform using your package manager by running `terraform -v` in terminal in terminal

4 Creating a Terraform Project

Create a new directory for your project. Inside a file named `main.tf`. Define your provider and resources in `main.tf` file.

5 Basic Terraform Commands

- `terraform init`
- `terraform plan` creates an `execution plan` in `main.tfplan` file
- Resource definitions for an EC2 instance.



6 Working with Variables and Outputs

Variables

- Define variables to make information about your infrastructure after applying the configuration

Outputs

- Define outputs to display information about your infrastructure after applying the configuration
- Outputs can be defined in `outputs.tf`.

7 Modules

Modules, containers for multiple resources that are used together. Using allows to Organize configuration and reuse code.

8 AWS: Creating IAM and EC2 Instances



- Define an AWS provider and create a IAM user in `main.tf`.
- Add resource definitions for an EC2 instance.

9 Terraform Project Structure

A typical Terraform project structure includes:

- `main.tf` the main configuration
- File `variables.tf`
- `outputs.tf`
- `terraform.tfvars` for variable values