HashiCorp Terraform

aws

Terraform Commands

iamrishmish Rishabh Mishra Rishabh Mishra DevOps i_m_rishmish Rishmish94

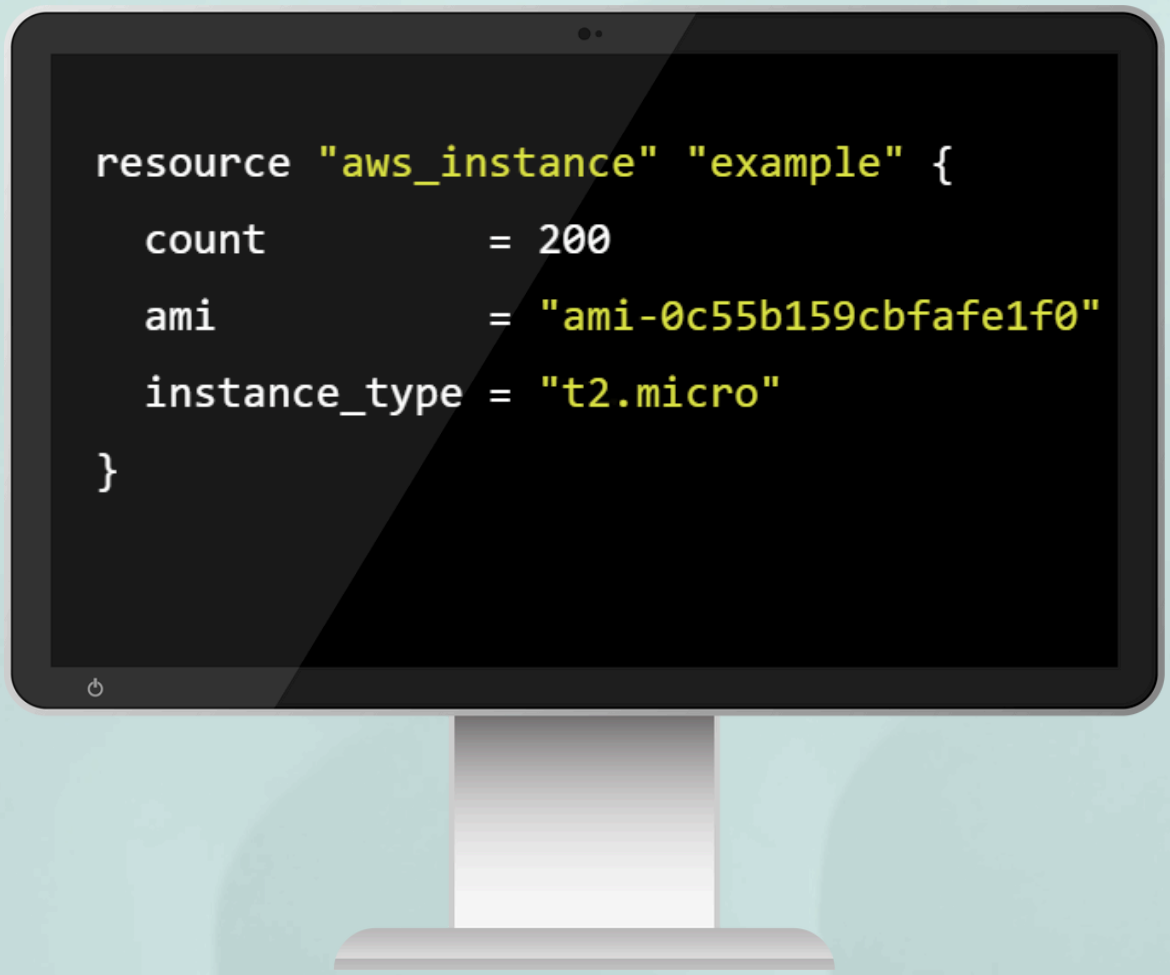
Terraform

Terraform is one of the most popular Infrastructure as Code (IaC) tools that helps automate cloud infrastructure deployment. But to use it effectively, you need to know its essential commands.

1. Setting Up Terraform
2. Planning & Applying Changes
3. Destroying Infrastructure
4. Managing Terraform State
5. Working with Modules & Outputs

iamrishmish Rishabh Mishra Rishabh Mishra DevOps i_m_rishmish Rishmish94

Before diving into commands, let's look at a basic Terraform configuration:



```
resource "aws_instance" "example" {
  count         = 200
  ami           = "ami-0c55b159cbfafe1f0"
  instance_type = "t2.micro"
}
```

This code tells Terraform to create an AWS EC2 instance using a specific AMI and instance type.

terraform init

– Initialize a Terraform Project

This command sets up a new or existing Terraform project. It does the following:

- **Downloads necessary provider plugins (like AWS, Azure, Google Cloud, etc.).**
- **Configures the backend where Terraform stores its state file.**
- **Prepares Terraform to manage infrastructure.**

```
touch main.tf  # Create a new Terraform file
terraform init  # Initialize Terraform in the directory
```

After running terraform init, Terraform will download the AWS provider plugin needed to communicate with AWS.

terraform fmt

– Format Terraform Code

Terraform code should be clean and readable.

This command automatically formats the code properly.

terraform init
terraform init

```
terraform fmt  # Reformats all Terraform files in the directory
```

Running terraform fmt will correct any indentation or spacing issues in the configuration file.

terraform validate

- Check for Errors

Before applying changes, it's good practice to validate your Terraform code.

This command checks for syntax errors or missing configurations.

terraform init
terraform init

```
terraform validate  # Checks for syntax errors
```

If there are syntax errors in the configuration, Terraform will highlight them before applying changes.

iamrishmish Rishabh Mishra Rishabh Mishra DevOps i_m_rishmish Rishmish94

terraform plan

– Preview Changes

Before making changes to your infrastructure, you should preview what Terraform is about to do.
This command shows what will be added, changed, or destroyed without applying anything.

```
terraform plan  # Shows what Terraform will do
```

Terraform will output a plan detailing that it will create one EC2 instance.

terraform apply

- Apply Changes

Once you're happy with the plan, use this command to actually create the infrastructure.

```
terraform apply  # Deploys the infrastructure
```

terraform init

Terraform provisions the EC2 instance in AWS.

```
terraform apply -auto-approve  # Deploys without asking for confirmation
```

You will be asked to confirm before applying. If you want to skip the confirmation step, use the above code

terraform destroy

- Delete Everything

Sometimes, you need to delete all resources created by Terraform. This command does that.

```
terraform destroy  # Deletes all infrastructure managed by Terraform
```

The EC2 instance created earlier will be deleted.

```
terraform  destroy  -target=aws_instance.example
# Deletes a specific EC2 instance
```

Destroy a Specific Resource: If you want to delete only one resource instead of everything, use the above code

terraform show

- View Current State

This command displays the details of your Terraform-managed infrastructure.

terraform init
terraform init

```
terraform show  # Shows all existing resources
```

Terraform will show the EC2 instance details, including its public IP and instance ID.

terraform state list

- List All Resources

This command shows all resources Terraform is currently managing.

terraform init
terraform init

```
terraform state list  # Lists all managed resources
```

Terraform will list the managed EC2 instance.

iamrishmish Rishabh Mishra Rishabh Mishra DevOps i_m_rishmish Rishmish94

terraform state rm <resource>

- Remove a Resource from State

If you need Terraform to forget about a resource without deleting it from the cloud, use this command.

terraform init
terraform init

```
terraform state rm aws_instance.example
 # Removes EC2 from Terraform state
```

Terraform will stop managing the EC2 instance, but the instance will still exist in AWS.

terraform get

- Download Modules

Modules in Terraform help you reuse configurations. This command downloads the required modules before applying changes.

terraform init
terraform init

```
terraform get # Fetches modules used in the configuration
```

terraform output

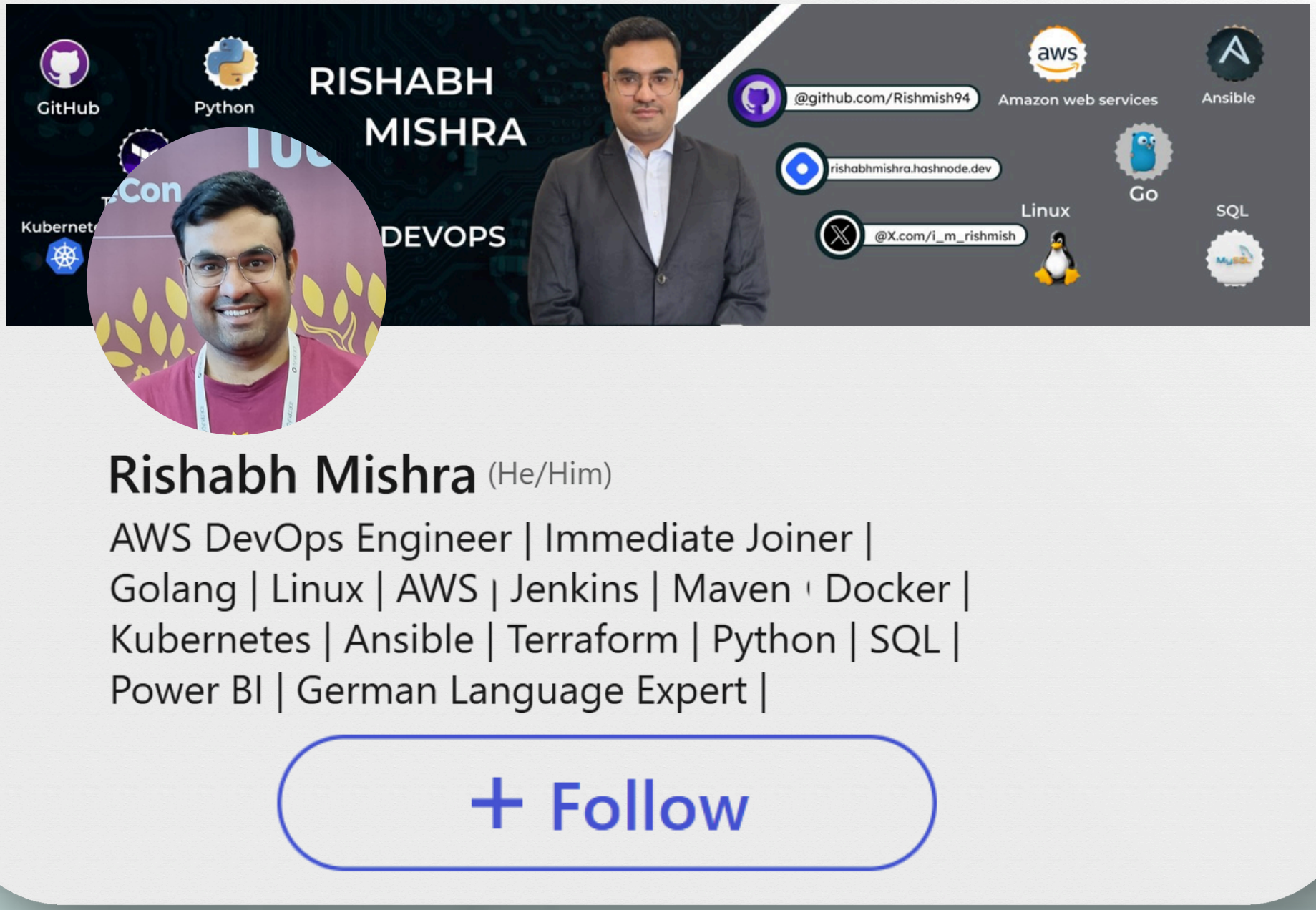
- View Output Values

Terraform allows defining output values in configuration files to capture important information, such as resource attributes.

```
terraform output instance_ip # Displays the public IP of the instance
```

If an output variable is defined for the EC2 instance's public IP, running terraform output will display that value.

Found this helpful? Repost to help others learn Terraform!



Follow me for more DevOps insights, cloud automation tips, and hands-on tutorials.

iamrishmish | Rishabh Mishra | Rishabh Mishra DevOps | i_m_rishmish | Rishmish94