


Destroy Infrastructure

 developer.hashicorp.com/terraform/tutorials/certification-associate-tutorials/aws-destroy

You have now created and updated an EC2 instance on AWS with Terraform. In this tutorial, you will use Terraform to destroy this infrastructure.

Once you no longer need infrastructure, you may want to destroy it to reduce your security exposure and costs. For example, you may remove a production environment from service, or manage short-lived environments like build or testing systems. In addition to building and modifying infrastructure, Terraform can destroy or recreate the infrastructure it manages.

Destroy

The `terraform destroy` command terminates resources managed by your Terraform project. This command is the inverse of `terraform apply` in that it terminates all the resources specified in your Terraform state. It does *not* destroy resources running elsewhere that are not managed by the current Terraform project.

Destroy the resources you created.

```
$ terraform destroy
Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
```

```
- destroy
```

Terraform will perform the following actions:

```
# aws_instance.app_server will be destroyed
- resource "aws_instance" "app_server" {
  - ami                      = "ami-08d70e59c07c61a3a" -> null
  - arn                      = "arn:aws:ec2:us-west-
2:561656980159:instance/i-0fd4a35969bd21710" -> null
##...
```

Plan: 0 to add, 0 to change, 1 to destroy.

```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.
```

```
Enter a value:
```

The `-` prefix indicates that the instance will be destroyed. As with `apply`, Terraform shows its execution plan and waits for approval before making any changes.

Answer `yes` to execute this plan and destroy the infrastructure.

Enter a value: yes

```
aws_instance.app_server: Destroying... [id=i-0fd4a35969bd21710]
aws_instance.app_server: Still destroying... [id=i-0fd4a35969bd21710, 10s elapsed]
aws_instance.app_server: Still destroying... [id=i-0fd4a35969bd21710, 20s elapsed]
aws_instance.app_server: Still destroying... [id=i-0fd4a35969bd21710, 30s elapsed]
aws_instance.app_server: Destruction complete after 31s
```

Destroy complete! Resources: 1 destroyed.

Just like with **apply**, Terraform determines the order to destroy your resources. In this case, Terraform identified a single instance with no other dependencies, so it destroyed the instance. In more complicated cases with multiple resources, Terraform will destroy them in a suitable order to respect dependencies.