

```
web_instance_ip = "3.110.195.193"
```

```
[root@node1 youtube]#
```

```
[root@node1 youtube]# ^C
```

```
[root@node1 youtube]# cd
```

```
[root@node1 ~]# vim auto.tf
```

```
[root@node1 ~]# terraform init
```

Initializing the backend...

Initializing provider plugins...

- Finding latest version of hashicorp/aws...

- Installing hashicorp/aws v3.68.0...

- Installed hashicorp/aws v3.68.0 (signed by HashiCorp)

Terraform has created a lock file `.terraform.lock.hcl` to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see any changes that are required for your infrastructure. All Terraform commands should now work.

If you ever set or change modules or backend configuration for Terraform, rerun this command to reinitialize your working directory. If you forget, other commands will detect it and remind you to do so if necessary.

```
[root@node1 ~]# terraform plan
```

```
█
```

```
}  
}  
  
# aws_launch_template.auto_scale will be created  
+ resource "aws_launch_template" "auto_scale" {  
  + arn                = (known after apply)  
  + default_version    = (known after apply)  
  + id                 = (known after apply)  
  + image_id           = "ami-0002bdad91f793433"  
  + instance_type      = "t2.micro"  
  + latest_version     = (known after apply)  
  + name               = (known after apply)  
  + name_prefix        = "auto_scale"  
  + tags_all           = (known after apply)  
  
  + metadata_options {  
    + http_endpoint      = (known after apply)  
    + http_protocol_ipv6 = (known after apply)  
    + http_put_response_hop_limit = (known after apply)  
    + http_tokens        = (known after apply)  
  }  
}
```

Plan: 2 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

[[[[root@node1 ~]#

```
# aws_launch_template.auto_scale will be created
+ resource "aws_launch_template" "auto_scale" {
  + arn                = (known after apply)
  + default_version    = (known after apply)
  + id                 = (known after apply)
  + image_id           = "ami-0002bdad91f793433"
  + instance_type      = "t2.micro"
  + latest_version     = (known after apply)
  + name               = (known after apply)
  + name_prefix        = "auto_scale"
  + tags_all           = (known after apply)

  + metadata_options {
    + http_endpoint           = (known after apply)
    + http_protocol_ipv6     = (known after apply)
    + http_put_response_hop_limit = (known after apply)
    + http_tokens             = (known after apply)
  }
}
```

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes




```
Activities  PuTTY SSH Client  Dec 3 11:56  root@node1:~

+ tags_all          = (known after apply)

+ metadata_options {
    + http_endpoint      = (known after apply)
    + http_protocol_ipv6 = (known after apply)
    + http_put_response_hop_limit = (known after apply)
    + http_tokens        = (known after apply)
  }
}

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: yes

aws_launch_template.auto_scale: Creating...
aws_launch_template.auto_scale: Creation complete after 1s [id=lt-0f7935f820a6ea4a3]
aws_autoscaling_group.bar: Creating...
aws_autoscaling_group.bar: Still creating... [10s elapsed]
aws_autoscaling_group.bar: Still creating... [20s elapsed]
aws_autoscaling_group.bar: Still creating... [30s elapsed]
aws_autoscaling_group.bar: Still creating... [40s elapsed]
aws_autoscaling_group.bar: Still creating... [50s elapsed]
aws_autoscaling_group.bar: Creation complete after 57s [id=terraform-20211203062534282800000003]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
[root@node1 ~]#
```

Activities

Google Chrome

Dec 3 11:58

raheela2000

EC2 Management Console

(1) WhatsApp

Terraform/auto.tf at mas

+

←

→

↻

ap-south-1.console.aws.amazon.com/ec2autoscaling/home?region=ap-south-1#/details

→

aws

Services

Search for services, features, blogs, docs, and more

[Alt+S]

🔔

🔔

🔔

Mumbai

New EC2 Experience

Tell us what you think

×

EC2 Dashboard

Events

Tags

Limits

INSTANCES

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

IMAGES

AMIs

The old Auto Scaling groups console is no longer available. We will keep improving the new console based on your feedback.

Auto Scale your Amazon EC2 Instances Ahead of Demand

Explore how the new predictive scaling policy of EC2 Auto Scaling helps you improve availability for your applications.

Learn More

EC2

>

Auto Scaling groups

Auto Scaling groups (1)

🔄

Edit

Delete

Create an Auto Scaling group

🔍 Search your Auto Scaling groups

<

1

>

⚙️

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	terraform-2021120306253428	auto_scale20211203062534402000000	1	-	1	1	4	ap-south-1a