

MINGW64:/c/Users/Nitin.S/terraform-ec2-nginx/terraform-aws-nginx

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ terraform -v
Terraform v1.10.3
on windows_386
+ provider registry.terraform.io/hashicorp/aws v5.84.0
```

Your version of Terraform is out of date! The latest version is 1.10.5. You can update by downloading from <https://www.terraform.io/downloads.html>

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ ^C
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ ^C
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ terraform -v
Terraform v1.10.3
on windows_386
+ provider registry.terraform.io/hashicorp/aws v5.84.0
```

Your version of Terraform is out of date! The latest version is 1.10.5. You can update by downloading from <https://www.terraform.io/downloads.html>

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ aws configure
AWS Access Key ID [*****BDMW]: AKIAWFIPSULO6M36XSNO
AWS Secret Access Key [*****dav4]: mVC0EuF1ZkL1FOYoBmLtJbnjWewWsMmoYH
zQyHAI
Default region name [us-east-1]: us-east-1
Default output format [json]: json
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ mkdir terraform-aws-nginx
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ cd terraform-aws-nginx
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ pwd
/c/Users/Nitin.S/terraform-ec2-nginx/terraform-aws-nginx
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ ^C
```

```
MINGW64:/c/Users/Nitin.S/terraform-ec2-nginx/terraform-aws-nginx
AWS Secret Access Key [*****daV4]: mVC0EuF1ZkL1F0YoBmLtJbnjWewWsMmoYH
zQyHAI
Default region name [us-east-1]: us-east-1
Default output format [json]: json
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ mkdir terraform-aws-nginx
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx
$ cd terraform-aws-nginx
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ pwd
/c/Users/Nitin.S/terraform-ec2-nginx/terraform-aws-nginx
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ ^C
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ terraform init
Terraform initialized in an empty directory!
```

The directory has no Terraform configuration files. You may begin working with Terraform immediately by creating Terraform configuration files.

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ dir
main.tf.txt  outputs.tf.txt  provider.tf.txt  user_date.sh.txt  variable.tf.txt
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ mv main.tf.txt main.tf
mv outputs.tf.txt outputs.tf
mv provider.tf.txt provider.tf
mv user_date.sh.txt user_data.sh
mv variable.tf.txt variables.tf
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ rename main.tf.txt main.tf
rename outputs.tf.txt outputs.tf
rename provider.tf.txt provider.tf
rename user_date.sh.txt user_data.sh
rename variable.tf.txt variables.tf
bash: rename: command not found
bash: rename: command not found
bash: rename: command not found
bash: rename: command not found
bash: rename: command not found
```

MINGW64:/c/Users/Nitin.S/terraform-ec2-nginx/terraform-aws-nginx
guarantee to take exactly these actions if you run "terraform apply" now.

Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
\$ terraform apply -auto-approve

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
# aws_instance.nginx_us_east_1 will be created
+ resource "aws_instance" "nginx_us_east_1" {
  + ami                        = "ami-0c55b159cbfafa1f0"
  + arn                      = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone         = (known after apply)
  + cpu_core_count            = (known after apply)
  + cpu_threads_per_core      = (known after apply)
  + disable_api_stop          = (known after apply)
  + disable_api_termination   = (known after apply)
  + ebs_optimized              = (known after apply)
  + enable_primary_ipv6       = (known after apply)
  + get_password_data          = false
  + host_id                   = (known after apply)
  + host_resource_group_arn    = (known after apply)
  + iam_instance_profile       = (known after apply)
  + id                        = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle         = (known after apply)
  + instance_state             = (known after apply)
  + instance_type              = "t2.micro"
  + ipv6_address_count         = (known after apply)
  + ipv6_addresses             = (known after apply)
  + key_name                   = "my-key"
  + monitoring                 = (known after apply)
  + outpost_arn                = (known after apply)
  + password_data              = (known after apply)
  + placement_group            = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns                = (known after apply)
  + private_ip                 = (known after apply)
  + public_dns                 = (known after apply)
  + public_ip                  = (known after apply)
  + secondary_private_ips      = (known after apply)
  + security_groups            = (known after apply)
  + source_dest_check          = true
```


MINGW64:/c/Users/Nitin.S/terraform-ec2-nginx/terraform-aws-nginx
guarantee to take exactly these actions if you run "terraform apply" now.

Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
\$ terraform apply

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```
# aws_instance.nginx_us_east_1 will be created
+ resource "aws_instance" "nginx_us_east_1" {
  + ami                        = "ami-014d544cfef21b42d"
  + arn                       = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone          = (known after apply)
  + cpu_core_count             = (known after apply)
  + cpu_threads_per_core       = (known after apply)
  + disable_api_stop           = (known after apply)
  + disable_api_termination    = (known after apply)
  + ebs_optimized              = (known after apply)
  + enable_primary_ipv6        = (known after apply)
  + get_password_data          = false
  + host_id                    = (known after apply)
  + host_resource_group_arn     = (known after apply)
  + iam_instance_profile        = (known after apply)
  + id                         = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle          = (known after apply)
  + instance_state              = (known after apply)
  + instance_type               = "t2.micro"
  + ipv6_address_count          = (known after apply)
  + ipv6_addresses              = (known after apply)
  + key_name                    = (known after apply)
  + monitoring                  = (known after apply)
  + outpost_arn                 = (known after apply)
  + password_data               = (known after apply)
  + placement_group             = (known after apply)
  + placement_partition_number  = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns                 = (known after apply)
  + private_ip                  = (known after apply)
  + public_dns                  = (known after apply)
  + public_ip                   = (known after apply)
  + secondary_private_ips       = (known after apply)
  + security_groups             = (known after apply)
  + source_dest_check           = true
```

```
+ root_block_device (known after apply)
}

# aws_instance.nginx_us_west_1 will be created
+ resource "aws_instance" "nginx_us_west_1" {
+   ami                    = "ami-0abcdef1234567890"
+   arn                   = (known after apply)
+   associate_public_ip_address = (known after apply)
+   availability_zone      = (known after apply)
+   cpu_core_count         = (known after apply)
+   cpu_threads_per_core   = (known after apply)
+   disable_api_stop       = (known after apply)
+   disable_api_termination = (known after apply)
+   ebs_optimized          = (known after apply)
+   enable_primary_ipv6    = (known after apply)
+   get_password_data      = false
+   host_id                = (known after apply)
+   host_resource_group_arn = (known after apply)
+   iam_instance_profile   = (known after apply)
+   id                     = (known after apply)
+   instance_initiated_shutdown_behavior = (known after apply)
+   instance_lifecycle     = (known after apply)
+   instance_state         = (known after apply)
+   instance_type          = "t2.micro"
+   ipv6_address_count     = (known after apply)
+   ipv6_addresses         = (known after apply)
+   key_name               = (known after apply)
+   monitoring             = (known after apply)
+   outpost_arn            = (known after apply)
+   password_data          = (known after apply)
+   placement_group        = (known after apply)
+   placement_partition_number = (known after apply)
+   primary_network_interface_id = (known after apply)
+   private_dns            = (known after apply)
+   private_ip             = (known after apply)
+   public_dns             = (known after apply)
+   public_ip              = (known after apply)
+   secondary_private_ips  = (known after apply)
+   security_groups        = (known after apply)
+   source_dest_check       = true
+   spot_instance_request_id = (known after apply)
+   subnet_id              = (known after apply)
+   tags                   = {
+     + "Name" = "Nginx-US-West-1"
+   }
+   tags_all = {
+     + "Name" = "Nginx-US-West-1"
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$
```

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ terraform init
```

Initializing the backend...

Initializing provider plugins...

- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.84.0

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.

```
Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx
$ terraform apply
```

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

+ create

Terraform will perform the following actions:

aws_instance.nginx_us_east_1 will be created

```
+ resource "aws_instance" "nginx_us_east_1" {
+   ami                  = "ami-014d544cfef21b42d"
+   arn                  = (known after apply)
+   associate_public_ip_address = (known after apply)
+   availability_zone     = (known after apply)
+   cpu_core_count        = (known after apply)
+   cpu_threads_per_core  = (known after apply)
+   disable_api_stop      = (known after apply)
+   disable_api_termination = (known after apply)
+   ebs_optimized         = (known after apply)
+   enable_primary_ipv6   = (known after apply)
+   get_password_data     = false
+   host_id               = (known after apply)
+   host_resource_group_arn = (known after apply)
+   iam_instance_profile   = (known after apply)
+   id                    = (known after apply)
+   instance_initiated_shutdown_behavior = (known after apply)
```



```
MINGW64/c/Users/Nitin.S/terraform-ec2-nginx/terraform-aws-nginx
+ capacity_reservation_specification (known after apply)
+ cpu_options (known after apply)
+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
```

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

Changes to Outputs:

```
+ nginx_us_west_1_public_ip = (known after apply)
```

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_instance.nginx_us_west_1: Creating...

aws_instance.nginx_us_west_1: Still creating... [10s elapsed]

aws_instance.nginx_us_west_1: Creation complete after 16s [id=i-0eab070e632fc0c3d]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

nginx_us_east_1_public_ip = "52.91.196.249"

nginx_us_west_1_public_ip = "54.193.58.151"

Nitin.S@DESKTOP-HF82NIK MINGW64 ~/terraform-ec2-nginx/terraform-aws-nginx

\$ |

Instances (1) Info

Last updated 1 minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

< 1 >

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	Nginx-US-East-1	i-0af47674a08b7912f	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-52-9

Select an instance

- Dashboard
- EC2 Global View
- Events
- Instances
 - Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
- Images
 - AMIs

Instances (1) Info Last updated less than a minute ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

[All states](#)

<input type="checkbox"/>	Name ✎	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	Nginx-US-Wes...	i-0eab070e632fc0c3d	Running	t2.micro	2/2 checks passed	View alarms +

Select an instance