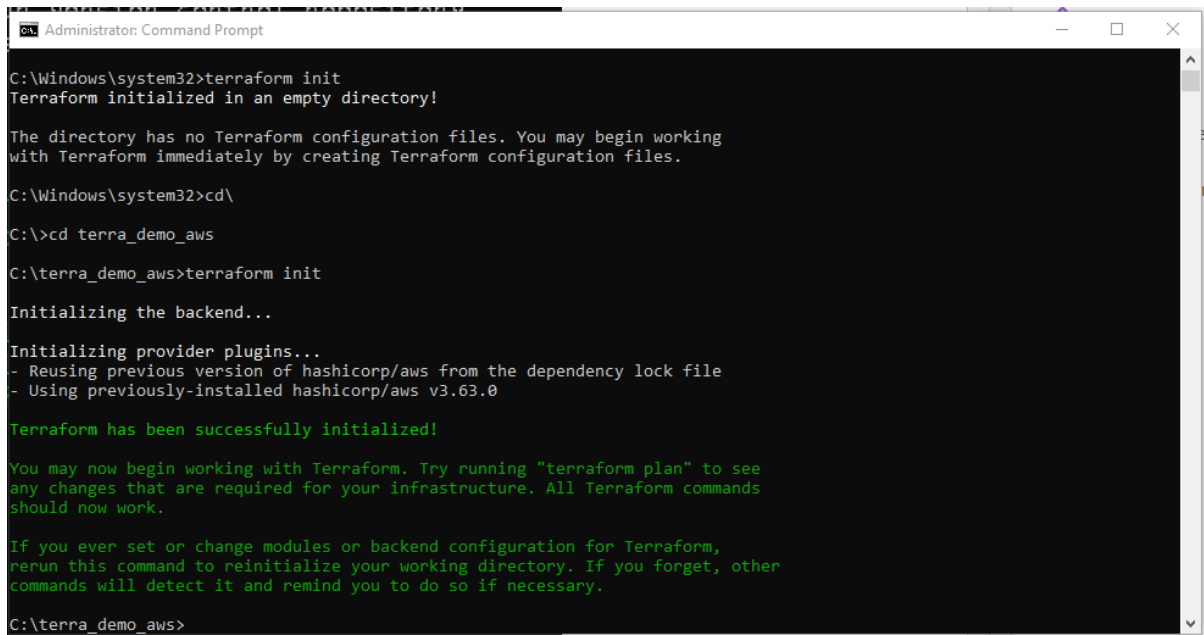


Terraform Creating AWS Virtual Machines

1. Install Terraform and AWS CLI



```
Administrator: Command Prompt

C:\Windows\system32>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.

C:\Windows\system32>cd\

C:\>cd terra_demo_aws

C:\terra_demo_aws>terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v3.63.0

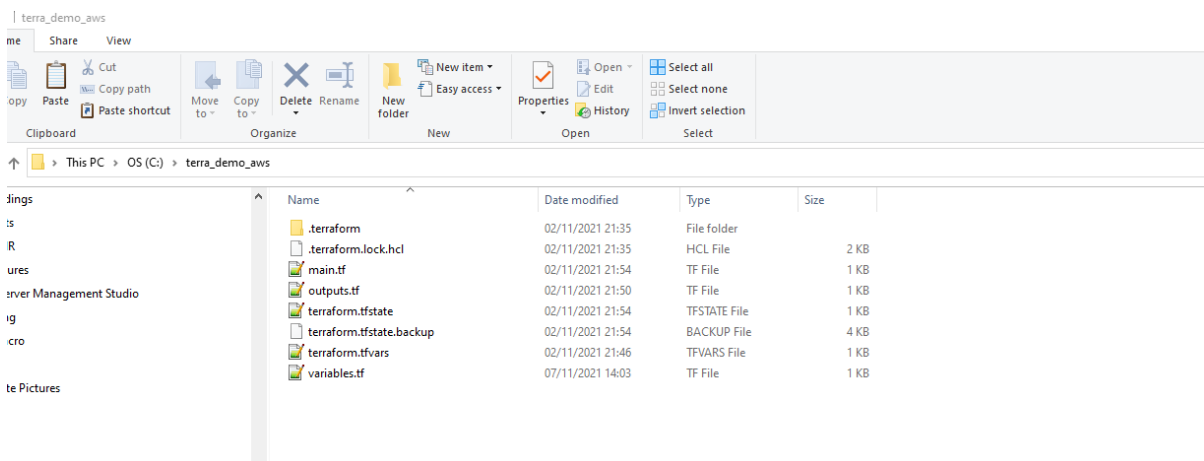
Terraform has been successfully initialized!

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.

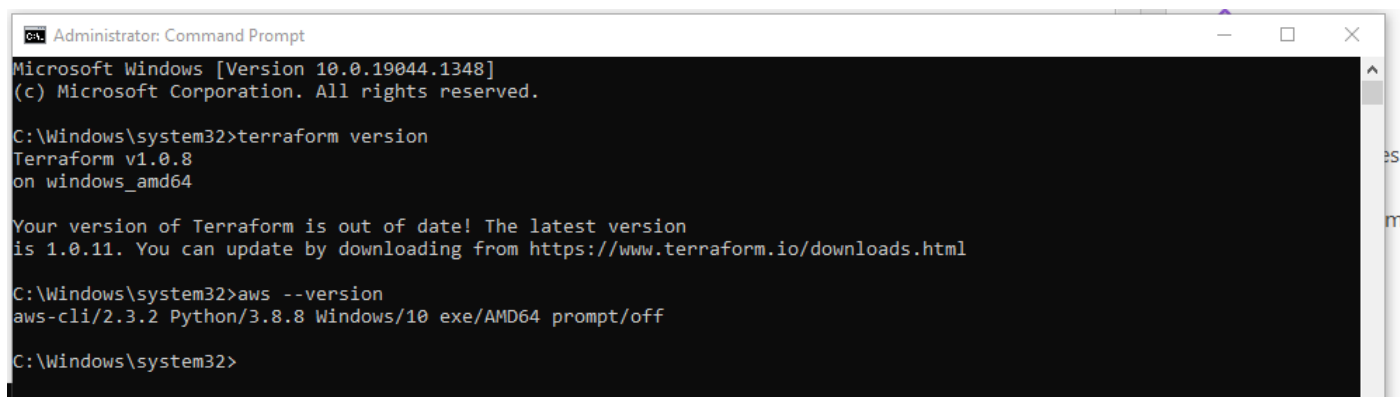
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

C:\terra_demo_aws>
```

2. Terraform folder with files created



3. Terraform Versions



```
Administrator: Command Prompt

Microsoft Windows [Version 10.0.19044.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>terraform version
Terraform v1.0.8
on windows_amd64

Your version of Terraform is out of date! The latest version
is 1.0.11. You can update by downloading from https://www.terraform.io/downloads.html

C:\Windows\system32>aws --version
aws-cli/2.3.2 Python/3.8.8 Windows/10 exe/AMD64 prompt/off

C:\Windows\system32>
```

4. Terraform Init & Validate

```
Administrator: Command Prompt

C:\terra_demo_aws>terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v3.63.0

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.

C:\terra_demo_aws>terraform validate
Success! The configuration is valid.

C:\terra_demo_aws>
```

5. Terraform apply

```
Administrator: Command Prompt - terraform apply

C:\terra_demo_aws>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.app_server will be created
+ resource "aws_instance" "app_server" {
  + ami                  = "ami-08d70e59c07c61a3a"
  + 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_termination = (known after apply)
  + ebs_optimized        = (known after apply)
  + get_password_data     = false
  + host_id              = (known after apply)
  + id                   = (known after apply)
  + instance_initiated_shutdown_behavior = (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
  + subnet_id            = (known after apply)
  + tags                 = {
    + "Name" = "L00090217AppServerInstance"
  }
  + tags_all              = {
    + "Name" = "L00090217AppServerInstance"
  }
}
```

```
Administrator: Command Prompt

+ http_tokens = (known after apply)
}

+ network_interface {
+ delete_on_termination = (known after apply)
+ device_index = (known after apply)
+ network_interface_id = (known after apply)
}

+ root_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ tags = (known after apply)
+ throughput = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (known after apply)
}
}

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

Changes to Outputs:
+ instance_id = (known after apply)
+ instance_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.app_server: Creating...
aws_instance.app_server: Still creating... [10s elapsed]
aws_instance.app_server: Still creating... [20s elapsed]
aws_instance.app_server: Still creating... [30s elapsed]
aws_instance.app_server: Still creating... [40s elapsed]
aws_instance.app_server: Still creating... [50s elapsed]
aws_instance.app_server: Creation complete after 53s [id=i-0b997f21043bcc73d]

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

Outputs:

instance_id = "i-0b997f21043bcc73d"
instance_public_ip = "35.85.51.178"

C:\terra_demo_aws>
```

6. Virtual machines being installed with previous version of the instance.

Instances | EC2 Management Console

us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Instances:

Services Search for services, features, blogs, docs, and more [Alt+S]

New EC2 Experience Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits

Instances
Instances New
Instance Types

Successfully terminated i-083c4dcf5c29f7e91

Instances (4) Info

Filter instances

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	L00090217Ap...	i-083c4dcf5c29f7e91	Terminated	t2.micro	-	No alarms	us-west-2a	-	-	-
<input type="checkbox"/>	L00090217Ap...	i-03c373077a1754ea9	Terminated	t2.micro	-	No alarms	us-west-2a	-	-	-
<input type="checkbox"/>	L00090217Ap...	i-02968d549f1996a6f	Terminated	t2.micro	-	No alarms	us-west-2a	-	-	-
<input type="checkbox"/>	L00090217Ap...	i-0b997f21043bcc73d	Pending	t2.micro	-	No alarms	us-west-2a	ec2-35-85-51-178.us-w...	35.85.51.178	-

7. Virtual machines created.

The screenshot displays the AWS Management Console interface for an EC2 instance. The top navigation bar shows the 'Instances' page. The left sidebar contains a navigation menu with options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main content area shows the 'Instance summary for i-0b997f21043bcc73d (L00090217AppServerInstance)'. The instance is in a 'Terminated' state. The summary includes details such as the Instance ID, Public IPv4 address (35.85.51.178), Private IPv4 address (ip-172-31-18-152.us-west-2.compute.internal), VPC ID (vpc-06f37873cdf63962), Subnet ID (subnet-06f5665d06e6f8ad5), Platform (Ubuntu), AMI ID (ami-830c94e3), AMI name (ubuntu/images/hvm/ubuntu-precise-12.04-amd64-server-20170502), and Launch time (Mon Nov 22 2021 20:37:04 GMT+0000).

Instance summary for i-0b997f21043bcc73d (L00090217AppServerInstance)

Updated less than a minute ago

Instance ID: i-0b997f21043bcc73d (L00090217AppServerInstance)

Public IPv4 address: 35.85.51.178 | [open address](#)

Private IPv4 address: ip-172-31-18-152.us-west-2.compute.internal

VPC ID: vpc-06f37873cdf63962

Subnet ID: subnet-06f5665d06e6f8ad5

Instance state: Pending

Instance type: t2.micro

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)

Platform: Ubuntu (Inferred)

Platform details: Linux/UNIX

Launch time: Mon Nov 22 2021 20:37:04 GMT+0000 (Greenwich Mean Time) (1 minute)

AMI ID: ami-830c94e3

AMI name: ubuntu/images/hvm/ubuntu-precise-12.04-amd64-server-20170502

AMI location: 099720109477/ubuntu/images/hvm/ubuntu-precise-12.04-amd64-server-

Monitoring: disabled

Termination protection: Disabled

Lifecycle: normal

8. Terraform Destroy

The screenshot displays the AWS Management Console interface for the 'Instances' page. The top navigation bar shows the 'Instances' page. The left sidebar contains a navigation menu with options like 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Images', 'Elastic Block Store', and 'Network & Security'. The main content area shows a list of instances. The list includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 address, and Elastic IP. The instances listed are all in a 'Terminated' state.

Instances (4)

Filter instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
L00090217Ap...	i-083c4dcf5c29f7e91	Terminated	t2.micro	-	No alarms	us-west-2a	-	-	-
L00090217Ap...	i-03c373077a1754ea9	Terminated	t2.micro	-	No alarms	us-west-2a	-	-	-
L00090217Ap...	i-02968d549f1996a6f	Terminated	t2.micro	-	No alarms	us-west-2a	-	-	-
L00090217Ap...	i-0b997f21043bcc73d	Terminated	t2.micro	-	No alarms	us-west-2a	-	-	-