

# Infrastructure Automation using Terraform

Working with Workspaces



# Table of Contents

Prerequisite:.....	3
Walkthrough: .....	3
Part 1: Initializing Terraform Directory .....	3
Part 2: Working with Workspaces.....	6
Part 3: Destroying Resources .....	10

## Infrastructure Automation using Terraform – Lab Guide

This Activity demonstrates the working of Workspaces in Terraform. Using Workspaces, we can create multiple copies of infrastructure having same configuration files.

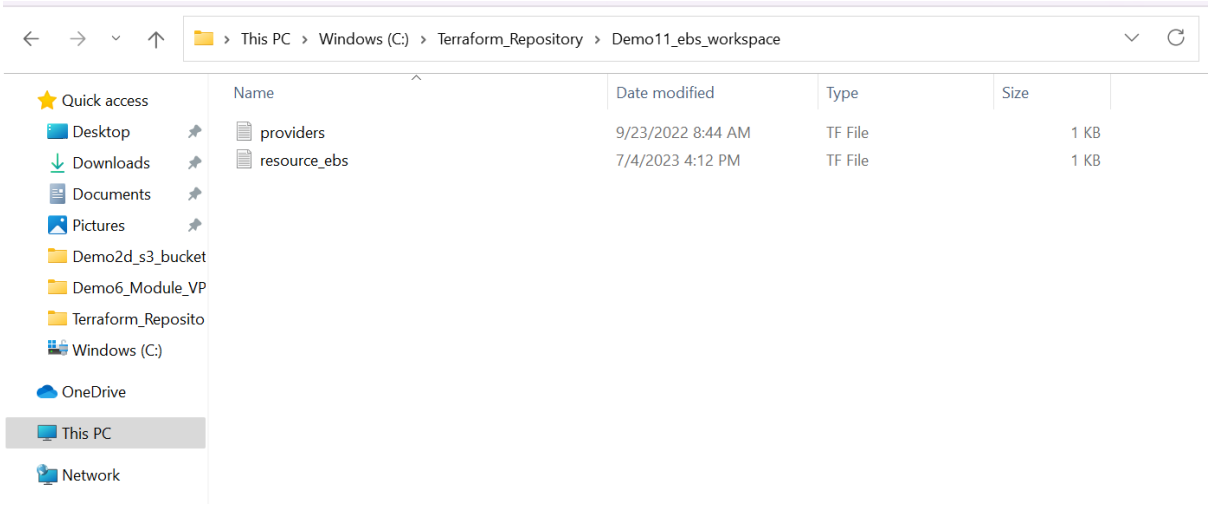
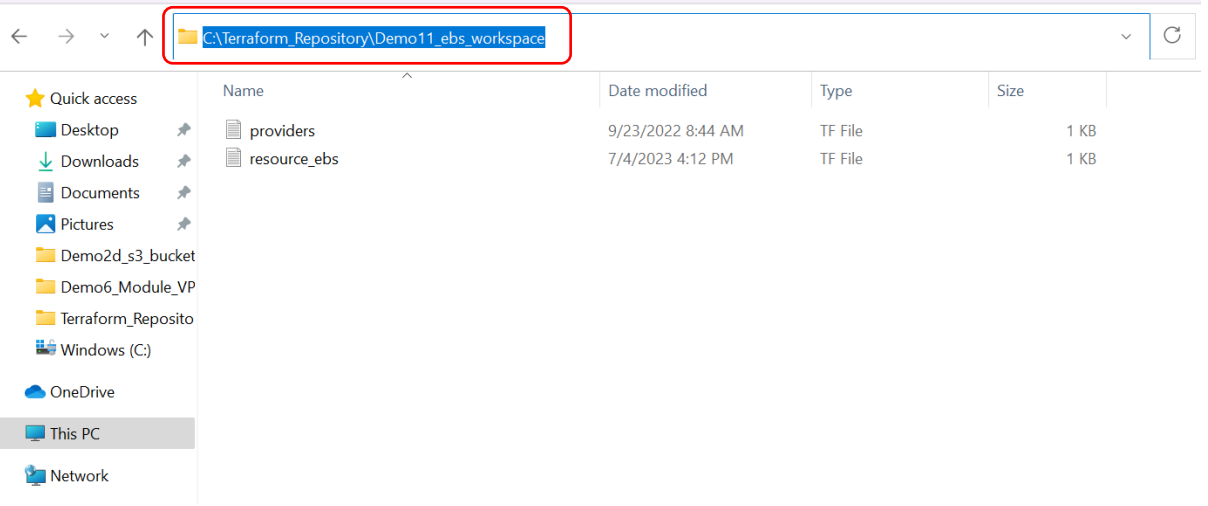
### Prerequisite:

- 1) Download the zip file shared by the trainer and extract it.

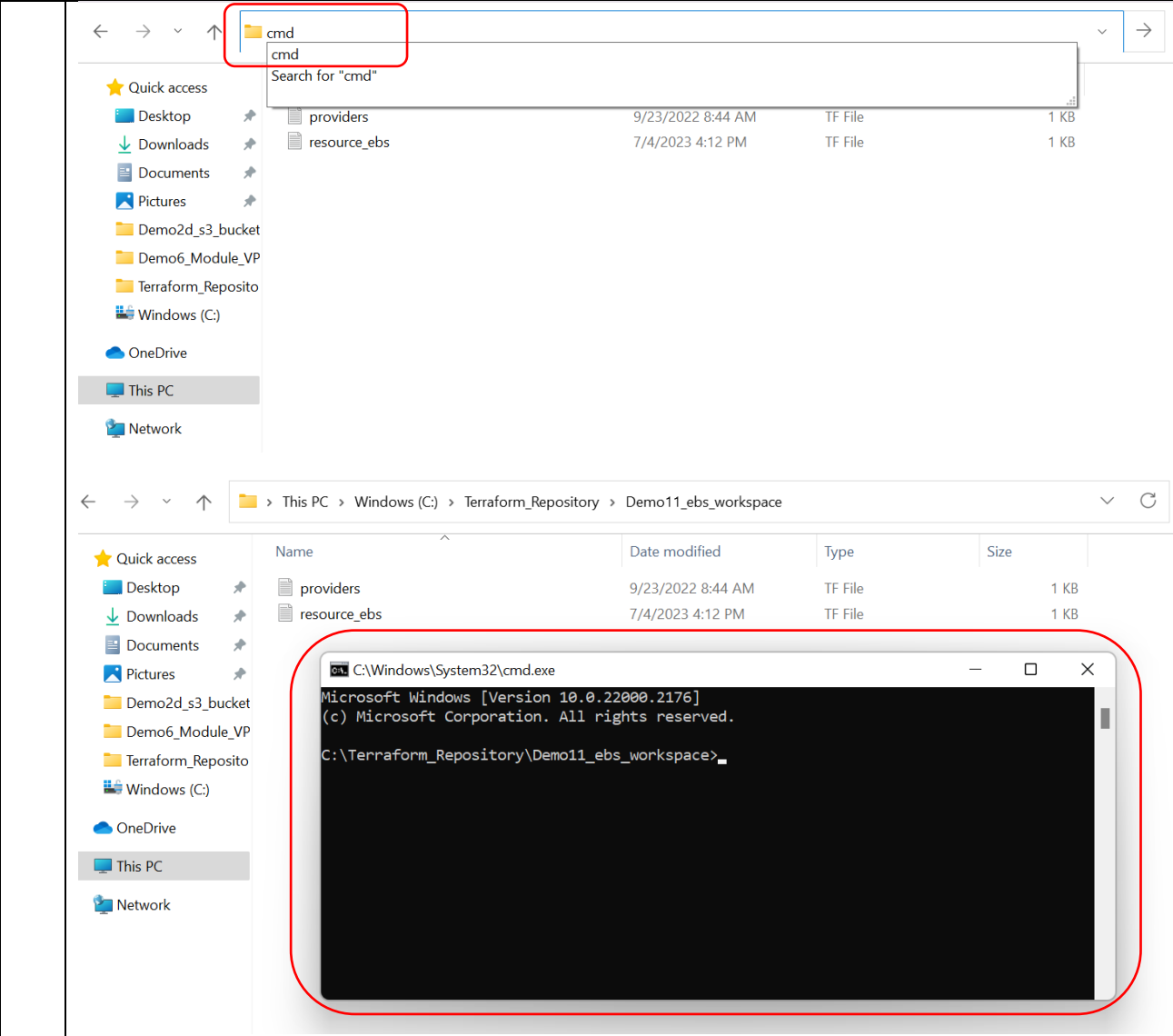
### Walkthrough:

1. Initializing Terraform Directory
2. Working with Workspaces
3. Destroying Workspaces

#### Part 1: Initializing Terraform Directory

1	<p>Open the extracted folder and navigate to “.tf” files.</p> 
2	<p>Click on the Address bar and type cmd. Press Enter ( It will open a command prompt from that location ).</p> 

## Infrastructure Automation using Terraform – Lab Guide



The screenshot shows a Windows File Explorer window with the address bar set to `C:\Terraform_Repository\Demo11_ebs_workspace`. The left sidebar shows the 'Quick access' pane with 'This PC' selected. The main pane displays a list of files:

Name	Date modified	Type	Size
providers	9/23/2022 8:44 AM	TF File	1 KB
resource_ebs	7/4/2023 4:12 PM	TF File	1 KB

Overlaid on the File Explorer is a Command Prompt window titled `C:\Windows\System32\cmd.exe`. The prompt shows the current directory as `C:\Terraform_Repository\Demo11_ebs_workspace>`.

3 Execute below command to initialize the current directory as Terraform directory which enables us to run terraform commands to manage Infrastructure.

**Command :**

```
C:\Terraform_Repository\Demo11_ebs_workspace>terraform init
```

**Result :**

	<pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform init  Initializing the backend...  Initializing provider plugins... - Finding latest version of hashicorp/aws... - Installing hashicorp/aws v5.9.0... - Installed hashicorp/aws v5.9.0 (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.  C:\Terraform_Repository\Demo11_ebs_workspace&gt; </pre>
4	<p>Next execute below command to validate syntax and configuration of terraform configuration files. If everything is proper, it will return a success message otherwise it will display the errors.</p> <p><b>Command :</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform validate </pre> <p><b>Result :</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform validate Success! The configuration is valid.  C:\Terraform_Repository\Demo11_ebs_workspace&gt; </pre>
5	<p>Next run below command and observe the output. The output contains information depicting all the changes which will happen in the AWS cloud. It is like dry-run to ensure whatever we are trying to do using terraform commands is what we want.</p> <p><b>Command :</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform plan -out "ws.tfplan" </pre> <p><b>Result :</b></p>

	<pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform plan -out "ws.tfplan"  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_ebs_volume.myvol will be created + resource "aws_ebs_volume" "myvol" {   + arn                = (known after apply)   + availability_zone   = "eu-west-1a"   + encrypted          = true   + final_snapshot     = false   + id                 = (known after apply)   + iops               = (known after apply)   + kms_key_id         = (known after apply)   + size               = 10   + snapshot_id        = (known after apply)   + tags_all           = (known after apply)   + throughput         = (known after apply)   + type               = (known after apply) }  Plan: 1 to add, 0 to change, 0 to destroy.  Changes to Outputs: + EBSDetails = {   + arn                = (known after apply)   + availability_zone   = "eu-west-1a" </pre>
6	<p>For creating resources in default workspace, execute below command.</p> <p><b>Command:</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform apply "ws.tfplan" </pre> <p><b>Result:</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform apply "ws.tfplan" aws_ebs_volume.myvol: Creating... aws_ebs_volume.myvol: Still creating... [10s elapsed] aws_ebs_volume.myvol: Creation complete after 12s [id=vol-0243d1fd15fbf0168]  Apply complete! Resources: 1 added, 0 changed, 0 destroyed.  Outputs:  EBSDetails = {   "arn" = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-0243d1fd15fbf0168"   "availability_zone" = "eu-west-1a"   "encrypted" = true   "final_snapshot" = false   "id" = "vol-0243d1fd15fbf0168"   "iops" = 100   "kms_key_id" = "arn:aws:kms:eu-west-1:386057849409:key/fe35c816-b038-485b-bd40-672bc3707ec6"   "multi_attach_enabled" = false   "outpost_arn" = ""   "size" = 10   "snapshot_id" = ""   "tags" = tomap(null) /* of string */   "tags_all" = tomap({})   "throughput" = 0   "timeouts" = null /* object */   "type" = "gp2" }  C:\Terraform_Repository\Demo11_ebs_workspace&gt; </pre>

## Part 2: Working with Workspaces

1	<p>For finding out in which workspace we are creating resources, we can execute below command.</p> <p><b>Command:</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace show </pre>
---	---

	<p><b>Result :</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace show default</pre> <p>C:\Terraform_Repository\Demo11_ebs_workspace&gt;_</p> <p><b>NOTE :</b> In Previous Step, we have created resources in default workspace.</p>
2	<p>For creating same kind of infrastructure with same configuration files and manage it with a different state file. We can create new workspaces using below command.</p> <p><b>Command :</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace new "Dev"</pre> <p><b>Result :</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace new "Dev" Created and switched to workspace "Dev"!  You're now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.  C:\Terraform_Repository\Demo11_ebs_workspace&gt;</pre>
3	<p>After creating a new workspace, we can utilize same configuration file to create another set of same resources in new workspace.</p> <p><b>Command :</b> ( for previewing the changes )</p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform plan -out "dev_ws.tfplan"</pre> <p><b>Result :</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform plan -out "dev_ws.tfplan"  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_ebs_volume.myvol will be created + resource "aws_ebs_volume" "myvol" { +   arn                = (known after apply) +   availability_zone  = "eu-west-1a" +   encrypted          = true +   final_snapshot     = false +   id                 = (known after apply) +   iops               = (known after apply) +   kms_key_id         = (known after apply) +   size               = 10 +   snapshot_id        = (known after apply) +   tags_all           = (known after apply) +   throughput         = (known after apply) +   type               = (known after apply) }  Plan: 1 to add, 0 to change, 0 to destroy.  Changes to Outputs: + EBSDetails = { +   arn                = (known after apply) +   availability_zone  = "eu-west-1a" +   encrypted          = true +   final_snapshot     = false +   id                 = (known after apply) +   iops               = (known after apply) +   kms_key_id         = (known after apply) +   size               = 10 +   snapshot_id        = (known after apply) +   tags_all           = (known after apply) +   throughput         = (known after apply) +   type               = (known after apply) }</pre> <p><b>Command :</b> ( for creating resources )</p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform apply "dev_ws.tfplan"</pre> <p><b>Result :</b></p>

	<pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform apply "dev_ws.tfplan" aws_ebs_volume.myvol: Creating... aws_ebs_volume.myvol: Still creating... [10s elapsed] aws_ebs_volume.myvol: Creation complete after 12s [id=vol-05e86de720c50cc0b]  Apply complete! Resources: 1 added, 0 changed, 0 destroyed.  Outputs:  EBSDetails = {   "arn" = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-05e86de720c50cc0b"   "availability_zone" = "eu-west-1a"   "encrypted" = true   "final_snapshot" = false   "id" = "vol-05e86de720c50cc0b"   "iops" = 100   "kms_key_id" = "arn:aws:kms:eu-west-1:386057849409:key/fe35c816-b038-485b-bd40-672bc3707ec6"   "multi_attach_enabled" = false   "outpost_arn" = ""   "size" = 10   "snapshot_id" = ""   "tags" = tomap(null) /* of string */   "tags_all" = tomap({})   "throughput" = 0   "timeouts" = null /* object */   "type" = "gp2" } </pre>
4	<p>In similar way, create another workspace and create resources in the new workspace.</p> <p><b>Command :</b> ( for creating a new workspace )</p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace new "Staging" </pre> <p><b>Result :</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace new "Staging" Created and switched to workspace "Staging"!  You're now on a new, empty workspace. Workspaces isolate their state, so if you run "terraform plan" Terraform will not see any existing state for this configuration.  C:\Terraform_Repository\Demo11_ebs_workspace&gt; </pre> <p><b>Command :</b> ( for previewing changes )</p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform plan -out "staging.tfplan" </pre> <p><b>Result :</b></p>



	<pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform plan -out "staging.tfplan"  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_ebs_volume.myvol will be created + resource "aws_ebs_volume" "myvol" {   + arn                = (known after apply)   + availability_zone   = "eu-west-1a"   + encrypted          = true   + final_snapshot     = false   + id                 = (known after apply)   + iops               = (known after apply)   + kms_key_id         = (known after apply)   + size               = 10   + snapshot_id        = (known after apply)   + tags_all           = (known after apply)   + throughput         = (known after apply)   + type               = (known after apply) }  Plan: 1 to add, 0 to change, 0 to destroy.  Changes to Outputs: + EBSDetails = {   + arn                = (known after apply)   + availability_zone   = "eu-west-1a"   + encrypted          = true </pre> <p><b>Command :</b> ( for creating resources )</p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform apply "staging.tfplan" </pre> <p><b>Result :</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform apply "staging.tfplan" aws_ebs_volume.myvol: Creating... aws_ebs_volume.myvol: Still creating... [10s elapsed] aws_ebs_volume.myvol: Creation complete after 12s [id=vol-0d6da05ae5b57f12f]  Apply complete! Resources: 1 added, 0 changed, 0 destroyed.  Outputs:  EBSDetails = {   "arn" = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-0d6da05ae5b57f12f"   "availability_zone" = "eu-west-1a"   "encrypted" = true   "final_snapshot" = false   "id" = "vol-0d6da05ae5b57f12f"   "iops" = 100   "kms_key_id" = "arn:aws:kms:eu-west-1:386057849409:key/fe35c816-b038-485b-bd40-672bc3707ec6"   "multi_attach_enabled" = false   "outpost_arn" = ""   "size" = 10   "snapshot_id" = ""   "tags" = tomap(null) /* of string */   "tags_all" = tomap({})   "throughput" = 0   "timeouts" = null /* object */   "type" = "gp2" }  C:\Terraform_Repository\Demo11_ebs_workspace&gt; </pre>
5	<p>To list out all the available workspaces , we can execute below command. Your current workspace will be indicated by <code>***</code> symbol beside the workspace name.</p> <p><b>Command :</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace list </pre> <p><b>Result :</b></p>

	<pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace list default Dev * Staging  C:\Terraform_Repository\Demo11_ebs_workspace&gt;_</pre>
6	<p>To switch from one workspace to another workspace we can execute below command.</p> <p><b>Command :</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace select Dev</pre> <p><b>Result :</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace select Dev Switched to workspace "Dev".  C:\Terraform_Repository\Demo11_ebs_workspace&gt;</pre> <p><b>Command :</b> (To identify in which workspace you are present, execute below command)</p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace show</pre> <p><b>Result :</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace show Dev  C:\Terraform_Repository\Demo11_ebs_workspace&gt;_</pre>

### Part 3: Destroying Resources

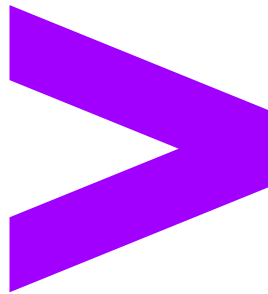
1	<p>Execute below command to destroy the resources which we have created in Dev Workspace. After you execute below command, it will show you what changes will be done and before doing those changes it will ask for your approval. So, if you want to proceed with destroying resources, provide "yes".</p> <p><b>Command:</b> ( In Dev Workspace )</p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform destroy</pre> <p><b>Result:</b></p>
---	---

	<pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform destroy aws_ebs_volume.myvol: Refreshing state... [id=vol-05e86de720c50cc0b]  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_ebs_volume.myvol will be destroyed - resource "aws_ebs_volume" "myvol" {   - arn                = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-05e86de720c50cc0b" -&gt; null   - availability_zone   = "eu-west-1a" -&gt; null   - encrypted           = true -&gt; null   - final_snapshot      = false -&gt; null   - id                  = "vol-05e86de720c50cc0b" -&gt; null   - iops                = 100 -&gt; null   - kms_key_id          = "arn:aws:kms:eu-west-1:386057849409:key/fe35c816-b038-485b-bd40-672bc3707ec6" -&gt; null   - multi_attach_enabled = false -&gt; null   - size                = 10 -&gt; null   - tags                = {} -&gt; null   - tags_all            = {} -&gt; null   - throughput          = 0 -&gt; null   - type                = "gp2" -&gt; null }  Plan: 0 to add, 0 to change, 1 to destroy.  Changes to Outputs:   - EBSDetails = {     - arn                = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-05e86de720c50cc0b"   } </pre> <p>Do you really want to destroy all resources in workspace "Dev"?  Terraform will destroy all your managed infrastructure, as shown above.  There is no undo. Only 'yes' will be accepted to confirm.</p> <p>Enter a value:</p> <p>Do you really want to destroy all resources in workspace "Dev"?  Terraform will destroy all your managed infrastructure, as shown above.  There is no undo. Only 'yes' will be accepted to confirm.</p> <p>Enter a value: yes</p> <pre> aws_ebs_volume.myvol: Destroying... [id=vol-05e86de720c50cc0b] aws_ebs_volume.myvol: Still destroying... [id=vol-05e86de720c50cc0b, 10s elapsed] aws_ebs_volume.myvol: Destruction complete after 12s  Destroy complete! Resources: 1 destroyed.  C:\Terraform_Repository\Demo11_ebs_workspace&gt; </pre>
2	<p>After clearing up all the resources in Dev Workspace, we can proceed to delete Dev Workspace by executing below command. Before deleting Dev Workspace, we need to switch to another workspace.</p> <p><b>Command:</b> ( for switching workspaces execute below command )</p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace select Staging </pre> <p><b>Result:</b></p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace select Staging Switched to workspace "Staging".  C:\Terraform_Repository\Demo11_ebs_workspace&gt; </pre> <p><b>Command:</b> ( for deleting workspaces execute below command )</p> <pre> C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace delete Dev </pre> <p><b>Result:</b></p>

	<pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace delete Dev Deleted workspace "Dev"!  C:\Terraform_Repository\Demo11_ebs_workspace&gt;_</pre> <p><b>NOTE:</b> If we delete workspace before clearing up the resources in that workspace, then along with the workspace state file will be getting deleted. So, we will lose control on those resources. If we want to delete them, then we must do it either manually or import those resources under Terraform and then destroy them.</p>
3	<p>Execute below command to destroy the resources in Staging Workspace. After you execute below command, it will show you what changes will be done and before doing those changes it will ask for your approval. So, if you want to proceed with destroying resources, provide "yes".</p> <p><b>Command:</b> ( In Staging Workspace )</p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform destroy</pre> <p><b>Result:</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform destroy aws_ebs_volume.myvol: Refreshing state... [id=vol-0d6da05ae5b57f12f]  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_ebs_volume.myvol will be destroyed - resource "aws_ebs_volume" "myvol" {   - arn              = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-0d6da05ae5b57f12f" -&gt; null   - availability_zone = "eu-west-1a" -&gt; null   - encrypted         = true -&gt; null   - final_snapshot    = false -&gt; null   - id                = "vol-0d6da05ae5b57f12f" -&gt; null   - iops              = 100 -&gt; null   - kms_key_id        = "arn:aws:kms:eu-west-1:386057849409:key/fe35c816-b038-485b-bd40-672bc3707ec6" -&gt; null   - multi_attach_enabled = false -&gt; null   - size              = 10 -&gt; null   - tags              = {} -&gt; null   - tags_all          = {} -&gt; null   - throughput        = 0 -&gt; null   - type              = "gp2" -&gt; null }  Plan: 0 to add, 0 to change, 1 to destroy.  Changes to Outputs:   - EBSDetails = {     - arn              = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-0d6da05ae5b57f12f"</pre> <pre>Do you really want to destroy all resources in workspace "Staging"? Terraform will destroy all your managed infrastructure, as shown above. There is no undo. Only 'yes' will be accepted to confirm.  Enter a value:</pre> <pre>Do you really want to destroy all resources in workspace "Staging"? Terraform will destroy all your managed infrastructure, as shown above. There is no undo. Only 'yes' will be accepted to confirm.  Enter a value: yes  aws_ebs_volume.myvol: Destroying... [id=vol-0d6da05ae5b57f12f] aws_ebs_volume.myvol: Still destroying... [id=vol-0d6da05ae5b57f12f, 10s elapsed] aws_ebs_volume.myvol: Destruction complete after 12s  Destroy complete! Resources: 1 destroyed.  C:\Terraform_Repository\Demo11_ebs_workspace&gt;</pre>
4	<p>After clearing up all the resources in Staging Workspace, we can proceed to delete Staging Workspace by executing below command. Before deleting Staging Workspace, we need to switch to another workspace.</p>

	<p><b>Command:</b> ( for switching workspaces execute below command )</p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace select default</pre> <p><b>Result:</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace select default Switched to workspace "default". C:\Terraform_Repository\Demo11_ebs_workspace&gt;</pre> <p><b>Command:</b> ( for deleting workspaces execute below command )</p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace delete Staging</pre> <p><b>Result:</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform workspace delete Staging Deleted workspace "Staging"! C:\Terraform_Repository\Demo11_ebs_workspace&gt;_</pre>
5	<p>Execute below command to destroy the resources in default Workspace. After you execute below command, it will show you what changes will be done and before doing those changes it will ask for your approval. So, if you want to proceed with destroying resources, provide “yes”.</p> <p><b>Command:</b> ( In default Workspace )</p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform destroy</pre> <p><b>Result:</b></p> <pre>C:\Terraform_Repository\Demo11_ebs_workspace&gt;terraform destroy aws_ebs_volume.myvol: Refreshing state... [id=vol-0243d1fd15fbf0168]  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_ebs_volume.myvol will be destroyed - resource "aws_ebs_volume" "myvol" {   - arn                = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-0243d1fd15fbf0168" -&gt; null   - availability_zone  = "eu-west-1a" -&gt; null   - encrypted          = true -&gt; null   - final_snapshot     = false -&gt; null   - id                 = "vol-0243d1fd15fbf0168" -&gt; null   - iops               = 100 -&gt; null   - kms_key_id         = "arn:aws:kms:eu-west-1:386057849409:key/fe35c816-b038-485b-bd40-672bc3707ec6" -&gt; null   - multi_attach_enabled = false -&gt; null   - size               = 10 -&gt; null   - tags               = {} -&gt; null   - tags_all           = {} -&gt; null   - throughput         = 0 -&gt; null   - type               = "gp2" -&gt; null }</pre> <p>Plan: 0 to add, 0 to change, 1 to destroy.</p> <p>Changes to Outputs:</p> <pre>- EBSDetails = {   - arn                = "arn:aws:ec2:eu-west-1:386057849409:volume/vol-0243d1fd15fbf0168"</pre> <p>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.</p> <p>Enter a value:</p>

```
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: yes  
  
aws_ebs_volume.myvol: Destroying... [id=vol-0243d1fd15fbf0168]  
aws_ebs_volume.myvol: Still destroying... [id=vol-0243d1fd15fbf0168, 10s elapsed]  
aws_ebs_volume.myvol: Destruction complete after 11s  
  
Destroy complete! Resources: 1 destroyed.  
  
C:\Terraform_Repository\Demo11_ebs_workspace>
```



Copyright © 2023 Accenture  
All rights reserved.  
Accenture and its logo are trademarks of Accenture.