

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

## Terraform Azure pre-requisites:

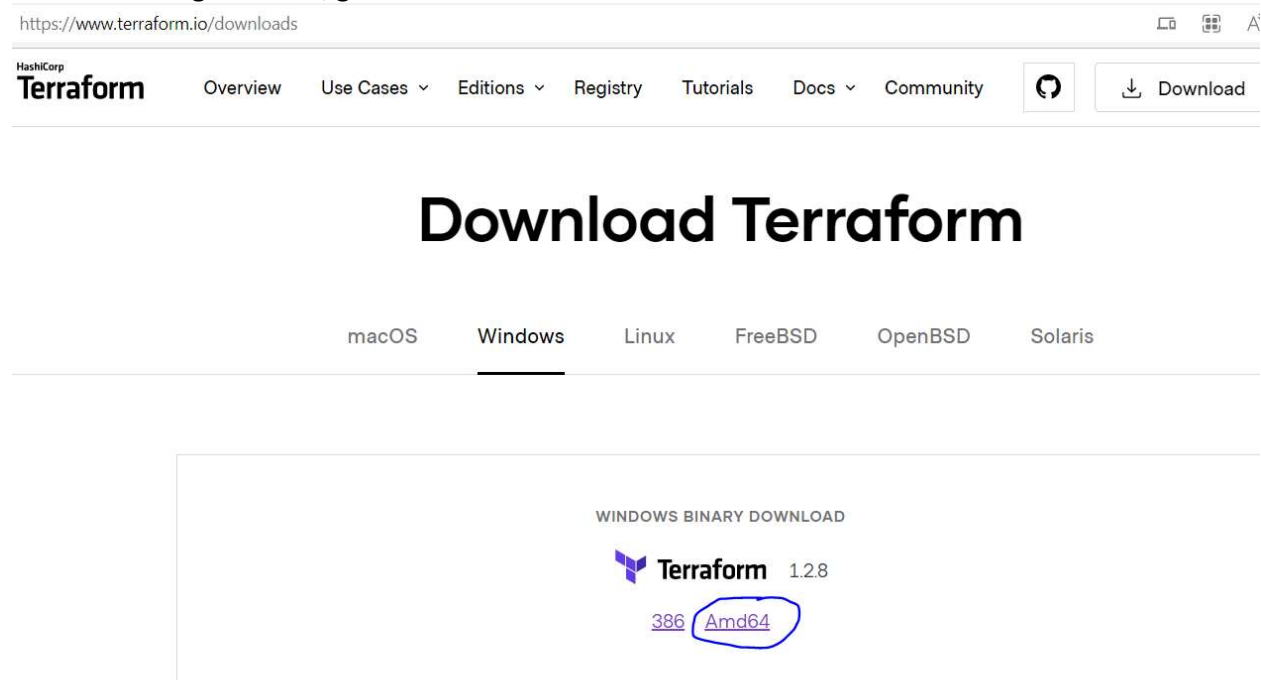
- Install Terraform
- Install azure cli

## Pre-requisite: Install Terraform

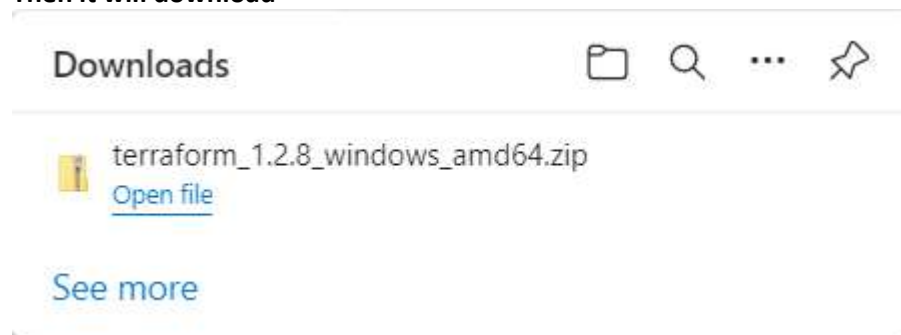
Download terraform exe file in terraform site:

<https://www.terraform.io/downloads>

Since I am using windows, go to windows tab and click Amd64



Then it will download



Then extract file you can see terraform executable file

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

> This PC > DATA (D:) > edge downloads > terraform\_1.2.8\_windows\_amd64

Name	Date modified	Type	Size
terraform	30/08/2022 4:05 am	Application	62,195 KB

**Then create a directory and copy the file**

> This PC > DATA (D:) > Terraform > bin

Name	Date modified	Type	Size
terraform	30/08/2022 4:05 am	Application	62,195 KB

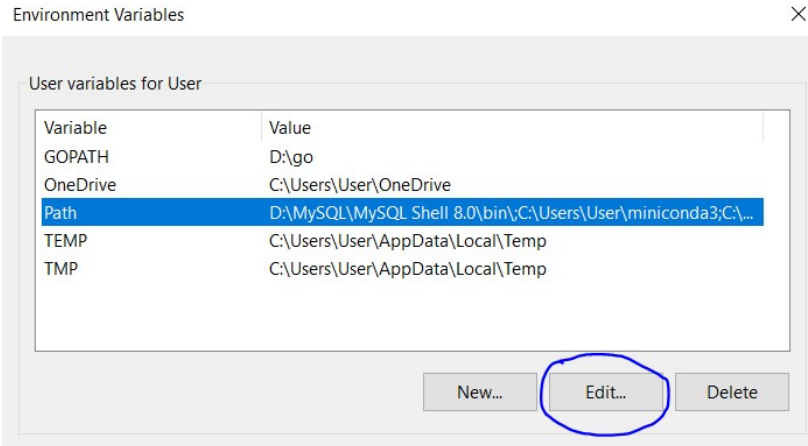
**Then edit windows user path variable**

The screenshot shows the Windows Start menu search interface. The search bar at the top contains the text 'Type here to search'. Below the search bar, a list of recent search results is displayed. The first result, 'Edit environment variables for your a...', is highlighted with a blue border. Other results include 'Snipping Tool', 'Word', 'Edit the system environment variables', 'Command Prompt', and 'Sign-in options'. On the right side of the search results, there is a section for 'Search people at sandboxcomp' with four user avatars: AD (Allan Deyoung), DS (Diego Siciliani), JS (Joni Sherman), and LR (Lynne Robbins). Below this, there is a section for 'Microsoft 365' with the text 'Find files or co-workers' and 'Get fast access to your info or team members.' There are also buttons for 'My documents', 'My profile', and 'My manager'.

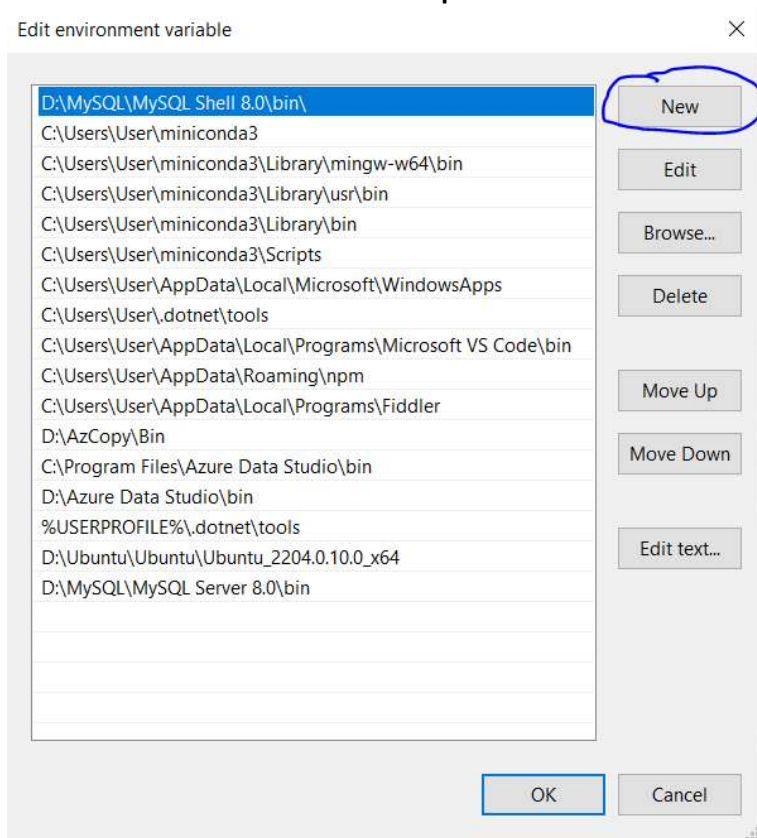
**Go to Path variable and click Edit... button**

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

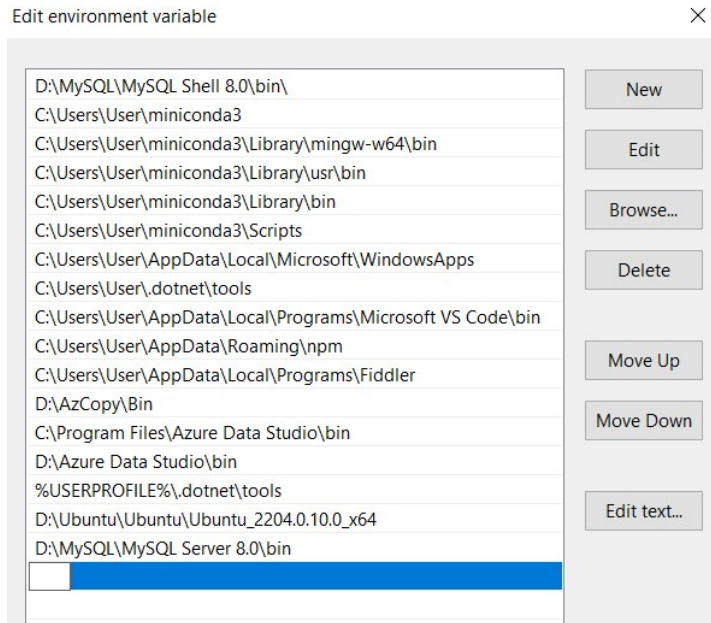


**Click New button to Add Terraform path**

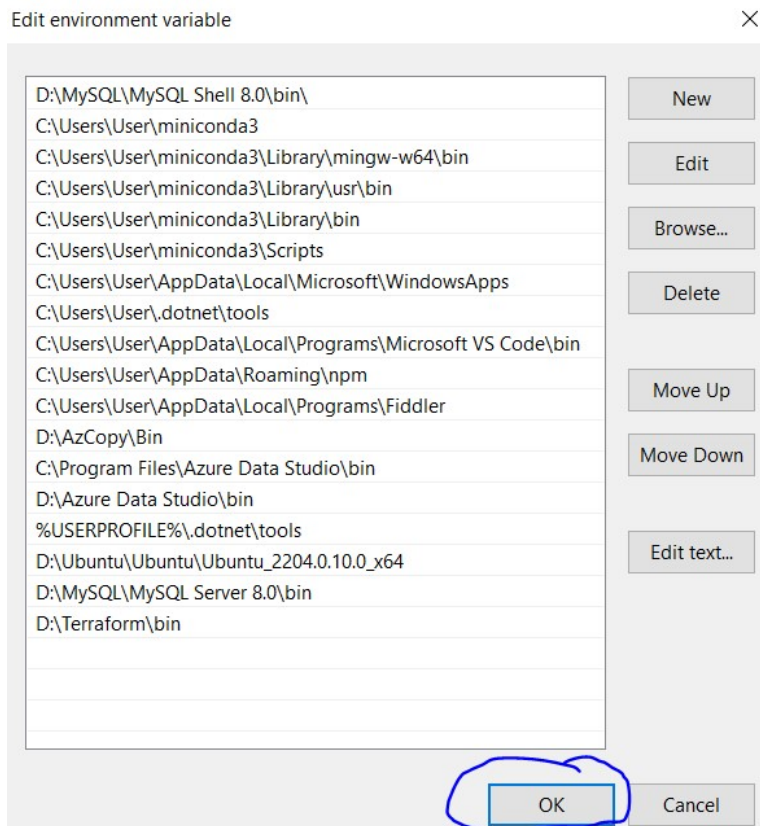


# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim



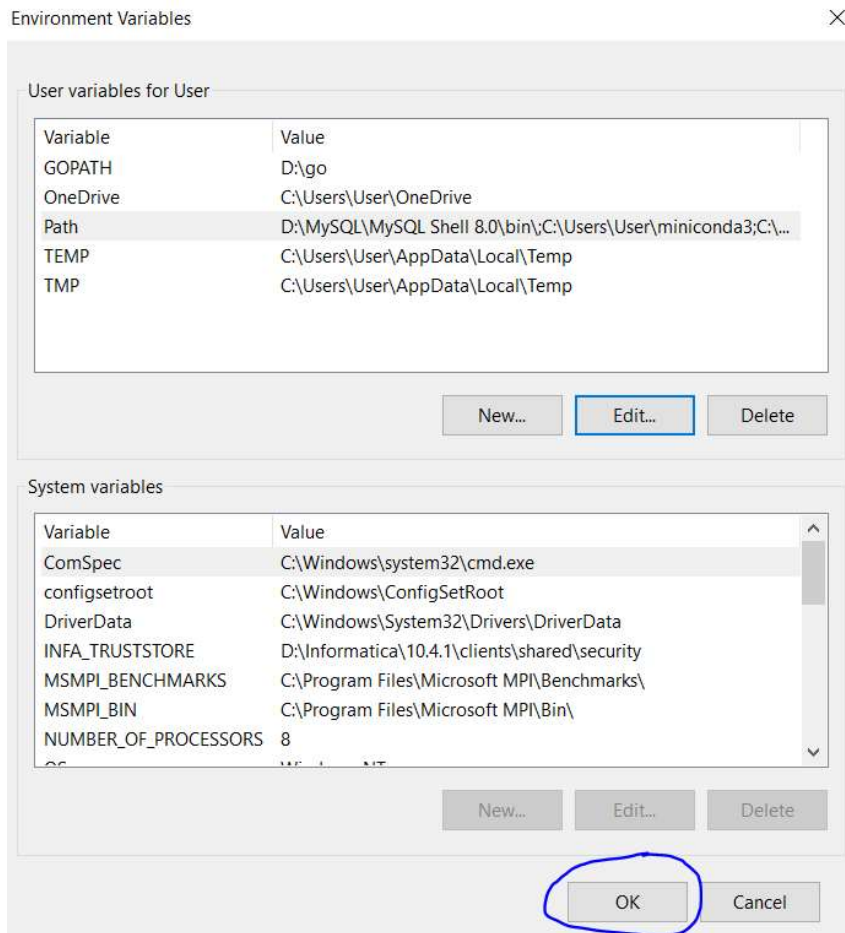
Click OK Button after specifying the path



Click OK Button again

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim



Open cmd and try to check if terraform is already accessible in any directory by typing:  
***terraform --version***

```
Microsoft Windows [Version 10.0.19043.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>terraform --version
Terraform v1.2.8
on windows_amd64

C:\Users\User>
```

Pre-requisite: Install Azure CLI

Go to the link below and click Latest release of the Azure CLI button to download the installer  
<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli>

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli

Microsoft Installer (MSI) | Microsoft Installer (MSI) with Command | Windows Package Manager

## Latest version

Download and install the latest release of the Azure CLI. When the installer asks if it can make changes to your computer, click the "Yes" box.

**Latest release of the Azure CLI**

## Specific version

To download the MSI installer for specific version, change the version segment in URL `https://azcliprod.blob.core.windows.net/msi/azure-cli-<version>.msi` and download it. Available versions can be found at [Azure CLI release notes](#).

**Downloads**

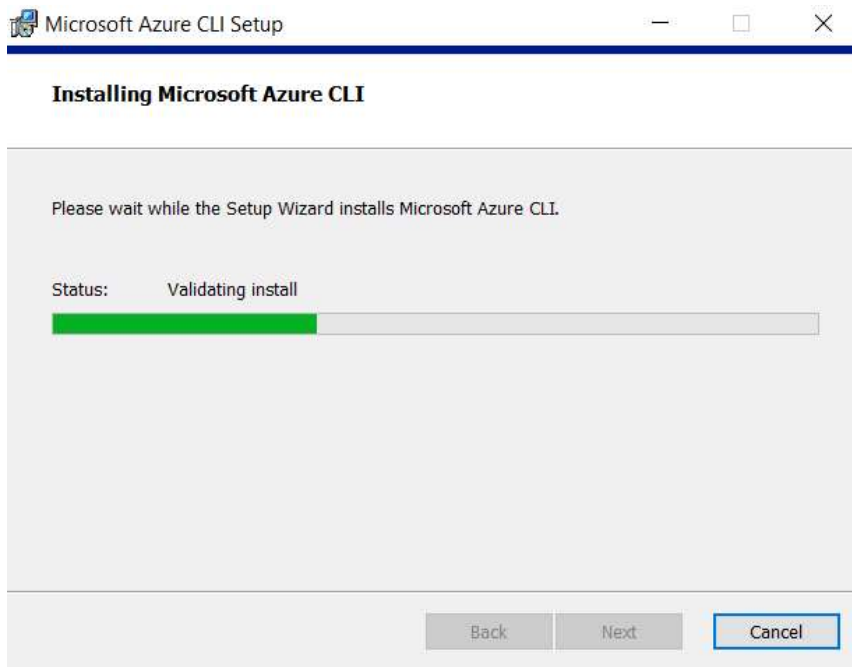
azure-cli-2.39.0.msi  
[Open file](#)

[See more](#)

Then open the installer and install the CLI

# Terraform Azure Sample: Azure Storage

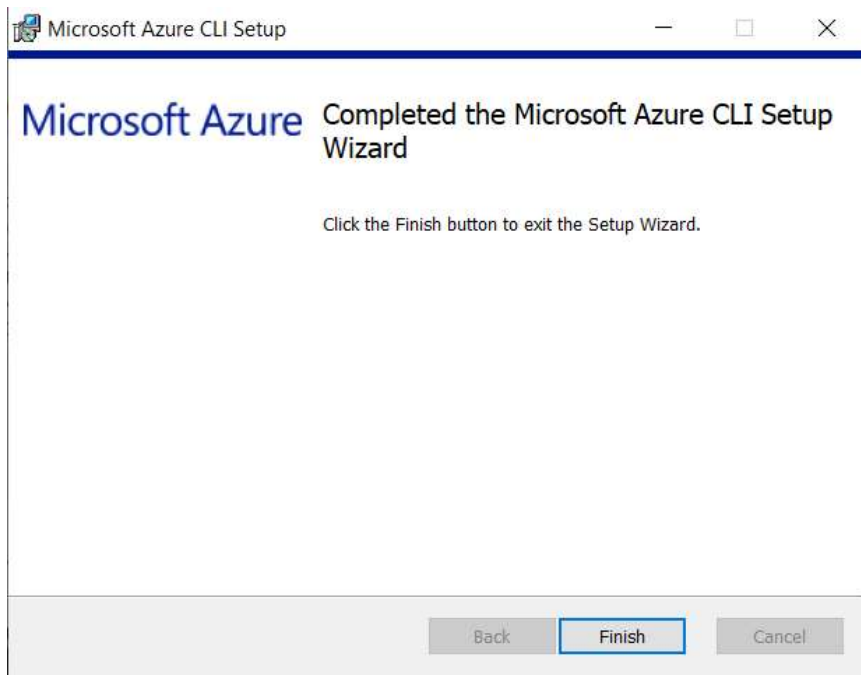
By Ronald Stewart Lim





# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim



Verify that the Azure CLI is installed successfully via cmd using command below  
**az --version**

```
C:\Users\User>az --version
azure-cli                        2.39.0

core                            2.39.0
telemetry                       1.0.6 *

Dependencies:
msal                            1.18.0b1
azure-mgmt-resource             21.1.0b1

Python location 'C:\Program Files (x86)\Microsoft SDKs\Azure\CLI2\python.exe'
Extensions directory 'C:\Users\User\.azure\cliextensions'

Python (Windows) 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 15:58:59) [MSC v.1929 32 bit (Intel)]

Legal docs and information: aka.ms/AzureCliLegal

You have 1 updates available.

Please let us know how we are doing: https://aka.ms/azureclihats
and let us know if you're interested in trying out our newest features: https://aka.ms/CLIUXstudy
```

## Terraform Implementation: Authentication via Azure CLI

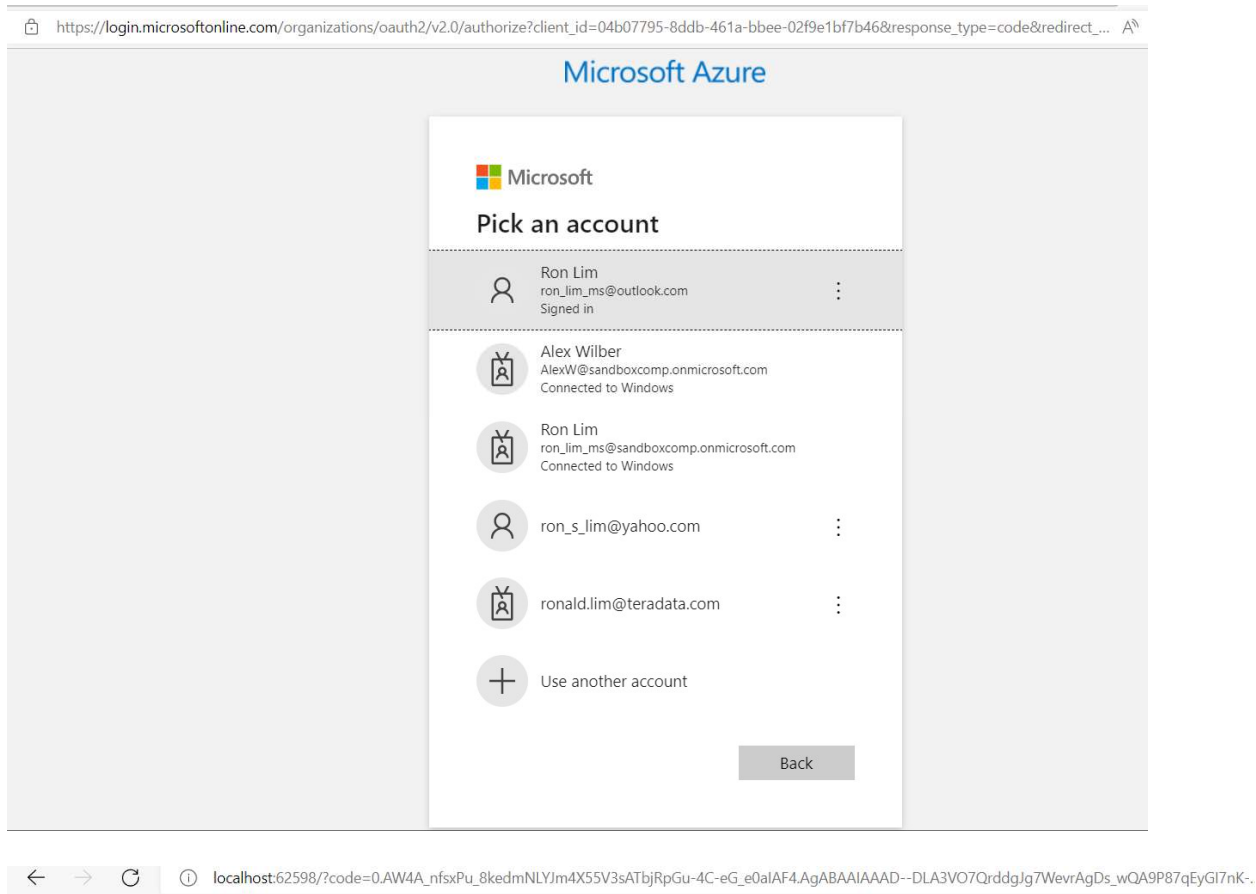
### Login to Azure via Azure CLI in cmd

```
D:\Terraform\terraform_azure>az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser.
open, use device code flow with 'az login --use-device-code'.
```



# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim



**You have logged into Microsoft Azure!**

You can close this window, or we will redirect you to the [Azure CLI documents](#) in 10 seconds.

```
D:\Terraform\terraform_azure>az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in
open, use device code flow with `az login --use-device-code`.
The following tenants require Multi-Factor Authentication (MFA). Use 'az login --tenant TENANT_ID' to explicitly login to a tenant.
f9a595e6-97f3-41bc-89c2-989f9f432da5 'Ron IT Company'
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "604c1504-c6a3-4080-81aa-b33091104187",
    "id": "6000e767-b051-49d9-8735-6e478f088167",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Concierge Subscription",
    "state": "Enabled",
    "tenantId": "604c1504-c6a3-4080-81aa-b33091104187",
    "user": {
      "name": "ron_lim_ms@outlook.com",
      "type": "user"
    }
  }
]
D:\Terraform\terraform_azure>
```

Go to the location of the desired main.tf and edit the details.

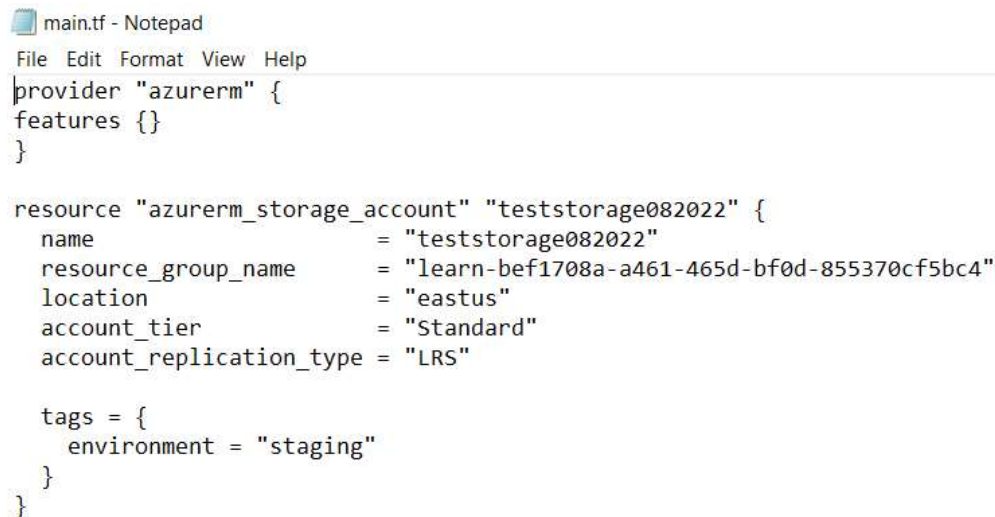
**Note:** Resource Group and Location are very important here. Also, provider block with features is also very important

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
D:\Terraform\terraform_azure\Storage>notepad main.tf  
D:\Terraform\terraform_azure\Storage>
```

Sample main.tf for Azure Storage Creation:



```
main.tf - Notepad  
File Edit Format View Help  
provider "azurerm" {  
  features {}  
}  
  
resource "azurerm_storage_account" "teststorage082022" {  
  name                = "teststorage082022"  
  resource_group_name = "learn-bef1708a-a461-465d-bf0d-855370cf5bc4"  
  location            = "eastus"  
  account_tier        = "Standard"  
  account_replication_type = "LRS"  
  
  tags = {  
    environment = "staging"  
  }  
}
```

After that close the file and initialize the Terraform using command below:

***terraform init***

```
D:\Terraform\terraform_azure\Storage>terraform init  
  
Initializing the backend...  
  
Initializing provider plugins...  
- Finding latest version of hashicorp/azurerm...  
- Installing hashicorp/azurerm v3.20.0...  
- Installed hashicorp/azurerm v3.20.0 (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.  
D:\Terraform\terraform_azure\Storage>
```

(Optional) You can change the main.tf file into terraform-based format by command below:

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

*terraform fmt*

If no output means that the main.tf file is already in terraform format

```
D:\Terraform\terraform_azure\Storage>terraform fmt
D:\Terraform\terraform_azure\Storage>
```

If there is an output, the output mentions the file that was converted to terraform format

```
D:\Terraform\terraform_azure\Storage>terraform fmt
main.tf
D:\Terraform\terraform_azure\Storage>
```

To check and validate the terraform-based codes, execute the command below:

*terraform validate*

If the validation found no errors, then it will output the success message

```
D:\Terraform\terraform_azure\Storage>terraform validate
Success! The configuration is valid.
D:\Terraform\terraform_azure\Storage>
```

(Optional) Check the Terraform execution plan by using the command below:

*terraform plan*

Sample output below:

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
D:\Terraform\terraform_azure\Storage>terraform plan

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:

# azurerm_storage_account.teststorage082022 will be created
+ resource "azurerm_storage_account" "teststorage082022" {
  + access_tier                = (known after apply)
  + account_kind               = "StorageV2"
  + account_replication_type   = "LRS"
  + account_tier               = "Standard"
  + allow_nested_items_to_be_public = true
  + cross_tenant_replication_enabled = true
  + default_to_oauth_authentication = false
  + enable_https_traffic_only      = true
  + id                          = (known after apply)
  + infrastructure_encryption_enabled = false
  + is_hns_enabled              = false
  + large_file_share_enabled      = (known after apply)
  + location                    = "eastus"
  + min_tls_version              = "TLS1_2"
  + name                        = "teststorage082022"
  + nfsv3_enabled               = false
  + primary_access_key           = (sensitive value)
  + primary_blob_connection_string = (sensitive value)
  + primary_blob_endpoint        = (known after apply)
  + primary_blob_host            = (known after apply)
  + primary_connection_string     = (sensitive value)
  + primary_dfs_endpoint         = (known after apply)
  + primary_dfs_host             = (known after apply)
  + primary_file_endpoint        = (known after apply)
  + primary_file_host            = (known after apply)
  + primary_location             = (known after apply)
  + primary_queue_endpoint       = (known after apply)
  + primary_queue_host           = (known after apply)
  + primary_table_endpoint       = (known after apply)
  + primary_table_host           = (known after apply)
  + primary_web_endpoint         = (known after apply)
  + primary_web_host             = (known after apply)
  + queue_encryption_key_type     = "Service"
  + resource_group_name          = "learn-bef1708a-a461-465d-bf0d-855370cf5bc4"
  + secondary_access_key         = (sensitive value)
  + secondary_blob_connection_string = (sensitive value)
  + secondary_blob_endpoint      = (known after apply)
  + secondary_blob_host          = (known after apply)
}
```

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
+ secondary_connection_string = (sensitive value)
+ secondary_dfs_endpoint      = (known after apply)
+ secondary_dfs_host         = (known after apply)
+ secondary_file_endpoint    = (known after apply)
+ secondary_file_host        = (known after apply)
+ secondary_location         = (known after apply)
+ secondary_queue_endpoint   = (known after apply)
+ secondary_queue_host       = (known after apply)
+ secondary_table_endpoint   = (known after apply)
+ secondary_table_host       = (known after apply)
+ secondary_web_endpoint     = (known after apply)
+ secondary_web_host         = (known after apply)
+ shared_access_key_enabled  = true
+ table_encryption_key_type   = "Service"
+ tags                       = {
  + "environment" = "staging"
}

+ blob_properties {
  + change_feed_enabled      = (known after apply)
  + change_feed_retention_in_days = (known after apply)
  + default_service_version  = (known after apply)
  + last_access_time_enabled = (known after apply)
  + versioning_enabled       = (known after apply)

  + container_delete_retention_policy {
    + days = (known after apply)
  }

  + cors_rule {
    + allowed_headers = (known after apply)
    + allowed_methods = (known after apply)
    + allowed_origins = (known after apply)
    + exposed_headers = (known after apply)
    + max_age_in_seconds = (known after apply)
  }

  + delete_retention_policy {
    + days = (known after apply)
  }
}

+ network_rules {
  + bypass          = (known after apply)
  + default_action  = (known after apply)
  + ip_rules        = (known after apply)
  + virtual_network_subnet_ids = (known after apply)

  + private_link_access {
```

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
+ endpoint_resource_id = (known after apply)
+ endpoint_tenant_id   = (known after apply)
}
}

+ queue_properties {
+   cors_rule {
+     allowed_headers = (known after apply)
+     allowed_methods = (known after apply)
+     allowed_origins = (known after apply)
+     exposed_headers = (known after apply)
+     max_age_in_seconds = (known after apply)
+   }

+   hour_metrics {
+     enabled           = (known after apply)
+     include_apis      = (known after apply)
+     retention_policy_days = (known after apply)
+     version           = (known after apply)
+   }

+   logging {
+     delete           = (known after apply)
+     read             = (known after apply)
+     retention_policy_days = (known after apply)
+     version          = (known after apply)
+     write            = (known after apply)
+   }

+   minute_metrics {
+     enabled           = (known after apply)
+     include_apis      = (known after apply)
+     retention_policy_days = (known after apply)
+     version           = (known after apply)
+   }
+ }

+ routing {
+   choice           = (known after apply)
+   publish_internet_endpoints = (known after apply)
+   publish_microsoft_endpoints = (known after apply)
+ }

+ share_properties {
+   cors_rule {
+     allowed_headers = (known after apply)
+     allowed_methods = (known after apply)
+     allowed_origins = (known after apply)
+     exposed_headers = (known after apply)
+     max_age_in_seconds = (known after apply)
+   }

+   retention_policy {
+     days = (known after apply)
+   }

+   smb {
+     authentication_types = (known after apply)
+     channel_encryption_type = (known after apply)
+     kerberos_ticket_encryption_type = (known after apply)
+     versions = (known after apply)
+   }
+ }
}
```

Plan: 1 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.

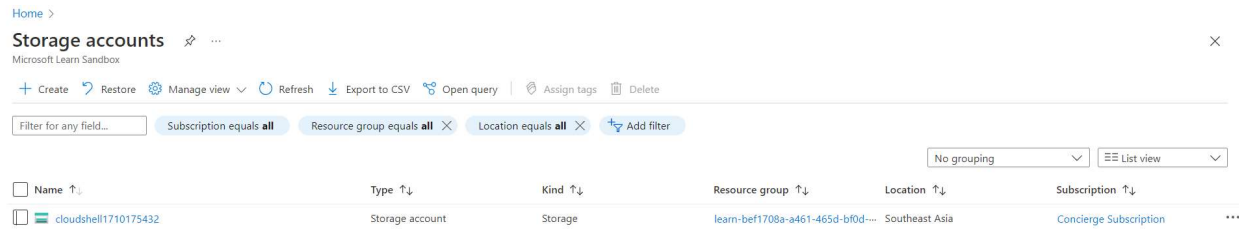
D:\Terraform\terraform\_azure\Storage>

**Before applying changes, check the before state of the Azure Storage. Notice that we only have cloud shell storage**



# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim



Execute the command below and confirm changes to create Azure Storage:

***terraform apply***

**Note:** terraform apply command already contains terraform plan so no need to execute terraform plan

**terraform apply = terraform plan + confirmation to apply changes**

sample output:

```
D:\Terraform\terraform_azure\Storage>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:

# azurerm_storage_account.teststorage082022 will be created
+ resource "azurerm_storage_account" "teststorage082022" {
  + access_tier                = (known after apply)
  + account_kind               = "StorageV2"
  + account_replication_type   = "LRS"
  + account_tier               = "Standard"
  + allow_nested_items_to_be_public = true
  + cross_tenant_replication_enabled = true
  + default_to_oauth_authentication = false
  + enable_https_traffic_only     = true
  + id                         = (known after apply)
  + infrastructure_encryption_enabled = false
  + is_hns_enabled             = false
  + large_file_share_enabled     = (known after apply)
  + location                   = "eastus"
  + min_tls_version            = "TLS1_2"
  + name                       = "teststorage082022"
  + nfsv3_enabled              = false
  + primary_access_key          = (sensitive value)
  + primary_blob_connection_string = (sensitive value)
  + primary_blob_endpoint       = (known after apply)
  + primary_blob_host           = (known after apply)
  + primary_connection_string    = (sensitive value)
  + primary_dfs_endpoint        = (known after apply)
  + primary_dfs_host            = (known after apply)
  + primary_file_endpoint       = (known after apply)
  + primary_file_host           = (known after apply)
  + primary_location            = (known after apply)
  + primary_queue_endpoint      = (known after apply)
  + primary_queue_host          = (known after apply)
  + primary_table_endpoint      = (known after apply)
  + primary_table_host          = (known after apply)
  + primary_web_endpoint        = (known after apply)
  + primary_web_host            = (known after apply)
  + queue_encryption_key_type    = "Service"
  + resource_group_name         = "learn-bef1708a-a461-465d-bf0d-855370cf5bc4"
  + secondary_access_key        = (sensitive value)
  + secondary_blob_connection_string = (sensitive value)
  + secondary_blob_endpoint     = (known after apply)
  + secondary_blob_host         = (known after apply)
}
```



# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
+ secondary_connection_string = (sensitive value)
+ secondary_dfs_endpoint      = (known after apply)
+ secondary_dfs_host          = (known after apply)
+ secondary_file_endpoint     = (known after apply)
+ secondary_file_host         = (known after apply)
+ secondary_location          = (known after apply)
+ secondary_queue_endpoint    = (known after apply)
+ secondary_queue_host        = (known after apply)
+ secondary_table_endpoint    = (known after apply)
+ secondary_table_host        = (known after apply)
+ secondary_web_endpoint      = (known after apply)
+ secondary_web_host          = (known after apply)
+ shared_access_key_enabled   = true
+ table_encryption_key_type    = "Service"
+ tags                        = {
  + "environment" = "staging"
}

+ blob_properties {
  + change_feed_enabled           = (known after apply)
  + change_feed_retention_in_days = (known after apply)
  + default_service_version       = (known after apply)
  + last_access_time_enabled      = (known after apply)
  + versioning_enabled           = (known after apply)

  + container_delete_retention_policy {
    + days = (known after apply)
  }

  + cors_rule {
    + allowed_headers = (known after apply)
    + allowed_methods = (known after apply)
    + allowed_origins = (known after apply)
    + exposed_headers = (known after apply)
    + max_age_in_seconds = (known after apply)
  }

  + delete_retention_policy {
    + days = (known after apply)
  }
}

+ network_rules {
  + bypass           = (known after apply)
  + default_action    = (known after apply)
  + ip_rules          = (known after apply)
  + virtual_network_subnet_ids = (known after apply)
}
```

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
+ private_link_access {
  + endpoint_resource_id = (known after apply)
  + endpoint_tenant_id   = (known after apply)
}

+ queue_properties {
  + cors_rule {
    + allowed_headers   = (known after apply)
    + allowed_methods   = (known after apply)
    + allowed_origins   = (known after apply)
    + exposed_headers   = (known after apply)
    + max_age_in_seconds = (known after apply)
  }

  + hour_metrics {
    + enabled           = (known after apply)
    + include_apis      = (known after apply)
    + retention_policy_days = (known after apply)
    + version           = (known after apply)
  }

  + logging {
    + delete           = (known after apply)
    + read             = (known after apply)
    + retention_policy_days = (known after apply)
    + version          = (known after apply)
    + write            = (known after apply)
  }

  + minute_metrics {
    + enabled           = (known after apply)
    + include_apis      = (known after apply)
    + retention_policy_days = (known after apply)
    + version           = (known after apply)
  }
}

+ routing {
  + choice               = (known after apply)
  + publish_internet_endpoints = (known after apply)
  + publish_microsoft_endpoints = (known after apply)
}

+ share_properties {
  + cors_rule {
    + allowed_headers   = (known after apply)
    + allowed_methods   = (known after apply)
```

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
+ allowed_origins    = (known after apply)
+ exposed_headers    = (known after apply)
+ max_age_in_seconds = (known after apply)
}

+ retention_policy {
+   days = (known after apply)
}

+ smb {
+   authentication_types          = (known after apply)
+   channel_encryption_type       = (known after apply)
+   kerberos_ticket_encryption_type = (known after apply)
+   versions                      = (known after apply)
}
}
```

Plan: 1 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.

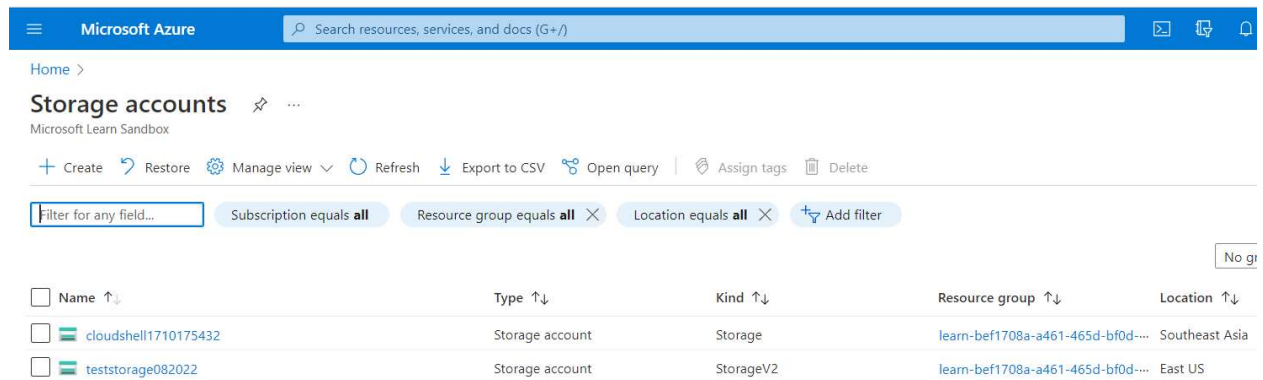
Enter a value: yes

azurerm\_storage\_account.teststorage082022: Creating...  
azurerm\_storage\_account.teststorage082022: Still creating... [10s elapsed]  
azurerm\_storage\_account.teststorage082022: Still creating... [20s elapsed]  
azurerm\_storage\_account.teststorage082022: Still creating... [30s elapsed]  
azurerm\_storage\_account.teststorage082022: Creation complete after 30s [id=/subscriptions/6000e767-b0e8-4b31-b1e1-1e3e3e3e3e3e/storageAccounts/teststorage082022]

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

D:\Terraform\terraform\_azure\Storage>

After applying changes, refresh the web browser and verify that the storage is already created in Azure.



Storage accounts				
Microsoft Learn Sandbox				
Filter for any field...				
Subscription equals all Resource group equals all Location equals all				
Add filter				
No grid				
Name ↑↓ Type ↑↓ Kind ↑↓ Resource group ↑↓ Location ↑↓				
<input type="checkbox"/>	cloudshell1710175432	Storage account	Storage	learn-bef1708a-a461-465d-bf0d-... Southeast Asia
<input type="checkbox"/>	teststorage082022	Storage account	StorageV2	learn-bef1708a-a461-465d-bf0d-... East US

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

The screenshot shows the Azure portal interface for a storage account named 'teststorage082022'. The left sidebar contains navigation links: Overview, Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, and Storage browser. The main area displays the 'Essentials' tab with the following details:

- Resource group (move): learn-bef1708a-a461-465d-bf0d-85537...
- Location: East US
- Subscription (move): Concierge Subscription
- Subscription ID: 6000e767-b051-49d9-8735-6e478f088...
- Disk state: Available
- Tags (edit): environment : staging
- Performance: Standard
- Replication: Locally-redundant storage (LRS)
- Account kind: StorageV2 (general purpose v2)
- Provisioning state: Succeeded
- Created: 8/31/2022, 6:25:09 AM

At the bottom, there are tabs for Properties, Monitoring, Capabilities (7), Recommendations, Tutorials, and Developer Tools.

Then let us try to destroy what we created for learning purposes using command below and confirm:  
***terraform destroy***

sample output:

```
D:\Terraform\terraform_azure\Storage>terraform destroy
azurerm_storage_account.teststorage082022: Refreshing state... [id=/subscriptions/6000e767-b051-49d9-8735-6e478f088167/resourceGroups/learn-bef1708a-a461-465d-bf0d-855370cf5bc4/providers/Microsoft.Storage/storageAccounts/teststorage082022]

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:

# azurerm_storage_account.teststorage082022 will be destroyed
- resource "azurerm_storage_account" "teststorage082022" {
  - access_tier                = "Hot" -> null
  - account_kind               = "StorageV2" -> null
  - account_replication_type   = "LRS" -> null
  - account_tier               = "Standard" -> null
  - allow_nested_items_to_be_public = true -> null
  - cross_tenant_replication_enabled = true -> null
  - default_to_oauth_authentication = false -> null
  - enable_https_traffic_only      = true -> null
  - id                          = "/subscriptions/6000e767-b051-49d9-8735-6e478f088167/resourceGroups/learn-bef1708a-a461-465d-bf0d-855370cf5bc4/providers/Microsoft.Storage/storageAccounts/teststorage082022" -> null
  - infrastructure_encryption_enabled = false -> null
  - is_hns_enabled             = false -> null
  - location                   = "eastus" -> null
  - min_tls_version            = "TLS1_2" -> null
  - name                       = "teststorage082022" -> null
  - nfsv3_enabled              = false -> null
  - primary_access_key         = (sensitive value)
  - primary_blob_connection_string = (sensitive value)
  - primary_blob_endpoint       = "https://teststorage082022.blob.core.windows.net/" -> null
  - primary_blob_host           = "teststorage082022.blob.core.windows.net" -> null
  - primary_connection_string   = (sensitive value)
  - primary_dfs_endpoint        = "https://teststorage082022.dfs.core.windows.net/" -> null
  - primary_dfs_host            = "teststorage082022.dfs.core.windows.net" -> null
  - primary_file_endpoint       = "https://teststorage082022.file.core.windows.net/" -> null
  - primary_file_host           = "teststorage082022.file.core.windows.net" -> null
  - primary_location            = "eastus" -> null
  - primary_queue_endpoint      = "https://teststorage082022.queue.core.windows.net/" -> null
  - primary_queue_host          = "teststorage082022.queue.core.windows.net" -> null
  - primary_table_endpoint      = "https://teststorage082022.table.core.windows.net/" -> null
  - primary_table_host          = "teststorage082022.table.core.windows.net" -> null
  - primary_web_endpoint        = "https://teststorage082022.z13.web.core.windows.net/" -> null
  - primary_web_host            = "teststorage082022.z13.web.core.windows.net" -> null
  - queue_encryption_key_type    = "Service" -> null
  - resource_group_name         = "learn-bef1708a-a461-465d-bf0d-855370cf5bc4" -> null
}
```

# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
- secondary_access_key          = (sensitive value)
- secondary_connection_string    = (sensitive value)
- shared_access_key_enabled      = true -> null
- table_encryption_key_type      = "Service" -> null
- tags                           = {
  - "environment" = "staging"
} -> null

- blob_properties {
  - change_feed_enabled          = false -> null
  - change_feed_retention_in_days = 0 -> null
  - last_access_time_enabled     = false -> null
  - versioning_enabled           = false -> null
}

- network_rules {
  - bypass                      = [
    - "AzureServices",
  ] -> null
  - default_action              = "Allow" -> null
  - ip_rules                    = [] -> null
  - virtual_network_subnet_ids = [] -> null
}

- queue_properties {
  - hour_metrics {
    - enabled          = true -> null
    - include_apis     = true -> null
    - retention_policy_days = 7 -> null
    - version           = "1.0" -> null
  }

  - logging {
    - delete          = false -> null
    - read            = false -> null
    - retention_policy_days = 0 -> null
    - version          = "1.0" -> null
    - write            = false -> null
  }

  - minute_metrics {
    - enabled          = false -> null
    - include_apis     = false -> null
    - retention_policy_days = 0 -> null
    - version           = "1.0" -> null
  }
}
```



# Terraform Azure Sample: Azure Storage

By Ronald Stewart Lim

```
- share_properties {
  - retention_policy {
    - days = 7 -> 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: yes

azurerm\_storage\_account.teststorage082022: Destroying... [id=/subscriptions/6000e767-b051-49d9-8735-6e478f088167/resourceGroups/learn-teststorage082022]  
azurerm\_storage\_account.teststorage082022: Destruction complete after 7s

Destroy complete! Resources: 1 destroyed.

D:\Terraform\terraform\_azure\Storage>

After applying changes, refresh the web browser and verify that the storage is already removed or deleted in Azure.

Microsoft Azure

Search resources, services, and docs (G+)

Home >

## Storage accounts

Microsoft Learn Sandbox

+ Create ↺ Restore ⚙ Manage view ▾ ↻ Refresh ⬇ Export to CSV 🔗 Open query | 🏷 Assign tags 🗑 Delete

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

No groups

<input type="checkbox"/> Name ↑↓	Type ↑↓	Kind ↑↓	Resource group ↑↓	Location ↑↓
<input type="checkbox"/> cloudshell1710175432	Storage account	Storage	learn-bef1708a-a461-465d-bf0d-...	Southeast Asia