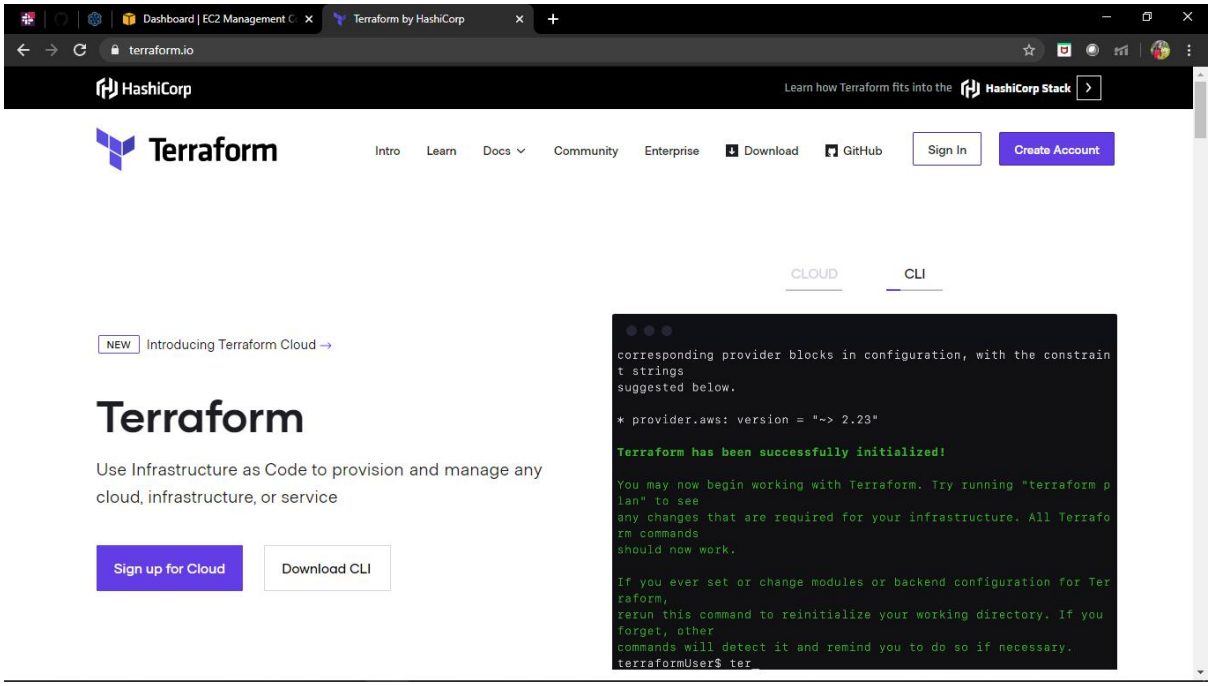
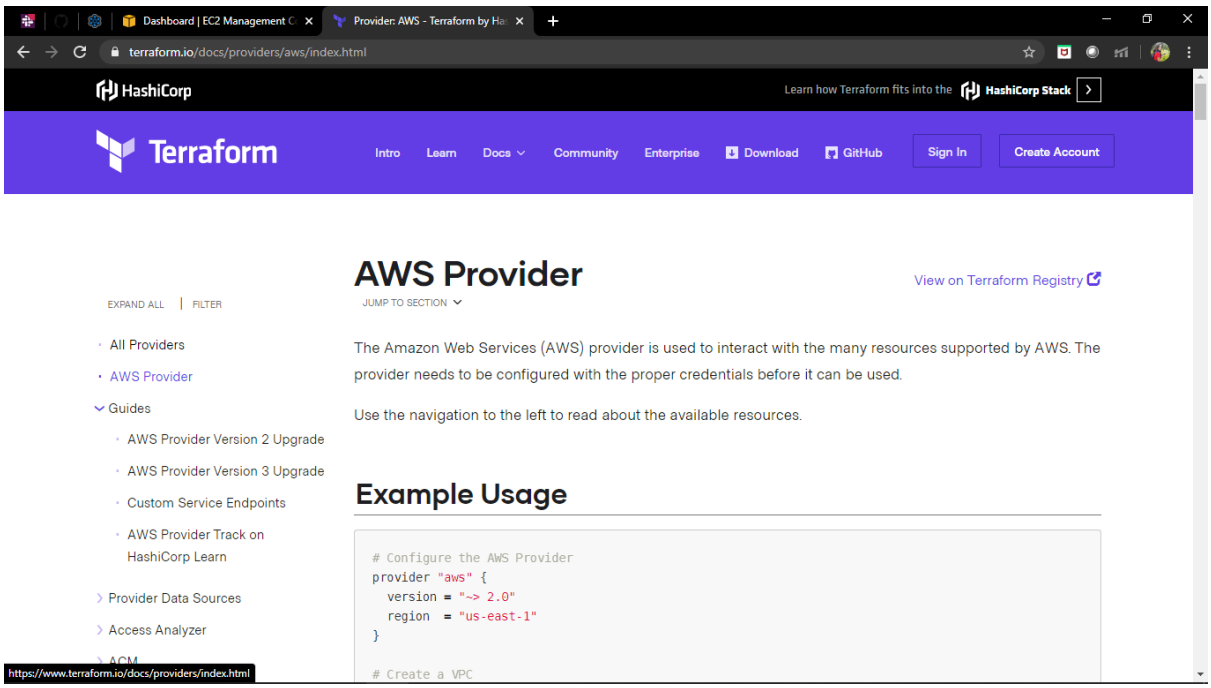
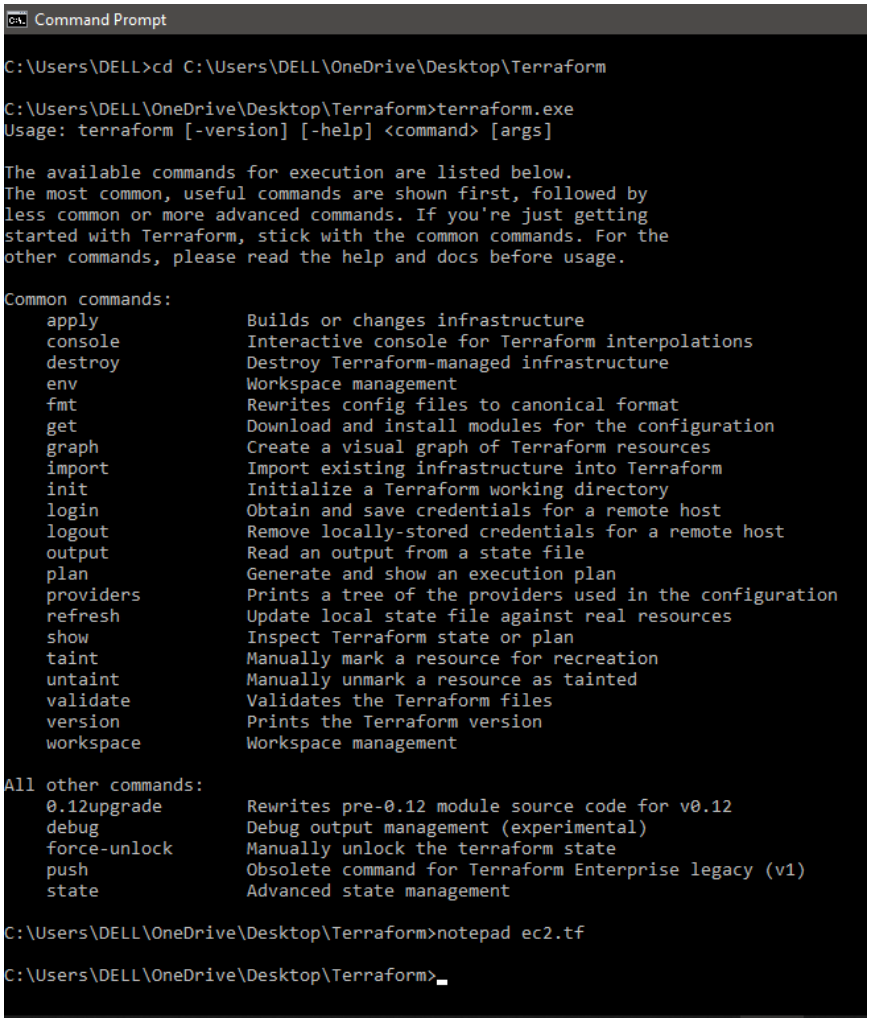
Hybrid Multi Cloud Day 8 : 9th June 2020

Terraform

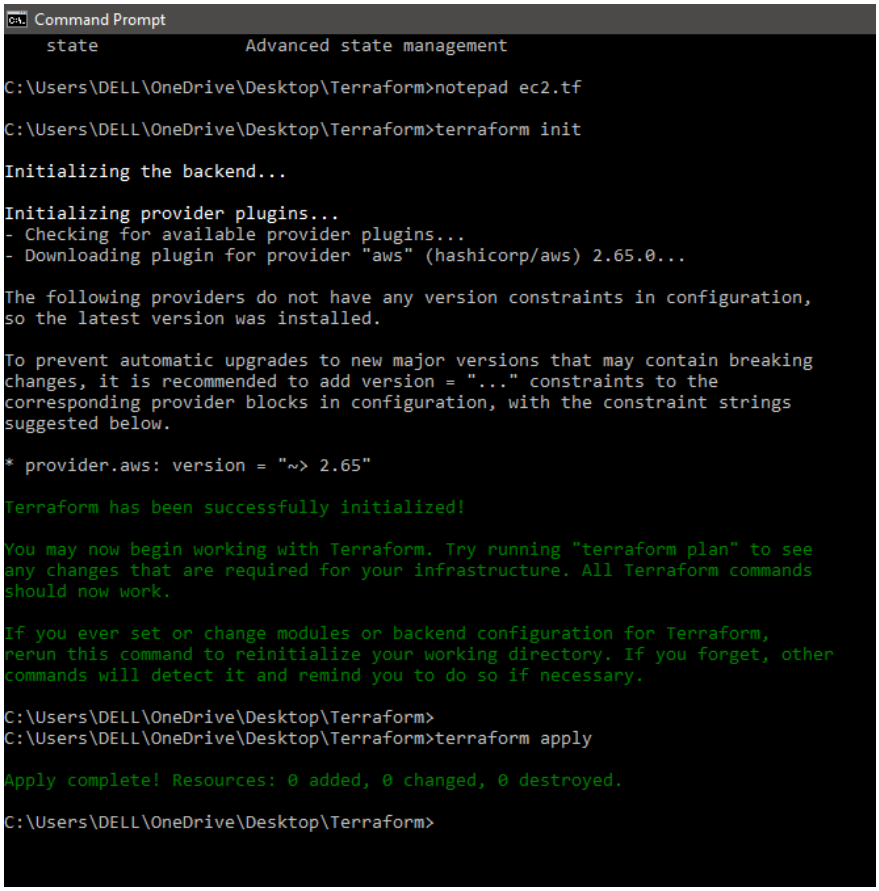
Use Infrastructure as Code to provision and manage any cloud, infrastructure, or service.

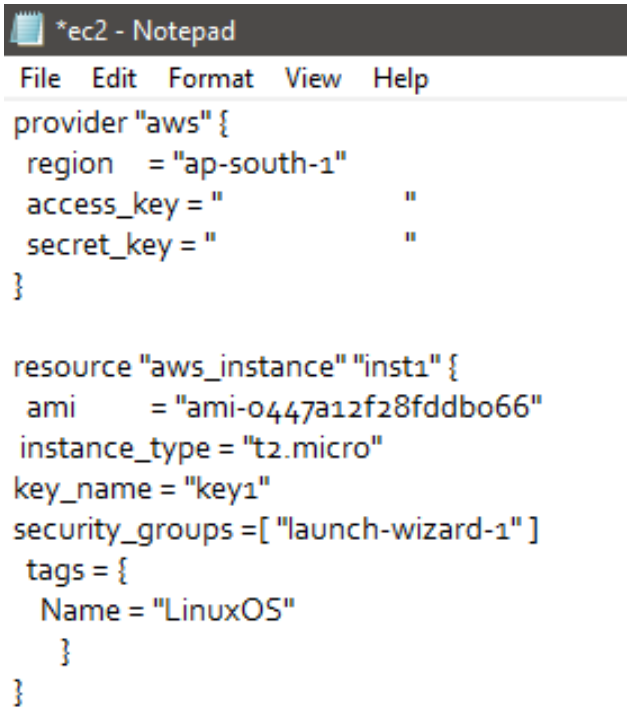





Command Prompt
C:\Users\DELL>cd C:\Users\DELL\OneDrive\Desktop\Terraform
C:\Users\DELL\OneDrive\Desktop\Terraform>terraform.exe
Usage: terraform [-version] [-help] <command> [args]
The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.
Common commands:
apply Builds or changes infrastructure
console Interactive console for Terraform interpolations
destroy Destroy Terraform-managed infrastructure
env Workspace management
fmt Rewrites config files to canonical format
get Download and install modules for the configuration
graph Create a visual graph of Terraform resources
import Import existing infrastructure into Terraform
init Initialize a Terraform working directory
login Obtain and save credentials for a remote host
logout Remove locally-stored credentials for a remote host
output Read an output from a state file
plan Generate and show an execution plan
providers Prints a tree of the providers used in the configuration
refresh Update local state file against real resources
show Inspect Terraform state or plan
taint Manually mark a resource for recreation
untaint Manually unmark a resource as tainted
validate Validates the Terraform files
version Prints the Terraform version
workspace Workspace management
All other commands:
0.12upgrade Rewrites pre-0.12 module source code for v0.12
debug Debug output management (experimental)
force-unlock Manually unlock the terraform state
push Obsolete command for Terraform Enterprise legacy (v1)
state Advanced state management
C:\Users\DELL\OneDrive\Desktop\Terraform>notepad ec2.tf
C:\Users\DELL\OneDrive\Desktop\Terraform>

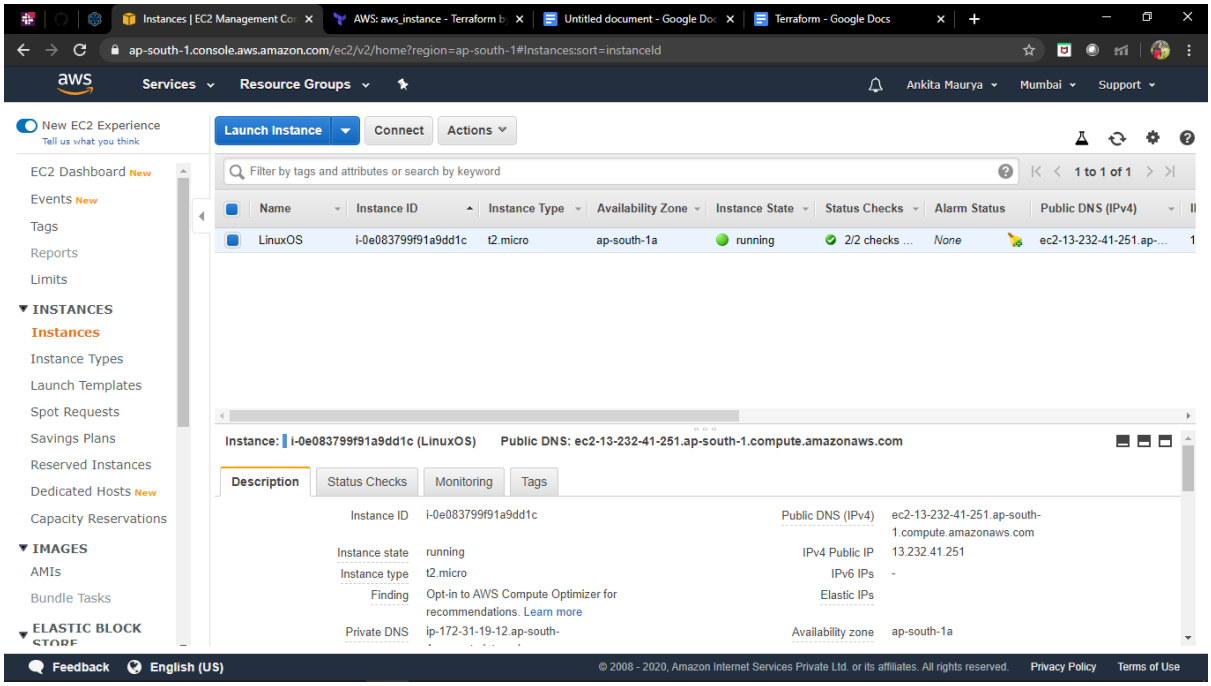

*ec2 - Notepad
File Edit Format View Help
provider "aws" {
region = "ap-south-1"
access_key = "
secret_key = "
}


Command Prompt
state Advanced state management
C:\Users\DELL\OneDrive\Desktop\Terraform>notepad ec2.tf
C:\Users\DELL\OneDrive\Desktop\Terraform>terraform init
Initializing the backend...
Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (hashicorp/aws) 2.65.0...
The following providers do not have any version constraints in configuration,
so the latest version was installed.
To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.
* provider.aws: version = "~> 2.65"
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:\Users\DELL\OneDrive\Desktop\Terraform>
C:\Users\DELL\OneDrive\Desktop\Terraform>terraform apply
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
C:\Users\DELL\OneDrive\Desktop\Terraform>


*ec2 - Notepad
File Edit Format View Help
provider "aws" {
region = "ap-south-1"
access_key = "
"
secret_key = "
"
}
resource "aws_instance" "inst1" {
ami = "ami-0447a12f28fddb066"
instance_type = "t2.micro"
key_name = "key1"
security_groups =["launch-wizard-1"]
tags = {
Name = "LinuxOS"
}
}

```
Command Prompt

C:\Users\DELL\OneDrive\Desktop\Terraform>terraform apply

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.inst1 will be created
  + resource "aws_instance" "inst1" {
      + ami                          = "ami-0447a12f28fddb066"
      + 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)
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (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                     = "key1.pem"
      + network_interface_id         = (known after apply)
      + outpost_arn                  = (known after apply)
      + password_data                = (known after apply)
      + placement_group              = (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)
      + security_groups              = [
          + "launch-wizard-1",
        ]
      + source_dest_check            = true
      + subnet_id                    = (known after apply)
      + tags                         = {
          + "Name" = "LinuxOS"
        }
      + tenancy                      = (known after apply)
      + volume_tags                  = (known after apply)
      + vpc_security_group_ids       = (known after apply)
```

```
Command Prompt
        }

      + ephemeral_block_device {
          + device_name  = (known after apply)
          + no_device    = (known after apply)
          + virtual_name = (known after apply)
        }

      + metadata_options {
          + http_endpoint               = (known after apply)
          + http_put_response_hop_limit = (known after apply)
          + 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)
          + 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.

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.inst1: Creating...
aws_instance.inst1: Still creating... [10s elapsed]
aws_instance.inst1: Still creating... [20s elapsed]
aws_instance.inst1: Creation complete after 23s [id=i-0e083799f91a9dd1c]

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




```
Command Prompt

C:\Users\DELL\OneDrive\Desktop\Terraform>terraform destroy
aws_instance.inst1: Refreshing state... [id=i-0e083799f91a9dd1c]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  - destroy

Terraform will perform the following actions:

  # aws_instance.inst1 will be destroyed
  - resource "aws_instance" "inst1" {
      - ami                          = "ami-0447a12f28fddb066" -> null
      - arn                          = "arn:aws:ec2:ap-south-1:220287739124:instance/i-0e083799f91a9dd1c" -> null
      - associate_public_ip_address  = true -> null
      - availability_zone            = "ap-south-1a" -> null
      - cpu_core_count               = 1 -> null
      - cpu_threads_per_core         = 1 -> null
      - disable_api_termination      = false -> null
      - ebs_optimized                = false -> null
      - get_password_data            = false -> null
      - hibernation                  = false -> null
      - id                           = "i-0e083799f91a9dd1c" -> null
      - instance_state               = "running" -> null
      - instance_type                = "t2.micro" -> null
      - ipv6_address_count           = 0 -> null
      - ipv6_addresses               = [] -> null
      - key_name                     = "key1" -> null
      - monitoring                   = false -> null
      - primary_network_interface_id = "eni-061f963d2e13c0a7a" -> null
      - private_dns                  = "ip-172-31-19-12.ap-south-1.compute.internal" -> null
      - private_ip                   = "172.31.19.12" -> null
      - public_dns                   = "ec2-13-232-41-251.ap-south-1.compute.amazonaws.com" -> null
      - public_ip                    = "13.232.41.251" -> null
      - security_groups              = [
          - "launch-wizard-1",
        ] -> null
      - source_dest_check            = true -> null
      - subnet_id                    = "subnet-aa95d4c2" -> null
      - tags                         = {
          - "Name" = "LinuxOS"
        } -> null
      - tenancy                      = "default" -> null
      - volume_tags                  = {} -> null
      - vpc_security_group_ids       = [
          - "sg-00f7d866857c79aa9",
```

Command Prompt

```
      - source_dest_check           = true -> null
      - subnet_id                   = "subnet-aa95d4c2" -> null
      - tags                        = {
          - "Name" = "LinuxOS"
        } -> null
      - tenancy                     = "default" -> null
      - volume_tags                 = {} -> null
      - vpc_security_group_ids      = [
          - "sg-00f7d866857c79aa9",
        ] -> null

      - credit_specification {
          - cpu_credits = "standard" -> null
        }

      - metadata_options {
          - http_endpoint               = "enabled" -> null
          - http_put_response_hop_limit = 1 -> null
          - http_tokens                 = "optional" -> null
        }

      - root_block_device {
          - delete_on_termination = true -> null
          - device_name           = "/dev/xvda" -> null
          - encrypted             = false -> null
          - iops                  = 100 -> null
          - volume_id             = "vol-0aeded138ca494275" -> null
          - volume_size           = 8 -> null
          - volume_type           = "gp2" -> 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

aws_instance.inst1: Destroying... [id=i-0e083799f91a9dd1c]
aws_instance.inst1: Still destroying... [id=i-0e083799f91a9dd1c, 10s elapsed]
aws_instance.inst1: Still destroying... [id=i-0e083799f91a9dd1c, 20s elapsed]
aws_instance.inst1: Destruction complete after 20s

Destroy complete! Resources: 1 destroyed.
```