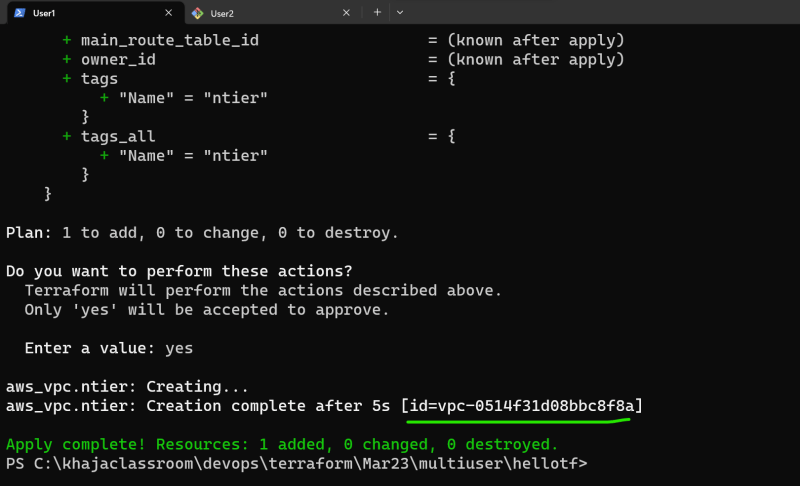
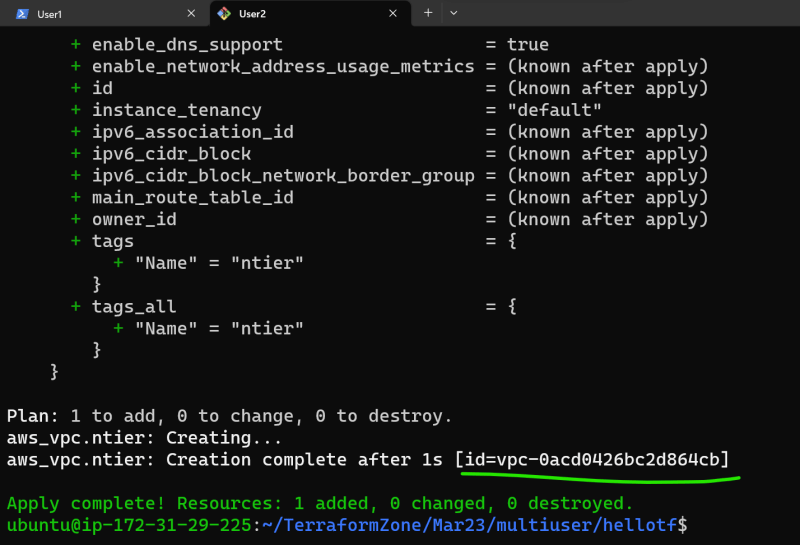
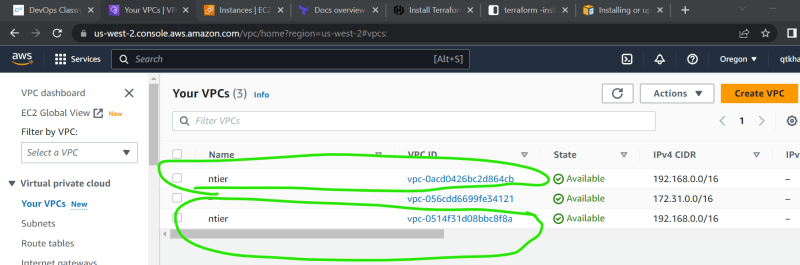
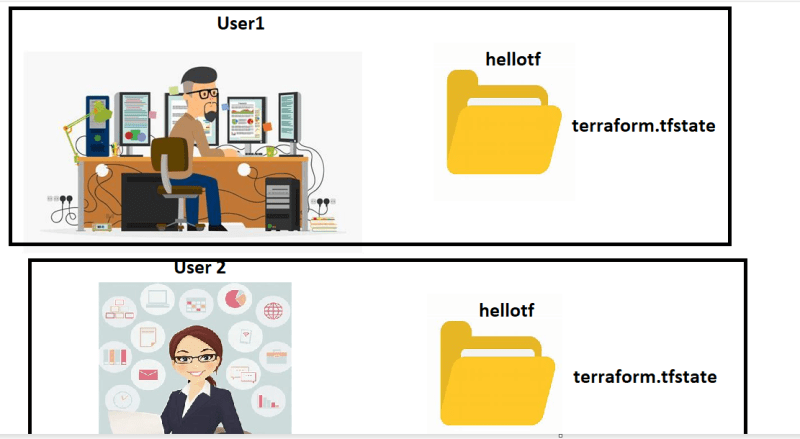
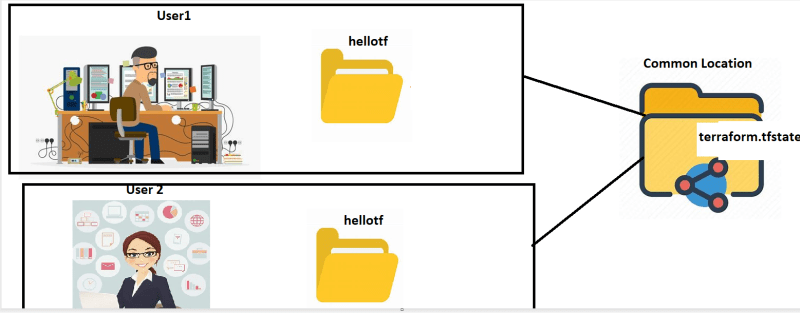
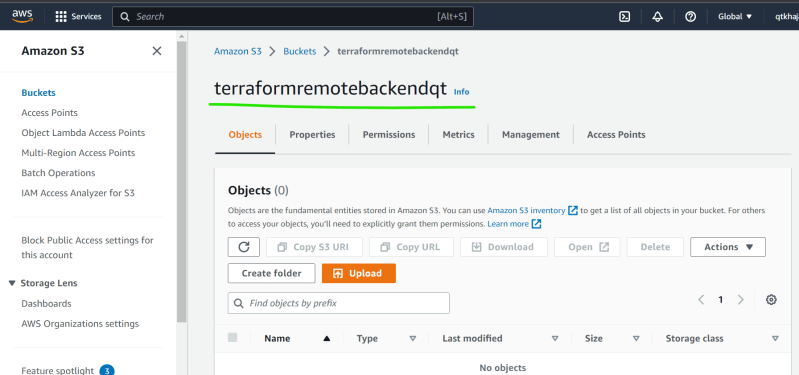
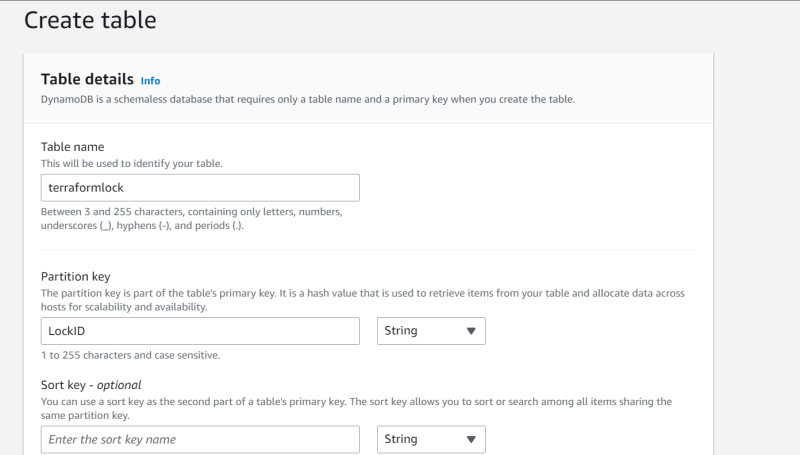
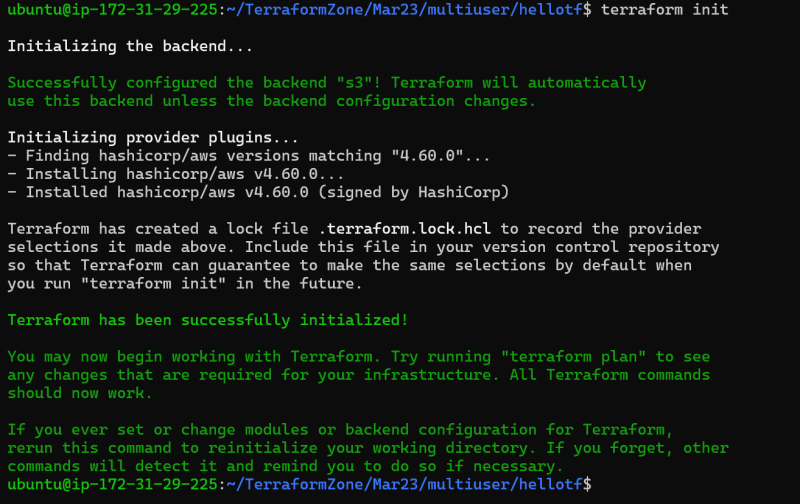
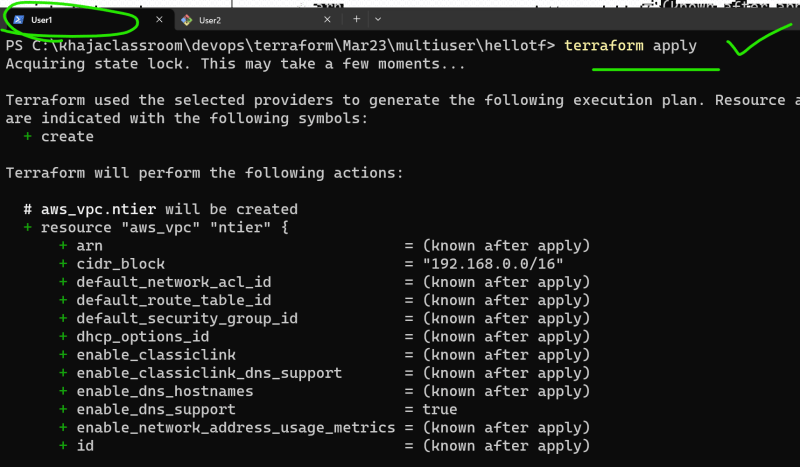
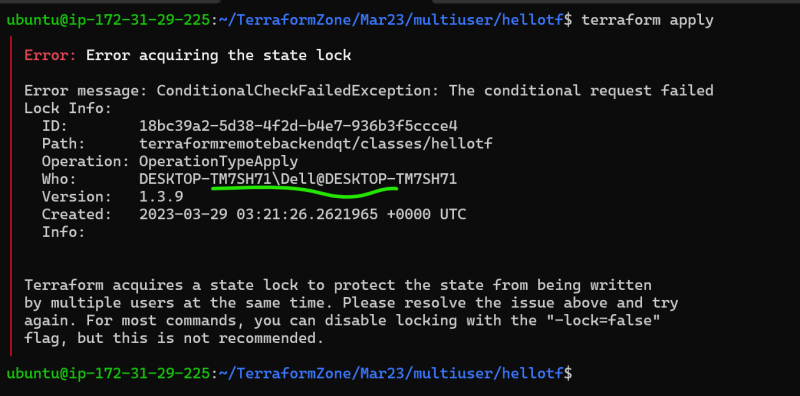
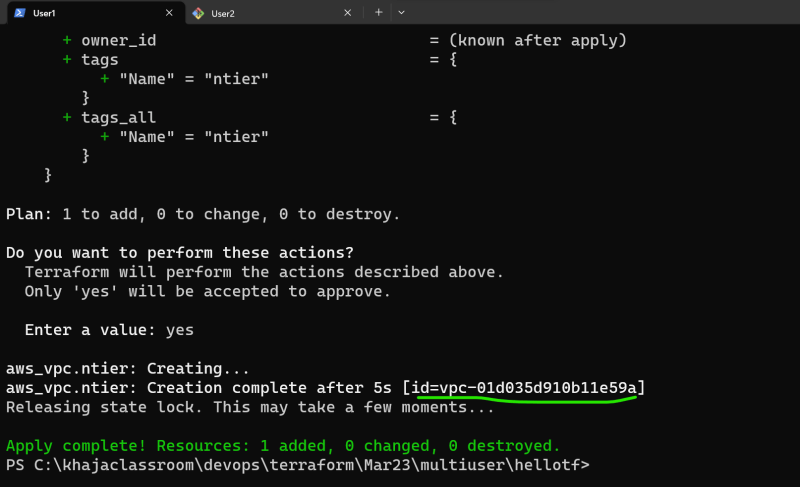
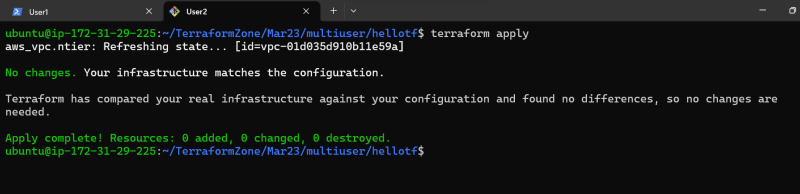
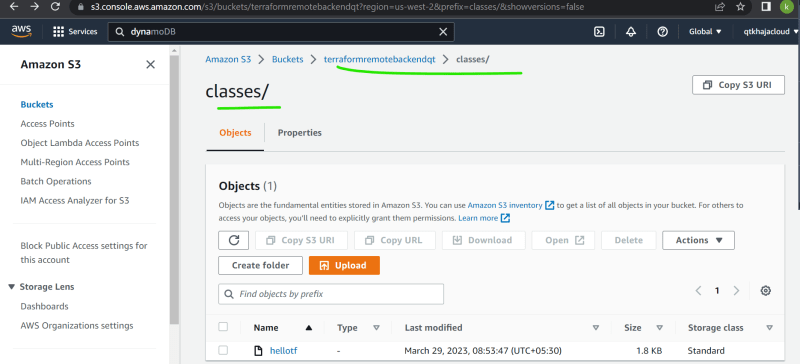
**Multi user setup in terraform**

* Let’s create a simple terraform template to create vpc <https://github.com/asquarezone/TerraformZone/commit/0ebffe06aea8fbc28f8f99fd634e8795c1019fe7> for the changes
* Execute this from two different machines.
* Consider both of them are working for same purpose
* User 1:  
  
* User 2:  
  
* Execution Create two different resources, which is not desired  
  
* As of now state is stored locally i.e., when user1 executes it is stored in user1 system and same for user2  
  
* To solve this problem, we need to store the state in some common place  
  
* The location of state file in terraform is defined by backend.

**Terraform Backends**

* Backend defines where the state has to be stored
* <https://developer.hashicorp.com/terraform/language/state/backends> for official docs and <https://developer.hashicorp.com/terraform/language/settings/backends/configuration> for configuring backend
* There are two types of backends
  + local-backend:
    - This is default backend
  + remote-backend
* <https://developer.hashicorp.com/terraform/language/settings/backends/configuration#available-backends> for available backends
* As common state for terraform for multiple users will have concurrency problem, Terraform backends need locking and unlocking
* S3 bucket can be used as terraform backend, S3 buckend doesnot support locking, if you need locking add dynamo db details

**Remote Backends with S3**

* [Refer Here](https://developer.hashicorp.com/terraform/language/settings/backends/s3) for official docs
* Create an s3 bucket  
  
* Create a dynamo DB table with any name and partition key LockID  
  
* <https://github.com/asquarezone/TerraformZone/commit/84b6692c43a357dea049b69599993ef62b73f63e> for the changes done to add s3 backend
* Perform init on both user machines  
  
* Now lets user1 apply the changes  
  
* Now while user1 is still applying let user 2 also apply  
  
* Let user1 finish applying and create the resources  
  
* Now let user2 try applying  
    
  
* Exercise: Configure Azurerm backend <https://developer.hashicorp.com/terraform/language/settings/backends/azurerm> which has inbuilt locking facility

Summary:

1. How to avoid duplicate resource creation: use statefile in remote backend.
2. Destroy delete all things from statefile as well.
3. When we are creating user it must have read/write access to S3 and DynmoDB.