**Understand terraform refresh in detail**

* This commands comes under **Terraform Inspecting State**
* Understanding terraform refresh clears a lot of doubts in our mind and terraform state file and state feature
* The terraform refresh command is used to reconcile the state Terraform knows about (via its state file) with the real-world infrastructure.
* This can be used to detect any drift from the last-known state, and to update the state file.
* This does not modify infrastructure, but does modify the state file. If the state is changed, this may cause changes to occur during the next plan or apply.
* **terraform refresh:** Update local state file against real resources in cloud
* **Desired State:** Local Terraform Manifest (All \*.tf files)
* **Current State:** Real Resources present in your cloud
* **Command Order of Execution:** refresh, plan, make a decision, apply

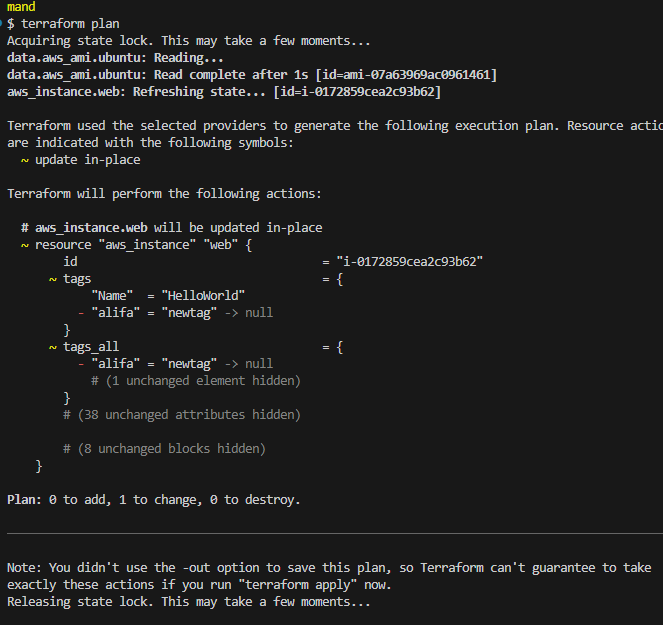
Current state= the actual state of the infrastructure

Desired state= the state according to the terraform configuration as declared in the state file.

Here we made an EC2 instance where the state file is stored in s3 and lock file in dynamodb.

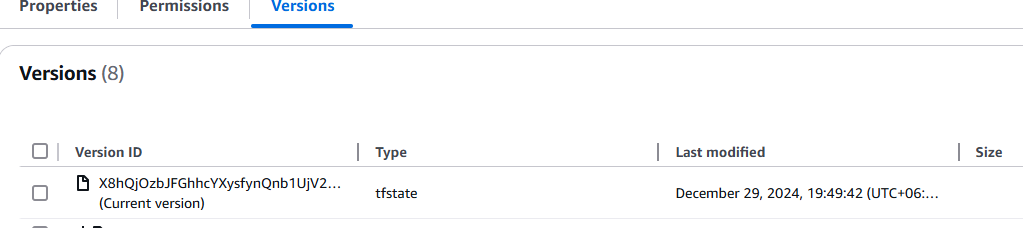
We applied terraform plan and terraform apply. Then using aws console we made changes to the ec2 by adding an extra tag called “alifa = newtag”

When we applied for terraform plan,



We observe that it refreshed the state file as a new tag was added.. 1 change is there. But there was no new state file created as we only did terraform plan, so it got refreshed, as we know in terraform plan its temporary.

**There is a update in place**.

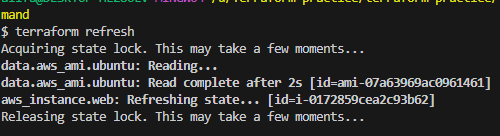


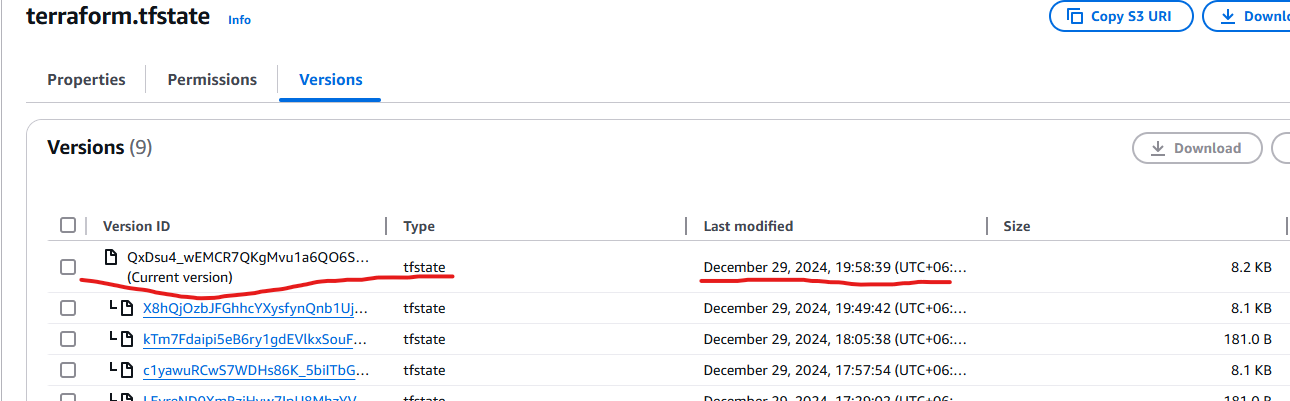
No new version of the state. It only runs in memory as it was refreshed but not in the memory.

Now we applied

terraform refresh

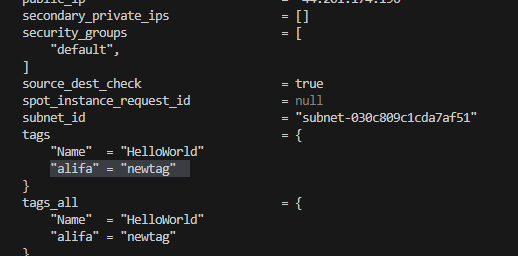
this will update the terraform state file.





New version of the tfstate file was created. In this statefile, the new tag that we created was also added.

terraform show



**Why you need to the execution in this order (refresh, plan, make a decision, apply) ?**

* There are changes happened in your infra manually and not via terraform.
* Now decision to be made if you want those changes or not.
* **Choice-1:** If you dont want those changes proceed with terraform apply so manual changes we have done on our cloud EC2 Instance will be removed.
* **Choice-2:** If you want those changes, refer terraform.tfstate file about changes and embed them in your terraform manifests (example: c4-ec2-instance.tf) and proceed with flow (referesh, plan, review execution plan and apply)

If we choose option 1 then we just apply terraform plan and apply then the tag will be dropped and changes will be removed.

If we choose option 2, then in code we make the changes like adding the tag name and value in the code. And then write terraform plan to see no changes.

