Terraform-Akash

Create-before-destroy lifecycle

step1

Create a main.tf file

```
resource "aws_instance" "webserver" {
ami = "ami-0c02fb55956c7d316"
instance_type = "t2.medium"
tags = {
Name = "lifecycle-update"
}
lifecycle {
create_before_destroy = true
}
}
```

Run terraform apply

step2

Now lets made some update in this code so we can understand lifecycle

Because create_before_destroy will create a new instance before destroy older one

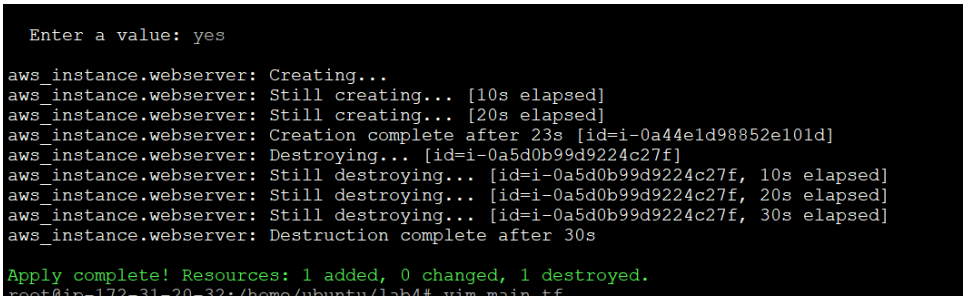
```
resource "aws_instance" "webserver" {
ami = "ami-0c02fb55956c7d316"
instance_type = "t2.medium"
key_name = "23MarchAfternoon"
tags = {
Name = "lifecycle-update"
}
lifecycle {
create_before_destroy = true
}
```

}

Again run terraform apply

It will create new instance and then destroy older one

Output for crreate-before-destroy



2 ignore_changes-

Create a another folder and in this create a main.tf

Step 1

Add in main.tf

```
resource "aws_instance" "webserver" {
 ami           = "ami-0c02fb55956c7d316"
 instance_type = "t2.micro"
 tags = {
   Name = "webA"
 }
lifecycle {
   ignore_changes = [
     tags,
   ]
 }
}
```

Run terraform apply

Step 2

Change the tag name from webA to webB and run terraform apply cmd

It will not change anything

```
root@ip-172-31-20-32:/home/ubuntu/lab5# terraform apply
aws_instance.webserver: Refreshing state... [id=i-03602f7d519aca176]

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

Then remove lifecycle policy

And run terraform apply

```
  Enter a value: yes

aws_instance.webserver: Modifying... [id=i-03602f7d519aca176]
aws_instance.webserver: Modifications complete after 1s [id=i-03602f7d519aca176]

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

3.

Prevent-destroy-

Step 1.

As name suggests, it prevent destroy of instances

Create a main.tf and run apply

```
resource "aws_instance" "webserver" {
 ami           = "ami-0c02fb55956c7d316"
 instance_type = "t2.micro"
 tags = {
   Name = "webserverB"
 }
lifecycle {
prevent_destroy=true
}
}
```

Step 2

Try to run

Terraform destroy

It will throw a error msg

```
root@ip-172-31-20-32:/home/ubuntu/lab5# terraform destroy
aws_instance.webserver: Refreshing state... [id=i-03602f7d519aca176]

Error: Instance cannot be destroyed

  on main.tf line 1:
   1: resource "aws_instance" "webserver" {

Resource aws_instance.webserver has lifecycle.prevent_destroy set, but the
plan calls for this resource to be destroyed. To avoid this error and continue
with the plan, either disable lifecycle.prevent_destroy or reduce the scope of
the plan using the -target flag.
```