**Update RDS db instance time zone to Asia/Kolkata**

* This will create new RDS Database Parameter Group
* Update Database Parameter group of existing RDS instance.
* Then reboot the instance

Terraform keep track of the previous state of deployment and applies incremental changes, basically maintains state of the environment.

But in our case, environment was not created using terraform so it won’t have the record. That’s why we need to import our environment (aws resources – rds db).

Also after importing env or doing any changes in it, terraform stores state of the environment in “terraform.tfstate” file.

So before applying same code to another aws account (that means another environment), we need to delete previous state stored in “terraform.tfstate” file. So we need to delete that file.

**STEPS :**

1. del terraform.tfstate
2. del terraform.tfstate.backup
3. aws configure
4. terraform init
5. terraform import aws\_db\_instance.db\_instance <DB Instance Identifier>
6. terraform apply

As per my knowledge, we don’t have any direct way in terraform to reboot instances. That’s why I’ve used "null\_resource" to execute aws cli command to reboot the rds database instance.

Since we are using aws cli command in terraform code, we need to configure aws on cli -> “aws configure”

Here I’ve set trigger so that it will only reboot the instance when parameter group of database is modified.

Also it will wait until database instance becomes available.