1. **Explain end to end flow of terraform.**

[install → init → HCL configuration → validate → plan → apply → manage(optional) → destroy]

you have to explain each and every step starting from init to destroy.

2. **project requirement is to re-create a degraded resource next apply, what will do on terraform to achieve this?**

Ans. We need to mark the resource as tainted by taint command.

First run $ terraform state list –get the resource name

$ terraform taint <resource name>

Terraform marks the resource as my not be functional due to some issue. Nothing changes on infrastructure but terraform state file updates with this information. Terraform plan will show the plan in that, we can see terraform will show destroy and recreate on that particular resource. Then terraform apply will recreate that resource.

3. **we deployed resources through terraform, but someone modified the resource properties in azure portal which are not reflected in terraform state file, by which command you are going to update this change terraform state file?**

Ans. When someone changed a terraform deployed resource through azure portal, the change will not reflect in terraform state file. To reflect these in terraform state file we need to use below command to update the state file.

$ terraform refresh

4. **what is state file and how do you make it available to other developers in the team.**

Ans. Terraform stores information of infrastructure which are crated using configuration in terraform state file and keeps track of these details.

By using Terraform backends make it available to other developers in the team.

5. **what are meta-arguments in terraform, explain with example?**

Ans. Meta arguments in terraform allows you to customize the default behaviour of terraform.

Meta arguments:

a. Count: allows you to crate similar resources with same resource block
b. For-each: allows you to create similar resources with same resource block with key value pairs.
c. Depends-on: handles hidden resources of module dependencies that terraform cannot automatically infer.
d. Life cycle policy: create before destroy, prevent destroy, ignore changes
e. Provider: crate multiple provider aliases to create resources in different region.

6. **who creates the rerraform.tfstate.backup file and which scenario its created.**

Ans. Terraform creates the terraform.tfstate.backup file when terraform destroy command performed. $ terraform destroy