

SECTION 1: GIT

1. If you using git stash, where will it save data? What is diff b/w index and staging area?

When using git stash, the data is saved in a hidden directory called .git in the root of the project directory. The data is saved in a file called refs/stash, which is a reference to the latest stash.

In git , index is a file which contains all the files that git knows about. The index is like a pre-commit, it holds a snapshot of the working tree content.

The staging area is a set of files that have been added to the index, but not committed. The staging area is where we choose what we want to go to our next commit.

2. When would individuals use git rebase, git fast-forward, or a git fetch then push?

We use git rebase to polish a feature branch before merging it into the main code base. It allows us to easily change a series of commits, modifying the history of our repository. We would use rebase to edit previous commit messages.

Git fast-forward, when we merge a branch that is ahead of our checked-out branch. It literally move our main branch's tip forward to the end of our feature branch, that keeps all commits created in feature branch sequential.

We use git fetch to download contents from a remote repository.

Git push, to move local changes to Central repo

3. How to revert already pushed changes?

Using git revert, git revert <unwanted_commit_hash>

4. What is the difference between cherry picking commits vs trying a hard reset. What is the final outcome of the head reference?

If we want to replay exactly one of those commits into our repository, we use git cherry-pick.

If we want to restore the project's history, we use git reset --hard

5. Explain the difference between git remote and git clone?

Git remote is used to refer to a remote repository or central repository. For example, when we want to add a remote repo as our origin, we use this command. 'git remote add' just creates an entry in git config that specifies a name for particular URL

Git clone is used to copy or clone a remote repo locally. It creates a new git repo by copying an existing one located at the given URL

SECTION 2: TERRAFORM

1. What is the difference between terraform count and for_each meta data function? and give a scenario-based example to use them?

When we define a resource block in Terraform, by default, only one resource element will be created. To manage multiple resource creation, we can use either count or for_each.

Count is an older version of for_each to create multiple instances of the same resource.

For example,

```
resource "aws_instance" "my-instance" {  
  
    ami = "ami-id"  
  
    instance_type = "type-of-instance"  
  
    key_name = "pem-file-name"  
  
    count = 5  
  
    tags = {  
  
        Name = "Terraform-Instance ${count.index}"  
  
    }  
  
}
```

For_Each, it uses key as map of the index i.e instead of specifying the number of resources, for_each meta-argument accepts a map or a set of strings.

For example,

```
resource "aws_instance" "my-instance" {  
  
    ami = "ami-id"  
  
    instance_type = "type-of-instance"  
  
    key_name = "pem-file-name"  
  
    for_each = toset (var.my-instance)  
  
    tags {  
  
        Name = each.value  
  
    }  
  
}
```

2. What is Terraform taint ? When to use it? When would you use terraform state rm vs terraform taint?

Terraform taint is used to manually mark a terraform managed resource as tainted. Once a resource is marked as tainted, it will be destroyed or recreated on next apply.

terraform taint <resource-type>.resource_name → To taint a resource

We use terraform state rm to remove a binding to an existing object without first destroying it.

3. How would you show a diagram of all terraform resources in the state file? When is this useful?

Terraform stores the information about our infrastructure in a state file. This state file keeps the track of resources created by our configuration and Maps them to real-world resources.

If something has been changed manually in the cloud or to the infrastructure, Terraform has the capability to identify these changes and re-configure to the original point by comparing it to the statefile.

4. Solve this expression:

```
count           = var.run_remote_environment ? var.TFC_RUN_ID  
!=["Yes"]) : null
```

In the above expression,

Var.TFC_RUN_ID != "Yes" is the condition we want to check. If it evaluates to true i.e., var.TFC_RUN_ID is not equal to "Yes", then count is set to 1, means the resource will be created once.

If the condition is false, count is set to 0, means the resource will not be created.

5. How would you apply terraform to multiple accounts simultaneously? We want to ensure this follows security best practices.

Pre-requisites: AWS account with necessary permissions. Terraform installed on local machine.

1. We need to set up multiple AWS accounts. Each account must have a unique email address. After setting up the accounts, creating an IAM user in each account with programmatic access.
2. Then, we configure the AWS credentials for each account on our local machine.
3. We create the Terraform configuration files, separate file for each AWS account.
4. We can deploy our infrastructure and navigate to the directory containing our Terraform files and run the commands like terraform init, terraform plan and terraform apply

Deploying to multiple accounts with Terraform, this approach provides the best way to manage our cloud resources, enhancing security and facilitates resource management.

SECTION 3: AWS

- 1. You have an EC2 instance that has an unencrypted volume. You want to create another Encrypted volume from this unencrypted volume. Which of the following steps can achieve this? How would you share this encrypted volume to another account? What must you ensure to make sure this cross-account encryption is shared?**

To create encrypted volume from unencrypted volume,

1. We select our unencrypted volume
2. Select Actions, then 'Create Snapshot'
3. After snapshot is complete, we select 'Snapshots' under 'Elastic Block Store'. Then we select newly created snapshot
4. Select Actions, then 'Copy'
5. We check the box for 'Encryption'
6. We select the CMK for KMS to use as required
7. We click 'Copy'
8. Select the newly created snapshot
9. Select Actions, then 'Create Volume'
10. We will notice that the normal Encryption option is set to 'True' since the snapshot itself encrypted and this cannot be modified. The volume thus created from this snapshot will be encrypted

To share encrypted volume to another account,

1. In the source account, we create an EBS snapshot of the EBS volume
2. In the source account, we share the snapshot with the target account
3. When the snapshot is encrypted with a customer managed key, we share the customer managed key with the target account from source account
4. In the target account, we create a copy of the shared snapshot and encrypt the snapshot copy. We have to make sure to select our customer managed key. Otherwise, EBS encryption uses the default key
5. In the target account, we create a new volume from the copy of the shared snapshot

- 2. How will you implement service control policy and in which area are you using it?**

We create SCPs by building statements that deny or allow access to services and actions that we specify. The default configuration for working with SCPs is to use a 'block list' strategy

We sign in to the AWS organization console, we sign in as IAM user, assume an IAM role in the management account. On the SCP page, we choose Create Policy, On the Create new SCP page, we enter a Policy name and description, if needed. We add one or more tags, then entering a key and optional value.

3. How can you convert a public subnet to private subnet?

We remove IGW and NAT Gateway, Associate subnet in private route table

4. What is the default route for any newly created route table?

The default route 0.0.0.0/0 in IPv4, ::/0 in IPv6

5. How would you ensure routes in the route table DO NOT use the local routes?

To ensure that routes in the route table do not use the local routes, we can explicitly specify the routes we want in the custom route table. By default, a custom route table doesn't have any local routes i.e., routes for the VPC's local traffic within the VPC itself.