**Ronak Patel**

**Day3 20 May 2020**

**What is Branching :-**

The git branch command is actually something of a branch management tool. It can list the branches you have, create a new branch, delete branches and rename branches.

Most of [Git Branching](https://git-scm.com/book/en/v2/ch00/ch03-git-branching) is dedicated to the branch command and it’s used throughout the entire chapter. We first introduce it in [Creating a New Branch](https://git-scm.com/book/en/v2/ch00/_create_new_branch) and we go through most of its other features (listing and deleting) in [Branch Management](https://git-scm.com/book/en/v2/ch00/_branch_management).

In [Tracking Branches](https://git-scm.com/book/en/v2/ch00/_tracking_branches) we use the git branch -u option to set up a tracking branch.

Finally, we go through some of what it does in the background in [Git References](https://git-scm.com/book/en/v2/ch00/_git_refs).

**What is Merging :-**

The git merge tool is used to merge one or more branches into the branch you have checked out. It will then advance the current branch to the result of the merge.

The git merge command was first introduced in [Basic Branching](https://git-scm.com/book/en/v2/ch00/_basic_branching). Though it is used in various places in the book, there are very few variations of the merge command — generally just git merge <branch> with the name of the single branch you want to merge in.

We covered how to do a squashed merge (where Git merges the work but pretends like it’s just a new commit without recording the history of the branch you’re merging in) at the very end of [Forked Public Project](https://git-scm.com/book/en/v2/ch00/_public_project).

We went over a lot about the merge process and command, including the -Xignore-space-change command and the --abort flag to abort a problem merge in [Advanced Merging](https://git-scm.com/book/en/v2/ch00/_advanced_merging).

We learned how to verify signatures before merging if your project is using GPG signing in [Signing Commits](https://git-scm.com/book/en/v2/ch00/_signing_commits).

Finally, we learned about Subtree merging in [Subtree Merging](https://git-scm.com/book/en/v2/ch00/_subtree_merge).

**What is an Elastic IP and how it is different from Dynamic IP?**

**Ans :-**

An *Elastic IP address* is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

An Elastic IP address is a public IPv4 address, which is reachable from the internet. If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet. For example, this allows you to connect to your instance from your local computer.

Elastic IPs get allocated to your account, and stay the same - it's up to you to attach them to an instance or not. You could say they are static public IP addresses.

In Dynamic IP - i.e. if you stop/start your instance you get reassigned a new Dynamic IP.

**What is the Client Server Model, Explain in detail?**

Ans :-

The client-server model describes how a [server](https://techterms.com/definition/server) provides resources and services to one or more [clients](https://techterms.com/definition/client). Examples of servers include [web servers](https://techterms.com/definition/web_server), [mail servers](https://techterms.com/definition/mail_server), and [file servers](https://techterms.com/definition/file_server). Each of these servers provide resources to client devices, such as [desktop computers](https://techterms.com/definition/desktop_computer), [laptops](https://techterms.com/definition/laptop), [tablets](https://techterms.com/definition/tablet), and [smartphones](https://techterms.com/definition/smartphone). Most servers have a one-to-many relationship with clients, meaning a single server can provide resources to multiple clients at one time.