**Git commands**

* **Git init:** create a new local repository in the current directory
* **Git clone:** copies an existing remote directory to your local machine
* **Git status:** shows the state of your current working directory and staging area
* **Git Add:** adds changes in your working directory to the staging area(which is a temporary area where you can prepare your next commit)
* **Git Commit:** records the changes in the staging area as a new snapshot in the local repository along with a message describing the changes.
* **Git Push:** upload the local changes to remote repository
* **Git Pull:** download the latest commits from the remote repository and merges them with your local branch.
* **Git Branch:** list, creates, renames or deletes branches in your local repository. A branch is a pointer to a specific commit.
* **Git Checkout:** switches your working directory to a different branch or commit, discarding any uncommitted changes
* **Git Merge:** combines the changes from one branch into another branch, creating a new commit if there are no conflicts.
* **Git Diff:** shows the difference b/w 2 commits, branches, files or the working directory and the staging area.
* **Git Log:** shows the history of commits in the current branch along with their messages, authors and dates

**Types of deployments:**

**Inplace:** we perform the update on the existing server.

**Rolling**: we specify the portion of the servers which we want to slowly roll out the update.

**Immutable**: means we are going to set up new infrastructure or servers with the new applications.

**Blue-green**: we already has the existing environment which is blue, then you are going to add the identical environment which is green that has the new code. Once you have the blue and green environment set up, you’ll route the portion of traffic away from the existing blue environment to the new green environment. We can shift this traffic by using load balancer, auto scaling and dns. These allows us to shift the percentage traffic to blue and some to green.

**Blue-Green traffic route can be done using below configuration.**

1. Canary: route equal amount of traffic to both the environment.

2. linear

3. All at once.

* You must install the CodeBuild plugin for Jenkins​ to integrate Jenkins with CodeBuild
* Install the code deploy agent on an EC2 instance when using it with CodeDeploy in a pipeline.
* Stage names must be unique within a single pipeline within an AWS account.

