Internship Summary (15 January – 15 February)

Jenkins

1. Setting up Jenkins on local as well as on remote server (AWS).
2. Integration of GitHub with Jenkins and cloning repository.
3. Creation of git web-hook.
4. Gathered information on other settings and services like poll scm, post actions, pipeline creation, concurrent job running, build periodically, and different plugins.

AWS

1. Completed AWS cloud practitioner essentials certification from Amazon Skill Builder.
2. EC2 (Elastic Compute Cloud):

* Launching instance
* Gathered knowledge of load balancer

1. AMI (Amazon Machine Image):

* Creation AMI of Jenkins installed instance and launching that AMI

1. Gathered and implemented the basics of security groups.
2. Basic knowledge of auto-scaling.
3. AWS S3 (Simple Storage Service):
4. Life cycle
5. Types of life cycle
6. Hosting static website using S3
7. Bucket versioning
8. CloudFront:

Hosted website using cloud-front

1. AWS VPC (Virtual Private Cloud):

* Basic knowledge of Virtual Private Clouds, route tables, subnets, and internet gateway.
* Basic knowledge of NAT gateway.

1. DynamoDB: Table creation
2. SNS: subscribed by mail address.
3. CloudWatch:

* Created a small project on triggering an alarm when the CPU threshold is crossed. In this alarm triggers SNS while we send to the subscribed email and the ec2 instance will be stopped.

Adding widgets on the dashboard.

* Created custom metrics in CloudWatch for tailored monitoring based on specific application requirements. (with the help of AWS documentation)

1. CloudFormation:

* Deployed AWS resources using CloudFormation templates.
* Created and developed templates for VPC configurations, security groups, and EC2 instances.

1. Also studied different services like AWS Amplify, AWS Guard Duty, security hub, and AWS WAF.

GIT/GitHub

1. **Git Configuration**:

- Configured global user name and email for Git.

- Checked Git configuration using `git config --list`.

2. **Repository Initialization and Cloning**:

- Initialized new Git repositories using `git init`.

- Cloned existing repositories using `git clone`.

3**. File Operations**:

- Added files to staging area with `git add`.

- Committed changes with `git commit -m`.

- Checked file differences using `git diff`.

4. **Branching and Merging**:

- Created and switched branches with `git branch` and `git checkout`.

- Merged branches with `git merge`.

- Monitored commit history using `git log`.

5. **Tags:**

- Created lightweight and annotated tags.

- Showed details of a specific tag.

- Deleted tags and pushed tags to remote.

6. **Pushing to GitHub**:

- Pushed changes to remote repositories.

- Pushed specific branches using `git push origin [branch\_name] `.

7. **Cherry-pick and Rebase**:

- Applied specific commits to current branch using `git cherry-pick`.

- Rebased branches using `git rebase`.

8. **Stashing**:

- Stashed changes temporarily using `git stash`.

- Applied stashed changes with `git stash pop`.

- Cleared and managed stash list.

9**. Searching in Git Repo**:

- Used `git grep` to search within the Git repository.

10**. Counting Commits**:

- Counted commits on the main branch using `git rev-list --count main`.

11**. Difference Commands**:

- Checked file differences before and after staging using `git diff` and `git diff --staged`.

12**. Resetting Changes**:

- Reset changes in files using `git reset`.

13. **Reverting Commits**:

- Reverted specific commits using `git revert`.

14**. Merge, and Pull**:

- Merged changes into the local branch using `git merge`.

- Pulled changes from remote repositories using `git pull`.

15**. Remote Repository Management:**

- Added remote repository using `git remote add origin`.

- Checked remote repositories using `git remote -v`.

Linux

1. Stdin, Stdout, and Stderr.
2. Logs
3. Log Rotation
4. Softlinks and hardlinks
5. Alias
6. Generated keys using Ssh-Keygen command