Wednesday
4th June, 2025

Q What is DevOps?

DevOps is a combination of Development & operations. It is a way of working where developers and operations teams work together to build, test, and release software faster and more reliably.

It focuses on automation, collaboration, and continuous delivery.

Q Why there is a need of DevOps?

There is a need for DevOps to speed up software development, improve collaboration between teams, and reduce errors through automation.

It helps in delivering updates faster, fixing bugs quickly, and maintaining high quality software.

DevOps ensures smooth and continuous delivery, making the ~~delivery~~ development process more efficient and reliable.

Q What if there is no DevOps?

Then devs & Ops teams work separately, leading to poor communication and delays. Software updates become slow, bugs take longer to fix, and manual processes cause more errors.

Without automation and collaboration, the overall software quality drops, and it becomes harder to deliver reliable products quickly.

Wednesday
4th June, 2025

Q What are the roles of the Developers, Operations team, and how does DevOps bring them together?
A Developers writes code, builds new features, and fixes bugs in the application.

The Operations team handles the deployment, servers management, monitoring, and ensures the system runs smoothly.

DevOps combines both roles by encouraging collaboration and using automation tools to build, test, and deploy software faster and more reliably.
It bridges the gap between development and operations, making the whole process seamless.

Q What is the meaning of DevOps core practices? And what are the core practices of DevOps?
DevOps core practices are the key methods and tools used to make software development and delivery faster, automated, reliable, and secure.

1. Continuous Integration (CI):
Devs regularly add (integrate) their code to ~~repo~~ a shared repo. The code is automatically tested to catch errors early.
Jenkins, Gitlab Actions, Gitlab CI/CD.
2. Continuous Delivery (CD): After CI, the tested code is automatically prepared for release so it can be deployed at any time without manual steps.
Gitlab CI/CD
3. Infrastructure as Code (IaC): Servers, networks, and infrastructure are managed using code instead of manual setup. This makes it faster, repeatable, and less error-prone.
- Terraform, Cloud Formation, Ansible
4. Containerization and Orchestration: Apps are packaged in containers (like boxes) so they run the same on any system. (Dockers)
Orchestration (Kubernetes) manages many containers automatically
5. Monitoring and Logging: Systems are continuously monitored to track issues, performance, and ensure everything runs well.
- Prometheus, Elk stack
6. Security & DevSecOps
7. Site Reliability Engineering: Focuses on keeping services reliable and available. It uses software tools to automate operations and solve problems before users face them.

Q What is Version Control System? A tool that tracks changes in code over time. It helps devs manage, save, and go back to earlier versions of their code, especially when working in teams.

Q What is Github? A cloud-based platform where devs can store their Git repos online. It allows code sharing, collaboration, and version tracking with others using Git.

Wednesday
4th June, 2025

Q. What is Git? A popular version control tool used to manage code locally. It lets you track changes, switch between versions, and collaborate with others.
Git is used on your system, and Github is where you can upload and share your Git-managed code.

Q Why should we use Git or Github?
Git is used to track changes, manage versions, and work safely on code without losing progress.

Github is used to store that code online; making it easy to collaborate with others, share projects, and backup our work.

Q What is Github and how is it different from Github?
Github is a web-based platform for hosting Git repos, similar to Github. Also supports version control, collaboration, and CI/CD.

The main difference is that Github offers a built in CICD system and can be fully hosted on your own servers (self-hosted), giving more control.
Github also supports CI/CD but often needs third-party tools or Github actions.

Github focuses more on an all in one DevOps solution.

Q Github Actions
are automation tools built into Github that let you run tasks automatically when something happens in your repo - like pushing code, creating a pull request etc.

help us to automate things like testing, building, or deploying our code, so we dont have to do it manually every time.

Ex. - Run tests auto when push
- Deploy app to a servers / cloud
- Build & package our code on every commit.

Q How to setup Github Actions?
1. Go to Github Repo
2. → "Actions" tab
3. Will suggest workflows, or click on "New workflow"
4. Create a new file in the .github/workflows/ folder (ex: main.yml)
5. Write a workflow in YAML format to define when & how it should run.
6. Commit the file. Github ~~with~~ Actions will run automatically based on the rules you set.

Q. What is the meaning of a Repository in Git or Github?
A repository is like a folder or storage space where all our project files & their full history of changes are saved.

In Git, it tracks our changes, commits, and versions locally.

In Github, it is the online version of our project - where we can collaborate, store, and share our code with others.

Wednesday
4th June, 2025

Example for running on push:

```
name: CI
on: [push]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: run a script
        run: echo "Hello"
```

This runs whenever you push code to the repo.

Q What is YAML?
YAML Ain't Markup Language is a simple, human-readable format used to write configuration files.
It uses identation to show structure and is easy to read and write.
In DevOps, YAML is commonly used in tools like Github Actions, kubernetes, and CI/CD pipelines to define setting & workflows.

Q Why does Android Studio ask for the Github URL when connecting a project for the first time?
Main Reason: Android Studio needs the repo URL, so it knows where to push our code on Github.

When we give URL, Github checks:
- is this our repo?
- is it public or private?
- Do we have permission to push to this repo?

So giving URL also starts the authentication process:
- if repo is private, it will ask us to log in or give a token.
- if repo is public, it still checks that you were allowed to push.

Q Token Authentication:
- Secure way to log in to Github without using our password. (a long, secret code)
- It proves our identity to Github when pushing.

Why? Because
- More secure
- required for private repos
- Github stopped allowing password logins for pushing code

Profile setting → developer setting.
→ Personal access token
→ Generate new token
→ classic
→ token-name
→ expiration
Permission → check repo (private)
check workflow (if using Github Actions)

Wednesday
4th June, 2025

Q What are the simple steps to create a GitHub Repository?

1. Go to github.com and log in.
2. Click the '+' icon on the top right, then select "New Repository".
3. Enter a Repository name and optional desc.
4. Choose public or private visibility.
5. (Optional) Tick "Initialize this repository with a README".
6. Click "Create Repository".

Q Whats are the additional options like License and .gitignore while creating a GitHub Repository.

License: Choose a license (ex, MIT, Apache) to define how others can use, modify, and share your code.

.gitignore: This file tells Git which files or folders to ignore (like temp files, build folders, etc.) so they are not pushed to the repo.

Q How to commit or push code using the GitHub web platform (without using Git or any IDE)?

1. Go to your repo on GitHub.com.
2. Click on the file you want to edit, or click "Add file" > "Create new file" to add new.
3. Make changes or write new code.
4. Scroll down to the "Commit changes" section.
5. Enter a message.
6. Choose: Commit directly to the main branch - if wants to save it directly.
   OR create a new branch - if you want to do it safely (for bigger changes)
7. Click "Commit Changes" btn.

Q What is the concept of branch in Git?
Explain with a real example.

- A branch in Git is like a separate line of work where you can make changes without affecting the main project.

Q How to commit or push code to GitHub repo using Android Studio?

1. Open project in Android Studio.
2. Go to VCS > Enable version control integration, and choose Git.
3. Then go to VCS > commit or use commit button
   - select files to commit
   - add a message
   - click commit
4. Now, go to Git > Manage Remotes
   - Click + and add GitHub repo url as origin
5. Finally, go to Git > Push
   - select branch
   - click push

Q. What is the meaning of fork & star.

Fork: A fork is your own copy of someone else's GitHub repo. It lets you freely make changes, or add new features without affecting the original project. You can also send a pull request later if you want to suggest your changes to the original repo.
Use it when you want to contribute to open-source or use someone's project as a base.

A star is like a bookmark or like. It shows appreciation for the project and helps you easily find it later in your GitHub profile under "starred repositories."