Pull Request Automation, Deployment Automation, Application Performance Managemant.

Pull Request Automation:

- The goal of DevOps engineer is to provide good quality service quickly and make sure it continues to be delightful and bug free for end users.
- **Developers commonly share the work with each other by proposing the new sets of changes. The likeliest tools are GitHub, GitLab, Bitbucket developers use git and push the set of changes and open a pull request to get those sets of changes integrated into the primary code bases.**
- Commonly each set of changes are reviewed by another developer on the same team, but the review process can include many stakeholders in larger teams. The people who review's it are
- Developers (code owners), product managers (incharge of functionality being proposed), designers (visual changes), translators, accessibility stake holders, security reviewers and other non-technical parties.
- The business goal of DevOps engineer working on pull request automation is to speed up review process.
- **Automated test runing with CI provider gives developers confidence that the change does not break the present code functionality.**
- Per charge ephermal environment helps parties to interact with proposed change to ensure it solves all the required business goals.
- Automated security scanning ensures that the proposed changes does not introduce new vulnerabilities into the product.
- Notification to reviewers so that the correct reviewers can quickly request changes to a pull request.
- Ideally changes should be reviewed and pulled within 24 hours of when they are made. Long review cycle is the common reason for slowed down software developers.

Deployment Automation:

- The efficiency of a build process isn't the only goal of deployment automation, other include such as
- Deploying a feature for certain set of users as a final test before rolling it out publicly.
- Starting new versions of services without causing downtime.
- Rolling back to prior version incase something goes wrong.
- **Broadly success in deployment automation is finding the appropriate deployment tools to fulfill business goals.**

Application Performance Management:

- The core goal of this pillar is to ensure that your **service continues to perform well in production**. Mainly it has four targets such as metrics, logging, monitoring and alerting.
- **Metrics:** Numerical measurements of keys in production. Usually in the form of finite resources (disc space, memory usage) and times (response time, job processing time).
- **Logging:** Text area of what is happening during the processing of logs often come with meta data often their source, time, related metrics.
- **Monitoring:** Takes the metrics, logs and converts them into health metrics (does the product feel slow, is it down, all features working with no errors).
- **Alerting:** If the monitor detect a problem, a correct problem solver should be notified automatically.

Test Driven Deployment:

- **Test Driven Deployment is a coding methodolgy where the tests are written before the code is written.**
- Example for test driven deployment is coffee machine.
- **Unit Test:** Ensure individual components to work on their own.
- **Integration test:** Few components work together.
- **System test:** Everything work together.
- **Acceptance test:** After being launched everything is perfect or not.
- **Before:** Choose something to work on. Build it. Test it.

- **After:** Choose sometimes to work on. Write test. Keep building untill all the test are passed.