CI/CD UDACITY-PROJECT

GIVE YOUR APPLICATION AUTO-DEPLOY POWERS

By Mahmoud Elsayed Aly

OVERVIEW OF PRESENTATION

- The concept of CI/CD.
- Benefits of CI/CD pipelines.
- Confronted challenges with automation Powers.
- CI/CD pipeline

THE CONCEPT OF CI/CD

- Continuous Integration
- Continuous Delivery
- Continuous Deployment

CONTINUOUS INTEGRATION

- In continuous integration, source code changes are frequently and rapidly turned into software release candidates, to be further evaluated in Continuous Delivery. Continuous integration purposes include:
  - Ensured sufficient quality on the software to enable subsequent integration.
  - Rapid feedback on changes and software quality to developers.
  - Rapid software release candidate for production.
  - Frequent software release candidate for production.

CONTINUOUS DELIVERY

- It can be defined as the continuous provision of ready-to-install integrated software through automated delivery mechanisms. With cloud-native applications, more opportunities arise to automate delivery and lifecycle management on a fine-grained microservice level.
- The process includes the automated validation from a functional and performance point of view with different test engines and tools integrated into the CI flow. Successful validation can mean to commit and deliver the new release to next level CI stages or deployment.

CONTINUOUS DEPLOYMENT

- It is the continuous provision of software functions and features which are available for testing or activation in a lab, staging or production environment through automated deployment. In the service provider context continuous deployment applies updates in phases across production sites or network slices.
- Continuous deployment can be thought of as an extension of continuous integration, aiming at minimizing lead time, the time elapsed between development writing one new line of code and this new code being used in commercial deployment.

BENEFITS OF CI/CD PIPELINES

- Using CI/CD pipelines ensures that fault isolations are faster to detect and easier to implement. Fault isolations combine monitoring the system, identifying when the fault occurred, and triggering its location.
- CI/CD continuously merges codes and continuously deploys them to production after thorough testing, keeping the code in a release-ready state.
- The advantages of CI/CD do not only fall into the technical aspect but also in an organization scope.
- CI/CD is a great way to get continuous feedback not only from your customers but also from your own team. This increases the transparency of any problems in the team and encourages responsible accountability.

BENEFITS OF CI/CD PIPELINES

- Automation in the CI/CD pipeline reduces the number of errors that can take place in the many repetitive steps of CI and CD. Doing so also frees up developer time that could be spent on product development as there aren't as many code changes to fix down the road if the error is caught quickly. Another thing to keep in mind: increasing code quality with automation also increases your ROI.
- Maintenance and updates are a crucial part of making a great product. However, it's important to note within a CI/CD process to perform maintenance during downtime periods, also known as the non-critical hour.
- Testing is a large part of that process because even if you are able to make your integrations and delivery faster, it would mean nothing if was done so without quality in mind. Also, the more steps of the CI/CD pipeline that can be automated, the faster quality releases can be accomplished.

CONFRONTED CHALLENGES WITH AUTOMATION POWERS.

Performance Issues

- Perform load simulation and performance testing to compare build performance.

Team Communication

- Transparency and cooperation in your team smoothen the CI/CD pipeline workflow.

CONFRONTED CHALLENGES WITH AUTOMATION POWERS.

Version Control

- Ensure that the new version is stable. Only then should you update it.

Flawed Automated Testing

- Don’t ignore potential red flags and ignore warnings only when they are genuinely not affecting your build performance.

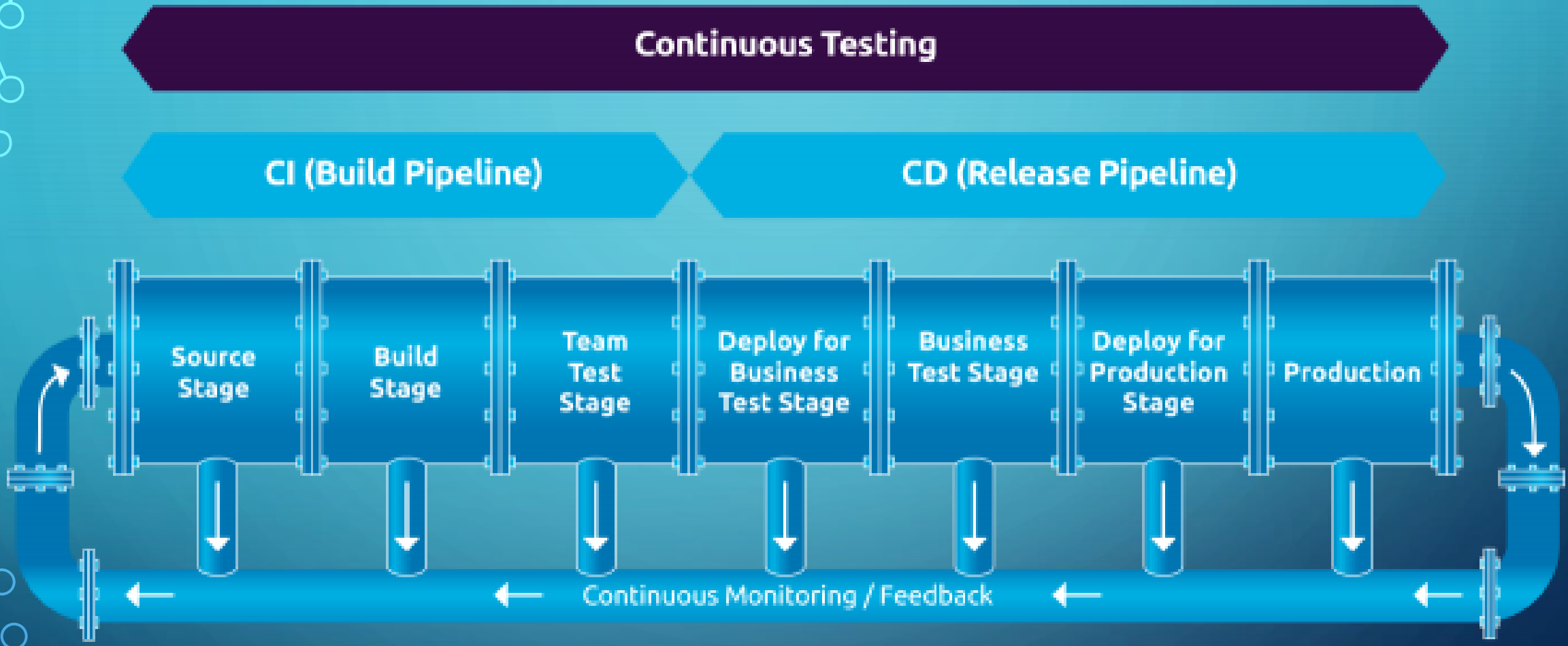
CONFRONTED CHALLENGES WITH AUTOMATION POWERS.

Security Vulnerabilities

- With an effective monitoring system, threats can be detected quickly, and you can defend the pipeline effectively.

Testing Infrastructure

- Choose a testing infrastructure that is flexible enough to handle updating versions, adding devices, and manage data capacity while maintaining stability.


Continuous Testing
CI (Build Pipeline)
CD (Release Pipeline)
Source Stage
Build Stage
Team Test Stage
Deploy for Business Test Stage
Business Test Stage
Deploy for Production Stage
Production
Continuous Monitoring / Feedback

THANK YOU