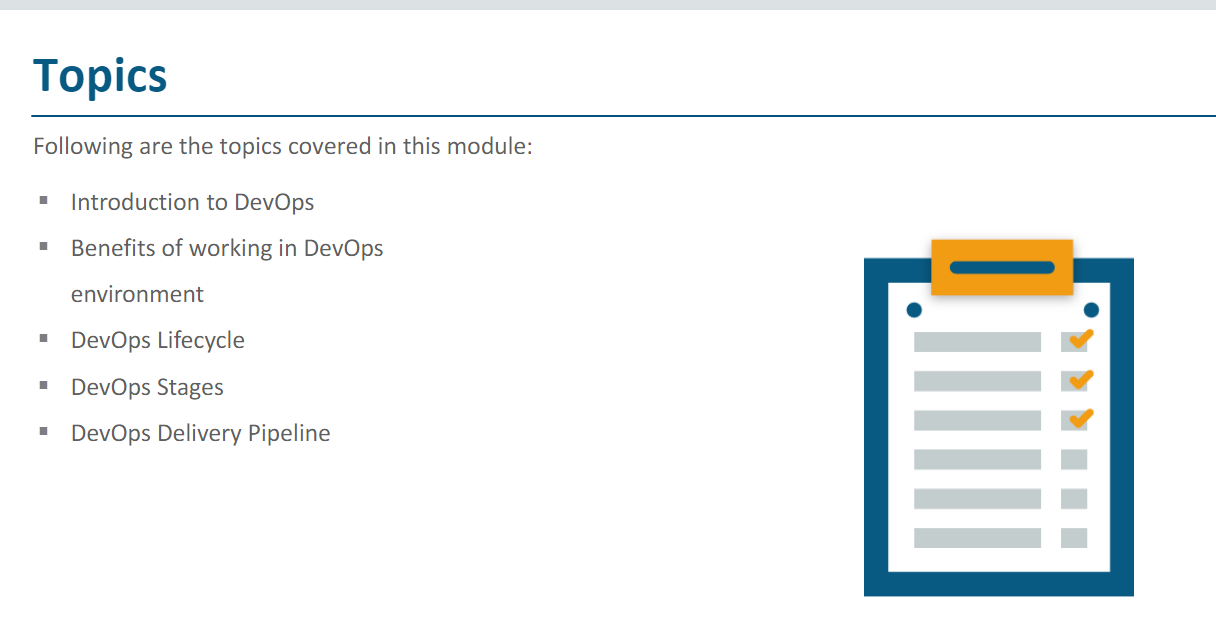
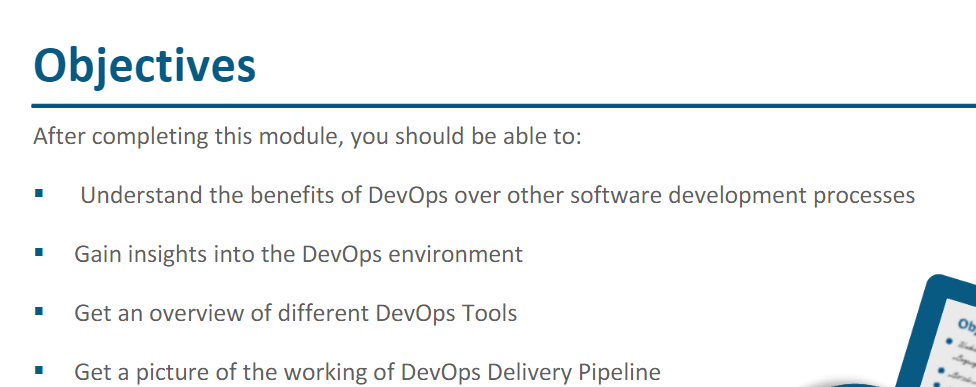
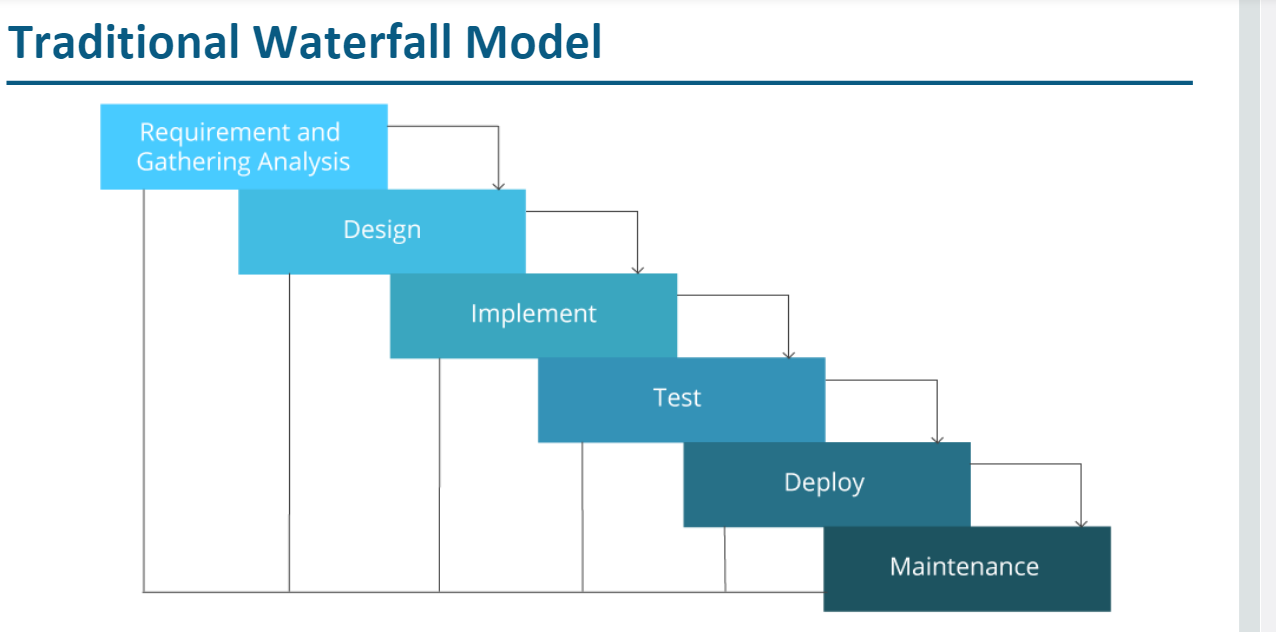
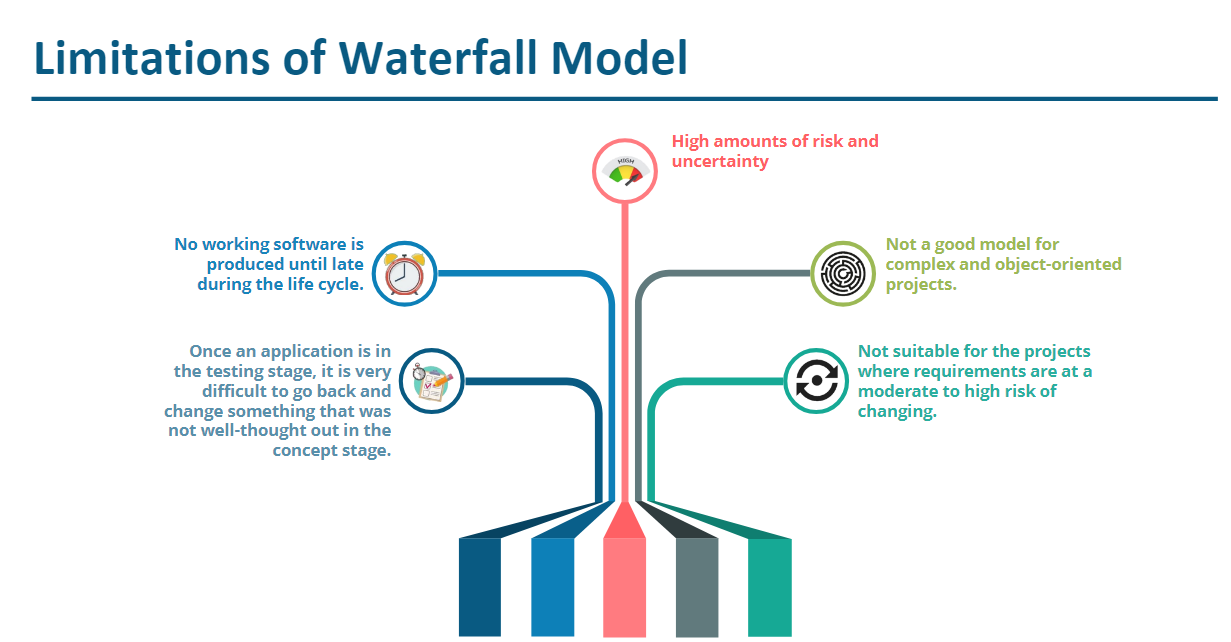
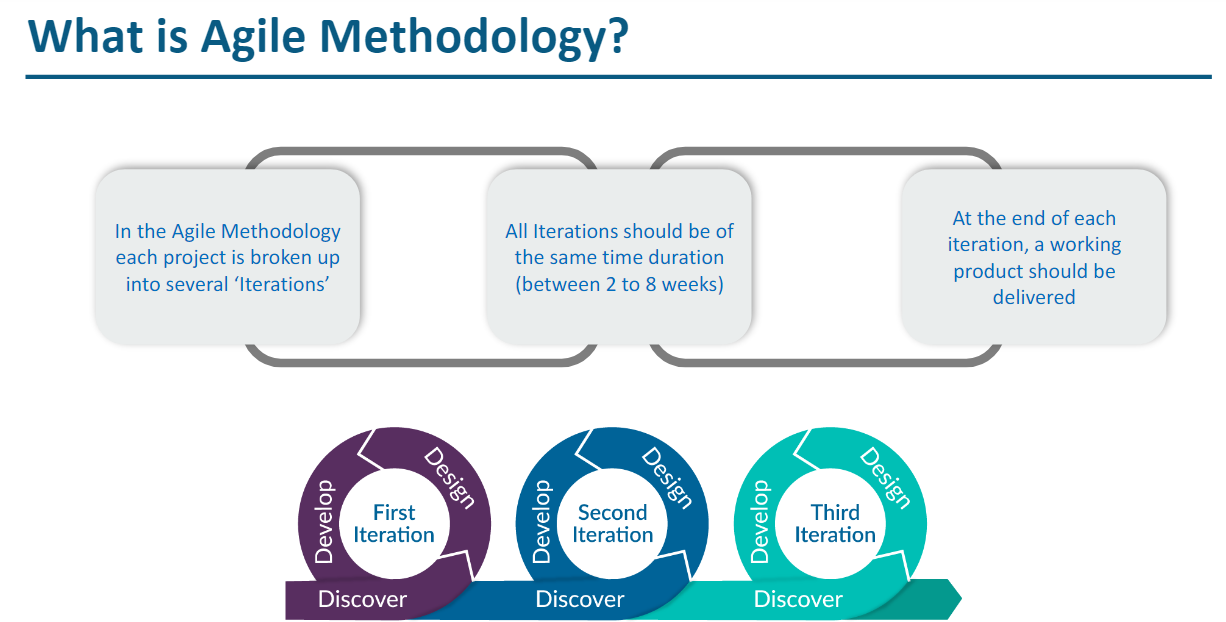
#### Intro. to DevOps!!!!!

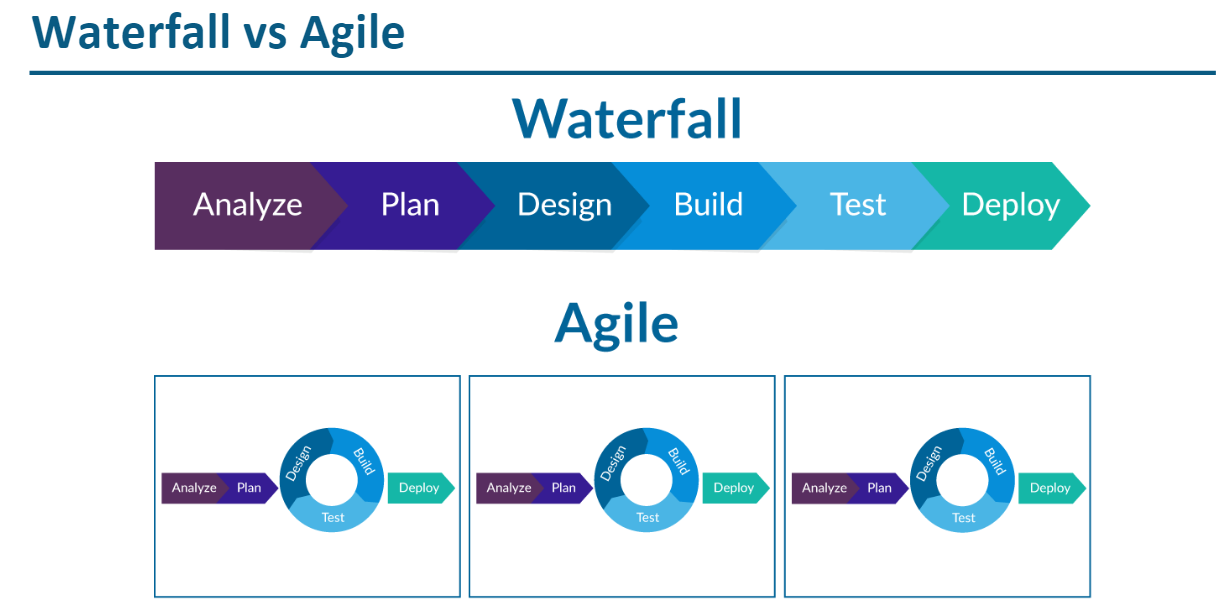












**SDLC - Software Development Lifecycle!!!**

**Requirement Analysis**

**Design**

**Development/Coding**

**Testing**

**Implementation / Deployment**

**Monitoring!**

**Waterfall ::::  -- it is Linear in fashion!**

**Requirement Analysis**

**Design**

**Development/Coding java/python**

**Testing**

**Implementation / Deployment**

**Monitoring!**

**To Create a software application/Project!**

Agile Methodologies!!!

Requirement Analysis

Design

Development/Coding java/python

Testing

Implementation / Deployment

Monitoring!

To Create a software application/Project! -- having various modules / iteration.

Iteration - 1 Module / Function

Requirement Analysis

Design

Development/Coding java/python

Testing

Implementation / Deployment

Monitoring!

Iteration - 2

Requirement Analysis

Design

Development/Coding java/python

Testing

Implementation / Deployment

Monitoring!

Iteration - 3

Requirement Analysis

Design

Development/Coding java/python

Testing

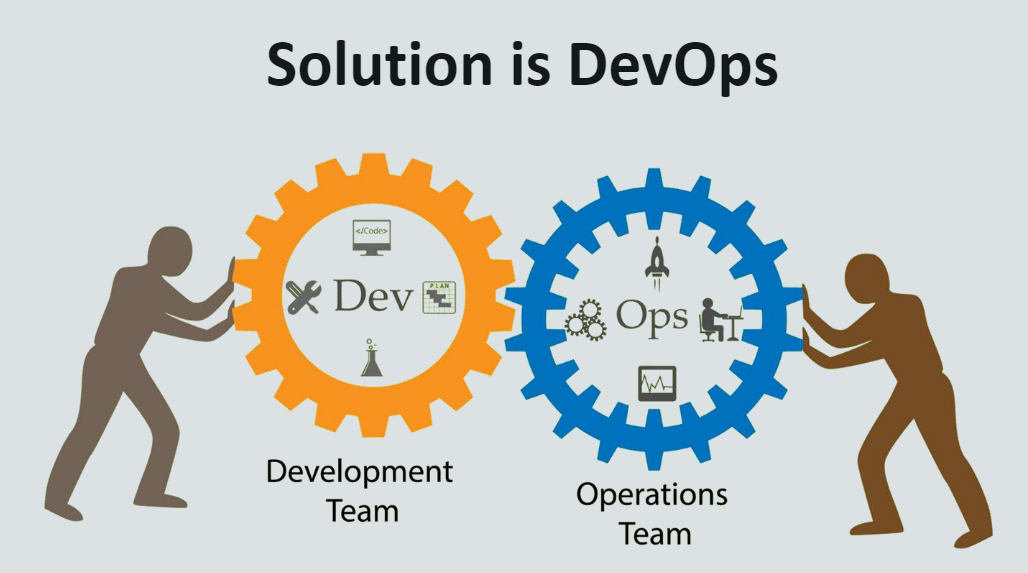
Implementation / Deployment

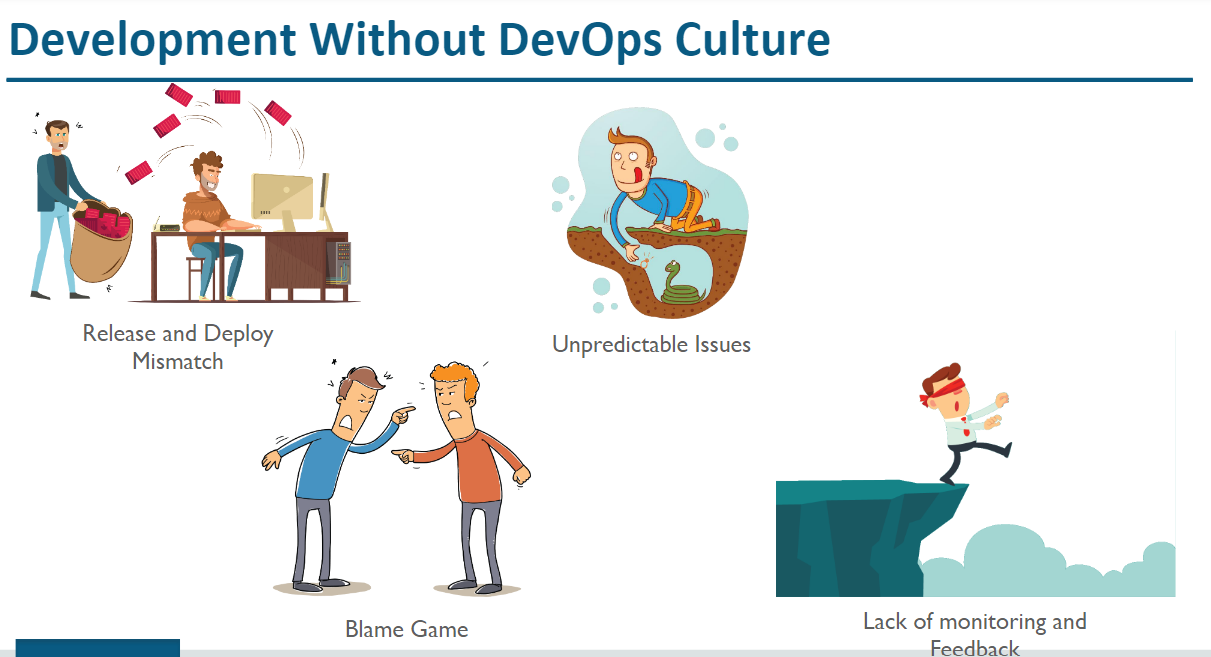
Monitoring!

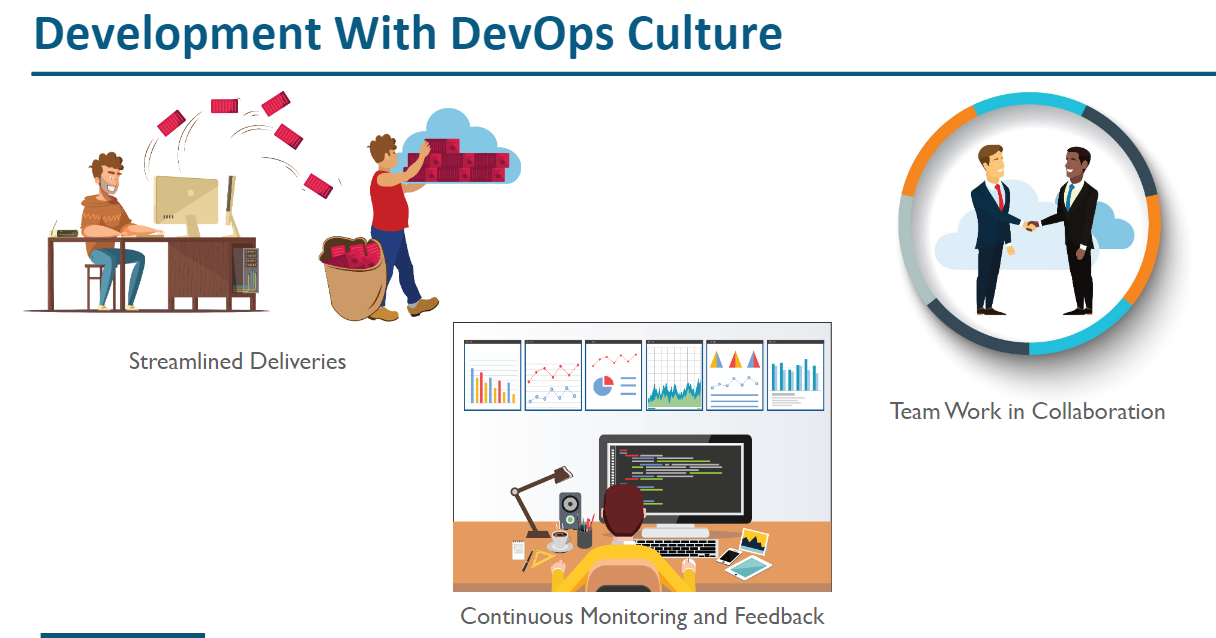
Finally, all these iterations are integrated and delivered as a software application.

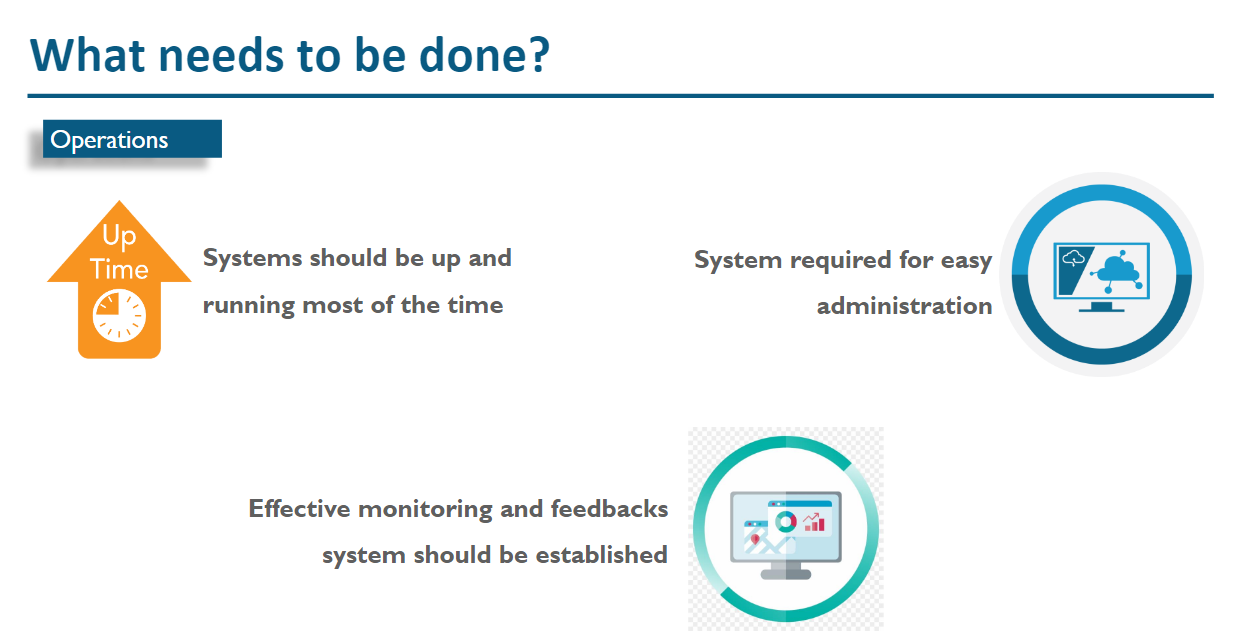


WHY DevOps ?

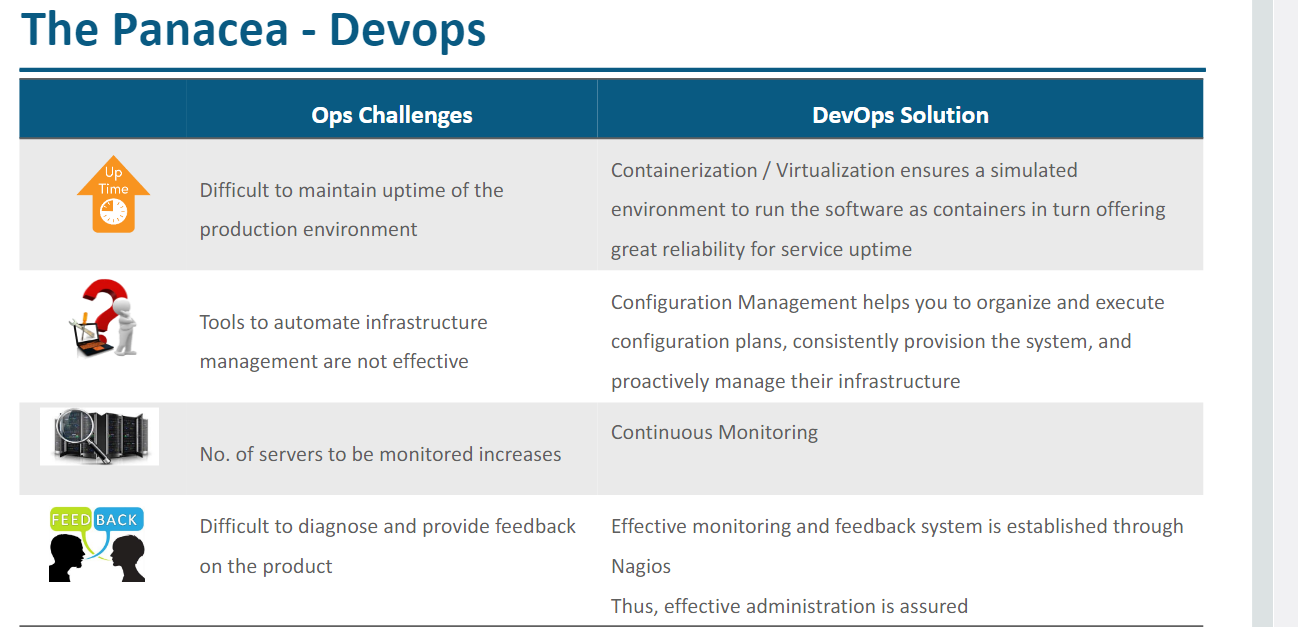






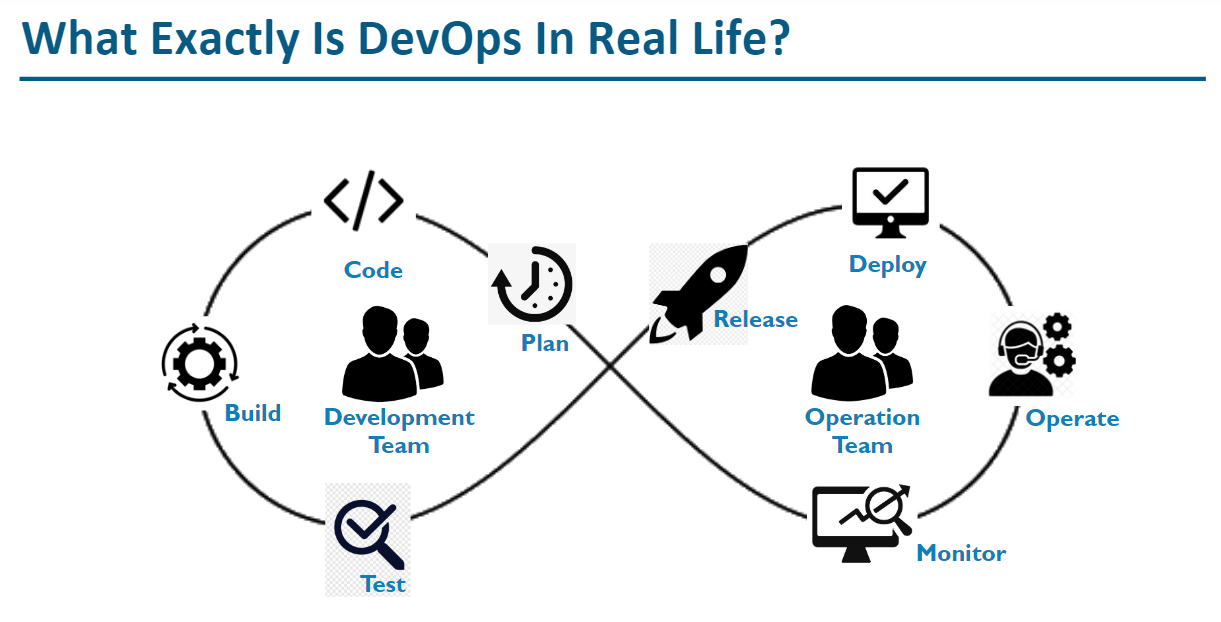




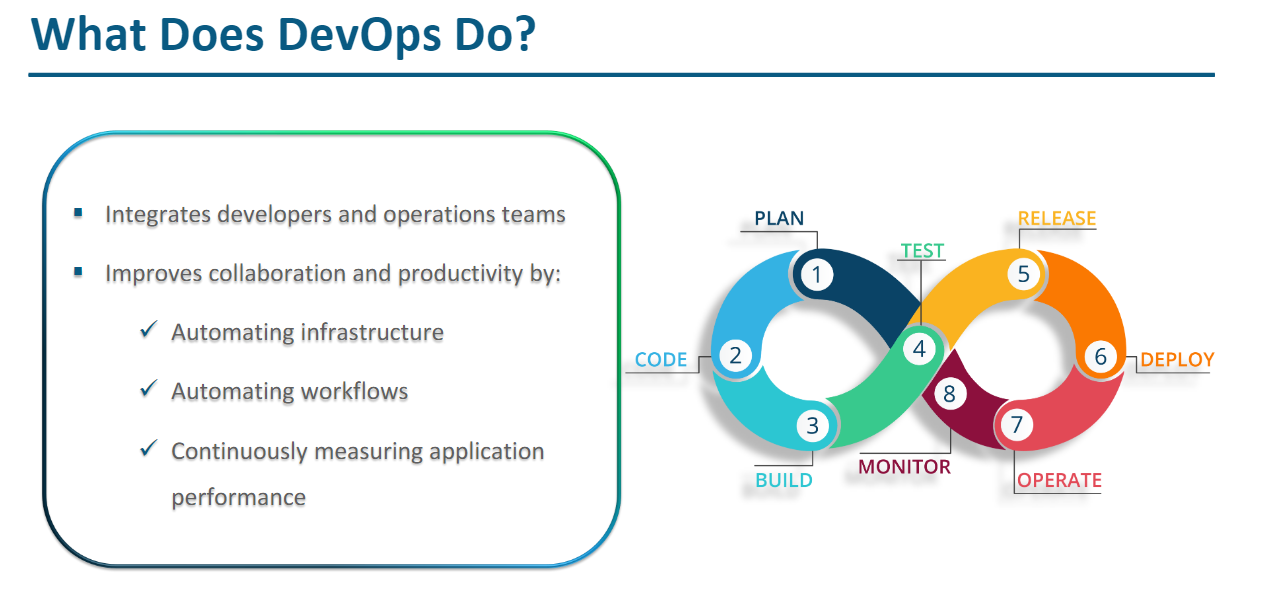




DevOps is a Software Development Strategy!







Continuous Development ! ===> related to Appln. DEV Team.

Continuous Integration !

Continuous Testing !

Continuous Delivery !

Continuous Deployment !

for code development, there are some time frame! --> 2 wks. 1

Integration !!!

Test 2 wks

Software Application/Product!

- Monolith Application Architecture!

All the process are deployment as one Application!!!

Process / tasks / Functions / Modules

- Microservices based Architecture!

Create a service

1 process === 1 service

E-commerce :

www.amazon.com

List of Process / tasks / Functions / Modules :::

User Sign-Up -- Developer-1 -- once the coding is completed(Testing)!

User Sign-In -- Developer-2 --

Search for products

Add the product to cart

Buy the product

fill the form

Make payment

Place the Order

Track the status

Application Owner / Architect

Difference between :

Continuous Integration !

Continuous Delivery !

Continuous Deployment !

Strategy -- Process -- --- Always subjected to Continuous Improvement!!!

Development Workflow :::

Code the changes / Module / Service ===> Java/Python/NodeJS/Angular/.Net (Programming Languages)

Build -- Compile and Create Artifacts(\*.war/\*.jar/\*.exe)!!!

Perform Unit Test

Testing Teams

QA

UAT

PreProd

PROD

Continuous Integration  :::

Code/Build/Create Artifacts and Integrate for further testing

Non-Prod Environment Prod Environment ( LIVE Environment )

DEV PROD

QA

UAT

The Production Release can be done by :

Continuous Delivery : Before the code is promoted to PROD envi. we need Manual Approval from various Teams.

Continuous Deployment :  The Code can be automatically promoted to PROD Environment without any Manual Intervention.

As a DevOps Engg. is to automate this end-to-end CI/Cd/CD

Code Build, Create Artifacts, Integrate for further testing(QA/UAT) and PROD Deployment

Banking Appln.

Facebook

Netflix

amazon.com

google

People

Process & Strategies

Tools

Business :::

DevOps Team :::

Identify the various Teams involved in the end-to-end SDLC Process

Infra

Dev

Test

Release Management

Monitoring Team

Prod Support Team

Security Team

Vendors

Creating CI/CD Pipeline

Detailed DevOps assessment

Waterfall --> Agile --> DevOps --> DevSecOps --> SRE --> GitOps --> AIOps --> FinOps

### DevOps Tools :::



Infra-structure Team -- Provision the servers and config the servers  - Terraform/Ansible/Puppet/Chef/CF/ARM

Dev (Version control system) -- VS Code / GIT / GITHub

Test -- Selenium/Junit

Release Management -- Anisible/Docker/Kubernetes/

Monitoring Team -- Promethus/Grafana/Nagios/

Prod Support Team -- AppDynamics

Security Team -- Static/Dynamic Code Analyzer - Sonarqube

Build Orchestrator or CI/CD Tools:

Jenkins

Bamboo

Gitlab-ci

Azure Pipeline

**Developer!!!**

**samplepgm.java index.jsp**

**Version Controlled!**

**It should be easily tracked!**

**Visual Studio Code (IDE) --> to create the source.**

**whenever we save the program -- it shd be assigned with a version.**

**index.jsp v1.0 Tested -- successful**

**index.jsp v1.1**

**index.jsp v1.2**

**index.jsp v1.3**

**index.jsp v1.4**

**index.jsp v1.5 Tested -- Not working as expected**

**GIT --> Distributed VC System**

**scripts :::**

**shell**

**groovy**

**yaml**

**python scripts**

**DevOps Engg. write Automation Scripts.**

**GIT ---> Distributed VCS ===> Local repo & Remote Repo**

**Git is a Local Repository**

**Remote GIT Repo / Servers**

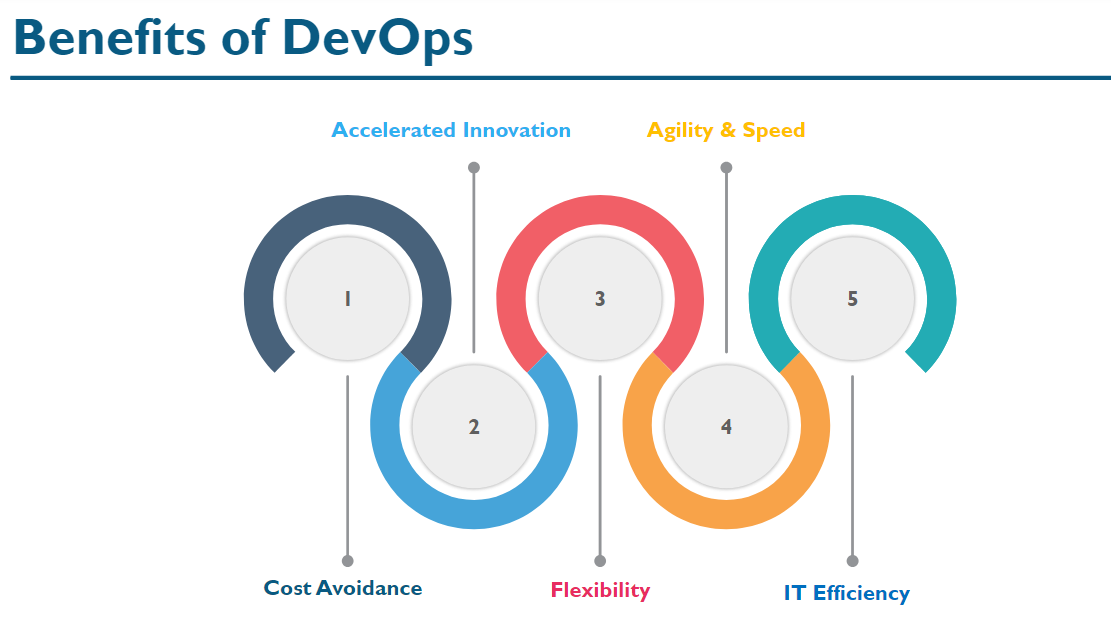
**GitHub is a Remote Git Repository**

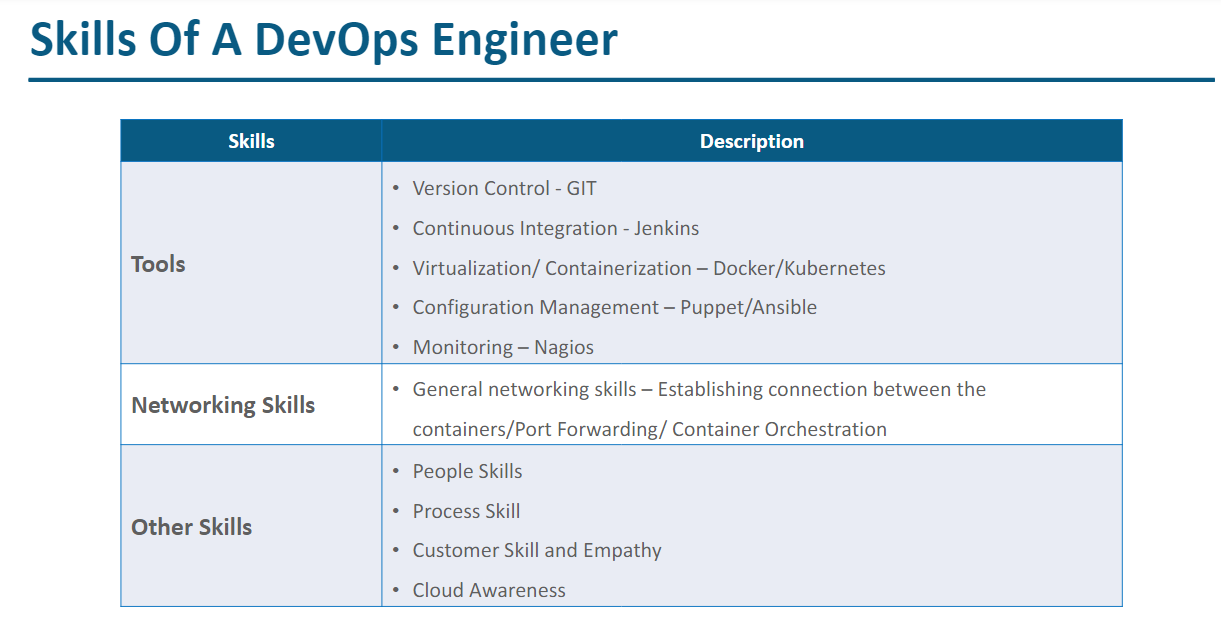
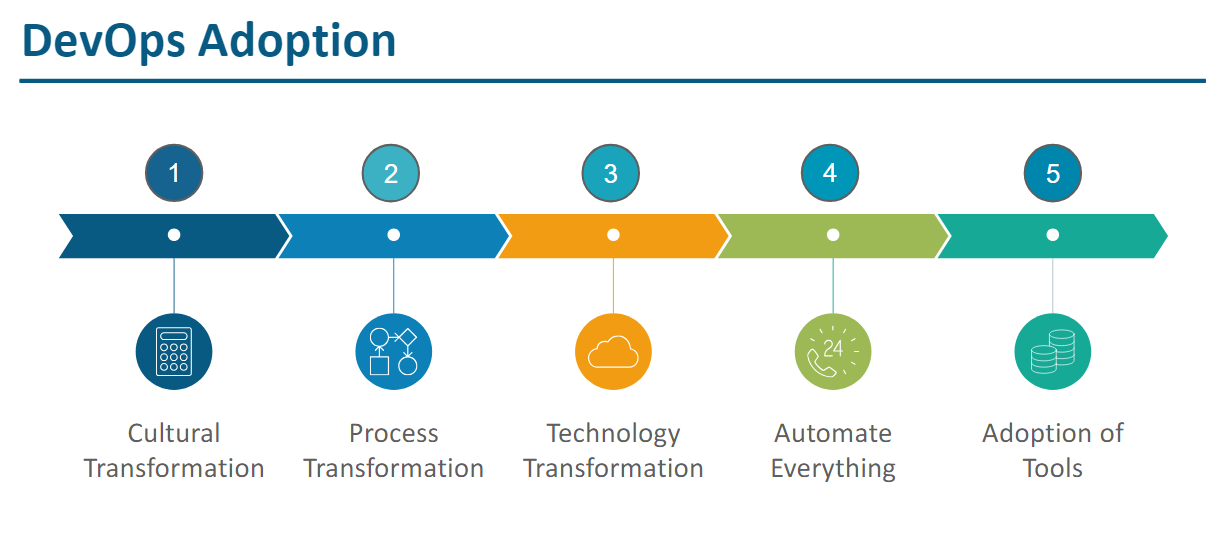
**Azure Repo**

**Bitbucket**

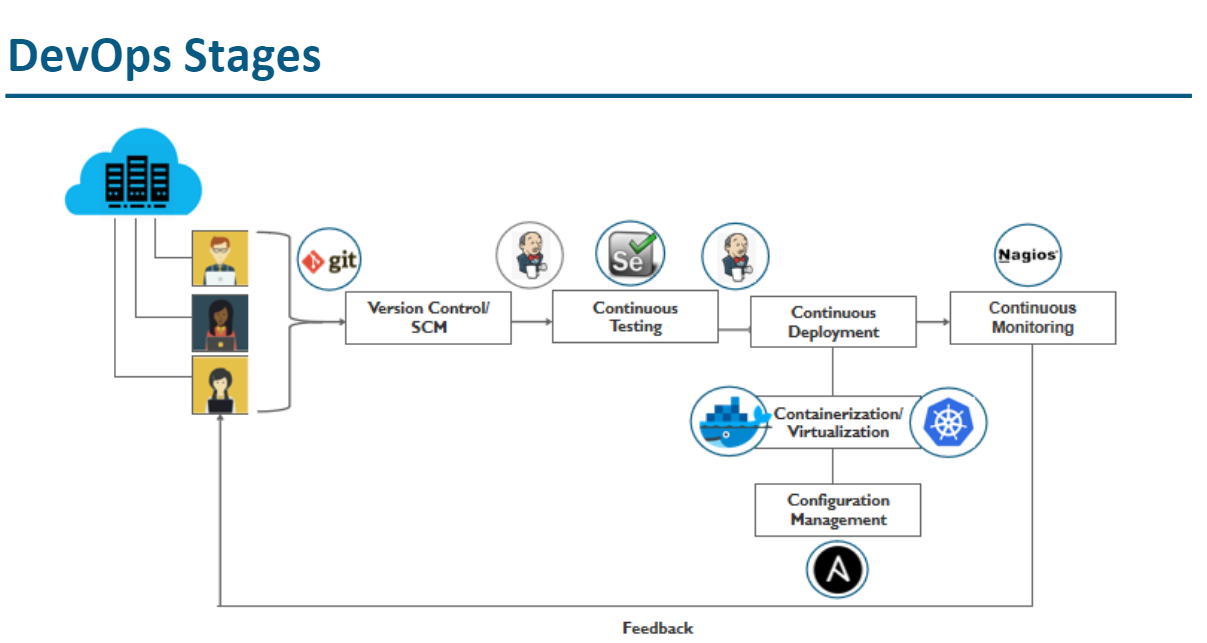
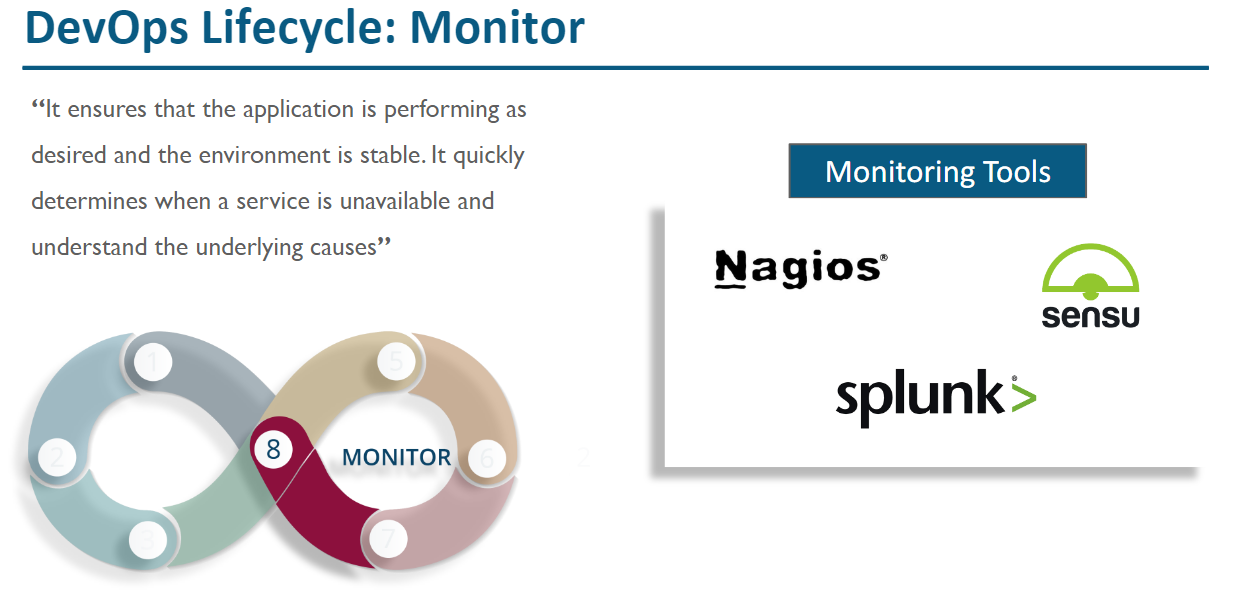
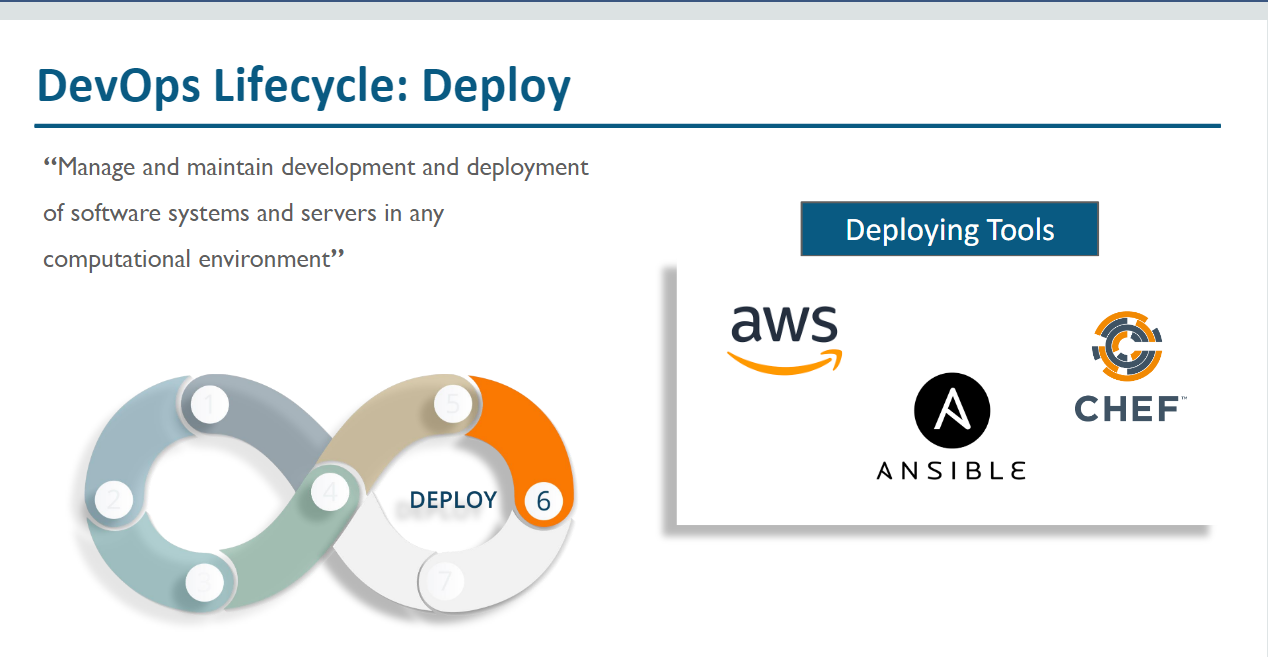
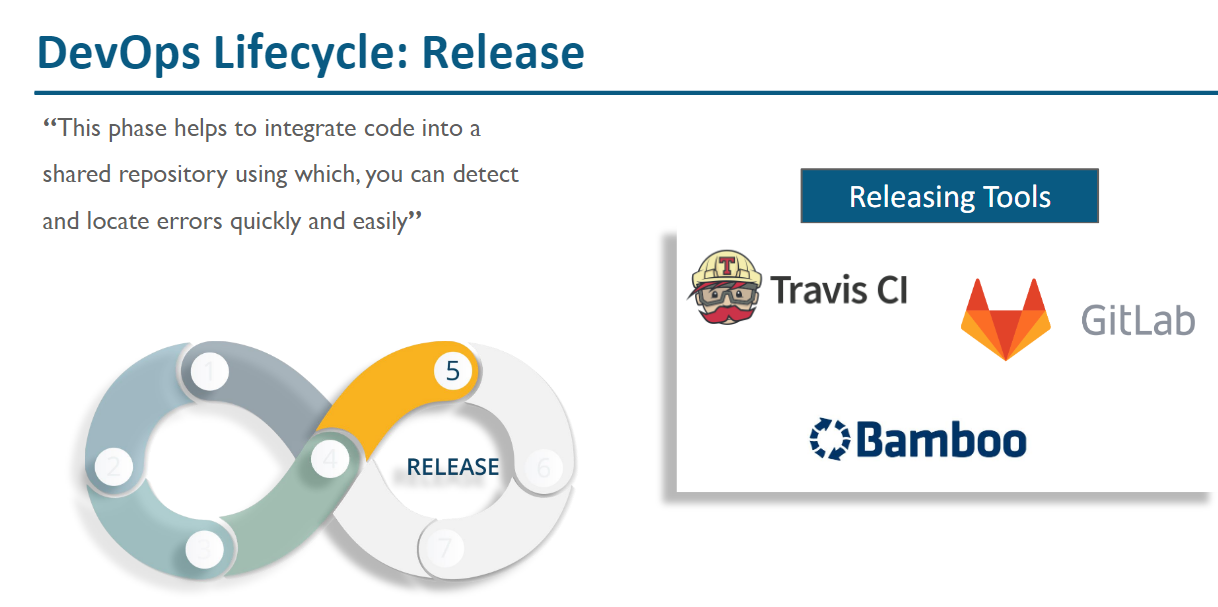
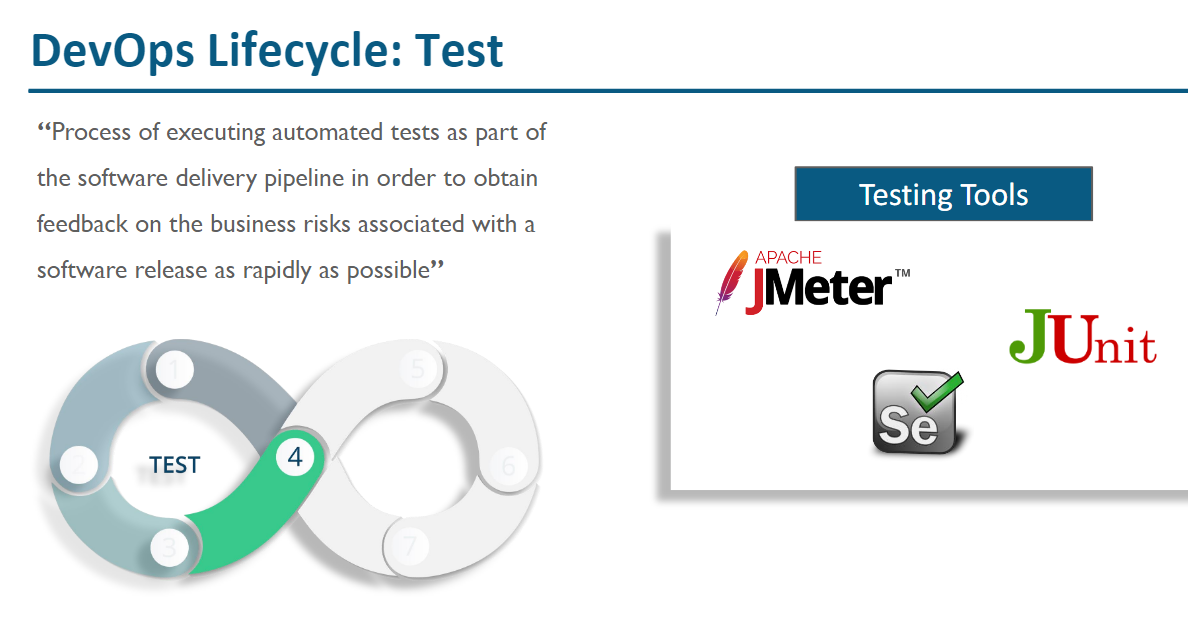
**Gitlab**

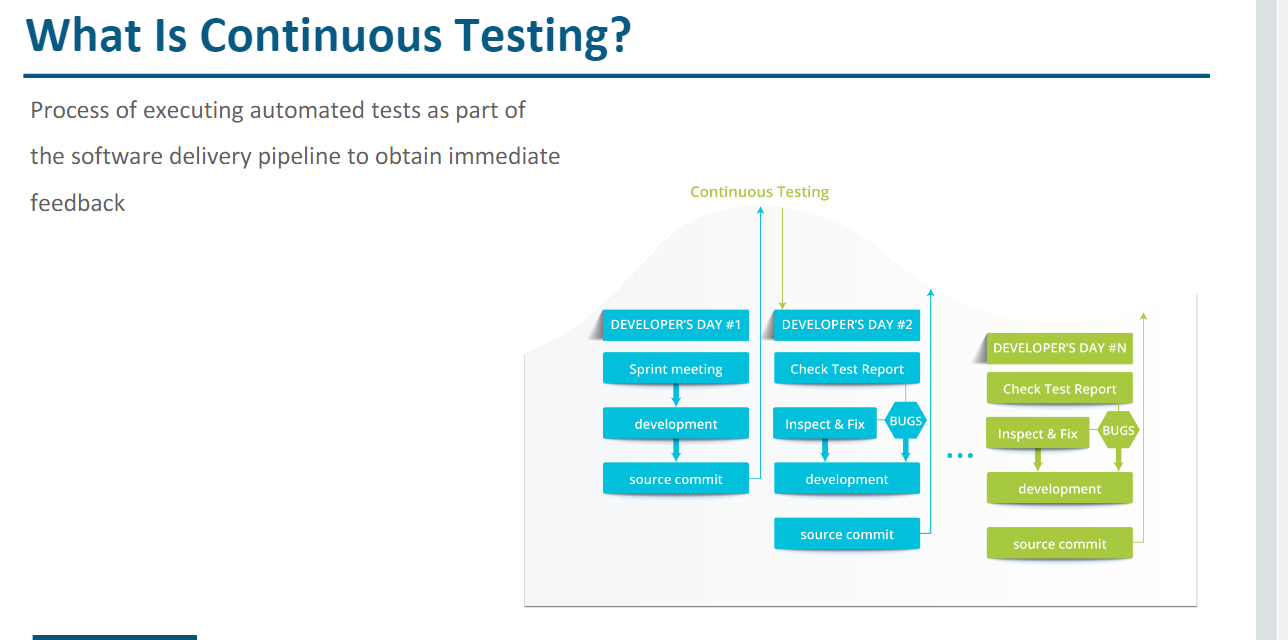
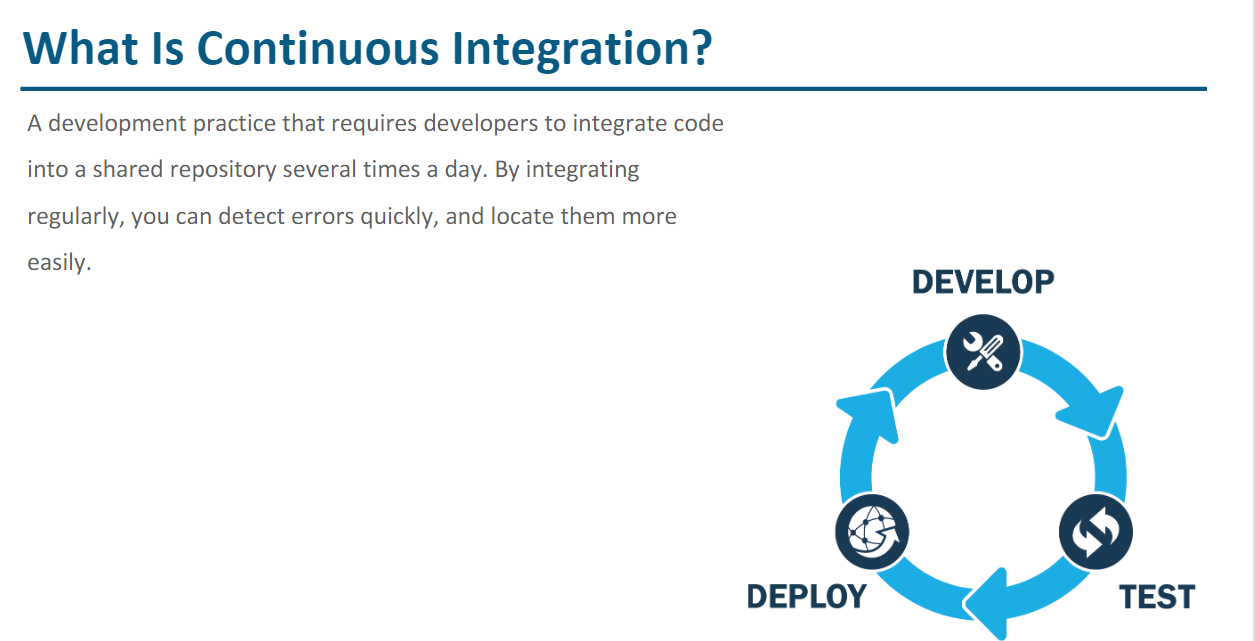
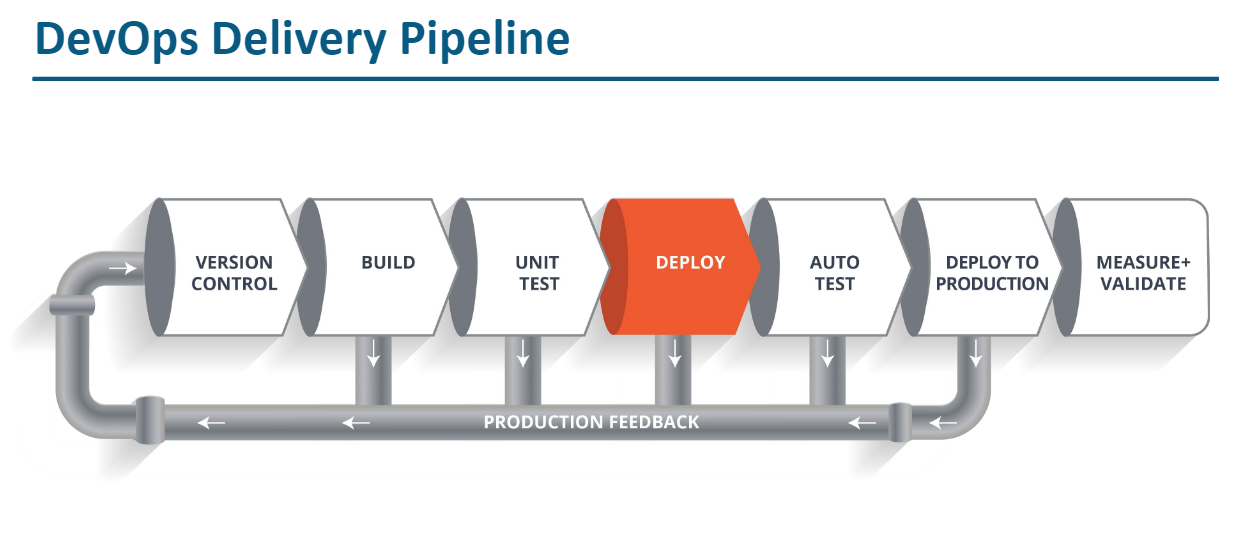
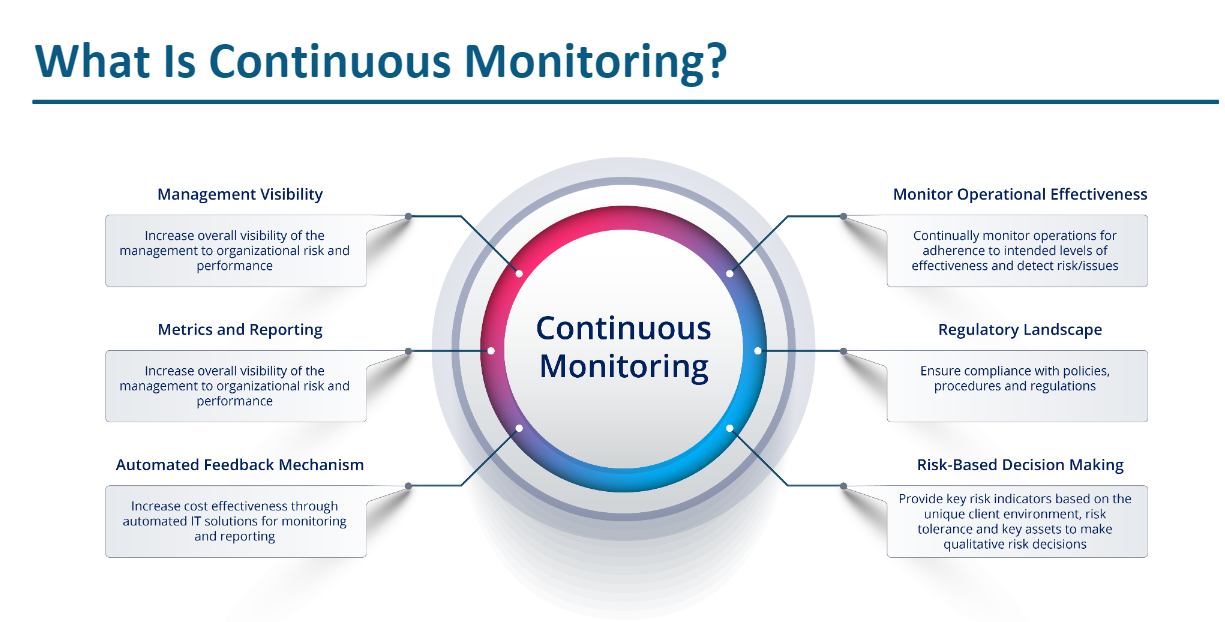
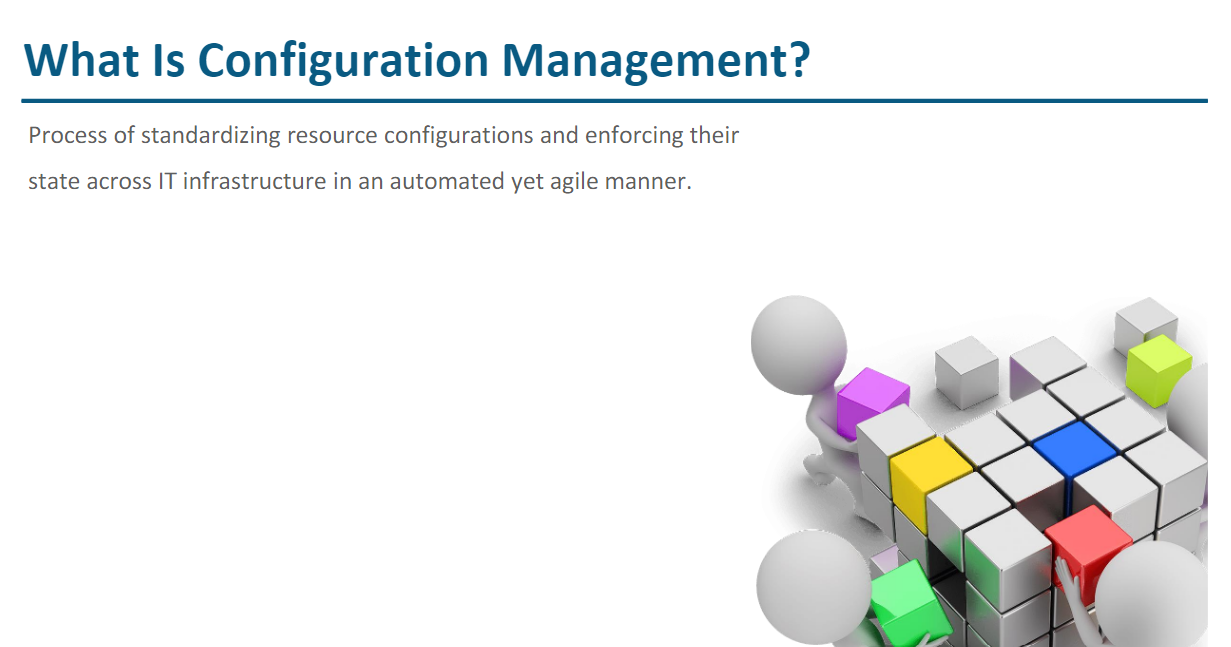
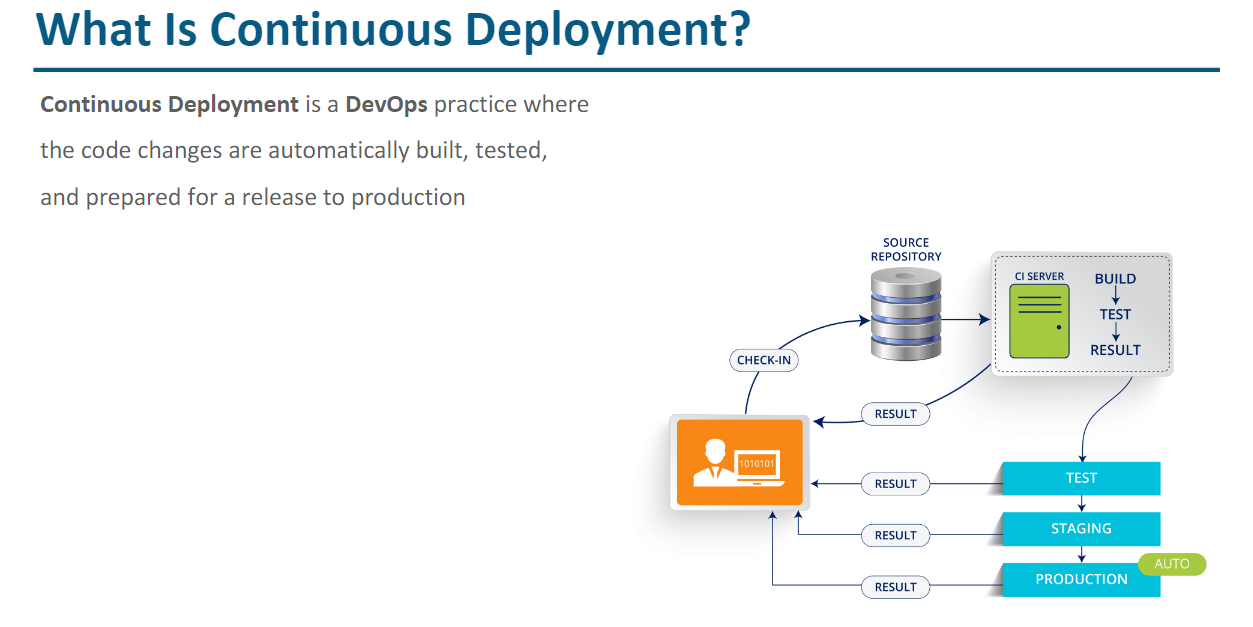
**AWS Code Commit**









k