



Introduction to DevOps on Cloud

Week 5

Course Outline
Module 12

1. Overview of DevOps

2. Oracle Virtual Box

3. Linux commands and file system

WEEK 1

4. Version Control with Git

5. Continuous Integration with Jenkins

6. Continuous Testing with Selenium

WEEK 2

7. Continuous Deployment:
Containerization with Docker

8. Containerization with Docker:
Ecosystem and Networking

WEEK 3

9. Container Orchestration using
Kubernetes

WEEK 4

10. Configuration Management with
Ansible

11. Continuous Monitoring Nagios

WEEK 5

12. Introduction to DevOps on Cloud

13. Introduction to SSH

WEEK 6

14. High Performance Server NGINX

Topics

- Introduction to Cloud Computing
- Introduction to AWS
- AWS Services
- DevOps using AWS

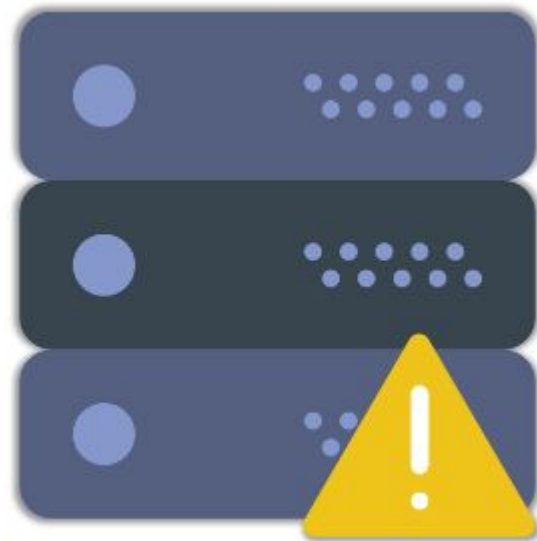
Objective

At the end of this module, you will be able to understand the below topics:

- Why Cloud
- Introduction to Cloud Computing
- Why Devops on Cloud
- Introduction to AWS
- Various AWS services
- DevOps using AWS

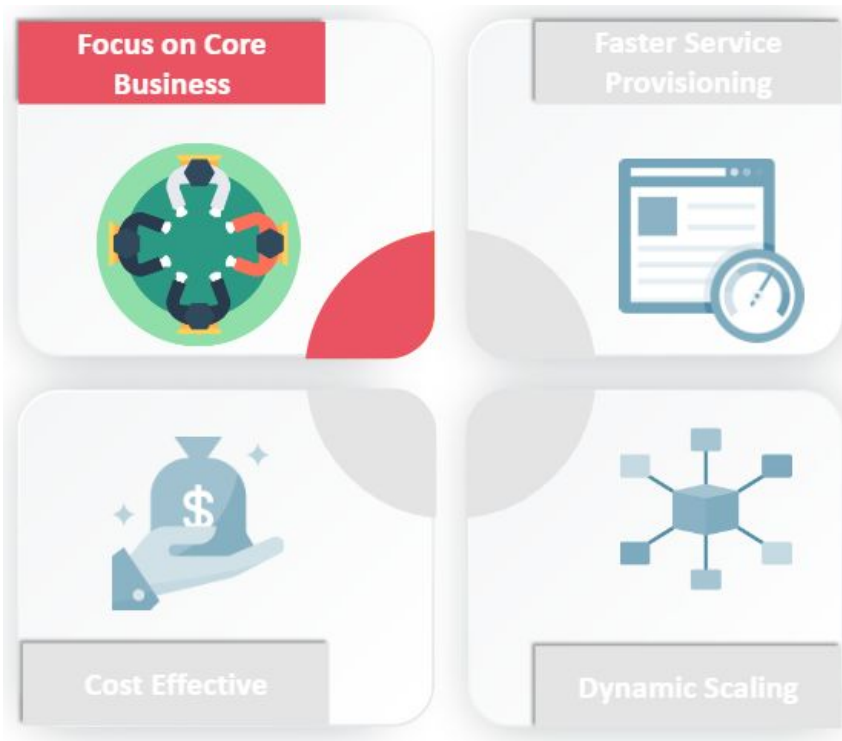
Problems Before Cloud Computing

- Cost of hardware and software and other resources were high
- Maintenance of server, installation of software or hardware was difficult
- Scaling of server was very difficult and expensive
- It was not easy to protect or recover crashed or lost data



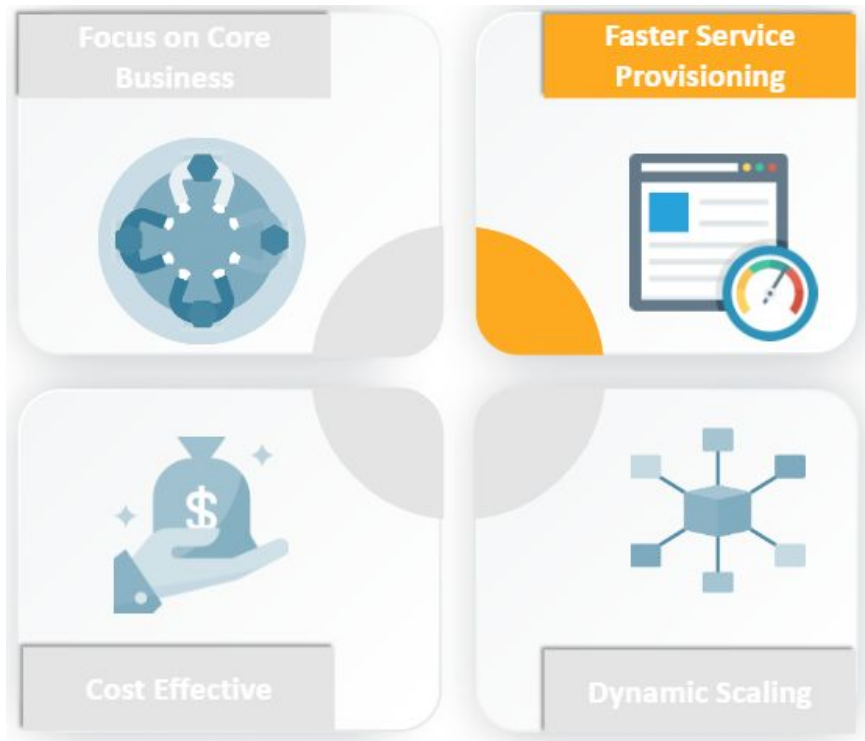
Why Cloud Computing

- Traditionally companies had to spend a lot of time and money on infrastructure, hardware and other operational aspects.
- But using cloud all these troubleshootings are managed by cloud service providers, so companies can now focus on their core business



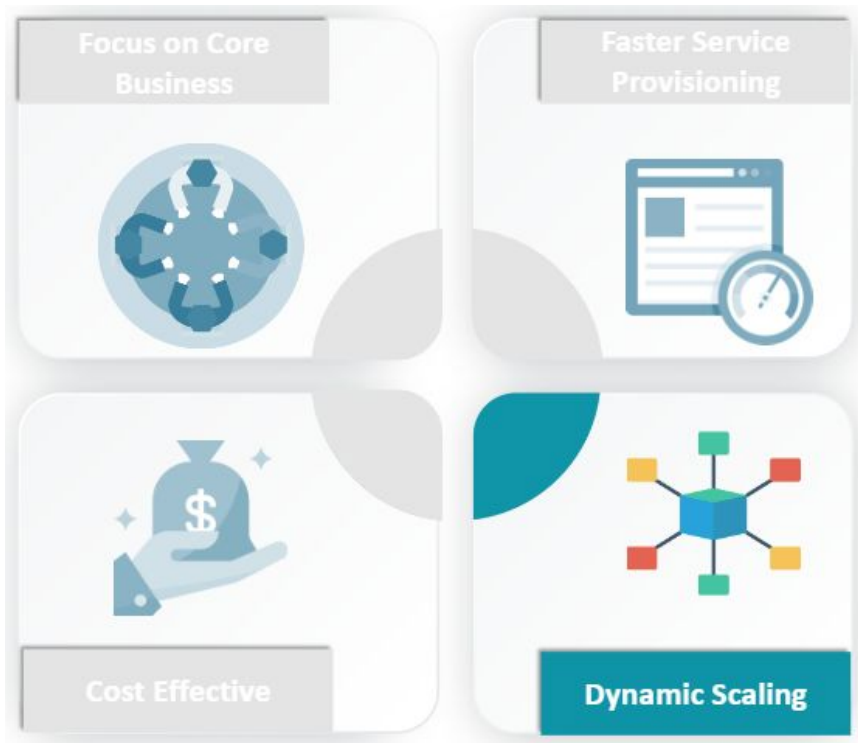
Why Cloud Computing (cont)

- Cloud enables the developers to get their applications to market quickly
- As long as there is internet connection available, users can access their required applications



Why Cloud Computing (cont)

- Cloud based services are ideal for business with growing and fluctuating bandwidth demands
- As it is flexible you can easily scale up or down the cloud capacity as per requirement



Why Cloud Computing (cont)

- Cloud computing reduces the cost of managing and maintaining IT systems
- Cloud service providers offer resources and using these resources rather than expensive systems and equipments can save a lot of money
- Cloud computing reduces both operational expense (OPEX) and capital expense (CAPEX)

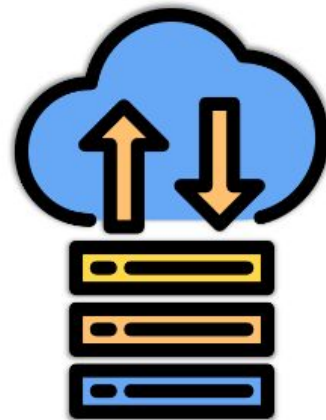


Intro to Cloud Computing: Cloud Storage

“Cloud Storage provides the functionality to store data remotely over the cloud or internet.”

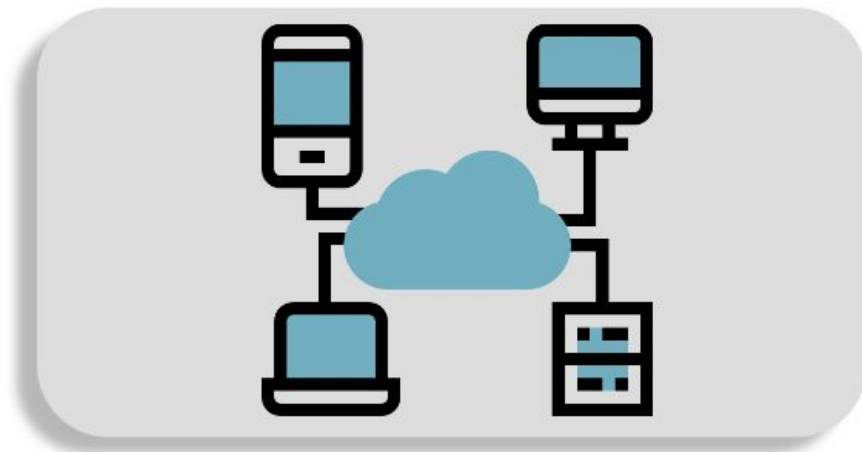
It helps in storing data such as :

- Documents,
- Pictures,
- Videos ,
- Any media



Cloud Computing

“Cloud computing can be simply understood as anything that involves delivering hosted services over the Internet.”



Benefits of Cloud Computing

01 Frequent and Easy Collaboration

If working in a team, it becomes very easy to configure and manage documents.

02 Security

Don't need to worry about data even if the hardware or system gets corrupted or destroyed or stolen, data will remain secured and safe, and it can be deleted also if confidential.

03 Scalability

You can scale up or down the storage according to your need of managing data.

04 Ease Of Software Updates

Software updates are taken care of by the suppliers which saves lot of time.

Types of Cloud Computing

Cloud Services can be described by two different ways:

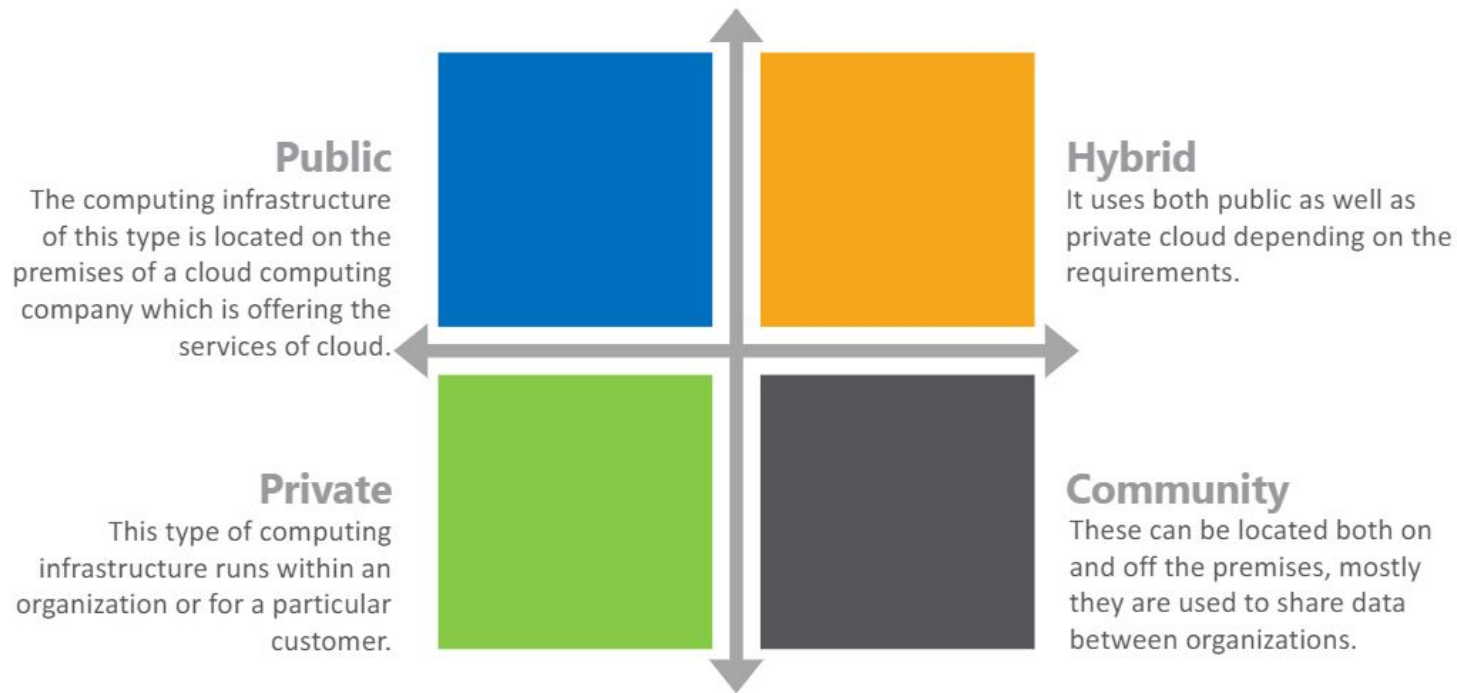
Based on location of cloud



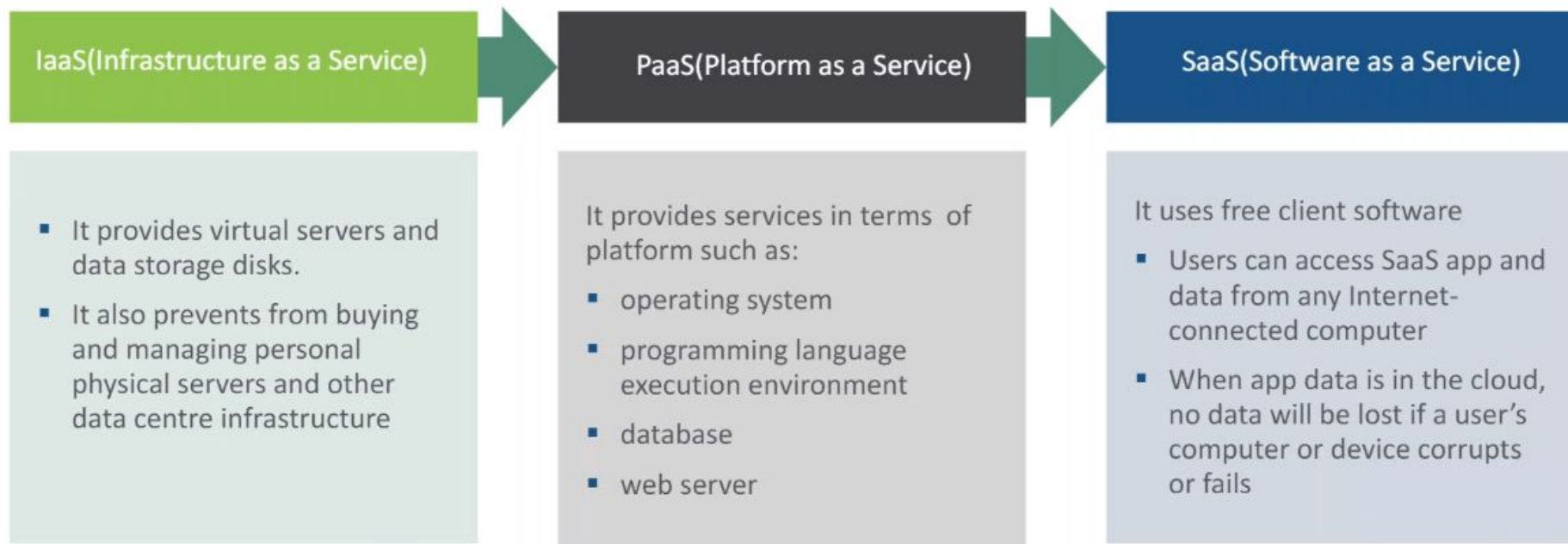
Based on Services provided by cloud



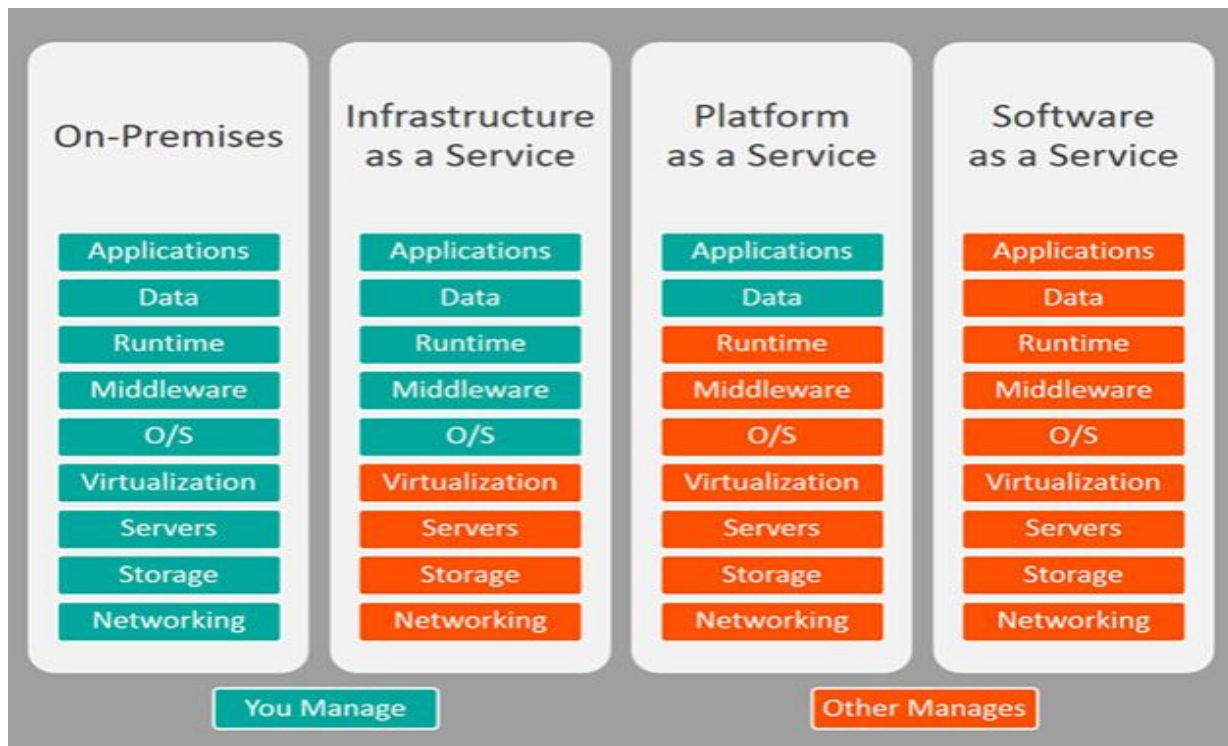
Cloud Computing Base On Location



Cloud Computing Based On Services



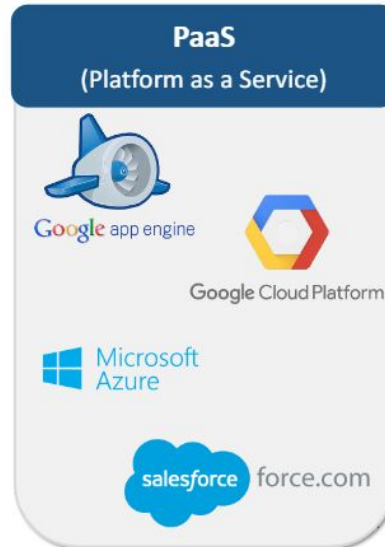
IaaS vs PaaS vs SaaS



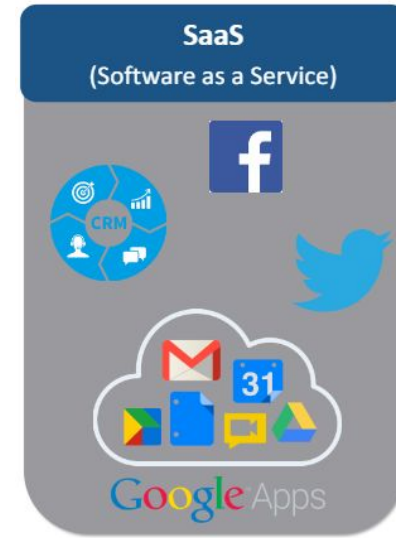
IaaS PaaS SaaS Examples



CSP provide VM, IT infra, storage, network & OS to their customers as a service and customers pay them as per use

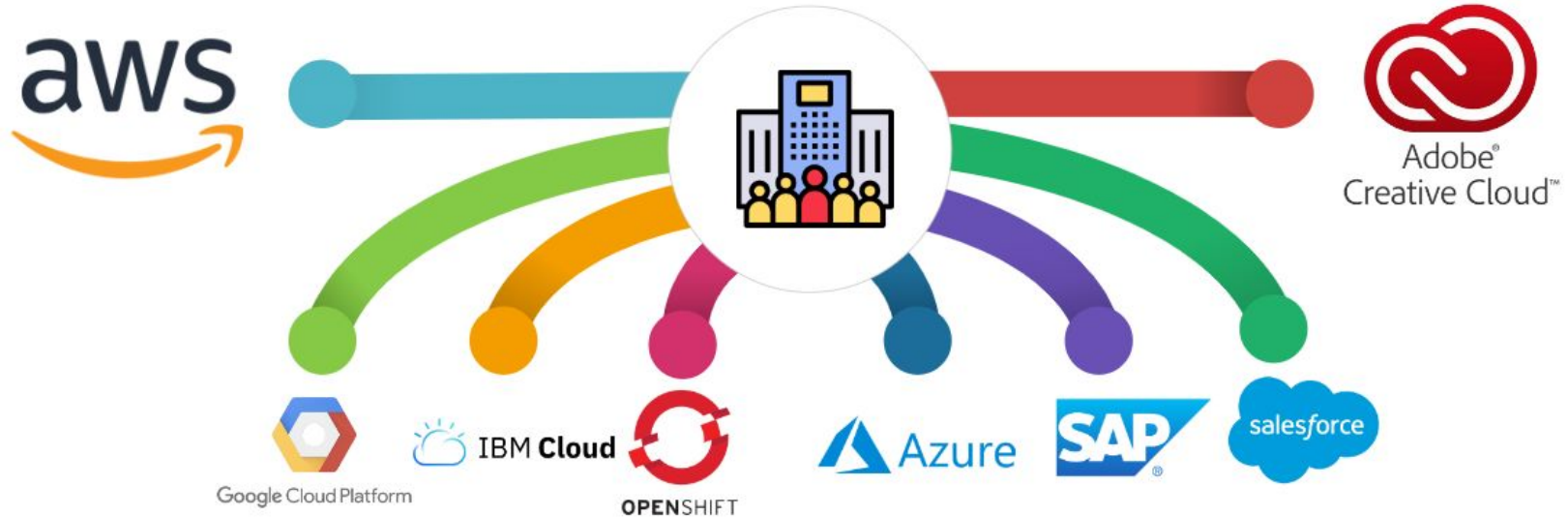


CSP provide on demand environment for testing, developing and delivering applications. Make it easy for developers to quickly create web and mobile app ready for market



Deliver SW app over internet, on demand based on subscription. CSP manage underlying Infra and handle SW updates and patches. User only connect to App via PC or Mobile

Top Companies Providing Cloud Computing Services



Service Comparison: AWS, Azure and GCP

Amazon AWS	Microsoft Azure	GCP
S3	Blob Storage	Storage
EC2	Virtual Machines	Compute Engine
EC2 Container Service	Container Service	Kubernetes Engine
Elastic Beanstalk	Cloud Services	App Engine
DynamoDB	Cosmos DB	Bigtable
Redshift	SQL Database	BigQuery
Lambda	Azure Functions	Cloud Functions
DynamoDB	Cosmos DB	Cloud Datastore

DevOps on Cloud

- DevOps and Cloud Computing go hand in hand. They can be implemented individually but are quite efficient and effective when joined together
- They can help an organization deliver new software features much faster and in a very efficient manner
- Many organizations are trying to fix their application development processes by shifting from waterfall to DevOps, they realized that DevOps alone cannot solve all their problems
- DevOps and public and private cloud solutions are now evolving together and most organizations are extending DevOps process and automation into public and/or private cloud

How DevOps combined with Cloud Improve Productivity

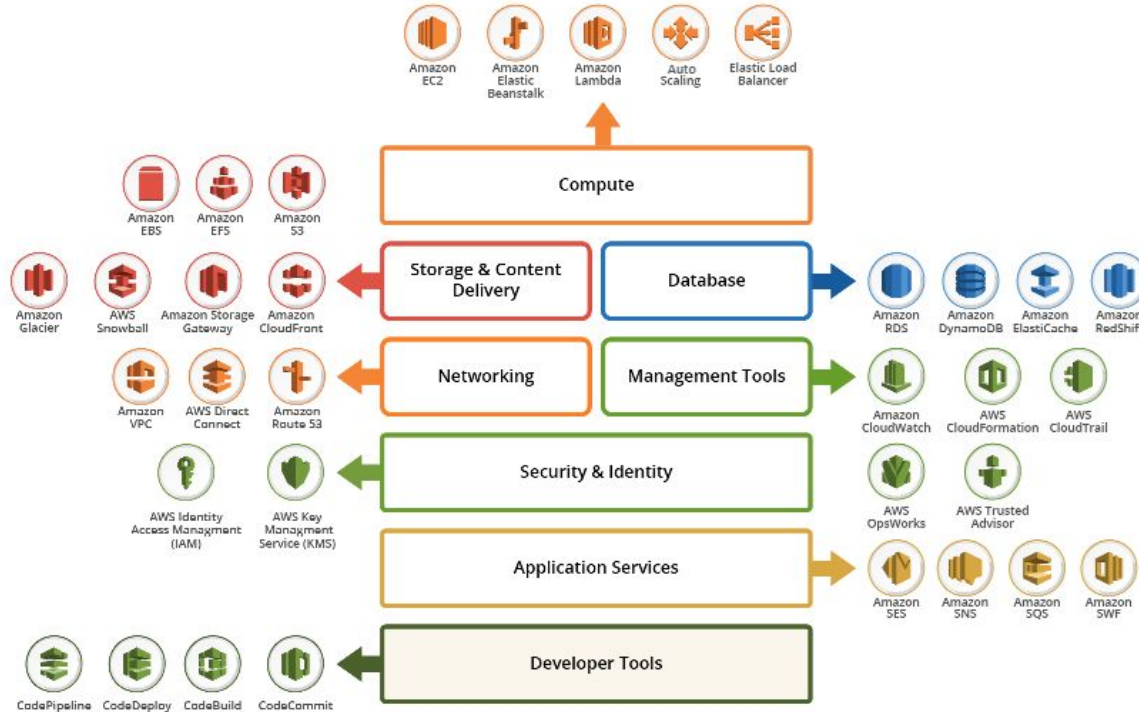
- Combined together they help bring products faster to market through quick access to development environment and streamline development process
- Infrastructure as code and automation together reduces cloud complexity and maintenance of servers and their resources
- Security is boosted with automated, repeatable processes that serve to eliminate error that can cause farther problem and more importantly build security controls from the very start
- Combined DevOps and Cloud Computing eliminate downtime and enhance scalability. Scalability is one of the most important factor for apps as they are developed. It also reduces the cost of infrastructure and increase global reach

Introduction to AWS

- AWS is a secure cloud service platform which offers compute power, database storage, content delivery and many other functionalities to help business scale and grow
- All over the world AWS technology is implemented at various server farms and is maintained by Amazon
- It provides 90 plus services
- Users are charged on a pay as you go model for resources used in Amazon



AWS Services



AWS Major Services



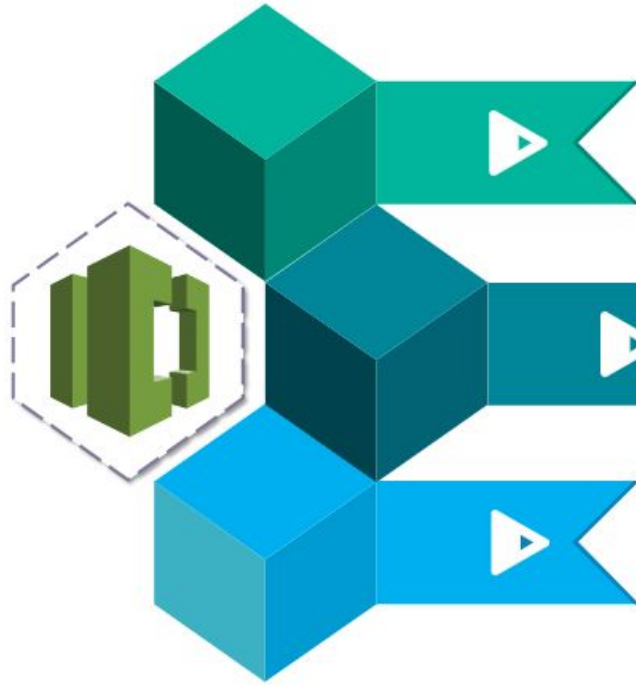
Important Steps Involved in AWS DevOps

DevOps when implemented on AWS, becomes lot more efficient and effective for a product lifecycle. Steps involved in AWS DevOps are mentioned below:

- Aws CodeCommit
- Aws CodePipeline
- Aws CodeDeploy



AWS CodeCommit



AWS CodeCommit is a fully managed source control service that hosts secure and highly scalable private **Git repository** without the need of operating the system

It is mainly designed for the **developers** who are supposed to **store** and **version** their code **securely** and **reliably**

Example- IT administrators to store their scripts and configurations; web designers to store HTML page and images etc.

AWS CodePipeline

AWS CodePipeline is a combination of **continuous integration** and **continuous delivery** service for quicker and more reliable infrastructure and application updates



It **automatically builds, tests and deploys** an user **code** whenever there is a **code change**, based on user defined release **process models**

It **integrates** with AWS services like **AWS CodeCommit**, **Amazon S3**, **AWS CodeDeploy**, **AWS Elastic Beanstalk**, **AWS OpsWorks**, and **AWS Lambda**



You can **configure** the pipeline with a **GUI** or **CLI**



AWS CodePipeline



Note: With AWS CodePipeline you have to **pay** only for what you **use**

AWS CodeDeploy

CodeDeploy is a service that coordinates your application deployment and updates across the fleet of AWS EC2 of any size

- 01 Automates code deployment to any instance
- 02 Handles the complexity of updating your instances
- 03 Avoid downtime during application deployment
- 04 Rolls-back automatically if failure detected
- 05 Integrates with third-party tools and AWS



AWS CodeDeploy

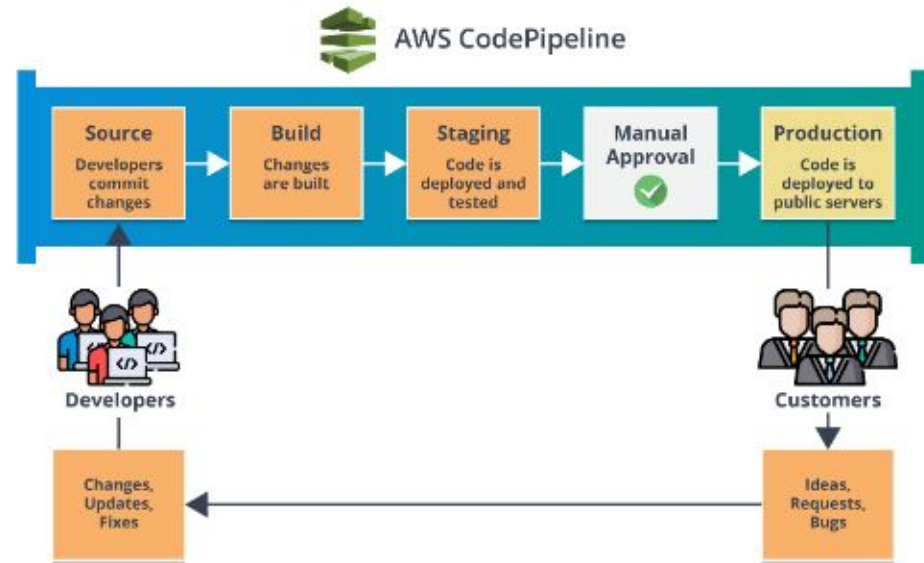
Why Should We Use CodePipeline?

By automating your software build, test, and release processes, AWS CodePipeline enables you to increase the speed and quality of your software updates by running all new changes through a consistent set of quality checks



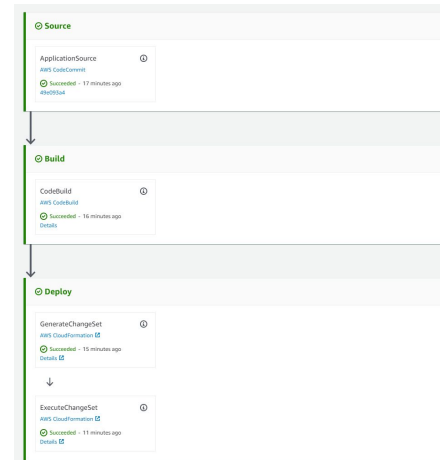
How does CodePipeline Work?

1. When a developer completes working on the code, he **commits** it to source repository
2. AWS CodePipeline **automatically deletes** the changes and **builds** those changes
3. After that, the build code is **deployed** to the staging server for testing
4. From the staging server, AWS CodePipeline runs **additional test**, such as **integration** or **load tests**
5. Once all tests are done, if code receives **manual approval** then AWS CodePipeline deploys the tested and approved code to **production instances**



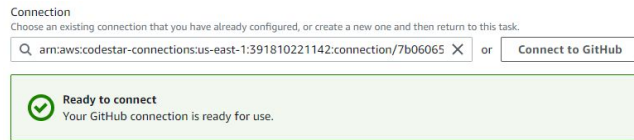
Create a CI/CD pipeline with AWS

- Select CodeStar from AWS console and create a new project (make sure your region is north virginia)
- Select a template - Select Python (Flask) using AWS Elastic Beanstalk
- Give a project name (example: demoproject), select codecommit as project repository and use default EC2 config
- In the next page review the steps and click on Create Project and wait for the project to complete provisioning
- Setup the Cloud9 IDE environment - Click on Set up AWS Cloud9 and assign an Environment name and click create
- Meanwhile browse the repository and it will go to CodeCommit - it is your code repository in Amazon
- Click on CodePipeline and you will see the entire CI/CD pipeline
- Click on EC2 to check your instances created



Create a Pipeline with CodePipeline

- Select CodePipeline from AWS console and click create pipeline
- Type a name for the pipeline, keep all settings as default and click next
- Add source as Github (version 2) then click Connect to Github and create a new connection
- A new window will pop up where you need to set a new connection name and click Connect to GitHub
- Click Install a new app and select your git repo then click save and click connect
- Select repository name and branch name as master and click next
- Select Build provider as AWS CodeBuild and click Build Project
- A new popup will appear and you will need to do the following:
 - Give a project name
 - Under Environment select operating system as Ubuntu, runtime as Standard and Image as Standard 5.0
 - Under Buildspec select Insert Build Command and type command “mvn -B package”
 - Click continue to CodePipeline
- Click next and skip deploy option and finally click create pipeline





Create a Pipeline with CodePipeline (cont)

- Under build you can click details to see the progress
- Your war file will be created and your new pipeline is now ready

mydemo-pipeline1

Source Succeeded
Pipeline execution ID: 3edb55cc-0c16-44dc-b54b-c01e3a7bfd2f

Source
GitHub (Version 2) ⓘ
Succeeded - 3 minutes ago
f74634d6 ⓘ
f74634d6 ⓘ Source: Update ContactService.java

↓ Disable transition

Build Succeeded
Pipeline execution ID: 3edb55cc-0c16-44dc-b54b-c01e3a7bfd2f

Build
AWS CodeBuild ⓘ
Succeeded - Just now
Details ⓘ
f74634d6 ⓘ Source: Update ContactService.java

```
8 04:22:23 Phase complete: BUILD State: SUCCEEDED
8 04:22:23 Phase context status code: Message:
8 04:22:23 Entering phase POST_BUILD
8 04:22:23 Phase complete: POST_BUILD State: SUCCEEDED
8 04:22:23 Phase context status code: Message:
```



Thank You!