



**TECHNION**

Azrieli Continuing Education and  
External Studies Division

# DevOps



**הטכניון**  
היחידה ללימודים המשך  
ולימודי חוץ ע"ש עזריאלי



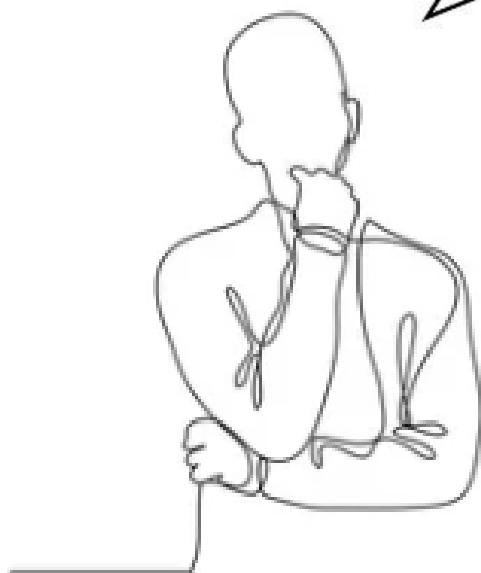


**What is  
DevOps?**

**Technologies  
Tools  
Principals**

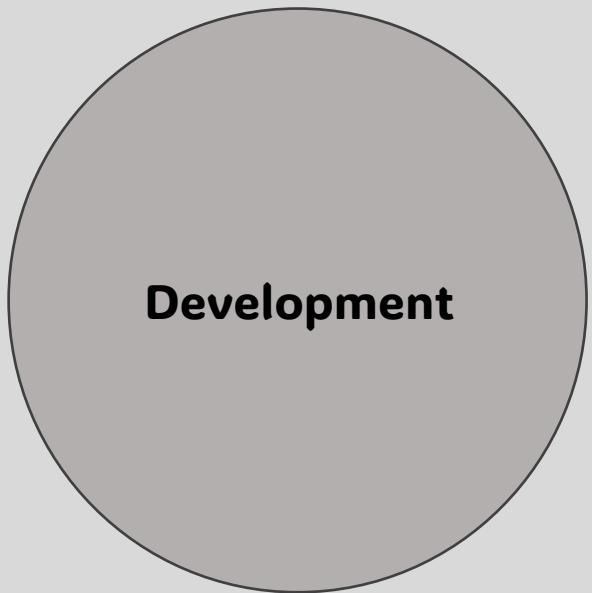


# What is DevOps?

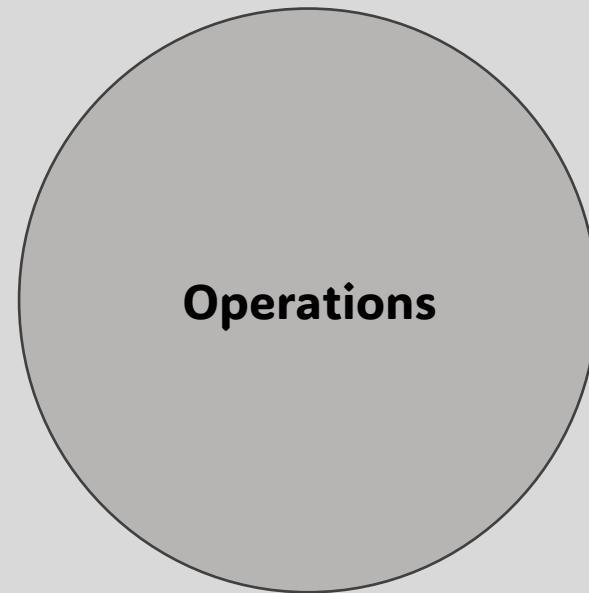


## TL;DR

DevOps is a collaborative approach that **integrates software development (Dev) with IT operations (Ops)** to streamline the software delivery process.



Dev  
Ops



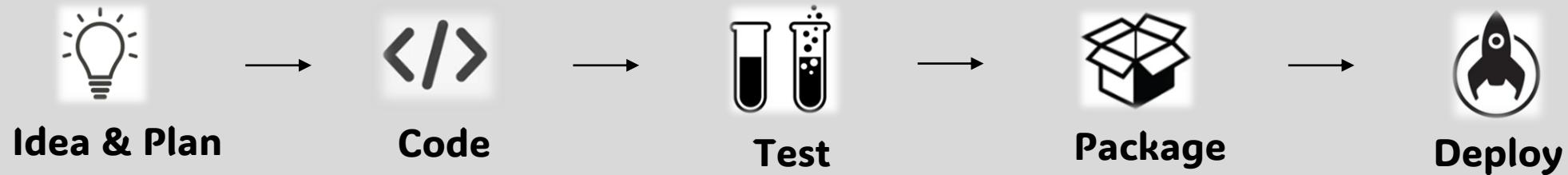


App



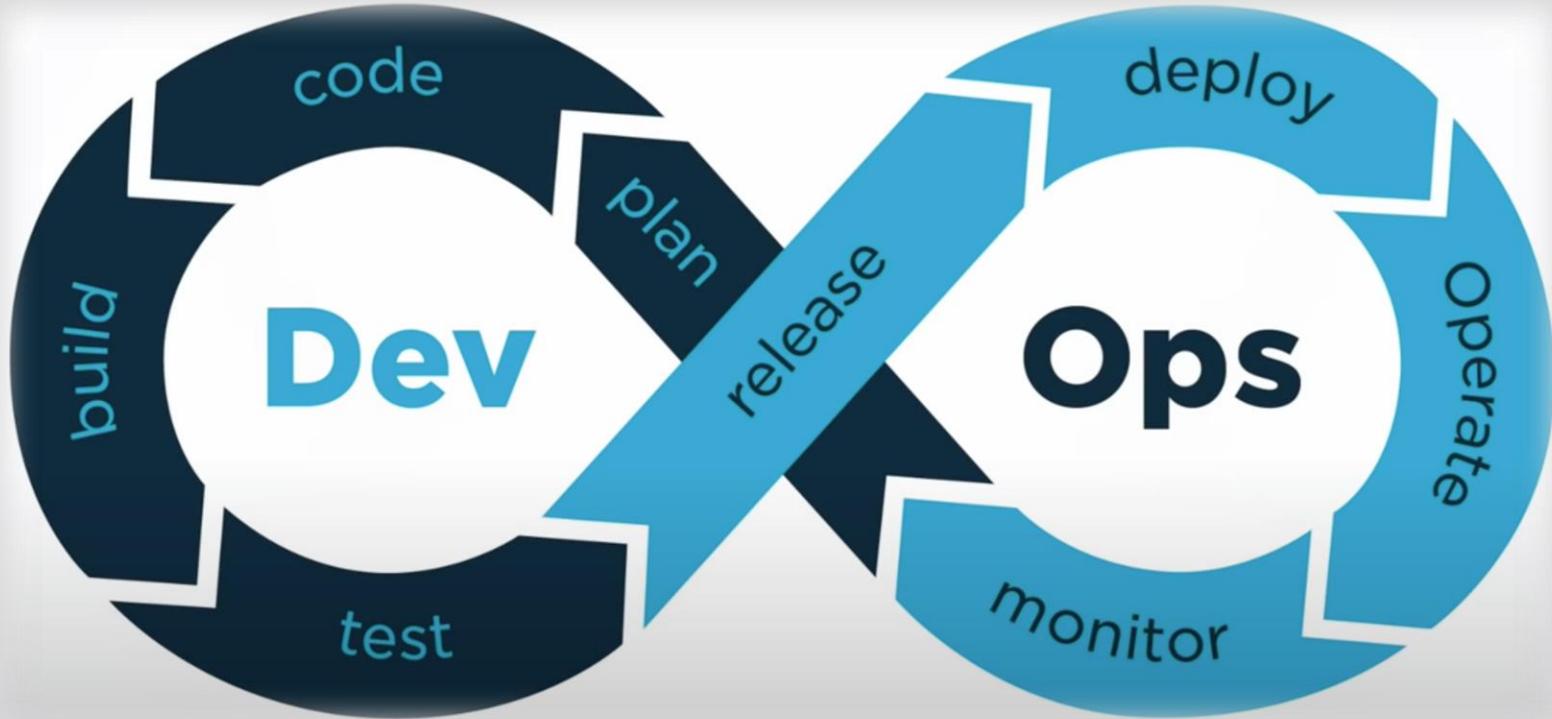
Users

## Process



## Infrastructure







CI/CD

Monitoring

Databases

Infrastructure

Security

Cost

Durability

Deployment

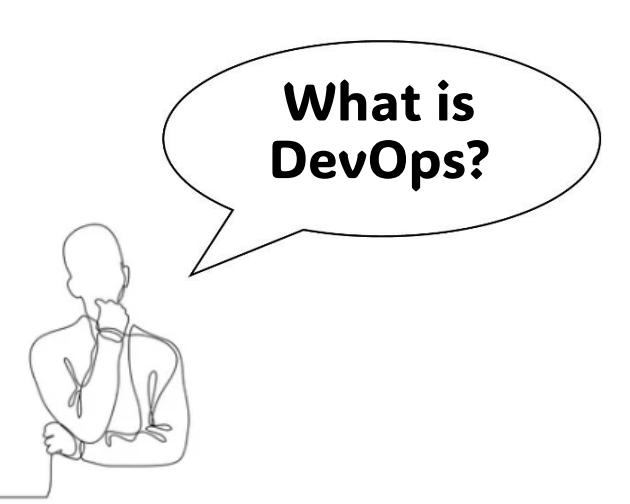
Versioning

Code

Networking

Platform

Automations

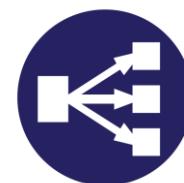




# NGINX



# aws



```
bash
$ env x='() { :}; echo vulnerable'
```



elasticsearch



What is  
DevOps?

# So Devops is...

Anything that creates the process of  
releasing the software **FAST** and with **HIGH QUALITY!**





# Let's Dive in...





## Development



## Code Repo

*Working with git.*

*Integrations for the developer.*



## Application

*Understanding the workflow.*

*Communication with other apps*



## Testing

*Automating the testing process*

*Building an artifact*



## Application



Servers Management



Servers - Infra



Linux & File System

```
File Edit View Search Terminal Help
mark@linux-desktop:~$ mkdir /tmp/tutorial
mark@linux-desktop:~$ cd /tmp/tutorial
mark@linux-desktop:/tmp/tutorial$ mkdir dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
mark@linux-desktop:/tmp/tutorial$ cd ~/Desktop
bash: cd: too many arguments
mark@linux-desktop:/tmp/tutorial$ ls
dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$
```

CLI

Shell & Bash



OSI

Communication Protocols

IP, DNS, HTTP/HTTPS,  
TCP/UDP, Ports, VPC,  
Subnets, Routing Tables

Load Balancers

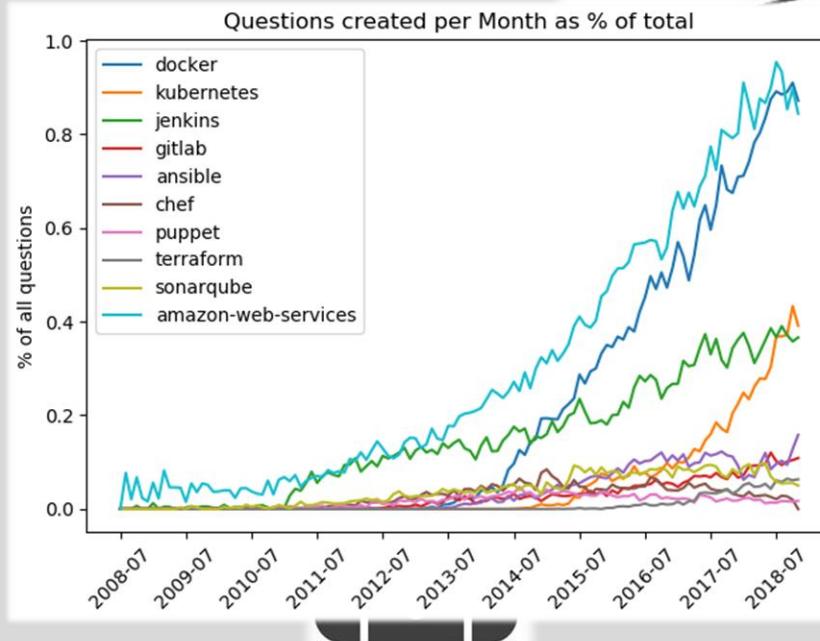
Web Servers, proxy/revers  
proxy

Firewall, Security Groups

SSL/TLS, Certificates



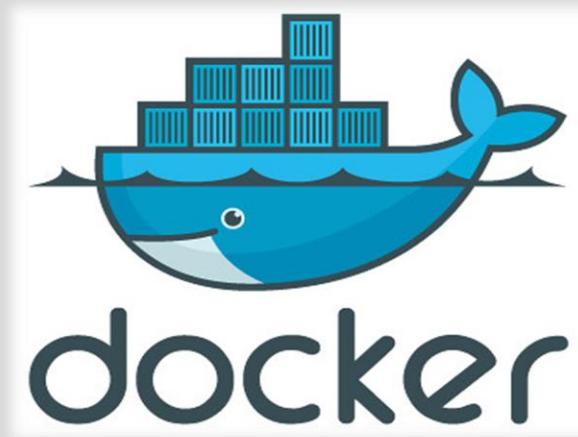
## Containers



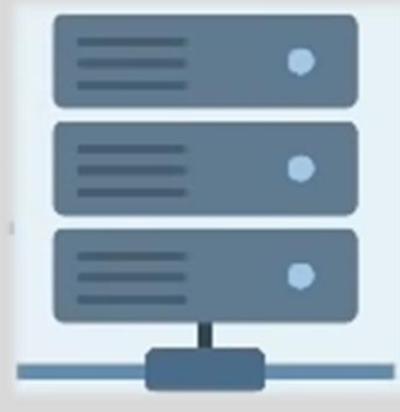
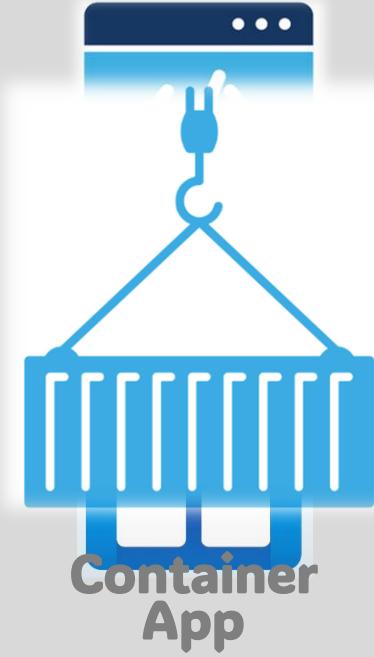
*containers to Build,  
Store and Run your  
Applications...*

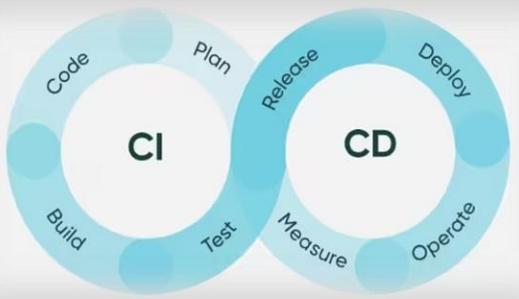


**Containers**

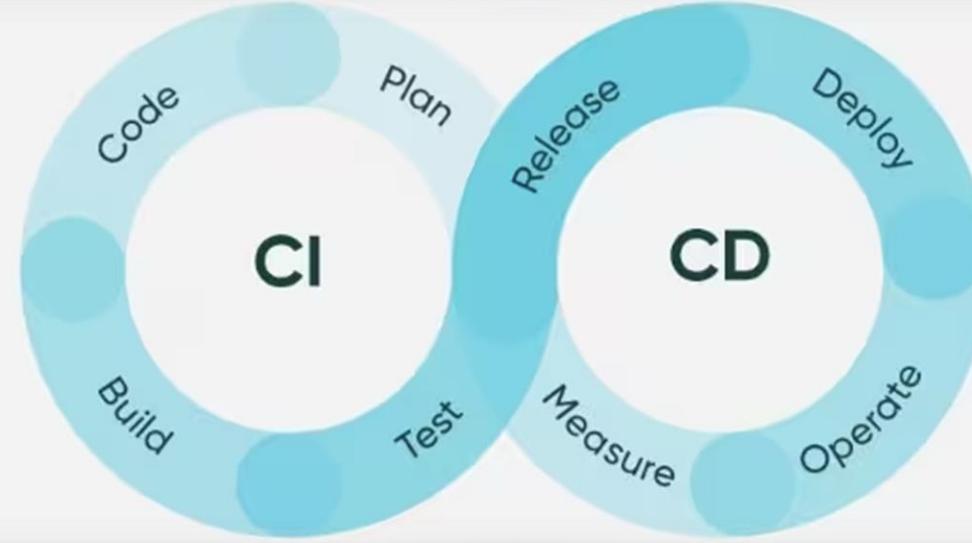


*A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.*

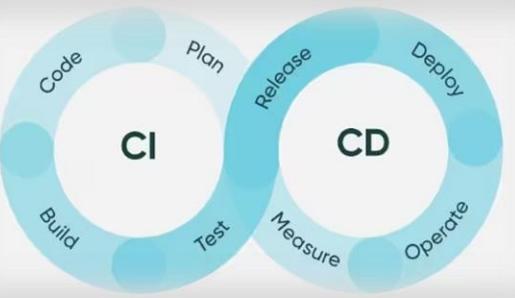




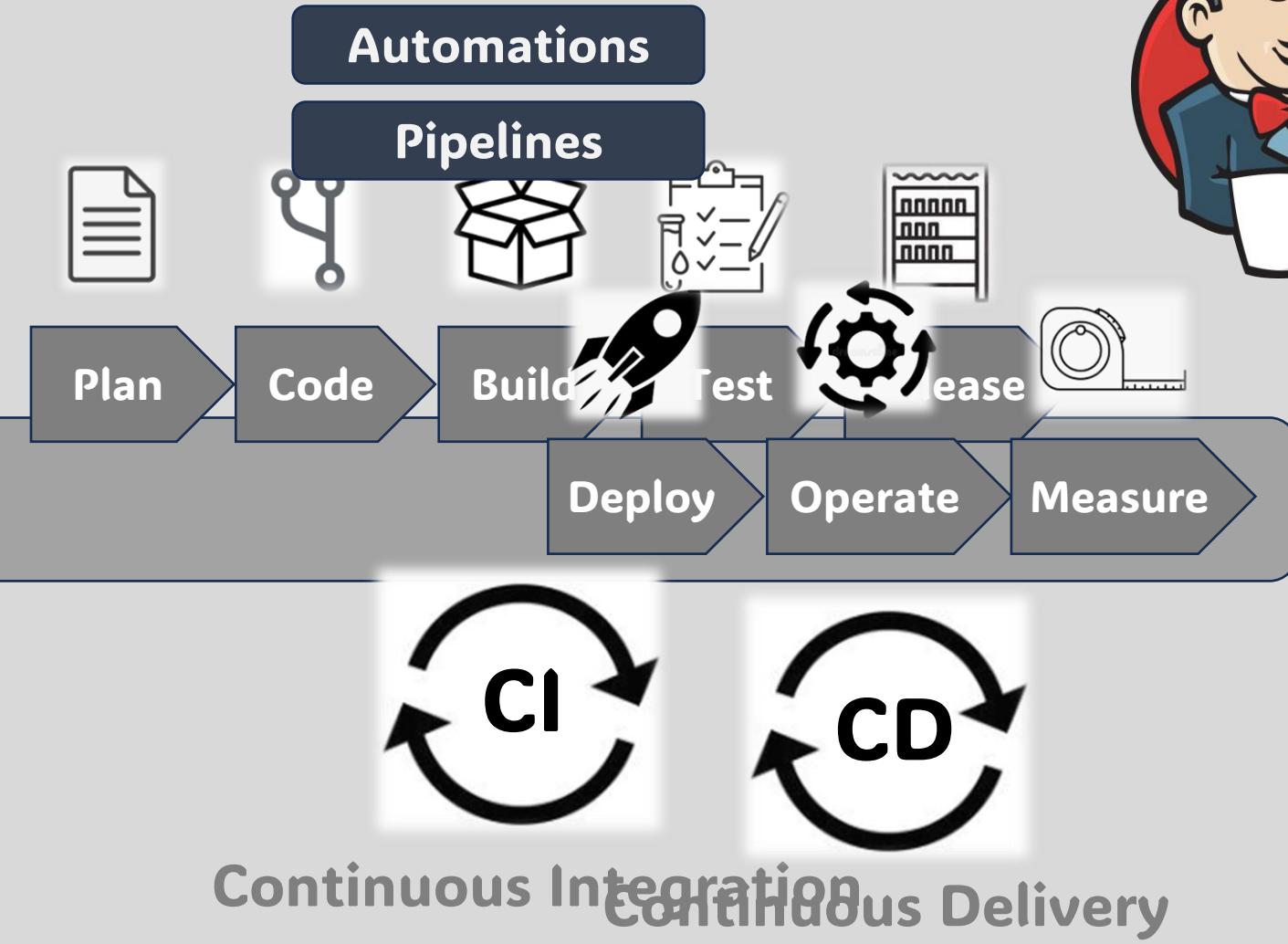
CI/CD & Automations



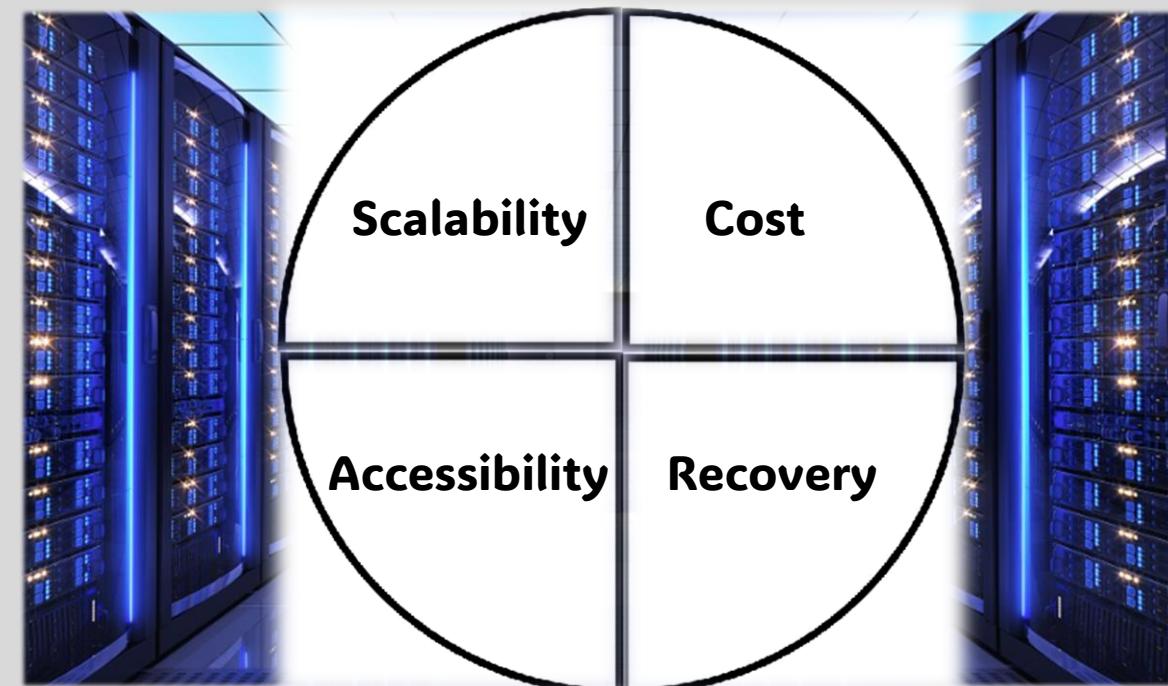
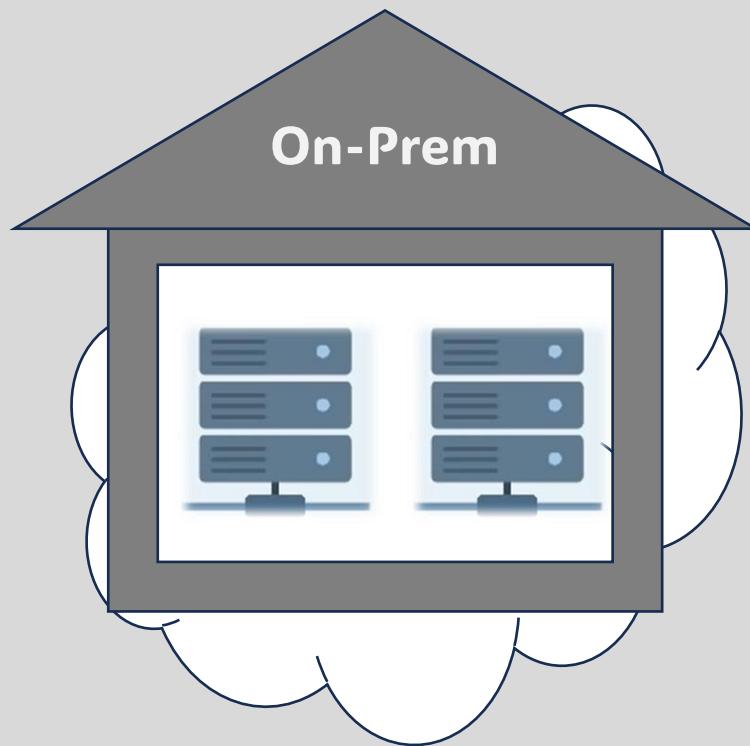
# CI/CD & Automations



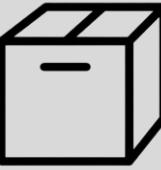
CI/CD & Automations



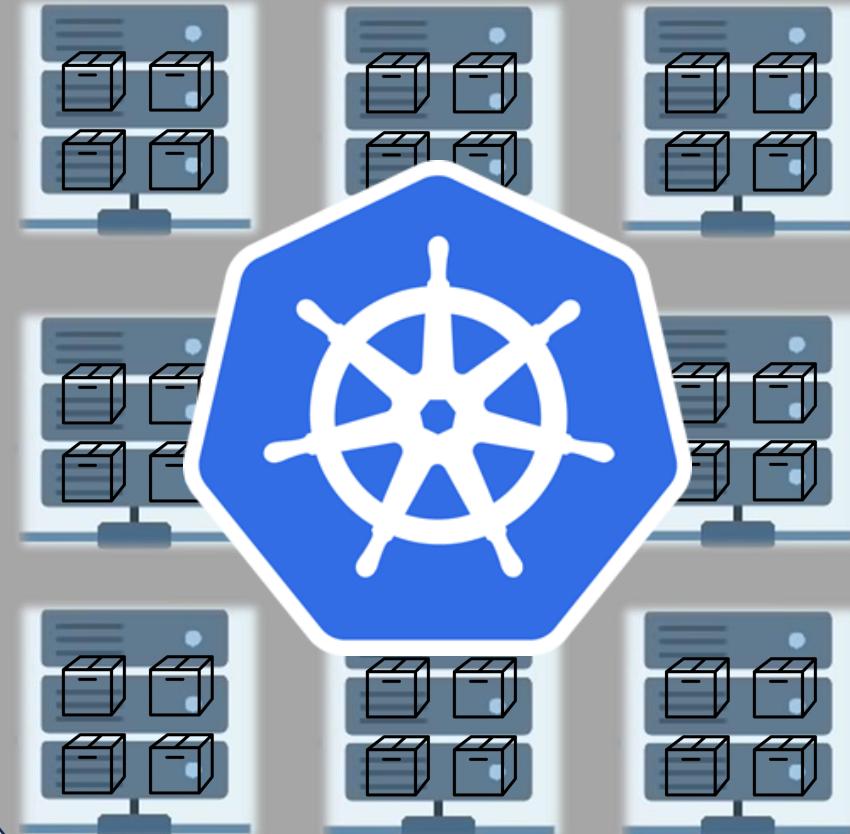
Cloud



## Container Orchestration



### Kubernetes Cluster

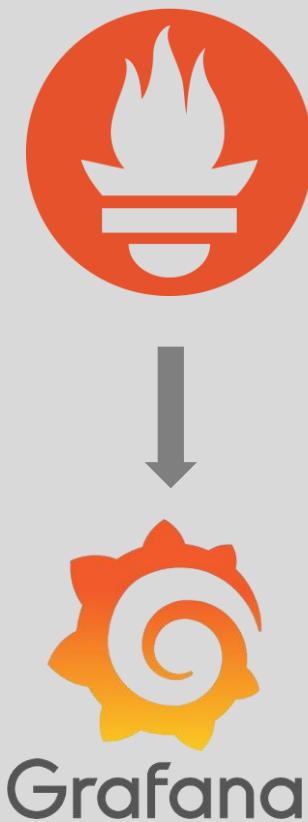


### Container Orchestrator

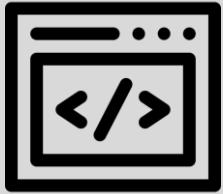
*Kubernetes (K8s) is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications.*



## Monitoring

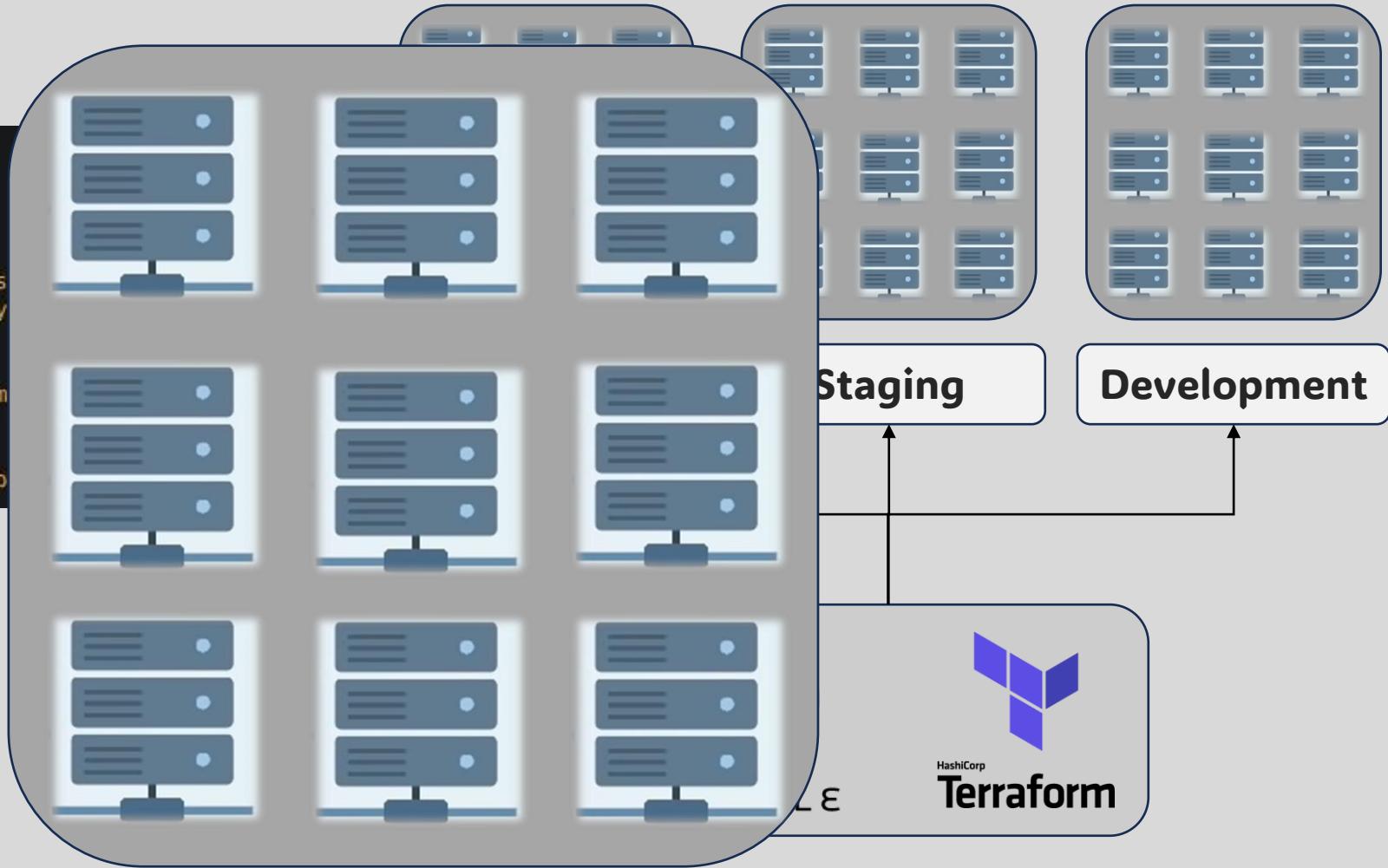


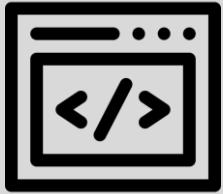
A screenshot of a UI kit interface. On the left, a sidebar shows "Main Nav" with "Dashboard" selected, followed by "Pages", "Components", and "UI Elements" which is expanded to show "Alerts", "Buttons", "Cards", "Carousel", "Dropdowns", "Grid", "Images", "Lightbox", and "Modals". In the center, there is a search bar and a section titled "Alerts" under "UI ELEMENTS → ALERTS". This section contains a list of "Primary Alerts" (Info Alert, Success Alert, Warning Alert, Danger Alert) and "Secondary Alerts" (Success Alert, Warning Alert, Danger Alert, Light Alert, Dark Alert). Each alert item has a close button. The background features a grid of server icons with some having red "X" marks indicating errors or issues.



# Infrastructure as Code

```
1 ## ec2.tf
2 provider "aws" {
3     access_key = "<aws_access
4     secret_key = "<secret_key"
5     region = "<aws_region"
6 }
7 resource "aws_instance" "exam
8     count = 5
9     ami = "ami-v1"
10    instance_type = "t2.micro"
```





# Infrastructure as Code

Infrastructure Provisioning



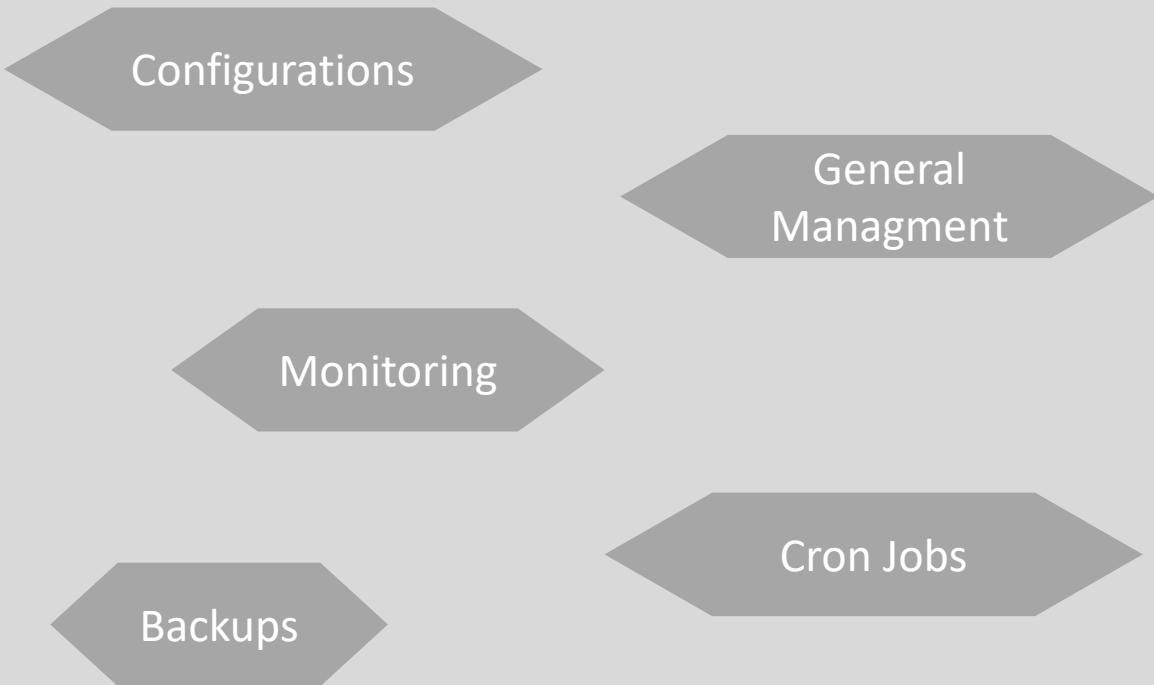
ARM Template

Configuration Management



< / >

## Scripting



### OS Scripting



### General Languages

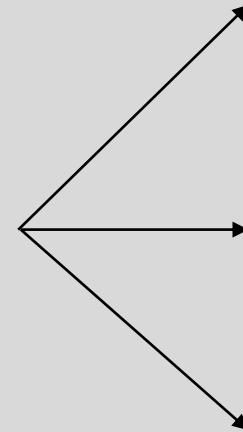


# Version Control

```
1 import numpy as np
2
3 with np.load("A") as T:
4     import random as rnd
5
6     #ask user for number of dice
7     nDice = input("How many dice are we playing with? \n")
8     nSides = input("\nhow many sides each dice has? \n")
9
10    X = msg = "we're roll ●●●"
11
12    try:
13        import lasio
14        def load_las_file(filename, start_depth=None):
15            f = lasio.read(filename)
16            d = f.dfs()
17
18            if start_depth is not None:
19                data = d[d.index > start_depth]
20
21            return d
22
23        def clay_volume(gamma_ray, gr_shale, gr_clean):
24            gr_index = (gamma_ray - gr_clean) / (gr_shale - gr_clean)
25            result = gr_index * 0.6
26
27            if result < 0:
28                result = 0
29            elif result > 1:
30                result = 1
31
32            return result
33
34        if __name__ == '__main__':
35            well_data = load_las_file('15-9-19_SR_COMP.las', start_depth=4310)
36            print(well_data.head())
37
```



Git



GitLab

 Bitbucket



Github

# So DevOps is ....

Development

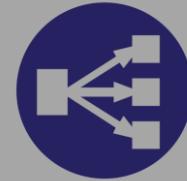


*Learning*

Operating Systems



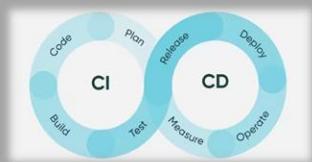
Security & Networking



Containers



CI/CD & Automations



Cloud



Container Orchestration



Monitoring



Infrastructure as Code



Scripting & Version Control





## DevOps:

*"Because sometimes even servers need a good therapist to talk about their feelings."*