

# Devops Skills Roadmap

Mohammed Oshari, Omar Elhaj

# Agendas

- Introduction - Omar
- OSI Model Understanding - Omar
- System Admin - Oshari
- Web Servers - Oshari
- Scripting - Omar
- Database Management - Omar
- Source Code Management - Oshari
- Containerization - Oshari
- Containers Orchestration - Oshari
- Cloud Computing - Omar
- Continuous Integration/ Continuous Deployment/Delivery (CI/CD) - Omar
- Infrastructure as Code (IaC) - Omar
- Secrets Management - Oshari
- Observability - Oshari

# Introduction

Problem?

Gap between Developers and Operations Teams

Why Devops?

- Enforce Reliability into code by Automation
- Promote Collaboration
- Remove Blind Spots between teams
- Secure Applications



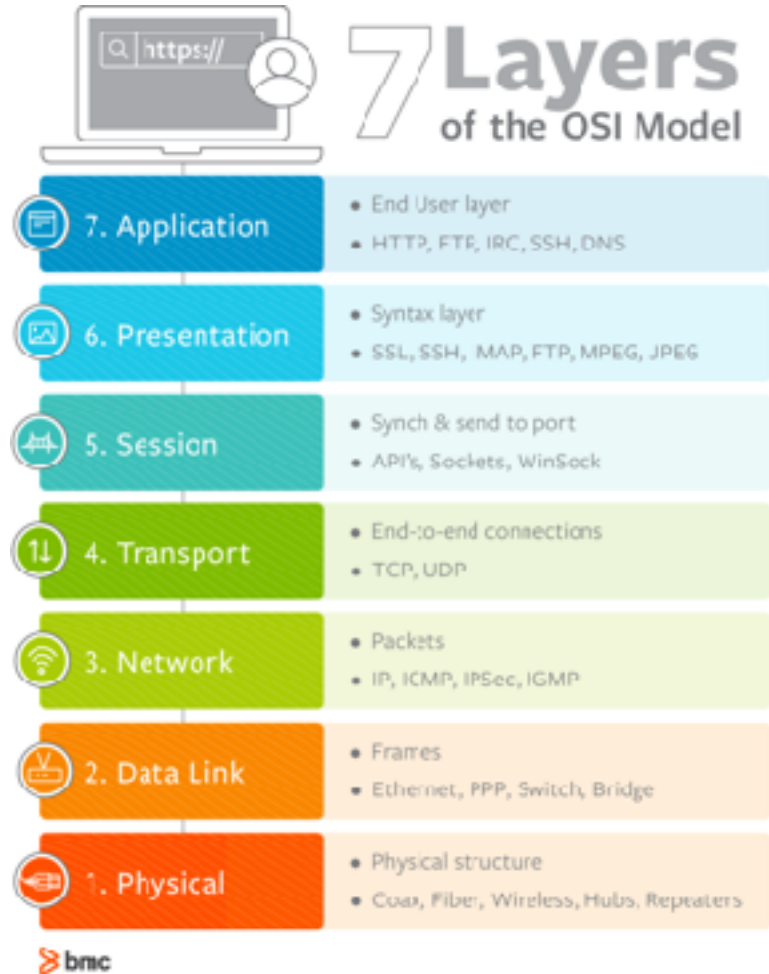
# OSI Model

What?

Conceptual View that standardize NW communications into **seven** layers

Why you need it as an Engineer?

- Troubleshooting

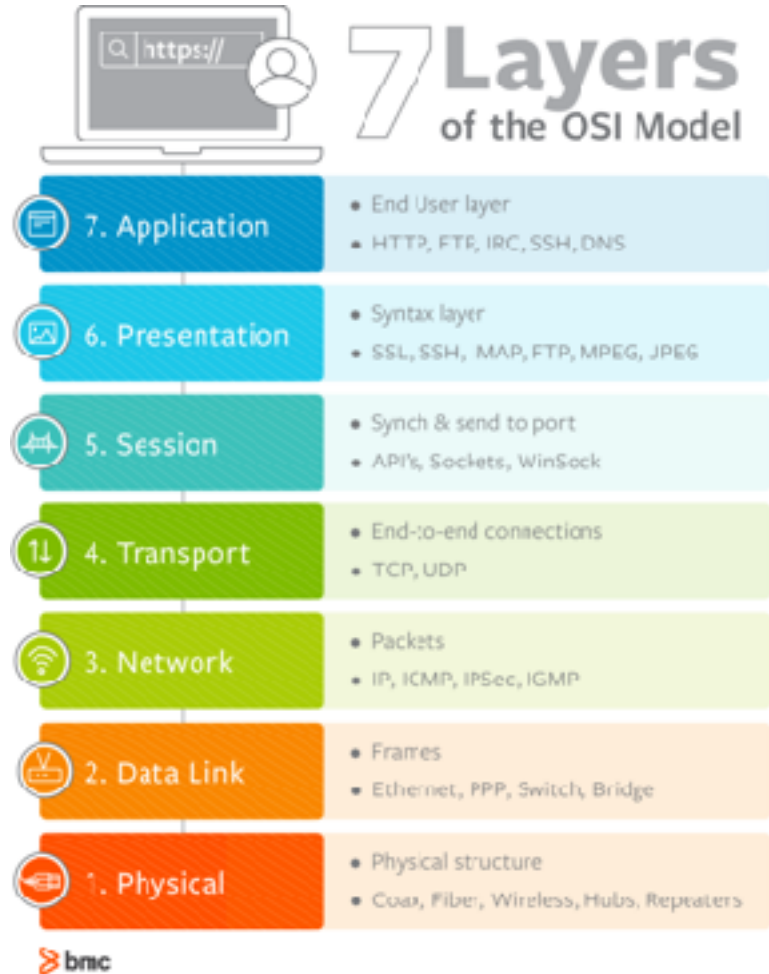


# OSI Model

## Troubleshooting Tools

- telnet (Layer 4)
- curl (layer 7)
- nmap (Layer3)
- traceroute (Layer3)
- Ping (Layer3)

Skills Weight - **Important**



# Systems Administration

What?

The skill of managing, configuring, securing, and troubleshooting Linux/Windows based systems

Business Value?

- Minimal and high performing Operating Systems for Applications Runtime
- Open Source, and Enterprise options
- Wide Coverage for Resources Skills to maintain

# Systems Administration

Famous Operating Systems:

Unix (FreeBSD), Linux (Redhat, Debian, SUSE),  
Windows (Windows Server)

Weight - **Very Critical**



# Systems Administration

```
0[||||| 15.0%] 3[|| 0.7%] 6[||||| 15.3%] 9[||||| 18.4%]
1[|| 5.8%] 4[||||| 14.2%] 7[|| 6.6%] 10[||||| 13.4%]
2[||||| 19.1%] 5[|| 2.6%] 8[||||| 12.5%] 11[|| 4.0%]
Mem[||||| 4.966/15.46] Tasks: 160, 962 thr: 2 running
Swp[ 0K/4.006] Load average: 1.39 1.46 1.28
Uptime: 01:03:55
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1277	rtkit	RT	1	150M	1552	1332	S	0.0	0.0	0:00.01	/usr/libexec/rt
3865	sagar	-21	-11	32032	5400	4692	S	0.0	0.0	0:00.00	/usr/bin/pipewi
3882	sagar	-21	-11	264M	18628	13416	S	0.0	0.1	0:00.00	/usr/bin/wirepl
3883	sagar	-21	-11	155M	15848	9140	S	0.0	0.1	0:04.67	/usr/bin/pipewi
3886	sagar	-21	-11	82148	22200	7420	S	0.0	0.1	0:01.67	/usr/bin/pipewi
4780	sagar	-11	5	12.4G	676M	258M	S	0.0	4.3	0:02.37	firefox
9979	sagar	-11	5	3028M	497M	112M	S	0.0	3.2	0:00.46	/usr/lib/firefo
3848	sagar	9	-11	155M	15848	9140	S	0.0	0.1	0:04.92	/usr/bin/pipewi
3849	sagar	9	-11	32032	5400	4692	S	0.0	0.0	0:00.00	/usr/bin/pipewi
3851	sagar	9	-11	82148	22200	7420	S	0.7	0.1	0:12.48	/usr/bin/pipewi
3855	sagar	9	-11	264M	18628	13416	S	0.0	0.1	0:00.69	/usr/bin/wirepl
3887	sagar	9	-11	264M	18628	13416	S	0.0	0.1	0:00.00	/usr/bin/wirepl
3893	sagar	9	-11	264M	18628	13416	S	0.0	0.1	0:00.00	/usr/bin/wirepl
3940	sagar	15	-5	1836M	99828	62936	S	7.2	0.6	1:47.09	/usr/lib/xorg/X
3941	sagar	15	-5	1836M	99828	62936	S	0.0	0.6	0:00.00	/usr/lib/xorg/X

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice - F8Nice + F9Kill F10Quit



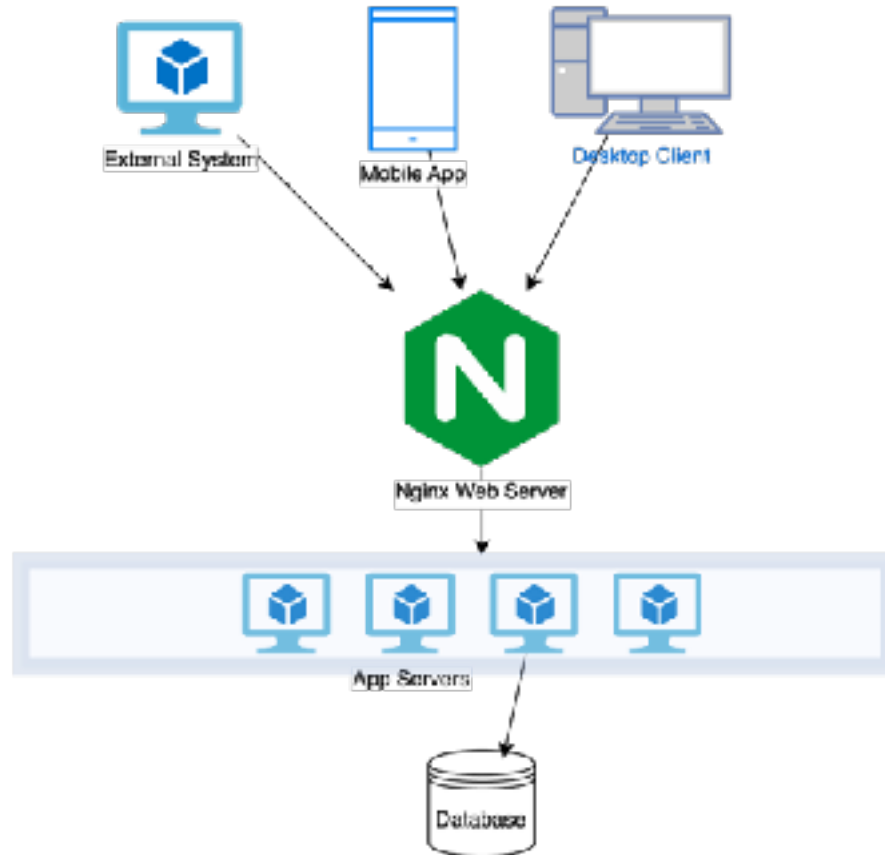
# Systems Administration



## Top 50 Linux Commands you must know

- |           |            |             |                      |                           |
|-----------|------------|-------------|----------------------|---------------------------|
| 1. ls     | 11. cat    | 21. diff    | 31. kill and killall | 41. apt, pacman, yum, rpm |
| 2. pwd    | 12. echo   | 22. cmp     | 32. df               | 42. sudo                  |
| 3. cd     | 13. less   | 23. comm    | 33. mount            | 43. cal                   |
| 4. mkdir  | 14. man    | 24. sort    | 34. chmod            | 44. alias                 |
| 5. mv     | 15. uname  | 25. export  | 35. chown            | 45. dd                    |
| 6. cp     | 16. whoami | 26. zip     | 36. ifconfig         | 46. whereis               |
| 7. rm     | 17. tar    | 27. unzip   | 37. traceroute       | 47. whatis                |
| 8. touch  | 18. grep   | 28. ssh     | 38. wget             | 48. top                   |
| 9. ln     | 19. head   | 29. service | 39. ufw              | 49. useradd               |
| 10. clear | 20. tail   | 20. ps      | 40. iptables         | 50. passwd                |

# Web Servers and Reverse Proxies



# Web Servers and Load Balancers

What?

Software that is responsible for Serving Web Content (Web Sites),  
Hosting APIs, HTTPs, Reverse Proxying, Load Balancing

Business Value?

- Communication Encryption
- Applying Authentication/ Authorization
- Load Balancing
- High Availability Setup
- Customized Routing Layer 7 Requests

# Web Servers and Load Balancers

Famous Reverse Proxies:

Nginx, Apache, Microsoft IIS, Cloud Resources



Weight - Important

# Scripting

```
1  #!/bin/bash
2  # This Works with Gitlab CI/CD as shell jobs
3
4  curl -s -X POST https://api.telegram.org/bots{myBotToken}/sendMessage -d chat_id={myGroupChatId} -d parse_mode="HTML" -d text=""
5  New code pushed to <b>SCI_COMMIT_BRANCH</b> branch! ✓
6  <b>Project</b> 🌿: $CI_PROJECT_NAME
7  <b>Committer</b> 👤: $GITLAB_USER_NAME
8  <b>Revision ID</b> 📄: $CI_COMMIT_SHORT_SHA
9  <b>Commit Message</b> 💬:
10  $CI_COMMIT_MESSAGE
11
12  Check code changes at the following commit URL 🔗:
13  $CI_PROJECT_URL/-/commit/$CI_COMMIT_SHA
14  ""
```

# Scripting

What?

Writing short programs (scripts) to automate tasks, often using languages like Bash, Python, or PowerShell.

## Typical Use Cases

- Database Backups and Archiving
- Logs Rotation
- CI/CD Customized Jobs
- Reducing Human Error

# Scripting

Top Scripting Languages:

Bash, Python, Powershell, CMD Batch

Weight - **Important**

Note:

Once you work a bit on OS skills,  
learning this skill will make more sense

# Database Management

What?

The process of storing, organizing, and accessing data using structured/ unstructured systems like relational or NoSQL databases.

Business Value/Real Life:

- Data
  - User Information in Facebook
  - Patients Information in Hospitals Systems
  - etc



# Database Management

## Famous Technologies

SQL/Relational Databases:

MySQL, PostgreSQL, Oracle, MSSQL, Cloud Resources

NoSQL Databases:

MongoDB, Redis, Cloud Resources

Expected from you:

- Provisioning and Maintenance
- Basic Level Queries
- Backups and Archiving
- Replication

Weight: **Normal**



# Source Code Management

What is SCM?

Source Code Management is a practice where developers can collaboratively write code and maintain a stable code base, and it is typically the starting point of CI/CD pipelines

Why we need it?

- Where's your change
- When did you change this
- We don't want this feature now, let's code it later
- I want to review your code changes
- I want to see the history of the changes in the code

# Source Code Management

Famous Technologies:

Git, Gitlab, Github,  
Bitbucket

Best Practice:

Gitflow Branching Strategy,  
Push Policy, Protect main  
branches

Weight: **Critical**



# Terms

Repo

Pull / Clone

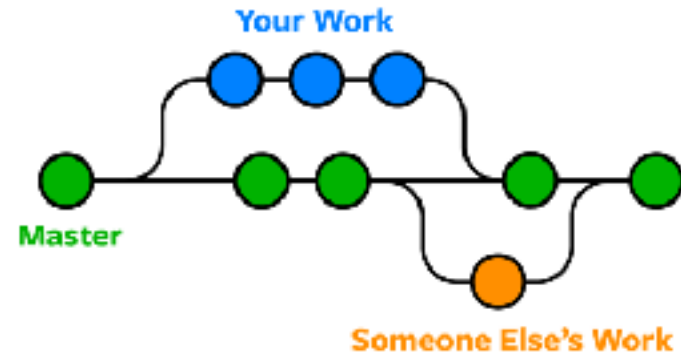
Commit

Push

Branch

Merge

Pull Request/Merge Request



# Who, When, What?

The screenshot displays a GitHub pull request interface. The top bar shows the repository name 'Use IsRepositoryWithGitHubRepository Helper' and the branch '1dy-dev'. The left sidebar contains a list of commits under the 'History' tab, including 'Use IsRepositoryWithGitHubRepository Helper' and 'consolidate repo rules checks, add from plan che...'. The main area shows a code diff for the file 'app/src/main/java/com/example/.../rules.ts'. The diff highlights changes in the 'import' statements, with a red background for the removal of 'import { Repository } from \'.../models/repository\'' and a green background for the addition of 'import { IsRepositoryWithGitHubRepository } from \'.../models/repository\''. The right sidebar shows the 'Changes' tab with a list of files, including 'app/src/main/java/com/example/.../rules.ts' and 'app/src/main/java/com/example/.../store.ts'.

Changes

History

Select Branch to Compare...

Use IsRepositoryWithGitHubRepository Helper  
1dy-dev · 3 days ago

consolidate repo rules checks, add from plan che...  
windell · 3 days ago

get plan info from existing API call  
windell · 3 days ago

the first commit  
windell · 4 days ago

always display total num of rule failures  
windell · 5 days ago

add error/warning states to viewer, update colors  
windell · 5 days ago

use spacing was instead of hardcoded  
windell · 5 days ago

update CSS variable names  
windell · 5 days ago

Merge pull request #17985 from desktop(jb-link)...  
Jose Corina · 5 days ago

Roll back changes to link focus colors  
Jose Corina · 5 days ago

Merge pull request #17982 from desktop(release...

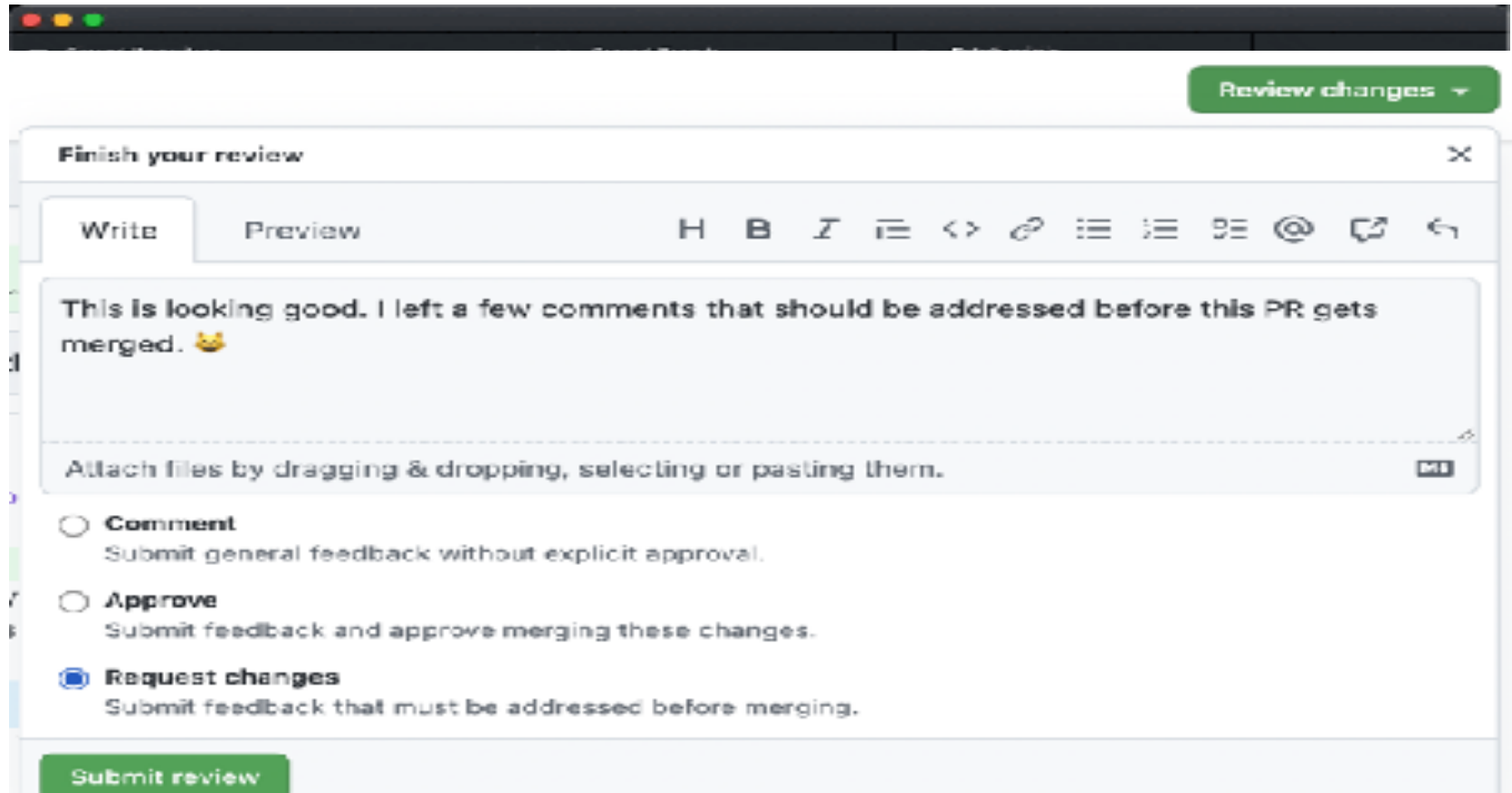
Use IsRepositoryWithGitHubRepository Helper  
1dy-dev · 2803183e7 ± 2 changed files +10 -19

app/src/main/java/com/example/.../rules.ts

app/src/main/java/com/example/.../store.ts

@@ -15,10 +15,18 @@ import {  
- import { enableRepoRules }  
+ from \'.../feature-flags\'  
+ import { supportRepoRules } f  
+ from \'.../endpoint-capabilities\'  
+ import { Account } from  
+ \'.../models/account\'  
- import { Repository } from  
+ \'.../models/repository\'  
- import { GitHubRepository } f  
+ from \'.../models/github-repositor  
+ y\'  
+ import {  
+ Repository,  
+ IsRepositoryWithGitHubReposi  
+ tory,  
+ } from \'.../models/repository  
+ \'  
+ }  
+ \* returns whether repo rules  
+ could potentially exist for the p  
+ rovided account and repository.  
+ @ -15,10 +17,18 @@ import {  
+ fromRepository } from \'.../mod  
+ els/github-repository\'

# Let me Review the Change before we take it live



The screenshot shows a web browser window with a dark title bar. The main content area is a pull request review form. At the top right, there is a green button labeled "Review changes" with a downward arrow. Below this, the form is titled "Finish your review" with a close button (X) in the top right corner. The form has two tabs: "Write" (active) and "Preview". The "Write" tab contains a rich text editor with a toolbar showing icons for bold, italic, link, unlink, list, and other formatting options. The text area contains the message: "This is looking good. I left a few comments that should be addressed before this PR gets merged. 🤔". Below the text area, there is a dashed line and a placeholder text: "Attach files by dragging & dropping, selecting or pasting them." with a "Media" button. At the bottom, there are three radio button options: "Comment" (unselected), "Approve" (unselected), and "Request changes" (selected). Each option has a description: "Submit general feedback without explicit approval.", "Submit feedback and approve merging these changes.", and "Submit feedback that must be addressed before merging." respectively. At the bottom left, there is a green button labeled "Submit review".

Review changes

Finish your review

Write Preview

H B I @

This is looking good. I left a few comments that should be addressed before this PR gets merged. 🤔

Attach files by dragging & dropping, selecting or pasting them. Media

☐ **Comment**  
Submit general feedback without explicit approval.

☐ **Approve**  
Submit feedback and approve merging these changes.

☒ **Request changes**  
Submit feedback that must be addressed before merging.

Submit review

# Cloud Computing

What?

Data Centers operated by Other companies accessed over the internet or private-network and Providing Infrastructure, Platforms, and Software as a Service

Business Value?

We Trust Microsoft Datacenters more than ours,

We want to eliminate Datacenters management overhead

We don't want to spend over Data centers, Cooling, etc

We Want to Manage our cost in a monthly basis

We Want a Guaranteed Availability and a Scalability for our Systems

# Cloud Computing

Famous Cloud Providers?

Azure, AWS, GCP, Oracle, and Digital Ocean

Weight - Important





# Containerization

What is a container

Packaging applications and their dependencies into isolated units (containers) that run consistently across environments.

Why to use?

- It works on my machine
- All In One Object. Ready to run and it **will run for sure**
- portability and consistency
- Microservice Architecture

# Containerization

Basic Concepts:

Image -> Container,  
Registry, Build, Push

Famous Tools:

Docker, Podman

Weight: **Critical**



# Containerization

TAG

[stable-perl](#)

Last pushed 9 days by [dolan](#)

```
docker pull nginix/stable-perl
```



Digest	OS/ARCH	 Vulnerabilities	Compressed size 
<a href="#">9c6dace91229</a>	linux/386	<div><div>C</div><div>3</div><div>1</div><div>58</div><div>2</div></div>	78.57 MB
<a href="#">acd52f1ebc12</a>	linux/amd64	<div><div>C</div><div>3</div><div>1</div><div>58</div><div>2</div></div>	80.09 MB
<a href="#">93b9f35c0c0e</a>	linux/armv5	<div><div>C</div><div>3</div><div>1</div><div>58</div><div>2</div></div>	70.00 MB
<a href="#">+5 more...</a>			

TAG

[stable-otel](#)

Last pushed 9 days by [dolan](#)

```
docker pull nginix/stable-otel
```

Digest	OS/ARCH	 Vulnerabilities	Compressed size 
<a href="#">3059d2819492</a>	linux/amd64	<div><div>C</div><div>3</div><div>1</div><div>60</div><div>2</div></div>	72.38 MB
<a href="#">c00e11b582d0</a>	linux/arm64/v8	<div><div>C</div><div>3</div><div>1</div><div>60</div><div>2</div></div>	68.87 MB

# Containerization

```
ubuntu@ip-172-31-2-195: ~  
FROM openjdk:11-jdk  
MAINTAINER gfg author  
LABEL env=production  
ENV apparea /data/app  
RUN mkdir -p $apparea  
ADD https://get.jenkins.io/war/2.397/jenkins.war $apparea  
WORKDIR $apparea  
EXPOSE 8080  
CMD ["java", "-jar", "jenkins.war"]  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

# Containerization

```
hemantasundaray@: ~/Desktop/docker-demo $ docker build . -t sample-flask-app:v1
[+] Building 7.2s (9/9) FINISHED
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 187B 0.0s
=> [internal] load .dockerignore 0.1s
=> => transferring context: 51B 0.0s
=> [internal] load metadata for docker.io/library/python:3.11-slim 6.8s
=> [1/4] FROM docker.io/library/python:3.11-slim@sha256:53a67c012da3b807905559fa59f 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 190B 0.0s
=> CACHED [2/4] WORKDIR /app 0.0s
=> CACHED [3/4] RUN pip install flask==2.3 0.0s
=> CACHED [4/4] COPY . /app 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:b409d5824cd3f9f8663b473dae9f9d570a99fb6d0b4632fa3096aabf 0.0s
=> => naming to docker.io/library/sample-flask-app:v1 0.0s
```

# Containers orchestration

What?

Automating deployment, scaling, and management of containerized applications across clusters.

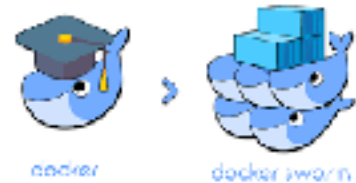
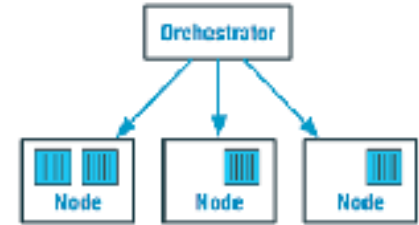
Why:

- Enables high availability and fault tolerance.
- Simplifies complex microservices management.
- Supports dynamic scaling and self-healing.

# Containers orchestration

Famous Tools:

- Kubernetes (k8s)
- Docker Swarm
- Red Hat Openshift



Weight: **Critical** (If being used)

# Kubernetes

De facto in Containers orchestration

Many flavours

Managed, Self-hosted

Examples



GKE



EKS



AKS



# Helm

What is?

Is a packaging tool for Kubernetes-based applications

It packs both App+Infrastructure resource

Why we use it?

Declarative syntax

Desired State Management

Gitops, Code Delivery approach with reviews

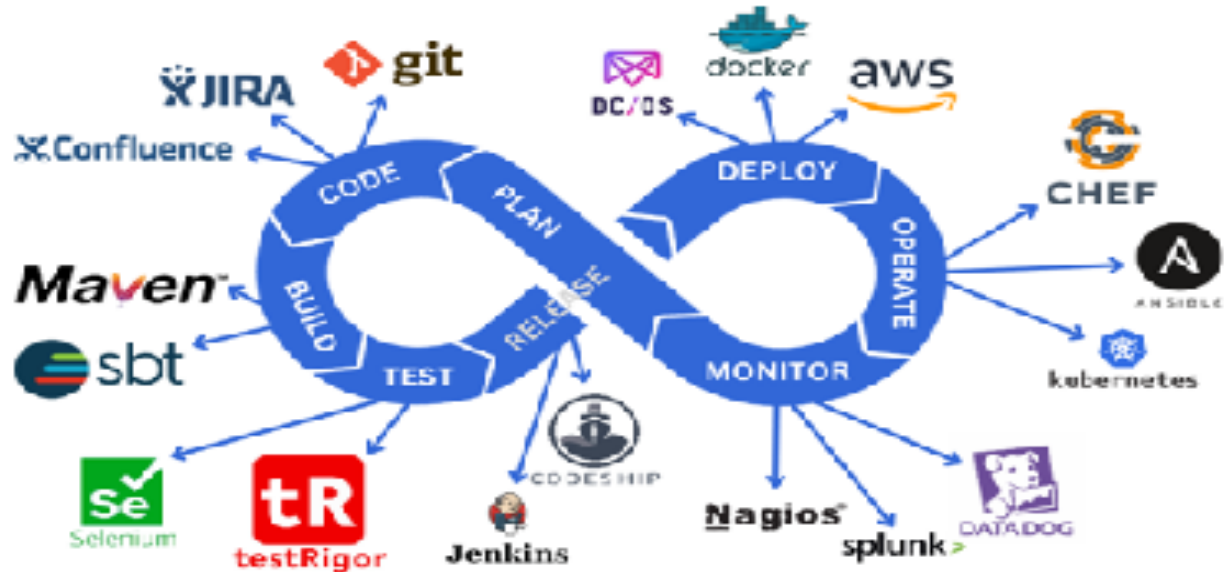
Weight - **Important** (if Kubernetes is used)



# Continuous Integration Continuous Deployment/Delivery CI/CD

What?

Building an Automation Lifecycle for Applications Delivery from Static Code to Production



# Continuous Integration Continuous Deployment/Delivery CI/CD

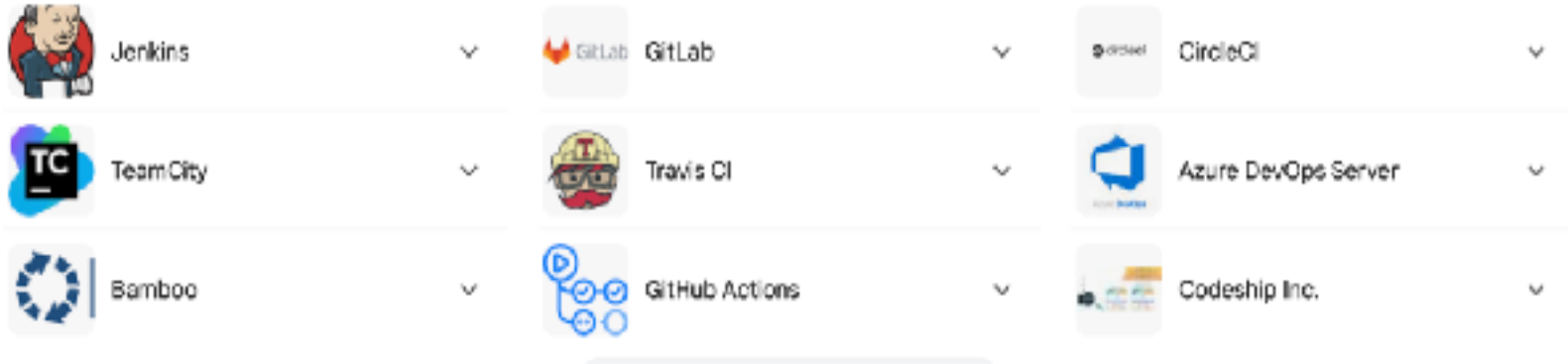
## Business Value?

- Speed up release cycles.
- Improve software quality and reliability.
- Reduce manual intervention and deployment risks.
- Enforce Application Security

# Continuous Integration Continuous Deployment/Delivery CI/CD

Famous Technologies?

Jenkins, Azure Devops, Github Actions, Gitlab CI, TravisCI, and CircleCI



Weight - **Most Critical**

# Infrastructure as Code (IaC)

What?

Building an Automation Lifecycle for Infrastructure Changes from Static Code to Production

Streamlining Infrastructure Changes through CI/CD pipelines, using declarative syntaxes for Desired State Management

Business Value?

Famous Technologies?

Weight

# Infrastructure as Code (IaC)

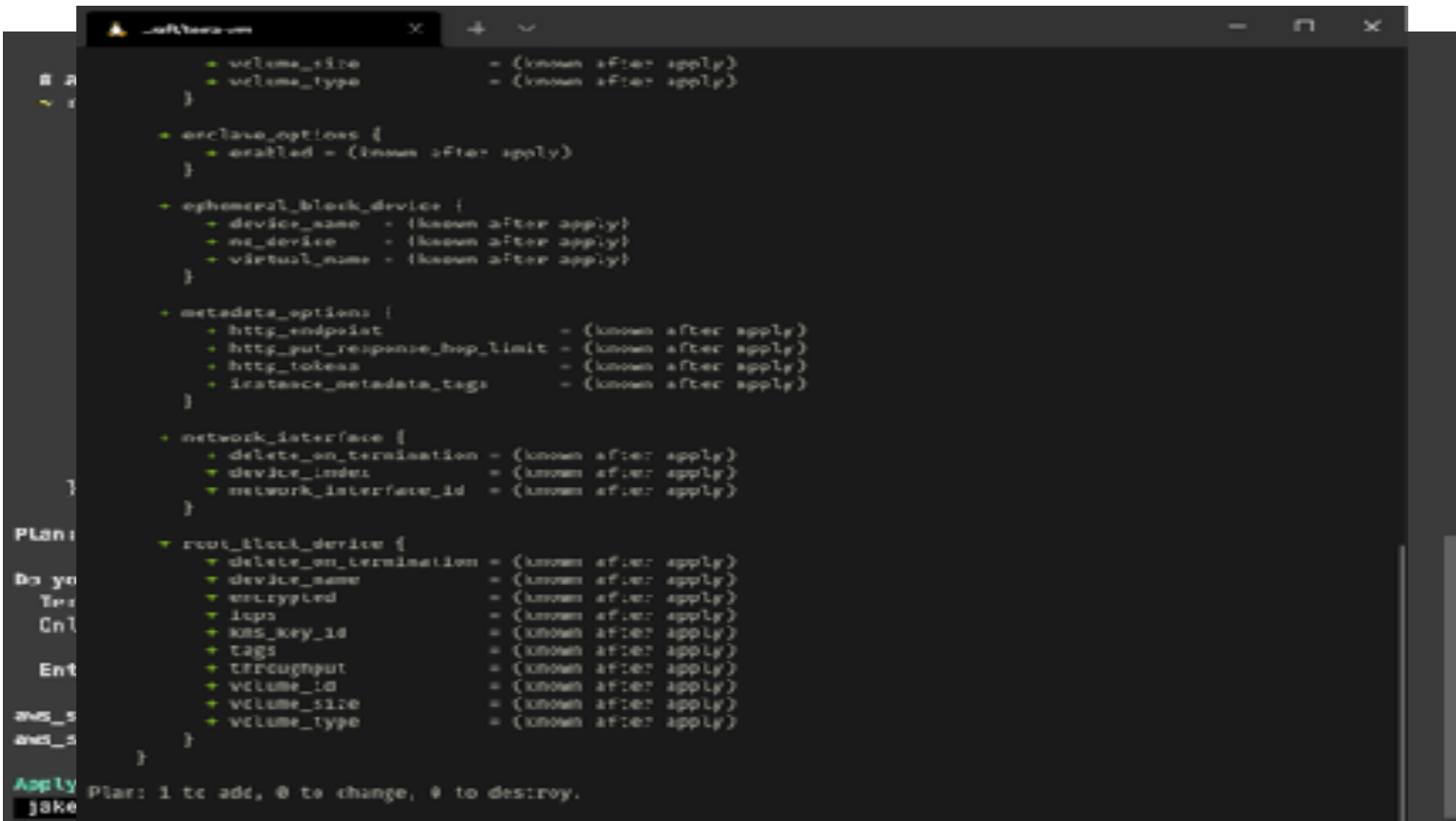
What?

Building an Automation Lifecycle for Infrastructure Changes from Static Code to Production

Business Value?

- Enables repeatable and consistent environments.
- Supports version control and automation.
- Reduces configuration drift and manual errors.

# Infrastructure as Code (IaC)



The screenshot shows a terminal window with a dark background. The title bar of the window reads "softlayer.com". The terminal displays a Terraform configuration snippet for an AWS instance. The configuration includes settings for volume size and type, enclave options, ephemeral block device, metadata options, network interface, and root block device. Each property is followed by a comment indicating its value is "known after apply".

```
    + volume_size          = (known after apply)
    + volume_type          = (known after apply)
  }

  + enclave_options {
    + enabled = (known after apply)
  }

  + ephemeral_block_device {
    + device_name = (known after apply)
    + no_device   = (known after apply)
    + virtual_name = (known after apply)
  }

  + metadata_options {
    + http_endpoint          = (known after apply)
    + http_put_response_hop_limit = (known after apply)
    + http_tokens            = (known after apply)
    + instance_metadata_tags = (known after apply)
  }

  + network_interface {
    + delete_on_termination = (known after apply)
    + device_index          = (known after apply)
    + network_interface_id = (known after apply)
  }

  + root_block_device {
    + delete_on_termination = (known after apply)
    + device_name           = (known after apply)
    + encrypted             = (known after apply)
    + iops                   = (known after apply)
    + kms_key_id            = (known after apply)
    + tags                  = (known after apply)
    + throughput            = (known after apply)
    + volume_id             = (known after apply)
    + volume_size           = (known after apply)
    + volume_type           = (known after apply)
  }
}
```

At the bottom of the terminal, there is a prompt "Apply" followed by a message: "Plan: 1 to add, 0 to change, 0 to destroy." and a cursor pointing to the word "joke".









# Infrastructure as Code (IaC)

Famous Technologies?

Terraform, Azure RM, AWS Cloudformation

## IaC tools

From sources across the web

 Terraform	▼	 Ansible	▼	 logres Chef	▼
 Pulumi	▼	 Puppet	▼	 AWS CloudFormation	▼
 Google Cloud Deployment M...	▼	 Salt	▼	 Resource management	▼

Weight - Normal



# Configuration Management

What is it

Manage your infra from central point

Why we need it

Govern your infrastructure management

Examples

Ansible

Weight - **Critical if Used**



# Secret Management

What is Secret Management?

Storing, distributing, rotating, and revoking secrets securely

Why we need it?

- Prevent credential leaks
- Preventing unauthorized access to secrets
- Removing Hardcoded passwords in Source Code

Code



## Secret Management

### Features:

Centralized Secret Store, Automation, Compliance

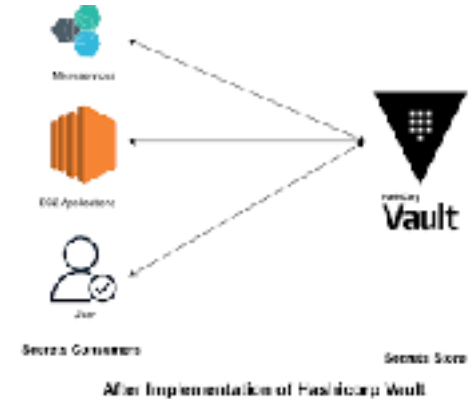
### Examples:

CI/CD Secretes, Azure Key Vault, Hashicorp Vault

What is expected from Devops Engineers:

Provisioning, Managing AuthN/AuthZ, Configuration

Weight - **Critical**



## Secret Management

Before: ❌

After: ✅

```
1 from azure.keyvault.secrets import SecretClient
2 from azure.identity import DefaultAzureCredential
3 import psycopg2
4
5 key_vault_name = "akvsalesaps"
6 key_vault_uri = f"https://{key_vault_name}.vault.azure.cn"
7 secret_db_user = "db_user_sales"
8 secret_db_password = "db_user_password_sales"
9
10 db_host = "db.salesaps.example.com"
11 db_port = "5432"
12
13 try:
14     credential = DefaultAzureCredential()
15     client = SecretClient(vault_url=key_vault_uri, credential=credential)
16
17     db_user = client.get_secret(secret_db_user).value
18     db_password = client.get_secret(secret_db_password).value
19
20     conn = psycopg2.connect(
21         database='sales_db',
22         user=db_user,
23         password=db_password,
24         host=db_host,
25         port=db_port
26     )
27
28     print("PostgreSQL connection established successfully!")
29
30 except Exception as ex:
31     print(f"Error: {ex}")
32
33 conn.close()
```

# Observability

What is it?

- The ability to understand system behavior and performance by analyzing telemetry data (logs, metrics, traces)

Why we need it?

- Accelerates root cause analysis and incident response.
- Improves reliability and user experience.
- Enables proactive performance tuning and capacity planning.

# Observability

## Famous Technologies

- Prometheus, Grafana, Loki
- Datadog
- Open Telemetry



# Terms

Query Language

Metrics

Values

Logs

Logs

Traces

Track request



# Dashboards

What is it

Prepared visualizations of Operational Logs

Aggregations

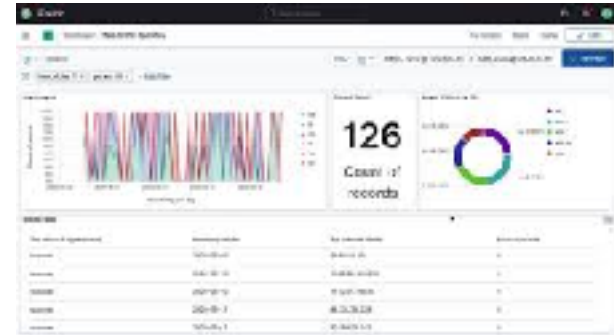
Ex: Failure Rate, Response Time

Why we need it

Golden Signals, Anomalies Detection, Alerting

Weight - **Normal** for Devops Engineers

Weight - **Critical** for SREs





# Extras - Gitops

What?

Manage both App and Infrastructure changes through Code Review Process

Business Value?

Reduce Manual Intervention Errors

Improve Developers Productivity

Adopt Code Review into Cloud Native applications  
changes

“I could make an Error, but **We** are less likely to do so”

# Extras - Gitops

Famous Technologies?

ArgoCD, FluxCD, Helm

Weight - Normal



# Extras - DevSecOps

What?

Enforcing Application Security Scans into CI/CD pipelines

Business Value?

Secure Applications from early development stages

Utilize Quality Gates for Secure Releases

Famous Technologies?

SonarQube, OWASP ZAP, Synk

Weight - Normal

# How To start and get into a job

- There are no Jr. Devops Roles
- Make yourself an Experience from hands-on projects
- Work with your hands-on
- Oracle Virtual Box
- Github, and Gitlab
- Kodekloud for Skills
- PluralSight for Certificates
- AWS/ Azure Free Plans

# Recommendations

- Change doesn't happen between a day and night?
- Always, think, speak, and be driven by business value/terms
- Focus on Concepts Only. Commands or Configurations are available everywhere
- Once you land a job:
  - Make your Standard and Enforce it "Work Smart not Hard"
  - Don't embrace workarounds, they will make you "work hard"
  - Follow Devops Institute, subscribe to their mailing list
  - Read about Site Reliability Engineering

## Sponsors

- <https://nogalliance.org/>

