DevOps Interview Questions + Answers!!

https://youtu.be/GX6fOvaS0Xs?si=9TkIptV-PLWNYFUB

Section 1 - `Self related Questions` 👨‍💻

Tell me about Yourself and The Projects you have worked on:

💡 - Mention your role as a DevOps engineer.
- Highlight specific projects, like CI/CD setup, Terraform automation, and Docker/Kubernetes optimization.

Do you know any programming or scripting language:

```
State your proficiency in Python and Bash scripting.
```

How I use Python as DevOps Engineer | Python for Devops

29K views • 5 months ago

Cloud Champ

Why you should learn **Python** as a **DevOps** engineer | **Python** for **DevOps** | **Python** for **DevOps** Automation This CloudChamp ...

Intro | Why you should Learn Python to become DevOps Engineer? | Python Use Case - Scripting an... 12 chapters

What are your favorite DevOps tools and why:

> - Share your preference for tools like Terraform, Jenkins, Ansible, Docker, Kubernetes and Prometheus. (Popular in Demand tools)
> - Briefly explain why you like them.

How do you stay up-to-date with the latest DevOps trends and technologies:

> Describe your approach to staying current, including reading **DevOps blogs, attending conferences, and participating in online communities.**

How Quickly can you learn:

> Mention your ability to quickly learn new technologies, typically within a few weeks.

If asked, can you architect an application and How quickly can you do it:

💡 Confirm your capability to architect applications and state that the timeline depends on the project's complexity.

What are some problems that you have faced while working on a project:

```
- Share a specific challenge you've encountered, such as underestimating infrastructure capacity, managing secrets, cost optimization etc.
- Emphasize the lesson learned from the experience.
```

Section 2 - `Linux` 🐧

Have you used any Linux Flavors if yes which one:

> List the Linux flavors you have experience with, such as Ubuntu, CentOS, or Amazon Linux.

What is the Command to change the ownership and Permission of a file or directory in Linux:

```
To change ownership: chown [new_owner] [file/directory]
To change permissions: chmod [permissions] [file/directory]
```

Example:

- Change ownership to "cloudchamp": `chown cloudchamp /var/www/myfile.txt`
- Change permissions to read and write for the owner: `chmod u+rw /var/www/myfile.txt`

How do you manage and view running processes in Linux:

💡 I use the ps command to list processes and top to view real-time system stats.

What is SSH:

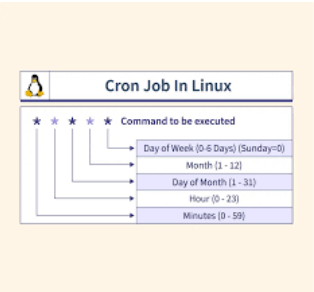
> SSH, or Secure Shell, is a secure way to remotely connect to and control Linux computers over the internet. It keeps your data safe through encryption and authentication.

How to check memory stats and CPU stats as a Linux Admin:

Use `free` for memory stats and `top` or `htop` for CPU stats.

What is a cronjob:

> A **cronjob** is a scheduled task in Unix-like operating systems. It allows you to automate repetitive tasks by specifying when and how often they should run.



What is alias in Linux:

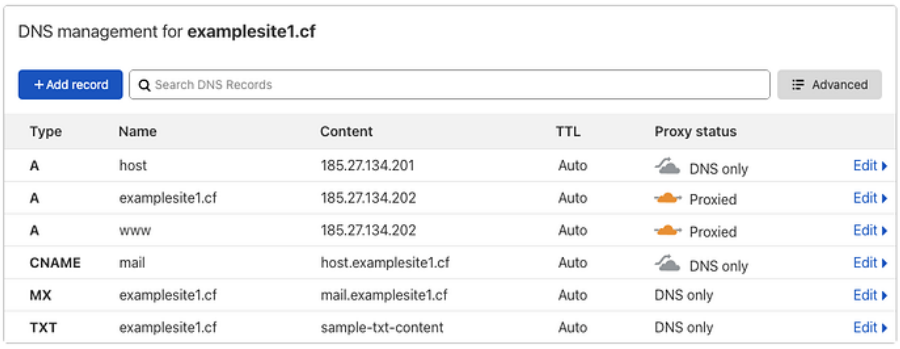
> An alias in Linux is a custom shorthand for longer commands. It helps save time and reduce typing errors.

```
alias alias_name='command or command sequence'
```

Section 3 - [redacted] 📶

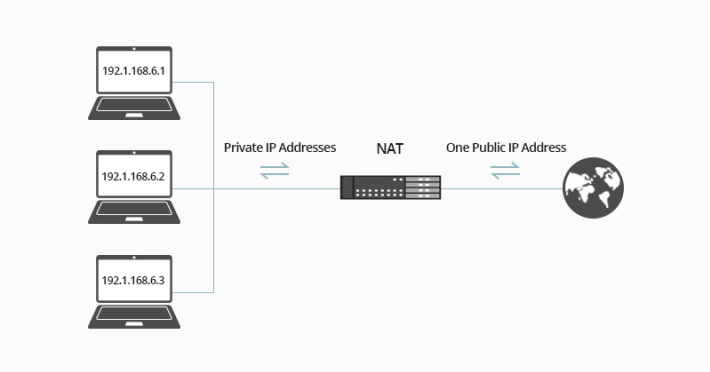
What is DNS (Domain Name System), and how does it work?

> DNS is a system that translates human-readable domain names (like example.com) into IP addresses (like 192.168.1.1) that computers use to identify each other on the network.



What is NAT (Network Address Translation), and why is it used?

> A Network Address Translation (NAT) is the process of mapping an internet protocol (IP) address to another by changing the header of IP packets while in transit via a router. This helps to improve security and decrease the number of IP addresses an organization needs.

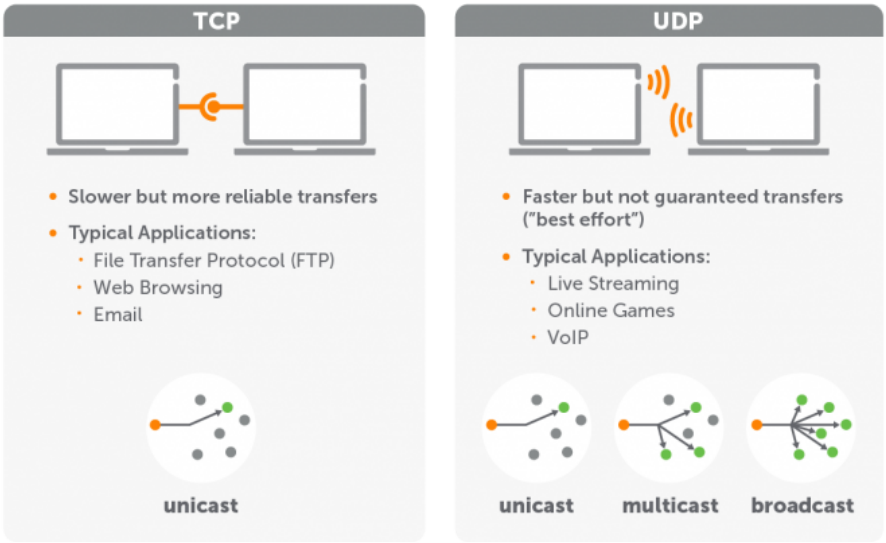


Explain the differences between TCP (Transmission Control Protocol) and UDP (User Datagram Protocol):

> 💡 TCP provides a reliable, connection-oriented communication with error checking and retransmission of lost data. UDP is connectionless, faster, and used when some packet loss is acceptable, such as in real-time video streaming.

TCP **Use case:** File transfer, web browsing, emails.

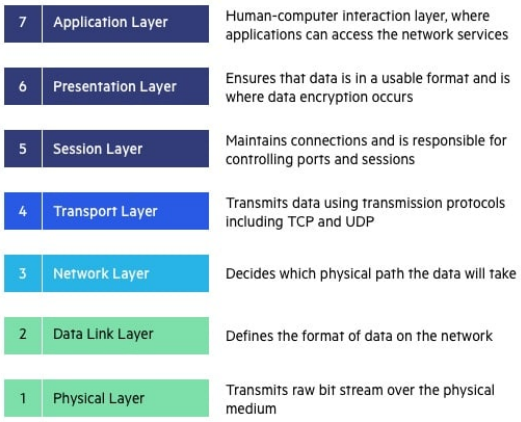
UDP **Use case:** Real-time video streaming, online gaming, VoIP



What are the seven layers in OSI model:

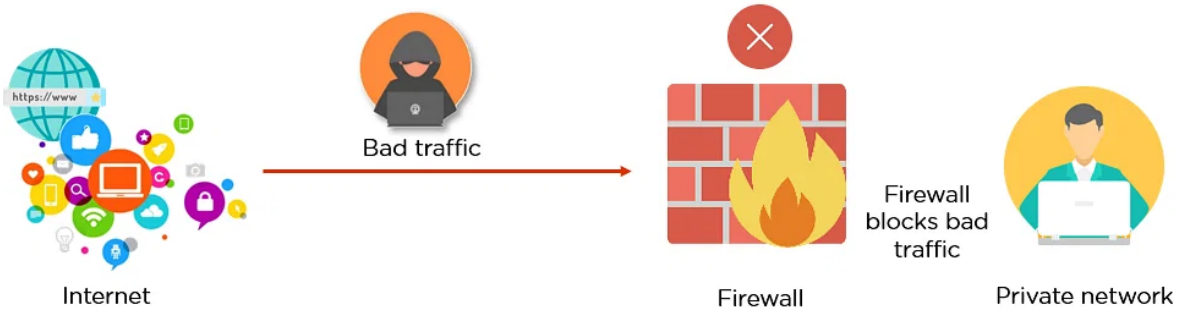
The OSI (Open Systems Interconnection) model has seven layers:

1. Physical
2. Data Link
3. Network
4. Transport
5. Session
6. Presentation
7. Application



What is a firewall:

A firewall is a network security device or software that monitors and controls incoming and outgoing network traffic based on predefined security rules. It acts as a barrier between a trusted internal network and untrusted external networks.



What is Ping command:

> The `ping` command is used to test network connectivity by sending ICMP echo requests to a target host and measuring the response time.

What is the `ifconfig` (or `ipconfig` on Windows) command used for, and how do you use it to display network interface information:

> 💡 `ifconfig` (or `ipconfig` on Windows) displays information about network interfaces on a system, including IP addresses, MAC addresses, and network configuration details.

Section 4 - [redacted] 🗃️

What is Git and how do we use it in DevOps:

> Git is a distributed version control system used to track changes in source code during software development. In DevOps, Git is used for version control, collaboration, and to automate code deployments through CI/CD pipelines.

Explain me the workflow of how you push your code from a local machine:

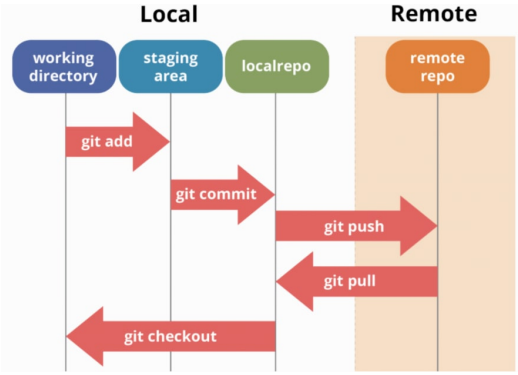
The typical workflow involves the following steps:

> `git add [files]`: Stage changes for commit.

> `git commit -m "Message"`: Create a commit with a descriptive message.

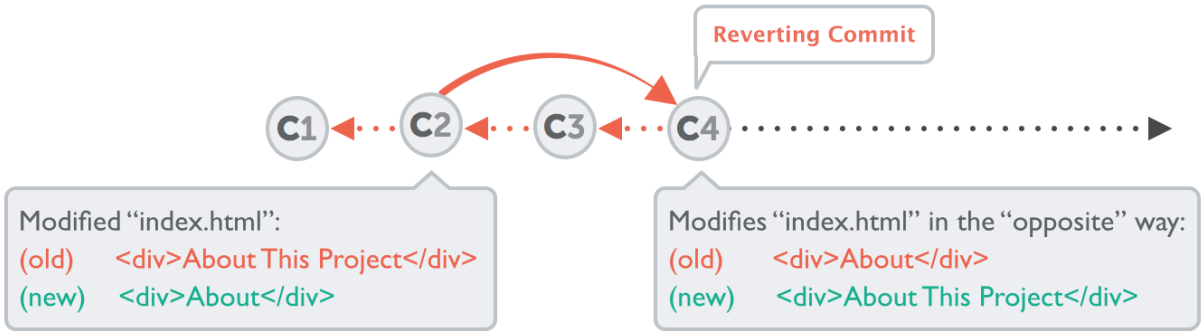
> `git pull origin [branch]`: Fetch changes from the remote repository.

> `git push origin [branch]`: Push changes to the remote repository.



How do you revert a commit that you made in your repository:

> Use `git revert [commit_hash]` to create a new commit that undoes the changes made by a previous commit.



What is a branch in Repository:

> A branch in a Git repository is a separate line of development that allows multiple developers to work on different features or bug fixes simultaneously without affecting the main codebase.

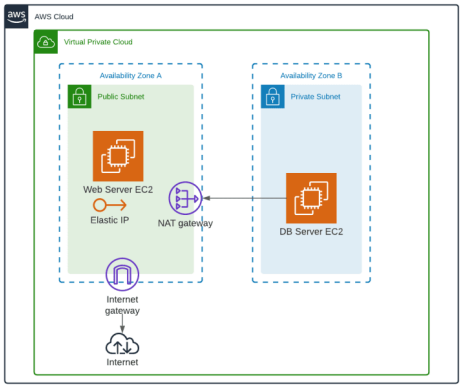
Section 5 -

What cloud are you familiar with:

> I'm familiar with AWS (Amazon Web Services), Azure, and GCP (Google Cloud Platform).

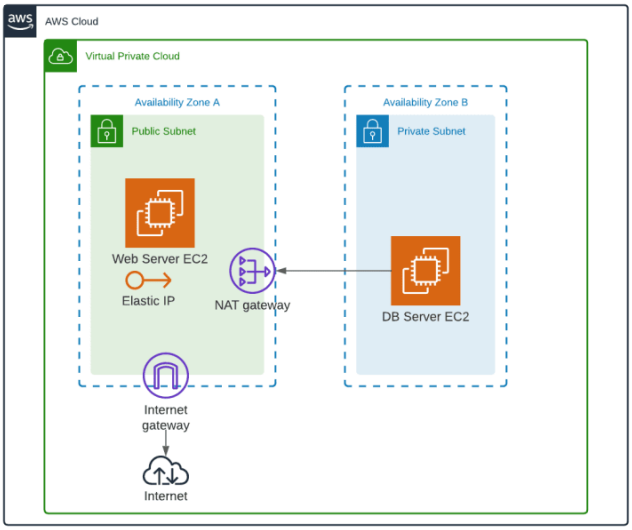
What is a VPC in cloud:

> A Virtual Private Cloud (VPC) is a virtual network within a cloud provider's infrastructure that you define and it allows you to isolate, control and deploy your resources, such as virtual servers and databases.



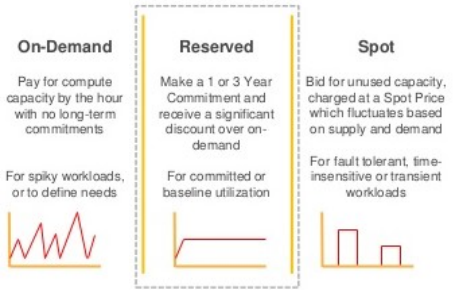
What is the difference between a Public and Private Subnet and what differentiates it:

> A public subnet is accessible directly from the internet, while a private subnet is not. Public subnets are typically used for resources that need public access, like web servers, while private subnets are used for resources that should not be directly exposed, like databases.



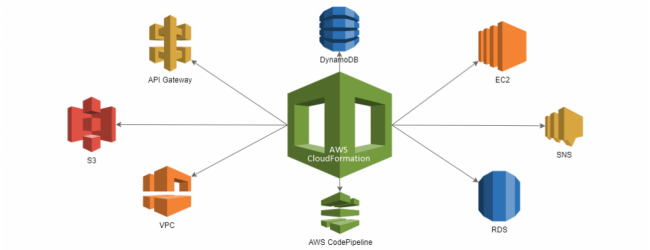
What is the difference between reserved instance and spot instances:

Reserved instances are long-term reservations of compute capacity, offering a significant discount in exchange for a commitment. Spot instances are spare capacity instances available at a lower price but can be terminated with short notice.



What is CloudFormation:

AWS CloudFormation is a service that allows you to define and provision AWS infrastructure as code (IaC). You can create, update, and delete AWS resources using templates written in JSON or YAML.



Section 6 - [redacted]

What are the popular IaC tools have you used:

I've used Terraform and AWS CloudFormation.

What is the difference between Terraform and Ansible:

Terraform is primarily used for infrastructure provisioning and management, while Ansible is a configuration management and automation tool.

Terraform focuses on describing the desired state of infrastructure, while Ansible defines how to configure servers.

Infrastructure as Code vs Configuration as Code: Difference between Ansible and Terraform

4.1K views • 8 months ago

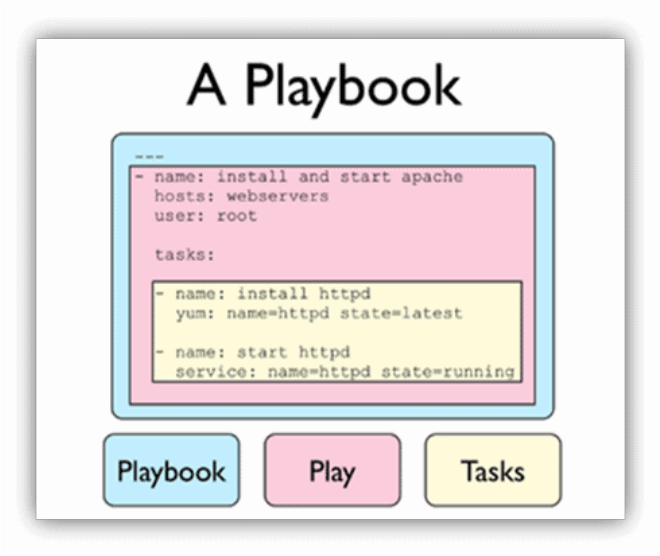
Cloud Champ

This **Cloudchamp** video tutorial on '**Terraform Vs Ansible**' will give you an overview of similarities & differences between **Terraform** ...

What is Infrastructure as code. | What is configuration as code. | Difference between IAC and... 3 chapters

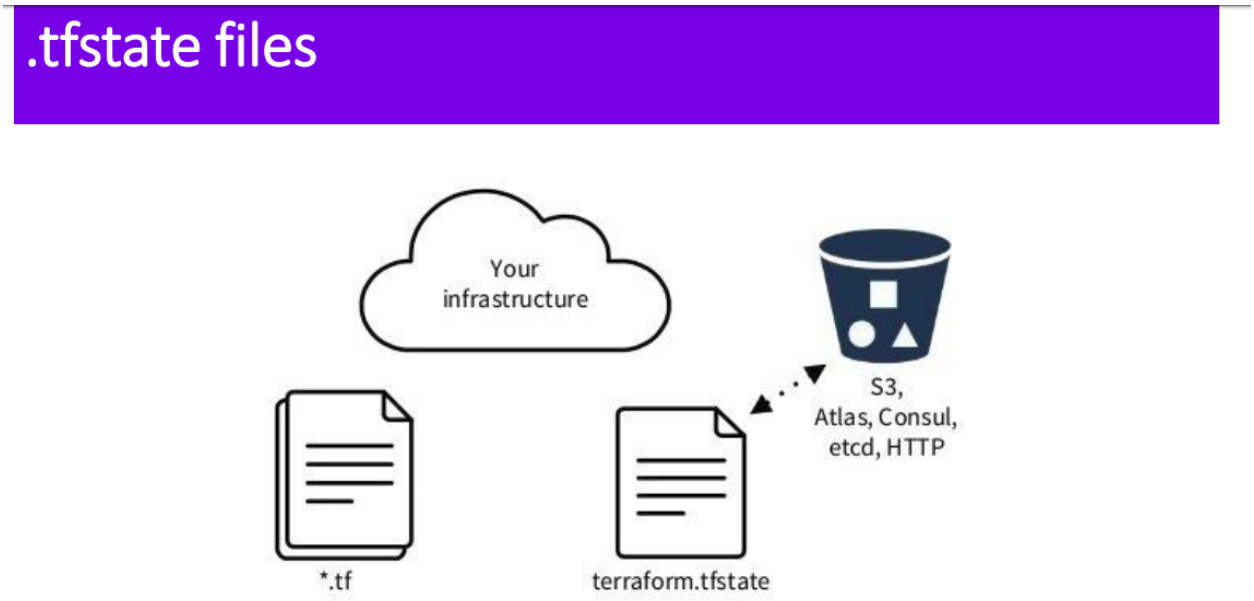
What is a playbook in Ansible:

A playbook in Ansible is a YAML file that defines a set of tasks and configurations to be executed on remote servers. It is used for automation and configuration management.



What is a state file in Terraform:

💡 The Terraform state file is used to keep track of the resources that Terraform manages. It stores information about the current state of the infrastructure and helps Terraform plan and apply changes accurately.



What is Terraform Remote state backend?

Terraform remote state is a mechanism that allows Terraform to store its state information in a centralized location, such as an object storage bucket or a remote key-value store.

```
terraform {
  backend "s3" {
    bucket         = "my-terraform-state-bucket"
    key            = "terraform.tfstate"
    region         = "us-east-1"
    encrypt        = true
    dynamodb_table = "my-lock-table"
  }
}
```

Section 7 - [redacted] ⚓

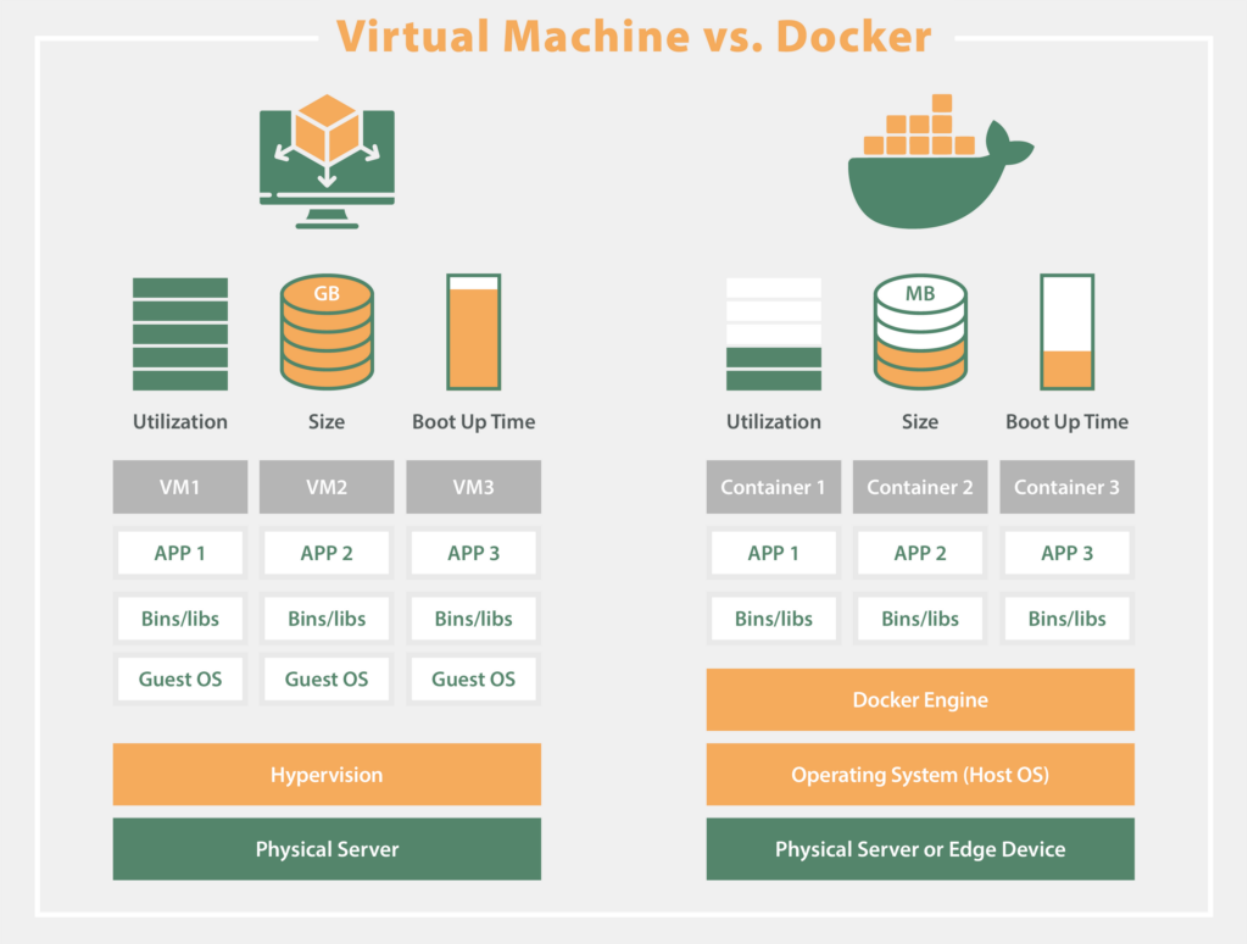
What is the difference between Virtualization and containerization:

Virtualization:

Running multiple virtual machines (VMs) on a single physical server, each with its own operating system. It's like having separate houses with different families in a single building.

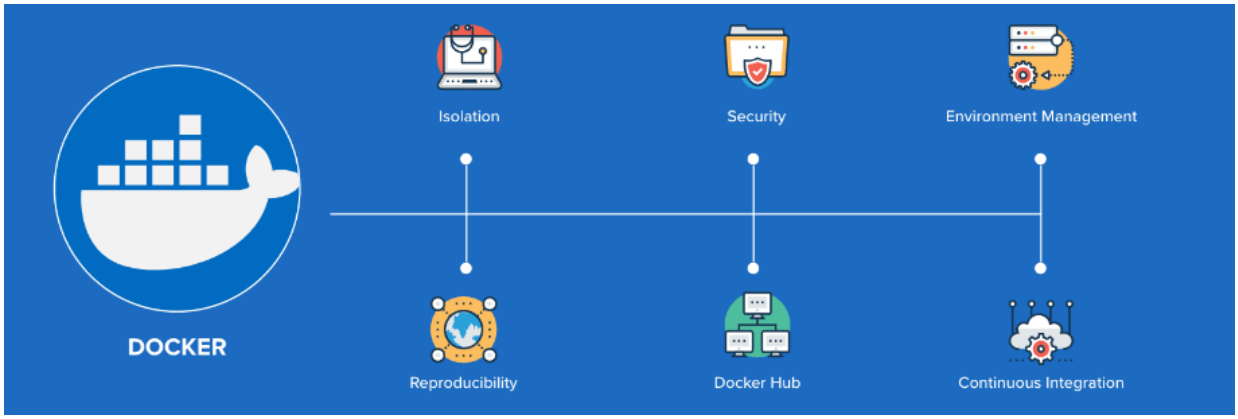
Containerization:

Running isolated applications (containers) on a shared operating system. It's like having apartments in a single building, where each apartment is self-contained but shares common infrastructure.



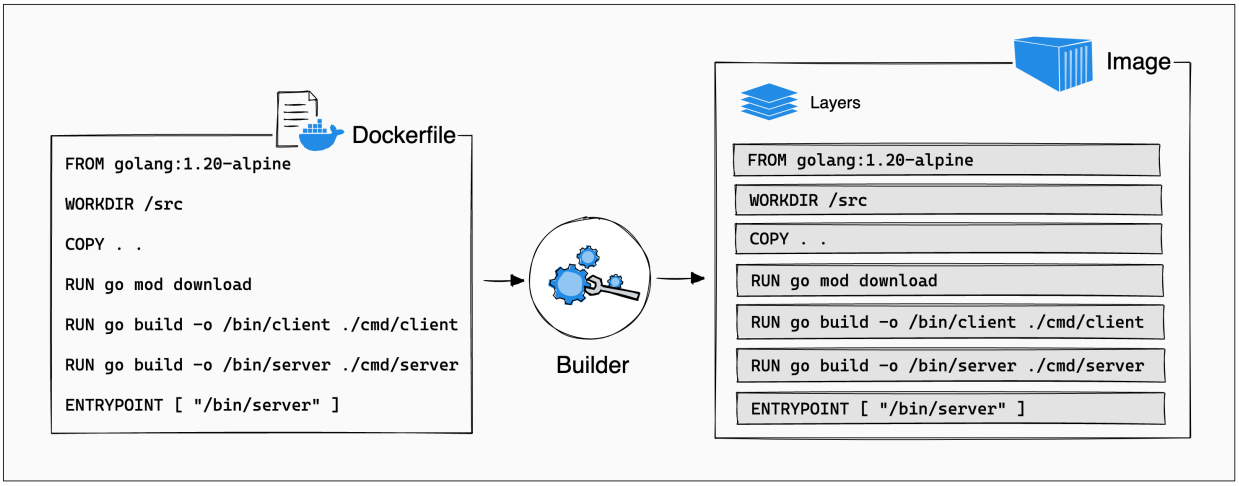
What problem does Docker solve:

> Docker solves the problem of consistent, portable, and efficient application deployment by packaging applications and their dependencies into containers, ensuring they run reliably across **different environments and systems while maintaining security and scalability.**



What is Dockerfile and why do you use it:

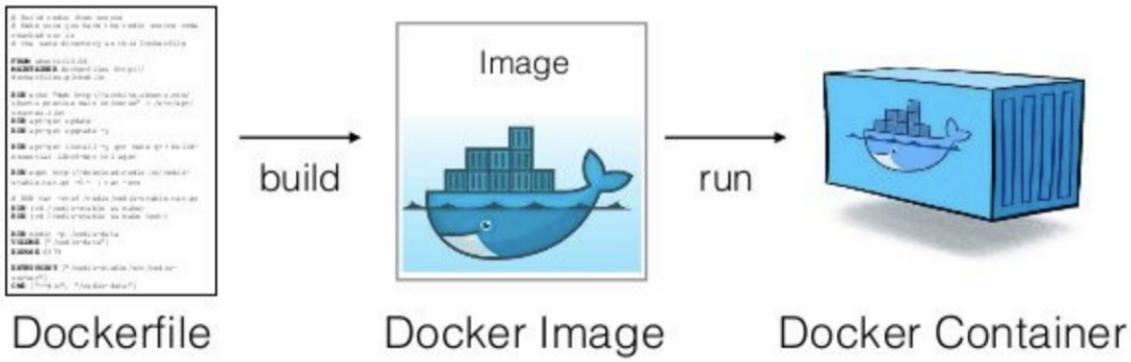
> A Dockerfile is a script that defines the steps to create a Docker container image. It's used to automate the containerization of an application, making it reproducible and shareable.



Explain the workflow of how a Docker Container is created?

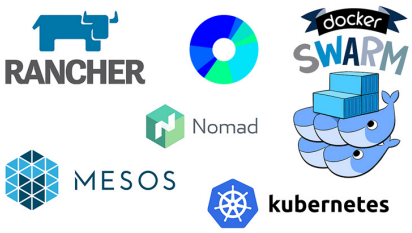
Docker Container Creation Workflow:

1. **Dockerfile**: Create a Dockerfile with application instructions.
2. **Build Image**: Use `docker build` to build an image from the Dockerfile.
3. **Run Container**: Run a container from the image using `docker run`.
4. **Container**: The container runs your application, isolated and self-contained.



How do you manage multiple containers:

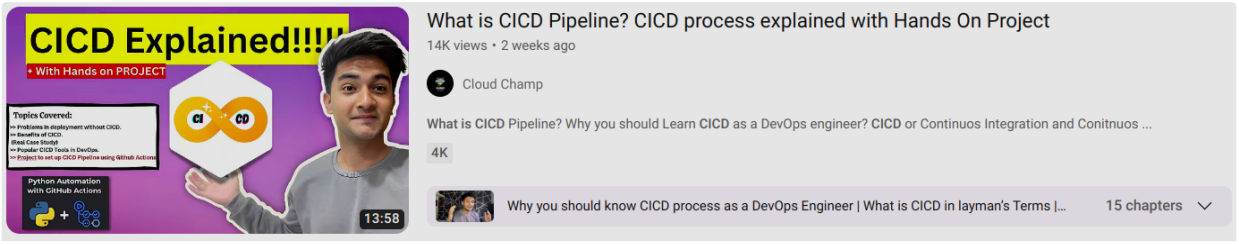
> I use container orchestration tools like Kubernetes to manage multiple containers, ensuring high availability, scalability, and load balancing.



Section 8 - `CICD`

What is CI and CD in CICD:

CI (Continuous Integration) is the practice of automatically building and testing code changes frequently. CD (Continuous Deployment/Delivery) is the automated process of deploying code to production after successful CI.



What CICD tools have you used in the past:

I've used Jenkins, GitHub actions, CircleCI, and GitLab CI/CD in various projects.

How will you create a CICD pipeline to update the website or app on every commit to a particular branch:

To create a CICD pipeline for this scenario, I would configure the following steps:

1. Set up a version control system (e.g., Git).

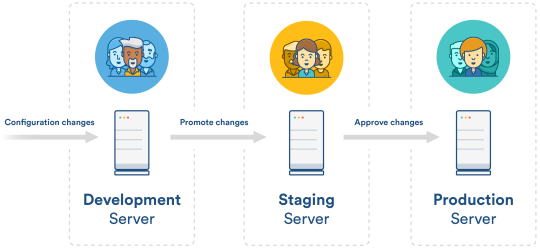
2. Use a CICD tool (e.g., Jenkins) to monitor the repository for commits to a specific branch.
3. Configure automated tests to ensure code quality.
4. Build and package the application.
5. Deploy the application to a staging environment for further testing (if required).
6. If tests pass, automatically deploy the application to the production environment.

Additionally, implement rollback mechanisms and monitoring to ensure the production environment remains stable.

Section 9 - [redacted] 🚀

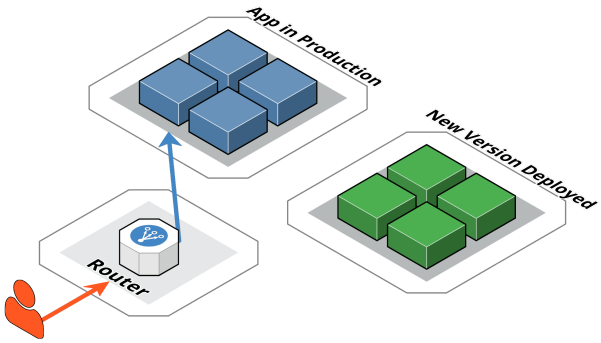
Explain staging, production and testing environment

- **Development/Testing Environment:** Where developers write and test code.
- **Staging Environment:** Pre-production environment for final testing before the live environment.
- **Production Environment:** Live environment for end-users.



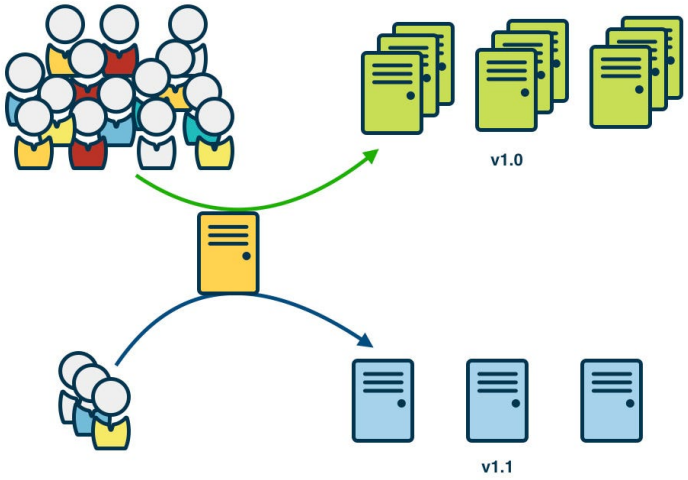
What is a Blue Green Deployment:

A Blue-Green Deployment is a deployment strategy where you have two identical environments (Blue and Green), and you switch traffic between them when deploying new versions. This minimizes downtime and allows quick rollbacks if issues arise.



Explain Canary Deployment:

Canary Deployment is a strategy where a new version of an application is deployed to a small subset of users or servers, allowing for testing and validation. If it performs well, it's gradually rolled out to the entire user base.



What is the biggest issue you faced:

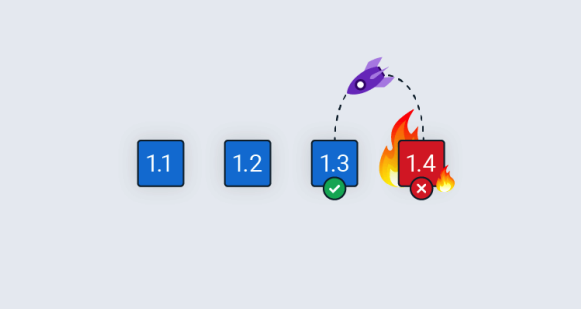
The most significant challenge I faced was managing a sudden influx of traffic during a product launch. We had to quickly scale our application to handle the load, which involved optimizing database queries and adding more servers.

How do you scale your application:

I use auto-scaling groups in AWS or Kubernetes Horizontal Pod Autoscaling to dynamically adjust resources based on traffic patterns and resource utilization.

How do you rollback if something fails:

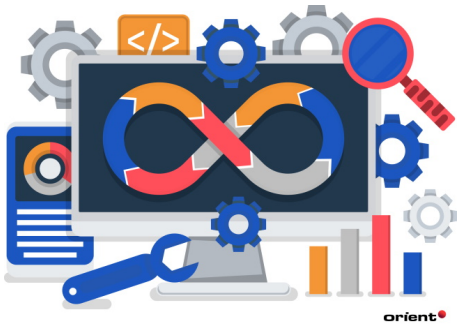
I roll back by deploying the previous version of the application or by using container orchestration tools to revert to the last stable state. I also ensure comprehensive monitoring and alerts to detect failures early.



How do you automate Deployment:

> I automate deployment using scripts and CI/CD pipelines with tools like Jenkins, Travis CI, or GitLab CI/CD. This includes building, testing, and deploying code automatically on every commit. Also

> continuos monitoring to make sure systems are up and properly working



Note: If you found this document helpful and want to get more content like this, please consisder SUBSCRIBING to CloudChamp on Youtube: 👉 SUBSCRIBE 👈