DAY 3: INFRASTRUCTURE MANAGEMENT

Introduction to Infrastructure Management

infrastructure is the set of hardware and software components used to develop, test, and deploy applications.

Some management tasks:

- Hardware needs to be installed and maintained.
- Thought must be given to power and cooling.
- Networks and databases need to be configured.
- Hardware failures and cyber-attacks are serious concerns with any infrastructure.

Dealing with all of these things is a tall task — even for seasoned professionals!

The main topics we will cover are:

- Scaling Infrastructure
- In-house Infrastructure
- Virtualization
- Containerization
- Infrastructure as Code
- Orchestration
- Cloud Infrastructure

———————————————————————

Scaling Infrastructure

Scalability is a system's ability to add resources to keep up with growing demand. An infrastructure with poor scalability will likely cause slowdowns or disruptions.

Vertical & Horizontal Scaling

scalability can be achieved either through vertical scaling or horizontal scaling.

- **Vertical scaling** means adding computing resources.
- **Horizontal scaling** means adding more servers.

A tool called a "load balancer" then distributes the work across the many servers.

Vertical scaling is relatively simple and affordable, as it only involves upgrading a machine. The only drawback is the downtime required to perform the upgrade.

Horizontal scaling has the benefit of not requiring any downtime for existing servers. This is why DevOps teams prefer this scaling method over the last option, however it is more expensive than upgrading existing machines.

The Price of Scaling
companies still need to consider the cost of their infrastructure. Scaling is about finding that sweet spot — enough to perform well but not so much that money is wasted.

Whereas scalability only deals with increases in resources, elasticity is the ability to automatically add *or subtract* resources to accommodate fluctuating demand. Elasticity comes in handy when using pay-per-use infrastructure services

DevOps uses several techniques to achieve both elasticity and scalability. Automation, cloud-based infrastructure, and microservices are some of these practices.

———————————————————————————

Virtualization
You will need to consider scale if you infrastructure is about to see a growth in activity. Scaling horizontally will enable us to handle the increase in traffic without interruptions!

Scaling horizontally will require more servers than we need, and could lead to waste. Another option is virtualization. Virtualization technology allows many virtual machines (VMs) to run on one physical computer. Virtualization would reduce the number of servers needed to run many instances of our blog application. This allows us to horizontally scale even more efficiently based on demand.

Virtualization management tools simplify the task of creating virtual machines. Using these tools is more efficient than installing and managing pieces of hardware by hand.

With virtualization, a single server can run multiple processes and achieve a higher usage percentage.

———————————————————————————

Containerization
Containerization is a form of virtualization in which users create virtual environments called **containers**. containers include instances of applications as well as their dependencies. It’s convenient solution to help applications behave consistently when moving through the deployment pipeline.

Containers do not include their own operating system. The lack of their own operating system makes containers smaller and faster to spin up than VMs, its in terms of seconds!

Since containers do not need their own operating system, they use less physical resources than virtual machines.

Containerization did not become widespread until 2013 with the release of Docker. Today, there are a handful of containerization tools that are used in addition to Docker.

Orchestration

Orchestration is the automated configuration, management, and coordination of infrastructure. Orchestration tools direct many individual components on how to play their part in order to achieve consistency across the entire infrastructure system.

Orchestration would automatically performs tasks such as:

- Deploying containers across many servers
- Restarting failed containers
- Rolling out updates without any downtime
- Horizontal scaling of containerized applications

Infrastructure as Code

Orchestration tools rely on infrastructure as Code. This is the act of defining infrastructure properties in configuration files. Without this change in tools the systems deployment would be prone to "Configuration Drift". This is happens over time, configurations across servers would become inconsistent due to human error.

Infrastructure as code solves this problem by relying on configuration files as the source of truth for infrastructure state. It is also simple to roll back changes if needed since these files are stored in version control systems.

IaC/orchestration tools also rely on the idea of *immutable infrastructure*. This means that servers are not changed once they are created. This guarantees that all servers are created in the same manner and eliminates the risk of configuration drift.

In the past this wouldn't have been an option due to the cost of destroying old servers. Now with virtual machines, however, can be destroyed and created in minutes with little cost.

IaC and immutable infrastructure ultimately lead to lower business costs and a better user experience.

———————————————————————————————

In-House Infrastructure
Historically, businesses owned and managed infrastructure on company premises with their staff. Traditional or in-house infrastructure refers to a system where a company obtains, sets up, and manages physical components such as servers, power supplies, and cooling on their own. This is a very complicated Solution, and these are the problems associated with that Solution:

- Hardware components such as power supplies, hard drives, and RAM fail over time.
- Malicious users attempt to disrupt web services and steal sensitive data.
- Software becomes outdated, requiring consistent patches and upgrades.
- Scaling up infrastructure as demands grow.

Even though this has some risks, it offers unparalleled customization over its resources. While more modern solutions exist, many companies still use traditional infrastructure.

———————————————————————————————

Cloud Infrastructure
Cloud-based infrastructure refers to infrastructure and computing resources that are available to users over the internet. a third-party company owns, houses, and manages the physical infrastructure, allowing application developers to focus on their configs.

The shift towards cloud computing was made possible by virtualization technology, which enables multiple instances of an application to run on physical pools of resources created by cloud providers.

Cloud-based infrastructure has several benefits:

- It allows specific companies to specialize in physical infrastructure management while others focus on business logic.
- It allows a company to scale quickly since cloud providers have physical resources readily available.
- It allows for efficient scaling by taking full advantage of virtualization.

There are still valid reasons to use on-premises infrastructure. These include:

- It provides the ultimate in flexibility and control.
- It decreases reliance on third-party vendors.
- Sensitive data can be kept inside the company.

DevOps uses cloud infrastructure because of benefits in specialization and scalability. Many companies still maintain on-premises infrastructure, especially for internal applications.

Review

In this lesson, we covered many facets of infrastructure management, including:

- Scalability
- Virtualization and Containerization
- Orchestration
- Infrastructure as Code
- Cloud vs On-Premises infrastructure

DevOps brought about many changes to infrastructure management. DevOps practices include an emphasis on containerization, cloud-based infrastructure, and Infrastructure as Code. These practices have brought about cost-effective, scalable infrastructure and ultimately better user experience.

#DEV-OPS-CERT-CODE-ACADEMY