CoGrammar

Welcome to this session:

Computer Networks

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators
answering questions.

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Cloud Web Development

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Skills Bootcamp Cloud Web Development

- For all **non-academic questions**, please submit a query: ***www.hyperiondev.com/support***

- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***

- We would love your feedback on lectures: *Feedback on Lectures*

- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

When a service is running on localhost, it mean

A. The service can be accessed by anyone over the internet

B. The service can only be accessed within a single system

C. The service can be accessed on the local network

D. None of the above

CoGrammar

Which of the following is the default port for SSH?

A. 5500

B. 22

C. 2222

D. 80

Learning Outcomes

- Identify the different layers that make up a computer server
- Define the role of networking in application deployment
- Explain how different services can communicate with each other
- Identify the link between Linux and computer networking
- Outline the process involved in deploying an application to a server.

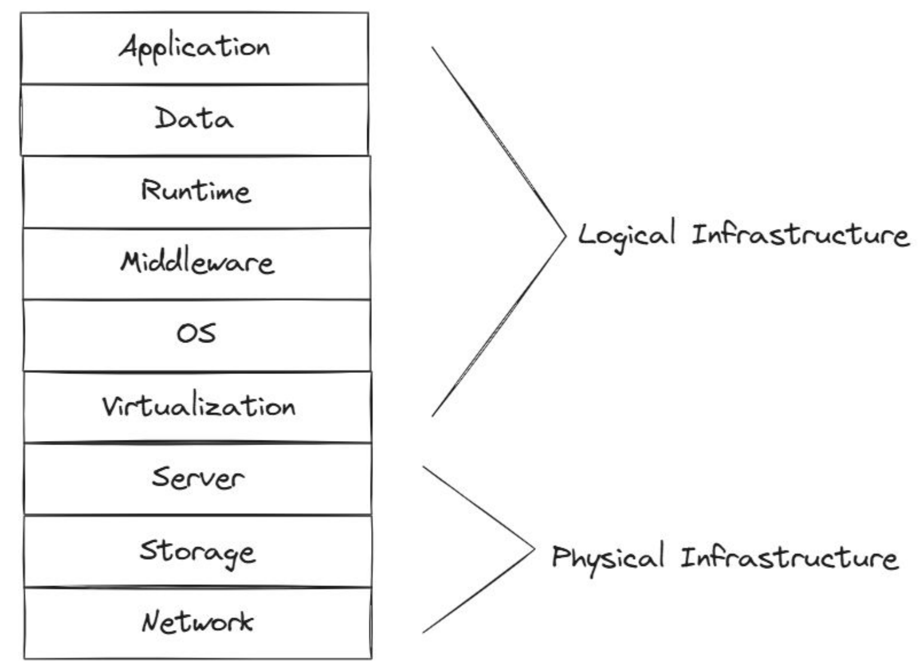
Computer Networks for Developers

- Importance of computers
  - **Development** - Developers need a local environment for building and testing their applications
  - **Deployment** - For other people to use the applications, they need to have equipment to access and run the systems
- Why it’s important to know how systems work
  - **Performance Optimization -** Lets developers make better decisions that take the effects of hardware and software environments into account.
  - **Troubleshooting and Debugging -** Allows developers to differentiate between hardware limitations and coding bugs
  - **Deployment -** Developers are able to understand the type of environment the deployed application will need to run in.

Computer Architecture for Developers

- Computer systems are made up of 3 main components
  - **Hardware** - Physical components responsible for performing complex operations.
  - **Networking** - Hardware that allows communication between different systems
  - **Software** - Tools that facilitate the communication between the system user and the hardware
- We often refer to the components that make up a computer as the infrastructure.
  - **Physical Infrastructure -** All of the components that we can see and touch (Hardware and networking equipment)
  - **Logical Infrastructure -** The parts of the system that we interact with electronically (software)

Computer Architecture for Developers



Application

Data

Runtime

Middleware

OS

Virtualization

Server

Storage

Network

Logical Infrastructure

Physical Infrastructure

Computer Architecture for Developers

- **Physical Infrastructure**
  - **Server -** A single computer that is made up of hardware components like **RAM, CPU and GPUs.**
  - **Storage -** Hardware for long term storage like **Hard Drives** and **Solid State Drives.**
  - **Networking -** Networking hardware like **Routers, Switches** and **Cables**.
- **Logical Infrastructure**
  - **Virtualization -** A software layer for creating logical servers that run on top of physical servers
  - **Operating System -** A tool that converts system instructions into operations that can be performed by hardware
  - **Middleware -** Software that runs on an operating system that allows for communication between other pieces of software running on a system
  - **Runtime -** Tools that allow a certain programming language to be translated into something that the operating system can understand and execute.
  - **Application and Data -** The applications that run on an OS that handle user interactions and store user data.

Understanding the Physical Infrastructure

- For developers, the physical infrastructure is usually packaged into a single device like a Laptop, Desktop, or a server in a data center.
- When deploying applications, developers have the following concerns about the physical infrastructure:
  - **Server -** Whether the server has enough CPU, RAM and GPU to run the application for each user without any noticeable performance issues.
  - **Storage -** Whether the server has enough storage to handle all of the application data, and whether the storage is fast enough to reduce any latency that is a result of slow read or write times
  - **Networking -** Whether the server has access to the internet to access external resources and whether the internet has access to the service to allow access to user.

Understanding the Logical Infrastructure

- **Logical Infrastructure** allows us to interact with hardware.
- As a developer, there is a high likelihood that you will need to interact with every logical infrastructure layer at some point in your career.
- Understanding the layers will allow you to build better applications and make it easier to manage cloud based applications.

Virtualisation

- Like a normal desktop PC, a server can only run a single operating system for it's given hardware
- Virtualisation allows us to increase the number of operating systems we can run on a single server
- With virtualisation, we're able to divide the hardware resources and create virtual servers
- Each virtual server will be logically isolated from other servers, this provides the folloiwng benefits:
  - **Security**
    - Virtual servers don't have direct access to hardware, any malware that targets hardware won't be effective
    - Malware in a single virtual server can't be spread to other servers unless they are communicating through the network
  - **Deployment**
    - Different services can run in their own environments without having their dependencies interrupted by other services.
  - **Portability**
    - Servers can be brought up and torn down at any time

Operating System

- Operating systems translate software operations to hardware instructions.
- We can run an operating system directly on hardware or we can run it through virtual environments depending on the use case.
- Different operating systems come with different built in features and different levels of support for certain tools

CoGrammar

Middleware

- An operating system consists of different applications and tools.
- Certain applications need to communicate with one another, but because they are isolated systems or built with different languages are runtimes, they can’t.
- **Middleware** are tools that facilitate communication between different applications
- Examples of middleware include
  - **MQTT** - For sending messages between applications
  - **Data Access -** Provides applications with a means to connect to database engines

CoGrammar

Runtime

- The code that we write cannot be automatically translated by the operating system.
- A **runtime** provides the tools required to translate code into something that a specific operating system can understand.
- Common runtimes include
  - **.NET -** C#, F#, visual basic
  - **Node -** JavaScript, TypeScript
  - **JVM -** Java, Scala, Kotlin

Application and Data

- The application and data layers represent the applications that we interact with
- These application will run on a specific **runtime** and talk to different pieces of **middleware.**
- Examples
  - Visual Studio Code
  - A custom Python application
  - MongoDB Shell

Let's take a break

CoGrammar

Components of an Application

- Modern applications are usually made up of:
  - **Backend -** Handling all of the operational logic
  - **Frontend -** Handling user interactions
  - **Database -** Storing structured / semi-structured data (eg, user information)
  - **Object or File Storage -** Storing unstructured data (eg, files, images, audio)
- Each component that makes up a full-stack application will have its own dependencies that allows it to operate

Distributed Services

- A **distributed service** is a service that operates across multiple networked computers instead if a single computer
- Each service run on its own machine with its own operating system and independent dependences
- Benefits
  - **Scalability -** Each service can independently increase it’s capacity based on traffic
  - **Fault Tolerance** - If a single service goes down, the rest of the system can continue and another instance of the same service can be brought up
- Distributed computing is the most common approach used when deploying applications to the cloud

Configuring Distributed Services

- **Physical Infrastructure and Virtualization**
  - When deploying to the cloud, one just need to be concerned about the resources allocated to the virtual machine.
  - The required hardware will depend on the services that you’re trying to run.
- **Operating System**
  - Linux is the most commonly used operating system when deploying services
  - You can choose a specific flavour of Linux based on the system requirements and built in tooling required.
- **Middleware and Runtimes**
  - In order to run the service, certain runtimes and tools will need to be installed.
  - Some services that are already compiled for the specific operating system might not require any additional runtimes and middleware.
- **Application and Data**
  - Once the environment is configured, the actual service that will run will need to be uploaded.
  - The may include uploading the source code or uploading a compiled application
  - The environment variables and any other secrets would also need to be setup

Deploying a Distributed Service

- After the VM and OS have been configured, you will need to install the dependencies for the service.
- Approaches:
  - **Manual Scripting -** SSH into the server and run the commands to install each dependency.
  - **Automated Scripting -** SSH into the server and run a bash script that will go over the installation steps
  - **IaaC (Advanced) -** Use **Ansible** or **Terraform** to describe the services that need to be configured on a single or multiple VMs to deploy a single service.

Bash Scripting

- When deploying service, we want the process to be fast and repeatable
- Bash scripting is the simplest automation technique we can use to configure servers
- To create a basic bash script, you just need to have an understanding of basic UNIX commands
- Most third party services that you might want to install on your VM will include a step that install a bash script from an external source and runs the commands to simplify the process of configuring the service
- If you’re setting up your own service, you can either use SCP to upload the bash file, download the script from a public repository, or write the script on the VM using vim/nano

Bash Scripting

- When a bash script is moved to the VM, it can’t be run, we need to set some permissions.
- **chmod +x <file-name>** - allows for file execution
- **./<file-name>** - By calling the file name in the terminal, the file will be executed.

CoGrammar

Communication

- Once the server has been setup, you'll want services to communicate with one another.
- **HTTP** is the most common approach for allowing communication.
- **LocalHost**
  - Allows services within the same server to communicate
  - Services will have the IP address: 127.0.0.1
  - Each service will need to run on it's own port
  - No service outside the server will have access to anything running on 127.0.0.1
- **Public**
  - Allows a service to be accessible over the internet
  - Will be deployed to **0.0.0.0** locally and will be publically accessible through the server's public IP address
  - Each service will run on it's own port, the port will need to be exposed through firewall rules so that external traffic can access it

What is a distributed system?

A. Where all of the services that make up a single system are hosted on the same server
B. When a full system is made up of different applications
C. When each service that makes up an application is deployed on a it’s own server.
D. None of the above

What is a way of making the setup process for a server faster?

A. Using Bash scripts
B. Running terminal commands
C. Using cURL
D. None of the above

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education