

CHAPTER-1

DAY TO DAY ACTIVITIES

Unn



Basavarajeshwari Group of Institutions
BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT
Autonomous Institute under VTU - Belagavi
"Jnana Gangotri" Campus, Bellary-Hospet Road, Near Allipura Village,
BALLARI - 583 104 (Karnataka)
Ph: 08392-237167/237153 Fax: 237197, e-mail: bitmbly@gmail.com
Website: www.bitm.edu.in



Internship Program on Python for BE-3rd Sem students
From 9th sep to 28th sep 2024 .

Student Name: Shaguftha Nikhar USN No: 3BR23CD087 Branch: CSE-DS

INDEX PAGE			
Day	Date	Content Covered	Signature of the faculty in-charge
1	09.09.24	Introduction to Python, Setup & Installation, First Python Program, Variables, Data Types, and Basic I/O	
2	10.09.24	Control Structures: If-else, Loops, Functions and Modules	
3	11.09.24	Lists, Tuples, and Dictionaries, File Handling	
4	12.09.24	Exception Handling, Practice exercises on Python basics	
5	13.09.24	Introduction to OOP, Classes, and Objects	
6	14.09.24	Inheritance, Polymorphism, and Encapsulation	
7	15.09.24	Abstract Classes and Interfaces	
8	17.09.24	Practice exercises on OOP concepts	
9	18.09.24	Introduction to DSA, Arrays, and Linked Lists	
10	19.09.24	Introduction to DSA, Arrays, and Linked Lists	
11	20.09.24	Introduction to stack and queue	
12	21.09.24	Practice exercise on basic concepts (Reduce lambda function and list comprehension)	
13	23.09.24	Introduction to tree data structure	

14	24.09.24	Introduction to graph data structure	
15	25.09.24	Searching algorithms Project Building & Presentations	
16	26.09.24	Project Building & Presentations	
17	27.09.24	Project Building & Presentations	
18	28.09.24	Project Building & Presentations	

CHAPTER-2

COMPANY PROFILE

Company Name: EZ Trainings and Technologies Pvt. Ltd.

Introduction:

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

Mission:

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

Services:

College Trainings:

- Tailored training programs designed to enhance the employability of students.
- Industry-aligned curriculum covering technical and soft skills.
- Placement assistance and career guidance.

Development Projects:

- End-to-end development services, from ideation to execution.
- Expertise in diverse technologies and frameworks.
- Custom solutions to meet specific business needs.

Locations: Hyderabad | Delhi NCR

At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence

CHAPTER-3

ABSTRACT

- Virtual network configuration refers to the process of creating and managing virtualized network resources within a cloud or data center environment.
- This approach enables the abstraction of physical network infrastructure, allowing for greater flexibility, scalability, and efficiency in managing network services.
- Key components include virtual switches, routers, and firewalls, which facilitate communication between virtual machines and external networks.
- The configuration process involves defining network topologies, assigning IP addresses, and implementing security protocols.
- Advanced features such as network slicing, dynamic resource allocation, and automated provisioning enhance performance and reduce operational costs.
- By leveraging software-defined networking (SDN) and network function virtualization (NFV), organizations can optimize their network resources to meet evolving demands and improve overall reliability.
- At the core of virtual network configuration are technologies such as software-defined networking (SDN) and network function virtualization (NFV).
- These technologies decouple network functions from hardware, allowing for the dynamic allocation of resources and simplified management.
- Key components include virtual switches, routers, and firewalls, which facilitate communication among virtual machines (VMs) and between VMs and external networks.
- The configuration process involves several critical tasks:
 - 1. ***Topology Design***: Defining the virtual network layout, including subnets, VLANs, and interconnections.
 - 2. ***IP Address Management***: Assigning and managing IP addresses for virtual devices to ensure seamless communication.
 - 3. ***Security Configuration***: Implementing firewalls, access controls, and encryption to protect data in transit.
 - 4. ***Performance Optimization***: Using monitoring tools to analyze traffic patterns and optimize resource allocation based on usage.
- Advanced features such as network slicing allow for the creation of multiple virtual networks within a single physical infrastructure, enhancing resource utilization.
- Automation tools enable rapid provisioning and scaling of network services, significantly reducing deployment times and operational overhead. .
- By leveraging virtualization technologies, organizations can achieve improved performance, enhanced security, and better alignment of network resources with business needs.
- In conclusion, virtual network configuration is a vital component of modern IT infrastructure, supporting the agility and efficiency required by today's enterprises. By leveraging virtualization technologies, organizations can achieve improved performance, enhanced security, and better alignment of network resources with business needs.

CHAPTER-4

INTRODUCTION OF THE PROJECT

- Virtual Network Configuration is the process of setting up and managing virtualized networking environments that mimic traditional physical networks.
- This involves defining the necessary components such as IP addresses, subnets, routing tables, and security rules to ensure resources within the network can communicate securely and efficiently.
- Purpose:
- The main purpose of virtual network configuration is to enable seamless communication between virtual resources (like virtual machines, containers, databases) while providing flexibility, scalability, and security.
- Virtual networks allow organizations to logically separate and manage workloads, reducing the need for physical networking hardware.
- Goals:
- Isolation: Segregate environments (e.g., development, testing, production) or workloads for better management and security.
- Connectivity: Facilitate communication between virtual resources, other networks, and external environments (like on-premises systems or the internet).
- Scalability: Easily scale the network to accommodate growing infrastructure demands without physical limitations.
- Security: Implement firewall rules, access control, and encryption to protect network traffic and prevent unauthorized access.
- Cost Efficiency: Reduce the need for physical network devices, lowering infrastructure costs and simplifying network management.
- Overall, virtual network configuration helps create a flexible, secure, and cost-effective networking environment that supports modern IT infrastructure needs.
- NetworkSetting: A class representing individual network settings.
- NetworkManager: Manages all network settings and encompasses the CRUD operations.
- VirtualNetwork: Handles setting up and managing virtual networks.

CHAPTER-5

MODULE DESCRIPTION

This module focuses on the principles and practices of configuring virtual networks within various environments, including data centers, cloud platforms, and enterprise networks.

Participants will learn how to design, implement, and manage virtual networks, ensuring optimal performance, security, and scalability.

The goal is to design and implement functionality for managing virtual network configurations, including the setup, update, and management of network settings through CRUD (Create, Read, Update, Delete)

operations.Scope:CRUD Operations for Network Settings:Implement basic CRUD operations for network settings, which include creating new configurations, reading or retrieving existing configurations, updating them as needed, and deleting unnecessary configurations.

Setup and Configuration of Virtual Networks:Develop functionality to initialize and configure virtual networks (setup_virtual_networks(network_config)), ensuring that different network parameters such as IP ranges, subnets, gateways, etc., are properly configured.

Updating Network Settings:Include the ability to modify existing network settings (update_network_settings(settings_id)).

This allows for dynamic updates to network configurations when changes in the network environment occur, such as modifying IP address ranges, enabling/disabling specific protocols, or adjusting security settings.

Functionality: Virtual Network Configuration

CRUD Operations for Network Settings:

Create (C): Allows users to add new network configurations. This could involve defining IP ranges, subnet masks, gateways, DNS settings, and security rules.

Example: Creating a new virtual network configuration for an isolated subnet within a cloud environment.

Read (R): Retrieve and view the existing network configurations.

Example: Displaying the details of an existing network setup such as IP ranges and active security protocols

.Update (U): Modify existing network settings (e.g., adjusting the subnet mask, updating DNS servers).

Example: Changing the IP range of an existing virtual network.

Delete (D): Remove unwanted or outdated network configurations.

Example: Deleting an unused virtual network to free up resources.

Setup Virtual Networks (setup_virtual_networks(network_config)):

This function sets up and initializes virtual networks based on given configurations. It involves configuring routing rules, security groups, and network interfaces.

Example: Setting up a virtual network to connect two different cloud instances, ensuring they can communicate securely.

It handles key network parameters: IP Address ranges (CIDR blocks)

Subnets

Gateways

Security groups

Update Network Settings (update_network_settings(settings_id)): This function allows dynamic updating of network settings for specific network configurations based on their unique ID (settings_id).

Example: Adjusting the bandwidth or latency settings for a virtual private network (VPN).

Functionality involves:

Editing the configurations such as modifying subnets or enabling/disabling firewall rules. Rolling out updates without interrupting network traffic.

Additional Functionalities : Validation of Network Configuration: Ensure that any network configuration created or updated passes validation rules (e.g., no overlapping IP ranges, valid CIDR blocks).

Monitoring and Alerts: Implement network monitoring to track the performance and health of the virtual networks, and send alerts in case of issues like connectivity drops or security breaches.

Security Features: Integration of security settings like firewalls, encryption, and access control lists (ACLs) to control network traffic between different virtual networks.

Logging and Auditing: Maintain a history of all network changes, including who made the change, when, and what settings were modified for auditing and compliance purposes.

Real-Life Example: Virtual Network Configuration for IT Infrastructure:

Imagine an IT department within a medium to large organization that manages multiple virtual networks for various departments, such as finance, human resources, and development. The Virtual Network Configuration system helps the IT team efficiently configure, manage, and update virtual networks, ensuring that network resources are used optimally and securely.

1. **CRUD: Network Settings** The system provides capabilities to create, read, update, and delete network settings, allowing the IT team to maintain accurate and current configurations for each virtual network.

Example:

- **Create:** When a new department is formed, the IT team creates network settings for the finance department, specifying the IP address range, subnet mask, and other relevant configurations.
- **Read:** The IT staff can view existing network settings to ensure they align with the department's requirements. For instance, they check the settings for the Development Network to verify the allocated resources.

- Update: If the organization undergoes a merger, the IT team can update the network settings to accommodate new departments, changing the IP address range and adding VLAN configurations as necessary.

- Delete: When a project is completed, the corresponding virtual network settings are deleted from the system to avoid clutter and confusion.

2. `setup_virtual_networks(network_config)` The tool enables the IT team to set up and configure virtual networks based on specific requirements and configurations.

Example:

- The IT department needs to set up a new virtual network for a project team working on a sensitive application. They use the system to input the network configuration details, such as security protocols, bandwidth limits, and access permissions.

- The configuration is processed, and the virtual network is created, providing isolated resources for the project team while maintaining compliance with organizational policies.

3. `update_network_settings(settings_id)` The system also allows for easy updates to existing network settings, ensuring that the configurations remain relevant and effective as needs change.

Example:

- The company decides to implement stricter security measures. The IT team identifies the network settings for the HR Network and updates them to include enhanced encryption protocols and stricter access controls.

- The system logs the changes made, providing a historical record of modifications for compliance and auditing purposes.

The Virtual Network Configuration system is an essential tool for IT departments by:

- Providing an organized and structured way to manage network settings across various virtual networks, ensuring that configurations are current and accurate.

- Enabling seamless setup of virtual networks tailored to specific departmental needs, allowing for flexibility and isolation of resources.

- Facilitating quick and efficient updates to network settings in response to changing organizational requirements, maintaining security and compliance. This system ensures that the organization's network infrastructure is robust, secure, and adaptable, ultimately supporting business operations and enhancing productivity across departments.

We'll use a Python dictionary to represent our network configurations. Each network will have a unique `network_id` and associated configuration:

`setup_virtual_networks(network_config)` This function sets up a new virtual network if does not exist, otherwise it suggests updating the existing network.

`network_settings(settings_id)`: This function allows you to update network settings based on `settings_id`. You can pass a dictionary (`new_config`) containing the new configuration values and the function will apply these updates to the existing network.

CHAPTER-6

ALGORITHM

- 1.Start
- 2.Initialize network settings (example: a dictionary to hold networks).

Create Network:

- 1.If create_network(network_id, name, ip_range, status) is called:
 - Check if network_id exists in network_settings.
 - ~If exists, print "Network with ID already exists."
 - ~Otherwise, add the new network to network_settings and print confirmation.

Read Network:

- 1.If read_network(network_id) is called:
 - Retrieve and print the network information for network_id.
 - ~If network_id doesn't exist, print "Network with ID does not exist."

Update Network:

- 1.If update_network(network_id, name, ip_range, status) is called:
 - Check if network_id exists in network_settings.
 - ~If exists, update the network fields (name, IP range, status) as provided, and print confirmation.
 - ~If not, print "Network with ID does not EXISTS".

Delete Network:

- 1.If delete_network(network_id) is called:
 - Check if network_id exists in network_settings.
 - ~If exists, delete the network from network_settings and print confirmation.
 - ~If not, print "Network with ID does not exist."

Setup Virtual Network:

- 1.If setup_virtual_network(network_id, name, ip_range, status) is called:
 - Check if network_id exists in network_settings.
 - ~If not, call create_network() and set up the new network.

~Otherwise, print that the network already exists.

Update Network Settings:

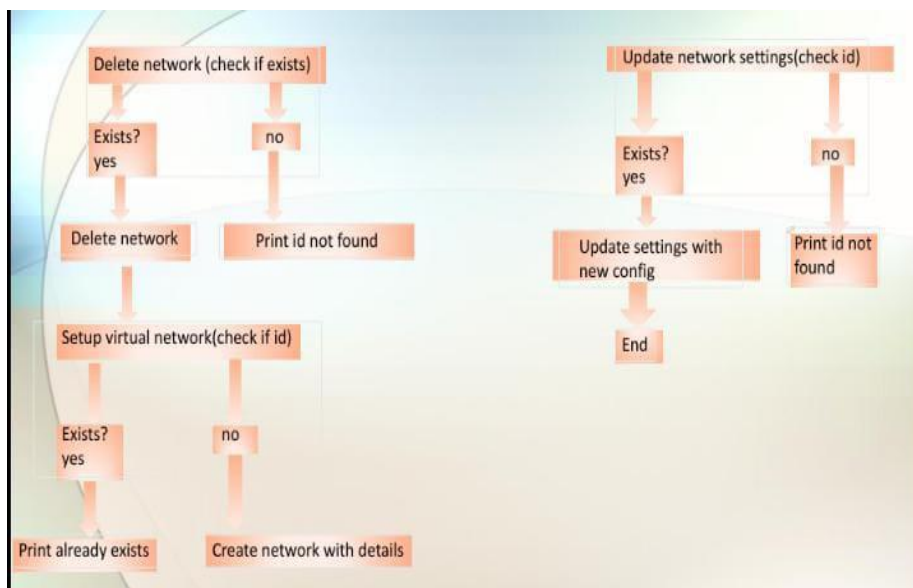
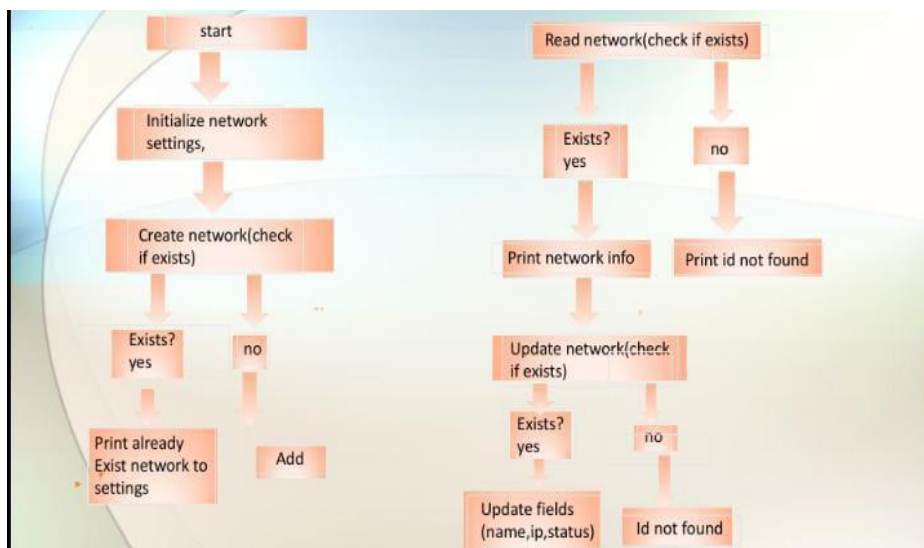
1.If update_network_settings(settings_id, new_config) is called:

-Check if settings_id exists in network_settings.

~If exists, update the network settings with new_config and print confirmation.

~If not, print "Network with ID does not exist."End

FLOW CHART:



CHAPTER-7

SOURCE CODE

```
network_settings = {
    1: {'name': 'Network_1', 'ip_range': '192.168.1.0/24', 'status': 'active'},
    2: {'name': 'Network_2', 'ip_range': '192.168.2.0/24', 'status': 'inactive'}
}

#print(network_settings)

def create_network(network_id, name, ip_range, status):
    if network_id in network_settings:
        print(f"Network with ID {network_id} already exists.")
    else:
        network_settings[network_id] = {'name': name, 'ip_range': ip_range, 'status': status}
        print(f"Network '{name}' created with ID {network_id}.")
print(create_network)

def read_network(network_id):
    network = network_settings.get(network_id)
    if network:
        print(f"Network {network_id}: {network}")
    else:
        print(f"Network with ID {network_id} does not exist.")

def update_network(network_id, name=None, ip_range=None, status=None):
    if network_id in network_settings:
        if name:
```

```

        network_settings[network_id]['name'] = name
    if ip_range:
        network_settings[network_id]['ip_range'] = ip_range
    if status:
        network_settings[network_id]['status'] = status
    print(f"Network {network_id} updated.")
else:
    print(f"Network with ID {network_id} does not exist.")

def delete_network(network_id):
    if network_id in network_settings:
        del network_settings[network_id]
        print(f"Network {network_id} deleted.")
    else:
        print(f"Network with ID {network_id} does not exist.")

def setup_virtual_network(network_id, name, ip_range, status):
    if network_id not in network_settings:
        create_network(network_id, name, ip_range, status)
        print(f"Virtual network '{name}' is set up with IP range {ip_range} and status {status}.")
    else:
        print(f"Network with ID {network_id} already exists, consider updating it.")

def update_network_settings(settings_id, new_config):
    if settings_id in network_settings:
        network_settings[settings_id].update(new_config)

```

```
    print(f'Network {settings_id} updated with new configuration: {new_config}')
else:
    print(f'Network with ID {settings_id} does not exist.')

create_network(1, 'Network_1', '192.168.1.0/24', 'active')
read_network(1)
update_network(1, name='New_Network_1', status='inactive')
delete_network(1)
setup_virtual_network(3, 'Network_3', '192.168.3.0/24', 'active')
new_settings = {'ip_range': '192.168.1.0/25', 'status': 'inactive'}
update_network_settings(1, new_settings)
```

CHAPTER-8

OUTPUT

<function create_network at 0x03A7B438>

Network with ID 1 already exists.

Network 1: {'name': 'Network_1', 'ip_range': '192.168.1.0/24', 'status': 'active'}

Network 1 updated.

Network 1 deleted.

Network 'Network_3' created with ID 3.

Virtual network 'Network_3' is set up with IP range 192.168.3.0/24 and status active.

Network with ID 1 does not exist.

<function create_network at 0x03A7B438>

Network with ID 1 already exists.

Network 1: {'name': 'Network_1', 'ip_range': '192.168.1.0/24', 'status': 'active'}

Network 1 updated.

Network 1 deleted.

Network 'Network_3' created with ID 3.

Virtual network 'Network_3' is set up with IP range 192.168.3.0/24 and status active.

Network with ID 1 does not exist.

CHAPTER-9

CONCLUSION

- Providing an organized and structured way to manage network settings across various virtual networks, ensuring that configurations are current and accurate.
- Enabling seamless setup of virtual networks tailored to specific departmental needs, allowing for flexibility and isolation of resources.
- Facilitating quick and efficient updates to network settings in response to changing organizational requirements, maintaining security and compliance.

It ensures that the network infrastructure is robust, secure, and adaptable, ultimately supporting business operations and enhancing productivity across

- 1. AI-Driven Automation: Automate network setup, management, and self-healing using AI.
- 2. Enhanced Security: Implement context-aware security, zero trust, and AI-driven threat detection.
- 3. Advanced Monitoring: Real-time interactive maps, predictive analytics, and detailed traffic insights.
- 4. User-Friendly Interfaces: Drag-and-drop design, pre-built templates, and virtual testing environments.
- 5. Multi-Cloud Integration: Seamless networking across multiple cloud providers and edge computing support.
- 6. Scalability: Use NFV, dynamic bandwidth, and latency optimization. and so on....

CHAPTER-11

REFERENCES

<https://www.sdxcentral.com/security/definitions/data-security-in-the-cloud-best-practices/virtual-network-security-works/>

<https://docs.oracle.com/cd/E19504-01/802-5753/6i9g71m52/index.html>

<https://docstore.mik.ua//manuals/hp-ux/en/T2767-90141/ch08s01.html>

CHAPTER-11

REFERENCES

<https://www.sdxcentral.com/security/definitions/data-security-in-the-cloud-best-practices/virtual-network-security-works/>

<https://docs.oracle.com/cd/E19504-01/802-5753/6i9g71m52/index.html>

<https://docstore.mik.ua//manuals/hp-ux/en/T2767-90141/ch08s01.html>