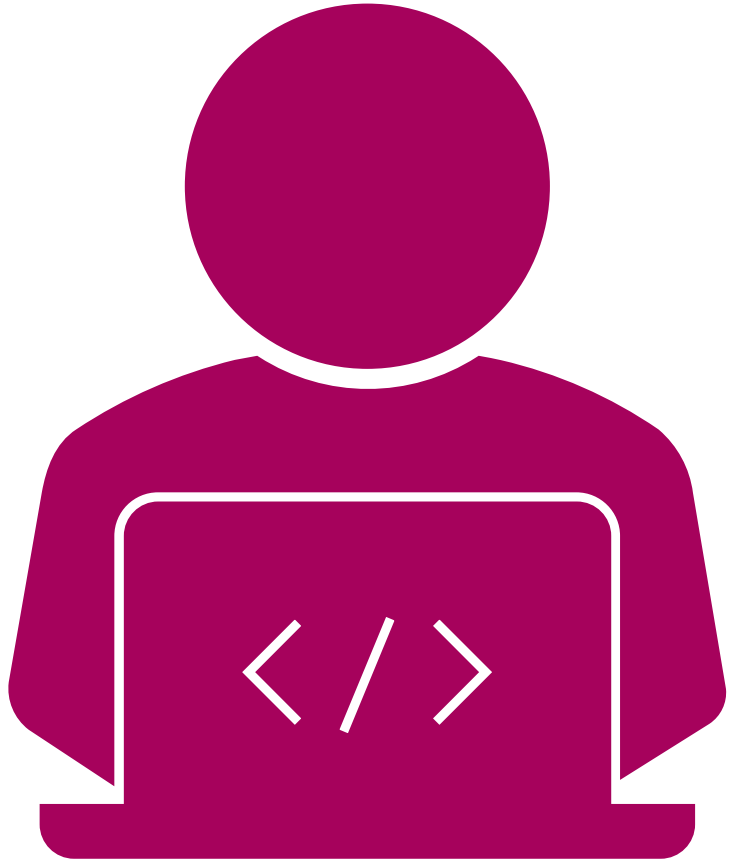
CLUSTER, IT'S TYPES AND CLUSTER CONFIGURATIONS

EXPLORING DIFFERENT HPC ARCHITECTURE

INTRODUCTION

- Name: Muhammad Shazam Razzaq
- Academic Year: 3rd Year
- Major: Computer Science
- University: Namal University
- Role: Research Assistant
- Facility: HPC Research Facility at Namal University
- Mentor: Dr. Tasadoq Hussain



Contact: 0311-6517962
Email: shazamrazzaq1234@gmail.com

OUTLINE

- What is a cluster?
- Types of Cluster
  - Cloud-Based Cluster
  - Containerized Cluster
  - Bare-Metal Cluster
- Implementation
- Demo

WHAT IS A CLUSTER?

A cluster is a group of interconnected computers or servers.

These systems work together to collectively perform tasks.

Clusters provide enhanced computational power, reliability, and scalability.

Common uses include scientific simulations, data analysis, and web hosting.

Clusters enable resource sharing, load balancing, and fault tolerance.

OUTLINE

- **What is a cluster?**
- Types of Cluster
  - Cloud-Based Cluster
  - Containerized Cluster
  - Bare-Metal Cluster
- Implementation
- Demo

TYPES OF CLUSTER CONFIGURATION

Cloud-Based Clusters: OpenStack Cluster

Containerized Clusters: Kubernetes Cluster

Bare-Metal Clusters: OpenHPC Cluster

CLOUD-BASED CLUSTER

A cloud-based cluster virtualized cluster hosted on cloud infrastructure.

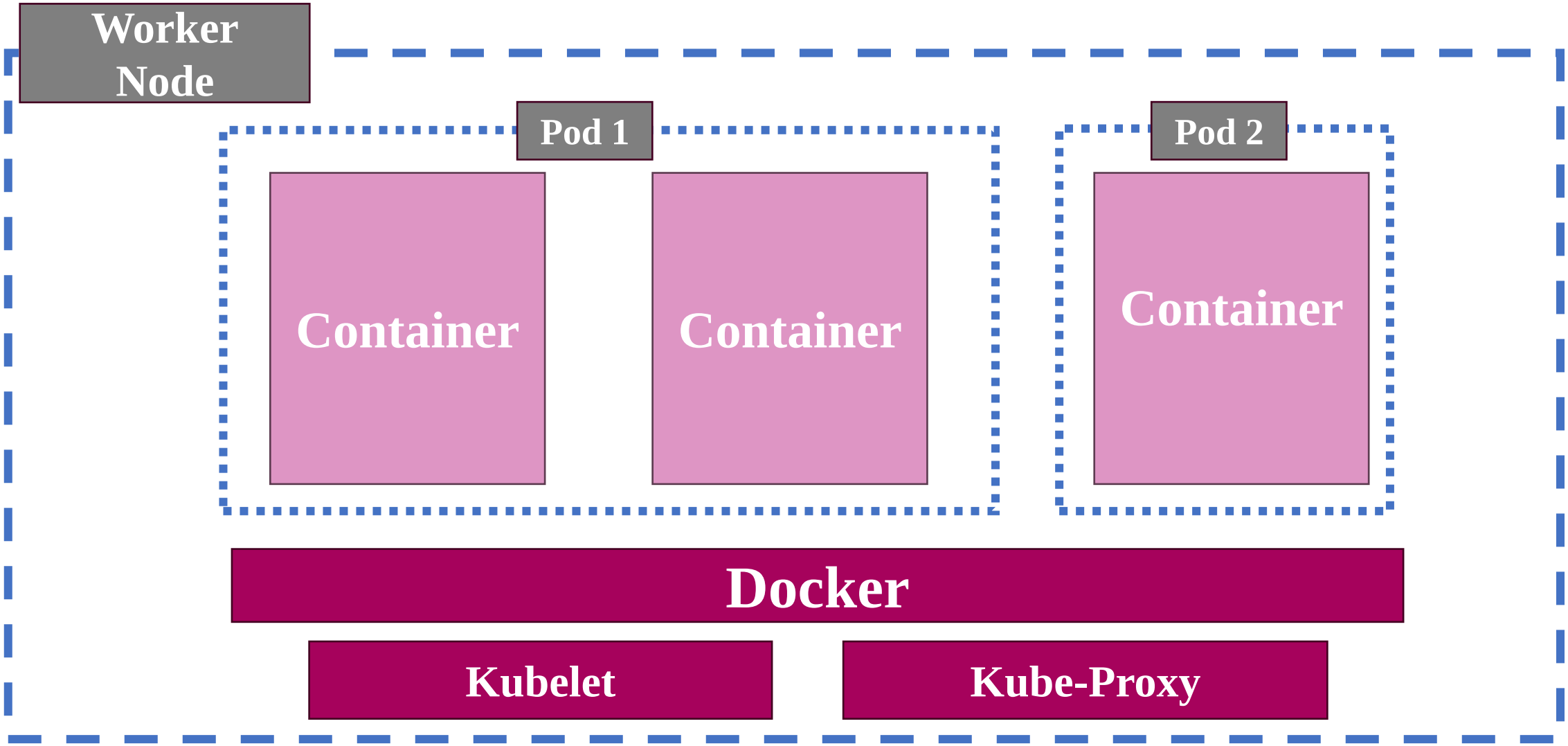
Functionality	Use Cases	Characteristics
Provides services for public and private clouds	Used by organizations to create and manage IaaS clouds.	Highly adaptable to diverse needs.
Includes compute, storage, networking, and identity.	Offers virtual machines, storage, and networking resources.	Suitable for enterprises and service providers.
Suitable for diverse cloud environments.	Enables provisioning of resources for user needs.	Offers customization but may demand setup and maintenance expertise.

CONTAINERIZED CLUSTERS

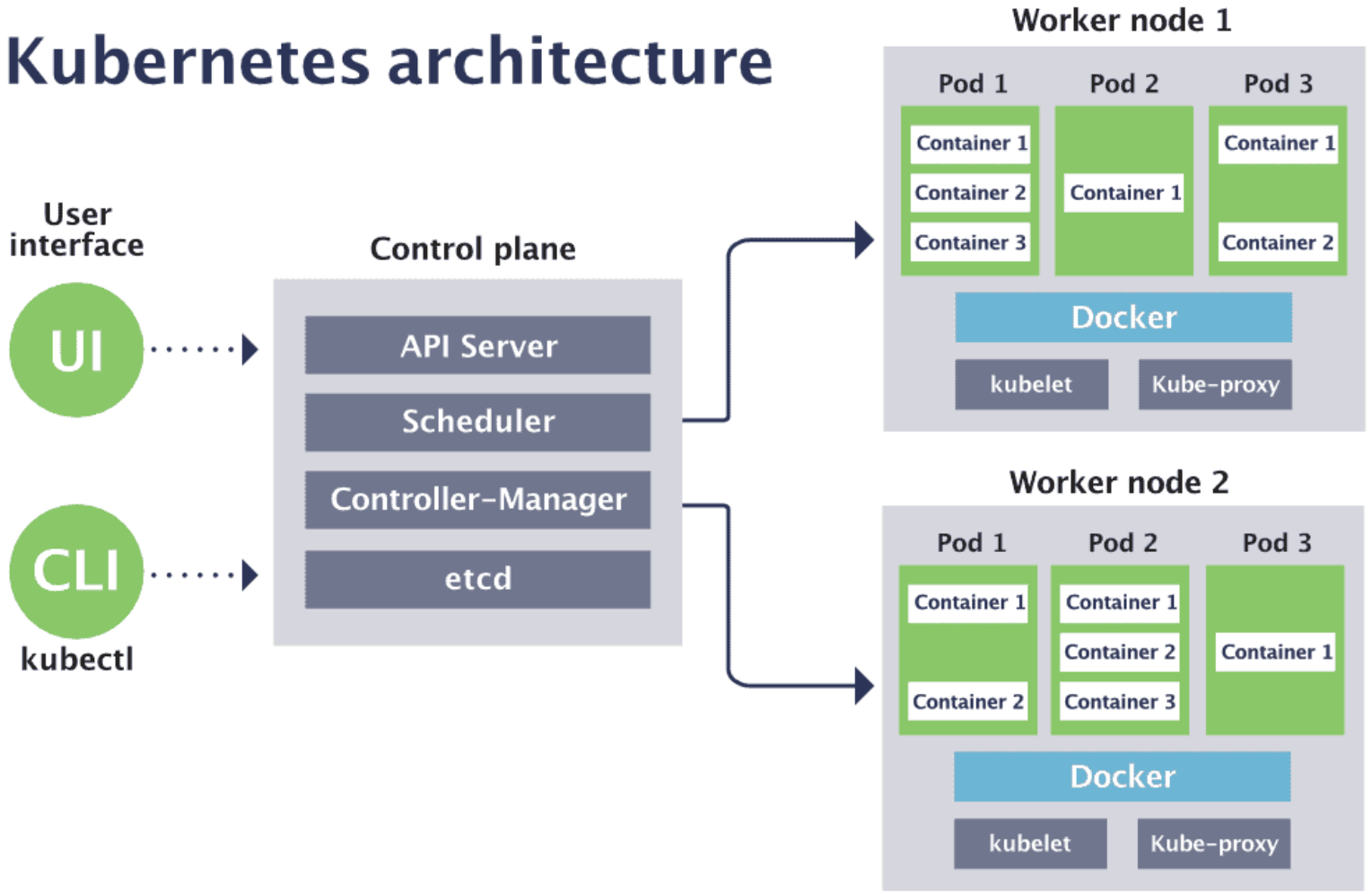
A containerized cluster encapsulates and manages applications within isolated containers for efficient deployment and scalability.

Functionality	Use Cases	Characteristics
Platform for automating deployment, scaling, and management.	Manages containerized apps uniformly.	Simplifies app deployment across diverse environments.
Focuses on containerized application environments.	Scales applications effectively to match demand.	Ensures applications remain available and responsive.
Automates scaling to handle varying workloads.	Well-suited for modern microservices setup.	Optimizes resource allocation for efficiency.

CONTAINERIZED CLUSTER INFRASTRUCTURE OF WORKER NONE



CONTAINERIZED CLUSTER INFRASTRUCTURE



BARE-METAL CLUSTERS

A bare-metal cluster is a high-performance computing framework that directly utilizes physical hardware for efficient parallel processing tasks.

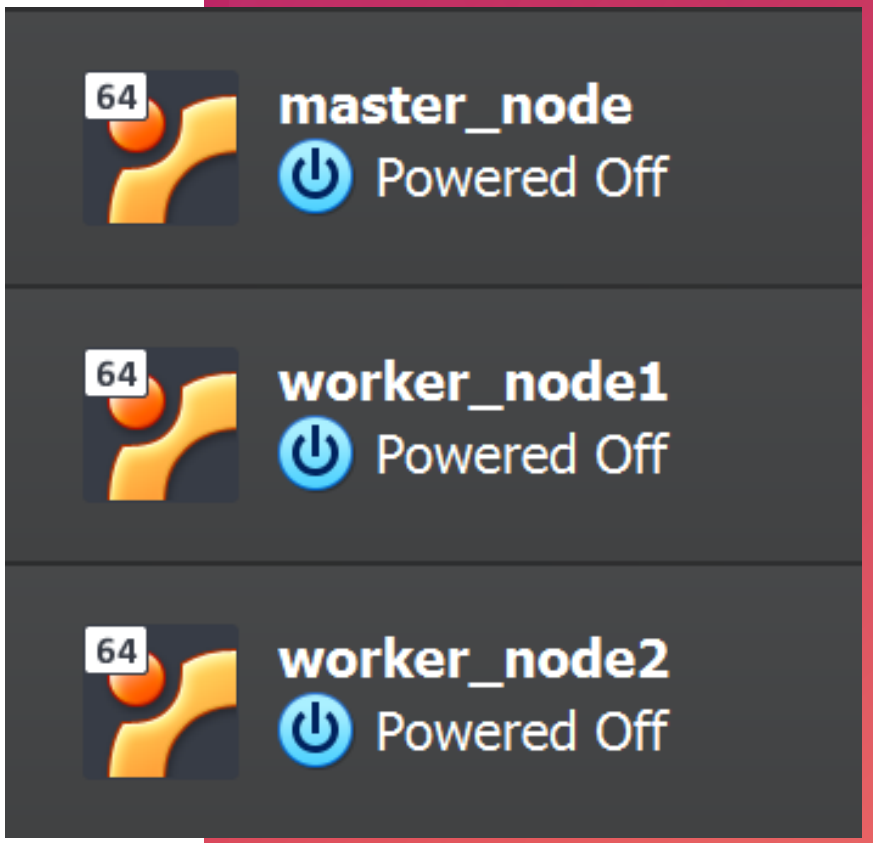
Functionality	Use Cases	Characteristics
Offers HPC software components and tools.	Utilized in research for simulations, data analysis, and modeling tasks.	Focuses on streamlining setup and configuration of HPC clusters.
Provides a collection of components for high-performance computing.	Supports resource-intensive computations in scientific fields.	Tailored for the needs of researchers in scientific computing.
Enables parallel processing for complex calculations and simulations.	Supports resource-intensive computations in scientific fields.	Efficiently manages computing resources for complex calculations.

OUTLINE

- **What is a cluster?**
- **Types of Cluster**
  - **Cloud-Based Cluster**
  - **Containerized Cluster**
  - **Bare-Metal Cluster**
- Implementation
- Demo

I M P L E M E N T A T I O N

- **Virtualization Platform:** Utilized VirtualBox for Simulation
- **Virtual Machines Setup:** Configured Three Virtual Machines:
  - Master Node: Ubuntu Desktop
  - Worker Nodes: Ubuntu Live Server
- **Hardware Resource Allocation:** Leveraged Laptop's Physical Resources for Each Virtual Machine



MASTER NODE: KEY FEATURES

Master Node Configuration: Prepared the master node with Ubuntu OS and essential packages.

Central Control: Master orchestrates tasks and distributes workloads to workers.

Cluster Management: Responsible for configuration, coordination, and monitoring.

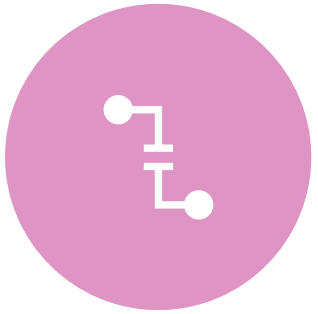
Control Plane: Hosts components like scheduler, API server, and etcd for seamless operation.

SCHEDULING IN CLUSTER MANAGEMENT

- **Task Allocation:** Master node allocates tasks to worker nodes for efficient execution.
- **Optimal Resource Usage:** Ensures tasks are assigned to nodes with available resources.
- **Load Balancing:** Distributes tasks evenly to achieve balanced resource utilization.
- **Task Prioritization:** Accommodates prioritization for crucial tasks' timely execution.
- **Automation Advantage:** Automates task allocation for seamless operation.
- **Enhanced Efficiency:** Effective scheduling enhances cluster performance and responsiveness.

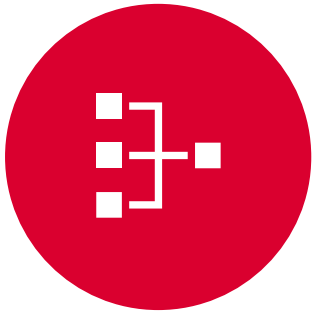
WORKER NODE: KEY FEATURES

Worker Node Configuration: Set up worker nodes using Ubuntu Live Server for streamlined performance.



Processing Power: Worker nodes execute computational tasks assigned by the master.

Data Storage: Store temporary data during processing and computations.



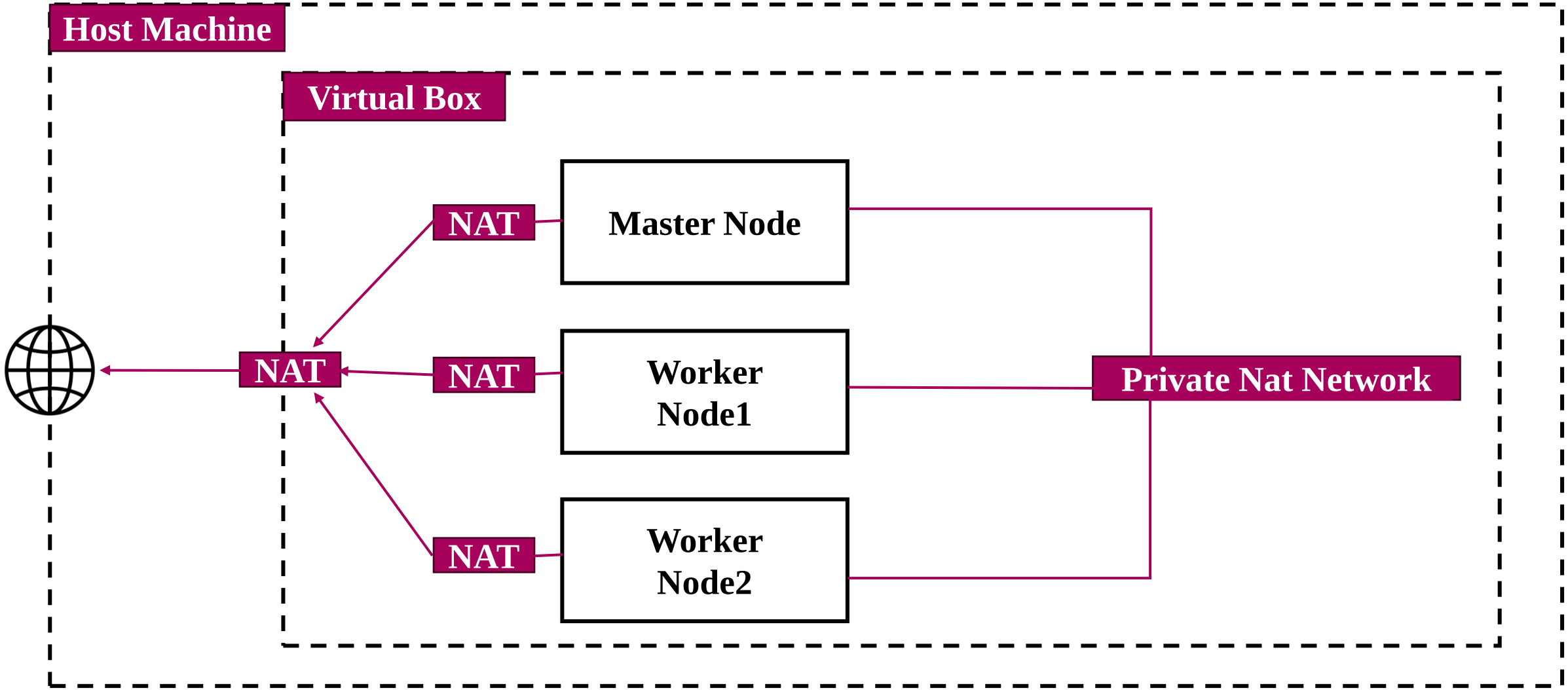
Node Independence: Multiple worker nodes collectively form the processing backbone.

IP CONFIGURATION

Private Network: Established a private network for exclusive cluster communication.

IP Assignment: Assigned static IP addresses to master and worker nodes.

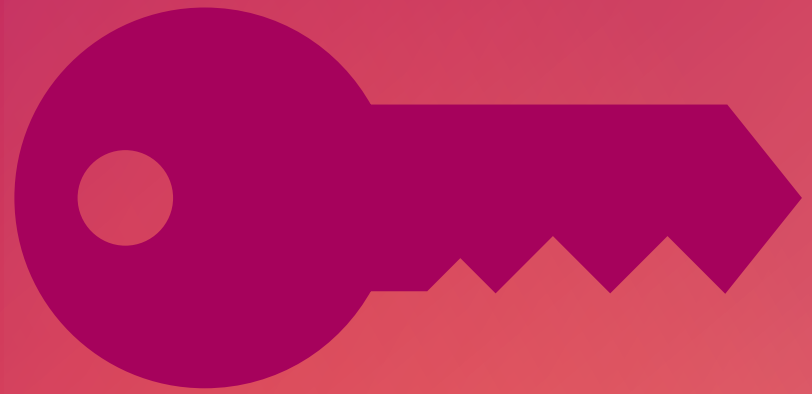
Isolation: Ensured distinct identities for each node, facilitating efficient data exchange.



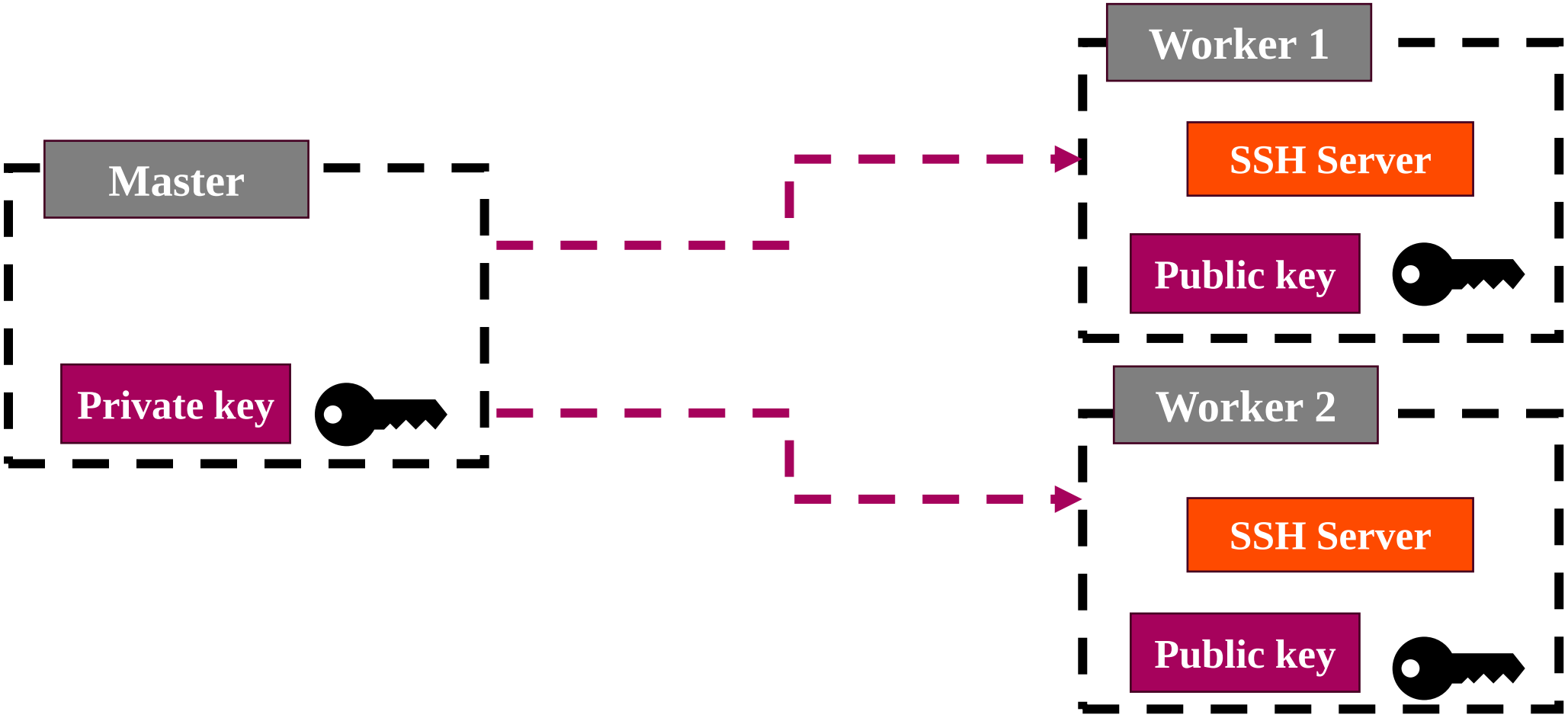
PRIVATE NETWORK

SECURE ACCESS WITH OPENSSH

- **Introduction:** Establishing Secure Communication within the Cluster
- **OpenSSH Server:** Installed OpenSSH Server Package on Master Node
- **Key Pair Generation:** Created Private and Public Key Pair
- **Key Distribution:** Shared Public Key with Worker Nodes
- **Secure Communication:** Utilized SSH for Encrypted and Protected Access
- **Importance of SSH:** Secure Shell Ensures Data Privacy and Identity Protection



CLUSTERS INFRASTRUCTURE



SOFTWARE STACK CONFIGURATION

Software stacks: Layered tools for app development.

Stack layers: Presentation, Application, Data, Infrastructure.

Interdependency: Each layer builds on the one below.

Customization: Choose components for specific needs.

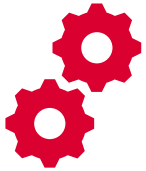
Benefits: Efficiency, Scalability, Rapid Development.

KUBERNETES COMPONENTS INSTALLATION

Install key Kubernetes components.

Docker: Container platform for app packaging.

kubelet: Node agent managing node operations.



kubeadm: Tool for cluster setup and management.

kubectl: Command-line tool for cluster interaction.

OUTLINE

- **What is a cluster?**
- **Types of Cluster**
  - **Cloud-Based Cluster**
  - **Containerized Cluster**
  - **Bare-Metal Cluster**
- **Implementation**
- Demo