AF-IAST
Skilling Pakistan

SPCAI
Sino-Pak Center for Artificial Intelligence

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY

Type-1 Hypervisor and Container Orchestration Platform on High-Performance Computing Cluster for AI and Big Data Applications

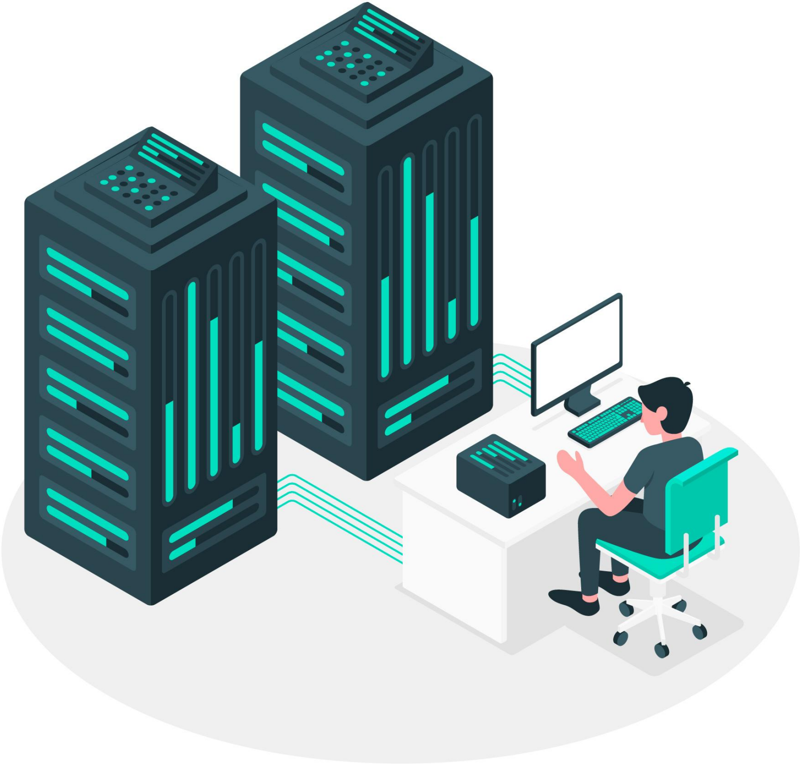
Dr. Sohail Khan
Associate Professor
Sino-Pak Center for Artificial Intelligence,
Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology

Choudhry Shehryar
Student (BS-Software Engineering, 4th Semester)

Date: 22-April 2024

Contents

Understanding Type-1 Hypervisor and Container Orchestration Platform



Type-1 Hypervisor

Type-1 hypervisors, also known as bare-metal hypervisors, are virtualization solutions that run directly on the host's hardware to manage guest operating systems. Unlike Type-2 hypervisors, they don't rely on a host operating system.

Key Features:

- Direct access to hardware resources
- High performance and efficiency
- Suitable for enterprise-level virtualization

VMware vSphere/ESXi

Microsoft Hyper-v

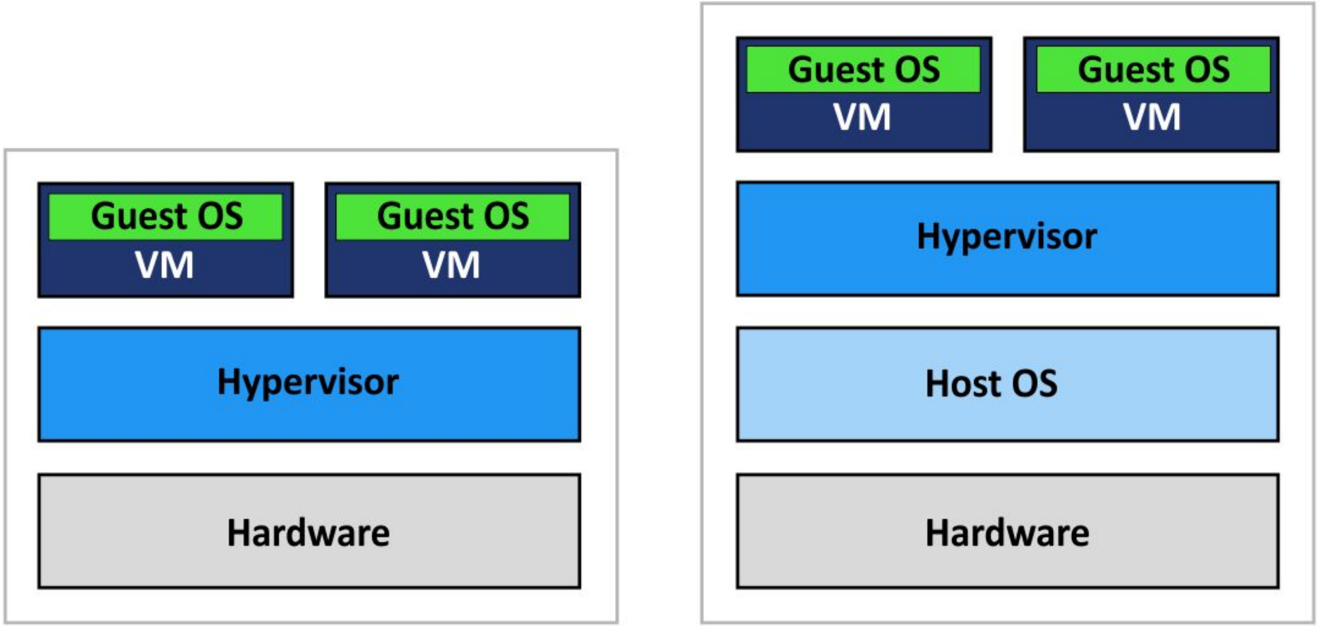
Xen Server™

PROXMOX

KVM

PROXMOX | Type-1 Hypervisor

Introduction to Proxmox Cluster



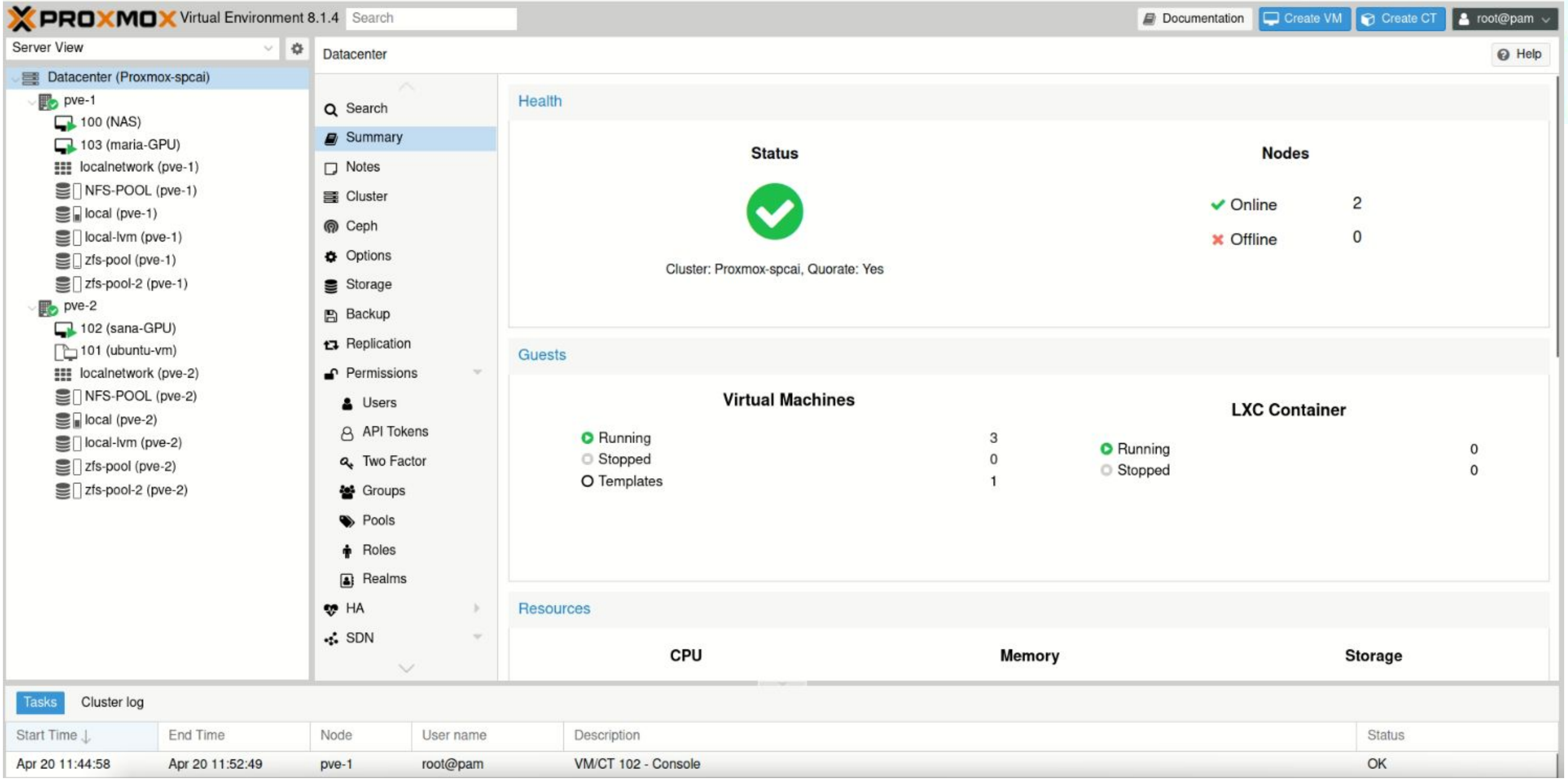
Type 1 Hypervisor
(Bare-Metal Architecture)

Type 2 Hypervisor
(Hosted Architecture)

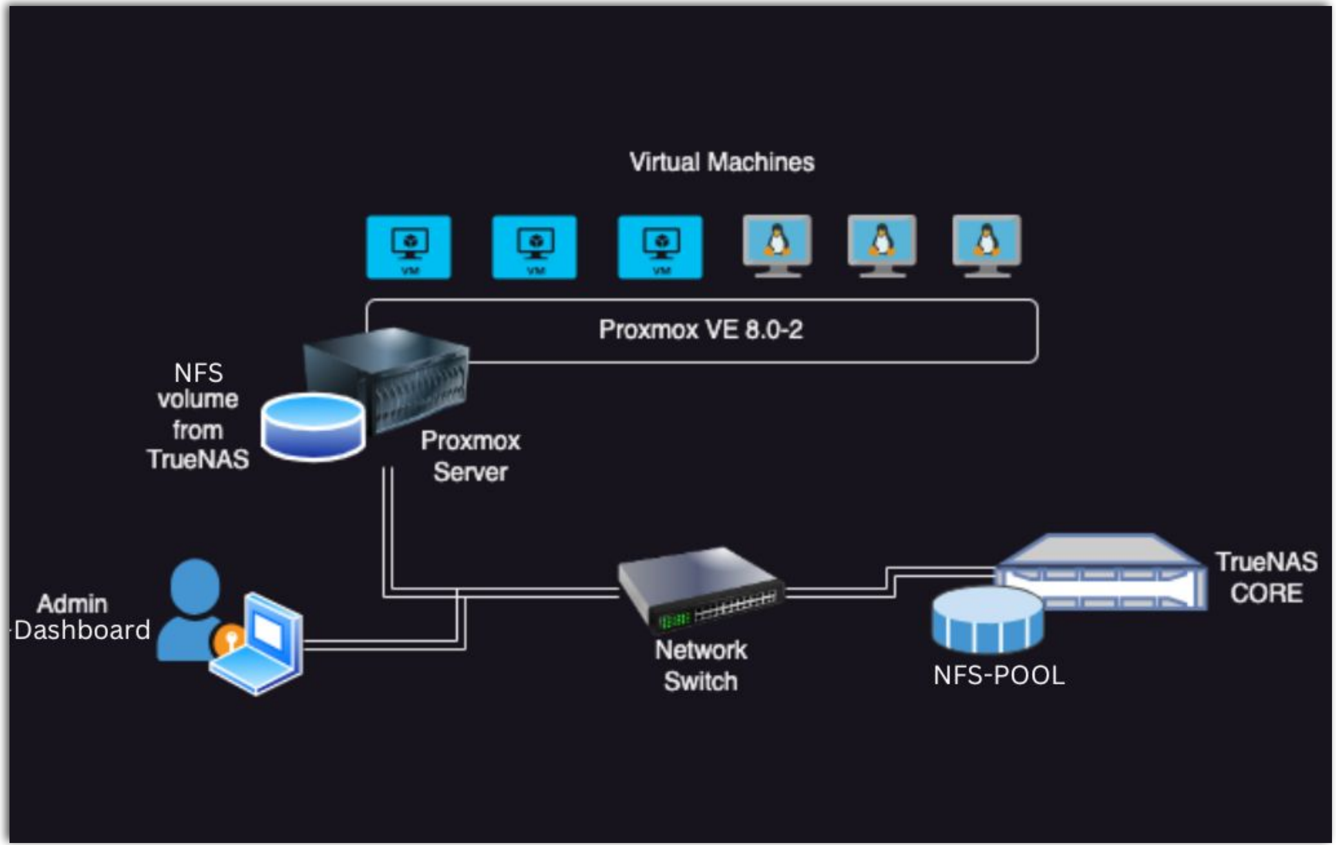
Features of Proxmox

- **Proxmox Usage:** We are using Proxmox to deploy VM and LXC Containers.
- **VM Deployment:** Proxmox hosts virtual machines, manages them and those machines can be scaled, migrated and are Highly available.
- **LXC Deployment:** Lightweight Linux containers run on Proxmox, supporting efficient application deployment and utilizing scalability features.
- **Resource Management:** Proxmox optimizes CPU, memory, and storage usage across VMs and containers.
- **High Availability:** Our Hypervisor has Features like live migration ensure minimal downtime and continuous service availability.
- **Scalability:** We can always scale up our Vms and containers according to needs.
- **Backup and Recovery:** Vms and containers images can be stored into backup drives seamlessly to avoid loss of any data in node failures.
- **Centralized Management:** Proxmox unified interface simplifies monitoring and administration tasks.

Central Dashboard for Proxmox



Architecture Overview



Explanation of High Availability Concept in Proxmox

Status

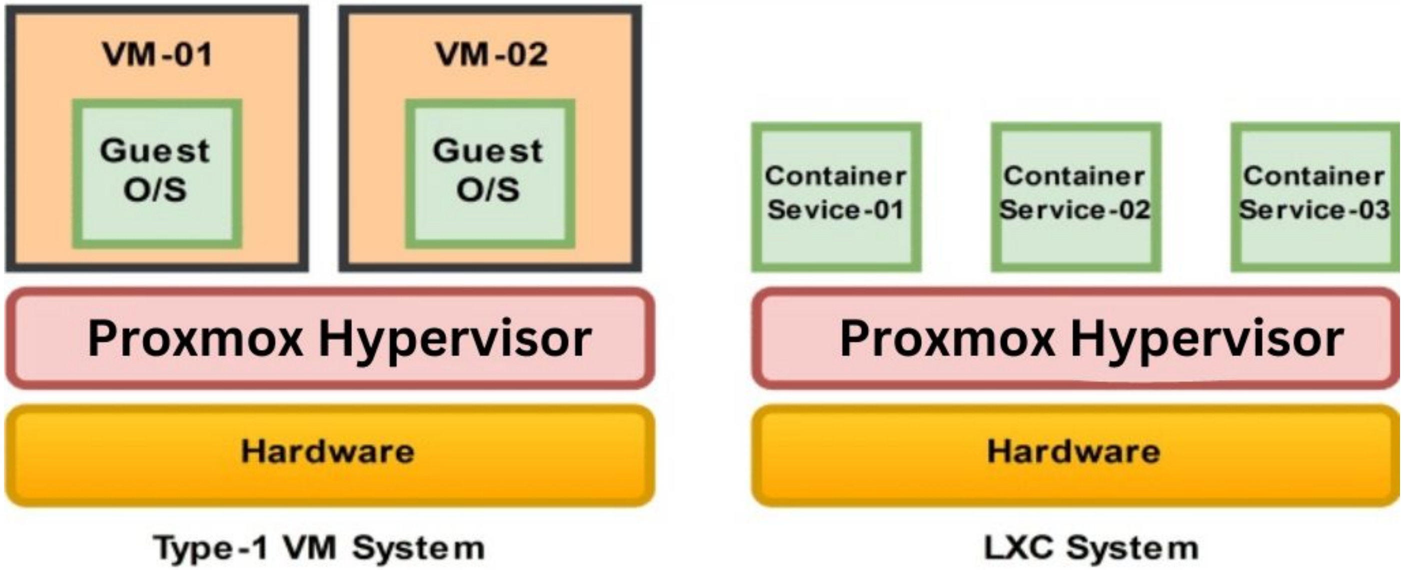
Type	Status
quorum	OK
master	pve-1 (active, Sun Apr 21 23:39:10 2024)
lrm	pve-1 (idle, Sun Apr 21 23:39:14 2024)
lrm	pve-2 (active, Sun Apr 21 23:39:13 2024)

Resources

Add Edit Remove

ID	State	Node	Name	Max. Restart	Max. Reloc...	Group	Description
ct:104	started	pve-2	ubuntu	1	1		

DEMO: Creating vms and lxc



Explaining Steps within HA Quorum

1. **Node Failure Detection:** The HA system detects the failure of a node within the cluster.
2. **Quorum Recalculation:** The Quorum algorithm recalculates the minimum number of active nodes required for service availability based on the remaining nodes.
3. **Quorum Threshold Evaluation:** The system evaluates whether the remaining active nodes meet the new Quorum threshold for maintaining service availability.
4. **Automatic Failover Trigger:** If the remaining active nodes fall below the Quorum threshold, the HA system triggers automatic failover mechanisms.
5. **Resource Redistribution:** Resources and workloads from the failed node are redistributed and migrated to the remaining active nodes to ensure continued service availability.
6. **Service Restoration:** Once resources are successfully migrated, the HA system restores service availability and resumes normal operations on the surviving nodes.
7. **Node Recovery:** If the failed node becomes available again, it undergoes a recovery process to rejoin the cluster and resume its role in maintaining high availability.

Showing Quorum in Action

Migrating VMs & LXC to other Nodes

Migrating VMs and LXC from one node to another is lightning fast, thanks to NFS deployed using TRUE NAS.
Our cluster leverages shared storage to ensure minimal downtime and efficient migration of workloads, enabling us to maintain Vms Availability during node maintenance or upgrades with minimum downtime.

Minimized Downtime: Shared storage enables quick and seamless migration of VMs and LXC, minimizing disruptions and downtime.
Improved Resource Utilization: Migration optimizes resource usage by balancing workloads across the cluster.
Enhanced Flexibility: Proxmox migration capabilities allow dynamic resource scaling and performance optimization without service interruptions.

DEMO: Migrating vms over nodes

Task viewer: VM 103 - Migrate (pve-1 ---> pve-2)

Output | Status

Stop | Download

```
2024-04-22 00:04:17 starting VM 103 on remote node 'pve-2'
2024-04-22 00:04:20 start remote tunnel
2024-04-22 00:04:21 ssh tunnel ver 1
2024-04-22 00:04:21 starting online/live migration on unix:/run/qemu-server/103.migrate
2024-04-22 00:04:21 set migration capabilities
2024-04-22 00:04:21 migration downtime limit: 100 ms
2024-04-22 00:04:21 migration cachesize: 1.0 GiB
2024-04-22 00:04:21 set migration parameters
2024-04-22 00:04:21 start migrate command to unix:/run/qemu-server/103.migrate
2024-04-22 00:04:22 migration active, transferred 94.6 MiB of 8.0 GiB VM-state, 130.8 MiB/s
2024-04-22 00:04:23 migration active, transferred 206.4 MiB of 8.0 GiB VM-state, 403.7 MiB/s
2024-04-22 00:04:24 migration active, transferred 315.3 MiB of 8.0 GiB VM-state, 2.5 GiB/s
2024-04-22 00:04:25 migration active, transferred 422.9 MiB of 8.0 GiB VM-state, 135.4 MiB/s
2024-04-22 00:04:26 migration active, transferred 534.8 MiB of 8.0 GiB VM-state, 111.7 MiB/s
2024-04-22 00:04:27 migration active, transferred 647.1 MiB of 8.0 GiB VM-state, 113.7 MiB/s
2024-04-22 00:04:28 migration active, transferred 719.9 MiB of 8.0 GiB VM-state, 112.7 MiB/s
2024-04-22 00:04:29 migration active, transferred 832.1 MiB of 8.0 GiB VM-state, 112.5 MiB/s
2024-04-22 00:04:30 migration active, transferred 929.5 MiB of 8.0 GiB VM-state, 130.8 MiB/s
2024-04-22 00:04:31 migration active, transferred 1.0 GiB of 8.0 GiB VM-state, 120.5 MiB/s
2024-04-22 00:04:33 migration active, transferred 1.2 GiB of 8.0 GiB VM-state, 112.0 MiB/s
2024-04-22 00:04:34 migration active, transferred 1.3 GiB of 8.0 GiB VM-state, 113.1 MiB/s
2024-04-22 00:04:35 migration active, transferred 1.4 GiB of 8.0 GiB VM-state, 116.7 MiB/s
2024-04-22 00:04:35 average migration speed: 586.3 MiB/s - downtime 49 ms
2024-04-22 00:04:35 migration status: completed
2024-04-22 00:04:38 migration finished successfully (duration 00:00:21)
TASK OK
```

49 ms Downtime

Scalability of VMs and LXC:

Proxmox supports horizontal scaling of VMs and LXC instances by dynamically adjusting **CPU, GPU, memory, and storage resources.**
We can easily modify resource allocations to accommodate changing workload requirements, ensuring optimal performance and efficiency.

Mounting IO & PCI Devices:
Proxmox facilitates the mounting of IO devices like network adapters, storage controllers, and GPUs to VMs and LXC instances, enhancing functionality and performance.
This flexibility enables leveraging specialized hardware resources, tailoring virtualized environments to specific application needs.

Scalability demo

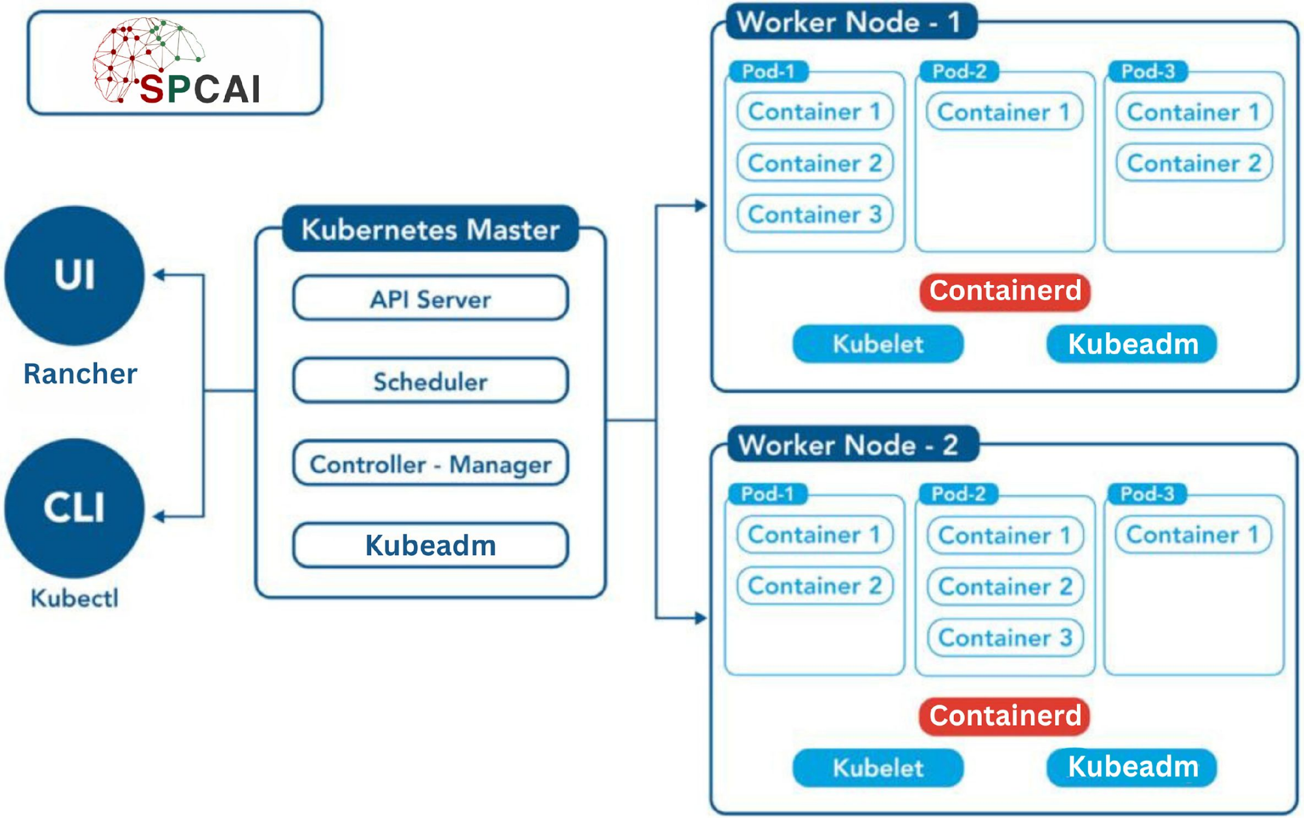
KUBERNETES | Container Orchestration

Introduction to Kubernetes: Container Orchestration Platform

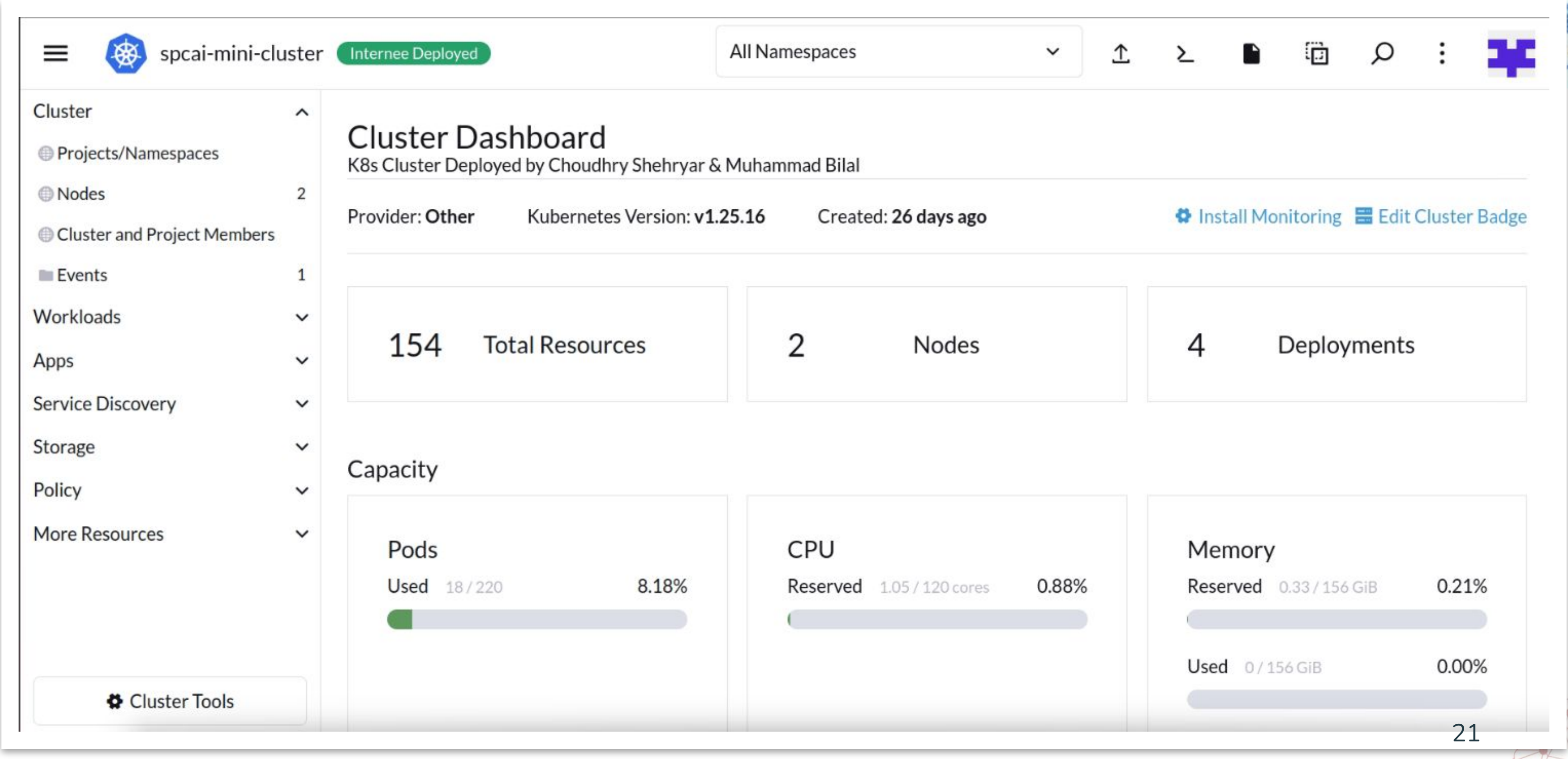
Kubernetes is an open-source container orchestration platform developed by Google. It automates the deployment, scaling, and management of containerized applications.

- Efficient container management
- Seamless Scalability on demand
- Service discovery and load balancing
- Self-healing: Restarts containers that fail and replaces them with healthy ones.
- Declarative configuration: Kubernetes uses YAML files for configuration
- Resource utilization: It optimizes resource allocation, ensuring efficient use of CPU and memory.

Architecture of kubernetes cluster:



Rancher Dashboard



Exploring Rancher Dashboard: Simplifying Kubernetes Management

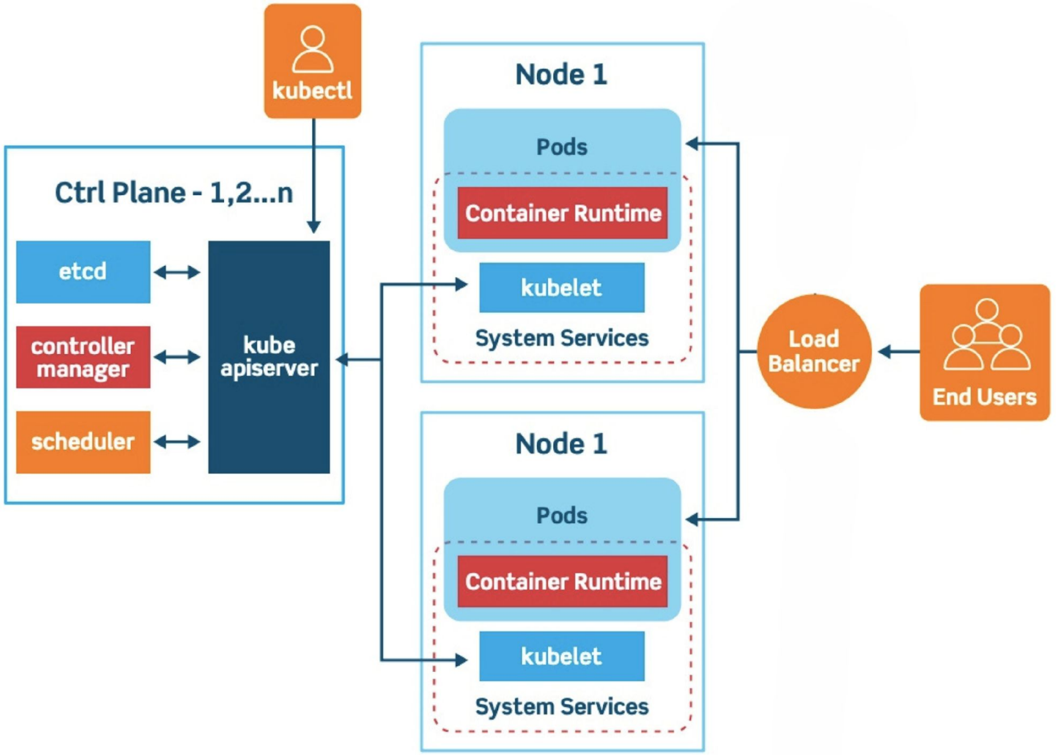
Rancher Dashboard is serving as a central hub for managing and analyzing our Kubernetes cluster.

Key Functions:

Cluster Management: Rancher Dashboard provides a user-friendly interface for managing all aspects of our Kubernetes cluster, including nodes, workloads, and configurations.

Monitoring and Analysis, Resource Allocation, Deployment and Orchestration: We utilize Rancher Dashboard for deploying and orchestrating applications within our Kubernetes cluster

Kubernetes Pods:



Pod Deployment

Deploying Pods through Rancher Dashboard:
Offers a user-friendly graphical interface for visual deployment and management of Pods, suitable for users who prefer a point-and-click approach.

Deploying Pods through kubectl Command-line Tool:
Provides flexibility and automation capabilities for deploying Pods programmatically, ideal for advanced users and scripting deployment workflows.

Pods: Use-Cases and Benefits

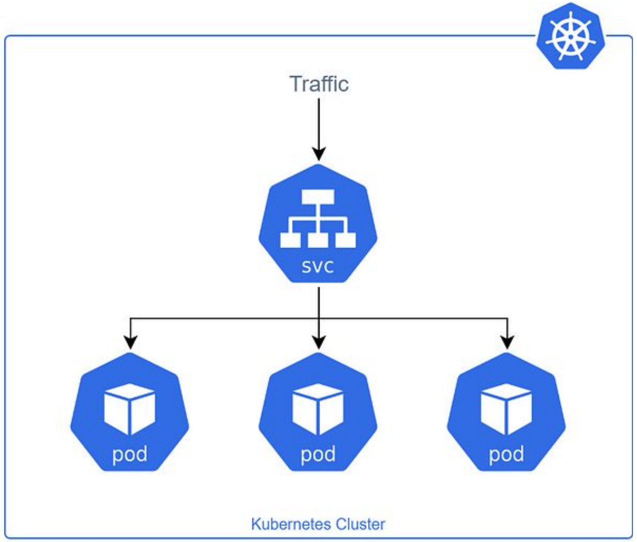
Key Use Cases:

- **Microservices Architecture:** Pods break down applications into manageable components for microservices-based deployments.
- **Batch Processing Jobs:** Kubernetes Pods execute batch processing tasks like data analytics and report generation.
- **Stateful Applications:** Pods provide persistent storage for stateful applications via external storage volumes.
- **High-Performance Computing (HPC):** Pods support high-performance computing workloads such as scientific simulations.

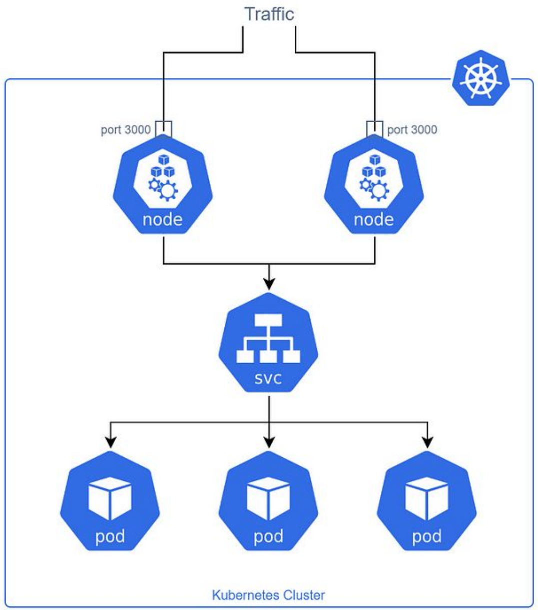
Benefits:

- **Scalability:** Pods enable horizontal scaling of applications to meet fluctuating demand
- **Flexibility:** With Pods, we can deploy diverse types of applications, from stateless web services to data-intensive tasks
- **Resilience:** Kubernetes manages Pod lifecycle, ensuring fault tolerance and automatic recovery in case of Pod failures

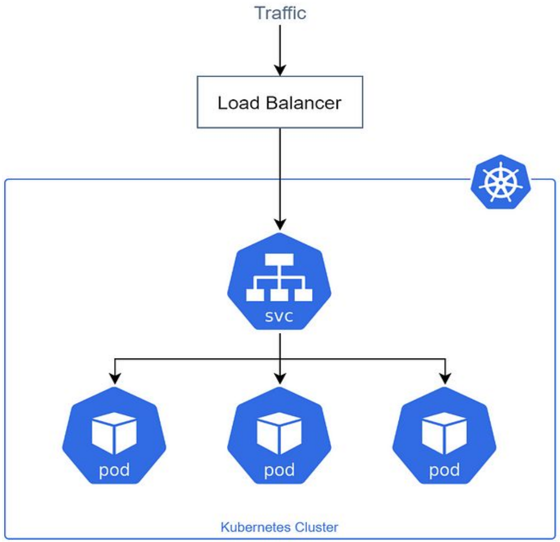
Services:



ClusterIP



NodePort



LoadBalancer

Service Types and Use Cases:

ClusterIP:

Use Case: Internal microservices communication, databases, and backend APIs.

Description: Exposes Pods internally for secure microservices communication.

NodePort:

Use Case: Testing and development environments, external access for debugging or validation.

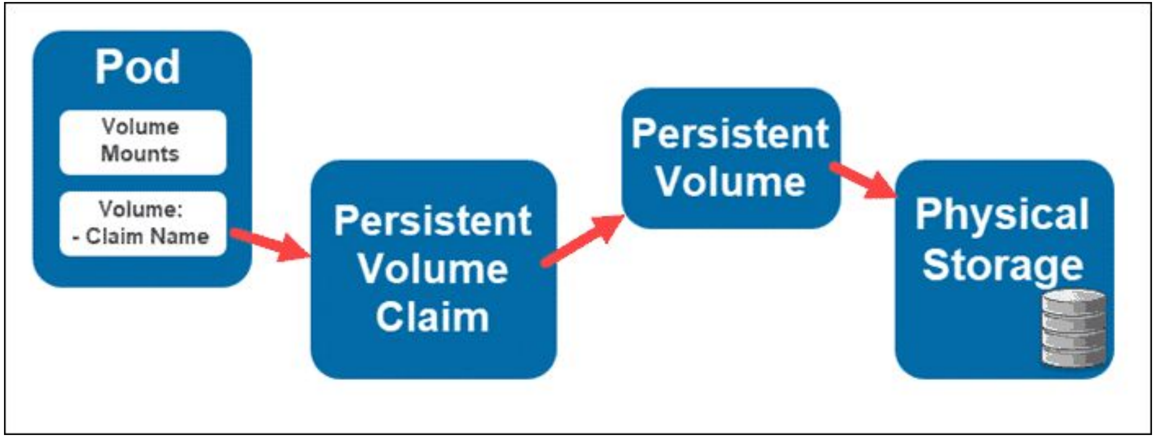
Description: Exposes Pods on static ports across all nodes for external access.

Load Balancer:

Use Case: Production applications requiring external access.

Description: Distributes external traffic across Pods for high availability.

Persistent Volumes for Pods



Databases: PVs are commonly used to provide persistent storage for databases, ensuring data persistence and reliability across Pod restarts or migrations.

File Storage: For applications requiring shared file storage, PVs with RWX access mode can be used to provide shared access to files across multiple Pods.

DEMO: Deploying **JupyterHub**

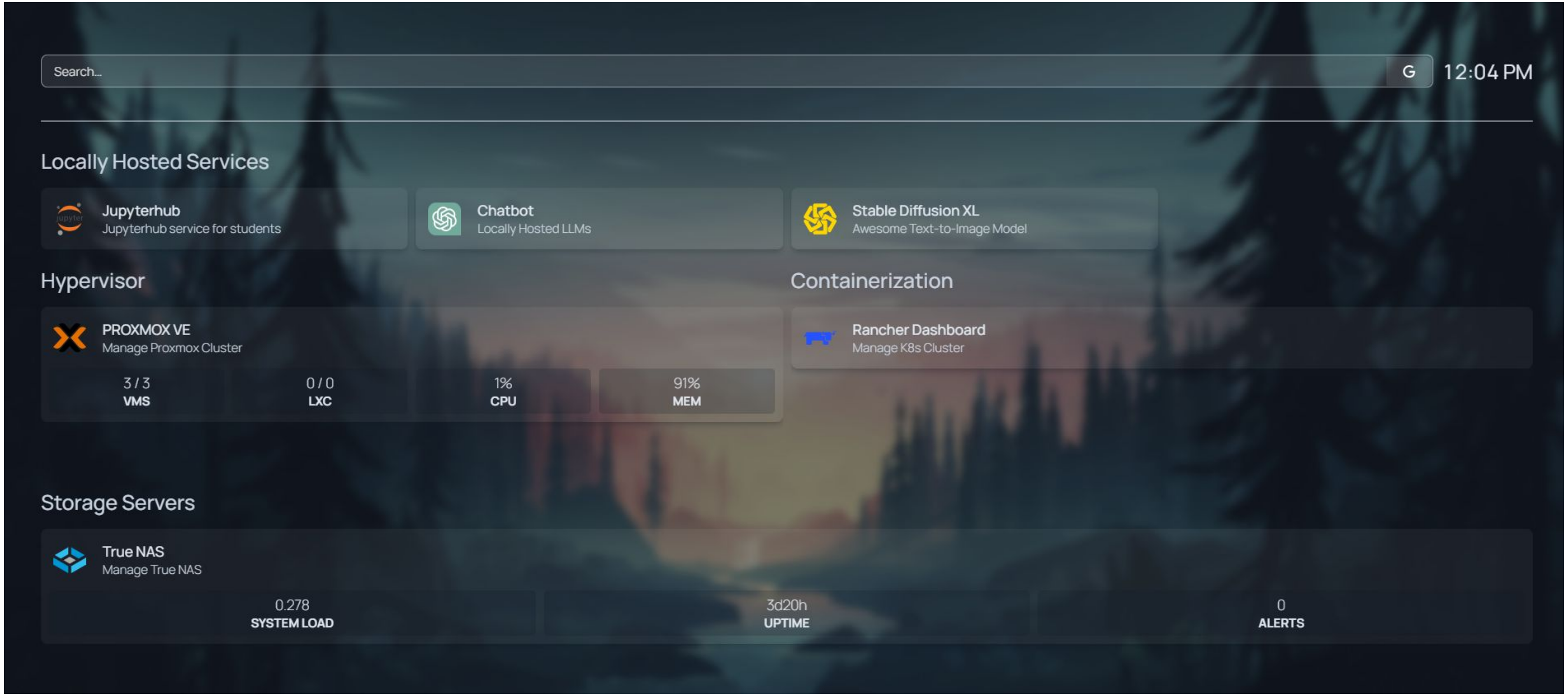
DEMO:
Deploying a Static Website

Applications Running on Cluster

Overview of Our Local Deployments:

- Multimodal Model LLAVA 1.6 **(Supports vision and text)**
- LLMs **(llama-2, llama-3, Mistral 7B, Codegemma)**
- Text-to-image model **(Stable Diffusion XL)**
- JupyterHub for Students

Central dashboard for Services



http://cluster.paf-iast.edu.pk

The login credentials of the subdomain are:
Login: SpcaiComputeServices
Password: h\OuXIn$&DP`_XK1h%O:qU#i

Conclusion

- Recap of Key Points
- Importance of Proxmox and Kubernetes in Efficient Resource Utilization
- Handling different nature of tasks
- Flexible IDE for handling both platform from a single domain

Q&A

Open Floor for Questions and Answers