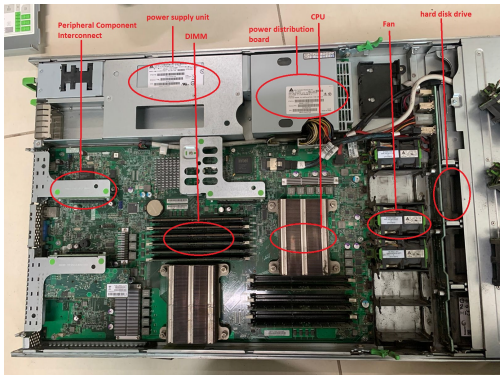
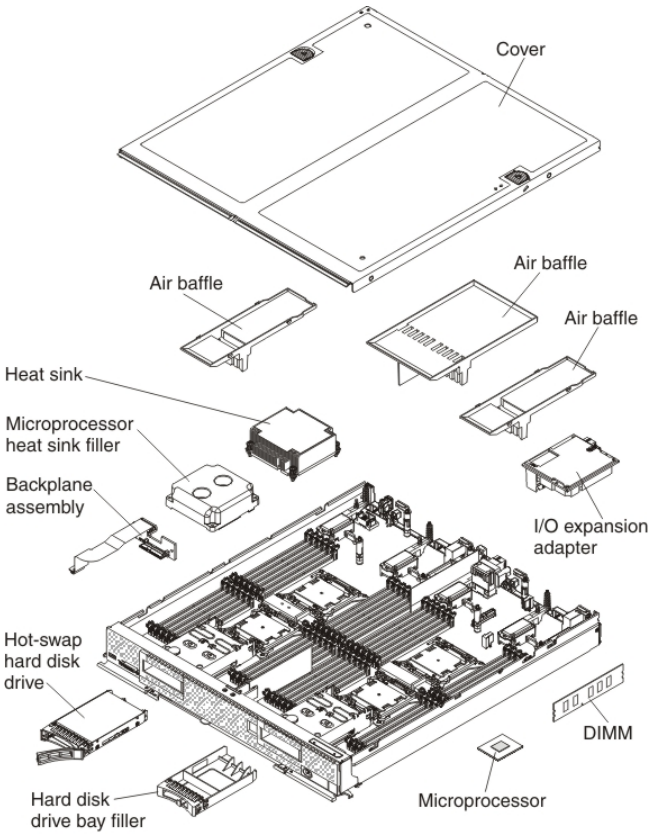
Implementation of private cloud using old HPC servers

Task-01

- **S:** Examine the components of a compute node's
- **T:** Open the node and physically determine each component present.
- **A:** Pushing the buttons and removing the lid as directed on the node allowed you to open it.
- **R:** All of the components required for the instruction to be executed were visible, and they were photographed and labeled as well.

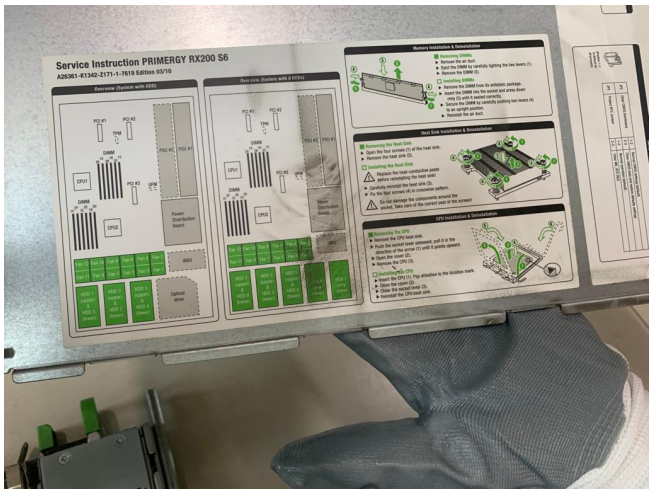


- A compute node is **a system running Oracle Linux using KVM**. A compute node runs the bare minimum of services



to manage virtual machine instances.

- The major components of the compute node include **field replaceable units (FRUs), customer replaceable units (CRUs)**, and optional devices. The side illustration shows the major components of the compute node.



- Here, a **Fujitsu Primergy RX200 S6 Server** serves as the compute node.
- The picture shows the instructions on how to open each component and also a simple diagram of the basic components of the compute node.

Task-02

- S: Install OS on compute nodes.
- T: Boot and install Ubuntu on the USB flash drive before installing the operating system on the compute node and then installing it on the compute nodes.

- A: Booting and installation were performed with balenaEtcher (commonly referred to and formerly known as Etcher, which is **a free and open-source utility used for writing image files such as .iso and .img files, as well as zipped folders onto storage media to create live SD cards and USB flash drives**). Then opened the boot-device selection menu and did select the option that boots from the USB flash drive and hence started the setup.
- R: Tried installing the OS on different nodes but due to **DIMM failure(**A memory error is **an event that leads to the logical state of one or multiple bits being read differently from how they were last written**. For example, If 1 was written in a memory cell and while reading the same memory cell, it returns 0) the installation process was declined.

Compute NODE	CPU status	MEMORY status	PCI status	OS installation
NODE- 26	OK	FAILED- DIMM-1C ENABLED- 8 EMPTY- 3	ENABLED- 1 EMPTY- 2	DECLINED
NODE- 17	OK	FAILED- DIMM-1F ENABLED- 6 EMPTY- 5	ENABLED- 1 EMPTY- 2	DECLINED
NODE- 18	OK	FAILED- DIMM-1D ENABLED- 4 EMPTY- 7	ENABLED- 1 EMPTY- 2	DECLINED
NODE- 19	OK	FAILED- 0 ENABLED- 9 EMPTY- 3	ENABLED- 1 EMPTY- 2	DECLINED

Memory- 152115 MB

- Then switched from PRIMERGY RX200 S6 to RX600 S6 where Ubuntu was successfully installed and set up.

Task-03

- **S:** A **hardware** solution for controlling and managing your servers.
- **T:** Configuring IPMI support
- **A:** The following are the steps required for IPMI configuration.

  ▼ Installing the **IPMItool** command

- To enable one of the drivers that use IPMI protocol for power and management actions, the **ipmitool** command must be present on the service node(s).
- Please refer to the IPMI driver for information on how to configure and use IPMItool-based drivers.

▼ Configuring Hardware

IPMI is a relatively old protocol and may require additional setup on the hardware side that the Bare Metal service cannot do automatically:

1. Make sure IPMI is enabled and the account you use has the permissions to change power and boot devices.
2. Make sure the boot mode corresponds to the expected boot mode on the node.

▼ Validating and Troubleshooting

If the above command doesn't return the power status of the bare metal server, check that

- The IPMI controller on your bare metal server is turned on.
- The IPMI controller credentials and IP address passed in the command are correct.
- The conductor node has a route to the IPMI controller. This can be checked by just pinging the IPMI controller IP from the conductor node.

- **R:** Configure the IPMI of a node and the following are the **takeaways** from this task.
  1. The Intelligent Platform Management Interface (IPMI) is **a specification for remote server management**.
  2. It is a way to manage a node by using a network connection to the hardware rather than to an operating system or login shell.

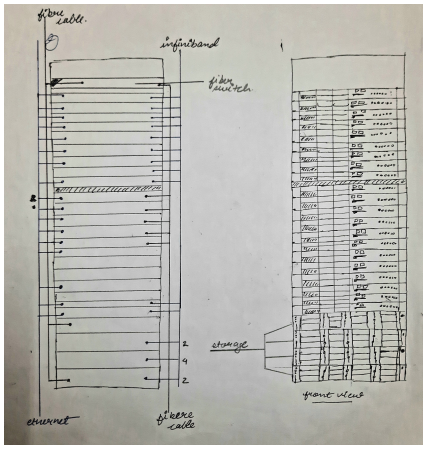
Task-04

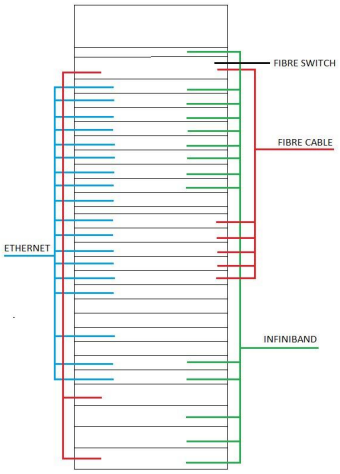
- **S: How to use MAAS with Juju.**
- **T:** Study materials related to MAAS and Juju

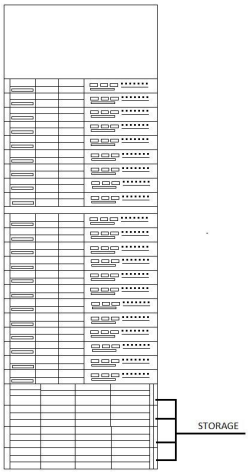
- **A:** Studied in detail about MAAS and Juju and how to cluster MAAS with Juju, and the following are the steps involved in the process.
    1. Create your MAAS network in VMM.
    2. Create a MAAS Controller
    3. Configure the MAAS Controller
    4. Configure the MAAS Controller network
    5. Add node(s) to MAAS
    6. Set up Juju
- **R:** The following are the **takeaways** from this task.
    1. MAAS provisions hardware using existing and well-established protocols such as IPMI. Juju makes it easy to visualize, design, deploy, and scale application infrastructures across multiple environments, such as bare-metal hardware or as workloads on top of cloud infrastructure ecosystems like OpenStack, HP Cloud, Microsoft Azure, and Amazon AWS.
    2. Juju, coupled with MAAS, enables the deployment of services on bare-metal hardware.
    3. Most of the functionality of MAAS is contained in a series of controllers. There are two basic types: **a region controller and one or more rack controllers**. The region controller deals with operator requests while the rack controller(s) provides high-bandwidth services to the individual machines.

Task-05

- **S:** Identify the connections between storage, nodes, and fiber switches in the server sack.
- **T:** Draw a labeled diagram of the server sack's front and back views.
- **A:** Observed and identified the connection cables and their connections and drew the following diagram based on that.







- **R:** There are three types of connection cables:
  1. **ETHERNET:** Ethernet enables devices to communicate with each other via a protocol, which is a set of rules or common network language. Ethernet cables are connected to the nodes.
  2. **Fiber Cable:** Fiber cables are used for **long-distance and high-performance data networking**. In the server sack, first fiber cable connections are made between storage and the fiber switch, then cables are connected to nodes.
  3. **INFINIBAND:** InfiniBand is **a channel-based fabric that facilitates high-speed communications between interconnected nodes**. In the server sack, InfiniBand connections are made between the nodes, but according to the new configuration of our server sack, InfiniBand is not required as fiber cables facilitate its function.

Task-06

- S: There are different types of nodes in the rack such as **storage nodes and compute nodes** and they have different configurations.
- T: To identify all types of nodes and label them on the rack for easy identification.
- A: With the help of **JUJU identify** all the different types of nodes in the rack.
- R:
  - Node identification: Labeled rack:

NODE 40	storage, compute

NODE 41	compute
NODE 42	storage
NODE 43	storage, compute
NODE 44	-
NODE 45	-
-	-
NODE 29	-
MAAS enlisting node	
NODE 28	-
MAAS	
JUJU	

Task-07

- S: The Image service (glance) enables users to discover, register, and retrieve virtual machine images.
- T: Upload images on OpenStack.
- A: Download images from the internet in iso format then convert those images to qcow2 format and upload them on OpenStack.
    - To convert windows images from iso format to qcow2 create VM inside VirtualBox, then install windows and drivers and use Cloud-Init for the conversion.
    - For the other images, use qemu-img command in the terminal for the conversion.
- R: The uploaded images on OpenStack are:
    - RedHat-7.9, 8.2, 8.4, 9.0
    - Zorin-OS-16.1
    - SUSE Linux-15
    - Ubuntu-22.04
    - Fedora-36-1.5

    - Windows desktop images
    - Windows Server images

Task-08

- S: OpenStack virtual machines are called **instances**, mostly because they are instances of an image that is created upon request and that is configured when launched.
- T: To create instances on OpenStack for all the different images and different sizes.
- A: Launch the uploaded images on Openstack and created instances for the same.
- R: Upto 10 instances can be made on OpenStack at a time.