Cloud Computing

CSI ZG527 / SS ZG527 / SE ZG527

BITS Pilani

Module 3
Infrastructure as a Service

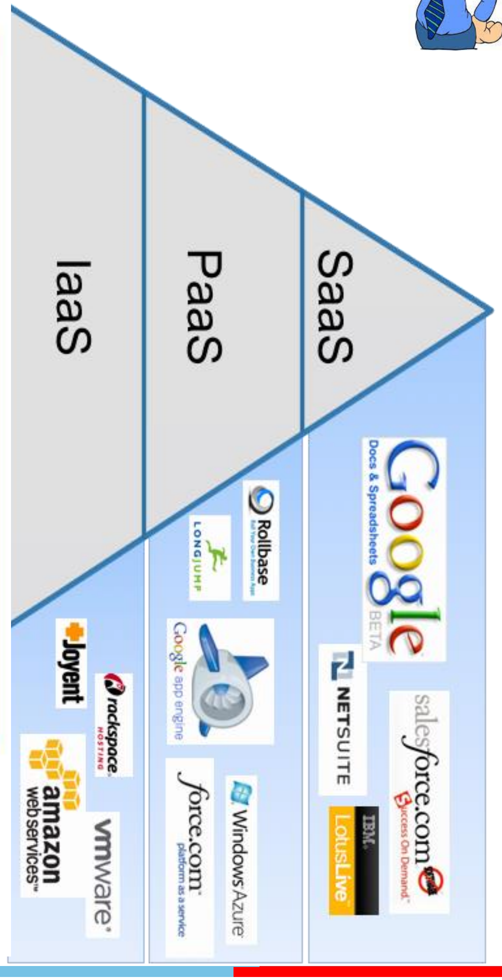
heard of 3 models of Cloud Computing?

Yes, Yes, IaaS, PaaS and SaaS

OnPremises	IaaS	PaaS	SaaS
🟧 Applications	🟧 Applications	🟧 Applications	🟦 Applications
🟧 Data	🟧 Data	🟧 Data	🟦 Data
🟧 Middleware	🟧 Middleware	🟦 Middleware	🟦 Middleware
🟧 Operating System	🟧 Operating System	🟦 Operating System	🟦 Operating System
🟧 Virtualization	🟦 Virtualization	🟦 Virtualization	🟦 Virtualization
🟧 Server	🟦 Server	🟦 Server	🟦 Server
🟧 Storage	🟦 Storage	🟦 Storage	🟦 Storage
🟧 Network	🟦 Network	🟦 Network	🟦 Network

🟧 Own Responsibility

🟦 Cloud Provider Responsibility



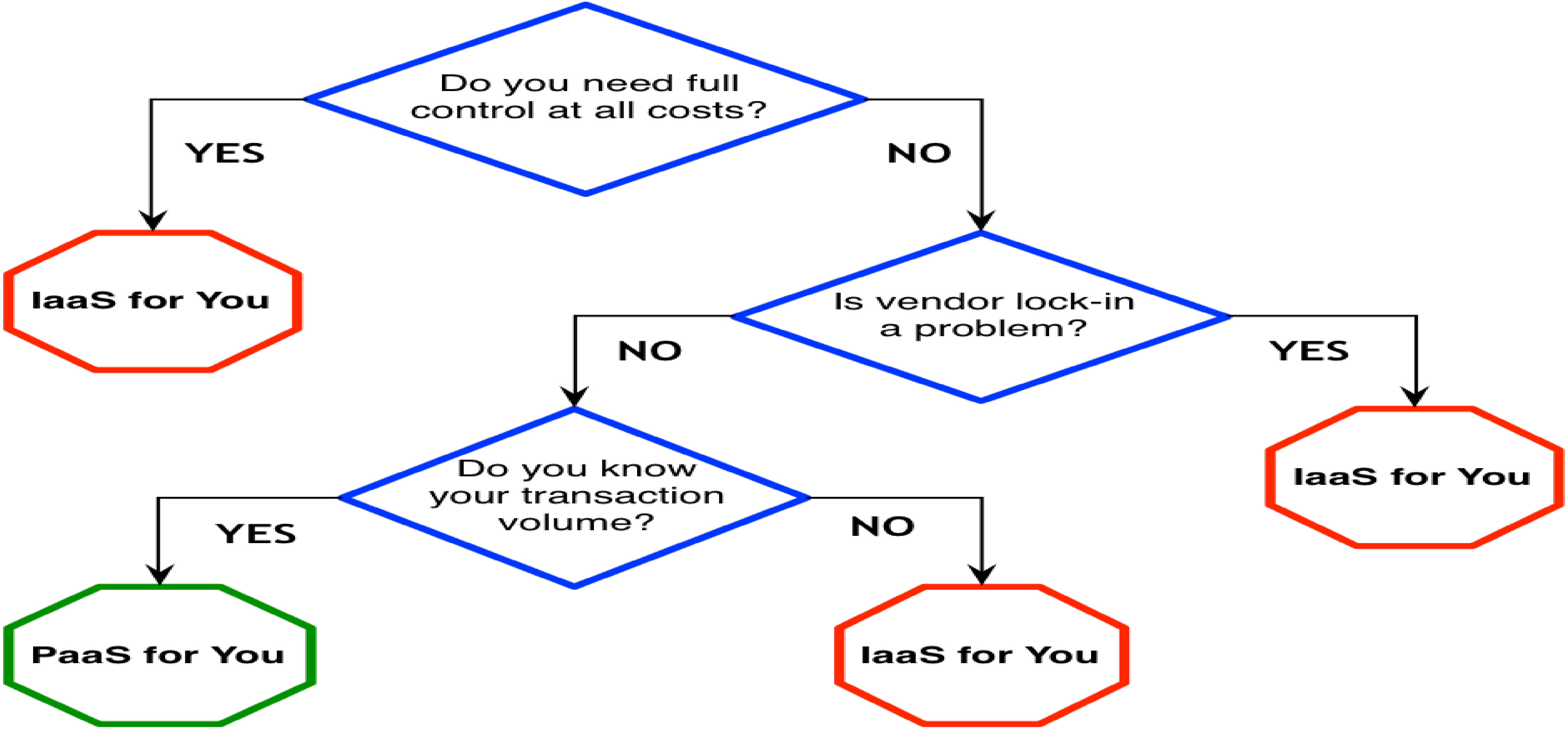
Infrastructure as a Service (IaaS)

- Under the IaaS cloud computing model, **cloud service providers make computing and storage resources (such as servers and storage) available as a service**.
- This offers maximum flexibility for users to work with the cloud infrastructure, wherein exactly how the virtual computing and storage resources are used is left to the cloud user.

Features of IaaS

- Geographic Presence
- User Interfaces and Access to Servers
- Advance Reservation of Capacity
- Automatic Scaling and Load Balancing
- Automatic Scaling and Load Balancing
- Service-Level Agreement (SLA)
- Hypervisor and Operating System Choice

IaaS or PaaS Decision Tree


Do you need full control at all costs?
YES
NO
IaaS for You
Is vendor lock-in a problem?
NO
YES
IaaS for You
Do you know your transaction volume?
YES
NO
PaaS for You
IaaS for You

The value of IaaS

For businesses, the greatest value of IaaS is through a concept known as ***cloudbursting***—the process of off-loading tasks to the cloud during times when the most compute resources are needed.

To take advantage of IaaS in this capacity, IT departments must be able to build and implement the software that handles the ability to re-allocate processes to an IaaS cloud.

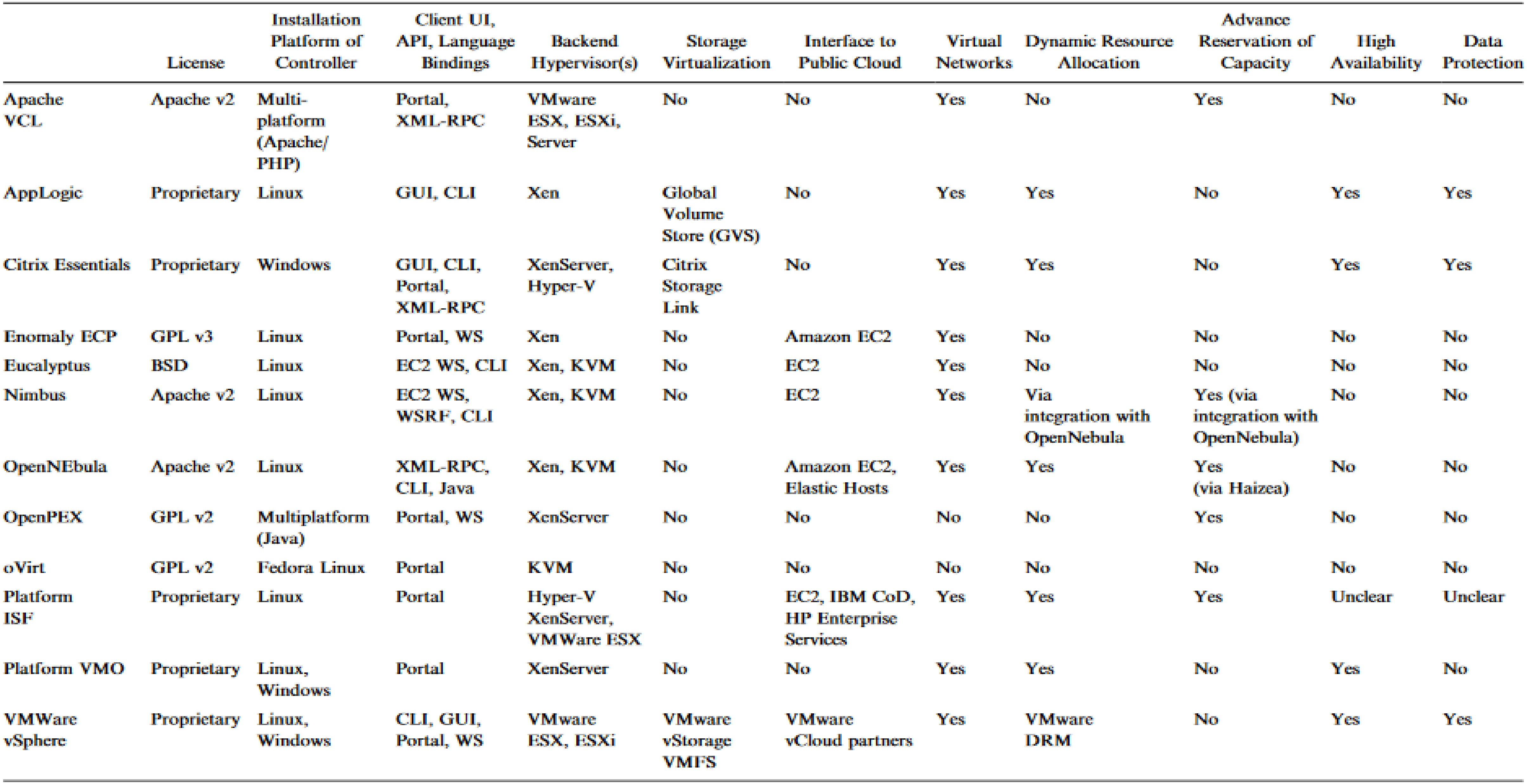
There are **four important considerations** to build and implement software that can manage such reallocation processes.

4 considerations:

- Developing for a specific vendor's proprietary IaaS could prove to be a costly mistake.
- The complexity of well-written resource allocation software is significant and do not come cheap
- What will you be sending off to be processed in the cloud? Sending data such as personal identities, financial information, and health care data put an organization's compliance at risk
- Understand the dangers of shipping off processes that are critical to the day-to-day operation of the business.

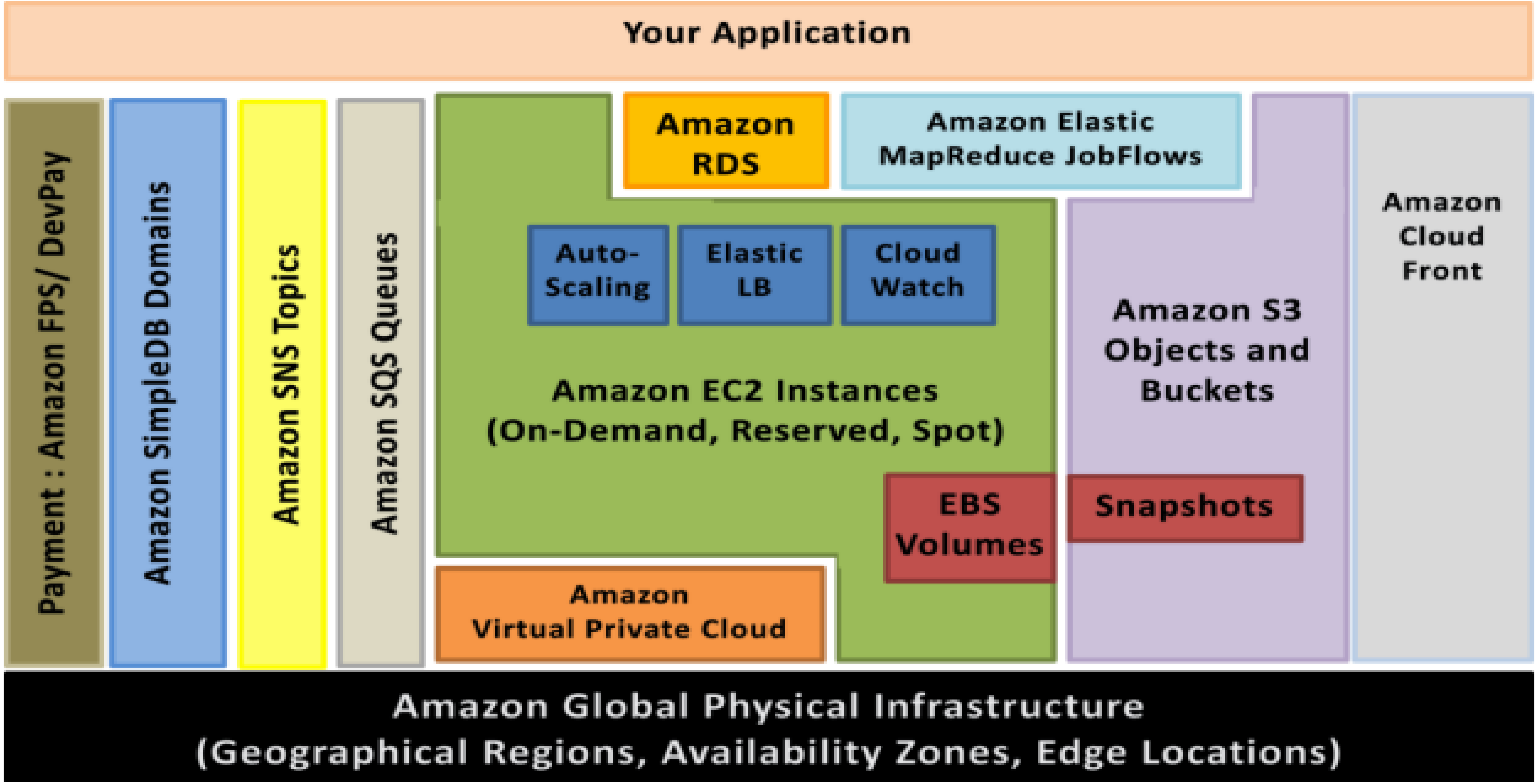
- http://www.ibm.com/developerworks/cloud/library/cl-cloudservices1iaas/

Feature Comparison of Virtual Infrastructure Managers



	License	Installation Platform of Controller	Client UI, API, Language Bindings	Backend Hypervisor(s)	Storage Virtualization	Interface to Public Cloud	Virtual Networks	Dynamic Resource Allocation	Advance Reservation of Capacity	High Availability	Data Protection
Apache VCL	Apache v2	Multi-platform (Apache/ PHP)	Portal, XML-RPC	VMware ESX, ESXi, Server	No	No	Yes	No	Yes	No	No
AppLogic	Proprietary	Linux	GUI, CLI	Xen	Global Volume Store (GVS)	No	Yes	Yes	No	Yes	Yes
Citrix Essentials	Proprietary	Windows	GUI, CLI, Portal, XML-RPC	XenServer, Hyper-V	Citrix Storage Link	No	Yes	Yes	No	Yes	Yes
Enomaly ECP	GPL v3	Linux	Portal, WS	Xen	No	Amazon EC2	Yes	No	No	No	No
Eucalyptus	BSD	Linux	EC2 WS, CLI	Xen, KVM	No	EC2	Yes	No	No	No	No
Nimbus	Apache v2	Linux	EC2 WS, WSRF, CLI	Xen, KVM	No	EC2	Yes	Via integration with OpenNebula	Yes (via integration with OpenNebula)	No	No
OpenNEbula	Apache v2	Linux	XML-RPC, CLI, Java	Xen, KVM	No	Amazon EC2, Elastic Hosts	Yes	Yes	Yes (via Haizea)	No	No
OpenPEX	GPL v2	Multiplatform (Java)	Portal, WS	XenServer	No	No	No	No	Yes	No	No
oVirt	GPL v2	Fedora Linux	Portal	KVM	No	No	No	No	No	No	No
Platform ISF	Proprietary	Linux	Portal	Hyper-V XenServer, VMWare ESX	No	EC2, IBM CoD, HP Enterprise Services	Yes	Yes	Yes	Unclear	Unclear
Platform VMO	Proprietary	Linux, Windows	Portal	XenServer	No	No	Yes	Yes	No	Yes	No
VMWare vSphere	Proprietary	Linux, Windows	CLI, GUI, Portal, WS	VMware ESX, ESXi	VMware vStorage VMFS	VMware vCloud partners	Yes	VMware DRM	No	Yes	Yes

AWS infrastructure services


Your Application
Payment : Amazon FPS/ DevPay
Amazon SimpleDB Domains
Amazon SNS Topics
Amazon SQS Queues
Amazon RDS
Amazon Elastic MapReduce JobFlows
Auto-Scaling
Elastic LB
Cloud Watch
Amazon EC2 Instances (On-Demand, Reserved, Spot)
EBS Volumes
Snapshots
Amazon S3 Objects and Buckets
Amazon Cloud Front
Amazon Virtual Private Cloud
Amazon Global Physical Infrastructure (Geographical Regions, Availability Zones, Edge Locations)

Amazon Web Services (AWS)

IaaS that AWS offer:

- Storage as a Service
  - Amazon Simple Storage Service
  - Amazon SimpleDB
  - Amazon Relational Database Service
- Compute as a Service
  - Amazon Elastic Compute Cloud (EC2)

Different types of Data

- Enterprises have varied requirements for data, **including structured data in relational databases** that power an e-commerce business, or **documents that capture unstructured data** about business processes, plans and visions.
- Enterprises may also need to store objects on behalf of their customers, like an **online photo album or a collaborative document** editing platform.
- Further, some of the data may be **confidential and must be protected**, while

others data should be easily shareable.

- In all cases, **business critical data should be secure and available** on demand in the face of hardware and software failures, network partitions and inevitable user errors.

Storage as a Service: Amazon Storage Services

- **Simple Storage Service** (**S3**): An object store
- **SimpleDB**: A Key-value store
- **Relational Database Service (RDS):** MySQL instance

and so on.

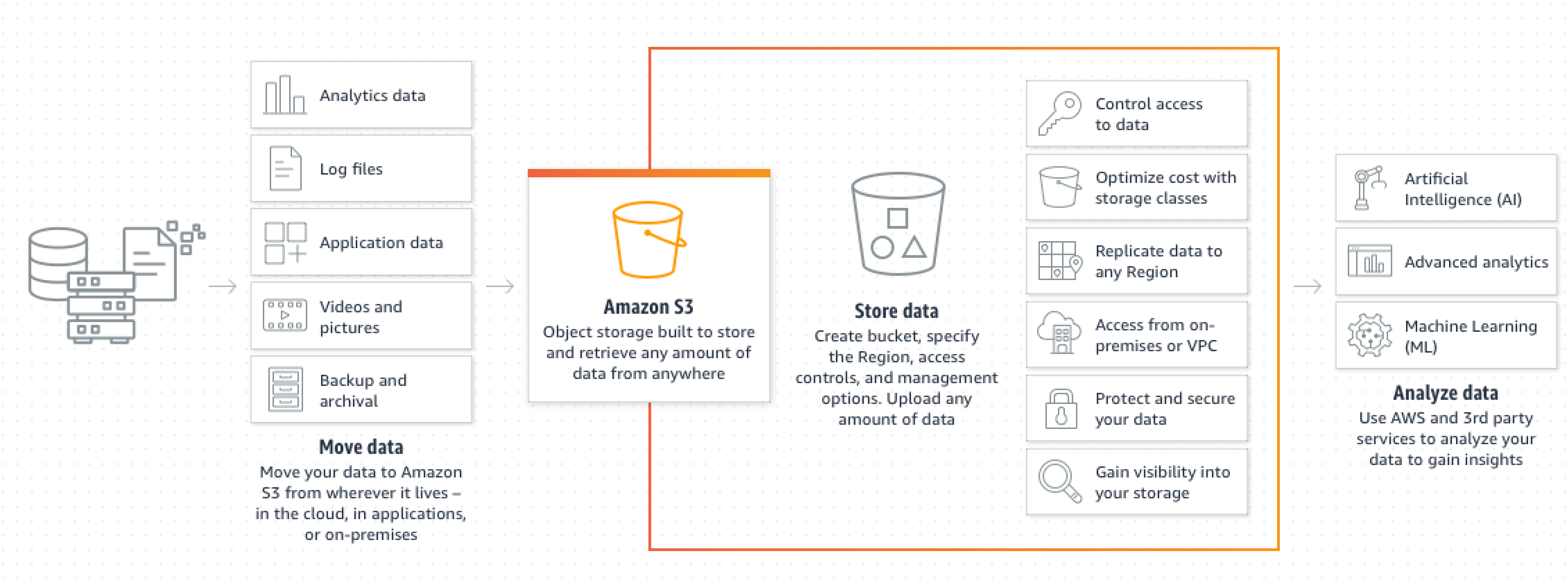
Amazon's Simple Storage Server (S3)

- Amazon S3 is a **highly reliable**, **highly available**, **scalable** and **fast storage** in the cloud for storing and retrieving large amounts of data just through simple web services.
- S3 is a storage service, several S3 browsers exist that allow users to explore their S3 account as if it were a directory (or a folder). There are also file system implementations that let users treat their S3 account as just another directory on their local disk.

S3 Access Methods:

- AWS Console
- Amazon's RESTful API
- SDKs for Ruby and other languages

Amazon S3: How it works?.



Analytics data

Log files

Application data

Videos and pictures

Backup and archival

Move data

Move your data to Amazon S3 from wherever it lives – in the cloud, in applications, or on-premises

Amazon S3

Object storage built to store and retrieve any amount of data from anywhere

Store data

Create bucket, specify the Region, access controls, and management options. Upload any amount of data

Control access to data

Optimize cost with storage classes

Replicate data to any Region

Access from on-premises or VPC

Protect and secure your data

Gain visibility into your storage

Artificial Intelligence (AI)

Advanced analytics

Machine Learning (ML)

Analyze data

Use AWS and 3rd party services to analyze your data to gain insights

Ref: https://aws.amazon.com/s3/

Organizing Data In S3: Buckets, Objects and Keys

- **Files** are called **objects** in S3.
- Objects are referred to with **keys** – basically an optional **directory path name** followed by the **name** of the object.
- Objects **in S3 are replicated across multiple geographic locations** to make it resilient to several types of failures (however, consistency across replicas is not guaranteed).
- **If object versioning is enabled**, recovery from inadvertent deletions and modifications is possible.
- S3 objects can be up to **5 Terabytes in size** and there are no limits on the number of objects that can be stored.
- All objects in S3 **must be stored in a bucket**.

Organizing Data In S3: Buckets, Objects and Keys (Cont…)

- Buckets provide a way to keep related objects in one place and separate them from others. **There can be up to 100 buckets** per account and an unlimited number of objects in a bucket.

S3 objects

- Each object has a **key**, which can be used as the path to the resource in an HTTP URL.
- For example, if the bucket is named **johndoe** and the key to an object is

resume.doc, then its HTTP URL is **http://s3.amazonaws.com/johndoe/resume.doc**

or alternatively, http://johndoe.s3.amazonaws.com/resume.doc

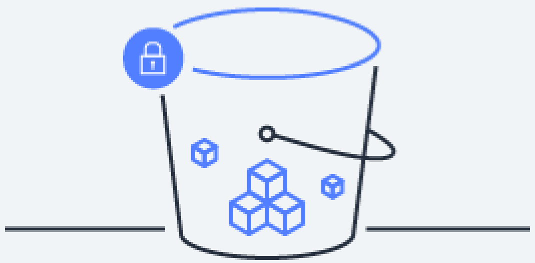
- URL needs **authentication parameters**; S3 objects **are private by default**

and requests should carry authentication parameters that prove the requester has rights to access the object, unless the object has “Public” permissions.

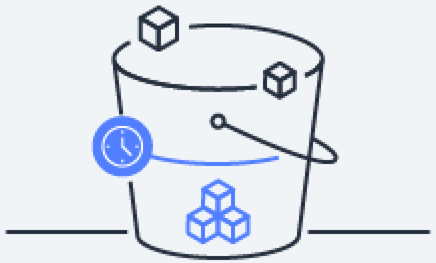
Note: The bucket namespace is shared; i.e., it is not possible to create a bucket with a name that has already been used by another S3 user.

Amazon S3 security and access management

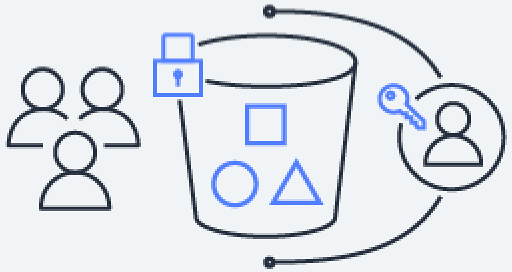
Block Public Access



Object Lock

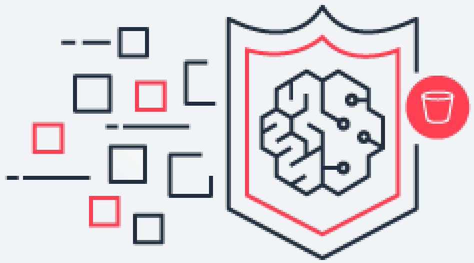


Object Ownership



Identity and Access Management

Amazon Macie



Encryption

Amazon S3 security and access management

Large Objects and Multi-part Uploads on S3

- The object size limit for S3 is 5 terabytes.
- If this limit is not sufficient, the **object then can be stored in smaller chunks** with the **splitting and re-composition** being managed in the application, using the data.

Note: Uploading large objects on Amazon S3 will still take some time even though it has high aggregate bandwidth available. Additionally, if an upload fails, the entire object needs to be uploaded again.

- Multi-part upload solves both the above problems elegantly.
- S3 provides APIs that allow the developer to write a program that splits a large object into several parts and uploads each part independently.
- These uploads can be parallelized for greater speed to maximize the network utilization.
- If a part fails to upload, only that part needs to be re-tried.

Simple use-case with S3 (Uploading photos)

1. Sign up for S3 at http://aws.amazon.com/s3/. While signing up, **obtain the AWS Access Key and the AWS Secret Key**. These are similar to userid and password that is used to authenticate all transactions with Amazon Web Services (not just S3).
2. Sign in to the AWS Management Console for S3 at the below URL https://console.aws.amazon.com/s3/home
3. Create a **bucket** giving a name and geographical location where it can be stored. In S3 all files (called objects) are stored in a bucket, which represents a collection of related objects.
4. Click the Upload button and follow the instructions to upload files.
5. The photos or other files are now safely backed up to S3 and available for sharing with a URL if the right permissions are provided.

Note: From a developer perspective, this can also be accomplished programmatically.

Amazon Simple Database Service (SimpleDB)

- SimpleDB (SDB) provides a simple data store interface in the form of a key-value store.
- It allows storage and retrieval of a set of attributes based on a key.
- It is a highly available, flexible, and scalable non-relational data store that offloads the work of database administration. It provides the core database functions of data indexing and querying in the cloud.
- It provides a simple web services interface to create and store multiple data sets, query your data easily, and return the results.

Data Organization and Access in SimpleDB

- Data is organized into domains.
- Each item in a domain has a unique key that must be provided during creation.
- Each item can have up to 256 attributes, which are name-value pairs.

SDB provides a query language that is analogous to SQL, although there are methods to fetch a single item.

- Queries take advantage of the fact that SDB automatically indexes all attributes.

Availability and Administration in SimpleDB

SDB has a number of features to increase **availability and reliability.**

- Data stored in SDB is automatically replicated across different geographies for high availability.
- It also automatically adds **compute resources in proportion to the request rate** and **automatically indexes all fields** in the dataset for efficient access.
- SDB **is schema-less**; i.e., fields can be added to the dataset as the need arises.

Amazon **Relational Database Service (RDS)**

- RDS provides a traditional database abstraction in the cloud, specifically a MySQL instance in the cloud.
- An RDS instance can be created using the RDS tab in the AWS Management Console as shown in the next slide.
- AWS performs many of the administrative tasks associated with maintaining

a database for the user.

- The database is backed up at configurable intervals, which can be as frequent as 5 minutes.
- The backup data are retained for a configurable period of time which can be up to 8 days. Amazon also provides the capability to snapshot the database as needed.

COMPUTE AS A SERVICE: AMAZON ELASTIC COMPUTE CLOUD (EC2)

Compute as a Service:

- Here, computing resources are offered as a service to the users.
- It should be **possible to associate storage with the computing service** (so that the results of the computation can be made persistent).
- **Virtual networking** is needed as well, so that it is possible to communicate with the computing instance.

All these together make up Infrastructure as a Service.

Note: Amazon’s Elastic Compute Cloud (EC2) is one of the popular Compute as a Service offerings

EC2 Computational Resources

Computing Resources:

- The computing resources available on EC2, referred to as **EC2 instances**, consist of combinations of **computing power, together with other resources such as memory**.
- Amazon measures the **computing power** of an EC2 instance in **terms of EC2 Compute Units**.
- An EC2 Compute Unit (CU) is a standard **measure of computing power** in the same way that bytes are a standard measure of storage.
- One EC2 CU provides the same amount of computing power as a 1.0–1.2 GHz Opteron or Xeon processor in 2007

For example, if a developer requests a computing resource of 1 EC2 CU, and the resource is allocated on a2.4 GHz processor, they may get 50% of the CPU. This allows developers to request standard amounts of CPU power regardless of the physical hardware.

Instance Types in EC2

A developer can request a computing resource of one of the instance types shown in the table.

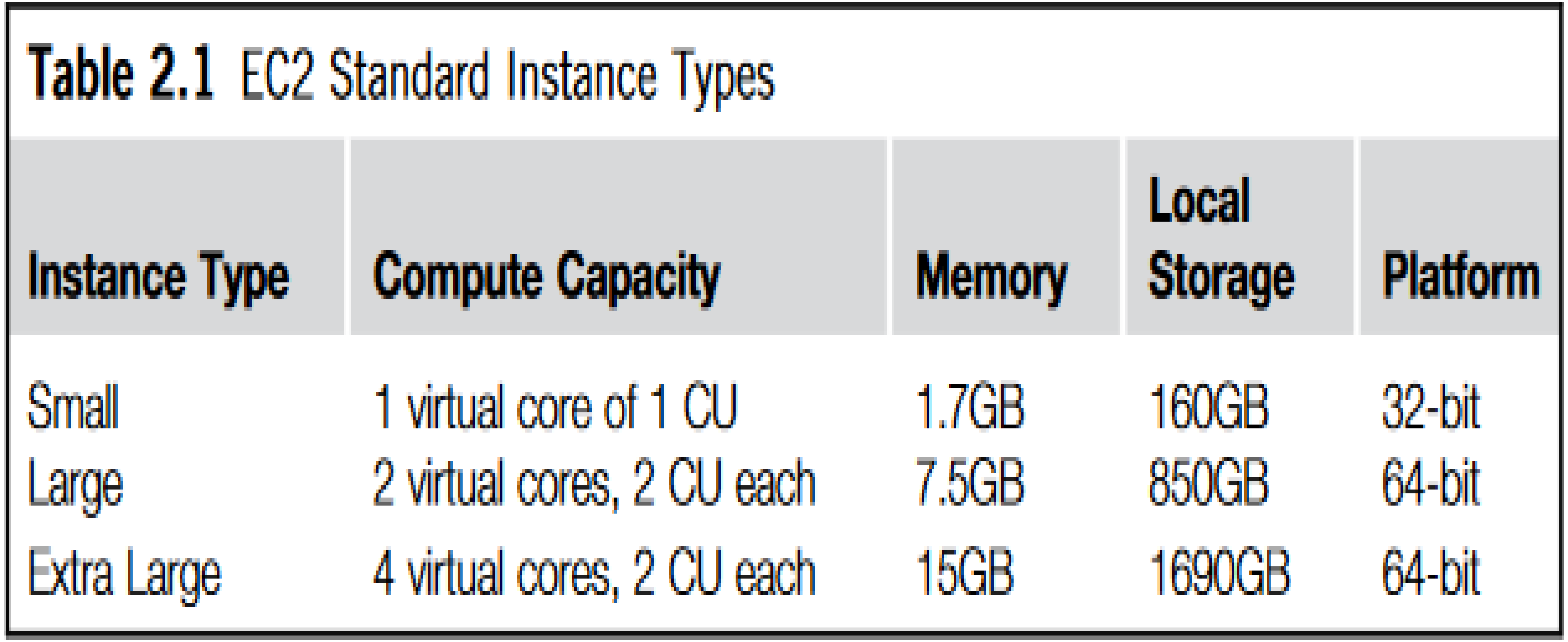


Table 2.1 EC2 Standard Instance Types

Instance Type	Compute Capacity	Memory	Local Storage	Platform
Small	1 virtual core of 1 CU	1.7GB	160GB	32-bit
Large	2 virtual cores, 2 CU each	7.5GB	850GB	64-bit
Extra Large	4 virtual cores, 2 CU each	15GB	1690GB	64-bit

Note: High memory instances are also available for databases and other memory-hungry applications

EC2 Softwares

- Amazon makes available operating system and application software in the form of **Amazon Machine Images(AMIs**).
- Operating systems available in AMIs include various flavors of **Linux**, such as **Red Hat Enterprise Linux and SuSE**, the **Windows server, and Solaris**.
- The required AMI has to be specified **when requesting the EC2 instance**, as demonstrated. The AMI running on an EC2 instance is also called the **root AMI**.
- Software available includes databases such as IBM DB2, Oracle and Microsoft SQL Server. A wide variety of other application software and middleware, such as Hadoop, Apache, and Ruby on Rails, are also available.

Accessing additional softwares on EC2

There are two ways of using additional software not available in standard AMIs.

1. Request a standard AMI, and then install the additional software needed. This AMI can then be saved as one of the available AMIs in Amazon.
2. Import a VMware image as an AMI using the **ec2-import-instance** and **ec2-import-disk-image** commands

Regions and Availability Zones of EC2

- EC2 offers **regions**, which are the same as the S3 regions described in the earlier slides of S3.
- Within a **region**, there are multiple **availability zones**, where each availability zone corresponds to a virtual data center that is isolated (for failure purposes) from other availability zones.

For example, an enterprise that wishes to have its EC2 computing instances in Europe could select the “Europe” region when creating EC2 instances.

- By creating two instances in different availability zones, the enterprise could have a highly available configuration that is tolerant to failures in any one availability zone.

Load Balancing and Scaling on EC2

- EC2 provides the **Elastic Load Balancer**, which is a service that balances the load across multiple servers.
- The default load balancing policy is to treat all requests as being independent.
- It is also possible to have **timer-based and application controlled sessions**, whereby successive requests from the same client are routed to the same server based upon time or application direction.
- The **load balancer** also scales the number of servers up or down depending upon the load. This can also be used as a failover policy, since failure of a server is detected by the Elastic Load Balancer.
- If the load on the remaining server is too high, the **Elastic Load Balancer** could start a new server instance.

EC2 Storage Resources

1. **Amazon S3:** Highly available object store (already covered in the earlier slides).
2. **Elastic Block Service:** Permanent block storage
3. **Instance Storage:** Transient block storage.

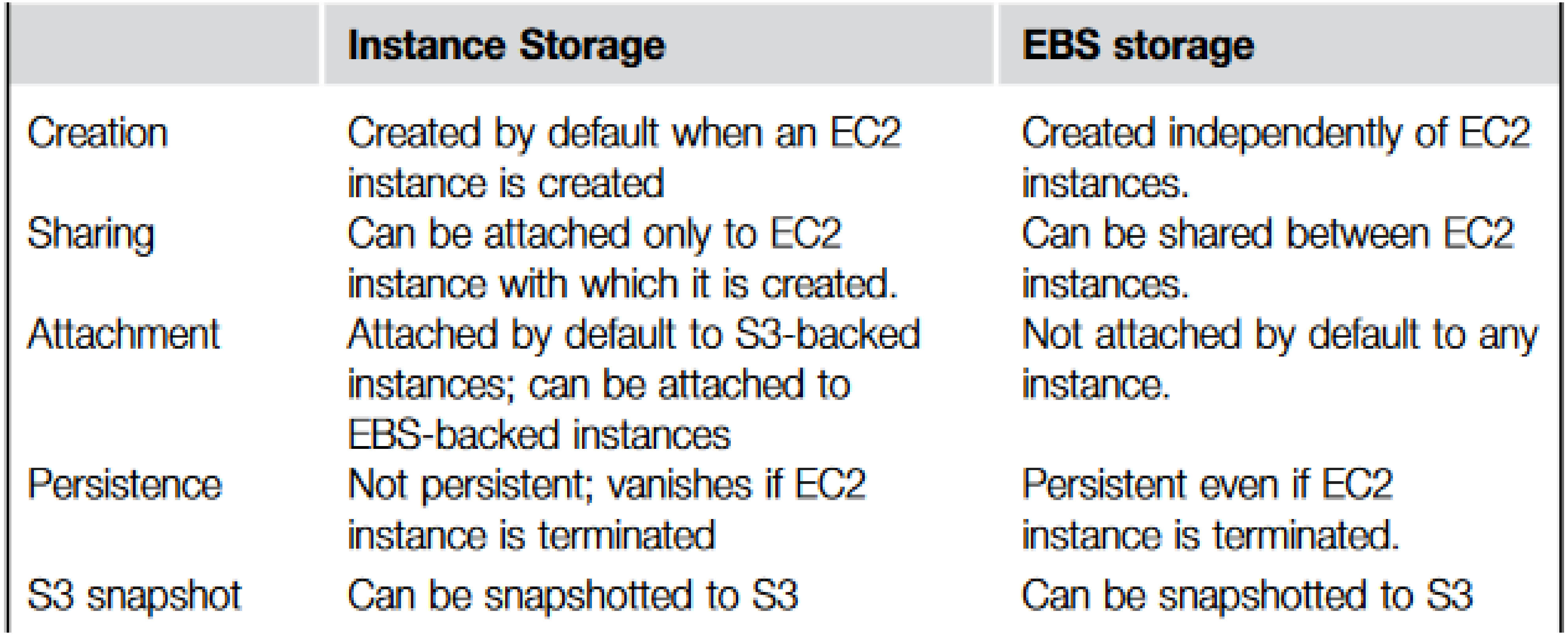
2. Elastic Block Service (EBS)

- In the same way that S3 provides file storage services, EBS provides a block storage service for EC2.
- It is possible to request an EBS disk volume of a particular size and attach this volume to one or multiple EC2 instances using the instance ID returned during the time the volume is created.
- The EBS volume has an existence independent of any EC2 instance, which is critical to have persistence of data

3. Instance Service

- Every EC2 instance has local storage that can be configured as a part of the compute resource. This is referred to as instance storage.
- Storage exists only as long as the EC2 instance exists, and cannot be attached to any other EC2 instance.
- If the EC2instance is terminated, the instance storage ceases to exist.
- To overcome this limitation of local storage of Instance service, developers can use either **EBS or S3** for persistent storage and sharing.

Comparison of Instance Storage and EBS Storage



	Instance Storage	EBS storage
Creation	Created by default when an EC2 instance is created	Created independently of EC2 instances.
Sharing	Can be attached only to EC2 instance with which it is created.	Can be shared between EC2 instances.
Attachment	Attached by default to S3-backed instances; can be attached to EBS-backed instances	Not attached by default to any instance.
Persistence	Not persistent; vanishes if EC2 instance is terminated	Persistent even if EC2 instance is terminated.
S3 snapshot	Can be snapshotted to S3	Can be snapshotted to S3

EC2 Networking Resources

- Network resources are also needed by applications in addition to compute and storage resources.
- For networking between EC2 instances, EC2 offers both a **public address** as well as a **private address** to each instance.
- It also offers **DNS services** for managing DNS names associated with these IP addresseees. **Amazon Route 53** effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS.
- **Elastic IP address:** It is a public IPv4 address, which is reachable from the internet. These addresses are independent of any instance, and can be used to support failover of servers

Virtual Private Cloud

- A virtual private cloud (VPC) is a secure, isolated private cloud hosted within a public cloud.
- Enterprises that desire more control over their networking configuration can use Virtual Private Cloud (VPC). Examples of VPC are provided below:

1. The ability to allocate both public and private IP addresses to instances from any address range.
2. The ability to divide the addresses into subnets and control the routing between subnets.
3. The ability to connect the EC2 network with an Intranet using a VPN tunnel.

Simple EC2 Example: Setting up a Web Server

The web server will be created as an EBS-backed instance, to avoid the necessity of having to periodically back up the storage to S3.

The process is broken down into four steps:

1. Selecting the AMI for the instance (Amazon Images" and "Amazon Linux" brings up a list of Linux images supplied by Amazon).
2. Creating the EC2 instance and installing the web server.
3. Creating an EBS volume for data, such as HTML files and so on.
4. Setting up networking and access rules (for allowing external access to the Web server).

Assumptions:

i. The data needed for the web server (HTML files, scripts, executables, and so on) are available, and have been uploaded to EC2.

ii. The web server needed also has to be uploaded to EC2 and then installed (in reality, a web server instance may be available as an image as well)

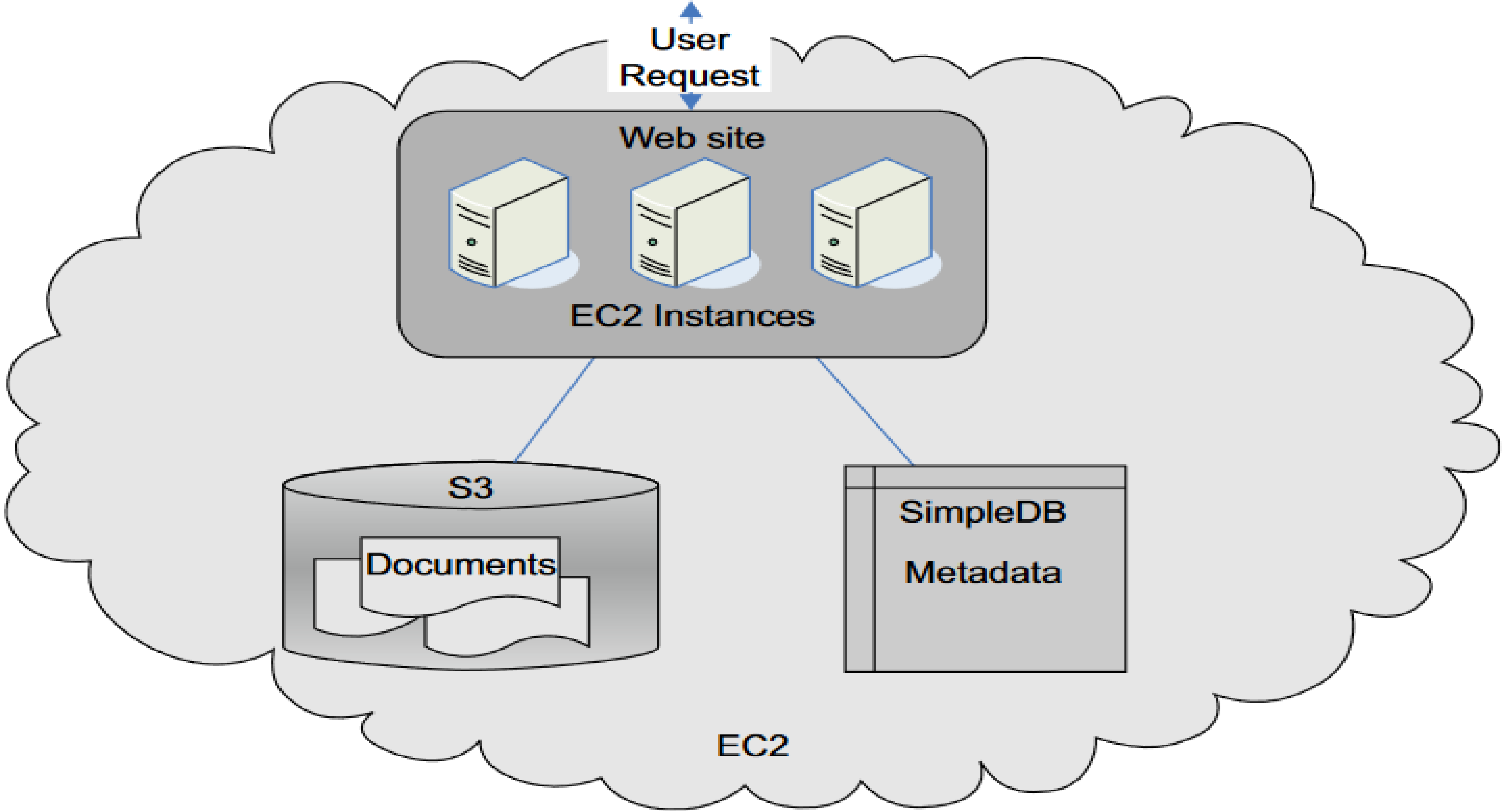
Case study of Using Amazon EC2 for Pustak Portal

What is Pustak Portal (a simple book publishing portal): It allows authors to upload and share book chapters or short articles in various formats with readers, who have to be registered with the portal.

Requirements:

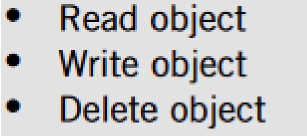
- It is necessary to store the documents, together with metadata such as the file type, and an access control list of readers who have been given access permission.
- As a particular article may become very popular due to its topical nature, the load on the portal could vary greatly, and it is necessary that the number of servers scale up and down with usage.

The High-level architecture of the enhanced Pustak Portal



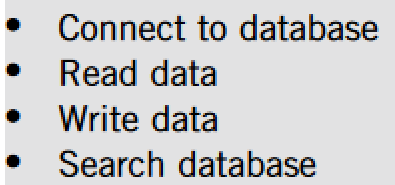
Pustak Portal (Cont…)

- Amazon S3 is used for storing articles.



- Read object
- Write object
- Delete object

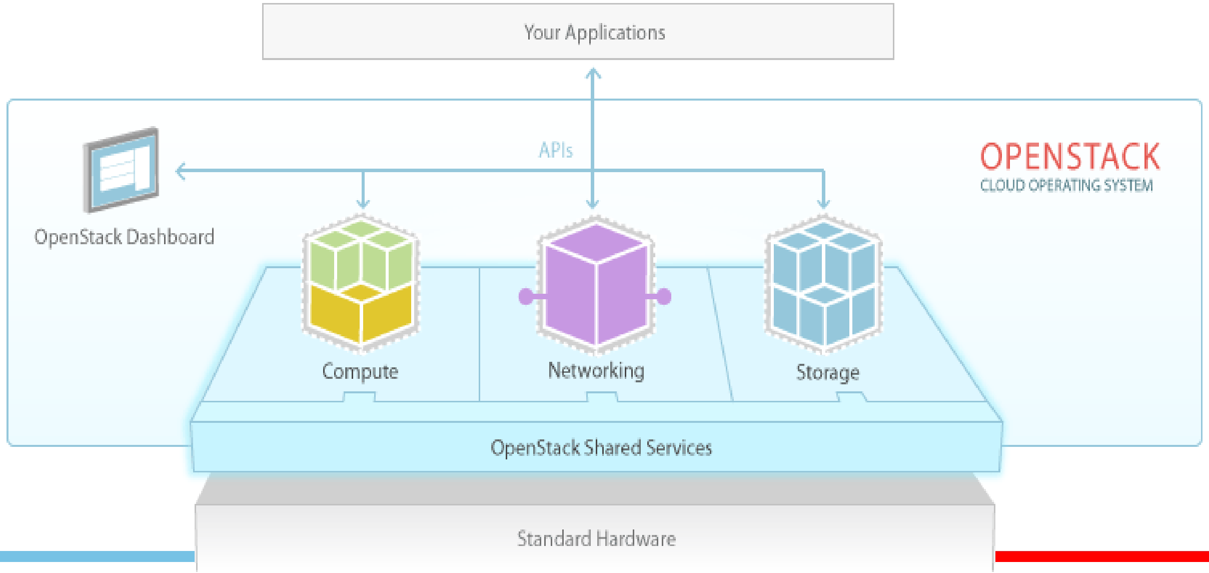
- The associated metadata of articles, such as the name of the article, author, and a list of readers, etc., are stored in SimpleDB.



- Connect to database
- Read data
- Write data
- Search database

Topic: Openstack

- OpenStack is considered as – Infrastructure as a Service (IaaS).
- It is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.
- It is one among several open-source cloud building software through which various organizations offer cloud services to their clients.
- The cloud can run on the commodity hardware that are available at economical costs.



Conceptual OpenStack Architecture

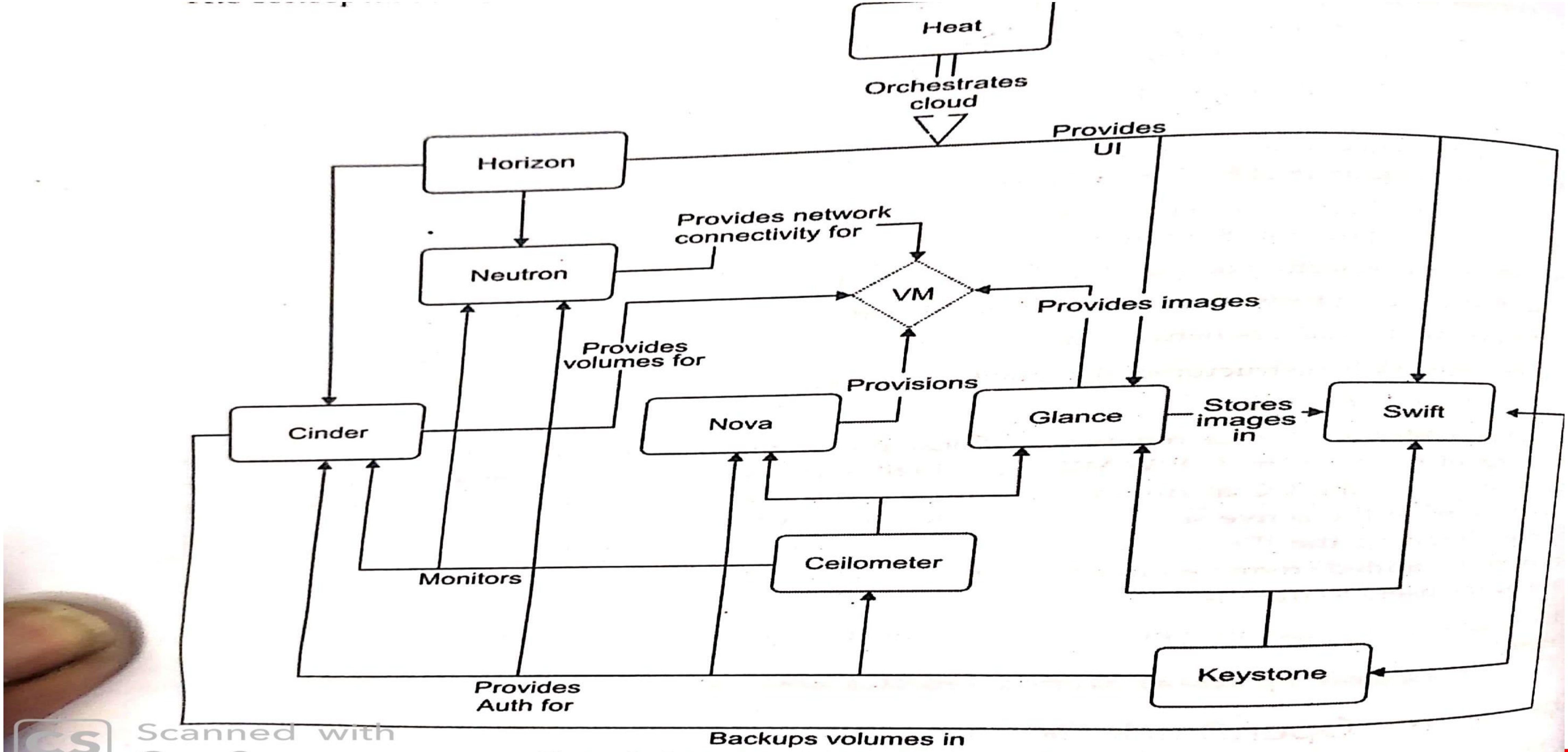


Figure 3: Conceptual OpenStack Architecture

Logical OpenStack Architecture

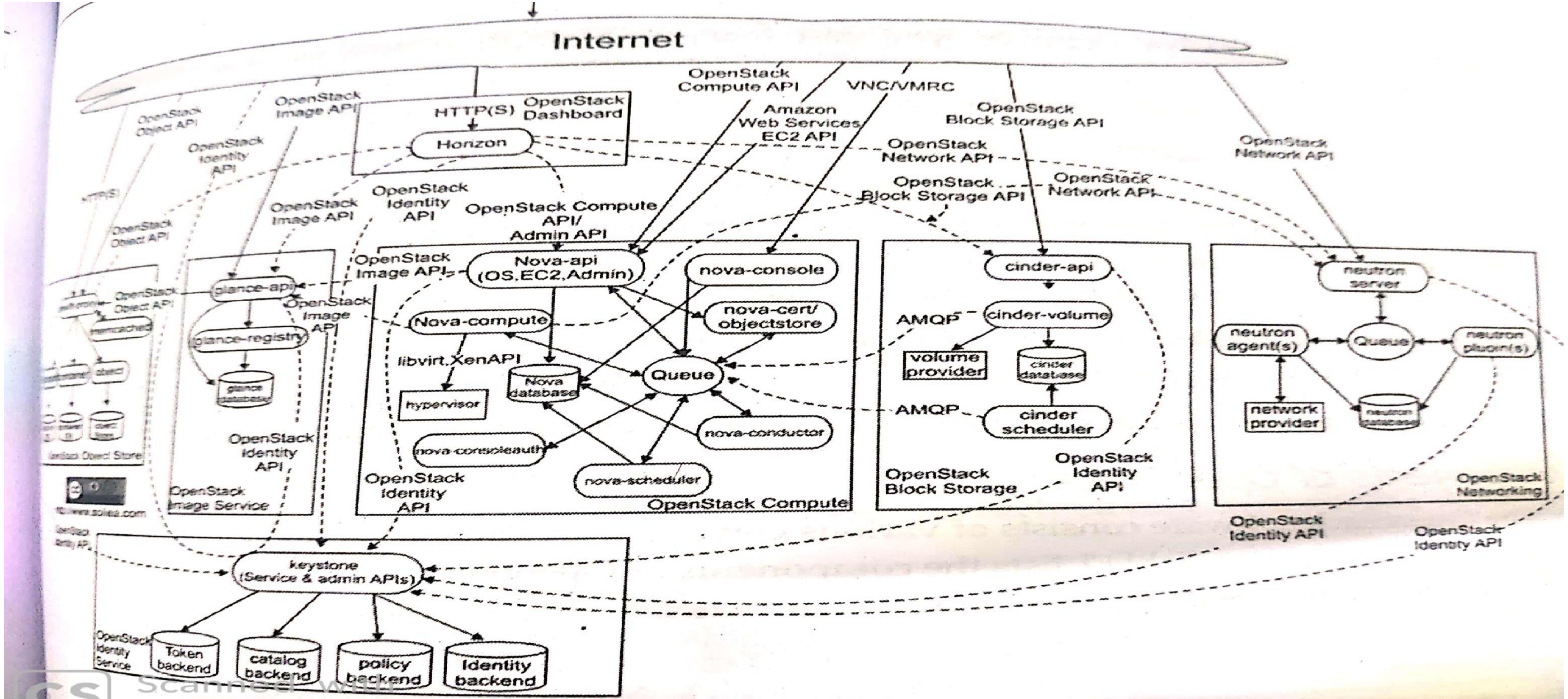
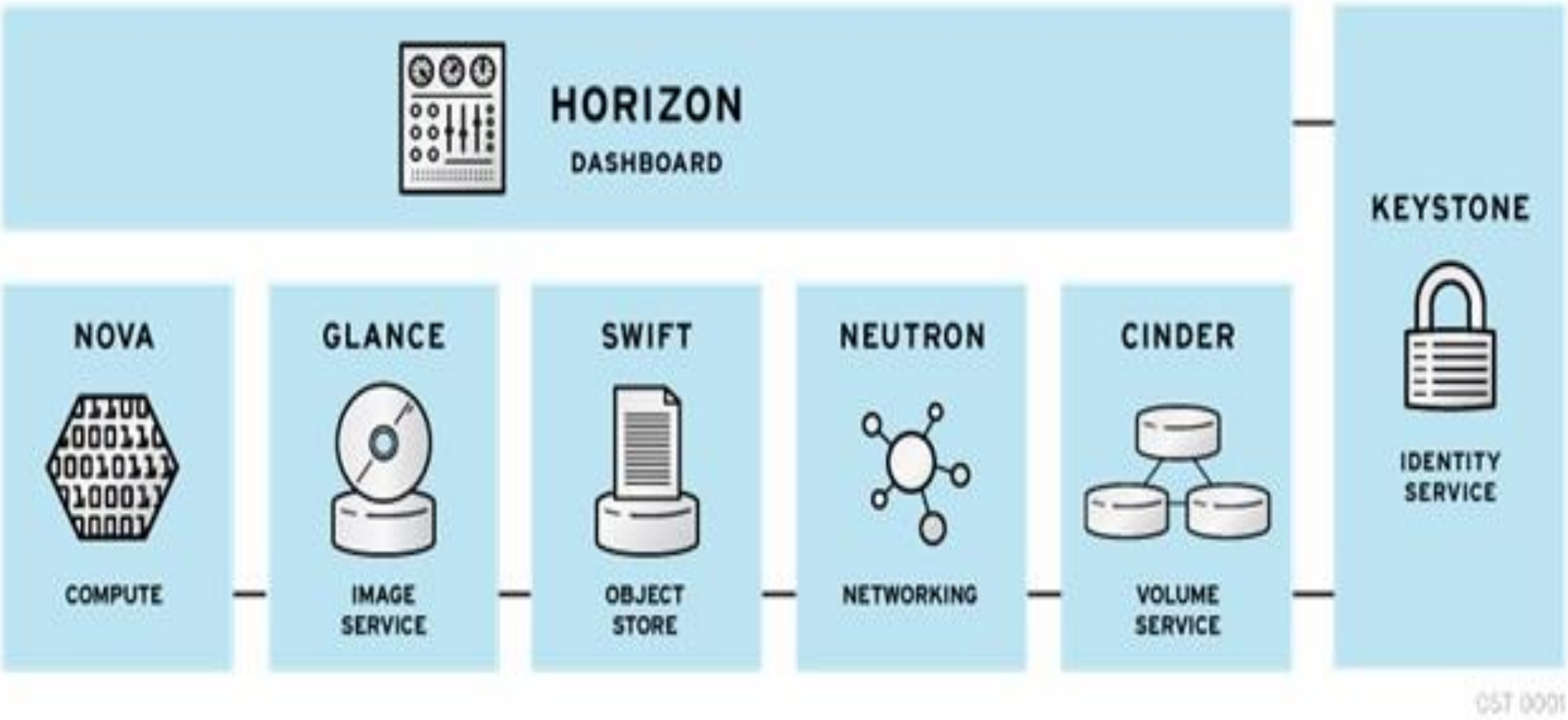


Figure 4: Logical OpenStack Architecture

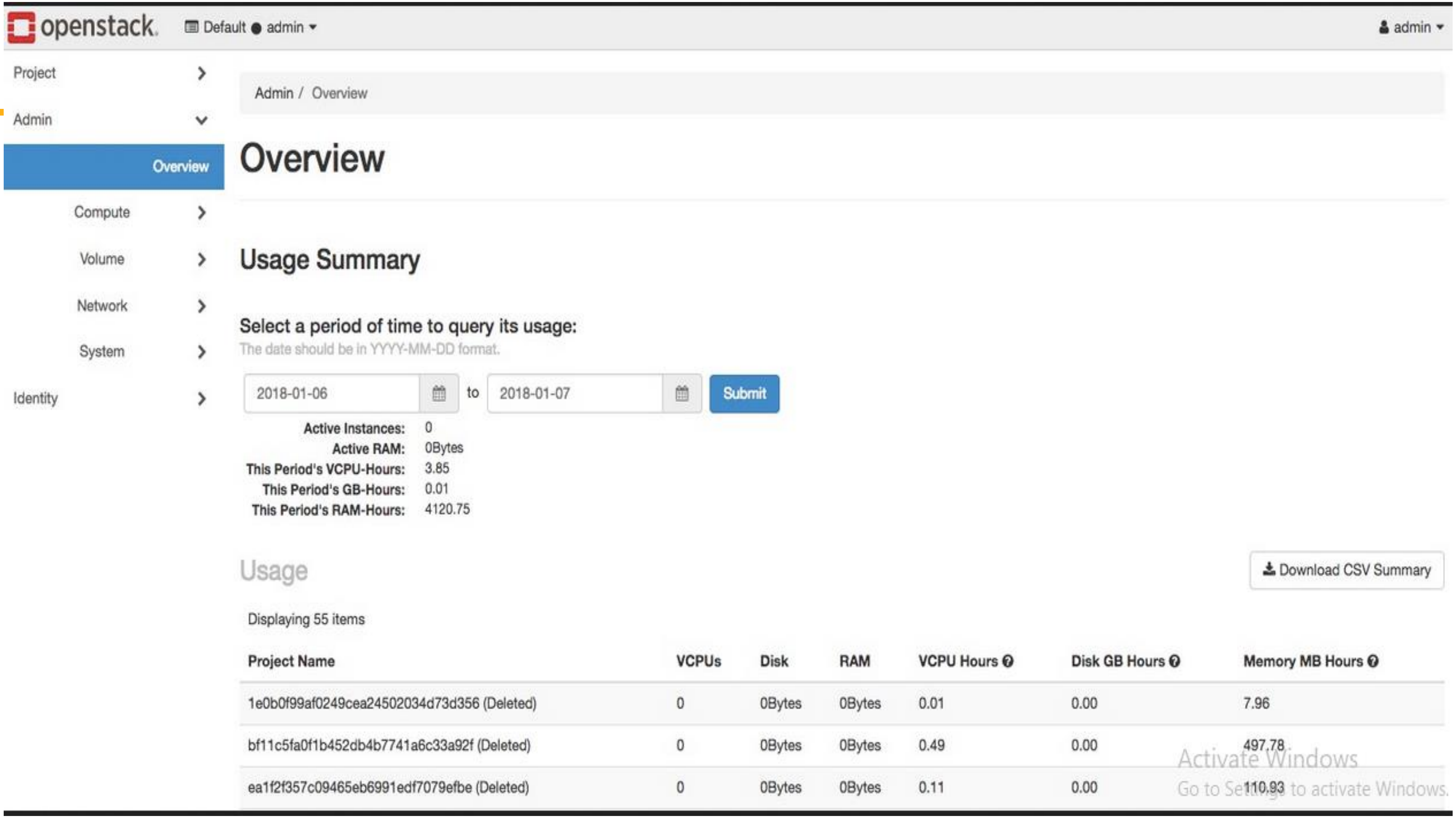
Openstack Key Components



Openstack Components. Cont.

Horizon – Dashboard

- It provides a modular web-based user interface for all the OpenStack services. With this web GUI, you can perform most operations on your cloud like **launching an instance**, **assigning IP addresses** and **setting access controls**.
- Its primary objective is to interact with the backend API's of other components and execute requests initiated by users.
- It interacts with keystone authentication service, to authorize requests before doing anything



Overview

Usage Summary

Select a period of time to query its usage:

The date should be in YYYY-MM-DD format.

2018-01-06 to 2018-01-07

Active Instances:	0
Active RAM:	0Bytes
This Period's VCPU-Hours:	3.85
This Period's GB-Hours:	0.01
This Period's RAM-Hours:	4120.75

Usage

Displaying 55 items

Project Name	VCPUs	Disk	RAM	VCPU Hours	Disk GB Hours	Memory MB Hours
1e0b0f99af0249cea24502034d73d356 (Deleted)	0	0Bytes	0Bytes	0.01	0.00	7.96
bf11c5fa0f1b452db4b7741a6c33a92f (Deleted)	0	0Bytes	0Bytes	0.49	0.00	497.78
ea1f2f357c09465eb6991edf7079efbe (Deleted)	0	0Bytes	0Bytes	0.11	0.00	110.93

OpenStack dashboard Admin tab

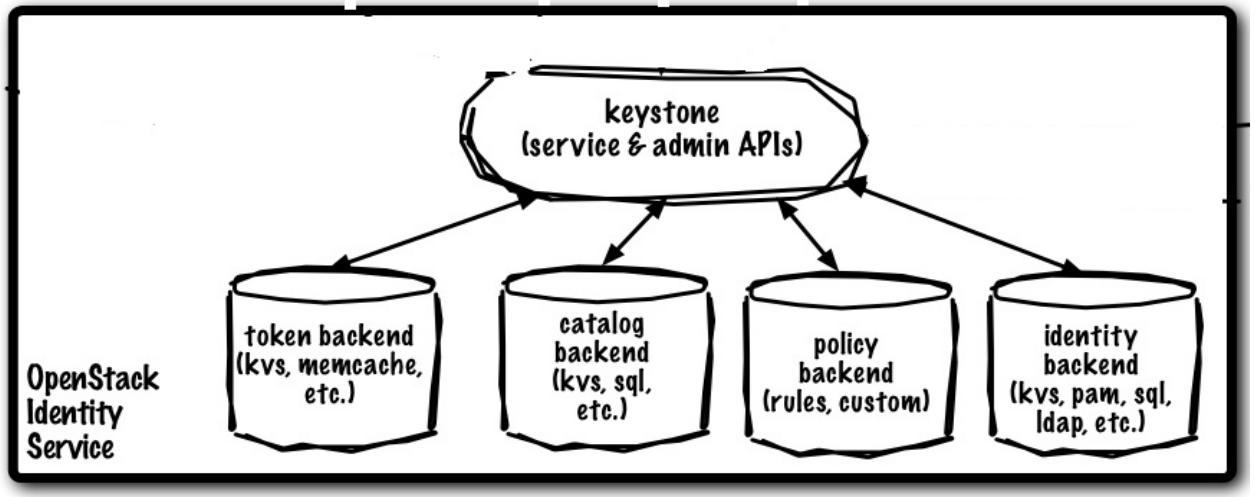
Keystone – Identity

Keystone is a framework for authentication and authorization for all the OpenStack services.

Keystone has two primary functions

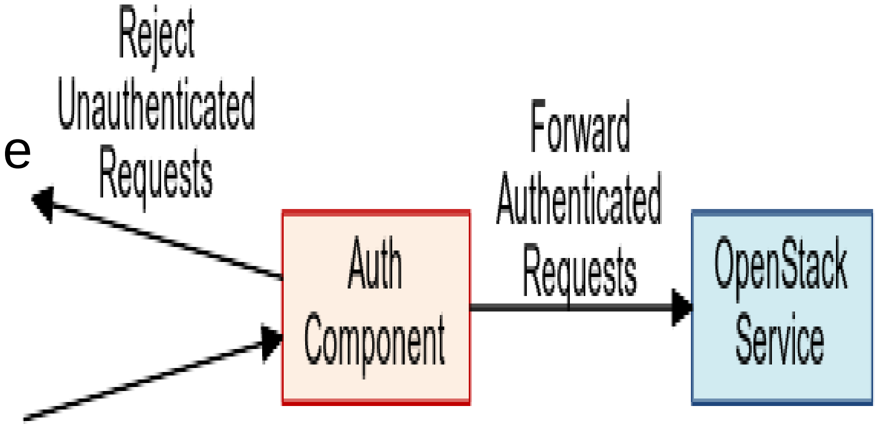
1) Manage Users. Like tracking of all users, and their permissions.

2) Service list/catalog. This is nothing but providing information regarding what services are available and their respective API endpoint details.



Keystone (OpenStack Identity Service):

The OpenStack Identity Service provides the cloud environment with an authentication and authorization system. In this system, users are a part of one or more projects. In each of these projects, they hold a specific role. Users need to have identity and a particular level of access in the cloud. When a user logs into the cloud, Keystone authenticates that he is indeed a user and authorises his level of access within the cloud.


Reject Unauthenticated Requests
Forward Authenticated Requests
Auth Component
OpenStack Service

Glance – Image Store

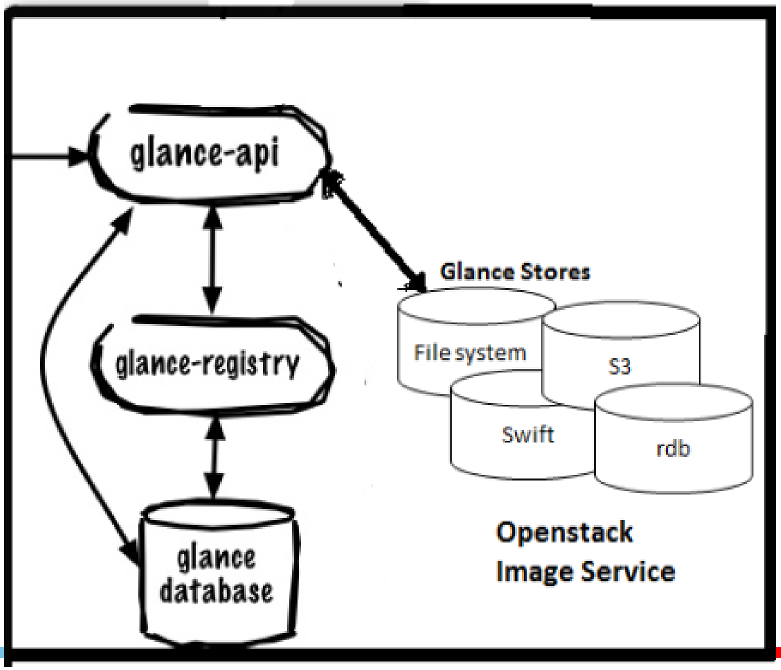
It provides discovery, registration and delivery services for disk and server images.
List of processes and their functions:

glance-api : It accepts Image API calls for image discovery, image retrieval and image storage.

glance-registry : it stores, processes and retrieves metadata about images (size, type, etc.).

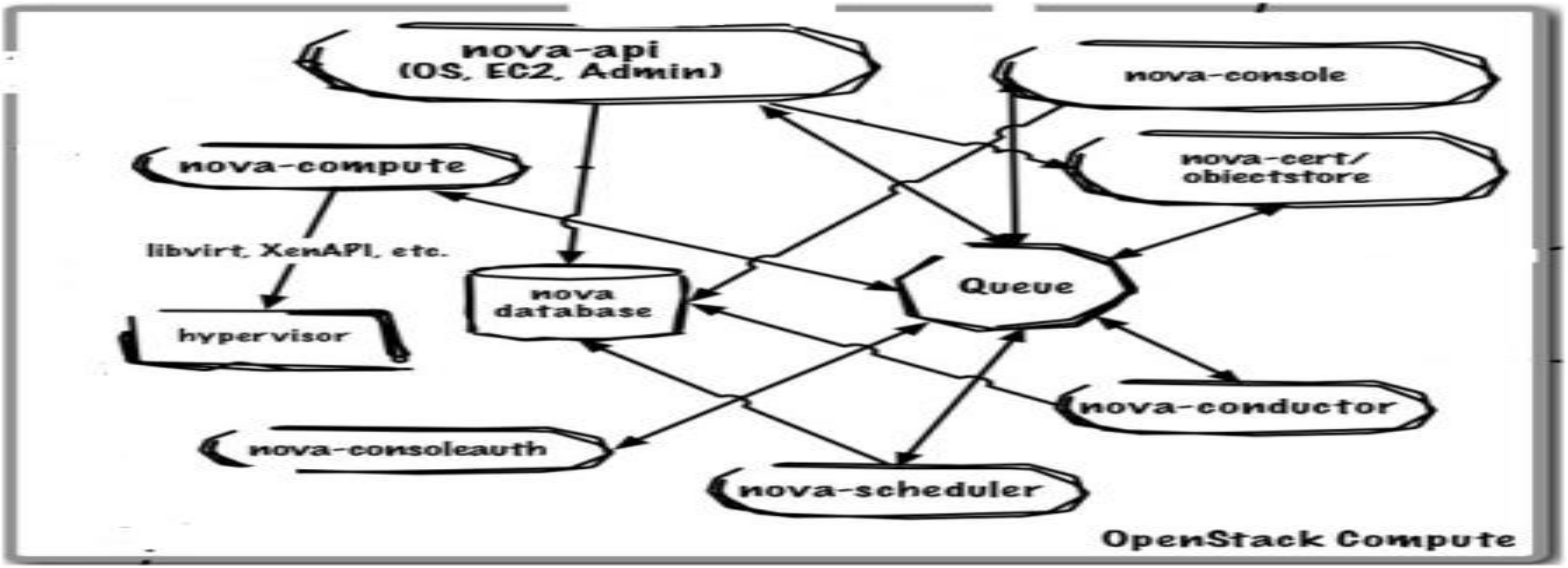
glance database : A database to store the image metadata.

A ***storage repository*** for the actual image files. Glance supports normal file-systems, Amazon S3, and Swift.



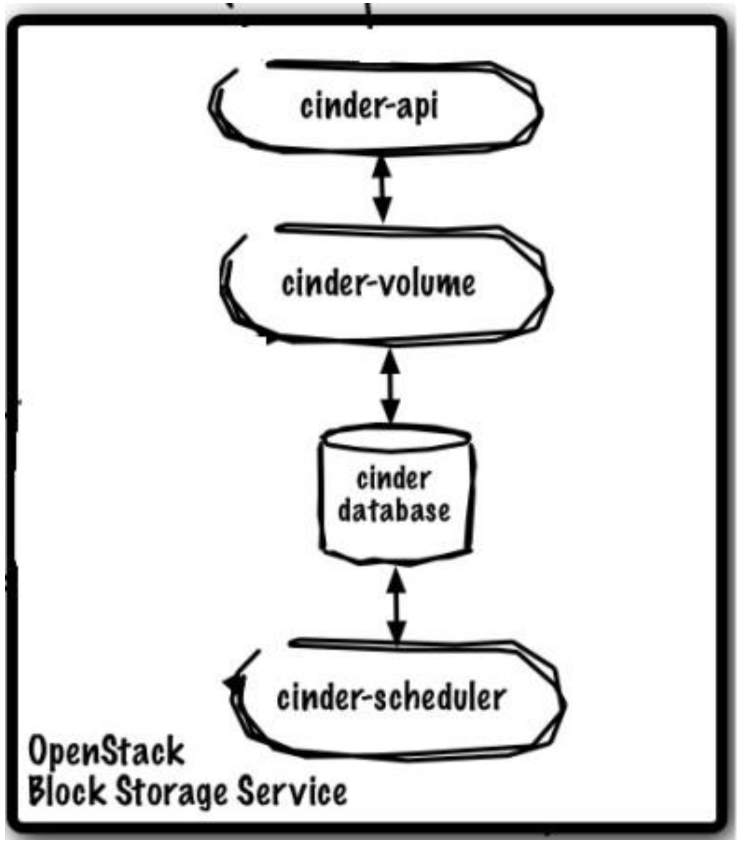
Nova – Compute

It provides virtual servers upon demand. Nova is the most complicated and distributed component of OpenStack. A large number of processes cooperate to turn end user API requests into running virtual machines.



Cinder – Block Storage

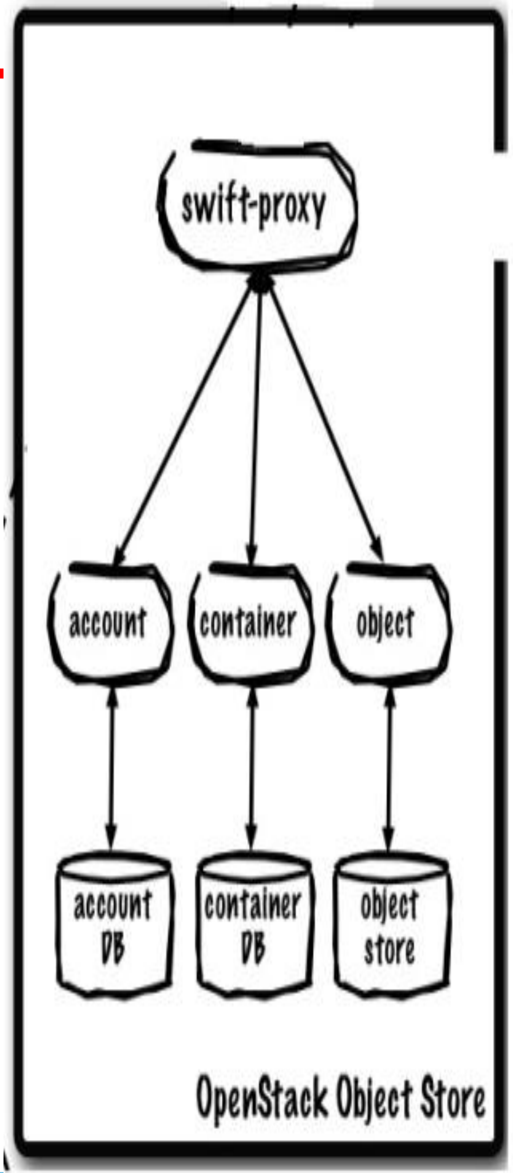
Cinder allows block devices to be exposed and connected to compute instances for expanded storage & better performance.



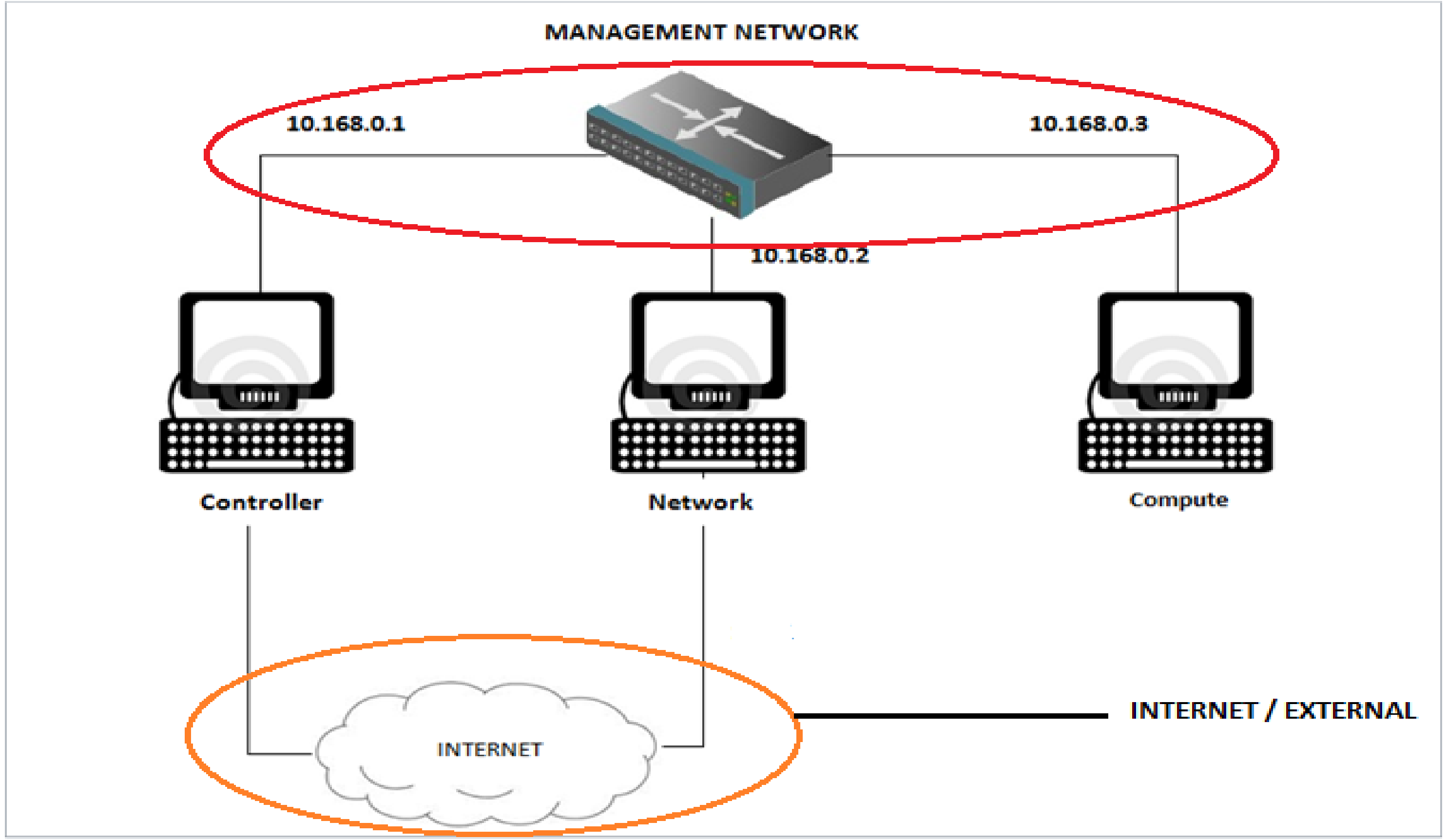
Swift – Object Storage

Object store allows you to store or retrieve files. It provides a fully distributed, API-accessible storage platform that can be integrated directly into applications or used for backup, archiving and data retention.

Note : Object Storage is not a traditional file system, but rather a distributed storage system for static data such as virtual machine images, photo storage, email storage, backups and archives.



Networking



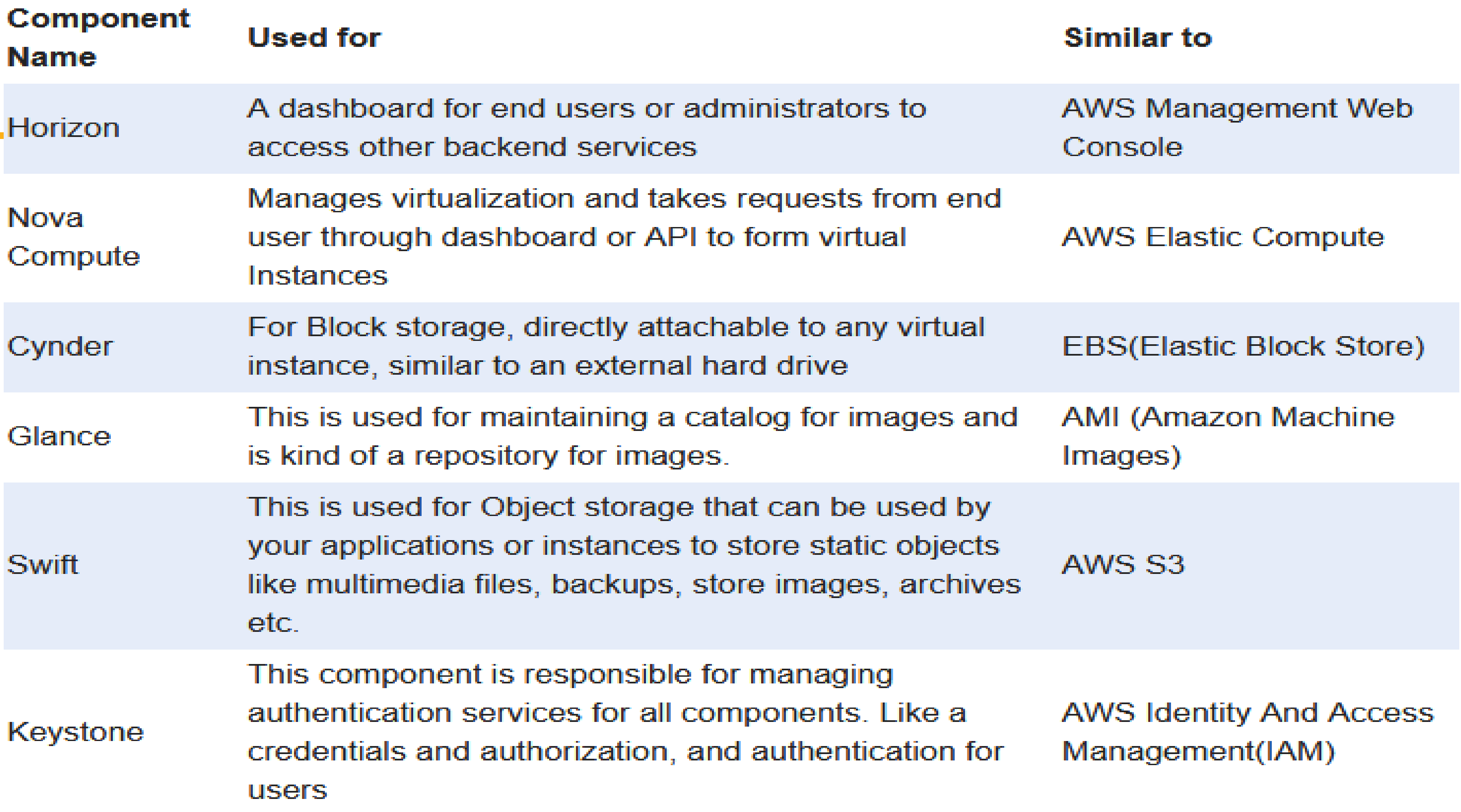
There are two networks :

1. Internal or Management network
2. External or Internet network

CONTROLLER NODE

- The controller is the central management system in a multinode cloud installation.

- The Controller node supplies API, scheduling, and other shared services for the cloud.
- The Controller node has the dashboard, the image store, and the identity service. Additionally, Nova compute management service as well as the Neutron server are also configured in this node.



Component Name	Used for	Similar to
Horizon	A dashboard for end users or administrators to access other backend services	AWS Management Web Console
Nova Compute	Manages virtualization and takes requests from end user through dashboard or API to form virtual Instances	AWS Elastic Compute
Cynder	For Block storage, directly attachable to any virtual instance, similar to an external hard drive	EBS(Elastic Block Store)
Glance	This is used for maintaining a catalog for images and is kind of a repository for images.	AMI (Amazon Machine Images)
Swift	This is used for Object storage that can be used by your applications or instances to store static objects like multimedia files, backups, store images, archives etc.	AWS S3
Keystone	This component is responsible for managing authentication services for all components. Like a credentials and authorization, and authentication for users	AWS Identity And Access Management(IAM)

OpenStack Installation References (Taxila)

- https://youtu.be/x5tuyzwq16k?list=PLvvQ7qimTOkmJFGS_uYlOA423PlvVmxOg (Installation)
- https://youtu.be/wzVSGGg4fsY?list=PLvvQ7qimTOkmJFGS_uYlOA423PlvVmxOg (Instance Creation)
- https://youtu.be/G1ZY4RorBiw?list=PLvvQ7qimTOkmJFGS_uYlOA423PlvVmxOg (Instance Creation with Volume)
- https://youtu.be/QFkfSgjJddI?list=PLvvQ7qimTOkmJFGS_uYlOA423PlvVmxOg (Swift Object Storage)
- Cloud Computing Black Book, Kailash Jaiswal, etc., 2020 Edition.

Topic: Managing Virtual Resources on the Cloud: **Provisioning and Migration**

Public cloud: Public cloud or external cloud describes cloud computing in a traditional mainstream sense, whereby resources are dynamically provisioned via publicly accessible Web applications/Web services (SOAP or RESTful interfaces) from an off-site third-party provider, who shares resources and bills on a fine-grained utility computing basis, the user pays only for the capacity of the provisioned resources at a particular time.

There are many examples for vendors who publicly provide **infrastructure as a service**. **Amazon Elastic Compute Cloud (EC2)** is the best known example. Few other examples **GoGrid**, **JoyentAccelerator**, **Rackspace**, **AppNexus**, **FlexiScale**, and **ManjrasoftAneka**.

Topic: Managing Virtual Resources on the Cloud: **Provisioning and Migration (Cont…)**

Private Cloud: A private cloud aims at providing public cloud functionality, but on private resources, **while maintaining control over an organization's data and resources to meet security and governance's requirements** in an organization. Private cloud exhibits a highly virtualized cloud data center **located inside your organization's firewall**.

- The best-known examples are **Eucalyptus** and **OpenNebula**.

High Availability: It allows virtual machines to automatically be restarted in case of an underlying hardware failure or individual VM failure.

- If one of your servers fails, the VMs will be restarted on other virtualized servers in the resource pool, restoring the essential services with minimal service interruption.

Two core services of getting resources for VMs

In Infrastructure as a Service (IaaS), the provisioning of required resources for systems and applications on a large number of physical machines is traditionally a time-consuming process with low assurance on deployment's time and cost.

Two core services are there that enable the users to get the best out of the IaaS model in public and private cloud setups.

1) **Virtual machine provisioning** and
2) **Migration services**

Why Virtual Machine Provisioning is required?

To provide a new virtual machine in a matter of minutes, **saving lots of time and effort.**

Generic procedure for virtual machine provisioning:

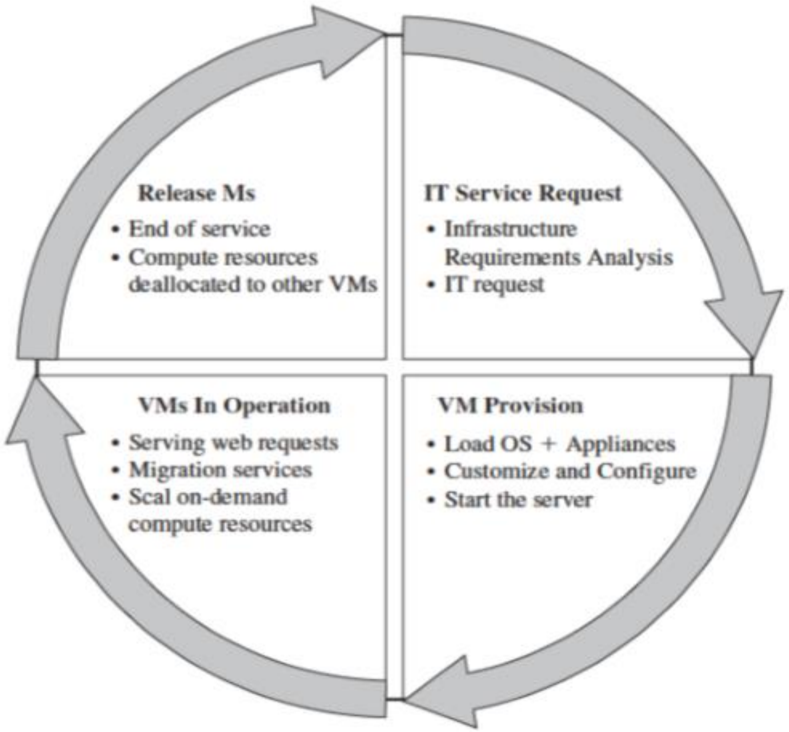
-Check the inventory for a new machine, get one, format, install OS required, and install services; a server is needed along with lots of security batches and appliances.

With the emergence of virtualization technology, **it is just a matter of minutes to achieve the above task of IT admin through** public cloud virtualization management software package or a private cloud management solution installed at your data center in order to provision the virtual machine inside the organization and within the private cloud setup.

Virtual Machine Provisioning and Manag eability Life Cycle

- The cycle starts by a request delivered to the IT department, stating the requirement for creating a new server for a particular service.
- This request is being processed by the IT administration to start seeing the servers’ resource pool, matching these resources with requirements
- Starting the provision of the needed virtual machine.
- Once it provisioned and started, it is ready to provide the required service according to an SLA(Service Level agreement).
- Virtual is being released; and free resources.

Virtual Machine Life Cycle



IT Service Request
- Infrastructure Requirements Analysis
- IT request

VM Provision
- Load OS + Appliances
- Customize and Configure
- Start the server

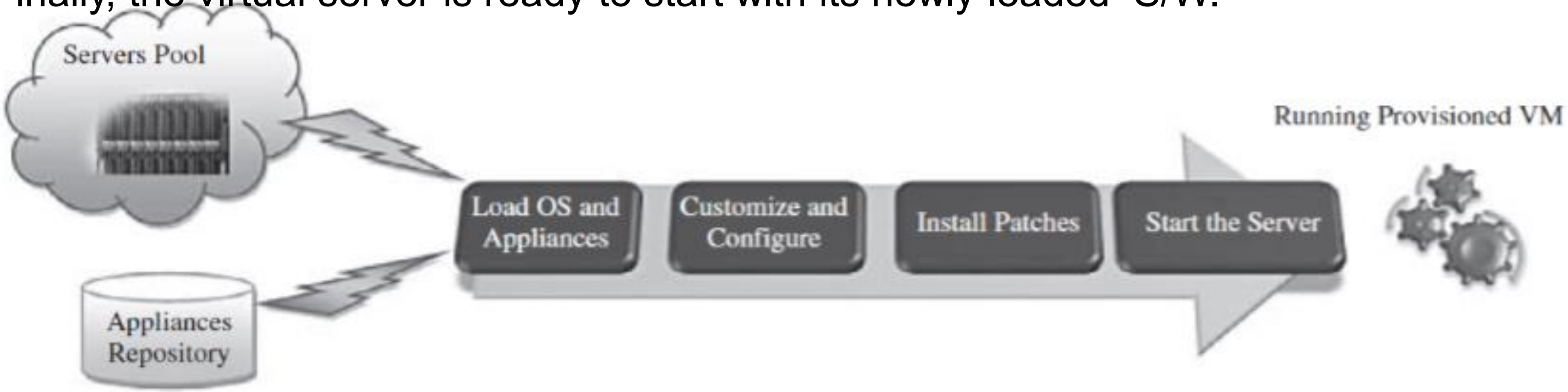
VMs In Operation
- Serving web requests
- Migration services
- Scal on-demand compute resources

Release Ms
- End of service
- Compute resources deallocated to other VMs

VM Provisioning Process

Steps to Provision VM -

- Select a server from a pool of available servers along with the appropriate OS template you need to provision the virtual machine.
- Load the appropriate software (operating system, device drivers, middleware, and the needed applications for the service), .
- Customize and configure the machine (e.g., IP address, Gateway) to an associated network and storage resources.
- Finally, the virtual server is ready to start with its newly loaded S/W.



VM Provisioning using templates

- Provisioning from a template reduces the time required to create a new virtual machine.
- Administrators can create different templates for different purposes.

  For example –

  - **Vagrant provision** tool using VagrantFile (template file) (demo)
  - **Heat** – Orchestration Tool of openstack (Heat template in YAML format) (demo – Instance creation in cloud, Load balancer in cloud)

This enables the administrator to quickly provision a correctly configured virtual server on demand.

Why Migration is required?

Migrations of a virtual machine is a matter of milliseconds: **saving time, effort, making the service alive for customers, and achieving the SLA/SLO agreements and quality-of-service (QoS) specifications required.**

- A particular VM is consuming more than its fair share of resources at the expense of other VMs on the same host, it will be eligible, for this machine, to either be moved to another underutilized host or assign more resources for it.

Virtual Machine Migration Services

Virtual Machine Migration Services

Migration service -

The process of moving a virtual machine from one host server or storage location to another. It plays an important role in datacenters by making it easy to adjust resource’s priorities to match resource’s demand conditions.

There are different techniques of VM migration-

- **Cold/regular migration,**
- **Hot/live migration, and**
- **Live storage migration of a virtual machine**.

In this process, all key machines’ components, such as **CPU, storage disks, networking, and memory,** are completely virtualized, thereby facilitating the entire state of a virtual machine to be captured by a set of easily moved data files.

Cold/regular migration

Cold migration is the migration of a **powered-off virtual machine** and is done in the following tasks:

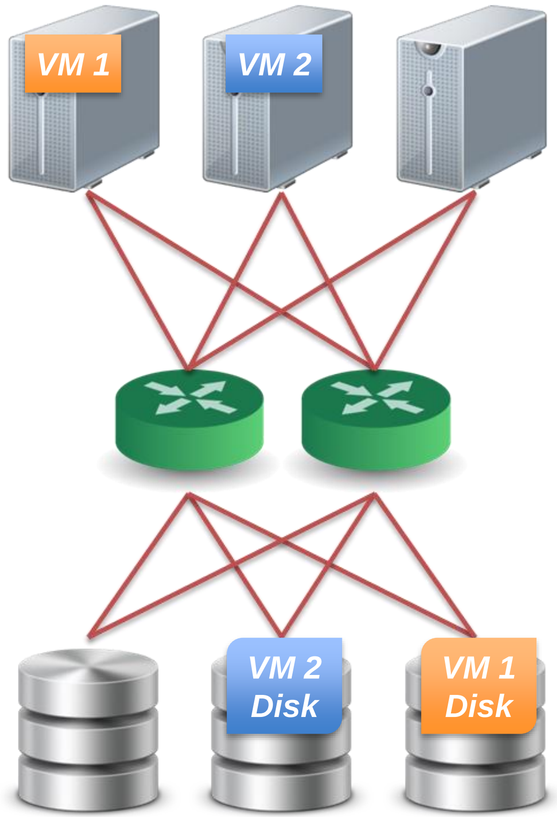
- If the option to move to a different datastore was chosen, the configuration files, including the **NVRAM file (BIOS settings), and log files** are moved from the source host to the destination host’s associated storage area. If you chose to move the virtual machine's disks, these are also moved.
- The virtual machine is registered with the new host.
- After the migration is completed, the old version of the virtual machine is deleted from the source host if the option to move to a different datastore was chosen.

Live Migration Technique (hot or real-time migration)

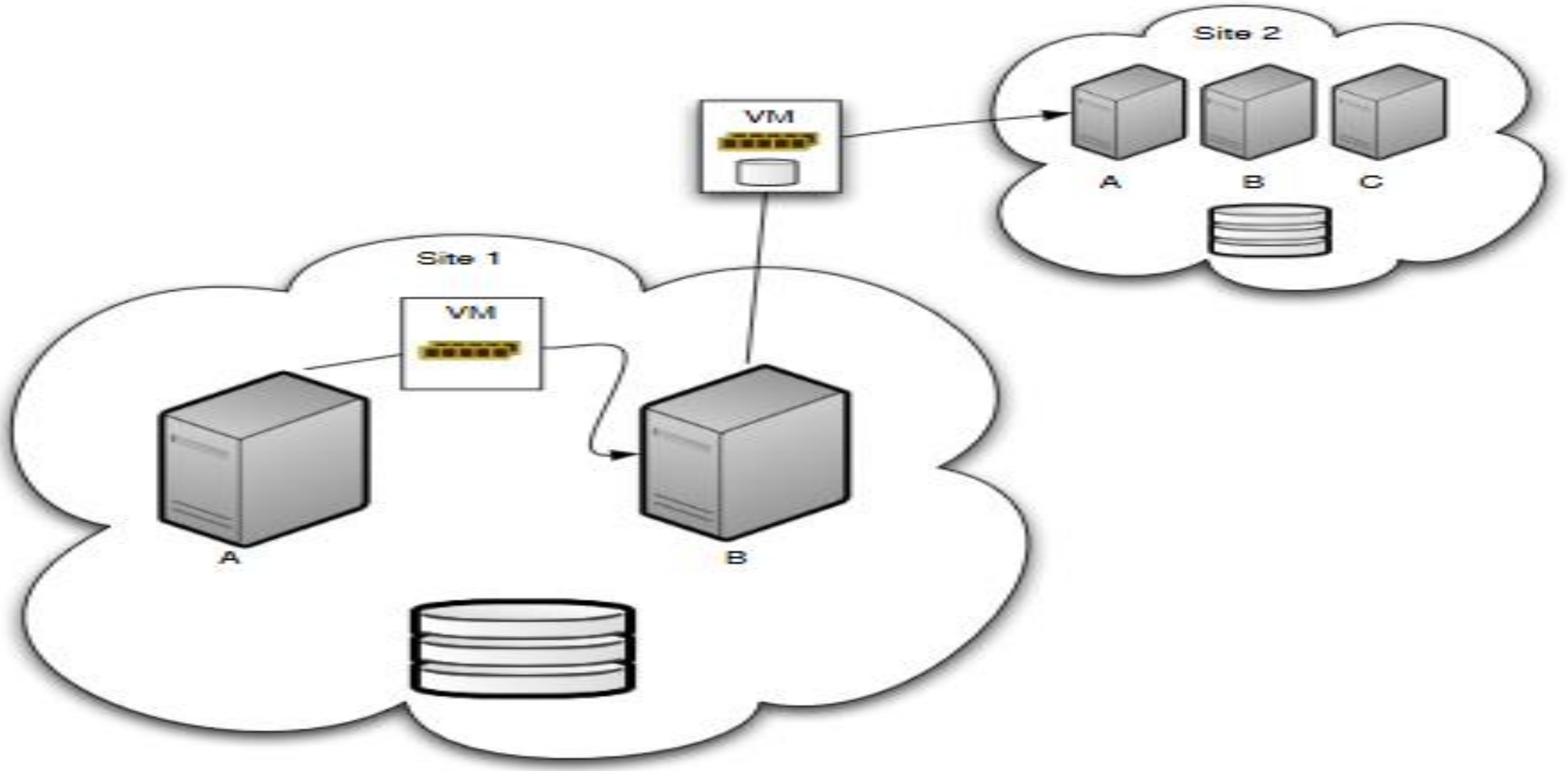
It can be defined as the movement of a virtual machine from one physical host to another while being powered on. When it is properly carried out, this process takes place without any noticeable effect from the end user's point of view (a matter of milliseconds). Live migration can also be used for load balancing.

Pre-assumption :

- We assume that all storage resources are separated from computing resources.
- Storage devices of VMs are attached from network :
  - ***NAS***: NFS, CIFS
  - ***SAN***: Fibre Channel
  - ***iSCSI***, network block device
  - ***drdb*** network RAID
- Require high quality network connection
  - Common L2 network (LAN)
  - L3 re-routing



In-site and Cross-site Migration



Live Migration Technique

Challenges of live migration :

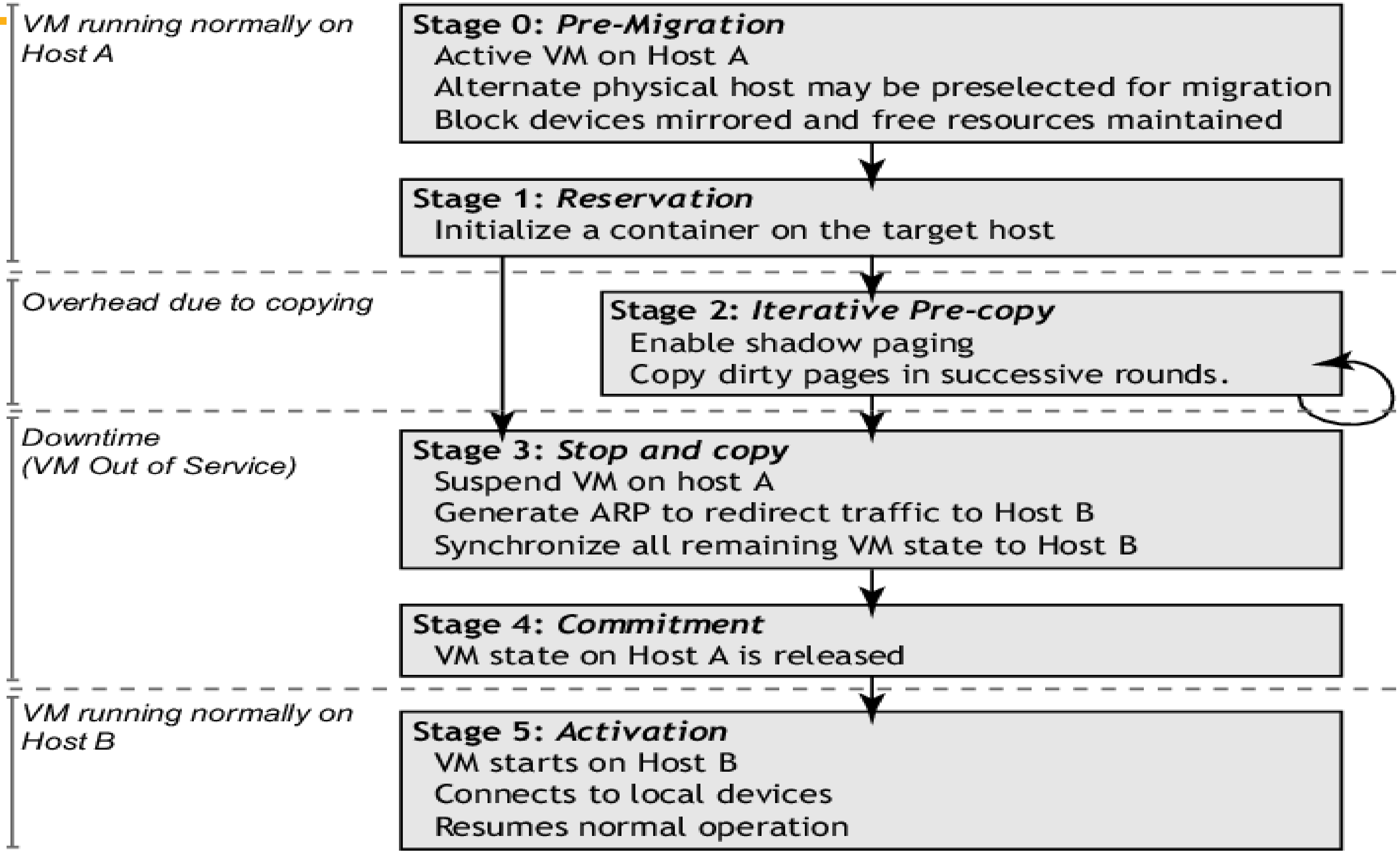
- VMs have lots of state in memory
- Some VMs have soft real-time requirements :
    - For examples, web servers, databases and game servers, ...etc.
    - Need to minimize down-time

Relocation strategy :

1. Pre-migration process
2. Reservation process
3. Iterative pre-copy
4. Stop and copy
5. Commitment

Reference (complete information of Live Migration) http://www.diva-portal.org/smash/get/diva2:7

Live Migration Technique



VM running normally on Host A

Stage 0: *Pre-Migration*
Active VM on Host A
Alternate physical host may be preselected for migration
Block devices mirrored and free resources maintained

Stage 1: *Reservation*
Initialize a container on the target host

Overhead due to copying

Stage 2: *Iterative Pre-copy*
Enable shadow paging
Copy dirty pages in successive rounds.

Downtime (VM Out of Service)

Stage 3: *Stop and copy*
Suspend VM on host A
Generate ARP to redirect traffic to Host B
Synchronize all remaining VM state to Host B

Stage 4: *Commitment*
VM state on Host A is released

VM running normally on Host B

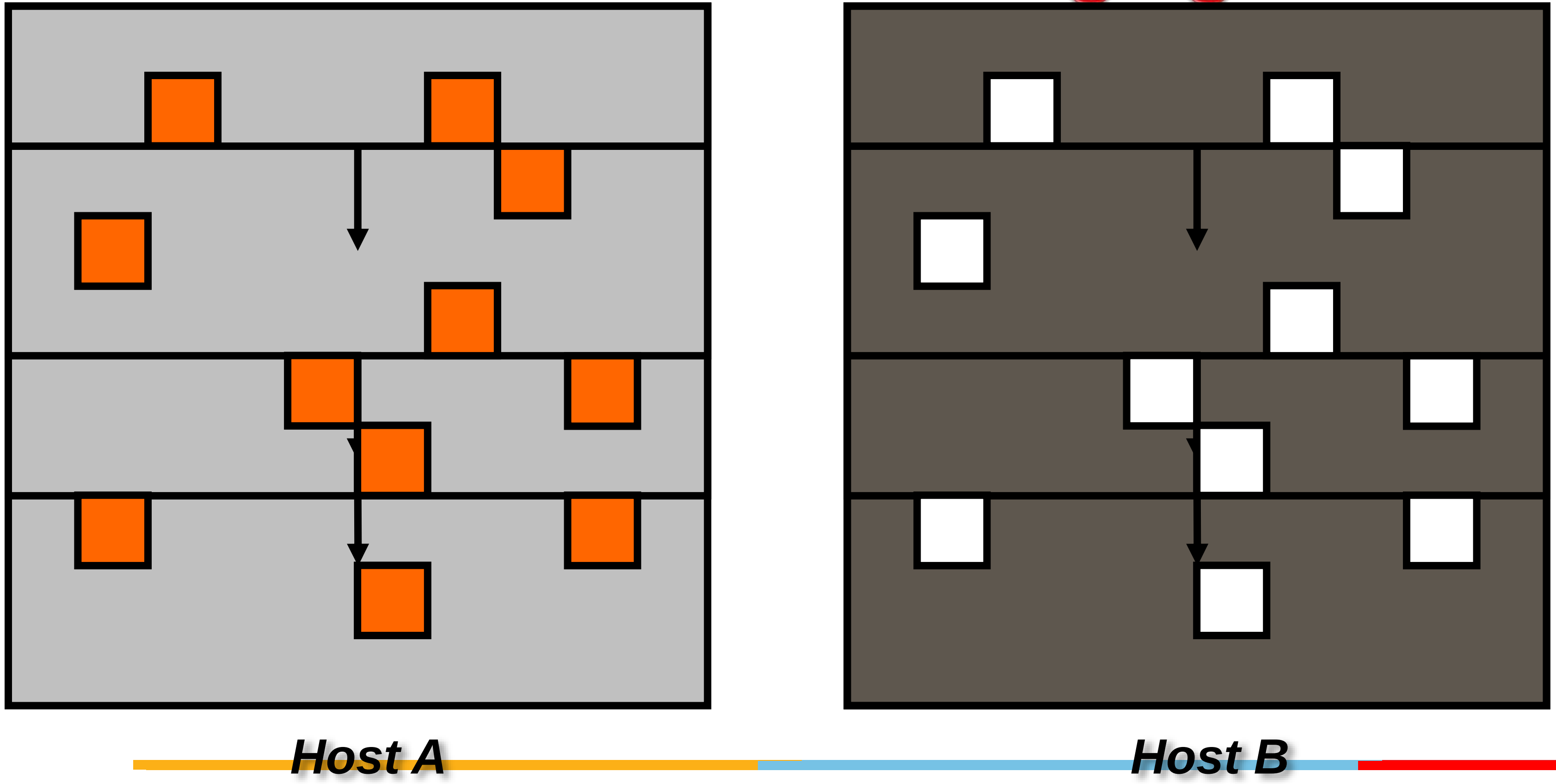
Stage 5: *Activation*
VM starts on Host B
Connects to local devices
Resumes normal operation

Note: We can migrate but with Short downtime.

Live Migration Technique

Live migration process :

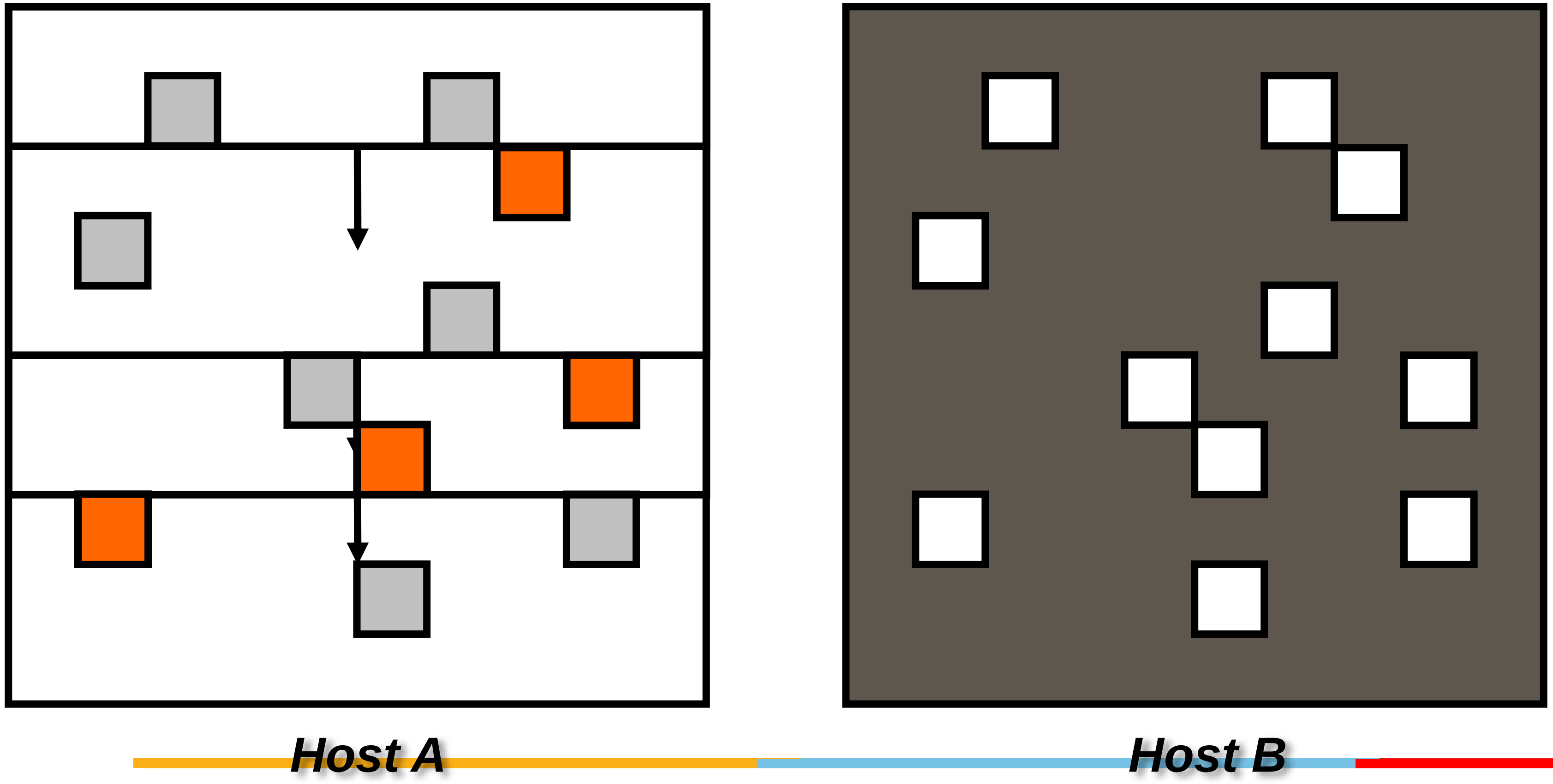
Pre-copy migration : Round 1, Enable Shadow Paging



Live Migration Technique

Live migration process :

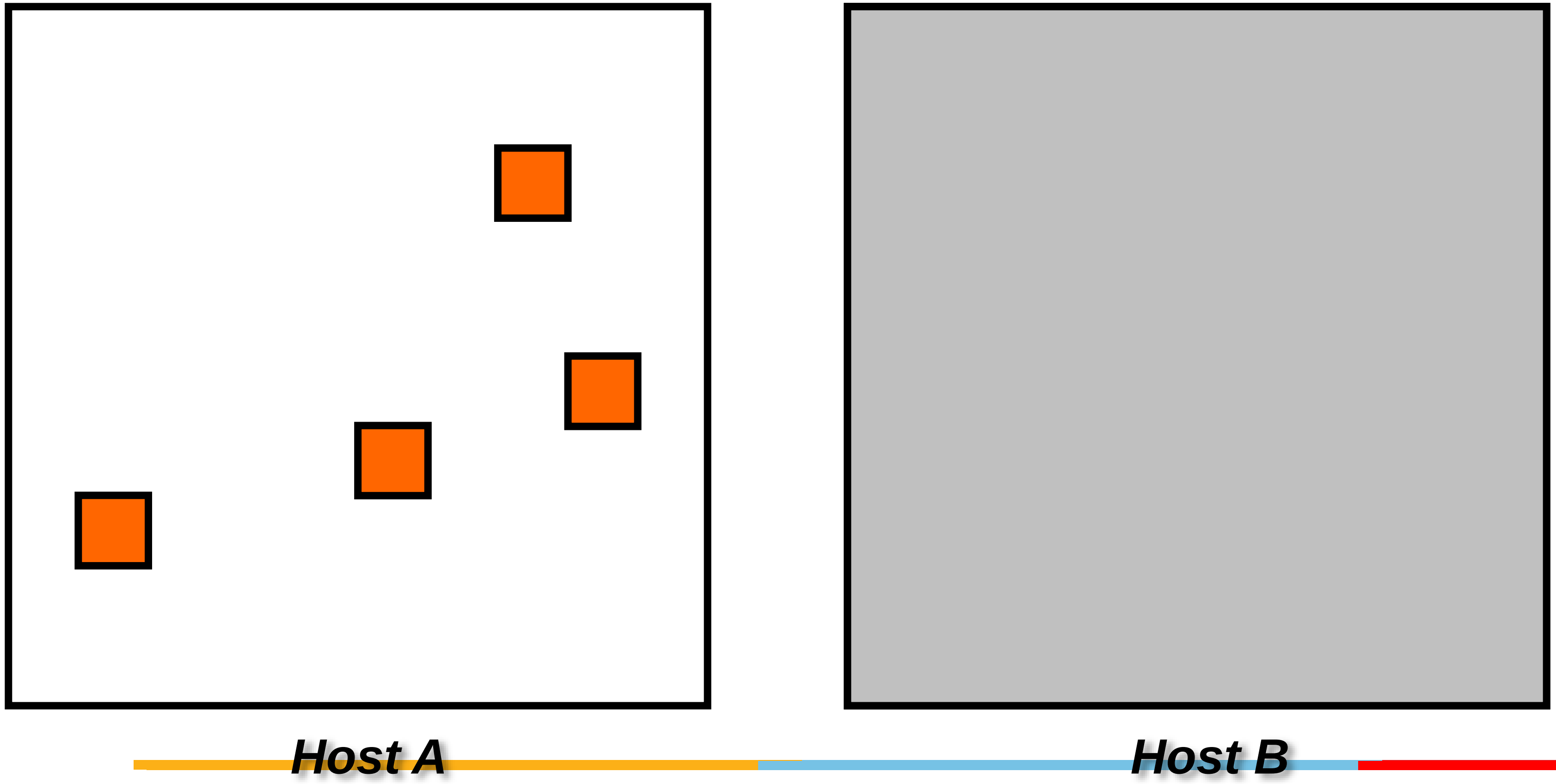
Pre-copy migration : Round 2



Live Migration Technique

Live migration process :

Stop and copy : Final Round



Live Storage Migration of Virtual Machine.

This migration technique constitutes moving the virtual disks or configuration file of a running virtual machine to a new data store without any interruption in the availability of the virtual machine’s service.

Refererences: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831656(v=ws.11)

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5071905 (A Live Storage Migration Mechanism over WAN for Relocatable Virtual Machine Services on Clouds)