BITS Pilani

Cloud Computing

SEWP ZG527

Agenda

- SaaS Recap
- Capacity management in Cloud computing
- Virtual Infrastructure management
- RESERVIOR Project
- Distributed management of virtual machines
- Reservation-based provisioning of virtualized resource
- Provisioning to meet SLA commitments
- Scheduling models and algorithms
- Haieza project

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Recap

Software as a Service - Summary

Capability

SaaS provides the following
- Hosted , Finished Product
- Subscription to Services

Why SaaS

Enabler : **Web Service**
✓ Accessibility & Portability

Characteristics

Not owned but subscribed to from an external service provider.
Designed to be Multi Tenant
Customer Configuration instead of Application configuration
Centralized management

Models

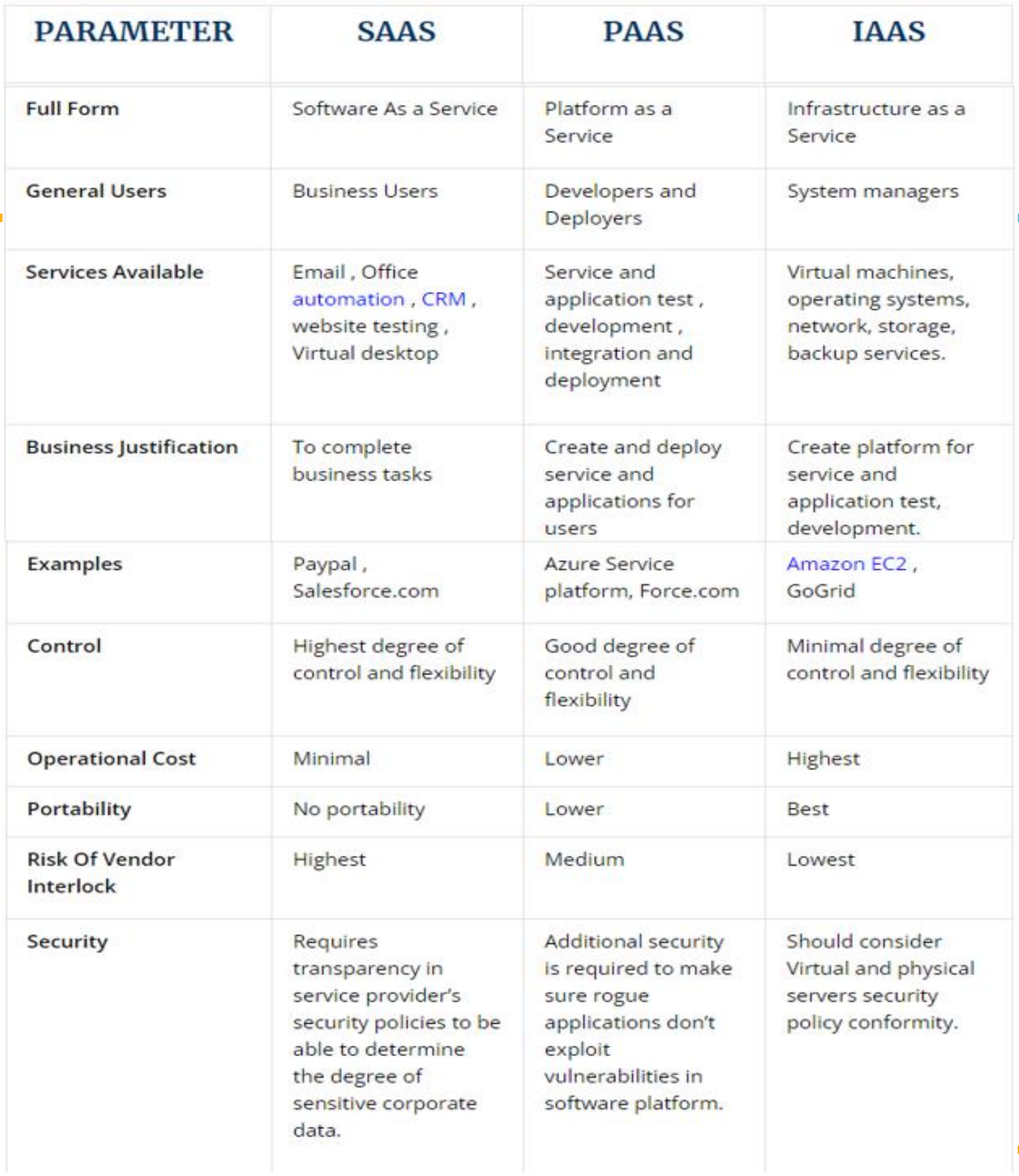
SaaS can be obtained as
(1) Managed Service
(2) Monthly Subscription

Benefit

No need to purchase Software or Licenses upfront
Centralized management helps SaaS Vendors
Lower TCO for users

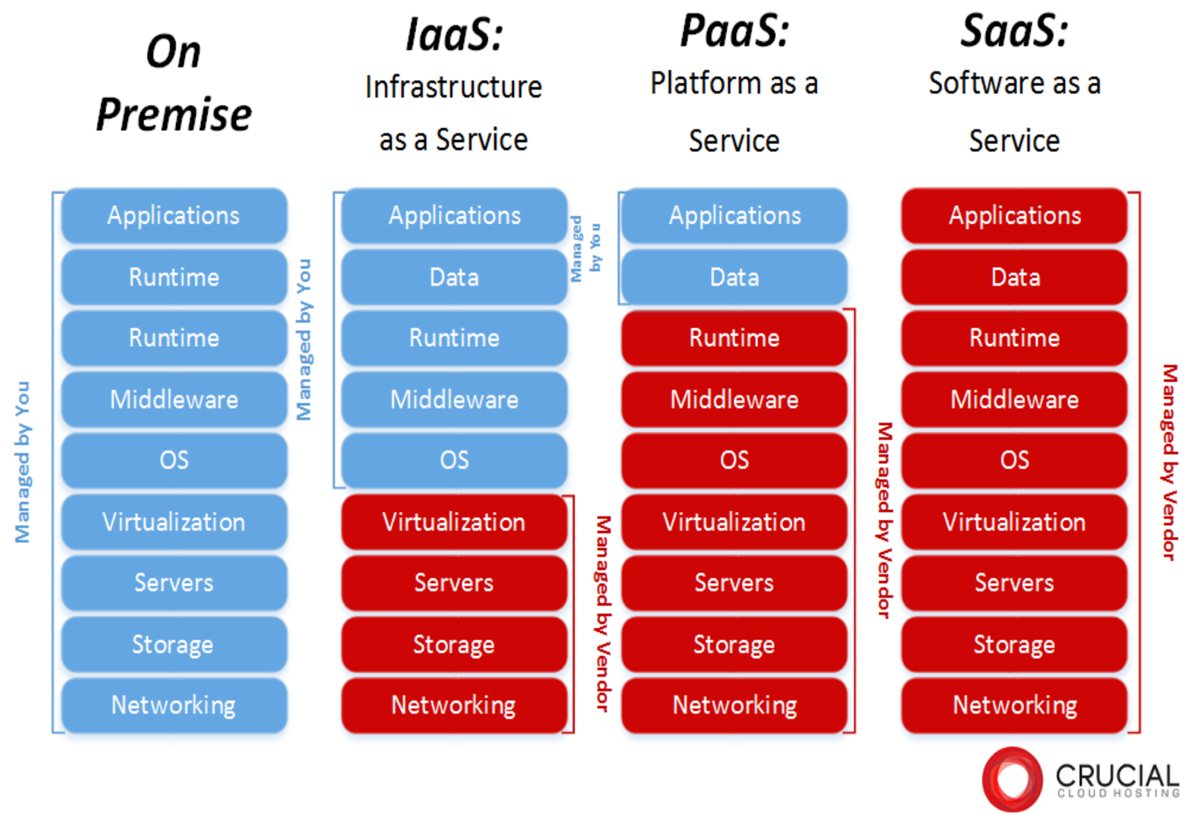
True SaaS Applicability

- **Enterprise Software Application**
    - Perform business functions
    - Organize internal and external information
    - Share data among internal and external users
    - The most standard type of software applicable to SaaS model
    - Example: Saleforce.com CRM application, Siebel On-demand application



PARAMETER	SAAS	PAAS	IAAS
Full Form	Software As a Service	Platform as a Service	Infrastructure as a Service
General Users	Business Users	Developers and Deployers	System managers
Services Available	Email , Office automation , CRM , website testing , Virtual desktop	Service and application test , development , integration and deployment	Virtual machines, operating systems, network, storage, backup services.
Business Justification	To complete business tasks	Create and deploy service and applications for users	Create platform for service and application test, development.
Examples	Paypal , Salesforce.com	Azure Service platform, Force.com	Amazon EC2 , GoGrid
Control	Highest degree of control and flexibility	Good degree of control and flexibility	Minimal degree of control and flexibility
Operational Cost	Minimal	Lower	Highest
Portability	No portability	Lower	Best
Risk Of Vendor Interlock	Highest	Medium	Lowest
Security	Requires transparency in service provider's security policies to be able to determine the degree of sensitive corporate data.	Additional security is required to make sure rogue applications don't exploit vulnerabilities in software platform.	Should consider Virtual and physical servers security policy conformity.

SaaS vs PaaS vs IaaS



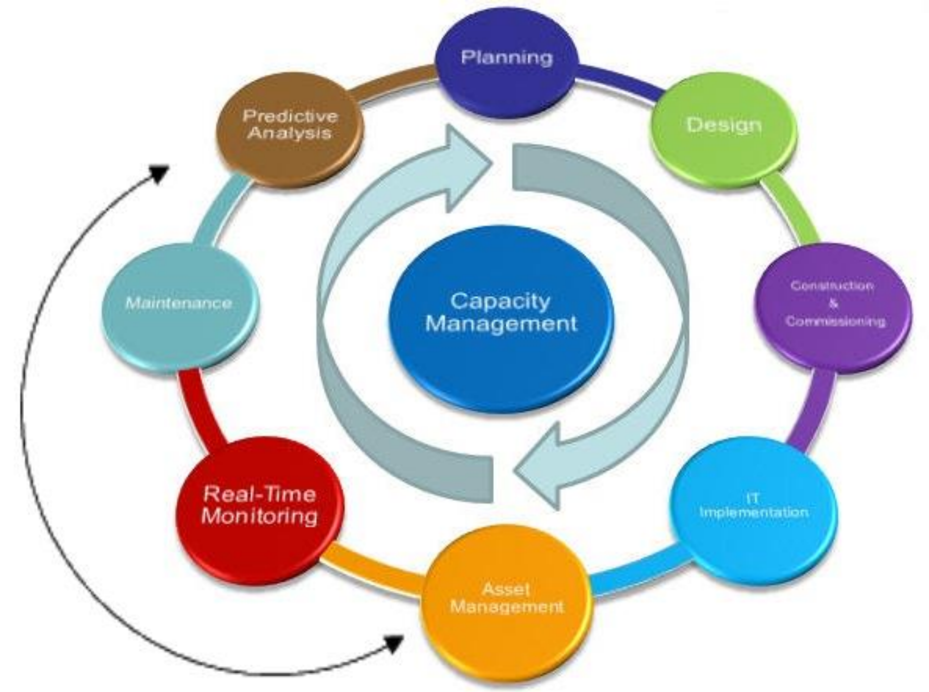
COMPARISION

innovate achieve lead

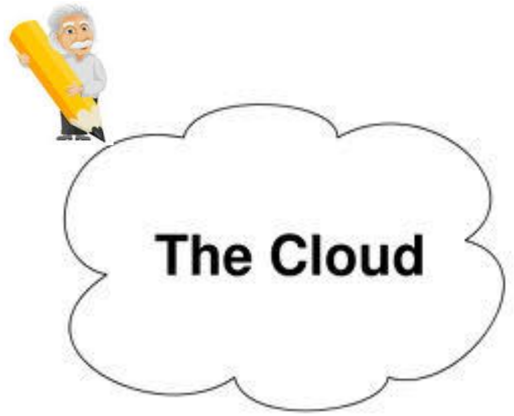
BITS Pilani

Pilani|Dubai|Goa|Hyderabad

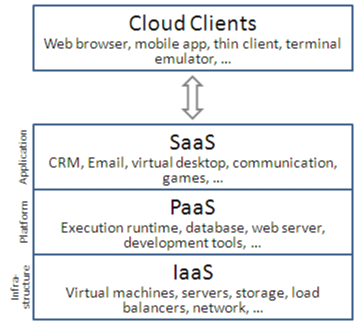
Capacity Management

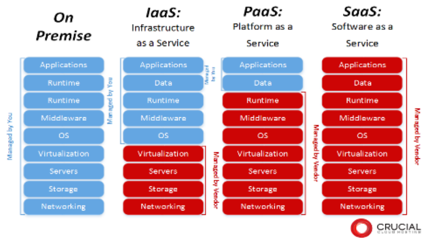


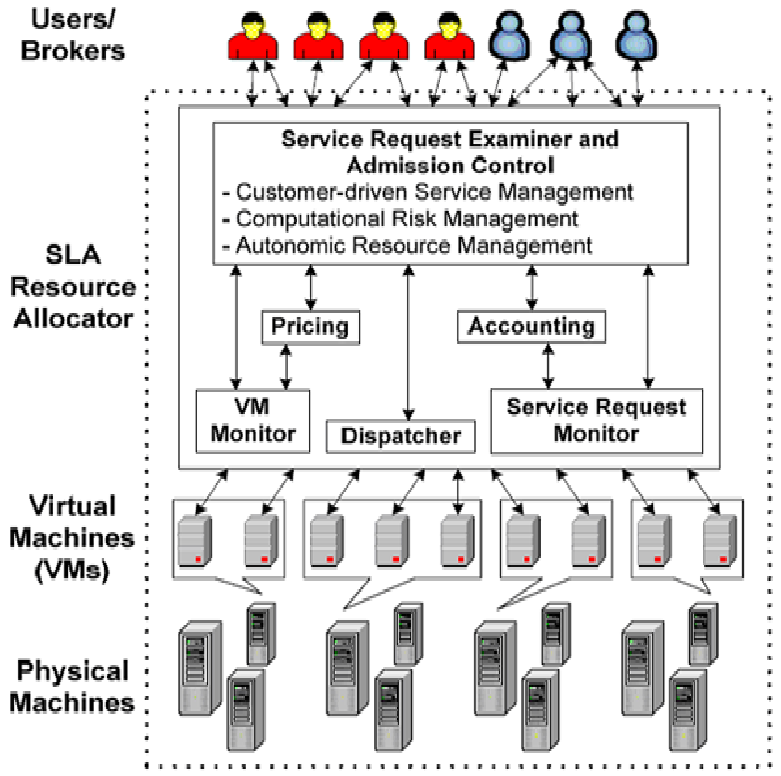
Anatomy of a Cloud Ecosystem



Is it So?????


Cloud Clients
Web browser, mobile app, thin client, terminal emulator, ...
SaaS
CRM, Email, virtual desktop, communication, games, ...
PaaS
Execution runtime, database, web server, development tools, ...
IaaS
Virtual machines, servers, storage, load balancers, network, ...
Application
Platform
Infra-structure




Users/ Brokers
SLA Resource Allocator
Service Request Examiner and Admission Control
- Customer-driven Service Management
- Computational Risk Management
- Autonomic Resource Management
Pricing
Accounting
VM Monitor
Dispatcher
Service Request Monitor
Virtual Machines (VMs)
Physical Machines

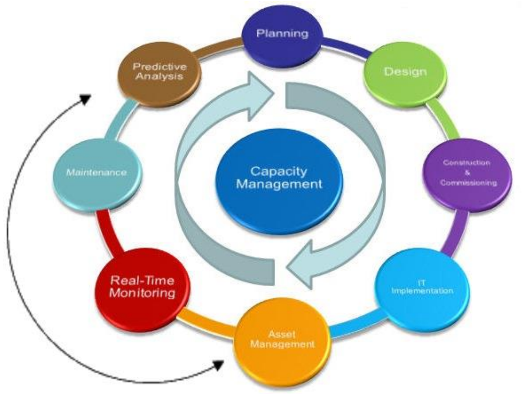
CLOUD requires managing…… But what will you manage?

- Cloud **<u>Distributed environment</u>**
    - With large scale of systems to manage
    - Support of multi-tenancy
    - Management to maintain SLAs

Why Capacity Management in Cloud?

Capacity Management is

- Strategic and Proactive
- Assists in managing service quality
- Assists in managing cost expenditures
- Aligns business and IT
- Match needs and cost during growth



Following are **five characteristics of Cloud**

•They provide on-demand provisioning of computational resources;

- They use virtualization technologies to lease these resources;
- They provide public and simple remote interfaces to manage those resources;
- They use a pay-as-you-go cost model, typically charging by the hour;
- They operate data centers large enough to provide a seemingly unlimited number of resources to their clients

What are the Challenges?

Capacity on Demand- Too Little, Too Much or Just Right?

Automation and Control

Security Protecting your loved ones‟

Scalability- In / Up or Out?

Old Habits – The Potential Risks

Virtual Infrastructure (VI) management—the management of virtual machines distributed across a pool of physical resources—becomes a key concern when building an IaaS cloud and **poses a number of challenges**.

- In traditional physical resources, virtual machines require a fair amount of configuration, including preparation of the machine's software environment and network configuration.
- In a virtual infrastructure, this configuration must be done on-the-fly, with as little time between the time the VMs are requested and the time they are available to the users.
- This is further complicated by the need to configure groups of VMs that will provide a specific service (e.g., an application requiring a Web server and a database server).
- Additionally, a virtual infrastructure manager must be capable of allocating resources efficiently, taking into account an organization's goals (such as minimizing power consumption and other operational costs) and reacting to changes in the physical infrastructure.

Importance of Capacity Management

When **Capacity** is **Too Little** (under capacity)

- Application Sizing not performed
- Controlled by limits
- Lower usage costs
- Service Levels affected – by how much?
- High impact on business?
  - Loss of service
  - SLA breach penalties

When **Capacity** is **Too Much** (over capacity)

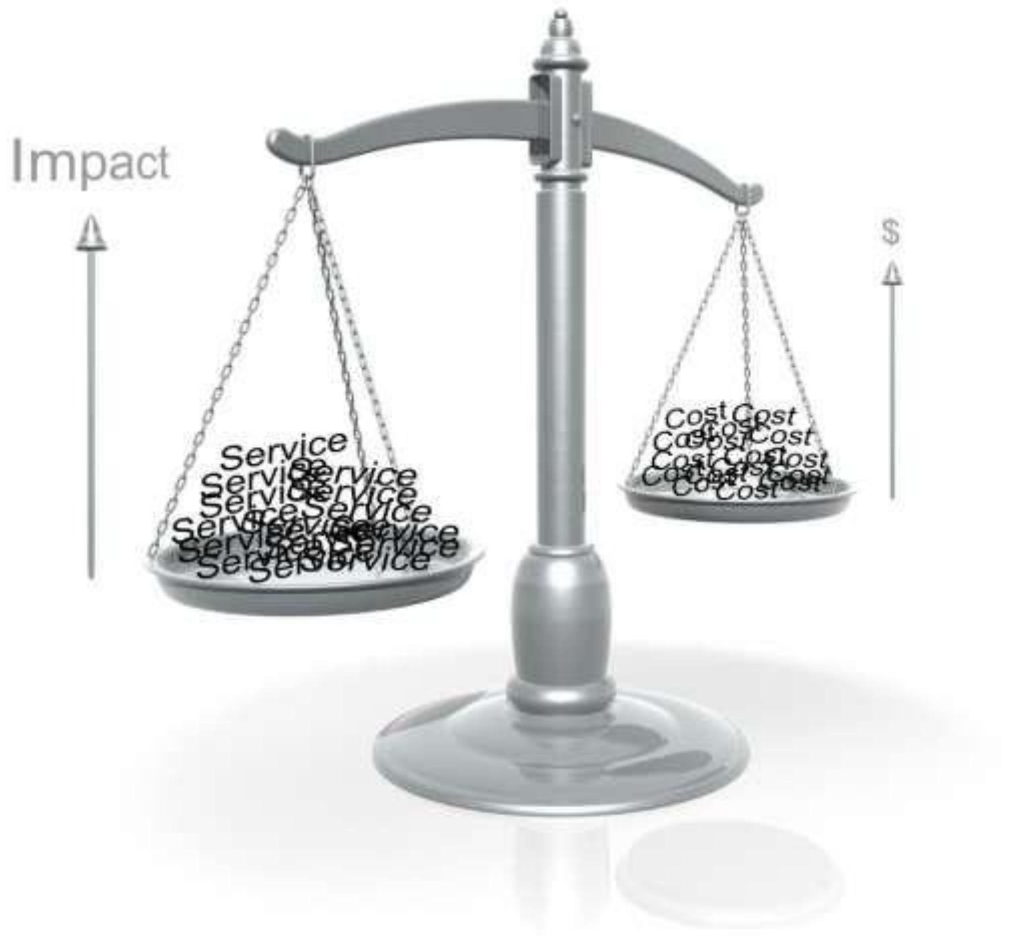
- Unlimited access to resources
- Service Levels unaffected
- Costs – higher resource usage, software licensing
- Impacts on other services
  - Poor performance
  - Wasted resources (multi virtual CPU (*vSMP*) & single-threaded applications)
  - VM Sprawl
  - Increased pressure to manage virtual resources

- **Gartner – *"most organizations overprovision their infrastructure by at least 100%"***

So What's the Way? – Find the Right Balance

We need to have the **Right Capacity**?

- Application Sizing performed
- Efficient use of IaaS
- Acceptable Service Levels
  - continuous review and adjustment
- Controlled by shares, limits and reservations
- Continuous monitoring and tuning
  - Configuration adjustments (CPU, Memory)
  - VM consolidation
  - Power off unused ESX hosts
- Right Balance
  - Find the equilibrium (service / cost)



How To ? – Find the Right Balance

Automation and Control

- Rapid Elasticity- Guest Migration and Portability, Templates, Golden Host
- Resource Pools (limits, shares and reservations)
- Load Balancing
  - DRS (Automatic, Partial or Manual)
- Affinity
  - VM to VM or VM to Host
  - CPU
    - Fine grained tweaking for performance gains – Single Threaded Apps
    - Licensing

Protecting your loved ones"

- Critical business applications
- Service Levels must be met
- Highest Priority (shares, unlimited), CPU affinity
- Must guarantee resources (reservations)
- High Availability Clusters
  - VMware - Fault Tolerance
- Trade offs
  - "just in case" capacity management could impact on other services
  - Significant impact? Think about scaling out or up .

How To ? – Find the Right Balance

Old Habits – Potential Risks

- Gartner – "*Through 2015, more than 70% of private cloud implementations will fail to deliver operational, energy and environmental efficiencies.*"
- Infrastructure or application hugging
  - Silo mentality "what's mine is mine"
- Lack of resource sharing
  - Through lack of trust and confidence
- "Just in case" capacity planning
  - Leads to over provisioning

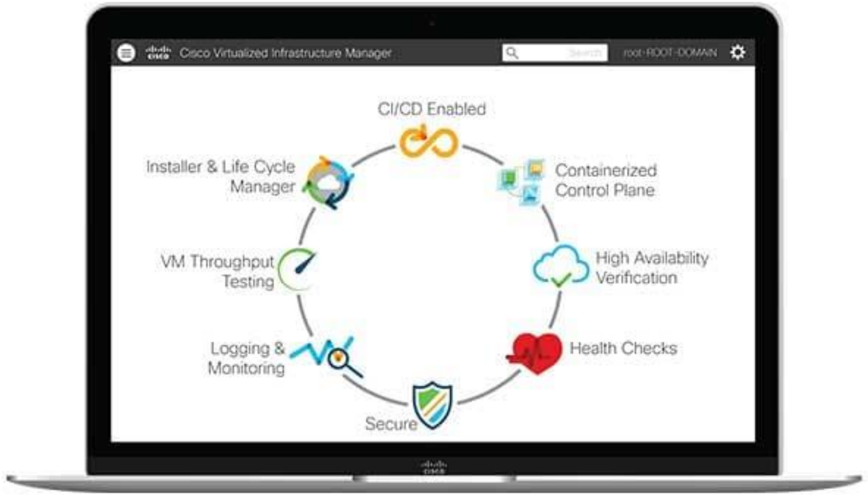
What is Virtual Infrastructure Management

Virtual Infrastructure (VI) management—the management of virtual machines distributed across a pool of physical resources—becomes a key concern when building an IaaS cloud and **poses a number of challenges**.

- In traditional physical resources, virtual machines require a fair amount of configuration, including preparation of the machine's software environment and network configuration.
- In a virtual infrastructure, this configuration must be done on-the-fly, with as little time between the time the VMs are requested and the time they are available to the users.

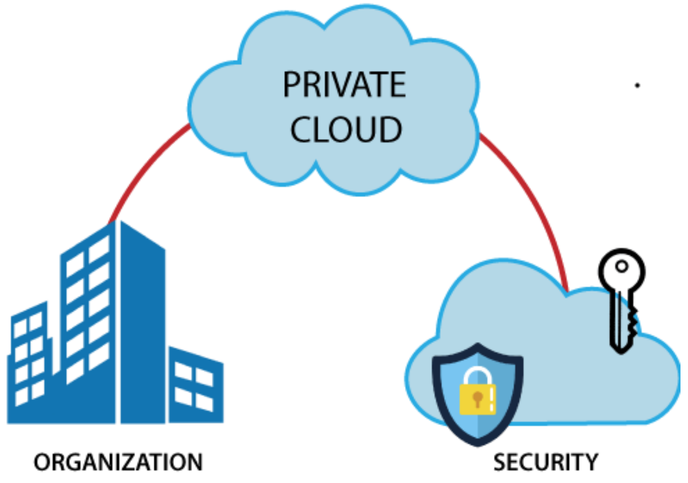
Virtual Infrastructure Management

- This is further complicated by the need to configure groups of VMs that will provide a specific service (e.g., an application requiring a Web server and a database server).
- Additionally, a virtual infrastructure manager must be capable of allocating resources efficiently, taking into account an organization’s goals (such as minimizing power consumption and other operational costs) and reacting to changes in the physical infrastructure.


Cisco Virtualized Infrastructure Manager
CI/CD Enabled
Containerized Control Plane
High Availability Verification
Health Checks
Secure
Logging & Monitoring
VM Throughput Testing
Installer & Life Cycle Manager

Virtual Infrastructure Management

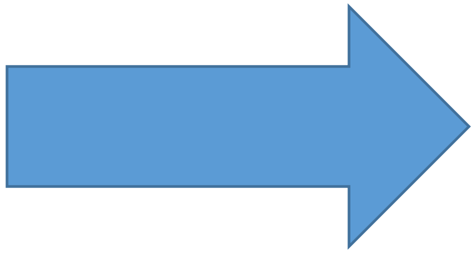
- Virtual infrastructure management in **private clouds** has to deal with an additional problem:
- **Unlike** large IaaS cloud providers such as Amazon, private clouds typically do not have enough resources to provide the illusion of "infinite capacity."
- The immediate provisioning scheme used in public clouds, where resources are provisioned at the moment they are requested, is **ineffective in private clouds**.



So What is the way out?

Managing virtual **infrastructures in a private/hybrid cloud is a different problem** than **managing a virtualized data center**, and existing tools lack several features that are required for building IaaS clouds.

- **Traditional methods** can only operate with some preconfigured placement policies, which are generally simple (round robin, first fit, etc.) and based only on CPU speed and utilization of a fixed and predetermined number of resources, such as memory and network bandwidth.
- Thus, there are still several gaps in existing **VI solutions**. Filling these gaps will require **addressing a number of research challenges** such as **virtual machine management**, **resource scheduling**, **SLAs**, **federation of resources**, and **security**.

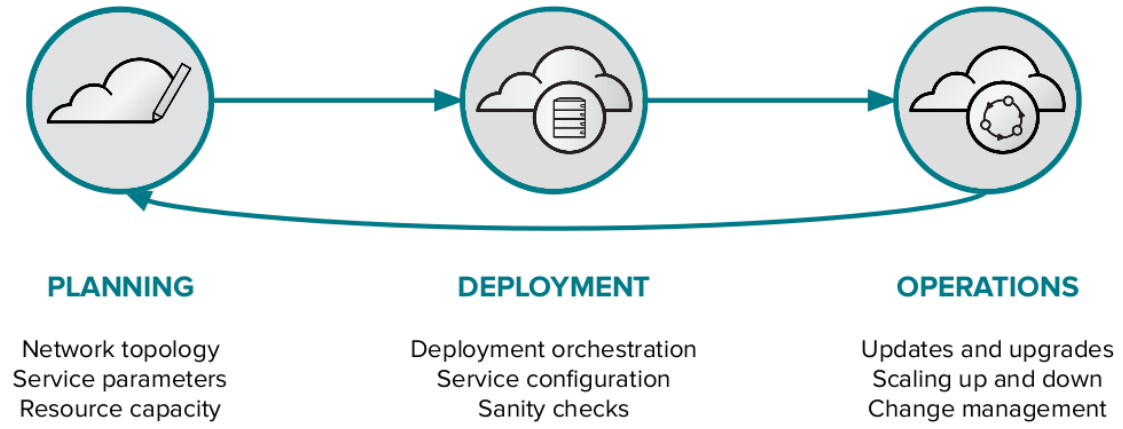


VIM

BITS Pilani

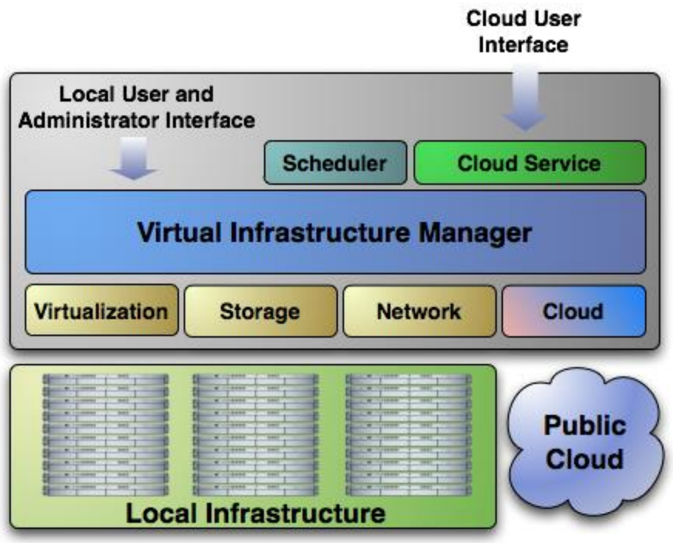
Pilani|Dubai|Goa|Hyderabad

Virtual Infrastructure Manager (VIM)



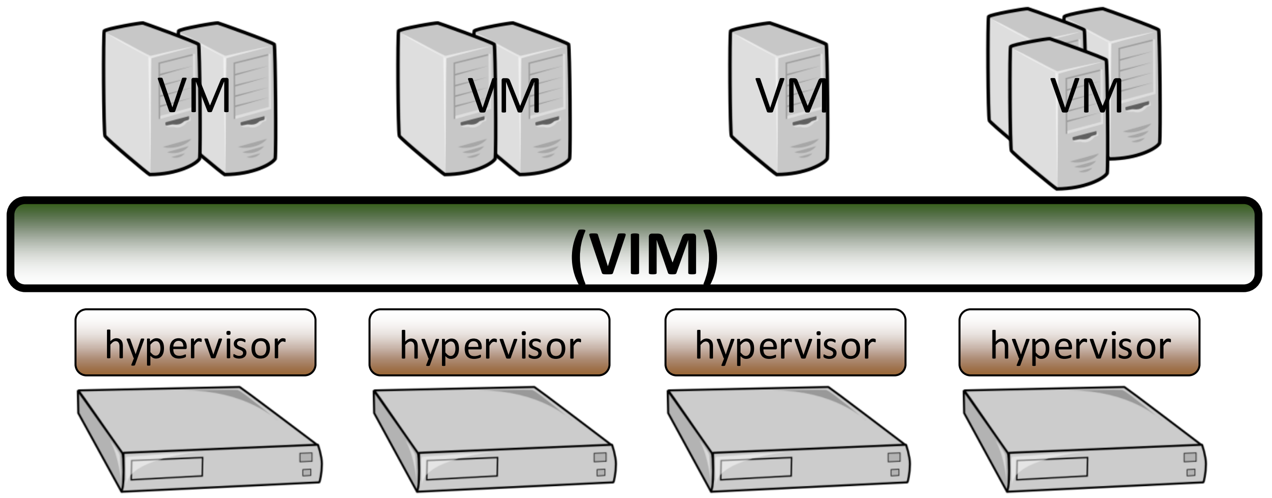
What is a Virtual Infrastructure Manager?

- A VIM runs on top of a hypervisor in a virtualized environment. The hypervisor allocates and manages virtual machines. The VIM deals with the allocation of resources in the virtual infrastructure. They include computational resources (processors), storage, and network resources. Virtual infrastructure management allows the allocation to happen based on current requirements, rather than being statically allocated.
- The VIM carries out several tasks:
- Allocating resources in accordance with traffic engineering rules.
- Support for defining operational rules.
- Definition of hub-to-facility mapping.
- Providing information for provisioning virtual infrastructure orchestration (VIO).



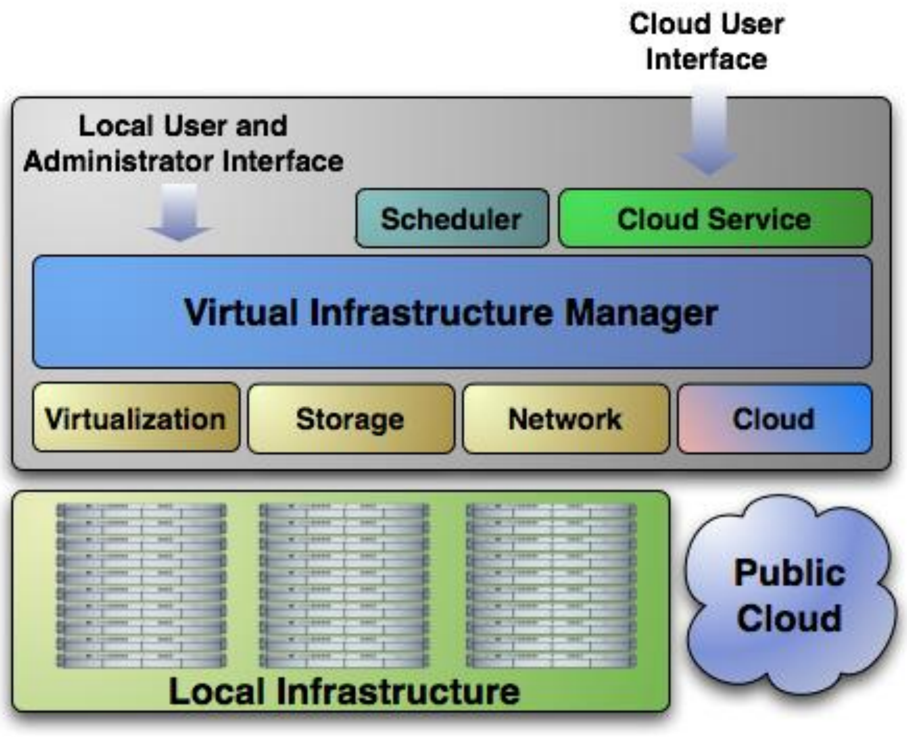
Why a Virtual Infrastructure Manager?

- VMs are great!!...but something more is needed
  - Where did/do I put my VM? (***scheduling & monitoring***)
  - How do I provision a new cluster node? (***clone***)
  - What IP addresses are available? (***networking***)
- Provide a ***uniform view*** of the resource pool
- ***Life-cycle management*** and monitoring of VM
- The VIM should ***integrate*** Image, Network and Virtualization



Why a Virtual Infrastructure Manager?

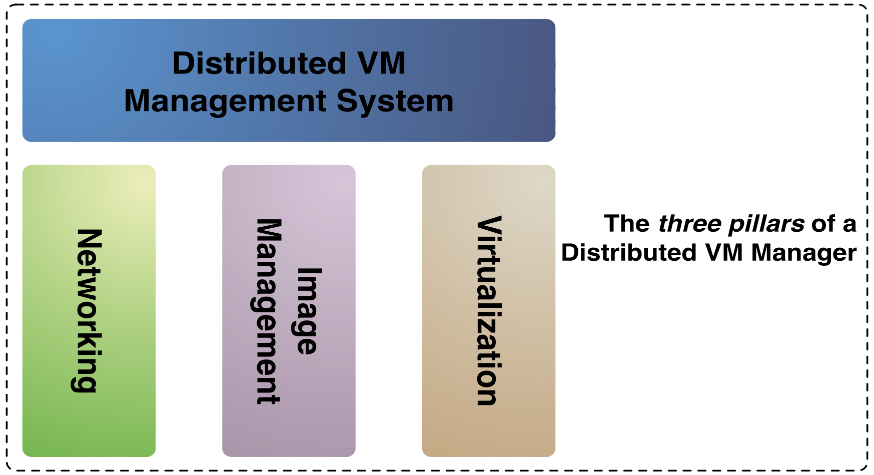
- Dynamic deployment and re-placement of virtual machines on a pool of physical resources
- Transform a rigid distributed physical infrastructure into a flexible and agile virtual infrastructure



- Backend of Public Cloud: Internal management of the infrastructure
- Private Cloud: Virtualization of cluster or data-center for internal users
- Cloud Interoperation: On-demand access to public clouds

Enter – Distributed Management of Virtual Resources

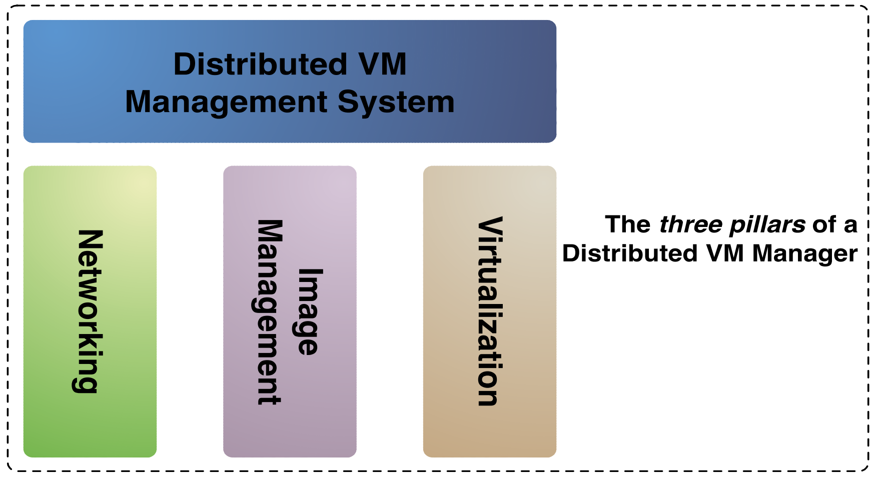
- The cloud ecosystem is inherently distributed. Resources are spread around.
- Location also plays an important role in the efficient selection / optimal selection & placement of resources.


Distributed VM Management System
Networking
Image Management
Virtualization
The *three pillars* of a Distributed VM Manager

- The problem of efficiently selecting or scheduling computational resources is well known. However, the state of the art in **VM-based resource scheduling** follows a **static approach**, where resources are initially selected using a greedy allocation strategy, with minimal or no support for other placement policies.
- To efficiently schedule resources, **VI managers** must be able to support **flexible and complex scheduling policies** and must *leverage* (use) the ability of VMs to suspend, resume, and migrate.

So What is the Solution

- Managing VMs in a pool of distributed physical resources is a key concern in IaaS clouds, requiring the use of a virtual infrastructure manager.
- **OpenNebula** is capable of managing groups of interconnected VMs—with support for the Xen, KVM, and VMWare platforms—within data centers and private clouds that involve a large amount of virtual and physical servers.
- **OpenNebula** can also be used to build hybrid clouds by interfacing with remote cloud sites.

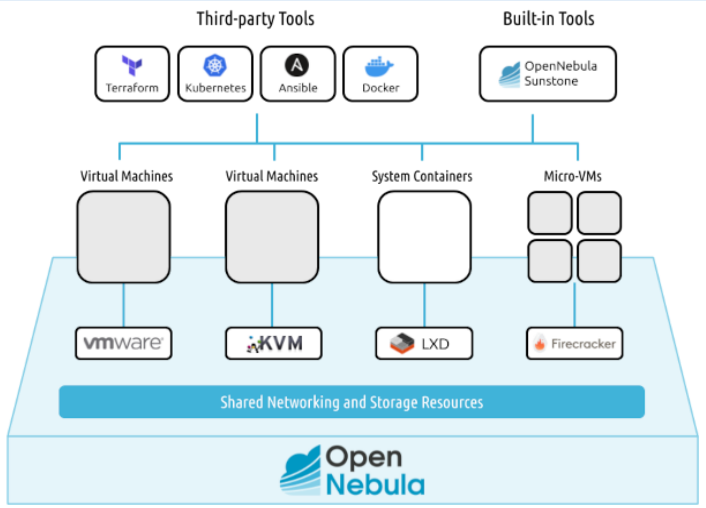

Distributed VM Management System
Networking
Image Management
Virtualization
The *three pillars* of a Distributed VM Manager

innovate achieve lead

BITS Pilani

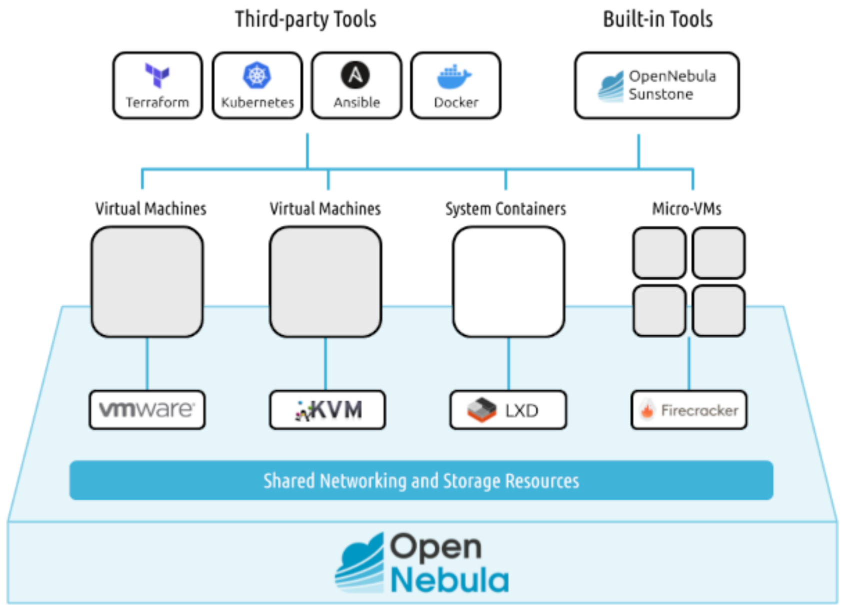
Pilani|Dubai|Goa|Hyderabad

OpenNebula VIM



What is OpenNebula

- *OpenNebula is a simple, feature-rich and flexible solution for the management of virtualized data centers.*
- *It enables private, public and hybrid clouds. Here are a few facts about this solution.*
- OpenNebula is an open source cloud middleware solution that manages heterogeneous distributed data center infrastructures.
- It is designed to be a simple but feature-rich, production-ready, customizable solution to build and manage enterprise clouds—simple to install, update and operate by the administrators; and simple to use by end users.
- OpenNebula combines existing virtualization technologies with advanced features for multi-tenancy, automated provisioning and elasticity.
- A built-in virtual network manager maps virtual networks to physical networks.
- Distributions such as Ubuntu and Red Hat Enterprise Linux have already integrated OpenNebula.
- OpenNebula supports Xen, KVM and VMware hypervisors.

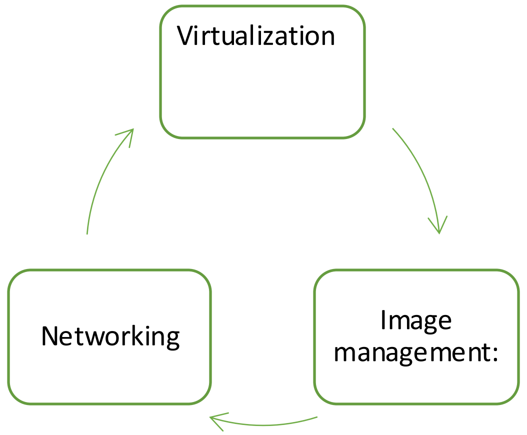


- **Private Cloud** to simplify and optimize internal operations
- Hybrid Cloud to supplement the capacity of the Private Cloud
- Public Cloud to expose your Private to external users

Stages of VM Life Cycle in OpenNebula

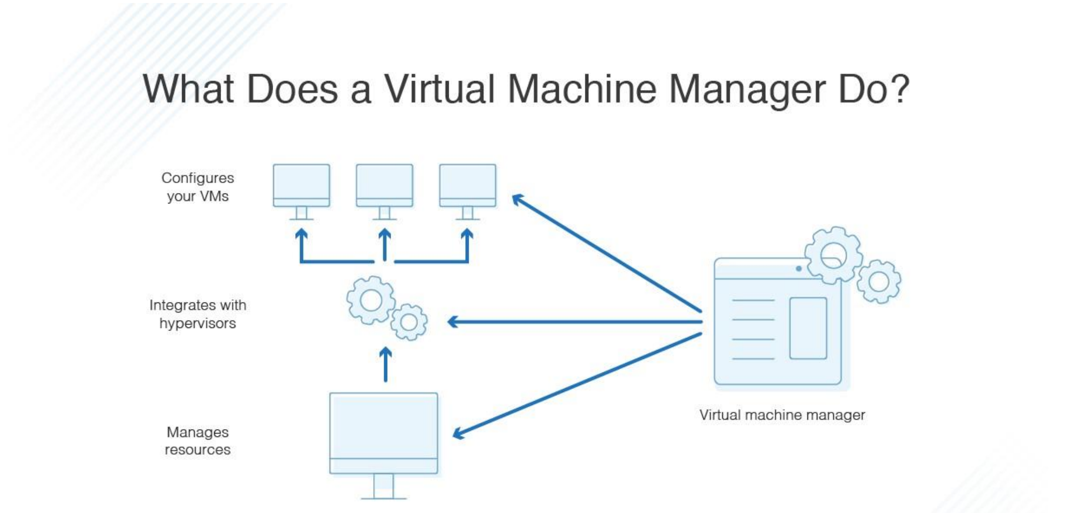
- **The life cycle of a VM within OpenNebula follows several stages:**
  - **Resource Selection**: allowing site administrators to configure the scheduler to prioritize the resources that are more suitable for the VM
  - **Resource Preparation:** The disk images of the VM are transferred to the target physical resource. During the boot process, the VM is contextualized, a process where the disk images are specialized to work in a given environment.
  - **VM Creation**: The VM is booted by the resource hypervisor
  - **VM Migration**: The VM potentially gets migrated to a more suitable resource(e.g.,to optimizethe power consumption ofthe physical resources)
  - **VM Termination**: When the VM is going to shut down, OpenNebula can transfer back its disk images to a known location. This way, changes in the VM can be kept for a future use.

- **Within OpenNebula, a VM is modeled as having the following attributes:**
  - A capacity in terms of memory and CPU
  - A set of NICs attached to one or more virtual networks
  - A set of disk images
  - A state file (optional) or recovery file



VM Management in OpenNebula

- OpenNebula manages VMs by interfacing with a physical resource's hypervisor, such as Xen, KVM, or VMWare, **Hyper-V** to control (e.g., boot, stop, or shutdown) the VM;
- using a set of **pluggable drivers** that decouple the managing process from the underlying technology.
- Thus, whenever the core needs to manage a VM, it uses **high-level commands** such as "start VM," "stop VM," and so on, which are translated by the drivers into commands that the virtual machine manager can understand. By decoupling the OpenNebula core from the virtualization technologies through the use of a **driver-based architecture**, adding support for additional virtual machine managers only requires writing a driver for it.


What Does a Virtual Machine Manager Do?
Configures your VMs
Integrates with hypervisors
Manages resources
Virtual machine manager

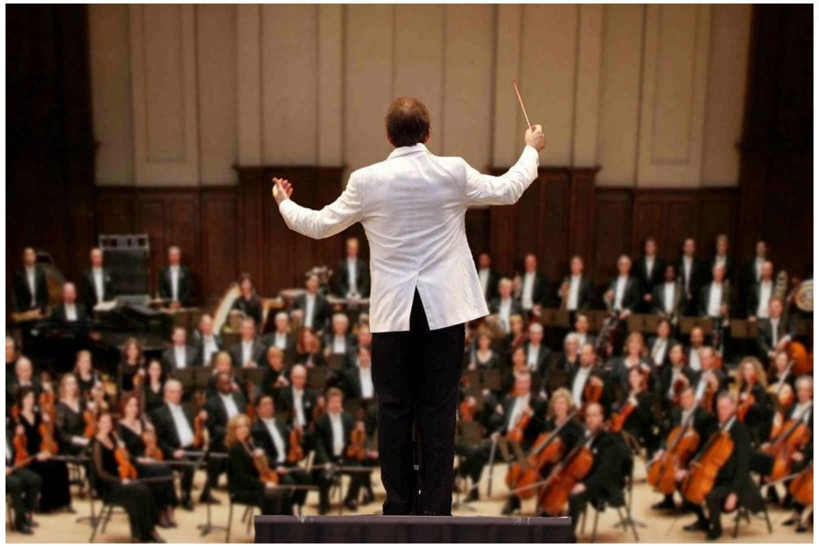
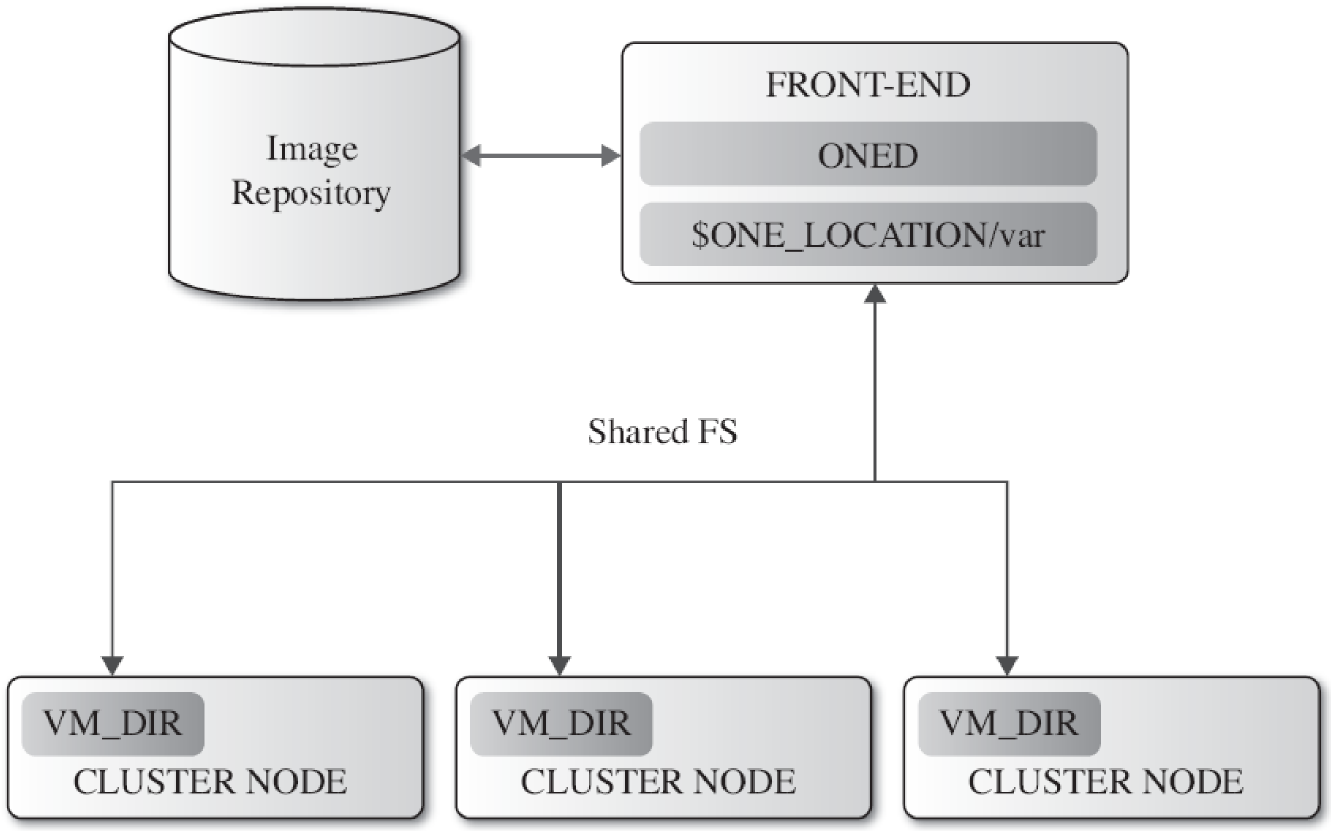


Image Management in OpenNebula

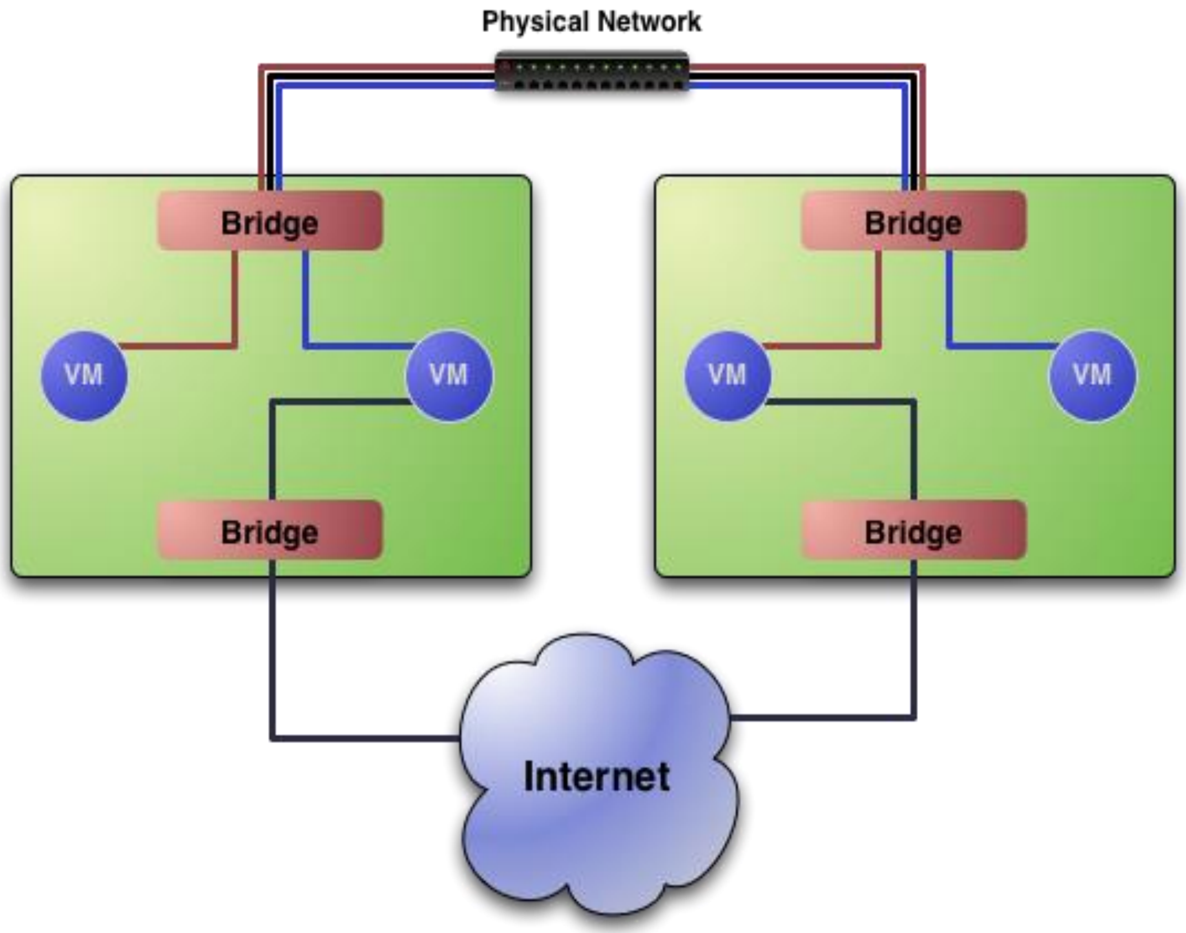
- Transferring the VM images from an image repository to the selected resource and by creating on-the-fly temporary images
- What is image?
  - Virtual disk contains the OS and other additional software
- Image management model



Networking OpenNebula

- In general, services deployed on a cloud require several interrelated VMs, with a virtual application network (VAN) being the primary link between them.

- OpenNebula dynamically creates these VANs and tracks the MAC addresses leased in the network to the service VMs.
- **physical cluster** as a set of hosts with one or more network interfaces, each of them connected to a different physical network.



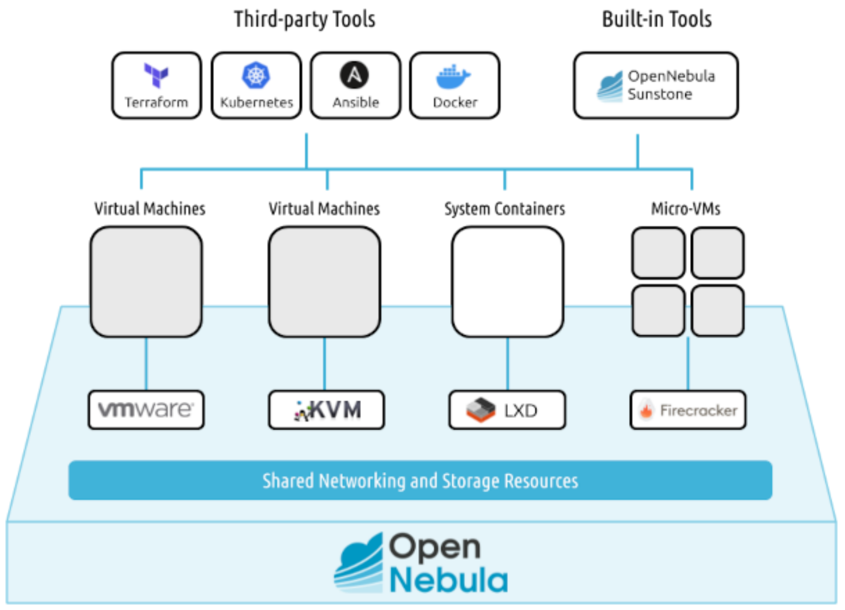
Benefits of OpenNebula

For the Infrastructure Manager

- Centralized management of VM workload and distributed infrastructures
- Support for VM placement policies: balance of workload, server consolidation…
- Dynamic resizing of the infrastructure
- Dynamic partition and isolation of clusters
- Dynamic scaling of private infrastructure to meet fluctuating demands
- Lower infrastructure expenses combining local and remote Cloud resources

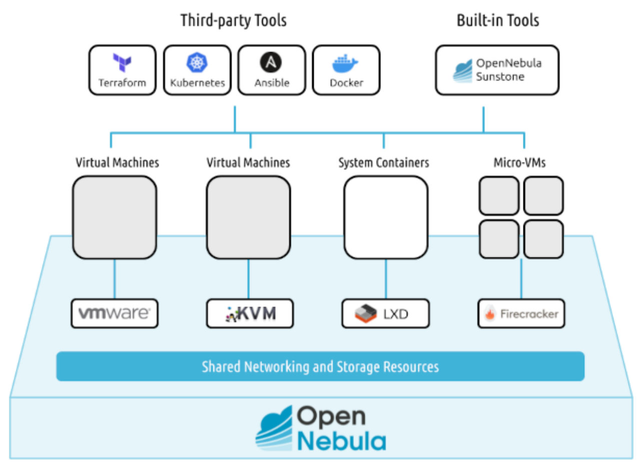
For the Infrastructure User

- Faster delivery and scalability of services
- Support for heterogeneous execution environments
- Full control of the lifecycle of virtualized services management



Features of OpenNebula

Feature	Function
Internal Interface	• Unix-like CLI for fully management of VM life-cycle and physical boxes • XML-RPC API and libvirt virtualization API
Scheduler	• Requirement/rank matchmaker allowing the definition of workload and resource-aware allocation policies • Support for advance reservation of capacity through Haizea
Virtualization Management	• Xen, KVM, and VMware • Generic libvirt connector (VirtualBox planned for 1.4.2)
Image Management	• General mechanisms to transfer and clone VM images
Network Management	• Definition of isolated virtual networks to interconnect VMs
Service Management and Contextualization	• Support for multi-tier services consisting of groups of inter-connected VMs, and their auto-configuration at boot time
Security	• Management of users by the infrastructure administrator
Fault Tolerance	• Persistent database backend to store host and VM information
Scalability	• Tested in the management of medium scale infrastructures with hundreds of servers and VMs (no scalability issues has been reported)
Flexibility and Extensibility	• Open, flexible and extensible architecture, interfaces and components, allowing its integration with any product or tool



Comparison with Similar Technologies

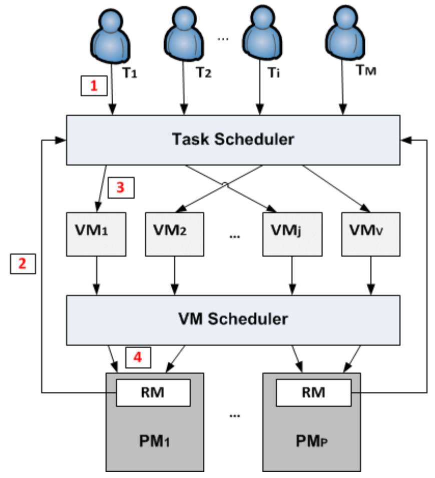
	Platform ISF	VMware Vsphere	Eucalyptus	Nimbus	OpenNebula
Virtualization Management	**VMware, Xen**	**VMware**	**Xen, KVM**	**Xen**	**Xen, KVM, VMware**
Virtual Network Management	**Yes**	**Yes**	No	**Yes**	**Yes**
Image Management	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Service Contextualizatio n	No	No	No	**Yes**	**Yes**
Scheduling	**Yes**	**Yes**	No	No	**Yes**
Administration Interface	**Yes**	**Yes**	No	No	**Yes**
Hybrid Cloud Computing	No	No	No	No	**Yes**
Cloud Interfaces	No	**vCloud**	**EC2**	**WSRF, EC2**	**EC2 Query, OGF OCCI**
Flexibility and Extensibility	**Yes**	No	**Yes**	**Yes**	**Yes**
Open Source	No	No	**GPL**	**Apache**	**Apache**

innovate achieve lead

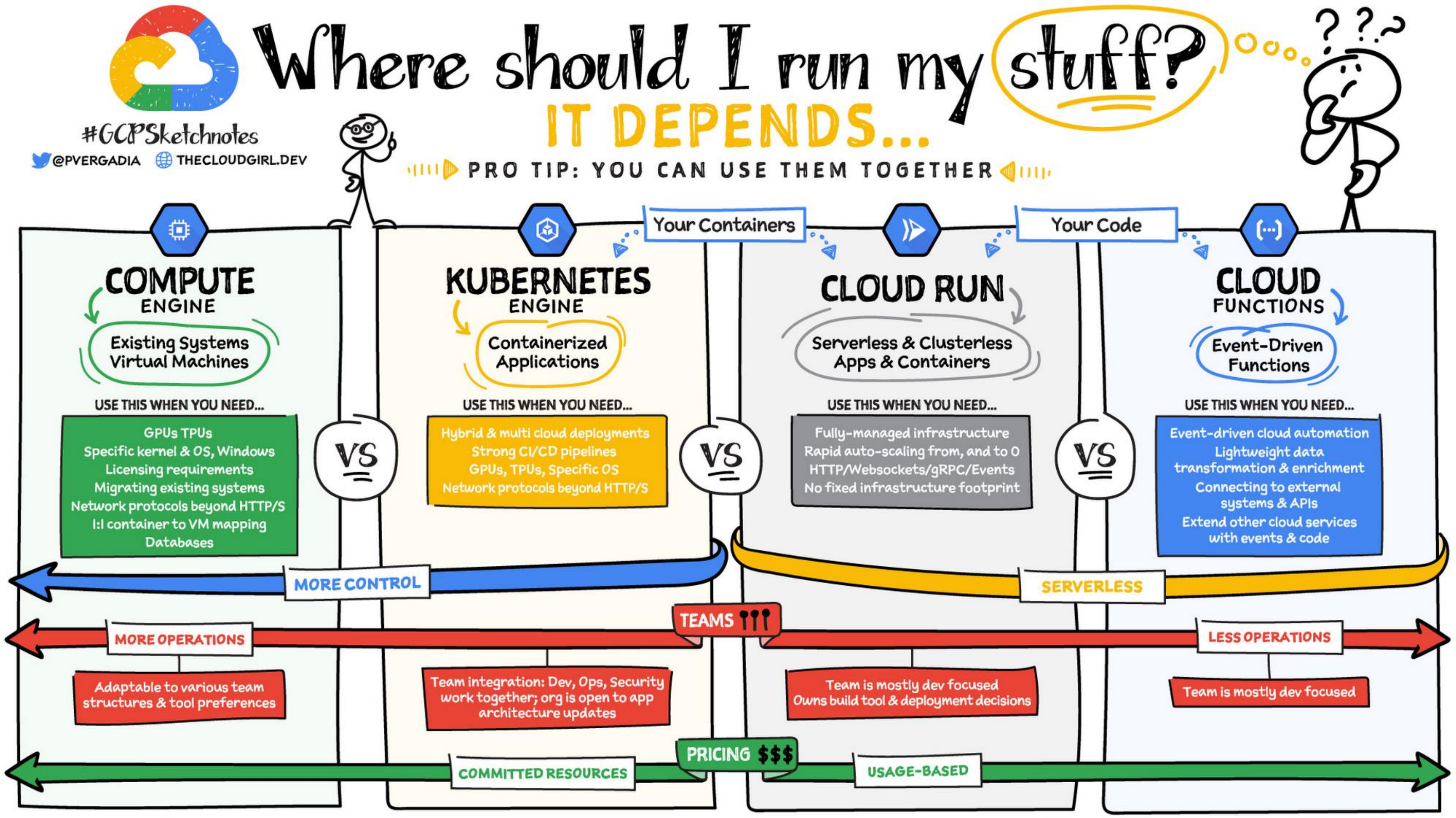
BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Scheduling VM Workloads



What is a Cloud Workload



COMPUTE ENGINE – Existing Systems Virtual Machines

USE THIS WHEN YOU NEED...
- GPUs TPUs
- Specific kernel & OS, Windows
- Licensing requirements
- Migrating existing systems
- Network protocols beyond HTTP/S
- 1:1 container to VM mapping
- Databases

Adaptable to various team structures & tool preferences

KUBERNETES ENGINE – Containerized Applications

USE THIS WHEN YOU NEED...
- Hybrid & multi cloud deployments
- Strong CI/CD pipelines
- GPUs, TPUs, Specific OS
- Network protocols beyond HTTP/S

Team integration: Dev, Ops, Security work together; org is open to app architecture updates

CLOUD RUN – Serverless & Clusterless Apps & Containers

USE THIS WHEN YOU NEED...
- Fully-managed infrastructure
- Rapid auto-scaling from, and to 0
- HTTP/Websockets/gRPC/Events
- No fixed infrastructure footprint

Team is mostly dev focused
Owns build tool & deployment decisions

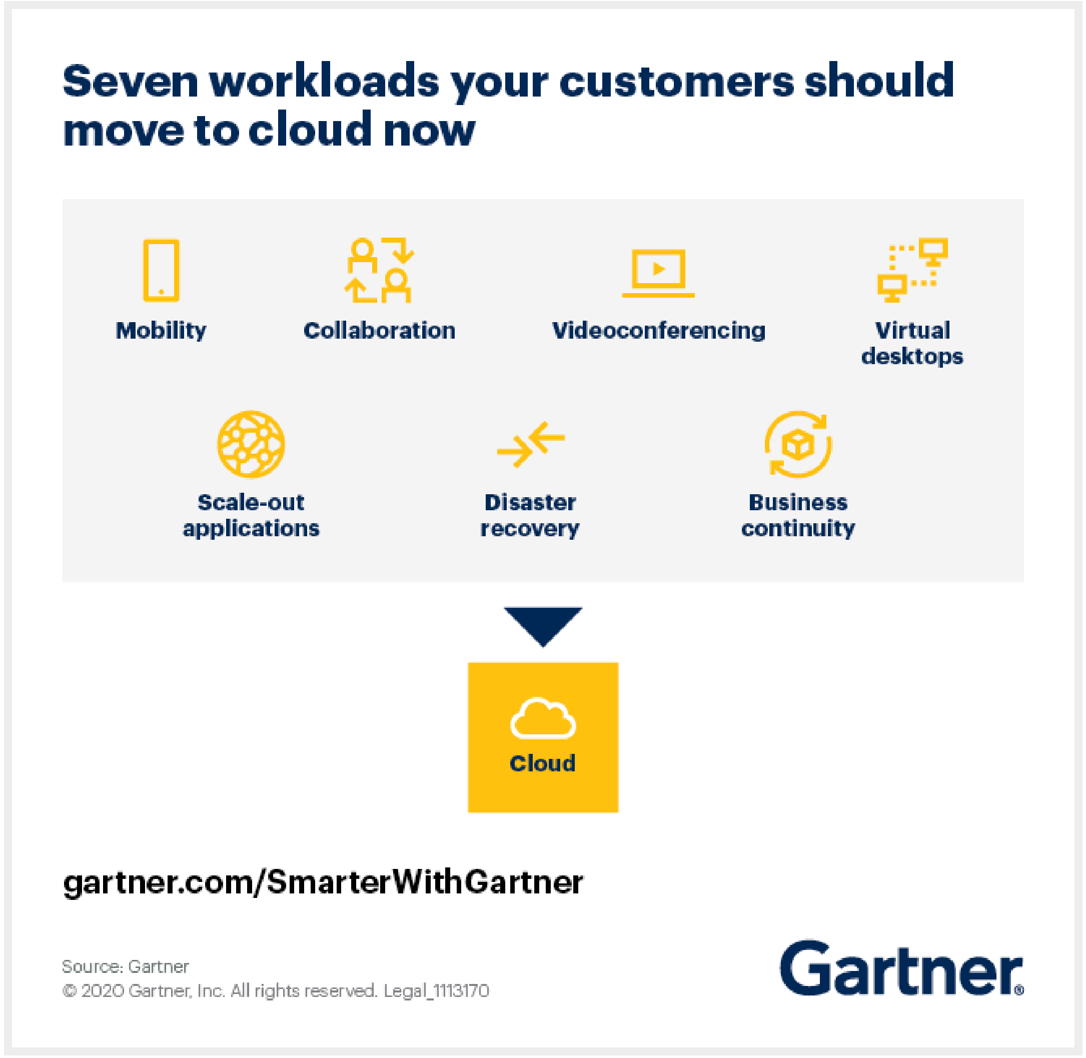
CLOUD FUNCTIONS – Event-Driven Functions

USE THIS WHEN YOU NEED...
- Event-driven cloud automation
- Lightweight data transformation & enrichment
- Connecting to external systems & APIs
- Extend other cloud services with events & code

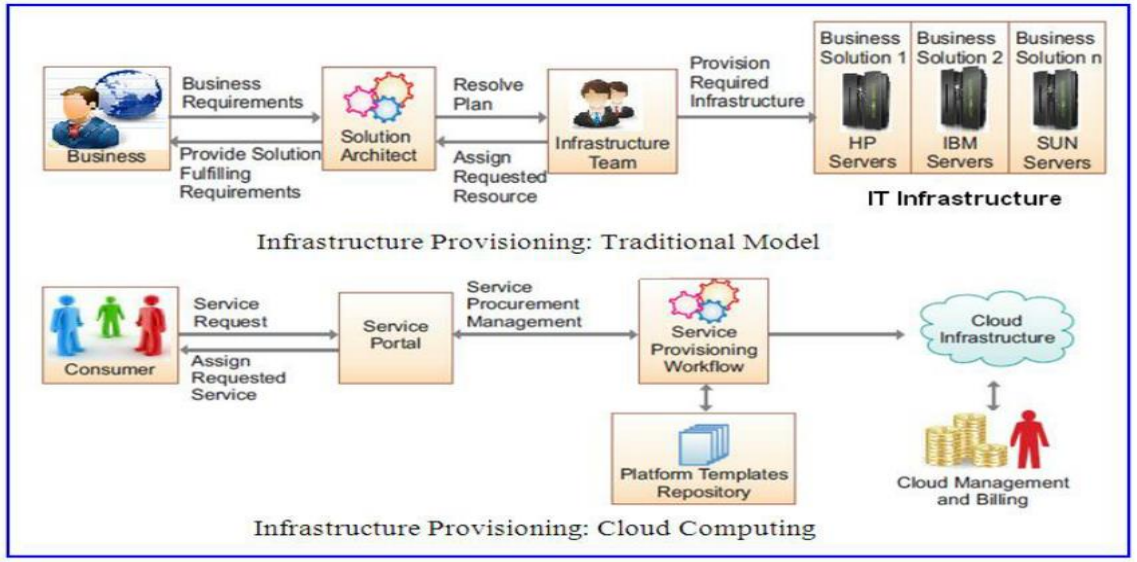
Team is mostly dev focused

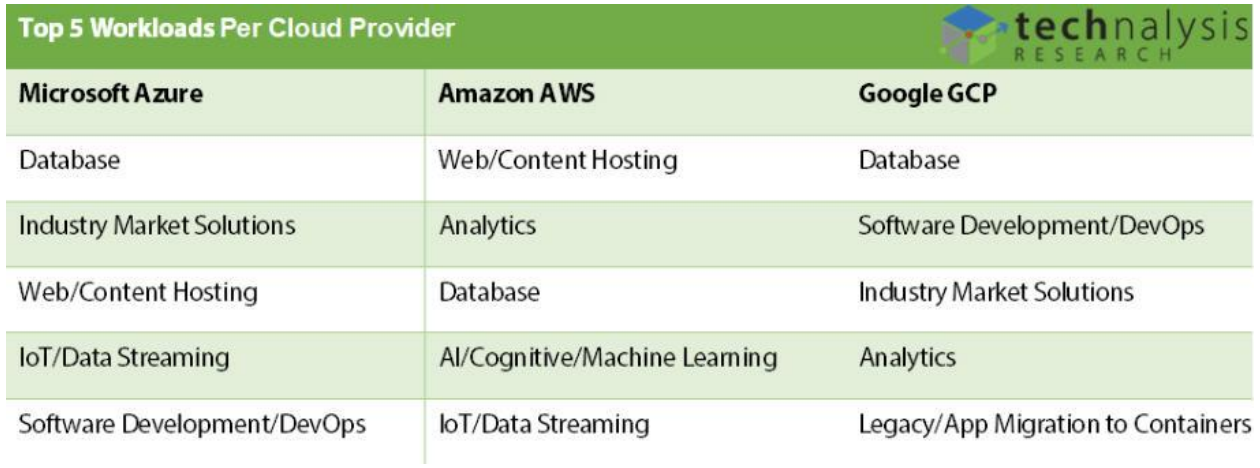
What is a Cloud Workload

- A cloud workload is a specific application, service, capability or a specific amount of work that can be run on a cloud resource. Virtual machines, databases, containers, Hadoop nodes and applications are all considered cloud workloads.
- A workload is **a collection of resources and code that delivers business value, such as a customer-facing application or a backend process**. A workload might consist of a subset of resources in a single cloud account or be a collection of multiple resources spanning multiple cloud accounts.
- Examples of workloads are **marketing websites, e-commerce websites, the back-ends for a mobile app, analytic platforms**, etc. Workloads vary in levels of architectural complexity, from static websites to architectures with multiple data stores and many components.


Seven workloads your customers should move to cloud now
Mobility
Collaboration
Videoconferencing
Virtual desktops
Scale-out applications
Disaster recovery
Business continuity
Cloud
gartner.com/SmarterWithGartner
Source: Gartner
© 2020 Gartner, Inc. All rights reserved. Legal_1113170
Gartner

What are we Talking about


Business
Business Requirements
Provide Solution Fulfilling Requirements
Solution Architect
Resolve Plan
Assign Requested Resource
Infrastructure Team
Provision Required Infrastructure
Business Solution 1
Business Solution 2
Business Solution n
HP Servers
IBM Servers
SUN Servers
IT Infrastructure
Infrastructure Provisioning: Traditional Model
Consumer
Service Request
Assign Requested Service
Service Portal
Service Procurement Management
Service Provisioning Workflow
Cloud Infrastructure
Platform Templates Repository
Cloud Management and Billing
Infrastructure Provisioning: Cloud Computing



Top 5 Workloads Per Cloud Provider — technalysis RESEARCH

Microsoft Azure	Amazon AWS	Google GCP
Database	Web/Content Hosting	Database
Industry Market Solutions	Analytics	Software Development/DevOps
Web/Content Hosting	Database	Industry Market Solutions
IoT/Data Streaming	AI/Cognitive/Machine Learning	Analytics
Software Development/DevOps	IoT/Data Streaming	Legacy/App Migration to Containers

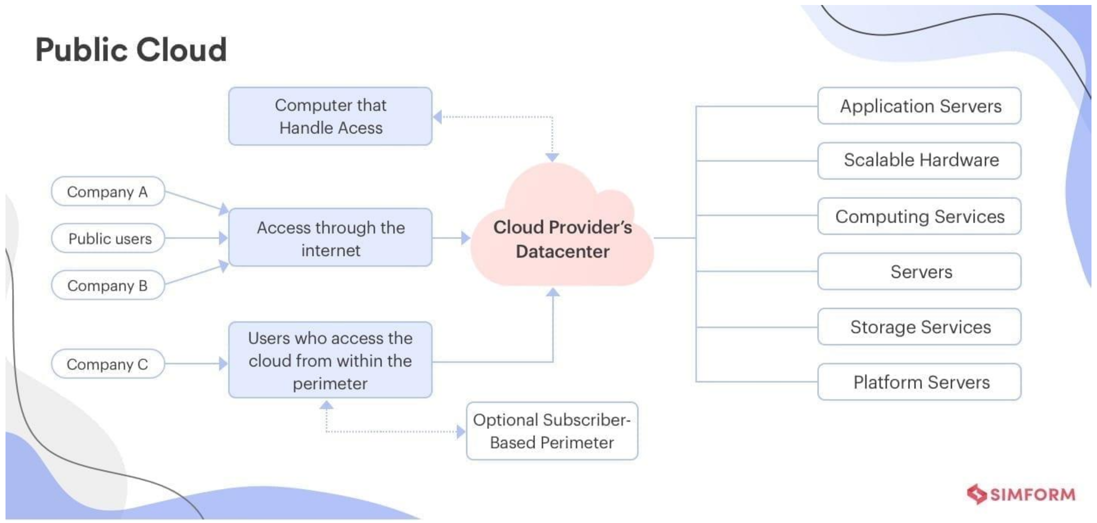
Workload ➔ Amount of work (or load) that software imposes on the underlying computing resources.

Workload ➔ Amount of time and computing resources required to perform a specific task or produce an output from inputs provided.

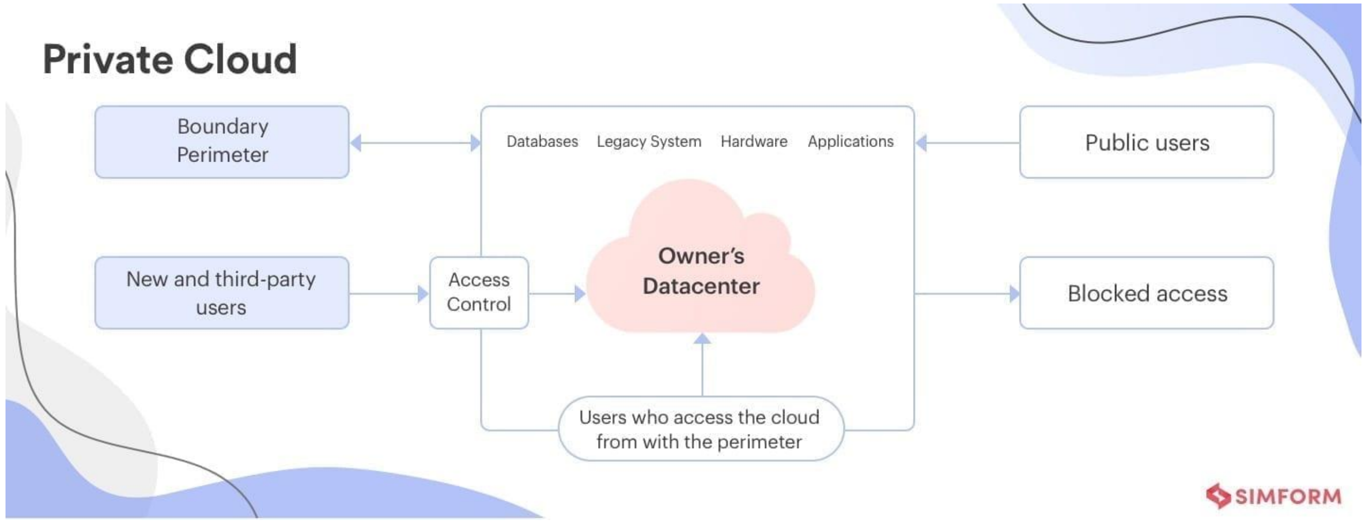
Types of Workload ➔Static (fixed work always) Dynamic (Ad-hoc requests) Analytical (Big Data) Transactional (Main Frame)

Standardized metrics used to measure and report on an application's performance or load are collectively referred to as *benchmarks*.

Where do we Run Workloads?



Public Cloud
Computer that Handle Acess
Company A
Public users
Company B
Access through the internet
Cloud Provider's Datacenter
Company C
Users who access the cloud from within the perimeter
Optional Subscriber-Based Perimeter
Application Servers
Scalable Hardware
Computing Services
Servers
Storage Services
Platform Servers
SIMFORM



Workload deployment -- determining where and how the workload runs -- is an essential part of workload management. Today, an enterprise can choose to deploy a workload on premises, as well as to a cloud.

Traditionally, workloads are deployed in the enterprise data center, which contains all of the server, storage, network, services and other infrastructure required to operate the workload. The business owns the data center facility and computing resources and fully controls the provisioning, optimization, and maintenance of those resources. The enterprise establishes policies and practices for the data center and workload deployment in order to meet prevailing business goals and regulatory obligations.

With the rise of the internet, cloud computing is now a viable alternative for many on-premises workload deployments.

The **challenge for any business is deciding just where to deploy a given workload**. Today, most general-purpose workloads can operate successfully in the public cloud, and, increasingly, applications are designed and developed to run natively and solely in a public cloud.

Workload Challenges

Technologically, the most demanding workloads may struggle in the public cloud. Some workloads require high-performance network storage or depend on internet throughput.

For example, database clusters that need high throughput and low latency may be unsuited to the cloud -- and the cloud provider may offer high-performance database services as an alternative. Applications that rely on low latency or are not designed for distributed computing infrastructures are usually kept on premises.

Technical issues aside, a business may decide to keep workloads on premises for business continuance or regulatory reasons.

Cloud clients have little actual insight into the underlying hardware and other infrastructure that hosts the workloads and data. That can be problematic for businesses obligated to meet data security and other regulatory requirements such as clear auditing and proof of data residency. Keeping those sensitive workloads in the local data center allows the business to control its own infrastructure and implement the necessary auditing and controls.

Workload Challenges

Cloud providers are also independent businesses that serve their own business interests and may not be able to meet an enterprise's specific uptime or resilience expectations for a workload. Outages happen and may last for hours -- even days -- adversely affecting client businesses and their customer base. Consequently, organizations often opt to keep critical workloads in the local data center where dedicated IT staff can maintain them.

Some organizations implement a hybrid cloud strategy that mixes on-premises, private cloud and public cloud services. This provides flexibility to run workloads and manage data where it makes the most sense, for reasons ranging from costs to security to governance and compliance. This presents tradeoffs -- for example, an organization may keep sensitive data and workloads in its own data center to preserve more direct control over them, but it also takes on more security responsibilities for them.

Cloud Workload – An Anatomy

<u>Key Terms to Understand</u>

Lease: A lease is defined as a contract between the Cloud Service Provider and the end user to facilitate the usage of resources available with the CSP to execute a workload of the end user.

Application: One or more business process encapsulated in code which requires computing resources to execute and provide business value.

Job: A Work Breakdown Structure of an application. An application may contain several jobs, and all of them need to be executed to complete execution of the application.

Tasks: Steps that need to be performed to complete a job. Task can be sequential or parallel.

When we talk about cloud workload we are referring to the completion of a specific job which provides some business values.

Resource Allocation vs Task Scheduling

- ***It might look similar, but there are lots of differences***
- ***RA➔ Identifying and allocating a resource of a type of work load***
- ***TS➔ Ability to shuffle and manage tasks to operate efficiently on the available resources.***

Amdahl's Law

Amdahl's Law Formula

$$S_{max} = \frac{1}{(1-p)+\frac{p}{s}}$$

In computer programming, Amdahl's law is that, in a program with parallel processing , a relatively few instructions that have to be performed in sequence will have a limiting factor on program speedup such that adding more processors may not make the program run faster.

This is generally an argument against parallel processing for certain applications and, in general, against overstated claims for parallel computing. Others argue that the kinds of applications for which parallel processing is best suited tend to be larger problems in which scaling up the number of processors does indeed bring a corresponding improvement in throughput and performance.

Amdahl's law formula calculates the expected speedup of the system if one part is improved. It has three parts: ***S_{max}, p, and s.***

S_{max} is the maximum possible improvement of the overall system. It is expressed as a decimal greater than 1. If the operation is improved to be done in half the time, $S_{max} = 2$. Higher means a greater improvement.

p is the part of the system to be improved, expressed as a number between 0-1. If the part is 45% of the system, $p = 0.45$.

s is the improvement factor of *p*, expressed by how many times faster *p* can be done. If it can be done in 1/3rd the time, then $s = 3$.

Essentially, the equation subtracts out the part to be improved, then puts it back in after it has been improved.

Scheduling Techniques

- While a VI manager like OpenNebula can handle all the minutiae of managing VMs in a pool of physical resources, **scheduling these VMs efficiently is a different and complex matter.**
- **Immediate provisioning model** is used by commercial cloud providers, such as Amazon, since their data centers' capacity is assumed to be infinite.
- Best-effort provisioning where requests have to be queued and prioritized
- **Advance provisioning** where resources are pre-reserved so they will be guaranteed to be available at a given time period.
- However, when managing a private cloud with limited resources, an immediate provisioning model is insufficient.
- **A lease-based** resource **provisioning model** that can act as a scheduling back-end for OpenNebula, supporting other **provisioning models** other than the immediate provisioning models in existing cloud providers. In particular, Haizea adds support for both best-effort provisioning and advance reservations, when managing a finite number of resources.

Scheduling Techniques – Existing Approaches

- Efficient reservation of resources in resource management systems has been studied considerably, particularly in the context of job scheduling.
- In fact, most modern job schedulers support **advance reservation of resources**, but their implementation falls short in several aspects.
- First of all, they are constrained by the **job abstraction**; when a user makes an advance reservation in a job-based system, the user does not have direct and unfettered access to the resources. Cloud users can access the VMs they requested, and are allowed to submit jobs to them.
- Example-1: XYZ Inc creates a new queue that will be bound to the reserved resources, guaranteeing that jobs submitted to that queue will be executed on them (assuming they have permission to do so)
- Example-2: ABC Inc, simply allow users to specify that a submitted job should use the reserved resources (if the submitting user has permission to do so).

Scheduling Techniques – Existing Approaches

- Additionally, advance reservations lead to utilization problems caused by the need to vacate resources before a reservation can begin.
- Traditional job schedulers are unable to efficiently schedule workloads combining both best-effort jobs and advance reservations.
- However, advance reservations can be supported more efficiently by using a scheduler capable of preempting running jobs at the start of the reservation and resuming them at the end of the reservation.
- Preemption can also be used to run large parallel jobs (which tend to have long queue times) earlier, and it is specially relevant in the context of urgent computing, where resources have to be provisioned on very short notice and the likelihood of having jobs already assigned to resources is higher.
- While preemption can be accomplished by canceling a running job, the least disruptive form of preemption is check pointing, where the preempted job's entire state is saved to disk, allowing it to resume its work from the last checkpoint.

Scheduling Techniques – Existing Approaches

- Additionally, some schedulers also support job migration, allowing check-pointed jobs to restart on other available resources, instead of having to wait until the preempting job or reservation has completed.
- •Check-pointing-based preemption, requires the job's executable itself to be checkpointable. An application can be made checkpointable by explicitly adding that functionality to an application (application-level and library-level checkpointing) OR transparently by using OS-level checkpointing, where the operating system (such as Cray, IRIX, and patched versions of Linux using BLCR [17]) checkpoints a process, without rewriting the program or relinking it with checkpointing libraries.
- •Thus, a job scheduler capable of checkpointing-based preemption and migration could be used to checkpoint jobs before the start of an advance reservation, minimizing their impact on the schedule.
- •However, the application and library-level checkpointing approaches burden the user with having to modify their applications to make them checkpointable, imposing a restriction on the software environment. On the other hand, OS-level checkpointing is a more appealing option, but still imposes certain software restrictions on resource consumers.

Scheduling Techniques – Existing Approaches

- An alternative approach to supporting advance reservations was proposed by Nurmi et al. [18], which introduced "virtual advance reservations for queues" (VARQ).
- •This approach overlays advance reservations over traditional job schedulers by first predicting the time a job would spend waiting in a scheduler's queue and then submitting a job (representing the advance reservation) at a time such that, based on the wait time prediction, the probability that it will be running at the start of the reservation is maximized.
- •Since no actual reservations can be done, VARQ jobs can run on traditional job schedulers, which will not distinguish between the regular best-effort jobs and the VARQ jobs.
- •Although this is an interesting approach that can be realistically implemented in practice (since it does not require modifications to existing scheduler), it still depends on the job abstraction.

Scheduling Techniques – VM Overheads

Virtualization technologies are a key enabler of many features found in IaaS clouds. Virtual machines are also an appealing vehicle for implementing efficient reservation of resources due to:

- Ability to be suspended,
- Potentially migrated,
- Resumed without modifying any of the applications running inside the VM.

However, virtual machines also raise additional challenges related to the overhead of using VMs:

- **Preparation Overhead.** When using VMs to implement reservations, a VM disk image must be either prepared on-the-fly or transferred to the physical node where it is needed. Since a VM disk image can have a size in the order of gigabytes, this preparation overhead can significantly delay the starting time of leases. This delay may, in some cases, be unacceptable for advance reservations that must start at a specific time.
- **Runtime Overhead**. Once a VM is running, scheduling primitives such as **checkpointing** and **resuming** can incur in significant overhead since a VM's entire memory space must be saved to disk, and then read from disk. Migration involves transferring this saved memory along with the VM disk image. Similar to deployment overhead, this overhead can result in noticeable delays.

Reservation Based Provisioning

- A particularly interesting problem when provisioning virtual infrastructures is **how to deal with situations where the demand for resources is known beforehand**—for example, when an experiment depending on some complex piece of equipment is going to run from 2 pm to 4 pm, and computational resources must be available at exactly that time to process the data produced by the equipment.
- Commercial clouds do have **infinite resources** to handle this situation. On the other hand, when dealing with **finite capacity**, a different approach is needed. However, the intuitively simple solution of reserving the resources beforehand is not so simple, because it is known to cause resources to be underutilized, due to the difficulty of scheduling other requests around an inflexible reservation.

Provisioning to meet SLA

- IaaS clouds can be used to deploy services that will be consumed by users other than the one that has deployed the services.
- There is a distinction between the **cloud consumer** (i.e., the **service owner**; for instance, the company that develops and manages the applications) and the **end users** of the resources provisioned on the cloud (i.e., the **service user**; for instance, the users that access the applications).
- Furthermore, **service owners** will enter into **service-level agreements (SLAs)** with their end users, covering guarantees such as the timeliness with which these services will respond.
- Requirements are formalized in infrastructure SLAs between **the service owner** and **cloud provider**, separate from the high-level **SLAs between the service owner** and **its end users**.

Provisioning to meet SLA

- In many cases, either the service owner is not resourceful enough to perform an exact service sizing or service workloads are hard to anticipate in advance.
- Therefore, to protect high-level SLAs, the cloud provider should cater for **elasticity on demand**.
- **Scaling and de-scaling** of an application is best managed by the application itself. The reason is that in many cases, resource allocation decisions are **application-specific** and are being driven by the **application level metrics**.

Scheduling Techniques for Advance Reservation of Capacity

- Virtualization technologies are a key enabler of many features found in IaaS clouds. Virtual machines are also an appealing vehicle for implementing efficient reservation of resources due to:
  - Ability to be suspended,
  - Potentially migrated,
  - Resumed without modifying any of the applications running inside the VM.
- However, virtual machines also raise additional challenges related to the overhead of using VMs:
  - **Preparation Overhead.** When using VMs to implement reservations, a VM disk image must be either prepared on-the-fly or transferred to the physical node where it is needed. Since a VM disk image can have a size in the order of gigabytes, this preparation overhead can significantly delay the starting time of leases. This delay may, in some cases, be unacceptable for advance reservations that must start at a specific time.
  - **Runtime Overhead**. Once a VM is running, scheduling primitives such as **checkpointing** and **resuming** can incur in significant overhead since a VM's entire memory space must be saved to disk, and then read from disk. Migration involves transferring this saved memory along with the VM disk image. Similar to deployment overhead, this overhead can result in noticeable delays.

Solution – Haizea Scheduler

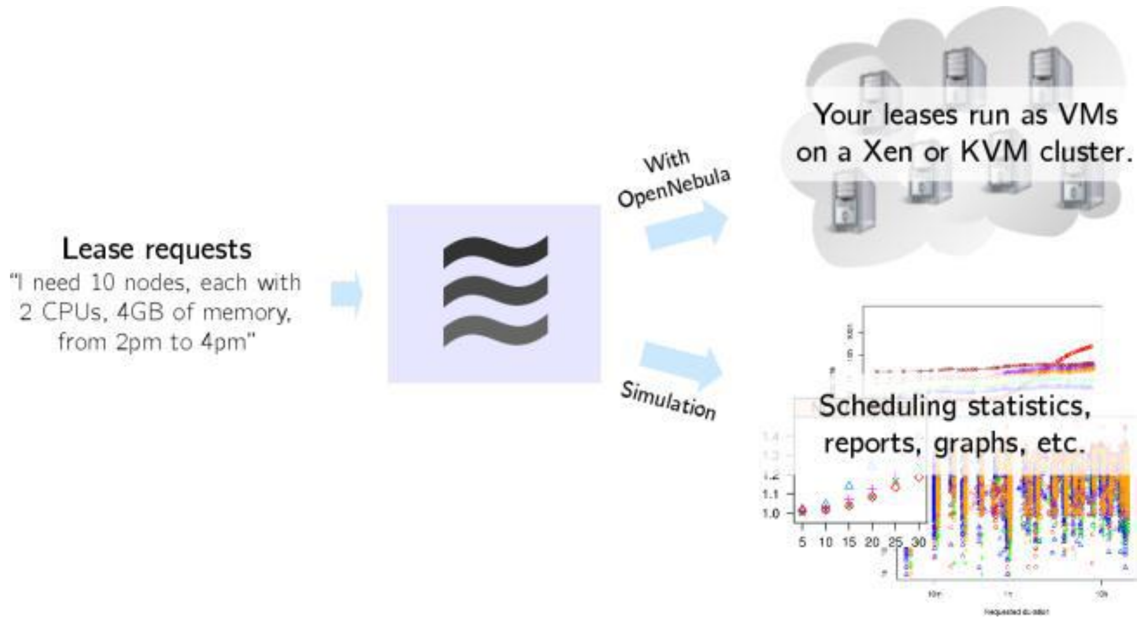
The Haizea project (http://haizea.cs.uchicago.edu/) was created to develop a scheduler that can efficiently support advance reservations efficiently by using the suspend/resume/migrate capability of VMs, but minimizing the overhead of using VMs.

The fundamental resource provisioning abstraction in Haizea is the lease, with three types of lease currently supported:

Advanced reservation leases, where the resources must be available at a specific time.

Best-effort leases, where resources are provisioned as soon as possible and requests are placed on a queue if necessary.

Immediate leases, where resources are provisioned when requested or not at all.



Lease requests
"I need 10 nodes, each with 2 CPUs, 4GB of memory, from 2pm to 4pm"
With OpenNebula
Your leases run as VMs on a Xen or KVM cluster.
Simulation
Scheduling statistics, reports, graphs, etc.

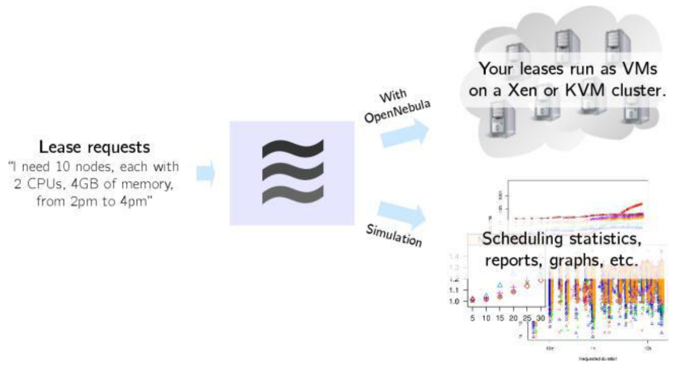
What is Haizea Scheduler

When managing a private cloud with limited resources, **an immediate provisioning model is insufficient**. A **lease-based resource provisioning model** that can act as a scheduling back-end for OpenNebula, **supporting other provisioning models other than the immediate provisioning models in existing cloud providers**. In particular, Haizea adds support for both best-effort provisioning and advance reservations, **when managing a finite number of resources.** The remainder of this section describes Haizea's leasing model and the algorithms Haizea uses to schedule these leases.

•We define a **lease** as "**a negotiated and renegotiable agreement between a resource provider and a resource consumer, where the former agrees to make a set of resources available to the latter, based on a set of lease terms presented by the resource consumer**."

•The terms must encompass the following:

- •the hardware resources required by the resource consumer, such as CPUs, memory, and network bandwidth;
- •a software environment required on the leased resources;
- •and an availability period during which a user requests that the hardware and software resources be available.


Lease requests
"I need 10 nodes, each with 2 CPUs, 4GB of memory, from 2pm to 4pm"
With OpenNebula
Your leases run as VMs on a Xen or KVM cluster.
Simulation
Scheduling statistics, reports, graphs, etc.

How Haizea Scheduler Works

We focus on the availability dimension of a lease and, in particular, on how to efficiently support advance reservations. Thus, we consider the following availability terms:

•Start time may be unspecified (a best-effort lease) or specified (an advance reservation lease). In the latter case, the user may specify either a specific start time or a time period during which the lease start may occur.

• Maximum duration refers to the total maximum amount of time that the leased resources will be available.

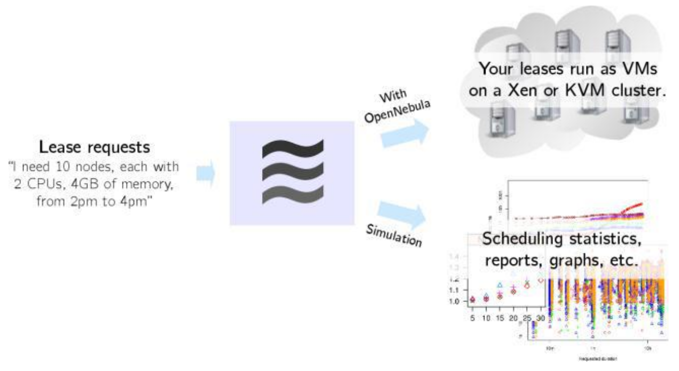
• Leases can be preemptable. A preemptable lease can be safely paused without disrupting the computation that takes place inside the lease.

Haizea's resource model considers that it manages ***W*** physical nodes capable of running virtual machines. Each node ***i*** has CPUs, megabytes (MB) of memory, and MB of local disk storage.

•We assume that all disk images required to run virtual machines are available in a repository from which they can be transferred to nodes as needed and that all are connected at a bandwidth of ***B*** MB/sec by a switched network.

•A lease is implemented as a set of ***N*** VMs, each allocated resources described by a ***tuple (p, m, d, b)***, where ***p*** is number of CPUs, ***m*** is memory in MB, ***d*** is disk space in MB, and ***b*** is network bandwidth in MB/sec.

•A disk image ***I*** with a size of size *(I)* MB must be transferred from the repository to a node before the VM can start. When transferring a disk image to multiple nodes, we use multicasting and model the transfer time as size (I)/B.



Lease requests
"I need 10 nodes, each with 2 CPUs, 4GB of memory, from 2pm to 4pm"
With OpenNebula
Your leases run as VMs on a Xen or KVM cluster.
Simulation
Scheduling statistics, reports, graphs, etc.

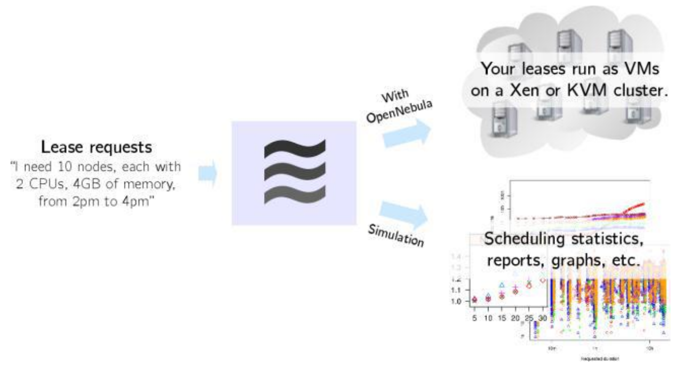
How Haizea Scheduler Works

Haizea is designed to process lease requests and determine how those requests can be mapped to virtual machines, leveraging their suspend/resume/migrate capability, in such a way that the leases' requirements are satisfied.

•The scheduling component of Haizea allow best-effort leases to be preempted if resources have to be freed up for advance reservation requests.

•Additionally, to address the preparation and runtime overheads mentioned earlier, the scheduler allocates resources explicitly for the overhead activities (such as transferring disk images or suspending VMs) instead of assuming they should be deducted from the lease's allocation.

•Besides guaranteeing that certain operations complete on time (e.g., an image transfer before the start of a lease), the scheduler also attempts to minimize this overhead whenever possible, most notably by reusing disk image transfers and caching disk images on the physical nodes.


Lease requests
"I need 10 nodes, each with 2 CPUs, 4GB of memory, from 2pm to 4pm"
With OpenNebula
Your leases run as VMs on a Xen or KVM cluster.
Simulation
Scheduling statistics, reports, graphs, etc.

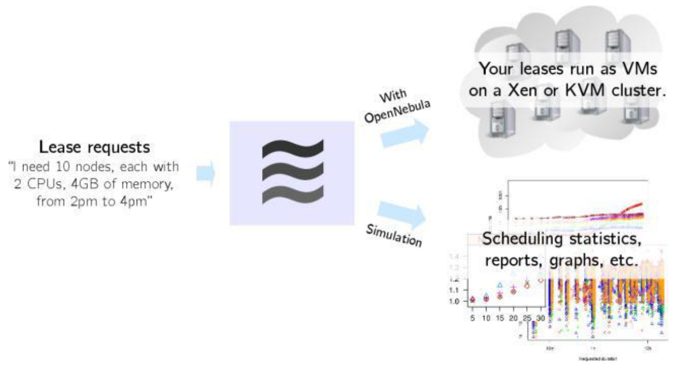
How Haizea Scheduler Works

Best-effort leases are scheduled using a queue. When a best-effort lease is requested, the lease request is placed at the end of the queue, which is periodically evaluated using a backfilling algorithm to determine if any leases can be scheduled.

•The scheduler does this by first checking the earliest possible starting time for the lease on each physical node, which will depend on the required disk images. For example, if some physical nodes have cached the required disk image, it will be possible to start the lease earlier on those nodes.

•Once these earliest starting times have been determined, the scheduler chooses the nodes that allow the lease to start the soonest.

•The use of VM suspension/resumption allows the best-effort leases to be scheduled even if there are not enough resources available for their full requested duration.


Lease requests
"I need 10 nodes, each with 2 CPUs, 4GB of memory, from 2pm to 4pm"
With OpenNebula
Your leases run as VMs on a Xen or KVM cluster.
Simulation
Scheduling statistics, reports, graphs, etc.

How Haizea Scheduler Works

Advance reservations, on the other hand, do not go through a queue, since they must start at either the requested time or not at all.

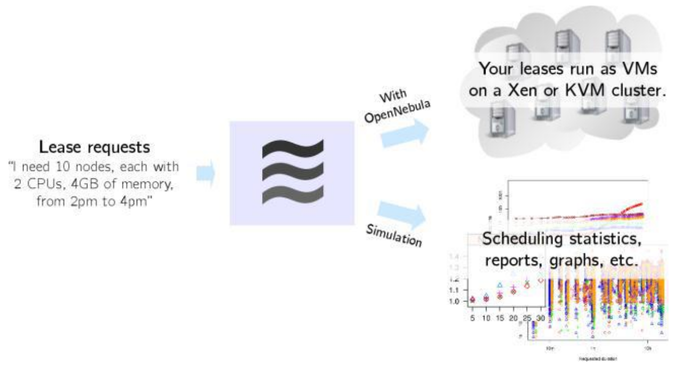
•Thus, scheduling this type of lease is relatively simple, because it mostly involves checking if there are enough resources available during the requested interval.

•However, the scheduler must also check if any associated overheads can be scheduled in such a way that the lease can still start on time.

•For preparation overhead, the scheduler determines if the required images can be transferred on time.

•These transfers are scheduled using an Earliest Deadline First (EDF) algorithm, where the deadline for the image transfer is the start time of the advance reservation lease.

•For runtime overhead, the scheduler will attempt to schedule the lease without having to preempt other leases; if preemption is unavoidable. The necessary suspension operations are scheduled; if they can be performed on time.


Lease requests
"I need 10 nodes, each with 2 CPUs, 4GB of memory, from 2pm to 4pm"
With OpenNebula
Your leases run as VMs on a Xen or KVM cluster.
Simulation
Scheduling statistics, reports, graphs, etc.

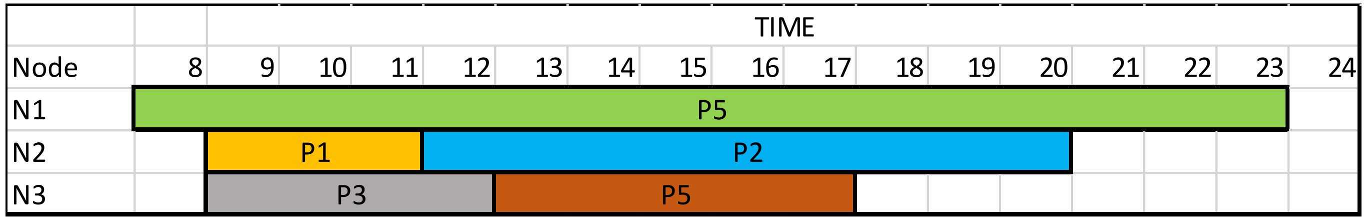
Leasing Schedule – Best Effort Lease (BEL)

- **Best-effort leases** are scheduled using a **queue**. When a best-effort lease is requested, the lease request is placed at the end of the queue, which is periodically evaluated using a backfilling algorithm to determine if any leases can be scheduled.

- The scheduler does this by **first checking the earliest possible starting time** for the lease on each physical node, which will depend on the required disk images. **For example**, if some physical nodes have cached the required disk image, it will be possible to start the lease earlier on those nodes.

- Once these earliest starting times have been determined, the scheduler chooses the nodes that allow the lease to start the soonest.

- The use of **VM suspension/resumption** allows the best-effort leases to be scheduled **even if there are not enough resources available for their full requested duration**.



e.g. If we have three nodes N1, N2 and N3 and five processes have to be scheduled with entry time and duration of resources as follows – Best effort queue

Process	Entry time	Time required	Node	Start time		
P1	9am	3 hours	N2	9am	Ended at 12pm	
P2	10am	10 hours	N2	12pm	Ended at 8pm	
P3	9.30am	4 hours	N3	9.30am	Ended at 1.30pm	
P4	11am	5hours	N3	1.30pm	Ended at 6.30pm	
P5	8am	15hours	N1	8am	Ended at 11pm	

Leasing Schedule – Advanced Reservation (AR)

- **Advance reservations**, on the other hand, **do not go through a queue, since they must start at either the requested time or not at all**.
- Thus, scheduling this type of lease is relatively simple, because it mostly involves checking if there are enough resources available during the requested interval.
- However, the scheduler must also check **if any associated overheads can be scheduled** in such a way that the lease can still start **on time**.
- **For preparation overhead**, the scheduler determines if the required images can be transferred on time.
- These transfers are scheduled using an **Earliest Deadline First (EDF) algorithm**, where **the deadline for the image transfer is the start time of the advance reservation lease**.
- **For runtime overhead**, the scheduler will attempt to schedule the lease without having to preempt other leases; if preemption is unavoidable. The necessary suspension operations are scheduled; if they can be performed on time.

e.g. If we have three nodes N1, N2 and N3 and five processes have to be scheduled with start time and duration with P2 and P5 being advanced reservation

Process	Entry time	Time required	Node	Start time		
P1	9am	3 hours				
P2	10am	10 hours				
P3	9.30am	4 hours				
P4	11am	5hours				
P5	8am	15hours				

CAPACITY MANAGEMENT TO MEET SLA COMMITMENTS

- If temporal behavior of services with respect to resource demands is highly predictable, then capacity can be efficiently scheduled using reservations.

- In this section we focus on **less predictable elastic workloads**. For these workloads, **exact scheduling of capacity may not be possible**. Rather than that, **capacity planning and optimizations are required**.

- IaaS providers perform two complementary management tasks:

  (1) **Capacity planning** to make sure that SLA obligations are met as contracted with the service providers and;

  (2) **Continuous optimization** of resource utilization in specific workload to make the most efficient use of the existing capacity.

CAPACITY MANAGEMENT TO MEET SLA COMMITMENTS - Infrastructure SLA’s

- IaaS can be regarded as a giant virtual hardware store, where computational resources such as virtual machines (VM), virtual application networks (VAN) and virtual disks (VD) can be **ordered on demand in the matter of minutes or even seconds**.
- Chandra et al. [29] quantitatively study advantages of **fine-grain resource allocation** in a shared hosting platform. As this research suggests, **fine-grain temporal and spatial resource allocation** may lead to substantial improvements in capacity utilization.
- Amazon EC2 [1] offers small, large, and extra large general-purpose VM instances and **high-CPU**, **medium** and **extra large** instances. It is possible that more instance types (**e.g., I/O high, memory high, storage high, etc.**) will be added in the future should a demand for them arise. Other IaaS providers—for example, GoGrid [3] and FlexiScale [4]—follow similar strategy.

CAPACITY MANAGEMENT TO MEET SLA COMMITMENTS - Infrastructure SLA's

- Thus, to deploy a service on a cloud, **a service provider orders suitable virtual hardware** and installs its application software on it.
- From the IaaS provider, a given service configuration is a virtual resource array of black box resources, which correspond to the number of instances of resource type.
- For example, a typical three-tier application may contain **ten general-purpose small instances** to run Web front-ends, **three large instances** to run an application server cluster with load balancing and redundancy, and **two large instances** to run a replicated database.
- A risk mitigation mechanism to protect user experience in the IaaS model is offered by infrastructure SLAs (i.e., the SLAs formalizing capacity availability) signed between service provider and IaaS provider.

CAPACITY MANAGEMENT TO MEET SLA COMMITMENTS - Infrastructure SLA’s

- There is **no universal approach to infrastructure SLAs**. As the IaaS field matures and more experience is being gained, some methodologies may become more popular than others. Also some methods may be more suitable for specific workloads than other. There are three main approaches as follows.
- **No SLAs.** This approach is based on two premises: **(a)** Cloud always has spare capacity to provide on demand, and **(b)** services are not QoS sensitive and can withstand moderate performance degradation. This methodology is best suited for the best effort workloads.
- **Probabilistic SLAs.** These SLAs allow **us to trade capacity availability for cost of consumption**. Probabilistic SLAs specify clauses that determine availability percentile for contracted resources computed over the SLA evaluation period. **The lower the availability percentile, the cheaper the cost of resource consumption.** This type of SLA is **suitable for small and medium businesses** and **for many enterprise grade applications**.
- **Deterministic SLAs.** These are, in fact, probabilistic SLAs where **resource availability percentile is 100%**. These SLAs are most stringent and difficult to guarantee. From the provider’s point of view, they do not admit capacity multiplexing. Therefore this is the most costly option for service providers, which may be applied for critical services.

CAPACITY MANAGEMENT TO MEET SLA COMMITMENTS - Infrastructure SLA’s

- We will focus on probabilistic SLAs, however, because they represent the more interesting and flexible option and lay the foundation for the rest of discussion on **statistical multiplexing of capacity**.
- Before we can proceed, we need to define the concept, **elasticity rules**, which are are scaling and de-scaling policies that guide transition of the service from one configuration to another to match changes in the environment. The main motivation for defining these policies stems from the pay-as-you-go billing model of IaaS clouds. The service owner is interested in paying only for what is really required to satisfy workload demands minimizing the over-provisioning overhead. There are **three types of elasticity rules**:
  - **Time-driven:** These rules **change the virtual resources array in response to a timer event**. These rules are useful for predictable workloads—for example, for services with well-known business cycles.
  - **OS Level Metrics-Driven:** These rules react on predicates defined in terms of the OS parameters (see Amazon Auto-scaling Service). These **auto-scaling policies are useful for transparently scaling and de-scaling services**. The problem is, however, that in many cases **this mechanism is not precise enough**.
  - **Application Metrics-Driven:** This is a unique RESERVOIR offering that **allows an application to supply application-specific policies** that will be transparently executed by IaaS middleware in reacting on the monitoring information supplied by the service-specific monitoring probes running inside VMs.

OpenNebula additional information available @

https://www.youtube.com/watch?v=Q8wHwnsEkRI

http://opennebula.org/

For additional information refer to **Chapter 6** (On the Management of Virtual Machines for Cloud Infrastructures) from the Textbook - Cloud Computing – Principles and Paradigms. John Wiley Pub, 2011

Rajkumar Buyya, James Broburg & Anderzej M.G,

Q & A……..

BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Credits

•Hwang, Kai; Dongarra, Jack; Fox, Geoffrey C.. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things (Kindle Locations 3532-3533). Elsevier Science. Kindle Edition.