UNIT – III

Infrastructure as a Service (IAAS) & Platform (PAAS): Virtual machines provisioning and Migration services, Virtual Machines Provisioning and Manageability, Virtual Machine Migration Services, VM Provisioning and Migration in Action. On the Management of Virtual machines for Cloud Infrastructures- Aneka—Integration of Private and Public Clouds.

INFRASTRUCTURE AS A SERVICE (IAAS) & PLATFORM (PAAS)

INFRASTRUCTURE AS A SERVICE PROVIDERS

Public Infrastructure as a Service providers commonly offer virtual servers containing one or more CPUs, running several choices of operating systems and a customized software stack. In addition, storage space and communication facilities are often provided.

Features

IAAS offers a set of specialized features that can influence the cost benefit ratio to be experienced by user applications when moved to the cloud.

The most relevant features are:

1. Geographic distribution of data centers.
2. Variety of user interfaces and APIs to access the system.
3. Specialized components and services that aid Particular applications (e.g., load- balancers, firewalls).
4. Choice of virtualization platform and operating systems and
5. Different billing methods and period (e.g., prepaid vs. postpaid, hourly vs. monthly).

Geographic Presence— To improve availability and responsiveness, a provider of worldwide services would typically build several data centers distributed around the world. For example, Amazon Web Services presents the concept of availability zones and regions for its EC2 service. Availability zones are distinct locations that are engineered to be insulated from failures in other availability zones and provide inexpensive, low-latency network connectivity to other availability zones in the same region. Regions, in turn, are geographically dispersed and will be in separate geographic areas or countries.

User Interfaces and Access to Servers— Ideally, a public IaaS provider must provide multiple access means to its cloud, thus catering for various users and their preferences. Different types of user interfaces (UI) provide different levels of abstraction, the most common being graphical user interfaces (GUI), command-line tools (CLI), and Web service (WS) APIs.

GUIs are preferred by end users who need to launch, customize, and monitor a few virtual servers and do not necessarily need to repeat the process several times. On the other hand, CLIs offer more flexibility and the possibility of automating repetitive tasks via scripts (e.g., start and shutdown a number of virtual servers at regular intervals).

Advance Reservation of Capacity— Advance reservations allow users to request for an IaaS provider to reserve resources for a specific time frame in the future, thus ensuring that cloud resources will be available at that time. However, most clouds only support best-effort requests that means users can request server whenever resources are available .

Amazon Reserved Instances is a form of advance reservation of capacity, allowing users to pay a fixed amount of money in advance to guarantee resource availability at anytime during an agreed period and then paying a discounted hourly rate when resources are in use. However, only long periods of 1 to 3 years are offered; therefore, users cannot express their reservations in finer granularities—for example, hours or days.

Automatic Scaling and Load Balancing— Automatic scaling is a highly desirable feature of IaaS clouds. It allows users to set conditions for when they want their applications to scale up and down, based on application-specific metrics such as transactions per second, number of simultaneous users, request latency, and so forth.

When the number of virtual servers is increased by automatic scaling, incoming traffic must be automatically distributed among the available servers. This activity enables applications to promptly respond to traffic increase while also achieving greater fault tolerance.

Service-Level Agreement- Service-level agreements (SLAs) are offered by IaaS providers to express their commitment to delivery of a certain QoS. To customers it serves as a warranty. An SLA usually include availability and performance guarantees. Additionally, metrics must be agreed upon by all parties as well as penalties for violating these expectations.

Most IaaS providers focus their SLA terms on availability guarantees, specifying the minimum percentage of time the system will be available during a certain period. For instance, Amazon EC2 states that "if the annual uptime Percentage for a customer drops below 99.95% for the service year, that customer is eligible to receive a service credit equal to 10% of their bill.3"

Hypervisor and Operating System Choice— Traditionally, IaaS offerings have been based on heavily customized open-source Xen deployments. IaaS providers needed expertise in Linux, networking, virtualization, metering, resource management, and many other low-level aspects to successfully deploy and maintain their cloud offerings.

More recently, there has been an emergence of turnkey IaaS platforms such as VMWare VCloud and Citrix Cloud Center (C3) which have lowered the barrier of entry for IaaS competitors, leading to a rapid expansion in the IaaS marketplace.

Case Studies

Amazon Web Services: Amazon WS4 (AWS) is one of the major players in the cloud computing market. It pioneered the introduction of IaaS clouds in 2006. It offers a variety cloud services, most notably: S3 (storage), EC2 (virtual servers), Cloudfront (content delivery), Cloudfront Streaming (video streaming), Simple DB (structured datastore), RDS (Relational Database), SQS (reliable messaging), and Elastic MapReduce (data processing). The ElasticCompute Cloud (EC2) offers Xen-based virtual servers (instances) that can be instantiated from Amazon Machine Images (AMIs). Instances are available in a variety of sizes, operating systems, architectures, and price. CPU capacity of instances is measured in Amazon Compute Units and, although fixed for each instance, vary among instance types from 1 (small instance) to 20 (high CPU instance). Each instance provides a certain amount of non persistent disk space; a persistence disk service (Elastic Block Storage) allows attaching virtual disks to instances with space up to 1TB. Elasticity can be achieved by combining the Cloud Watch, Auto Scaling and Elastic Load Balancing features, which allow the number of instances to scale up and down automatically based on a set of customizable rules, and traffic to be distributed across available instances. Fixed IP address (Elastic IPs) are not available by default, but can be obtained at an additional cost.

Flexiscale: Flexiscale is a UK-based provider offering services similar in nature to Amazon Web Services. Flexiscale cloud provides the following features: available in UK; Web services (SOAP), Web-based user interfaces; access to virtual server mainly via SSH (Linux) and Remote Desktop (Windows); 100% availability SLA with automatic recovery of VMs in case of hardware failure; per hour pricing; Linux and Windows operating systems; automatic scaling (horizontal/vertical).

Joyent: Joyent's Public Cloud offers servers based on Solaris containers virtualization technology. These servers, dubbed accelerators, allow deploying various specialized software- stack based on a customized version of Open- Solaris operating system, which include by default a Web-based configuration tool and several pre-installed software, such as Apache, MySQL, PHP, Ruby on Rails, and Java. Software load balancing is available as an accelerator in addition to hardware load balancers. A notable feature of Joyent's virtual servers is automatic vertical scaling of CPU cores, which means a virtual server can make use of additional CPUs automatically up to the maximum number of cores available in the physical host.

The Joyent public cloud offers the following features: multiple geographic locations inthe United States; Web-based user interface; access to virtual server via SSH and Web-based administration tool; 100% availability SLA; per month pricing; OS-level virtualization Solaris containers; Open- Solaris operating systems; automatic scaling(vertical).

GoGrid: GoGrid, like many other IaaS providers, allows its customers to utilize a range of pre- made Windows and Linux images, in a range of fixed instance sizes. GoGrid also offers "value- added" stacks on top for applications such as high- volume Web serving, e-Commerce, and database stores. It offers some notable features, such as a "hybrid hosting" facility, which combines traditional dedicated hosts with auto-scaling cloud server infrastructure. As part of its core IaaS offerings, GoGrid also provides free hardware load balancing, auto-scaling capabilities, and persistent storage, features that typically add an additional cost for most other IaaS providers.

Rackspace Cloud Servers: Rackspace Cloud Servers is an IaaS solution that provides fixed size instances in the cloud. Cloud Servers offers a range of Linux- based pre-made images. A user can request different-sized images, where the size is measured by requested

RAM, not CPU.

PLATFORM AS A SERVICE PROVIDERS

Public Platform as a Service providers commonly offer a development and deployment environment that allow users to create and run their applications with little or no concern to low- level details of the platform. In addition, specific programming languages and frameworks are made available in the platform, as well as other services such as persistent data storage and in memory caches.

Features

Programming Models, Languages, and Frameworks: Programming models made available by IaaS providers define how users can express their applications using higher levels of abstraction and efficiently run them on the cloud platform.

Each model aims at efficiently solving a particular problem. In the cloud computing domain, the most common activities that require specialized models are: processing of large dataset in clusters of computers (MapReduce model), development of request-based Web services and applications; definition and orchestration of business processes in the form of workflows (Workflow model); and high-performance distributed execution of various computational tasks.

For user convenience, PaaS providers usually support multiple programming languages. Most commonly used languages in platforms include Python and Java (e.g., Google AppEngine), .NET languages (e.g., Microsoft Azure), and Ruby (e.g., Heroku). Force.com has devised its own programming language (Apex) and an Excel-like query language, which provide higher levels of abstraction to key platform functionalities.

A variety of software frameworks are usually made available to PaaS developers, depending on application focus. Providers that focus on Web and enterprise application hosting offer popular frameworks such as Ruby on Rails, Spring, Java EE, and .NET.

Persistence Options: A persistence layer is essential to allow applications to record their state and recover it in case of crashes, as well as to store user data. Web and enterprise application developers have chosen relational databases as the preferred persistence method. These databases offer fast and reliable structured data storage and transaction processing, but may lack scalability to handle several peta bytes of data stored in commodity computers. In the cloud computing domain, distributed storage technologies have emerged, which seek to be robust and highly scalable, at the expense of relational structure and convenient query languages.

CASE STUDIES

Aneka: Aneka is a .NET-based service-oriented resource management and development platform. Each server in an Aneka deployment (dubbed Aneka cloud node) hosts the Aneka container, which provides the base infrastructure that consists of services for persistence, security (authorization, authentication and auditing), and communication (message handling and dispatching). Cloud nodes can be either physical server, virtual machines (Xen Server and VMware are supported), and instances rented from Amazon EC2. The Aneka container can also host any number of optional services that can be added by developers to augment the capabilities of an Aneka Cloud node, thus providing a single, extensible framework for

orchestrating various application models.

Several programming models are supported by such task models to enable execution of legacy HPC applications and Map Reduce, which enables a variety of data-mining and search applications. Users request resources via a client to a reservation services manager of the Aneka master node, which manages all cloud nodes and contains scheduling service to distribute request to cloud nodes.

App Engine: Google App Engine lets you run your Python and Java Web applications on elastic infrastructure supplied by Google. App Engine allows your applications to scale dynamically as your traffic and data storage requirements increase or decrease. It gives developers a choice between a Python stack and Java. The App Engine serving architecture is notable in that it allows real-time auto- scaling without virtualization for many common types of Web applications. However, such auto-scaling is dependent on the application developer using a limited subset of the native APIs on each platform, and in some instances you need to use specific Google APIs such as URLFetch, Data store, and mem cache in place of certain native API calls. For example, a deployed App Engine application cannot write to the file system directly (you must use the Google Data store) or open a socket or access another host directly (you must use Google URL fetch service). A Java application cannot create a new Thread either.

Microsoft Azure: Microsoft Azure Cloud Services offers developers a hosted. NET Stack (C#, VB.Net, ASP.NET). In addition, a Java & Ruby SDK for .NET Services is also available. The Azure system consists of a number of elements. The Windows Azure Fabric Controller provides auto-scaling and reliability, and it manages memory resources and load balancing. The .NET Service Bus registers and connects applications together. The .NET Access Control identity providers include enterprise directories and Windows LiveID. Finally, the .NET Workflow allows construction and execution of workflow instances.

Force.com: In conjunction with the Salesforce.com service, the Force.com PaaS allows developers to create add-on functionality that integrates into main Salesforce CRM SaaS application. Force.com offers developers two approaches to create applications that can be deployed on its SaaS plaform: a hosted Apex or Visualforce application. Apex is a proprietary Java-like language that can be used to create Salesforce applications. Visual force is an XML-like syntax for building UIs in HTML, AJAX, or Flex to overlay over the Salesforce hosted CRM system. An application store called App Exchange is also provided, which offers a paid & free application directory.

Heroku: Heroku is a platform for instant deployment of Ruby on Rails Web applications. In the Heroku system, servers are invisibly managed by the platform and are never exposed to users. Applications are automatically dispersed across different CPU cores and servers, maximizing performance and minimizing contention. Heroku has an advanced logic layer than can automatically route around failures, ensuring seamless and uninterrupted service at all times.

Public Cloud and Infrastructure Services

1. 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.

2. 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.

3. Examples for vendors who publicly provide IaaS:
    1. Amazon Elastic Compute Cloud (EC2).
    2. GoGrid
    3. Joyent Accelerator
    4. Rackspace
    5. AppNexus
    6. FlexiScale and Manjrasoft Aneka
4. Amazon Elastic Compute Cloud (EC2) is an IaaS service that provides elastic compute capacity in the cloud.
5. These services can be leveraged via Web services (SOAP or REST), a Web-based AWS (Amazon Web Service) management console, or the EC2 command line tools.
6. The Amazon service provides hundreds of pre-made AMIs (Amazon Machine Images) with a variety of operating systems (i.e., Linux, OpenSolaris, or Windows) and pre-loaded software.
7. Provides complete control of computing resources run on Amazon's computing and infrastructure environment easily
8. Reduces the time required for obtaining and booting a new server's instances to minutes
9. Allows a quick scalable capacity and resources, up and down as the computing requirements change Offers different instances' size according to
    1. The resources' needs (small, large, and extra large)
    2. The high CPU's needs it provides (medium and extra large high CPU instances)
    3. High-memory instances (extra large, double extra large, and quadruple extra large instance)
10. Amazon EC2 is a widely known example for vendors that provide public cloud services.
11. Eucalyptus and Open-Nebula are two complementary and enabling technologies for open source cloud tools, which play an invaluable role in infrastructure as a service and in building private, public, and hybrid cloud architecture.
    1. The Amazon EC2 (Elastic Compute Cloud) is a Web service that allows users to provision new machines into Amazon's virtualized infrastructure in a matter of minutes using a publicly available API
    2. EC2 instance is typically a virtual machine with a certain amount of RAM, CPU, and storage capacity.
12. Amazon EC2 provides its customers with three flexible purchasing models to make it

easy for the cost optimization:

1. On-Demand instances: which allow you to pay a fixed rate by the hourwith no commitment.

2. Reserved instances: which allow you to pay a low, one-time fee and in turn receive a significant discount on the hourly usage charge for that instance. It ensures that any reserved instance you launch is guaranteed to succeed (provided that you have booked them in advance). This means that users of these instances should not be affected by any transient limitations in EC2 capacity.

3. Spot instances: which enable you to bid whatever price you want for instance capacity, providing for even greater savings, if your applications have flexible start and end times.

13. Amazon Elastic Load Balancer is another service that helps in building fault- tolerant applications by automatically provisioning incoming application workload across available Amazon EC2 instances and in multiple availability zones.

Private Cloud and Infrastructure Services

A private cloud aims at providing public cloud functionality, but on private resources:

1. Maintaining control over an organization’s data and resources to meet security and governance’s requirements in an organization.
2. Private cloud exhibits a highly virtualized cloud data center located inside your organization’s firewall.
3. It may also be a private space dedicated for your company within a cloud vendor’s data center designed to handle the organization’s workloads.

Private clouds exhibit the following characteristics:

1. Allow service provisioning and compute capability for an organization’s users in a self-service manner.

2. Automate and provide well-managed virtualized environments.

3. Optimize computing resources, and servers’ utilization.

4. Support specific workloads.

Examples for vendors and frameworks that provide Iaas in private setups

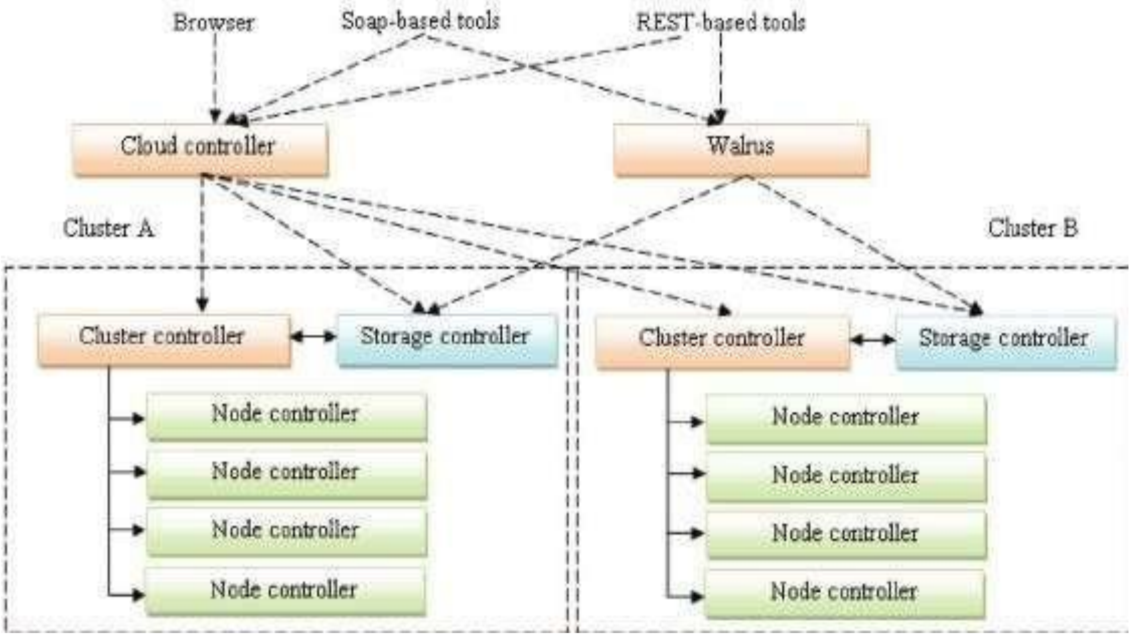
1. Eucalyptus (elastic utility computing architecture linking your programs to useful systems)
2. Open Nebula

Eucalyptus: Eucalyptus is an open-source infrastructure for the implementation of cloud computing on computer clusters. It is considered one of the earliest tools developed as a surge computing (in which data center's private cloud could augment its ability to handle workload's spikes by a design that allows it to send overflow work to a public cloud) tool. Its name is an acronym for **"elastic utility computing architecture for linking your programs to useful systems."**

Eucalyptus features :

1. Interface compatibility with EC2, and S3 (both Web service and Query/REST [Representational State Transfer] interfaces).
2. Simple installation and deployment.
3. Support for most Linux distributions (source and binary packages).
4. Support for running VMs that run atop the Xen hypervisor or KVM.
5. Support for other kinds of VMs, such as VMware, is targeted for future releases.
6. Secure internal communication using SOAP (Simple Object Access Protocol) with WS security.
7. Cloud administrator's tool for system's management and user's accounting.
8. The ability to configure multiple clusters each with private internal network addresses into a single cloud.
9. Eucalyptus aims at fostering the research in models for service's
10. provisioning, scheduling, SLA formulation, and hypervisors' portability.

Eucalyptus Architecture:



1. **Node controller (NC)**controls the execution, inspection, and termination of VM instances on the host where it runs.
2. **Cluster controller (CC)** gathers information about and schedules VM execution on specific node controllers, as well as manages virtual instance network.

3. **Storage controller (SC)** is a put/get storage service that implements Amazon's

S3(Simple Storage Service) interface and provides a way for storing and accessing VM images and user data.

4. **Cloud controller (CLC)** is the entry point into the cloud for users and administrators. It queries node managers for information about resources, makes high-level scheduling decisions, and implements them by making requests to cluster controllers.
5. **Walrus (W)** is the controller component that manages access to the storage services within Eucalyptus. Requests are been communicated to Walrus using the SOAP (Simple Object Access Protocol) or REST (Representational State Transfer) based interface

Hybrid Cloud and Infrastructure Services

A third type of cloud setup named Hybrid cloud

1. A combination of private/internal and external cloud resources existing together by enabling outsourcing of noncritical services and functions in public cloud and keeping the critical ones internal

2. Main function of Hybrid cloud is to release resources from a public cloud and handle sudden demand usage called cloud bursting

Distributed Management of Virtualization

Virtualization's benefits bring their own challenges and complexities presented in the need for a powerful management capabilities. That is why many commercial, open source products and research projects such as OpenNebula, IBM Virtualization Manager, Joyent, and VMware DRS are been developed to be dynamically provision virtual machines, utilizing the physical infrastructure. There are also some commercial and scientific infrastructure cloud computing initiatives, such as Globus VWS, Eucalyptus and Amazon, which provide remote interfaces for controlling and monitoring virtual resources.

One more effort in this context is the RESERVOIR initiative, in which grid interfaces and protocols enable the required interoperability between the clouds or infrastructure's providers.

High Availability

High availability is a system design protocol and an associated implementation that ensures a certain absolute degree of operational continuity during a given measurement period. Availability refers to the ability of a user's community to access the system—whether for submitting new work, updating or altering existing work, or collecting the results of the previous work.

Cloud and Virtualization Standardization Efforts

Standardization is important to ensure interoperability between virtualization management vendors, the virtual machines produced by each one of them, and cloud computing. In the past few years, virtualization standardization efforts led by the Distributed

Management Task Force (DMTF) have produced standards for almost all the aspects of virtualization technology.

DMTF initiated the VMAN (Virtualization Management Initiative), which delivers broadly supported interoperability and portability standards for managing the virtual computing lifecycle. VMAN's OVF (Open Virtualization Format) in a collaboration between industry key players: Dell, HP, IBM, Microsoft, XenSource, and Vmware.

OVF (Open Virtualization Format)

1. VMAN's OVF (Open Virtualization Format) in a collaboration between industry key players: Dell, HP, IBM, Microsoft, XenSource, and Vmware.
2. OVF specification provides a common format to package and securely distribute virtual appliances across multiple virtualization platforms.
3. VMAN profiles define a consistent way of managing a heterogeneous virtualized environment

OCCI and OGF

Open Grid Forum (OGF) organizing an official new working group to deliver a standard API for cloud IaaS, the Open Cloud Computing Interface Working Group (OCCIWG). This group is dedicated for delivering an API specification for the remote management of cloud computing's infrastructure and for allowing the development of interoperable tools for common tasks including deployment, autonomic scaling, and monitoring. The scope of the specification will be covering a high-level functionality required for managing the life-cycle virtual machines (or workloads), running on virtualization technologies (or containers), and supporting service elasticity. The new API for interfacing "IaaS" cloud computing facilities will allow

1. **Consumers** to interact with cloud computing infrastructure on an ad hoc basis.
2. **Integrators** to offer advanced management services.
3. **Aggregators** to offer a single common interface to multiple providers. Providers to offer a standard interface that is compatible with the available tools.
4. **Vendors** of grids/clouds to offer standard interfaces for dynamically scalable service's delivery in their products.

VM Provisioning Process

Typical life cycle of VM and its major possible states of operation, which make the management and automation of VMs in virtual and cloud environments easier Process & Steps to Provision VM. Here, we describe the common and normal steps of provisioning a virtual server:

1. Firstly, you need to select a server from a pool of available servers (physical servers with enough capacity) along with the appropriate OS template you need to provision the virtual machine.
2. Secondly, you need to load the appropriate software (operating system you selected in the previous step, device drivers, middleware, and the needed applications for the service required).
3. Thirdly, you need to customize and configure the machine (e.g., IP address, Gateway) to configure an associated network and storage resources.

4. Finally, the virtual server is ready to start with its newly loaded software. Typically, these are the tasks required or being performed by an IT or a data center's specialist to provision a particular virtual machine.

virtual machines can be provisioned by manually installing an operating system, by using a preconfigured VM template, by cloning an existing VM, or by importing a physical server or a virtual server from another hosting platform. Physical servers can also be virtualized and provisioned using P2V (physical to virtual) tools and techniques (e.g., virt-p2v).

After creating a virtual machine by virtualizing a physical server, or by building a new virtual server in the virtual environment, a template can be created out of it. Most virtualization management vendors (VMware, XenServer, etc.) provide the data center's administration with the ability to do such tasks in an easy way.

Provisioning from a template is an invaluable feature, because it reduces the time required to create a new virtual machine. Administrators can create different templates for different purposes. For example, you can create a Windows 2003 Server template for the finance department, or a Red Hat Linux template for the engineering department.

This enables the administrator to quickly provision a correctly configured virtual server on demand. This ease and flexibility bring with them the problem of virtual machine's sprawl, where virtual machines are provisioned so rapidly that documenting and managing the virtual machine's life cycle become a challenge

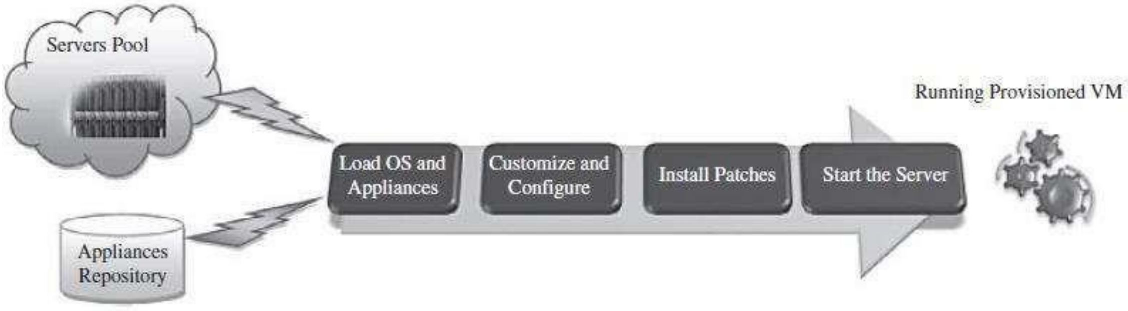


FIGURE 5.4. Virtual machine provision process.

VIRTUAL MACHINE MIGRATION SERVICES

Migration service, in the context of virtual machines, is the process of moving a virtual machine from one host server or storage location to another; there aredifferent techniques of VM migration, hot/life migration, cold/regular migration, and live storage migration of a virtual machine. In this process, all key machine 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. Here are some of the migration's techniques that most virtualization tools provide as a feature.

Migrations Techniques

Live Migration and High Availability.

Live migration (which is also called hot or real-time migration) 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). One of the most significant advantages of live migration is the fact that it facilitates proactive maintenance in case of failure, because the potential problem can be resolved before the disruption of service occurs. Live migration can also be used for load balancing in which work is shared among computers in order to optimize the utilization of available CPU resources.

Live Migration Anatomy, Xen Hypervisor Algorithm.

In this section we will explain live migration’s mechanism and how memory and virtual machine states are being transferred, through the network, from one host A to another host B, the Xen hypervisor is an example for this mechanism. The logical steps that are executed when migrating an OS are summarized in the diagram below In this research, the migration process has been viewed as a transactional interaction between the two hosts involved

Migration Techniques:

- Stage 0: Pre-Migration
    - An active virtual machine exists on the physical host A
- Stage 1: Reservation
    - A request is issued to migrate an OS from host A to B.
    - The necessary resources exist on B and on a VM container of that size.
- Stage 2: Iterative Pre-Copy
    - During the first iteration, all pages are transferred from A to B
    - Subsequent iterations copy only those pages dirtied during the previous transfer phase
- Stage 3: Stop-and-Copy
    - Running OS instance at A is suspended
    - The network traffic is redirected to B
    - CPU state and any remaining inconsistent memory pages are then transferred
    - At the end of this stage, there is a consistent suspended copy of the VM at both A and B.
    - Copy at A is considered primary and is resumed in case of failure
- Stage 4:Commitment
    - Host B indicates to A that it has successfully received a consistent OS image
    - Host A acknowledges this message as a commitment of the migration transaction
    - Host A may now discard the original VM
    - Host B becomes the primary host
- Stage 5: Activation
    - The migrated VM on B is now activated

Live Storage Migration of Virtual Machine.

This kind of migration 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.

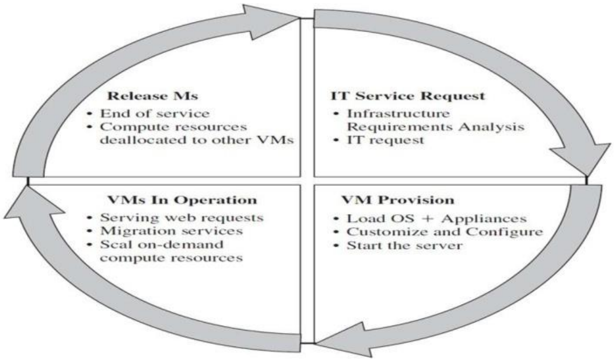
Migration of Virtual Machines to Alternate Platforms

One of the nicest advantages of having facility in data center’s technologies is to have the ability to migrate virtual machines from one platform to another. There are a number of ways for achieving this, such as depending on the source and target virtualization's platforms and on the vendor's tools that manage this facility—for example, the VMware converter that handles migrations between ESX hosts; the VMware server; and the VMware workstation. The VMware converter can also import from other virtualization platforms, such as Microsoft virtual server machines.

VIRTUAL MACHINES PROVISIONING AND MANAGEABILITY

The typical life cycle of VM and its major possible states of operation, which make the management and automation of VMs in virtual and cloud environments easier than in traditional computing environments.

As shown in the diagram below 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 the requirements, and starting the provision of the needed virtual machine. Once it is provisioned and started, it is ready to provide the required service according to an SLA, or a time period after which the virtual is being released; and free resources, in this case, won't be needed.



VM PROVISIONING AND MIGRATION IN ACTION

Now, it is time to get into business with a real example of how we can manage the life cycle, provision, and migrate a virtual machine by the help of one of the open source frameworks used to manage virtualized infrastructure.

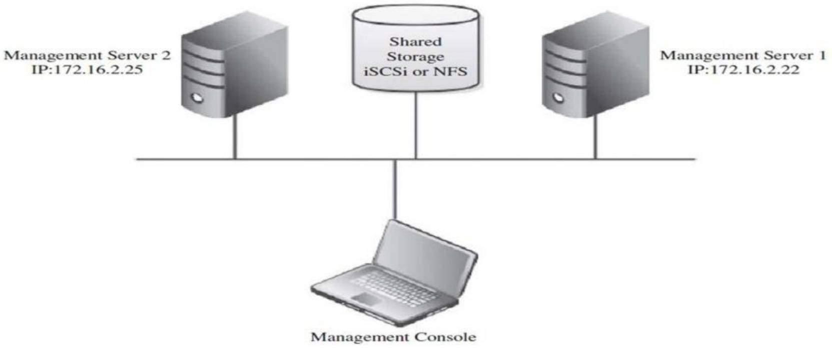
Here, we will use ConVirt (open source framework for the management of open source virtualization like Xen and KVM known previously as XenMan). Deployment Scenario. ConVirt deployment consists of at least one ConVirt workstation, where ConVirt is installed and ran, which provides the main console for managing the VM life cycle, managing images, provisioning new VMs, monitoring machine resources, and so on.

There are two essential deployment scenarios for ConVirt:

1. Basic configuration in which the Xen or KVM virtualization platform is on the local machine, where ConVirt is already installed.
2. An advanced configuration in which the Xen or KVM is on one or more remote servers. The scenario in use here is the advanced one. In data centers, it is very common to install centralized management software (ConVirt here) on a dedicated machine for use in managing remote servers in the data center.

In our example, we will use this dedicated machine where ConVirt is installed and used to manage a pool of remote servers (two machines). In order to use advanced features of

ConVirt (e.g., live migration), you should set up a shared storage for the server pool in use on which the disks of the provisioned virtual machines are stored.



Installation

The installation process involves the following:

1. Installing ConVirt on at least one computer. See reference 28 for installation details.
2. Preparing each managed server to be managed by ConVirt. See reference 28 for managed servers' installation details. We have two managing servers with the following Ips (managed server 1, IP:172.16.2.22; and managed server 2, IP:172.16.2.25) as shown in the deployment diagram (Figure 5.7).
3. Starting ConVirt and discovering the managed servers you have prepared.

Notes

1. Try to follow the installation steps existing in reference 28 according to the distribution of the operating system in use. In our experiment, we use Ubuntu 8.10 in our setup.
2. Make sure that the managed servers include Xen or KVM hypervisors installed.
3. Make sure that you can access managed servers from your ConVirt management console through SSH.

Environment, Software, and Hardware: ConVirt 1.1, Linux Ubuntu 8.10, three machines, Dell core 2 due processor, 4G RAM.

Adding Managed Servers and Provisioning VM: Once the installation is done and you are ready to manage your virtual infrastructure, then you can start the ConVirt management console.

Select any of servers' pools existing (QA Lab in our scenario) and on its context menu, select "Add Server".

1. You will be faced with a message asking about the virtualization platform you want to manage (Xen or KVM).
2. Choose KVM, and then enter the managed server information and credentials (IP, username, and password).
3. Once the server is synchronized and authenticated with the management console, it will appear in the left pane/of the ConVirt.
4. Select this server, and start provisioning your virtual machine.

5. Fill in the virtual machine's information (name, storage, OS template, etc) then you will find it created on the managed server tree powered-off. Note: While provisioning your virtual machine, make sure that you create disks on the shared storage (NFS or iSCSi). You can do so by selecting the “provisioning” tab, and changing the VM_DISKS_DIR to point to the location of your shared NFS.

6. Start your VM (Figures 5.14 and 5.15), and make sure the installation media of the operating system you need is placed in drive, in order to use it for booting the new VM and proceed in the installation process; then start the installation process as shown in Figure 5.16.

7. Once the installation finishes, you can access your provisioned virtual machine from the console icon on the top of your ConVirt management console.

8. Reaching this step, you have created your first managed server and provisioned virtual machine. You can repeat the same procedure to add the second managed server in your pool to be ready for the next step of migrating one virtual machine from one server to the other.

9. To start the migration of a virtual machine from one host to the other, select it and choose a migrating virtual machine.

10. You will have a window containing all the managed servers in your data center. Choose one as a destination and start

11. Once the virtual machine has been successfully placed and migrated to the destination host, you can see it still living and working.

ON THE MANAGEMENT OF VIRTUAL MACHINES FOR CLOUD INFRASTRUCTURES

In 2006, Amazon started offering virtual machines (VMs) to anyone with a credit card for just $0.10/hour through its Elastic Compute Cloud (EC2) service. Although not the first company to lease VMs, the programmer-friendly EC2 Web services API and their pay-as- you-go pricing popularized the “Infrastructure as a Service” (IaaS) paradigm, which is now closely related to the notion of a “cloud.”

Following the success of Amazon EC2, several other IaaS cloud providers, or public clouds, have emerged—such as Elastic- Hosts, GoGrid, and FlexiScale—that provide a publicly accessible interface for purchasing and managing computing infrastructure that is instantiated as VMs running on the provider's data center.

There is also a growing ecosystem of technologies and tools to build private clouds—where inhouse resources are virtualized, and internal users can request and manage these resources using interfaces similar or equal to those of public clouds—and hybrid clouds—where an organization's private cloud can supplement its capacity using a public cloud.

THE ANATOMY OF CLOUD INFRASTRUCTURES

There are many commercial IaaS cloud providers in the market, such as those citedearlier, and all of them share five characteristics:

(i) They provide on-demand provisioning of computational resources.

(ii) they use virtualization technologies to lease these resources.

(iii) they provide public and simple remote interfaces to manage those resources

(iv) they use a pay-as-you-go cost model, typically charging by the hour

(v) they operate data centers large enough to provide a seemingly unlimited amount of resources to their clients (usually touted as "infinite capacity" or "unlimitedelasticity").

1. Private and hybrid clouds share these same characteristics but, instead of selling capacity over publicly accessible interfaces, focus on providing capacity to an organization's internal users.

2. Virtualization technologies have been the key enabler of many of these salient characteristics of IaaS clouds by giving providers a more flexible and generic way of managing their resources. Thus, 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.

3. Virtual infrastructure management in private clouds has to deal with an additionalproblem: 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.

4. Several VI management solutions have emerged over time, such as platform ISF and VMware vSphere, along with open-source initiatives such as Enomaly Computing Platform and Ovirt.

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

Distributed Management of Virtual Machines

The first problem is how to manage the virtual infrastructures themselves. Although resource management has been extensively studied, particularly for job management in high-performance computing, managing VMs poses additional problems that do not arise when managing jobs, such as the need to set up custom software environments forVMs, setting up and managing networking for interrelated VMs, and reducing the various overheads involved in using VMs.

1. Thus, VI managers must be able to efficiently orchestrate all these different tasks. The problem of efficiently selecting or scheduling computational resources is well known.

2. 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.

3. To efficiently schedule resources, VI managers must be able to support flexible and complex scheduling policies and must leverage the ability of VMs to suspend, resume, and migrate. This complex task is one of the core problems that the RESERVOIR(Resources and Services Virtualization without Barriers) project tries to solve.

Reservation-Based Provisioning of Virtualized Resources

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 cloud providers, such as Amazon, have enough resources to provide the illusion of infinite capacity, which means that this situation is simply resolved by requesting the resources exactly when needed; if capacity is "infinite," then there will be resources available at 2 pm. On the other hand, when dealing with finite capacity, a different approach is needed. However, the intuitively simple solution of reserving the resources beforehand turns out to not be so simple, because it is known to cause resources to be underutilized, due to the difficulty of scheduling other requests around an inflexible reservation. VMs allow us to overcome the utilization problems typically associated withadvance reservations and we describe Haizea, a VM-based lease manager supporting advance reservation along with other provisioning models not supported in existing IaaS clouds, suchas best-effort provisioning.

Provisioning to Meet SLA Commitments

IaaS clouds can be used to deploy services that will be consumed by users other thanthe one that deployed the services. For example, a company might depend on an IaaS cloud provider to deploy three-tier applications (Web front-end, application server, and database server) for its customers. In this case, there is a distinction between the cloud consumer (i.e., the service owner) and the end users of the resources provisioned on the cloud (the serviceuser).

Furthermore, service owners will enter into service-level agreements (SLAs) withtheir end users, covering guarantees such as the timeliness with which these services will respond. However, cloud providers are typically not directly exposed to the service semantics or the SLAs that service owners may contract with their end users. The capacity requirements are less predictable and more elastic.

The cloud provider's task is, therefore, to make sure that resource allocation requestsare satisfied with specific probability and timeliness. These 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.

RESERVOIR proposes a flexible framework where service owners may register service-specific elasticity rules and monitoring probes, and these rules are being executed to match environment conditions.

Elasticity of the application should be contracted and formalized as part of capacity availability SLA between the cloud provider and service owner. This poses interesting research issues on the IaaS side, which can be grouped around two main topics:

1. SLA-oriented capacity planning that guarantees that there is enough capacity to guarantee service elasticity with minimal over-provisioning.
2. Continuous resource placement and scheduling optimization that lowers operational costs and takes advantage of available capacity transparently to the service while keeping the service SLAs.

ANEKA—INTEGRATION OF PRIVATE AND PUBLIC CLOUDS

1. Aneka is a software platform and a framework for developing distributed applicationson the cloud. It harnesses the computing resources of a heterogeneous network of workstations and servers or data centers on demand. Aneka provides developers witha rich set of APIs for transparently exploiting these resources by expressing the application logic with a variety of programming abstractions. System administrators can leverage a collection of tools to monitor and control the deployed infrastructure.
2. This can be a public cloud available to anyone through the Internet, a private cloud constituted by a set of nodes with restricted access within an enterprise, or a hybrid cloud where external resources are integrated on demand, thus allowing applicationsto scale. Diagram below provides a layered view of the framework.
3. Aneka is essentially an implementation of the PaaS model, and it provides a runtime environment for executing applications by leveraging the underlying infrastructure of the cloud. Developers can express distributed applications by using the API contained in the Software Development Kit (SDK) or by porting existing legacy applications to the cloud.
4. Such applications are executed on the Aneka cloud, represented by a collection of nodes connected through the network hosting the Aneka container.