UNIT 2

2 marks questions

1. **List the different layers of cloud computing architecture?**
   Physical Infrastructure, Core Middleware, IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-Service), SaaS (Software-as-a-Service)

2. **List the Characteristics of IaaS?**
   - Web access to the resources
   - Centralized management
   - Elasticity and dynamic scaling
   - Shared infrastructure
   - Preconfigured VMs
   - Metered services

3. **What is paas?**
   Platform-as-a-Service (PaaS) solutions provide a development and deployment platform for running applications in the cloud. PaaS consumers or developers can consume language runtimes, application frameworks, databases, message queues, testing tools, and deployment tools as a service over the Internet.

4. **List the advantages of SaaS?**
   Software-as-a-Service (SaaS) is a software delivery model that provides access to applications throughthe Internet as a Web-based service.

5. **List the different deployment models of cloud computing?**
   Public clouds, Private/enterprise clouds, Hybrid clouds, Community clouds

6. **What are hybrid clouds?**
   By bridging the public and private worlds with a layer of proprietary software, hybrid cloud computing gives the best of both worlds. With a hybrid solution, you may host the app in a safe environment while taking advantage of the public cloud's cost savings.

7. **List the advantages of public cloud?**
   - **Minimal Investment:** Because it is a pay-per-use service, there is no substantial upfrontfee, making it excellent for enterprises that require immediate access to resources.
   - **No setup cost:** The entire infrastructure is fully subsidized by the cloud service providers,thus there is no need to set up any hardware.
   - **Infrastructure Management is not required:** Using the public cloud does not necessitate infrastructure management.
   - **No maintenance:** The maintenance work is done by the service provider (not users).
   - **Dynamic Scalability:** To fulfill your company's needs, on-demand resources areaccessible.

8. **List the advantages of community clouds?**
   - **Cost Effective:** It is cost-effective because the cloud is shared by multiple organizations or communities.
   - **Security:** Community cloud provides better security.
   - **Shared resources:** It allows you to share resources, infrastructure, etc. with multiple organizations.
   - **Collaboration and data sharing:** It is suitable for both collaboration and data sharing.

9. **List three major components of virtualization?**
   Guest, host, and virtualization layer

10. **What are hypervisors? List three modules of it?**

A fundamental element of hardware virtualization is the hypervisor, or virtual machine manager (VMM). It recreates a hardware environment in which guest operating systems are installed. Three main modules are dispatcher, allocator, and interpreter.

11. **What is storage virtualization?**

Storage virtualization is a system administration practice that allows decoupling the physical organization of the hardware from its logical representation. Using this technique, users do not have to be worried about the specific location of their data, which can be identified using a logical path.

12. **What is operating system virtualization?**

Operating system virtualization offers the opportunity to create different and separated execution environments for applications that are managed concurrently. Differently from hardware virtualization, there is no virtual machine manager or hypervisor, and the virtualization is done within a single operating system, where the OS kernel allows for multipleisolated user space instances.

13. **Define virtualization.**

Virtualization is a proved technology that makes it possible to run multiple operating system and applications on the same server at same time.

14. **What is emulation?**

Guest programs are executed within an environment that is controlled by the virtualization layer, which ultimately is a program. Also, a completely different environment with respect to the host can be emulated, thus allowing the execution of guest programs requiring specific characteristics that are not present in the physical host.

15. **What are type II hypervisors?**

Type II hypervisors require the support of an operating system to provide virtualization services. This means that they are programs managed by the operating system, which
interact with it through the ABI and emulate the ISA of virtual hardware for guest operating systems. This type of hypervisor is also called a hosted virtual machine since it is hosted within an operating system.

16. **What are Type I hypervisors**

Type I hypervisors run directly on top of the hardware. Therefore, they take the place ofthe operating systems and interact directly with the ISA interface exposed by the underlying hardware, and they emulate this interface inorder to allow the management of guest operating systems. This type of hypervisor is also called a native virtual machine since it runs natively on hardware

17. **What is aggregation?**

A group of separate hosts can be tied together and represented to guests as a single virtual host. This functionality is implemented with cluster management software, which harnesses the physical resources of a homogeneous group of machines and represents them as a single resource.

18. **What is isolation?**

Virtualization allows providing guests—whether they are operating systems, applications, or other entities—with a completely separate environment, in which they are executed.

19. **List the disadvantages of the Community Cloud Model?**

- **Limited Scalability:** Community cloud is relatively less scalable as many organizations share the same resources according to their collaborative interests.
- **Rigid in customization:** As the data and resources are shared among different organizations according to their mutual interests if an organization wants some changes according to their needs they cannot do so because it will have an impact on other organizations.

20. **List the con's of virtualization?**

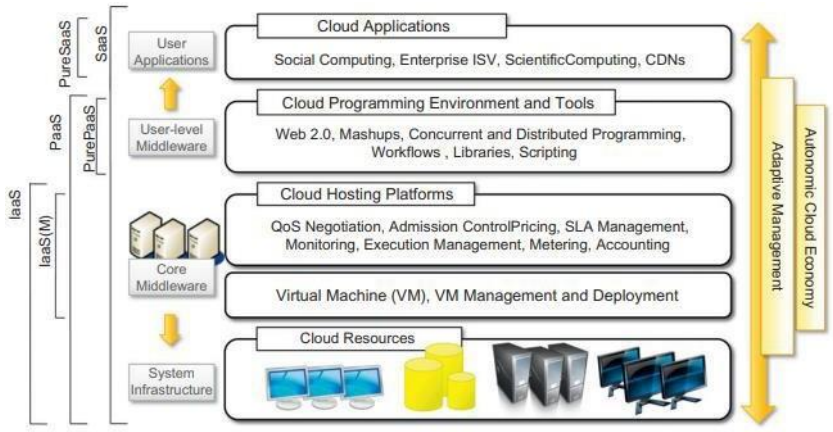
- **Data can be at Risk**
- **Learning New Infrastructure**
- **High Initial Investment**

Long answer questions

1) **Explain cloud reference model with diagram.**

Cloud computing supports any IT service that can be consumed as a utility and delivered through a network, most likely the Internet. Such characterization includes quite different aspects: infrastructure, development platforms, application and services.

Cloud Computing can be viewed as different layers as shown in the figure below:



Physical Infrastructure: This is where the actual hardware and resources are located, like data centres with many computers. It can be a mix of different technologies and includes storage anddatabase systems.

Core Middleware: This manages the physical infrastructure to create a suitable environment for applications. It uses virtualization to divide resources efficiently. It also handles quality, monitoring,and other functions.

IaaS (Infrastructure-as-a-Service): This combines the management layer and the physical infrastructure. Some provide only the management layer. It's good for designing system infrastructurebut doesn't offer many application services.

PaaS (Platform-as-a-Service): This is for developing applications specifically for the cloud. It includes tools and a development platform. Some PaaS options also include the infrastructure, while"Pure PaaS" provides only the development tools.

SaaS (Software-as-a-Service): These are cloud-based applications, like web applications, that use the cloud to serve users. It's what most people use on the internet.

The reference model described above also introduces the concept of **Everything as a Service (XaaS)**. Different cloud services can be combined to create a complete system. For example, you can use virtual machines from one service and add your software to run applications. This is useful for startupsbecause it reduces the cost of starting a business and allows them to grow their infrastructure according to their revenues.

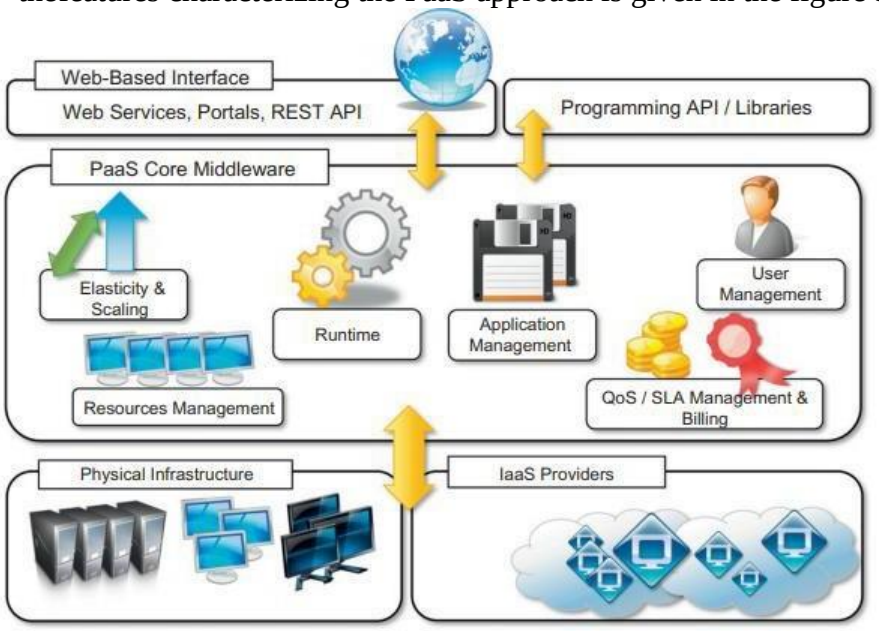
2) **Write a note on PaaS?**

Platform-as-a-Service (PaaS) solutions provide a development and deployment platform for running applications in the cloud. PaaS consumers or developers can consume language runtimes, application frameworks, databases, message queues, testing tools, and deployment tools as a service over the

Internet. Thus, it reduces the complexity of buying and maintaining different tools for developing an application.

Implementation

PaaS solutions constitute the middleware on top of which applications are built. A general overview of thefeatures characterizing the PaaS approach is given in the figure below:



The implementation of PaaS involves 3 layers: the *user interface*, the *middleware*, and the *physical infrastructure.*

The user interface layer: This layer allows programming and deploying applications on the cloud. These can be in the form of a Web-based interface or in the form of programming APIs and libraries. The user interface is responsible for presenting the application's interface to users, allowing them to access, configure, and manage the software running on the cloud platform.

The middleware layer: Application management is the core functionality of the middleware. PaaS implementations provide applications with a runtime environment and do not expose any service for managing the underlying infrastructure. They automate the process of deploying applications to the infrastructure, configuring application components, provisioning and configuring supporting technologies such as load balancers and databases, and managing system change based on policies set by the user. Developers design their systems in terms of applications and are not concerned with hardware (physical or virtual), operating systems, and other low-level services. The core middleware isin charge of managing the resources and scaling applications on demand or automatically, according tothe commitments made with users.

The Physical infrastructure layer: There are two ways of using infrastructure:

PaaS provides the tools or software to build the application and it provides the infrastructure to run the application.

PaaS provides the tools or software to build the application and the user must install it on his premises. This is known as Pure PaaS.

Categories of PaaS:

The various PaaS solutions can be categorized into three types: PaaS-I, PaaS-II, and PaaS-III.

PaaS-I: These follow the full cloud computing approach. They provide a web-based environment for creating and deploying applications.

Example: Force.com and Longjump, which offer both middleware and infrastructure.

PaaS-II: These focus on scalable infrastructure for web applications, like websites. Developers use the

provider's APIs to build apps.

Google AppEngine is a popular example, offering a scalable runtime for Java and Python. AppScale is an open-source version of it.

Joyent Smart Platform offers similar features.

Heroku and Engine Yard help with scalability for Ruby and Ruby on Rails-based websites.

PaaS-III: These solutions provide cloud programming platforms for various applications, not just web apps.

Microsoft Windows Azure is a well-known example, offering tools for building service-oriented cloud applications using .NET technology.

Other options in this category include Manjrasoft Aneka, Apprenda SaaSGrid, Appistry Cloud IQ Platform, DataSynapse, and GigaSpaces DataGrid, which offer middleware with different services.

Characteristics of PaaS:

- **Runtime Framework**: This is like the software foundation of PaaS. It's what runs your programsbased on your settings.
- **Abstraction**: PaaS makes things simpler. Instead of dealing with just virtual machines, it focuseson how applications work in the cloud.
- **Automation**: PaaS automatically does tasks like putting your application in the cloud and addingmore resources when needed, based on agreements between users and providers.
- **Cloud Services**: PaaS offers tools and shortcuts for developers to make cloud applications easily.These tools help create flexible and reliable cloud applications. Different PaaS options have different tools and features.

Suitability of PaaS

- **Collaborative Development**: When a project needs a place where developers and others can work together efficiently, PaaS provides a collaborative environment. It's great for projects where teamwork is essential.
- **Automated Testing and Deployment**: PaaS offers tools for automated testing and deploying applications. This speeds up development by reducing the time spent on manual testing. It's the right choice when you want to focus more on building and less on testing and deploying.
- **Quick Time to Market**: PaaS follows development methods that get your application to market fast. For example, it's perfect for agile development. If you want your software out there as soon as possible, PaaS is the way to go.

Advantages of PaaS

- **Quick Development and Deployment**: PaaS offers tools for developing, testing, and deploying software in one place. It automates testing and deployment, making it faster than traditional methods.
- **Reduces Costs**: PaaS eliminates the need to buy expensive development and testing tools. It also doesn't require high-end infrastructure, reducing costs for development companies.
- **Supports Agile Development**: Nowadays, most of the new generation applications are developed using agile methodologies. PaaS supports agile methodologies, which are popular for modern app development.
- **Collaborative Work**: PaaS allows teams from different places to collaborate on projects using an online platform.
- **User-Friendly**: PaaS offers various tools like web interfaces and APIs, making it easier for developers of all kinds to use.
- **Less Maintenance**: PaaS providers handle the underlying hardware and maintenance, freeing developers from that responsibility.
- **Scalability**: PaaS provides built-in scalability for web and SaaS applications, which can handle extra load without the need for additional servers.

Drawbacks of PaaS

- **Vendor Lock-in**: One major drawback is that PaaS providers may lock you into their services. This happens because there are no common standards among different providers, and they often use technologies that are specific to them. This can make it hard to switch between

providers.

- **Security Concerns**: Security is a big concern in PaaS, as data is stored on third-party servers. While many providers have security measures in place, some developers worry about safety compared to on-premises solutions. It's important for developers to carefully review the security policies of their chosen PaaS provider.
- **Limited Flexibility**: PaaS platforms may not allow developers to fully customize their application stack. While they offer various programming languages and tools, they may not cover all needs. Only a few PaaS providers allow extending tools with custom languages.
- **Internet Dependency**: Using PaaS requires a stable internet connection. While some providers allow offline access, most rely on being online. Slow internet can affect the usability and efficiency of the platform.

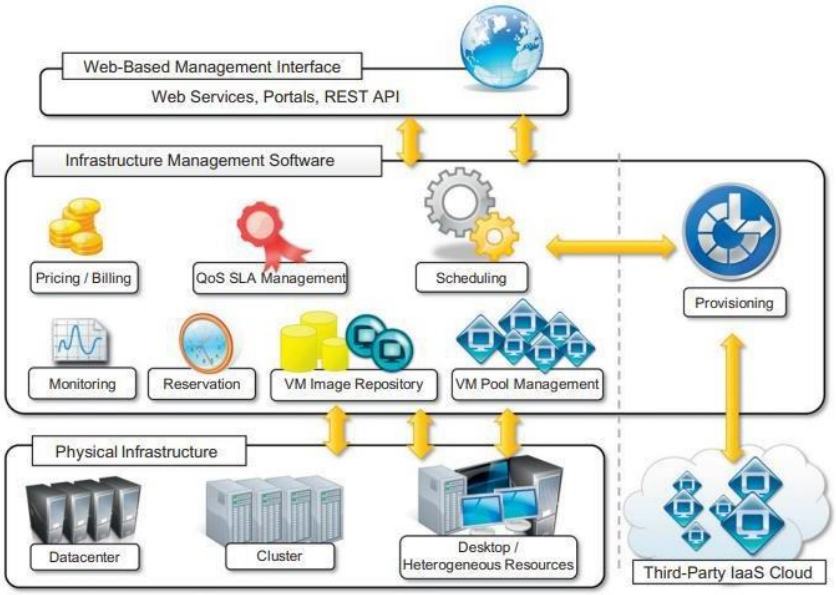
3) Write a note on IaaS?

Infrastructure- and Hardware-as-a-Service (IaaS/HaaS) solutions are the most popular and developed market segment of cloud computing. They deliver customizable infrastructure on demand.

IaaS changes computing from a physical infrastructure to a virtual infrastructure. IaaS provides virtual computing, storage, and network resources by abstracting physical resources. Technology virtualization is used to provide virtual resources. All the virtual resources are given to the virtual machines (VMs) that are configured by the service provider. The end users or IT architects will use the infrastructure resources in the form of VMs.

Implementation:

The implementation of IaaS involves 3 layers: the *user interface*, the *software management infrastructure*,and the *physical infrastructure.*



The user interface layer: At the top layer the user interface provides access to the services exposed by the software management infrastructure. Such an interface is generally based on Web 2.0 technologies: Web services, RESTful APIs, and mash-ups. These technologies allow either applications or final users to access the services exposed by the underlying infrastructure. Web 2.0 applications allow developing full-featured management consoles completely hosted in a browser or a web page. Web services and RESTful APIs allow programs to interact with the service without human intervention, thus providing complete integration within a software system.

The Software Management Infrastructure layer: A central role is played by the scheduler, which is in charge of allocating the execution of virtual machine instances. The scheduler interacts with the other components that perform a variety of tasks:

• The ***pricing and billing*** component takes care of the cost of executing each virtual machine instance

and maintains data that will be used to charge the user.
• The ***monitoring*** component tracks the execution of each virtual machine instance and maintains data required for reporting and analyzing the performance of the system.
• The ***reservation*** component stores the information of all the virtual machine instances that have been executed or that will be executed in the future.
• The ***QoS /SLA management component*** is used to ensure that a given virtual machine instance isexecuted with the desired quality of service.
• The ***VM repository component*** provides a catalog of virtual machine images that users can use tocreate virtual instances.
• A ***VM pool manager component*** is responsible for keeping track of all the live instances.

The physical infrastructure layer: The bottom layer is composed of the physical infrastructure, on top of which the management layer operates. The specific infrastructure used depends on the specific use of the cloud. A service provider will most likely use a massive data centre containing hundreds or thousands of nodes. A cloud infrastructure developed in house, in a small or medium-sized enterprise or within a university department, will most likely rely on a cluster. At the bottom of the scale, it is also possible to consider a heterogeneous environment where different types of resources—PCs, workstations, and clusters—can be aggregated.

Characteristics of IaaS:

- **Web access to the resources:** The IaaS model enables the IT users to access infrastructureresources over the Internet.
- **Centralized management**: Even though the physical resources are distributed, the managementwill be from a single place.
- **Elasticity and dynamic scaling:** IaaS provides elastic services where the usage of resources canbe increased or decreased according to the requirements.
- **Shared infrastructure:** IaaS follows a one-to-many delivery model and allows multiple IT usersto share the same physical infrastructure.
- **Preconfigured VMs:** IaaS providers offer preconfigured VMs with operating systems (OSs),network configuration, etc.
- **Metered services:** IaaS allows the IT users to rent computing resources instead of buying them.

Suitability of IaaS

IaaS can be used in the following situations:

- **Unpredictable spikes in usage**: When there is a significant spike in usage of computing resources, IaaS is the best option for IT industries. When demand is very volatile, we cannot predict the spikes and troughs in terms of demand of the infrastructure. In this situation, we cannot add or remove infrastructure immediately according to the demand in a traditional infrastructure. If there is an unpredictable demand for infrastructure, then it is recommended to use IaaS services.
- **Limited capital investment:** New start-up companies cannot invest more on buying infrastructure for their business needs. And so, by using IaaS, start-up companies can reduce the capital investment on hardware. IaaS is the suitable option for start-up companies with less capital investment in hardware.
- **Infrastructure on demand**: Some organizations may require large infrastructure for a short period of time. For this purpose, an organization cannot afford to buy more on-premises resources. Instead, they can rent the required infrastructure for a specific period of time. IaaS best suits the organizations that look for infrastructure on demand or for a short time period.

Advantages of IaaS:

- **Pay-as-you-use model:** The IaaS services are provided to the customers on a pay-per-use basis. This ensures that the customers are required to pay for what they have used. This model eliminates the unnecessary spending on buying hardware.
- **Reduced TCO (Total Cost of Ownership)**: Since IaaS providers allow the IT users to rent the

computing resources, they need not buy physical hardware for running their business. The IT users can rent the IT infrastructure rather than buy it by spending large amount. IaaS reducesthe need for buying hardware resources and thus reduces the TCO.

- **Elastic resources**: IaaS provides resources based on the current needs. IT users can scale up orscale down the resources whenever they want. This dynamic scaling is done automatically using some load balancers. This load balancer transfers the additional resource request to the new server and improves application efficiency.
- **Better resource utilization:** Resource utilization is the most important criteria to succeed inthe IT business. The purchased infrastructure should be utilized properly to increase the ROI(Return On Investment). IaaS ensures better resource utilization and provides high ROI for IaaS providers.

Drawbacks of IaaS

- **Security issues**: Since IaaS uses virtualization as the enabling technology, hypervisors play an important role. There are many attacks that target the hypervisors to compromise it. If hypervisors get compromised, then any VMs can be attacked easily. Most of the IaaS providers are not able to provide 100% security to the VMs and the data stored on the VMs.
- **Interoperability issues:** There are no common standards followed among the different IaaS providers. It is very difficult to migrate any VM from one IaaS provider to the other. Sometimes, the customers might face the vendor lock-in problem.
- **Performance issues:** IaaS is nothing but the consolidation of available resources from the distributed cloud servers. Here, all the distributed servers are connected over the network. The latency of the network plays an important role in deciding the performance. Because of latency issues, sometimes the VM contains issues with its performance.

4) Write a note on SaaS?

Software-as-a-Service (SaaS) is a software delivery model that provides access to applications through the Internet as a Web-based service. In the traditional software model, the software is delivered as a license-based product that needs to be installed in the end user device. Since SaaS is delivered as an on-demand service over the Internet, there is no need to install the software to the end user's devices. SaaS services can be accessed or disconnected at any time based on the end user's needs. SaaS services can be accessed from any lightweight web browsers on any devices such as laptops, tablets, and smartphones.

The SaaS model is popular for applications that can serve a broad user base and can be customized easily. This makes SaaS a "one-to-many" software delivery model, ideal for applications like CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning) used by various businesses. While basic requirements are the same, customization can address different needs. SaaS allows for hosted solutions where users receive the same applications and can tailor them to their requirements. This approach is naturally multitenant, benefiting both providers by streamlining management and users by reducing costs relative to software usage fees.

Characteristics of SaaS

- **One-to-many:** SaaS services are delivered as a one-to-many model where a single instance of the application can be shared by multiple tenants or customers.
- **Web access:** SaaS services provide web access to the software. It allows the end user to access the application from any location if the device is connected to the Internet.
- **Centralized management:** Since SaaS services are hosted and managed from the central location, management of the SaaS application becomes easier. Normally, the SaaS providers will perform automatic updates that ensure that each tenant is accessing the most recent version of the application without any user-side updates.
- **Multidevice support:** SaaS services can be accessed from any end user devices such as desktops, laptops, tablets and smartphones.
- **Better scalability:** Since most of the SaaS services leverage PaaS and IaaS for its development and deployment, it ensures a better scalability than the traditional software. The dynamic scaling of underlying cloud resources makes SaaS applications work efficiently even with varying

loads.

- **High availability**: SaaS services ensure the 99.99% availability of user data as proper backup and recovery mechanisms are implemented at the back end.
- **API integration:** SaaS services have the capability of integrating with other software or services through standard APIs.

Suitability of SaaS:

- **On-demand software**: The licensing-based software model requires buying full packaged software and increases the spending on buying software. If the end users are looking for on-demand software rather than the licensing-based full-term software, then the SaaS model is the best option.
- **Software for start-up companies:** When using any traditional software, the end user should buy devices with minimum requirements specified by the software vendor. This increases the investment in buying hardware for start-up companies. Since SaaS services do not require high-end infrastructure for accessing, it is a suitable option for start-up companies that can reduce the initial expenditure on buying high-end hardware.
- **Software compatible with multiple devices**: Some of the applications like word processors or mail services need better accessibility from different devices. The SaaS applications are adaptable with almost all devices.
- **Software with varying loads:** We cannot predict the load on popular applications such as social networking sites. The user may connect or disconnect from applications anytime. It isvery difficult to handle varying loads with the traditional infrastructure. With the dynamic scaling capabilities, SaaS applications can handle varying loads efficiently without disruptingthe normal behavior of the application

Advantages of SaaS

- **No client-side installation:** SaaS services do not require client-side installation of the software. The end users can access the services directly from the service provider data center without any installation. There is no need for high-end hardware to consume services. It can be accessed from thin clients or any handheld devices, thus reducing the initial expenditure on buying high-end hardware.
- **Cost savings:** Since SaaS services follow the utility-based billing or pay-as-you-go billing, it demands the end users to pay for what they have used. Most of the SaaS providers offer different subscription plans to benefit different customers. Sometimes, the generic SaaS services such as word processors are given for free to the end users.
- **Less maintenance:** SaaS services eliminate the additional overhead of maintaining the software from the client side. For example, in the traditional software, the end user is responsible for performing bulk updates. But in SaaS, the service provider itself maintains the automatic updates, monitoring, and other maintenance activities of the applications.
- **Ease of access:** SaaS services can be accessed from any device if it is connected to the Internet. Accessibility of SaaS services is not restricted to any particular device. It is adaptable to all the devices as it uses the responsive web UI.
- **Dynamic scaling:** SaaS services are popularly known for elastic dynamic scaling. It is very difficult for on-premises software to provide dynamic scaling capability as it requires additional hardware. Since the SaaS services leverage elastic resources provided by cloud computing, it can handle any type of varying loads without disrupting the normal behavior of the application.
- **Disaster recovery:** With proper backup and recovery mechanisms, replicas are maintained for every SaaS service. The replicas are distributed across many servers. If any server fails, the end user can access the SaaS from other servers. It eliminates the problem of single point of failure. It also ensures the high availability of the application.
- **Multitenancy:** Multitenancy is the ability given to the end users to share a single instance of the application. Multitenancy increases resource utilization from the service provider side.

Drawbacks of SaaS

- **Security:** Security is the major concern in migrating to SaaS application. Since the SaaS application is shared between many end users, there is a possibility of data leakage. Here, the data is stored in the service provider data centre. The end user should be careful while selecting the SaaS provider to avoid unnecessary data loss.
- **Connectivity requirements:** SaaS applications require Internet connectivity for accessing it. Sometimes, the end user's Internet connectivity might be very slow. In such situations, the user cannot access the services with ease. The dependency on high-speed Internet connection is a major problem in SaaS applications.
- **Loss of control:** Since the data is stored in a third-party and off premise location, the end user does not have any control over the data. The degree of control over the SaaS application and data is lesser than the on-premises application.

5) Explain different characteristics of virtualization.

1. **Increased Security** –

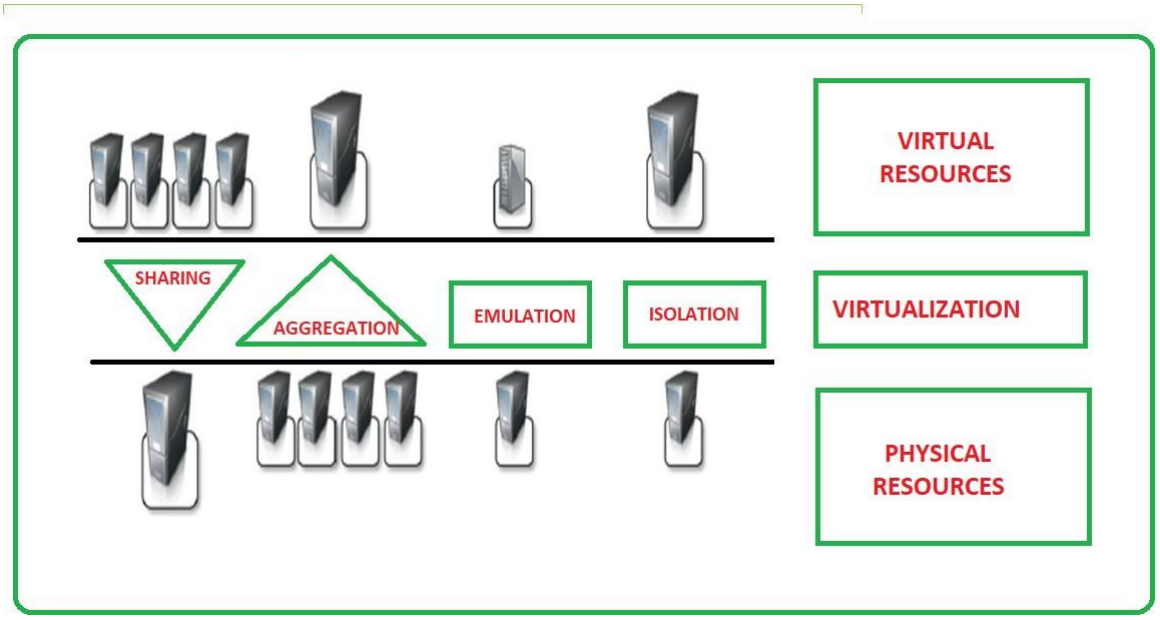
The ability to control the execution of a guest program in a completely transparent manner opens new possibilities for delivering a secure, controlled execution environment. All the operations of the guest programs are generally performed against the virtual machine, which then translates and applies them to the host programs.

A virtual machine manager can control and filter the activity of the guest programs, thus preventing some harmful operations from being performed. Resources exposed by the host can then be hidden or simply protected from the guest. Increased security is a requirement when dealing with untrusted code.

Example-1: Untrusted code can be analyzed in Cuckoo sandboxes environment. The term sandbox identifies an isolated execution environment where instructions can be filtered and blocked before being translated and executed in the real execution environment.

2. **Managed Execution** –

In particular, sharing, aggregation, emulation, and isolation are the most relevant features.



a) **Sharing** –

Virtualization allows the creation of a separate computing environment within the same host. This basic feature is used to reduce the number of active servers and limit power consumption.

b) **Aggregation** –

It is possible to share physical resources among several guests, but virtualization also allows aggregation, which is the opposite process. A group of separate hosts can be tied together and represented to guests as a single virtual host. This functionality is implemented with cluster management software, which harnesses the physical resources of a homogeneous group of machines and represents them as a single resource.

c) **Emulation** –

Guest programs are executed within an environment that is controlled by the virtualization layer, which ultimately is a program. Also, a completely different environment with respect to the host can be emulated, thus allowing the execution of guest programs requiring specific characteristics that are not present in the physical host.

d) **Isolation** –

Virtualization allows providing guests—whether they are operating systems, applications, or other entities—with a completely separate environment, in which they are executed. The guest program performs its activity by interacting with an abstraction layer, which provides access to the underlying resources. The virtual machine can filter the activity of the guest and prevent harmful operations against the host.

Besides these characteristics, another important capability enabled by virtualization is performance tuning. This feature is a reality at present, given the considerable advances in hardware and software supporting virtualization. It becomes easier to control the performance of the guest by finely tuning the properties of the resources exposed through the virtual environment. This capability provides a means to effectively implement a quality-of-service (QoS) infrastructure.

3. **Portability** –

The concept of portability applies in different ways according to the specific type of virtualization considered.

In the case of a hardware virtualization solution, the guest is packaged into a virtual image that, in most cases, can be safely moved and executed on top of different virtual machines. In the case of programming-level virtualization, as implemented by the JVM or the .NET runtime, the binary code representing application components (jars or assemblies) can run without any recompilation on any implementation of the corresponding virtual machine.

6) Explain different types of clouds.

The models for deploying and accessing cloud computing environments are

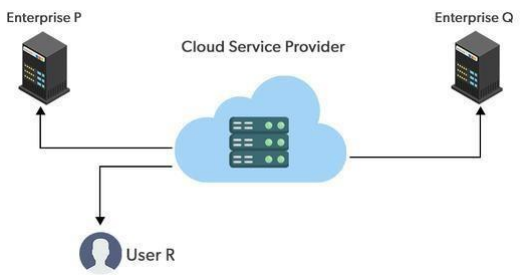
- Public clouds,
- Private/enterprise clouds, and
- Hybrid clouds
- Community clouds

1. Public clouds

The public cloud makes it possible for anybody to access systems and services. The public cloudmay be less secure as it is open to everyone. The public cloud is one in which cloud infrastructure services are provided over the internet to the general people or major industry groups.

The infrastructure in this cloud model is owned by the entity that delivers the cloud services, not by the consumer. It is a type of cloud hosting that allows customers and users to easily access systemsand services. This form of cloud computing is an excellent example of cloud hosting, in whichservice providers supply services to a variety of customers. In this arrangement, storage backup and retrieval services are given for free, as a subscription, or on a per-user basis.

For example, Google App Engine etc.



Advantages of the Public Cloud Model

- **Minimal Investment:** Because it is a pay-per-use service, there is no substantial upfront fee, making it excellent for enterprises that require immediate access to resources.
- **No setup cost:** The entire infrastructure is fully subsidized by the cloud service providers,

thus there is no need to set up any hardware.

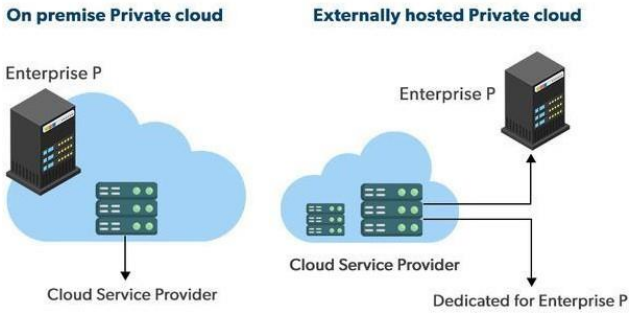
- **Infrastructure Management is not required:** Using the public cloud does not necessitate infrastructure management.
- **No maintenance:** The maintenance work is done by the service provider (not users).
- **Dynamic Scalability:** To fulfill your company's needs, on-demand resources are accessible.

Disadvantages of the Public Cloud Model

- **Data Security and Privacy Concerns** - Since it is accessible to all, it does not fully protect against cyber-attacks and could lead to vulnerabilities.
- **Reliability Issues** - Since the same server network is open to a wide range of users, it can lead to malfunction and outages
- **Service/License Limitation** - While there are many resources you can exchange with tenants, there is a usage cap.

2. Private cloud

The private cloud deployment model is the exact opposite of the public cloud deployment model. It's a one-on-one environment for a single user (customer). There is no need to share your hardware with anyone else. The distinction between private and public clouds is in how you handle all of the hardware. It is also called the "internal cloud" & it refers to the ability to access systems and services within a given border or organization. The cloud platform is implemented in a cloud-based secure environment that is protected by powerful firewalls and under the supervision of an organization's IT department. The private cloud gives greater flexibility of control over cloud resources.



Advantages of the Private Cloud Model

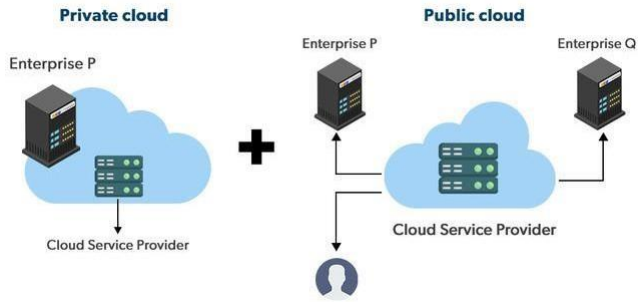
- **Better Control:** You are the sole owner of the property. You gain complete command over service integration, IT operations, policies, and user behavior.
- **Data Security and Privacy:** It's suitable for storing corporate information to which only authorized staff have access. By segmenting resources within the same infrastructure, improved access and security can be achieved.
- **Supports Legacy Systems:** This approach is designed to work with legacy systems that are unable to access the public cloud.
- **Customization:** Unlike a public cloud deployment, a private cloud allows a company to tailor its solution to meet its specific needs.

Disadvantages of the Private Cloud Model

- **Less scalable:** Private clouds are scaled within a certain range as there is less number of clients.
- **Costly:** Private clouds are more costly as they provide personalized facilities.

3. Hybrid Cloud

By bridging the public and private worlds with a layer of proprietary software, hybrid cloud computing gives the best of both worlds. With a hybrid solution, you may host the app in a safe environment while taking advantage of the public cloud's cost savings. Organizations can move data and applications between different clouds using a combination of two or more cloud deployment methods, depending on their needs.



Advantages of the Hybrid Cloud Model

- **Flexibility and control:** Businesses with more flexibility can design personalized solutions that meet their particular needs.
- **Cost:** Because public clouds provide scalability, you'll only be responsible for paying for the extra capacity if you require it.
- **Security:** Because data is properly separated, the chances of data theft by attackers are considerably reduced.

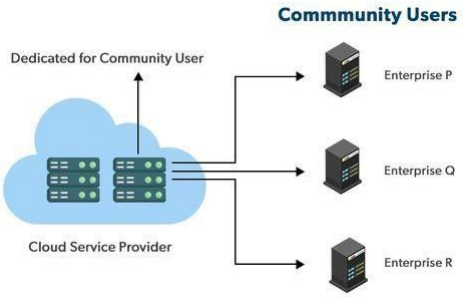
Disadvantages of the Hybrid Cloud Model

- **Difficult to manage:** Hybrid clouds are difficult to manage as it is a combination of both public and private cloud. So, it is complex.
- **Slow data transmission:** Data transmission in the hybrid cloud takes place through thepublic cloud so latency occurs.

4. Community Cloud

It allows systems and services to be accessible by a group of organizations. It is a distributed system that is created by integrating the services of different clouds to address the specific needs of a community, industry, or business.

The infrastructure of the community could be shared between the organization which has shared concerns or tasks. It is generally managed by a third party or by the combination of one or more organizations in the community.



Advantages of the Community Cloud Model

- **Cost Effective:** It is cost-effective because the cloud is shared by multiple organizations or communities.
- **Security:** Community cloud provides better security.
- **Shared resources:** It allows you to share resources, infrastructure, etc. with multiple organizations.
- **Collaboration and data sharing:** It is suitable for both collaboration and data sharing.

Disadvantages of the Community Cloud Model

- **Limited Scalability:** Community cloud is relatively less scalable as many organizations share the same resources according to their collaborative interests.
- **Rigid in customization:** As the data and resources are shared among different organizations according to their mutual interests if an organization wants some changes according to their needs they cannot do so because it will have an impact on other organizations.

7) Explain any three types of virtualization.

a. Server Virtualization

Application server virtualization abstracts a collection of application servers that provide the

same services as a single virtual application server by using load-balancing strategies and providing a high-availability infrastructure for the services hosted in the application server. This is a particular form of virtualization and serves the same purpose of storage virtualization: providing a better quality of service rather than emulating a different environment.

b. Application Virtualization

Application virtualization is a technique allowing applications to be run in runtime environments that do not natively support all the features required by such applications. These techniques are mostly concerned with partial file systems, libraries, and operating system component emulation. Such emulation is performed by a thin layer—a program or an operating system component—that is in charge of executing the application. Emulation can also be used to execute program binaries compiled for different hardware architectures. In this case, one of the following strategies can be implemented:

• ***Interpretation***: In this technique every source instruction is interpreted by an emulator for executing native ISA instructions, leading to poor performance. Interpretation has a minimal startup cost but a huge overhead, since each instruction is emulated.

• ***Binary translation***: In this technique every source instruction is converted to native instructions with equivalent functions. After a block of instructions is translated, it is cached and reused. Binary translation has a large initial overhead cost, but over time it is subject to better performance, since previously translated instruction blocks are directly executed.

One of the most popular solutions implementing application virtualization is Wine, which is a software application allowing Unix-like operating systems to execute programs written for the Microsoft Windows platform. Wine features a software application acting as a container for the guest application and a set of libraries, called Winelib, that developers can use to compile applications to be ported on Unix systems.

c. Storage Virtualization

Storage virtualization is a system administration practice that allows decoupling the physical organization of the hardware from its logical representation. Using this technique, users do not have to be worried about the specific location of their data, which can be identified using a logical path. There are different techniques for storage virtualization, one of the most popular being network-based virtualization by means of storage area networks (SANs). SANs use a network-accessible device through a large bandwidth connection to provide storage facilities.

8) Explain different advantages and disadvantages of virtualization.

Advantages of Virtualization in Cloud Computing :

- **Utilization of Hardware Efficiently –** With the help of Virtualization Hardware is Efficiently used by user as well as Cloud Service Provider. In this the need of Physical Hardware System for the User is decreases and this results in less costly. In Service Provider point of View, they will vitalize the Hardware using Hardware Virtualization which decrease the Hardware requirement from Vendor side which are provided to User is decreased. Before Virtualization, Companies and organizations have to set up their own Server which require extra space for placing them, engineer's to check its performance and require extra hardware cost but with the help of Virtualization the all these limitations are removed by Cloud vendor's who provide Physical Services without setting up any Physical Hardware system.
- **Availability increases with Virtualization –** One of the main benefit of Virtualization is that it provides advance features which allow virtual instances to be available all the times. It also has capability to move virtual instance from one virtual Server another Server which is very tedious and risky task in Server Based System. During migration of Data from one server to another it ensures its safety. Also, we can access information from any location and any time from anydevice.
- **Disaster Recovery is efficient and easy –** With the help of virtualization Data Recovery, Backup, Duplication becomes very easy. In traditional method, if somehow due to some disaster if Server system Damaged then the surety of Data Recovery is very less. But with the tools of Virtualization real time data backup recovery and mirroring become easy task and provide suretyof zero percent data loss.
- **Virtualization saves Energy –** Virtualization will help to save Energy because while moving from physical Servers to Virtual Server's, the number of Server's decreases due to this monthly

power and cooling cost decreases which will Save Money as well. As cooling cost reduces it means carbon production by devices also decreases which results in Fresh and pollution free environment.

- **Quick and Easy Set up –** In traditional methods Setting up physical system and servers are very time-consuming. Firstly Purchase them in bulk after that wait for shipment. When Shipment is done then wait for Setting up and after that again spend time in installing required software etc. Which will consume very time. But with the help of virtualization the entire process is done in very less time which results in productive setup.
- **Cloud Migration becomes easy –** Most of the companies those who already have spent a lot in the server have a doubt of Shifting to Cloud. But it is more cost-effective to shift to cloud services because all the data that is present in their server's can be easily migrated into the cloud server and save something from maintenance charge, power consumption, cooling cost, cost to Server Maintenance Engineer etc.

Disadvantages of Virtualization :

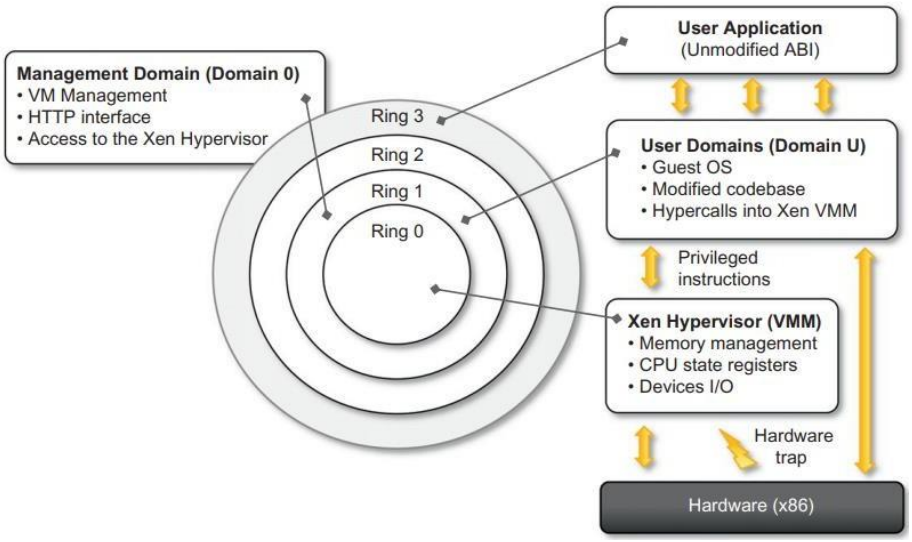
- **Data can be at Risk –** Working on virtual instances on shared resources means that our data is hosted on third party resource which put's our data in vulnerable condition. Any hacker can attack on our data or try to perform unauthorized access. Without Security solution our data is in threaten situation**.**
- **Learning New Infrastructure –** As Organization shifted from Servers to Cloud. They required skilled staff who can work with cloud easily. Either they hire new IT staff with relevant skill or provide training on that skill which increase the cost of company.
- **High Initial Investment –** It is true that Virtualization will reduce the cost of companies but also it is truth that Cloud have high initial investment. It provides numerous services which are not required and when unskilled organization will try to set up in cloud they purchase unnecessary services which are not even required to them.

9) Explain Xen architecture with diagram.

Xen is an open-source initiative implementing a virtualization platform based onparavirtualization. Initially developed by a group of researchers at the University of Cambridge in the United Kingdom, Xen now has a large open-source community backing it. Citrix also offers it as a commercial solution, XenSource.

Xen-based technology is used for either desktop virtualization or server virtualization, and recently it has also been used to provide cloud computing solutions by means of Xen Cloud Platform (XCP). At the basis of all these solutions is the Xen Hypervisor, which constitutes the core technology of Xen. Xen is the most popular implementation of paravirtualization, which, in contrast with full virtualization, allows high-performance execution of guest operating systems. This is made possible by eliminating the performance loss while executing instructions that require special management. This is done by modifying portions of the guest operating systems run by Xen with reference to the execution of such instructions. Therefore, it is not a transparent solution for implementing virtualization.

A Xen-based system is managed by the Xen hypervisor, which runs in the highest privileged mode and controls the access of guest operating system to the underlying hardware. Guest operating systems are executed within domains, which represent virtual machine instances. Moreover, specific control software, which has privileged access to the host and controls all the other guest operating systems, is executed in a special domain called Domain 0. This is the first one that is loaded once the virtual machine manager has completely booted, and it hosts a HyperText TransferProtocol (HTTP) server that serves requests for virtual machine creation, configuration, and termination.

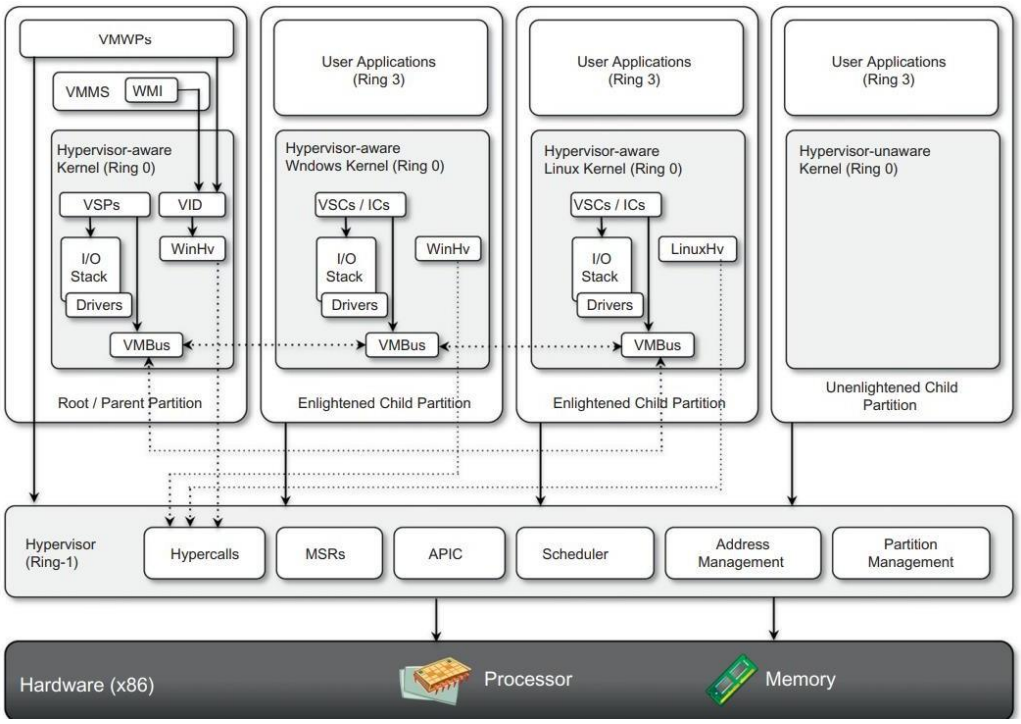


10) Write a note on Microsoft Hyper-V.

Hyper-V is an infrastructure virtualization solution developed by Microsoft for server virtualization. As the name recalls, it uses a hypervisor-based approach to hardware virtualization, which leverages several techniques to support a variety of guest operating systems.

a) **Architecture**: Hyper-V supports multiple and concurrent execution of guest operating systems by means of partitions. A partition is a completely isolated environment in which an operating system is installed and run. Hyper-V takes control of the hardware, and the host operating system becomes a virtual machine instance with special privileges, called the parent partition. The parent partition (also called the root partition) is the only one that has direct access to the hardware. It runs the virtualization stack, hosts all the drivers required to configure guest operating systems, and creates child partitions through the hypervisor. Child partitions are used to host guest operating systems and do not have access to the underlying hardware, but their interaction with it is controlled by either the parent partition or the hypervisor itself. Hypervisor: The hypervisor is the component that directly manages the underlying hardware(processors and memory). It is logically defined by the following components:

- Hypercalls interface: This is the entry point for all the partitions for the execution of sensitive instructions. This interface is used by drivers in the partitioned operating system to contact the hypervisor using the standard Windows calling convention. The parent partition also uses this interface to create child partitions.
- Memory service routines (MSRs): These are the set of functionalities that control thememory and its access from partitions. By leveraging hardware-assisted virtualization, the hypervisor uses the Input/Output Memory Management Unit to fast-track access to devices from partitions by translating virtual memory addresses.
- Advanced programmable interrupt controller (APIC): This component represents the interrupt controller, which manages the signals coming from the underlying hardware when some event occurs. The hypervisor is responsible of dispatching, when appropriate, the physical interrupts to the synthetic interrupt controllers.
- Scheduler: This component schedules the virtual processors to run on available physical processors. The scheduling is controlled by policies that are set by the parent partition.
- Address manager. This component is used to manage the virtual network addresses that are allocated to each guest operating system.
- Partition manager: This component is in charge of performing partition creation, finalization, destruction, enumeration, and configurations. Its services are available through the hyper calls interface API previously discussed.



11) **Write a note of virtualization and cloud computing.**

Virtualization plays an important role in cloud computing since it allows for the appropriate degree of customization, security, isolation, and manageability that are fundamental for delivering IT services ondemand. Virtualization technologies are primarily used to offer configurable computing environments and storage. The important is the role of virtual computing environment and execution virtualization techniques. Hardware virtualization is an enabling factor for solutions in the Infrastructure-as-a-Service (IaaS) market segment, while programming language virtualization is a technology leveraged in Platform-as-a-Service (PaaS) offerings. Virtualization allows us to create isolated and controllable environments; it is possible to serve these environments with the same resource without them interfering with each other. If the underlying resources are capable enough,

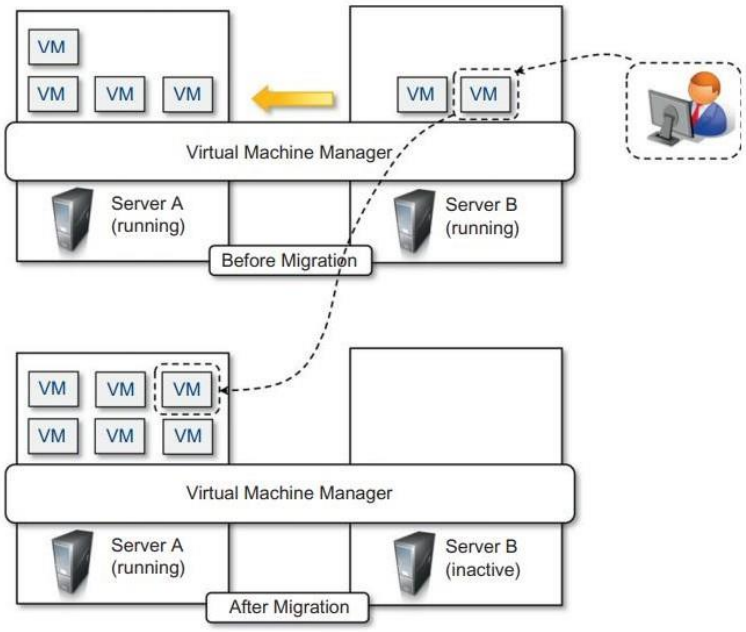


FIGURE 3.10

Live migration and server consolidation.

there will be no evidenceof such sharing. This opportunity is particularly attractive when resources are underutilized, because it allows reducing the number of active resources by aggregating virtual machines over a smaller numberof resources that become fully utilized. This practice is also known as server consolidation, while the movement of virtual machine instances is called virtual machine migration. This second techniques is known as live migration and in general is more complex to implement but more efficient since there is no disruption of the activity of the virtual machine instance.

Server consolidation and virtual machine migration are principally used in the case of hardware virtualization, even though they are also technically possible in the case of programming language virtualization. Storage virtualization constitutes an interesting opportunity given by virtualization technologies, often complementary to the execution of virtualization.

12) Write a note on hypervisor.

A fundamental element of hardware virtualization is the hypervisor, or virtual machine manager (VMM). It recreates a hardware environment in which guest operating systems are installed. There are two major types of hypervisors: Type I and Type II

- Type I hypervisors run directly on top of the hardware. Therefore, they take the place ofthe operating systems and interact directly with the ISA interface exposed by the underlying hardware, and they emulate this interface inorder to allow the management of guest operating systems. This type of hypervisor is also called a native virtual machine since it runs natively on hardware
- Type II hypervisors require the support of an operating system to provide virtualization services. This means that they are programs managed by the operating system, which interact with it through the ABI and emulate the ISA of virtual hardware for guest operating systems. This type of hypervisor is also called a hosted virtual machine since it is hosted within an operating system.

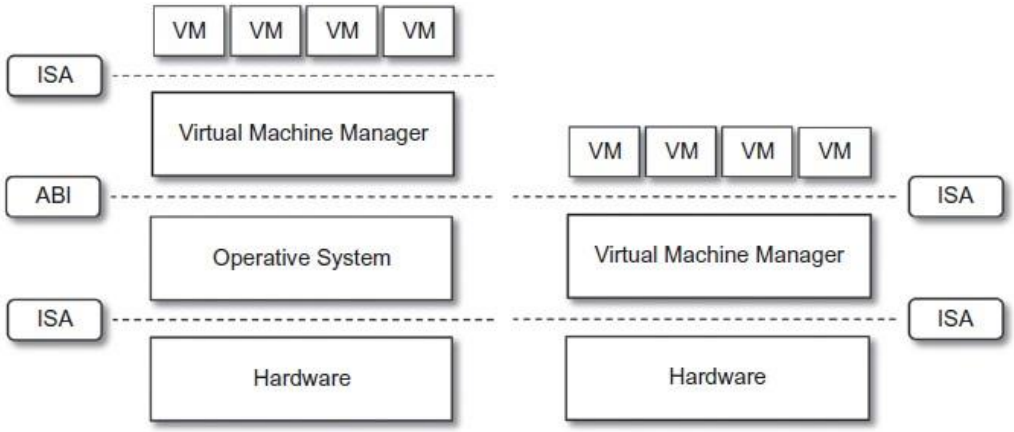


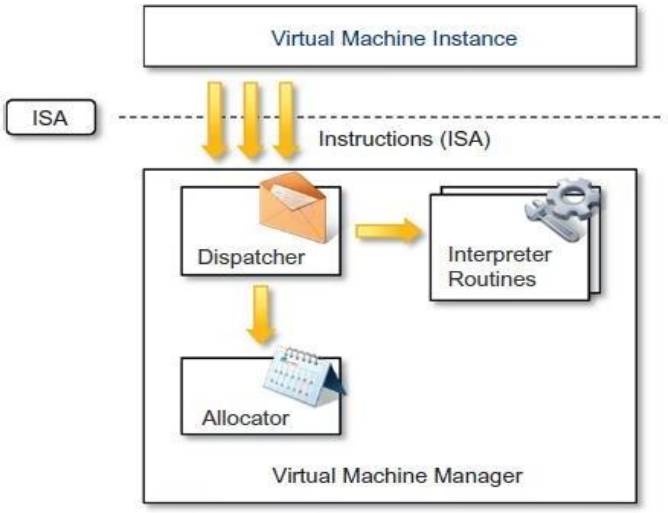
FIGURE 3.7

Hosted (left) and native (right) virtual machines. This figure provides a graphical representation of the two types of hypervisors.

Three main modules, dispatcher, allocator, and interpreter, coordinate their activity in order to emulate the underlying hardware. The dispatcher constitutes the entry point of the monitor and reroutes the instructions issued by the virtual machine instance to one of the two other modules. The allocator is responsible for deciding the system resources to be provided to the VM: whenever a virtual machine tries to execute an instruction that results in changing the machine resources associated with that VM, the allocator is invoked by the dispatcher. The interpreter module consists of interpreter routines. These are executed when ever a virtual machine executes a privileged instruction: atrap is triggered and the corresponding routine is executed. Three properties have to be satisfied:

- Equivalence. A guest running under the control of a virtual machine manager shouldexhibit the same behaviour as when it is executed directly on the physical host.
- Resource control. The virtual machine manager should be incomplete control ofvirtualized resources.

- Efficiency. A statistically dominant fraction of the machine instructions should beexecuted without the intervention from the virtual machine manager.



The major factor that determines whether these properties are satisfied is represented by the layout of the ISA of the host running a virtual machine manager. Popek and Goldberg provided a classification of the instruction set and proposed three theorems that define the properties that hardware instructions need to satisfy in order to efficiently support virtualization.

THEOREM 1 For any conventional third-generation computer, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.

THEOREM2 A conventional third-generation computer is recursively virtualizable if:

- It is virtualizable and
- A VMM without any timing dependencies can be constructed for it.

THEOREM3 A hybrid VMM may be constructed for any conventional third-generation machine in which the set of user-sensitive instructions is a subset of the set of privileged instructions.

UNIT 3

2 marks questions

1. **What is Aneka?**
   Aneka is Manjrasoft's solution for developing, deploying, and managing Cloud applications. Aneka is a software platform for developing Cloud computing applications
2. **What is Aneka container?**
   The Aneka container constitutes the building block of Aneka Clouds and represents the runtime machinery available to services and applications. The container is the unit of deployment in Aneka Clouds, and it is a lightweight software layer designed to host services and interact with the underlying operating system and hardware.
3. **List the services installed in the Aneka container**
   Fabric services , Foundation services, Application services

4. **Expand PAL & CLI.**
   PAL - Platform Abstraction Layer
   CLI - Common Language Infrastructure

5. **What is Platform Abstraction Layer (PAL)?**
   The Platform Abstraction Layer (PAL) addresses this heterogeneity and provides the container with a uniform interface for accessing the relevant hardware and operating system information, thus allowing the rest of the container to run unmodified on any platform supported.

6. **What is Common Language Infrastructure (CLI)?**
   The Common Language Infrastructure (CLI), which is the specification introduced in the ECMA-334 of standard defines a common runtime environment and application model for executing programs, but does not provide any interface to access the hardware or to collect performance data from the hosting operating system.

7. **List the services that are in charge of managing resources.**
   Index Service (or Membership Cata- logue), Reservation Service, and Resource Provisioning Service

8. **What is reporting service?**
   The Reporting Service manages the store for monitored data and makes them accessible to other services or external applications for analysis purposes.

9. **What are the 3 nodes used by the logical organization?**
   Master node, Worker node and Storage node

10. **Which services are hosted in Master node.**
    Index Service (master copy), Heartbeat Service, Logging Service, Reservation Service, Resource Provisioning Service, Accounting Service, Reporting and Monitoring Service, Scheduling Services for the supported programming models

11. **What are the different models supports by the Aneka clouds?**
    Aneka supports various deployment models for public, private, and hybrid clouds.

12. **What is Aneka daemon?**
    The infrastructure is deployed by harnessing a collection of nodes and installing on them the Aneka node manager, also called the *Aneka daemon*.

13. **What is Aneka repository?**

Aneka repository provides storage for all the libraries required to lay out and install the basic Aneka platform. These libraries constitute the software image for the node manager and the container programs. Repositories can make libraries available through a variety of communication channels, such as HTTP, FTP, common file sharing, and so on.

14. **What is private cloud deployment mode?**

A private deployment mode is mostly constituted by local physical resources and infrastructure management software providing access to a local pool of nodes, which mightbe virtualized.

15. **What is public cloud deployment mode?**

Public Cloud deployment mode features the installation of Aneka master and worker nodes over a completely virtualized infrastructure that is hosted on the infrastructure of one or more resource providers.

16. **What is hybrid cloud deployment mode?**

The hybrid deployment model constitutes the most common deployment of Aneka. In many cases, there is an existing computing infrastructure that can be leveraged to address the computing needs of applications.

17. **What are the categories of application models?**

a. Applications whose tasks are generated by the user
b. Applications whose tasks are generated by the runtime infrastructure

18. **What is service model?**

The Aneka Service Model defines the basic requirements to implement a service that can be hosted in an Aneka Cloud. The container defines the runtime environment in which services are hosted.

19. **Mention different features of Aneka management tools.**

Infrastructure management, Platform management, Application management

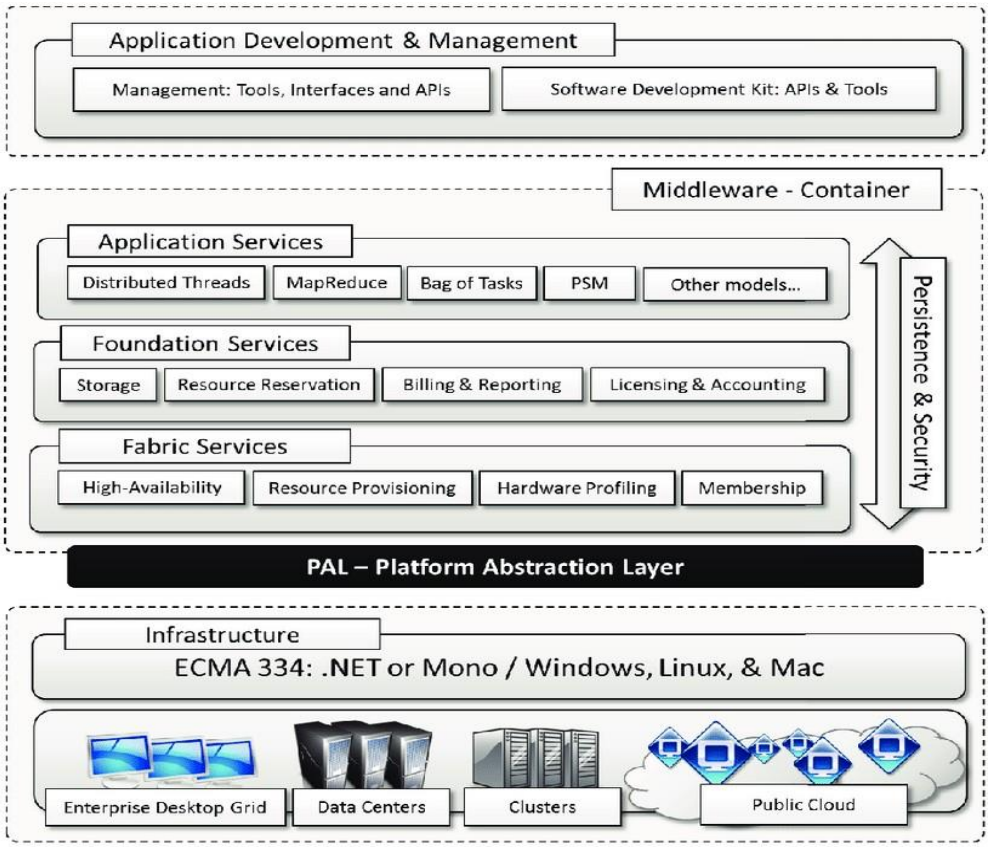
20. **What is infrastructure management?**

Aneka leverages virtual and physical hardware in orderto deploy Aneka Clouds. Virtual hardware is generally managed by means of the Resource Provisioning Service, which acquires resources on demand according to the need of applications, while physical hardware is directly managed by the Administrative Console by leveraging the Aneka management API of the PAL. The management features are mostly concerned with the provisioning of physical hardware and the remote installation of Aneka on the hardware.

Long answer questions

1. **Explain different components of Aneka framework.**

Aneka is a software platform for developing Cloud computing applications. According to the Cloud Reference Model , Aneka is a Pure PaaS solution for Cloud computing. Aneka is a Cloud middleware that can be deployed on a heterogeneous set of resources: a network of computers, a multi-core server, data centers, virtual Cloud infrastructures, or a mixture of them. The framework provides both a middleware for managing and scaling distributing applications and an extensible set of APIs for developing them.



This Figure provides a complete overview of the components of the framework. The core infrastructure of the system provides a uniform layer allowing the framework to be deployed over different platform and operating systems. The physical and virtual resourcesrepresenting the bare metal of the Cloud are managed by the Aneka container, which is installed on each node and constitutes the basic building block of the middleware. A collection of interconnected containers constitute the Aneka Cloud: a single domain in which services are made available to users and developers. The container features three different classes of services: Fabric Services, Foundation Services, and Execution Services. These respectively take care of infrastructure management, supporting services for the Cloud, and application management and execution. These services are made available to developers and administrators by the means of the application management and development layer, which includes interfaces and APIs for developing Cloud applications, and the management tools and interfaces for controlling Aneka Clouds.

Aneka implements a Service-Oriented Architecture (SOA), and services are the fundamental components of an Aneka Cloud. Services operate at container level and-except for theplatform abstraction layer-they provide developers, users, and administrators with all featuresoffered by the framework. Services also constitute the extension and customization point of Aneka Clouds: the infrastructure al- lows for the integration of new services or replacementof the existing ones with a different implementation. The framework includes the basic services for infrastructure and node management, application execution, accounting, and system monitoring; existing services can be extended and new features can be added to the Cloud by dynamically plugging new ones into the container. Such extensible and flexible infrastructure enables Aneka Clouds to support different programming and execution models for applications.

Within a Cloud environment, there are different aspects involved in providing a scalable and elastic infrastructure distributed runtime for applications. These services involve the following

(a) Elasticity and Scaling. With its dynamic provisioning service, Aneka supports dynami- cally up-sizing and down-sizing of the infrastructure available for applications.

(b) Runtime Management. The runtime machinery is responsible for keeping the infra- structure up and running, and serves as a hosting environment for services. It is primarily represented by the container and a collection of services managing service membership and lookup, infrastructure maintenance, and profiling.

(c) Resource Management. Aneka is an elastic infrastructure where resources are added andremoved

dynamically, according to the application needs and user requirements. In order to provide QoS based execution, the system not only allows dynamic provisioning, but also provides capabilities for reserving nodes for exclusive use by specific applications.

(d) Application Management. A specific subset of services is devoted to manage applications: these services include scheduling, execution, monitoring, and storage management.

(e) User Management. Aneka is a multi-tenant distributed environment where multiple applications, potentially belonging to different users, are executed. The framework provides an extensible user system where it is possible to define users, groups, and permissions. The services devoted to user management build up the security infrastructure of the system and constitute a fundamental element for the accounting management.

(f)QOS/SLA Management and Billing. Within a Cloud environment, application executionis metered and billed. Aneka provides a collection of services that coordinate together for taking into account the usage of resources by each application and billing the owning user accordingly.

All these services are available to specific interfaces and APIs, on top of which the software development kit and management kit are built.

2. Explain Fabric Services offered by Aneka Cloud Platform.

Fabric services define the lowest level of the software stack representing the Aneka Container. They provide access to the resource provisioning subsystem and to the monitoring facilities implemented in Aneka. Resource provisioning services are in charge of dynamicallyproviding new nodes on demand by relying on virtualization technologies, while monitoring services allows for hardware profiling and imple ment a basic monitoring infrastructure that can be used by all the services installed in the container.

1. Profiling and Monitoring

Profiling and monitoring services are mostly exposed through the Heartbeat, Monitoring, and Reporting services. The first makes available the information that is collected through the PAL, while the other two implement a generic infrastructure for monitoring the activity ofany service in the Aneka Cloud.

The Heartbeat service periodically collects the dynamic performance information about the node, and publishes this information to membership service in the Aneka Cloud. These data are collected by the index node of the Cloud, which makes them available for services suchas reservation and scheduling in order to optimize the usage of a heterogeneous infrastructure. The information published by the Heartbeat service is mostly concerned with the properties of the node. A specific component, called Node Resolver, is in charge of collecting these data and making them available to the Heartbeat service. For example, the retrieval of the public IP of the node is different in the case of physical machines or virtual instances hosted in the infrastructure of an laaS provider such as EC2 or GoGrid. In virtual deployment, different node resolver is used, so that all other components of the system can work transparently

The **Reporting Service** manages the store for monitored data and makes them accessible to other services or external applications for analysis purposes. On each node, an instance of the **Monitoring Service** acts as a gateway to the **Reporting Service** and forwards all the monitored data that has been collected on the node to it. Any service wanting to publish monitoring data can leverage the local monitoring service without knowing the details of the entire infrastructure.

Currently, several built-in services provide information through this channel:

- The Membership Catalogue tracks the performance information of nodes.
- The Execution Service monitors several time intervals for the execution of jobs.
- The Scheduling Service tracks the state transitions of jobs.
- The Storage Service monitors and makes available the information about datatransfer, such as upload and download times, file names, and sizes.
- The Resource Provisioning Service tracks the provisioning and lifetime informationof virtual nodes.

All these information can be stored on RDBMS or a flat file, and they can be further analyzedby specific Applications. For example, the management console provides a view on such datafor administrative purposes

2. Resource Management

Resource management is another fundamental feature of Aneka Clouds. It comprises several tasks: resource membership, resource reservation, and resource provisioning. Aneka provides a collection of services that are in charge of managing resources. These are: **Index Service (or Membership Catalogue), Reservation Service, and Resource Provisioning Service**.

a. The **Membership catalogue** is the fundamental component for resource management since it keeps track of the basic node information for all the nodes that are connected or disconnected.The **Member- ship Catalogue** implements the basic services of a directory service allowing the search for services by using attributes such as names, and nodes. During container startup,each instance publishes its information to the **Membership Catalogue** and updates it constantly during its lifetime. Services and external applications can query the membership catalogue in order to discover the available services and interact with them. In order to speed up and enhance the performance of queries, the membership catalogue works is organized asa distributed database. All the queries that pertain information main- tained locally are resolved locally, otherwise, the query is forwarded to the main index node, which has a global knowledge of the entire Cloud.

b. **Dynamic resource provisioning** allows the integration and the management, of virtual resources leased from laaS providers into the Aneka Cloud. Aneka defines a very flexible infrastructure for resource provisioning where it is possible to change the logic that triggers provisioning, support several back-ends, and change the runtime strategy with which aspecific back-end is selected for provisioning. The resource provisioning infrastructure built in Aneka is mainly concentrated in the Resource Provisioning Service which includes all the operations that are needed for provisioning virtual instances. The implementation of the service is based on resource pools. A resource pool abstracts the interaction with a specific laaS provider by exposing a common interface, so that all the pools can be managed uniformly.

c. Resource provisioning is a feature designed to support Quality of Service (QoS) requirements driven execution of applications. Therefore, it mostly serves requests coming from the **Reservation Service** or the Scheduling services. Despite this, external applications candirectly leverage the resource pro- visioning capabilities of Aneka by dynamically retrievinga client to the service and interacting with the infrastructure. This extends the resource provisioning scenarios that can be handled by Aneka, which can also be used as a virtual machine manager.

3. Write a note of Platform abstraction layer.

The Platform Abstraction Layer (PAL) addresses this heterogeneity and provides the container with a uniform interface for accessing the relevant hardware and operating system information, thus allowing the rest of the container to run unmodified on any platform supported.

The PAL is responsible for detecting the supported hosting environment, and providing the corresponding to interact with it for supporting the activity of the container.

It provides the following features:

- Uniform and platform-independent implementation interface for accessing the hostingplatform.
- Uniform access to extended and additional properties of the hosting platform
- Uniform and platform independent access to remote nodes .
- Uniform and platform independent management interfaces

The PAL is a small layer of software comprising a detection engine which automatically configures the container at boot time with the platform-specific component to access the above information, and an implementation of the abstraction layer for the Windows, Linux, and Mac OS X operating system. The collectible data that are exposed by the PAL are the following:

- Number of cores, frequency, and CPU usage
- Memory size and usage
- Aggregate available disk space.
- Network addresses and devices attached to the node

Moreover, additional custom information can be retrieved by querying the properties of the hardware. The PAL interface provides means for custom implementations to pull additional information by using

name-value pairs that can host any kind of information about the hosting platform.

4. Explain the infrastructure organization of Aneka clouds.

The scenario is a reference model for all the different deployments Aneka supports. A central role is played by the **Administrative Console**, which performs all the required management operations. A fundamental element for Aneka Cloud deployment is constituted by *repositories*. A repository provides storage for all the libraries required to lay out and install the basic Aneka platform. These libraries constitute the software image for the node manager and the container programs. Repositories can make libraries available through a variety of communication channels, such as HTTP, FTP, common file sharing, and so on. The **Management Console** can manage multiple repositories and select the one that best suits the specific deployment. The infrastructure is deployed by harnessing a collection of nodes and installing on them the Aneka node manager, also called the *Aneka daemon*. The daemon constitutes the remote management service used to deploy and control container instances. The collection of resulting containers identifies the Aneka Cloud. From an infrastructure point of view, the management of physical or virtual nodes is performed uniformly as long asit is possible to have an Internet connection and remote administrative access to the node. A different scenario is constituted by the dynamic provisioning of virtual instances; these are generally created by pre-packaged images already containing an installation of Aneka, which only need to be configured to join a specific Aneka Cloud.

FIGURE 5.3
Aneka cloud infrastructure overview.

5. Explain briefly about logical organization of Aneka clouds.

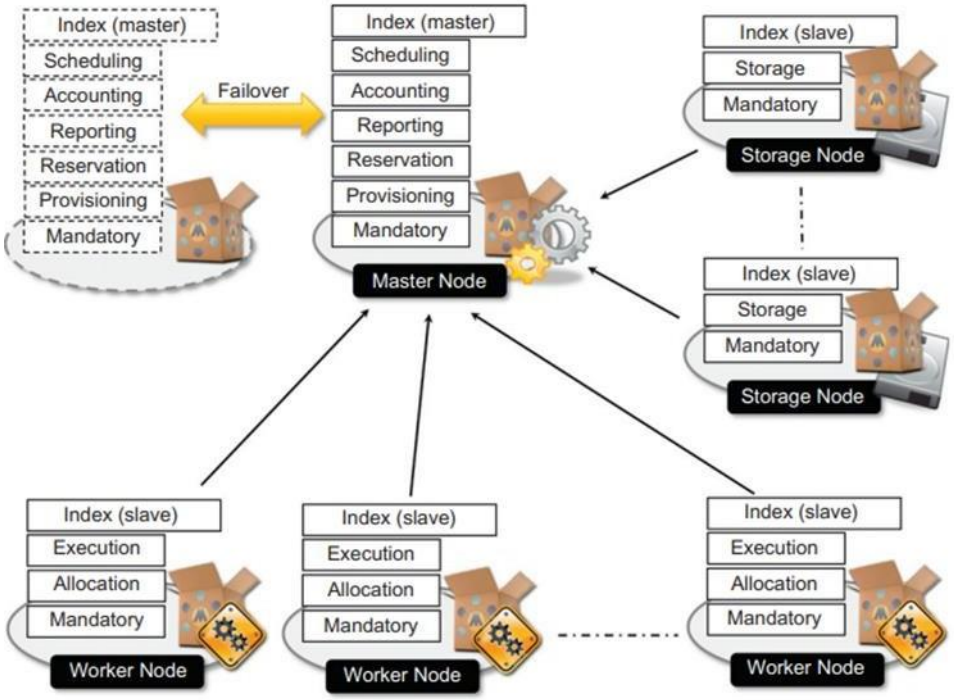
The logical organization of Aneka Clouds can be very diverse, since it strongly depends on the configuration selected for each of the container instances belonging to the Cloud. The most common scenario is to use a master-worker configuration with separate nodes for storage.

The master node features all the services that are most likely to be present in one single copy and that provide the intelligence of the Aneka Cloud. What specifically characterizes a node as a master node is the presence of the Index Service (or Membership Catalogue) configured in master mode; all the other services, except for those that are mandatory, might be presentor located in other nodes. A common configuration of the master node is as follows:

- Index Service (master copy)
- Heartbeat Service
- Logging Service
- Reservation Service
- Resource Provisioning Service
- Accounting Service
- Reporting and Monitoring Service
- Scheduling Services for the supported programming models

The **master node** also provides connection to an RDBMS facility where the state of several services is maintained. For the same reason, all the scheduling services aremaintained in the master node. They

share the application store that is normally persisted on the RDBMS in order to provide a fault-tolerant infrastructure. The master configuration can then be replicated in several nodes to provide a highly available infrastructure based on the failover mechanism.



The **worker nodes** constitute the workforce of the Aneka Cloud and are generally configuredfor the execution of applications. They feature the mandatory services and the specificexecution services of each of the supported programming models in the Cloud, A very common configuration is the following:

- Index Service
- Heartbeat Service
- Logging Service
- Allocation Service
- Monitoring Service
- Execution Services for the supported programming models

Storage nodes are optimized to provide storage support to applications. They feature, among the mandatory and usual services, the presence of the Storage Service. The number of storagenodes strictly depends on the predicted workload and storage consumption of applications. Storage nodes mostly reside on machines that have considerable disk space to accommodatea large quantity of files. The common configuration of a storage node is the following:

- Index Service
- Heartbeat Service
- Logging Service
- Monitoring Service
- Storage Service

6. Differentiate private, public and hybrid cloud deployment mode.

Difference	Public Cloud	Private Cloud	Hybrid Cloud

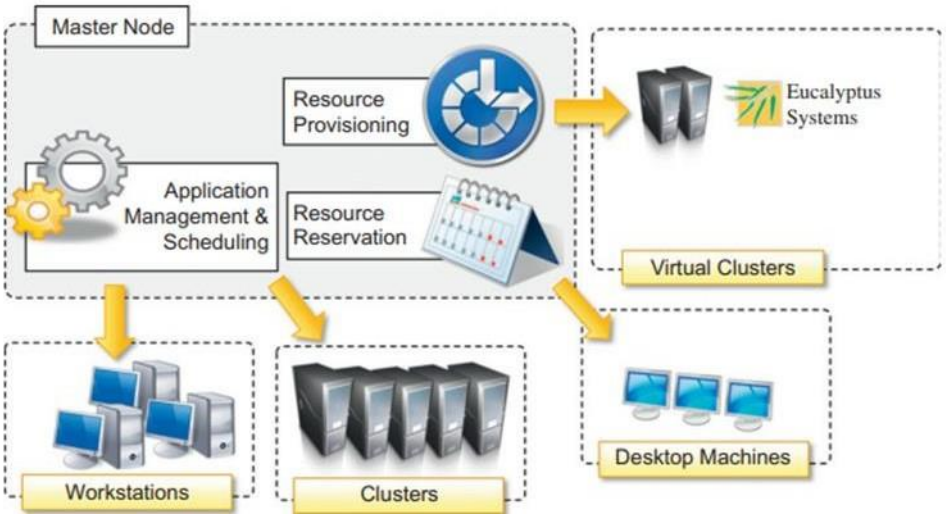
Data Tenancy	**Multi-tenancy:** The data of numerous companies is stored in a shared environment	**Single Tenancy:** The data of only a single organization is stored in the cloud	The data stored in the public cloud is shared, and the data stored in the private cloud is not shared and kept confidential
Cloud Services	Open to public	Only that specific organization can use the cloud services	Services on the public cloud can be accessed by everyone, whereas services in the private cloud can be accessed only by that organization
Connectivity	Over the internet	Over the organization's private network	Over the internet for public cloud services and organization's private network for private cloud services
Management of Cloud Services	Managed by the cloud service provider	Managed by the administrators of that specific organization	The public cloud is managed by the cloud service provider, whereas the administrators of that particular organization manage the private cloud
Software and Hardware Components	The cloud service provider manages these components	That particular organization operates these components	Public cloud components – Cloud Service providerPrivate cloud components – Organization
Costs	Less expensive as the cloud service provider offers all the resources	Very expensive as the organization has to purchase all the resources	Less costly for public cloud and more expensive for private cloud resources
Scalability and Flexibility	High	High	High
Security	Low	High	Public Cloud – LowPrivate Cloud – High

7. **Explain private cloud deployment mode.**

A private deployment mode is mostly constituted by local physical resources and infrastructure management software providing access to a local pool of nodes, which mightbe virtualized. In this scenario Aneka Clouds are created by harnessing a heterogeneous pool of resources such has desktop machines, clusters, or workstations. These resources can be partitioned into different groups, and Aneka can be configured to leverage these resources according to application needs.

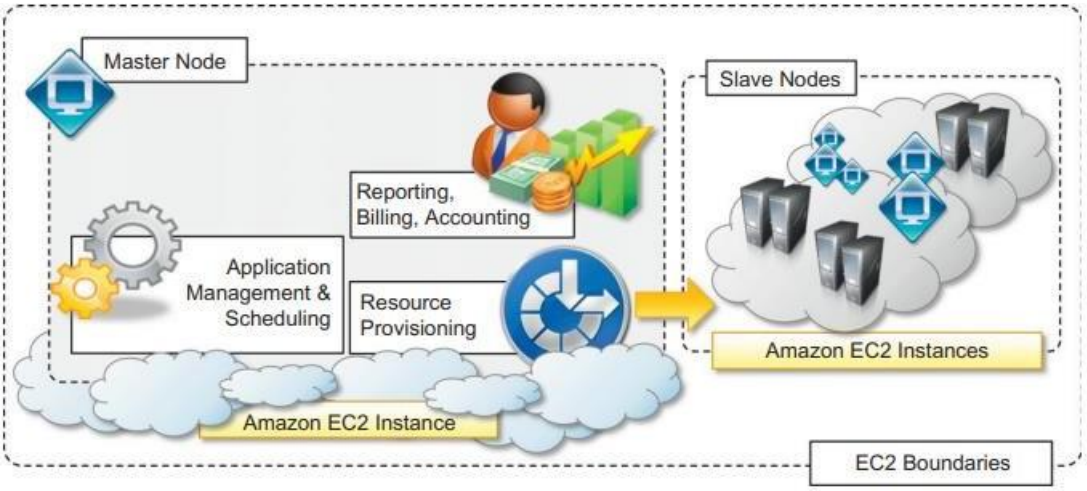
This deployment is acceptable for a scenario in which the workload of the system is predictable and a local virtual machine manager can easily address excess capacity demand. Most of the Aneka nodes are constituted of physical nodes with a long lifetime and a static configuration and generally do not need to be reconfigured often. The different nature of the machines harnessed in a private environment allows for specific policies on resource management and usage that can be accomplished by means of the Reservation Service. For example, desktop machines that are used during the day for office

automation can be exploited outside the standard working hours to execute distributed applications.



8. Explain public cloud deployment mode.

Public Cloud deployment mode features the installation of Aneka master and worker nodes over a completely virtualized infrastructure that is hosted on the infrastructure of one or more resource providers. In this case it is possible to have a static deployment where the nodes are provisioned beforehand and used as though they were real machines.



The deployment is generally contained within the infrastructure boundaries of a single IaaS provider. The reasons for this are to minimize the data transfer between different providers, which is generally priced at a higher cost, and to have better network performance. In this scenario it is possible to deploy an Aneka Cloud composed of only one node and to completely leverage dynamic provisioning to elastically scale the infrastructure on demand. A fundamental role is played by the Resource Provisioning Service, which can be configured with different images and templates to instantiate. Other important services that have to be included in the master node are the Accounting and Reporting Services.

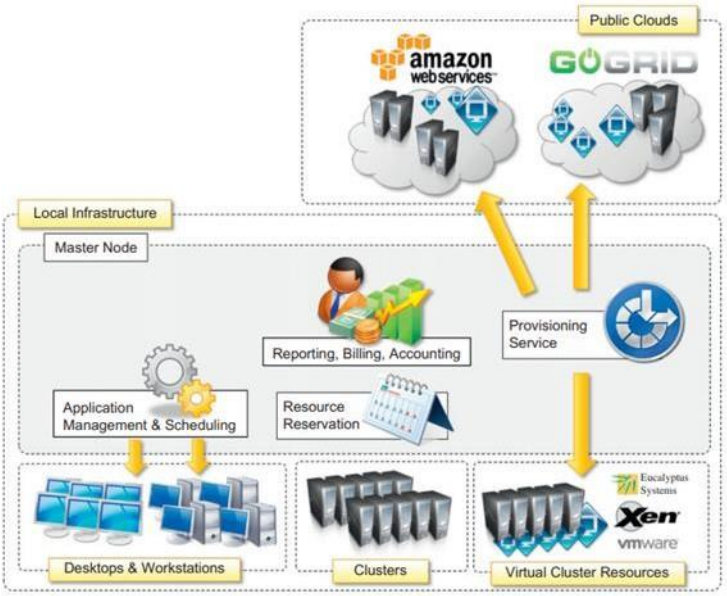
9. Explain hybrid cloud deployment mode.

The hybrid deployment model constitutes the most common deployment of Aneka. In many cases, there is an existing computing infrastructure that can be leveraged to address the computing needs of applications.

This scenario constitutes the most complete deployment for Aneka that is able to leverage all the capabilities of the framework:

- Dynamic Resource Provisioning
- Resource Reservation
- Workload Partitioning

- Accounting, Monitoring, and Reporting



In a hybrid scenario, heterogeneous resources can be used for different purposes. As we discussed in the case of a private cloud deployment, desktop machines can be reserved for low priority workload outside the common working hours. The majority of the applications will be executed on workstations and clusters, which are the nodes that are constantlyconnected to the Aneka Cloud. Any additional computing capability demand can be primarilyaddressed by the local virtualization facilities, and if more computing power is required, it is possible to leverage external IaaS providers.

10. Explain Aneka SDK.

Aneka provides APIs for developing applications on top of existing programming models, implementing new programming models, and developing new services to integrate into the Aneka Cloud. The development of applications mostly focuses on the use of existing featuresand leveraging the services of the middleware, while the implementation of newprogramming models or new services enriches the features of Aneka. The SDK provides support for both programming models and services by means of the Application Model and the Service Model

- **Application Model:** The Application Model represents the minimum set of APIs that is common to all the programming models for representing and programming distributed applications on top of Aneka. This model is further specialized according to the needs and the particular features of each of the programming models.

Each distributed application running on top of Aneka is an instance of the ApplicationBase <M> class, where M identifies the specific type of application manager used to control theapplication. Application classes constitute the developers' view of a distributed applicationon Aneka Clouds, whereas application managers are internal components that interact withAneka Clouds in order to monitor and control the execution of the application. Applicationmanagers are also the first element of specialization of the model and vary according tothe specific programming model used.

Aneka further specializes applications into two main categories:

(1) applications whose tasks are generated by the user
(2) applications whose tasks are generated by the runtime infrastructure

The first category is the most common and it is used as a reference for several programming models supported by Aneka: the Task Model, the Thread Model, and the Parameter Sweep Model. Applications that fall into this category are composed of a collection of units of work submitted by the user and represented by the WorkUnit class. Each unit of work can have input and output files, the transfer of which is transparently managed by the runtime. The specific type of WorkUnit class used to represent the unit of work depends on the programming model used. All the applications that fall into this category inherit or are instances of AnekaApplication <W,M>, where W is the specific

type of WorkUnit class used, and M is the type of application manager used to implement the ManualApplicationManager interface.

The second category covers the case of MapReduce and all those other scenarios in which the units of work are generated by the runtime infrastructure rather than the user. In this case there is no common unit-of-work class used, and the specific classes used by application developers strictly depend on the requirements of the programming model used. For example, in the case of the MapReduce programming model, developers expresstheir distributed applications in terms of two functions, map and reduce; hence, the MapReduce Application class provides an interface for specifying the Mapper <K,V> and Reducer <K,V> types and the input files required by the application. Other programming models might have different requirements and expose different interfaces. For this reason there are no common base types for this category except for Application Base <M>, whereM implements Auto Application Manager. The features that are available in the Aneka Application Model and the way they reflect into the supported programming model:

Table 5.1 Aneka's Application Model Features

Category	Description	Base Application Type	Work Units?	Programming Models
Manual	Units of work are generated by the user and submitted through the application.	*AnekaApplication<W,M>* *IManualApplicationManager<W>* *ManualApplicationManager<W>*	Yes	Task Model Thread Model Parameter Sweep Model
Auto	Units of work are generated by the runtime infrastructure and managed internally.	*ApplicationBase<M>* *IAutoApplicationManager*	No	*MapReduce* Model

- Service Model

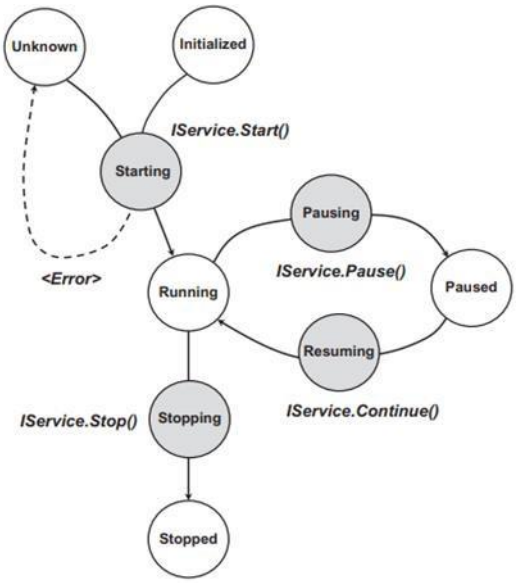
The Aneka Service Model defines the basic requirements to implement a service that can be hosted in an Aneka Cloud. The container defines the runtime environment in which services are hosted. Each service that is hosted in the container must be compliant with theIService interface, which exposes the following methods and properties:

- Name and status
- Control operations such as Start, Stop, Pause, and Continue methods
- Message handling by means of the HandleMessage method

This figure describes the reference life cycle of each service instance in the Aneka container. The shaded balloons indicate transient states; the white balloons indicate steady states. A service instance can initially be in the Unknown or Initialized state, a condition that refers to the creation of the service instance by invoking its constructor during the configuration of the container. Once the container is started, it will iteratively call the Startmethod on each service method. As a result the service instance is expected to be in a Starting state until the startup process is completed, after which it will exhibit the Running state. This is the condition in which the service will last as long as the container is active and running. This is the only state in which the service is able to process messages. If an exception occurs while starting the service, it is expected that the service will fall back to the Unknown state, thus signalling an error.

When a service is running it is possible to pause its activity by calling the Pause method and resume it by calling Continue. As described in the figure, the service moves first into the Pausing state, thus reaching the Paused state. From this state, it moves into theResuming state while restoring its activity to return to the Running state. Not all the services need to support the pause/continue operations, and the current implementation of the framework does not feature any service with these capabilities.

When the container shuts down, the Stop method is iteratively called on each service running, and services move first into the transient Stopping state to reach the final Stoppedstate, where all resources that were initially allocated have been released.



This figure describes the reference life cycle of each service instance in the Aneka container. The shaded balloons indicate transient states; the white balloons indicate steady states. A service instance can initially be in the Unknown or Initialized state, a condition that refers to the creation of the service instance by invoking its constructor during the configuration of the container. Once the container is started, it will iteratively call the Start method on each service method. As a result the service instance is expected to be in a Starting state until the startup process is completed, after which it will exhibit the Runningstate. This is the condition in which the service will last as long as the container is active and running. This is the only state in which the service is able to process messages. If an exception occurs while starting the service, it is expected that the service will fall back to the Unknown state, thus signalling an error.

When a service is running it is possible to pause its activity by calling the Pause method and resume it by calling Continue. As described in the figure, the service moves first into the Pausing state, thus reaching the Paused state. From this state, it moves into theResuming state while restoring its activity to return to the Running state. Not all the services need to support the pause/continue operations, and the current implementation of the framework does not feature any service with these capabilities.

When the container shuts down, the Stop method is iteratively called on each service running, and services move first into the transient Stopping state to reach the final Stoppedstate, where all resources that were initially allocated have been released.

11. Write a node on Application model of Aneka SDK.

The Application Model represents the minimum set of APIs that is common to all the programming models for representing and programming distributed applications on top of Aneka. This model is further specialized according to the needs and the particular features of each of the programming models.

Each distributed application running on top of Aneka is an instance of the ApplicationBase <M> class, where M identifies the specific type of application manager used to control theapplication. Application classes constitute the developers' view of a distributed applicationon Aneka Clouds, whereas application managers are internal components that interact withAneka Clouds in order to monitor and control the execution of the application. Applicationmanagers are also the first element of specialization of the model and vary according tothe specific programming model used.

Aneka further specializes applications into two main categories:

(3) applications whose tasks are generated by the user
(4) applications whose tasks are generated by the runtime infrastructure

The first category is the most common and it is used as a reference for several programming models supported by Aneka: the Task Model, the Thread Model, and the Parameter Sweep Model.

Applications that fall into this category are composed of a collection of units of work submitted by the user and represented by the WorkUnit class. Each unit of work can have input and output files, the transfer of which is transparently managed by the runtime. The specific type of WorkUnit class used to represent the unit of work depends on the programming model used. All the applications that fall into this category inherit or are instances of AnekaApplication <W,M>, where W is the specific type of WorkUnit class used, and M is the type of application manager used to implement the ManualApplicationManager interface.

The second category covers the case of MapReduce and all those other scenarios in which the units of work are generated by the runtime infrastructure rather than the user. In this case there is no common unit-of-work class used, and the specific classes used by application developers strictly depend on the requirements of the programming model used. For example, in the case of the MapReduce programming model, developers expresstheir distributed applications in terms of two functions, map and reduce; hence, the MapReduce Application class provides an interface for specifying the Mapper <K,V> and Reducer <K,V> types and the input files required by the application. Other programming models might have different requirements and expose different interfaces. For this reason there are no common base types for this category except for Application Base <M>, whereM implements Auto Application Manager. The features that are available in the Aneka Application Model and the way they reflect into the supported programming model:

Table 5.1 Aneka's Application Model Features

Category	Description	Base Application Type	Work Units?	Programming Models
Manual	Units of work are generated by the user and submitted through the application.	*AnekaApplication<W,M>* *IManualApplicationManager<W>* *ManualApplicationManager<W>*	Yes	Task Model Thread Model Parameter Sweep Model
Auto	Units of work are generated by the runtime infrastructure and managed internally.	*ApplicationBase<M>* *IAutoApplicationManager*	No	*MapReduce* Model

12. Explain service life cycle of Aneka service model.

The Aneka Service Model defines the basic requirements to implement a service that can be hosted in an Aneka Cloud. The container defines the runtime environment in which services are hosted. Each service that is hosted in the container must be compliant with theIService interface, which exposes the following methods and properties:

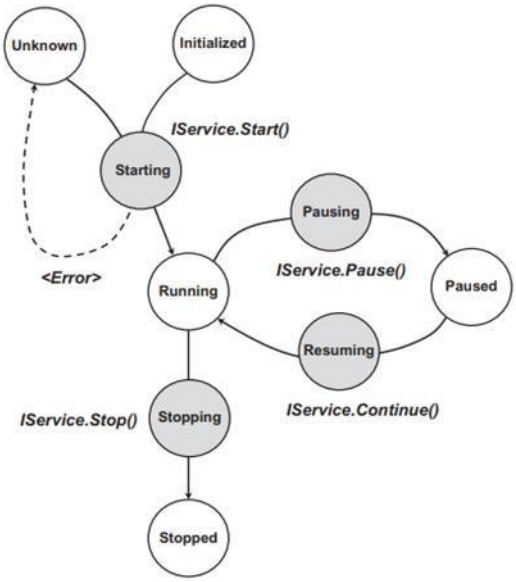
- Name and status
- Control operations such as Start, Stop, Pause, and Continue methods
- Message handling by means of the HandleMessage method

This figure describes the reference life cycle of each service instance in the Aneka container. The shaded balloons indicate transient states; the white balloons indicate steady states. A service instance can initially be in the Unknown or Initialized state, a condition that refers to the creation of the service instance by invoking its constructor during the configuration of the container. Once the container is started, it will iteratively call the Startmethod on each service method. As a result the service instance is expected to be in a Starting state until the startup process is completed, after which it will exhibit the Running state. This is the condition in which the service will last as long as the container is active and running. This is the only state in which the service is able to process messages. If an exception occurs while starting the service, it is expected that the service will fall back to the Unknown state, thus signalling an error.

When a service is running it is possible to pause its activity by calling the Pause method and resume it by calling Continue. As described in the figure, the service moves first into the Pausing state, thus reaching the Paused state. From this state, it moves into theResuming state while restoring its activity to return to the Running state. Not all the services need to support the pause/continue operations, and the current implementation of the framework does not feature any service with these capabilities.

When the container shuts down, the Stop method is iteratively called on each service running, and services move first into the transient Stopping state to reach the final Stoppedstate, where all

resources that were initially allocated have been released.



This figure describes the reference life cycle of each service instance in the Aneka container. The shaded balloons indicate transient states; the white balloons indicate steady states. A service instance can initially be in the Unknown or Initialized state, a condition that refers to the creation of the service instance by invoking its constructor during the configuration of the container. Once the container is started, it will iteratively call the Start method on each service method. As a result the service instance is expected to be in a Starting state until the startup process is completed, after which it will exhibit the Runningstate. This is the condition in which the service will last as long as the container is active and running. This is the only state in which the service is able to process messages. If an exception occurs while starting the service, it is expected that the service will fall back to the Unknown state, thus signalling an error.

When a service is running it is possible to pause its activity by calling the Pause method and resume it by calling Continue. As described in the figure, the service moves first into the Pausing state, thus reaching the Paused state. From this state, it moves into theResuming state while restoring its activity to return to the Running state. Not all the services need to support the pause/continue operations, and the current implementation of the framework does not feature any service with these capabilities.

When the container shuts down, the Stop method is iteratively called on each service running, and services move first into the transient Stopping state to reach the final Stoppedstate, where all resources that were initially allocated have been released.

13. Explain the different categories of management tools.

Aneka is a pure PaaS implementation and requires virtual or physical hardware to be deployed. Hence, infrastructure management, together with facilities for installing logical clouds on such infrastructure, is a fundamental feature of Aneka's management layer. This layer also includes capabilities for managing services and applications running in the Aneka Cloud

- **Infrastructure management:** Aneka leverages virtual and physical hardware in order to deploy Aneka Clouds. Virtual hardware is generally managed by means of the Resource Provisioning Service, which acquires resources on demand according to the need of applications, while physical hardware is directly managed by the Administrative Console by leveraging the Aneka management API of the PAL. The management features are mostly concerned with the provisioning of physical hardware and the remote installation of Aneka on the hardware.
- **Platform management:** Infrastructure management provides the basic layer on top ofwhich Aneka Clouds are deployed. The creation of Clouds is orchestrated by deploying a collection of services on the physical infrastructure that allows the installation and the management of containers. A collection of connected containers defines the platform on top of which

applications are executed. The features available for platform management are mostly concerned with the logical organization and structure of Aneka Clouds. It is possible to partition the available hardware into several Clouds variably configured for different purposes. Services implement the core features of Aneka Clouds and the management layer exposes operations for someof them, such as Cloud monitoring, resource provisioning and reservation, user management, and application profiling.

- **Application management:** Applications identify the user contribution to the Cloud. The management APIs provide administrators with monitoring and profiling features that help them track the usage of resources and relate them to users and applications. This is an important feature in a cloud computing scenario in which users are billedfor their resource usage. Aneka exposes capabilities for giving summary and detailed information about application execution and resource utilization.