Explain Distributed, Grid and Cloudcomputing

Distributed, Grid, and Cloud computing are all related to the concept of distributed computing, which is the use of a network of computers to perform a task.

Distributed computing refers to the use of multiple computers in a network to work together to solve a single problem. Each computer in the network is responsible for a portion of the task, and the results are combined to produce the final output. This type of computing is often used in scientific and engineering applications, such as weather modeling, simulations, and data analysis.

Grid computing builds on the concept of distributed computing by creating a virtual supercomputer by connecting multiple computers over a network. In grid computing, the computers can be geographically dispersed and owned by different organizations, and the resources are coordinated to work on a common goal. Grid computing is often used in scientific and research communities to handle large-scale computations that require significant processing power.

Cloud computing is a model for providing on-demand access to a shared pool of computing resources, such as servers, storage, and applications, over the internet. Cloud computing enables users to access resources as needed, without having to invest in and maintain their own infrastructure. The resources are typically provided by third-party service providers, and users pay only for what they use. Cloud computing is often used in businesses and organizations to reduce costs, increase flexibility, and improve scalability.

In summary, distributed computing refers to the use of multiple computers in a network to work together to solve a single problem, grid computing is a virtual supercomputer that connects multiple computers over a network, and cloud computing is a model for providing on-demand access to a shared pool of computing resources over the internet.

Demonstrate the working of Virtualization,container, kubernetes and cloud.

Virtualization, containers, Kubernetes, and cloud computing are all related to the concept of managing and deploying applications and services in a distributed computing environment.

Virtualization: Virtualization is a technology that enables the creation of multiple virtual machines (VMs) on a single physical server. Each virtual machine runs its own operating system and applications, which are isolated from other virtual machines on the same server. This allows multiple applications to be run on a single physical server, which can improve resource utilization and reduce hardware costs.

Working: In virtualization, a hypervisor is installed on the physical server, which allows multiple virtual machines to run on top of it. The hypervisor provides each virtual machine with its own virtual hardware, including CPU, memory, storage, and network interfaces, which are isolated from other virtual machines running on the same server.

Containers: Containers are a lightweight alternative to virtual machines that provide an isolated environment for running applications. Unlike virtual machines, which require a full operating system to be installed, containers share the host operating system, but have their own isolated file system, network stack, and process space. This makes containers more lightweight and faster to start up than virtual machines.

Working: In containers, a container engine is used to create and manage containers. The container engine interacts with the host operating system's kernel to create an isolated environment for the container. Each container has its own file system, network stack, and process space, which are isolated from other containers on the same host.

Kubernetes: Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Kubernetes enables the deployment of containers across a cluster of servers, and provides features for scaling, rolling updates, and self-healing of applications.

Working: In Kubernetes, a cluster of servers is used to run containerized applications. The Kubernetes control plane manages the cluster and provides features for scaling, scheduling, and managing containers across the cluster. Users interact with the control plane through a command-line interface or a graphical user interface to deploy and manage applications.

Cloud Computing: Cloud computing is a model for providing on-demand access to a shared pool of computing resources, such as servers, storage, and applications, over the internet. Cloud computing enables users to access resources as needed, without having to invest in and maintain their own infrastructure.

Working: In cloud computing, users access computing resources provided by a cloud service provider over the internet. The resources are typically provided as a service, such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), or Software-as-a-Service (SaaS). The cloud service provider manages the underlying infrastructure, including servers, storage, and network resources, while users manage the applications and services running on top of the infrastructure.

In summary, virtualization, containers, Kubernetes, and cloud computing are all important technologies for managing and deploying applications and services in a distributed computing environment. Virtualization allows the creation of multiple virtual machines on a single physical server, while containers provide a lightweight alternative to virtual machines. Kubernetes automates the deployment, scaling, and management of containerized applications across a cluster of servers, and cloud computing provides on-demand access to a shared pool of computing resources over the internet.

Analyze the deployment of services such as web service and database service.

Deploying web services and database services can be complex, as both require specific configurations and infrastructure to run reliably and securely. Here's a general analysis of the deployment of these services:

Web Services Deployment:

Web services are software systems designed to support interoperable machine-to-machine interaction over a network. They allow applications to communicate with each other using standardized protocols, such as HTTP or SOAP. Here are some key considerations for deploying web services:

Application server: A web server or application server is needed to host and run the web service. Popular web servers include Apache, Nginx, and IIS, while popular application servers include Tomcat, JBoss, and WebSphere.

Infrastructure requirements: The web service needs to be deployed on hardware and network infrastructure that can handle the expected traffic and performance requirements. This may include load balancing, caching, and other optimizations.

Security considerations: The web service needs to be secured to prevent unauthorized access, data breaches, and other security threats. This may include SSL/TLS encryption, firewalls, access controls, and other security measures.

Monitoring and management: The web service needs to be monitored and managed to ensure uptime, performance, and scalability. This may include logging, metrics, alerts, and other tools for monitoring and troubleshooting.

Database Services Deployment:

Database services are software systems designed to manage and store data. They can be deployed on-premises or in the cloud, and can be used for a wide range of applications, from transaction processing to data analytics. Here are some key considerations for deploying database services:

Database platform: A database management system (DBMS) is needed to manage and store data. Popular DBMSs include MySQL, PostgreSQL, Oracle, and Microsoft SQL Server.

Infrastructure requirements: The database service needs to be deployed on hardware and network infrastructure that can handle the expected workload and performance requirements. This may include storage, networking, and other infrastructure optimizations.

Backup and recovery: Data is critical, so a backup and recovery strategy is necessary to protect against data loss. This may include regular backups, point-in-time recovery, and other data protection measures.

Security considerations: The database service needs to be secured to prevent unauthorized access, data breaches, and other security threats. This may include access controls, encryption, and other security measures.

Monitoring and management: The database service needs to be monitored and managed to ensure uptime, performance, and scalability. This may include logging, metrics, alerts, and other tools for monitoring and troubleshooting.

In summary, deploying web services and database services requires careful consideration of the infrastructure, security, performance, and management requirements. While there are many tools and technologies available to simplify these tasks, it's important to have a good understanding of the underlying requirements to ensure that the services are deployed reliably and securely.

Economics of cloud computing

Cloud computing has a number of economic benefits, both for individuals and businesses. Here are some key economic advantages of cloud computing:

Cost savings: One of the biggest advantages of cloud computing is cost savings. By using cloud services, businesses can avoid the high up-front costs of purchasing and maintaining their own hardware and software. Instead, they can pay for cloud services on a subscription basis, which can be more cost-effective in the long run. Additionally, cloud services can be scaled up or down as needed, which means that businesses only pay for the resources they actually use.

Increased efficiency: Cloud computing can increase efficiency in a number of ways. For example, it can eliminate the need for businesses to maintain their own IT infrastructure, which can free up resources to focus on other areas of the business. Additionally, cloud services can be accessed from anywhere with an internet connection, which can enable remote work and increase productivity.

Improved security: Cloud computing can also improve security for businesses. Cloud providers typically invest heavily in security measures, such as encryption, firewalls, and other technologies, to protect their customers' data. Additionally, because data is stored in the cloud, it can be easier to recover in the event of a disaster, such as a natural disaster or cyber attack.

Increased innovation: Cloud computing can also enable businesses to innovate more quickly. By using cloud services, businesses can leverage the latest technologies and tools, without needing to invest in their own IT infrastructure. This can enable businesses to experiment with new ideas and products more easily, and to bring them to market more quickly.

However, it's worth noting that there are also some potential economic disadvantages of cloud computing. For example, if a business becomes heavily reliant on a particular cloud provider, it may be difficult to switch to a different provider if prices increase or if the quality of service declines. Additionally, businesses may need to invest in training for their employees to ensure that they are able to use cloud services effectively.

In summary, cloud computing can offer significant economic advantages, including cost savings, increased efficiency, improved security, and increased innovation. However, businesses should carefully consider the potential risks and disadvantages of cloud computing before making the switch.

advantage and disadvantages of rds with ec2 and without

Amazon Relational Database Service (RDS) is a cloud-based database service that enables easy deployment and management of relational databases. RDS can be used with Amazon Elastic Compute Cloud (EC2) instances to provide a complete solution for deploying and managing web applications. Here are some advantages and disadvantages of using RDS with EC2 and without:

Advantages of using RDS with EC2:

Easy deployment: When RDS is used with EC2, deploying a fully functional web application becomes easy as developers can focus on writing code instead of configuring the database. RDS simplifies database management and maintenance, including backups, upgrades, and patches, leaving the application developer free to concentrate on the application.

High availability: RDS can provide multi-AZ (Availability Zone) deployments, ensuring that data remains available in the event of an AZ failure. This ensures high availability of data and increased reliability of web applications.

Scalability: RDS can provide automatic scaling of storage and computing resources. When an application's data storage needs increase, the RDS database can be automatically scaled up or down to meet those needs, without the need for manual intervention.

Disadvantages of using RDS with EC2:

Cost: The use of RDS with EC2 can lead to higher costs compared to running a database on an EC2 instance alone. RDS can be more expensive than running a database on EC2, depending on the size of the database, the number of connections, and the level of traffic.

Limited access to the operating system: RDS is a managed service, which means that it is not possible to access the operating system directly. This can limit the ability of developers to fine-tune the database, especially in terms of performance and optimization.

Advantages of using EC2 without RDS:

Full control of the database: Running a database on EC2 gives developers full control of the database, including the operating system, database server, and data. This allows developers to fine-tune the database performance to suit the needs of the application.

Cost savings: Running a database on EC2 can be more cost-effective than using RDS, especially for small-scale applications. EC2 instances can be purchased on a pay-as-you-go basis, which can save costs compared to using RDS.

Disadvantages of using EC2 without RDS:

Complex deployment and management: Running a database on EC2 can be more complex compared to using RDS. This requires more effort in terms of configuration, management, and maintenance. The developer needs to handle database backups, replication, and other administrative tasks.

Limited availability: Running a database on EC2 alone may not provide the high availability and reliability required for web applications, especially in the case of a system failure.

In summary, the use of RDS with EC2 offers benefits such as easy deployment, high availability, and scalability, but can be more expensive than running a database on EC2 alone. Running a database on EC2 alone provides full control and cost savings, but requires more effort in terms of configuration, management, and maintenance.

containerization with docker explaination

Containerization is a method of packaging software applications with all the necessary dependencies and configurations, so they can run reliably and consistently across different computing environments. Docker is a popular platform for containerization that enables developers to package their applications into containers.

Here's how Docker containerization works:

Docker image creation: A Docker image is a read-only file that contains all the necessary code, libraries, and dependencies for a particular application. Developers can create an image by writing a Dockerfile, which is a script that specifies the configuration and dependencies of the application.

Docker image distribution: Once an image is created, it can be distributed to other computing environments. This can be done through a Docker registry, which is a repository of Docker images that can be accessed by other users.

Docker container creation: A Docker container is a lightweight, standalone executable package that includes everything needed to run an application, including the application code, runtime, system tools, libraries, and settings. Docker containers are created from Docker images and are designed to run on any operating system or platform that supports Docker.

Docker container deployment: Docker containers can be deployed to any computing environment that supports Docker, including development, testing, and production environments. This makes it easy to move applications between different environments, without the need for reconfiguration or redeployment.

Some of the benefits of using Docker for containerization include:

Portability: Docker containers can run on any computing environment that supports Docker, which makes it easy to move applications between development, testing, and production environments.

Consistency: Docker containers provide a consistent environment for running applications, regardless of the underlying computing infrastructure.

Resource efficiency: Docker containers are lightweight and consume fewer resources compared to traditional virtual machines, which can help optimize resource usage.

Scalability: Docker containers can be scaled up or down as needed, which makes it easy to adjust resource usage based on application demands.

In summary, containerization with Docker enables developers to package their applications into lightweight, standalone executable packages that are portable, consistent, and resource-efficient. This can help streamline the application development and deployment process, and make it easier to manage applications across different computing environments.

Containers Architecture.Containers vs Virtual machines.Basics of Using Containers in Production

Containers are a form of virtualization technology that allows applications to be run in a lightweight, isolated environment. In contrast to traditional virtual machines, which virtualize entire operating systems, containers virtualize only the application and its dependencies. Here's an overview of container architecture and how it differs from virtual machines:

Containers Architecture: Containers are built on top of a host operating system, which provides the system resources that the container needs to run. Each container has its own file system, network stack, and process namespace, which isolates the container from the host system and other containers.

Containers are created from container images, which are read-only templates that include all the necessary files and dependencies for running an application. Container images can be built from scratch, or they can be based on existing images that have been created by other developers.

Containers vs Virtual Machines: Virtual machines (VMs) are another form of virtualization technology that allow multiple operating systems to run on a single physical machine. Unlike containers, virtual machines virtualize the entire operating system, which can lead to increased resource consumption and overhead.

Because containers virtualize only the application and its dependencies, they are more lightweight and consume fewer resources compared to virtual machines. This makes containers more scalable and efficient, especially for running microservices-based applications.

Basics of Using Containers in Production: Using containers in production requires a few key considerations, including:

Container orchestration: In order to run containers in a production environment, you need a way to manage and orchestrate them. Tools like Kubernetes and Docker Swarm can help automate the deployment, scaling, and management of containerized applications.

Security: Container security is a critical concern in a production environment. Containers should be isolated from each other and from the host system, and access should be restricted to only those resources that are necessary for the application to function.

Monitoring: Monitoring and logging are essential for ensuring that containerized applications are running properly in a production environment. Tools like Prometheus and Fluentd can be used to collect and analyze performance metrics and logs from containerized applications.

Continuous integration and deployment: Containers can be integrated into a continuous integration and deployment (CI/CD) pipeline, which can help automate the testing, building, and deployment of containerized applications.

In summary, containers provide a lightweight and efficient way to virtualize applications, making them more portable and scalable. Using containers in production requires careful consideration of container orchestration, security, monitoring, and continuous integration and deployment.

Container Orchestration, KubernetesIntroduction to Microservices.Introduction to Kubernetes.Kubernetes Architecture.Kubernetes storage.Kubernetes Scalability

Container Orchestration and Kubernetes: Container orchestration is the process of managing and automating the deployment, scaling, and management of containerized applications. Kubernetes is a popular open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications.

Introduction to Microservices: Microservices are a software architecture approach that involves breaking down large, monolithic applications into smaller, independent services that communicate with each other through APIs. Each service is responsible for a specific task, and can be developed, deployed, and scaled independently from the rest of the application.

Introduction to Kubernetes: Kubernetes is an open-source container orchestration platform that provides a platform for deploying, scaling, and managing containerized applications. It automates the deployment and scaling of containerized applications across a cluster of machines, and provides tools for managing the lifecycle of containerized applications.

Kubernetes Architecture: The architecture of Kubernetes is based on a master-slave model, with a master node that controls the operation of the cluster and worker nodes that run the containerized applications. The master node includes several components, such as the API server, controller manager, scheduler, and etcd, which is a distributed key-value store for storing configuration data.

The worker nodes include a kubelet, which is responsible for managing the containers on the node, and a container runtime, such as Docker or CRI-O.

Kubernetes Storage: Kubernetes provides several options for storage, including persistent volumes and persistent volume claims. Persistent volumes are used to store data that needs to persist across container restarts or node failures. Persistent volume claims are used to request a specific amount of storage from the cluster, which can then be bound to a persistent volume.

Kubernetes Scalability: Kubernetes provides several mechanisms for scaling containerized applications, including horizontal pod autoscaling, which automatically scales the number of replicas of a deployment based on resource utilization, and vertical pod autoscaling, which adjusts the resource requests and limits of containers based on resource utilization.

In summary, Kubernetes provides a platform for deploying, scaling, and managing containerized applications, using a master-slave architecture, with several components for managing the operation of the cluster. It provides tools for managing storage, and for scaling containerized applications based on resource utilization. Kubernetes is widely used in production environments for managing containerized applications, and is a key component of modern cloud-native application architecture.

CI/CD Pipelines

CI/CD pipelines are a set of practices and tools used to automate the testing, building, and deployment of software applications. CI stands for Continuous Integration, which is the practice of automating the integration of code changes into a single codebase. CD stands for Continuous Delivery or Continuous Deployment, which is the practice of automating the release of code changes into a production environment.

CI/CD pipelines typically consist of the following stages:

Source Control Management: The first step in the CI/CD pipeline is to store the code in a version control system (VCS) like Git or SVN.

Continuous Integration: When a developer commits code to the VCS, the CI system automatically builds the application and runs automated tests to ensure that the new code changes have not introduced any errors.

Continuous Delivery: After the code has been built and tested, it is deployed to a staging environment where it undergoes further testing and quality assurance checks.

Continuous Deployment: Once the code has been fully tested and approved in the staging environment, it is automatically deployed to the production environment.

CI/CD pipelines provide a number of benefits, including:

Faster time-to-market: By automating the testing, building, and deployment processes, CI/CD pipelines enable organizations to release software faster and more frequently.

Higher quality: Automated testing and quality assurance checks help to identify and fix issues early in the development process, leading to higher quality software.

Improved collaboration: CI/CD pipelines promote collaboration between developers, testers, and operations teams by providing a single, unified process for managing software development and deployment.

Increased agility: CI/CD pipelines enable organizations to quickly respond to changes in the market or user requirements by delivering software faster and more frequently.

In summary, CI/CD pipelines are an essential part of modern software development, providing a set of practices and tools for automating the testing, building, and deployment of software applications. By adopting CI/CD pipelines, organizations can deliver high-quality software faster and more frequently, while promoting collaboration and agility across the development and operations teams.

Cloud Issues and Threats .Manage cloud security

Cloud computing has revolutionized the way businesses operate and store their data, but it also comes with certain issues and threats that need to be managed. Here are some of the main cloud issues and threats and some tips for managing cloud security:

Data Breaches: Cloud data breaches occur when unauthorized individuals gain access to sensitive data stored in the cloud. To manage this threat, organizations should use strong encryption, multi-factor authentication, and access control mechanisms to protect their data.

Insider Threats: Insider threats are risks posed by employees or other trusted individuals who misuse their privileges to access sensitive data. To manage this threat, organizations should implement strict access control policies and monitor user activity for any suspicious behavior.

DDoS Attacks: Distributed Denial of Service (DDoS) attacks are a common threat to cloud systems, in which a large number of requests are sent to a server in order to overload it and disrupt its normal functioning. To manage this threat, organizations should use DDoS protection services and implement firewalls and intrusion detection systems.

Compliance Risks: Cloud services need to comply with various regulatory standards and guidelines, such as HIPAA or GDPR. To manage this risk, organizations should ensure that their cloud service providers are compliant with the necessary regulations and maintain regular audits and assessments of their cloud infrastructure.

Shared Responsibility: Cloud providers and customers share responsibility for cloud security. To manage this risk, organizations should establish clear roles and responsibilities and maintain regular communication and collaboration with their cloud service providers.

Malware and Ransomware: Malware and ransomware can infect cloud systems and disrupt their functioning or hold data hostage. To manage this risk, organizations should use anti-virus software, firewalls, and other security measures to prevent and detect malware and ransomware attacks.

Data Loss: Data loss can occur due to accidental deletion, natural disasters, or other unforeseen events. To manage this risk, organizations should implement regular data backups and disaster recovery plans.

In summary, managing cloud security involves understanding the potential issues and threats that can arise, and taking proactive steps to protect against them. By using best practices and working closely with their cloud service providers, organizations can ensure the security and availability of their data and applications in the cloud.

Economics of cloud computing:OpEx (Operational expenditure)Capex (Capital Expenditures) in IT-industry.Analyzing and recommending cloud adoption.

The economics of cloud computing can be analyzed in terms of two key financial metrics: Operational expenditure (OpEx) and Capital expenditures (CapEx).

OpEx refers to ongoing expenses that are incurred as part of the day-to-day operation of a business. In the context of cloud computing, OpEx includes expenses such as monthly subscription fees, usage-based charges, and ongoing maintenance and support costs.

CapEx, on the other hand, refers to the one-time expenses associated with acquiring and setting up IT infrastructure, such as servers, networking equipment, and data storage. In the context of cloud computing, CapEx is reduced or eliminated because the cloud provider owns and maintains the underlying hardware, allowing customers to pay for only the resources they use.

When considering cloud adoption, organizations should evaluate the potential benefits and risks of moving to the cloud, and determine whether the OpEx savings and other benefits outweigh any potential CapEx investments required for migration.

Some key factors to consider when analyzing and recommending cloud adoption include:

Scalability: Cloud computing enables organizations to quickly and easily scale up or down their computing resources as needed, without the need for significant upfront investments in hardware. This makes it an attractive option for organizations with fluctuating demand for IT resources.

Cost Savings: Cloud computing can offer significant cost savings over traditional on-premise IT infrastructure, by eliminating the need for costly CapEx investments and reducing ongoing OpEx expenses through the use of shared, pay-as-you-go resources.

Security: While cloud providers are responsible for securing their infrastructure, it is important for organizations to understand and manage their own security risks and responsibilities in the cloud.

Compliance: Organizations must ensure that their cloud providers are compliant with relevant industry regulations and standards, and that they themselves maintain compliance with any applicable regulations.

Migration: Cloud migration can be complex and time-consuming, and it is important to plan and execute the migration process carefully in order to avoid disruption to business operations.

In summary, the economics of cloud computing can offer significant benefits to organizations in terms of OpEx savings, scalability, and other advantages. However, it is important for organizations to carefully evaluate the potential costs and risks of cloud adoption, and to develop a clear strategy and plan for migration.

microservices vs monolithic

Microservices and monolithic architectures are two common approaches to developing software applications. Here are some of the key differences between the two:

Architecture: Monolithic architectures consist of a single, large codebase, with all components of the application tightly coupled together. Microservices architectures, on the other hand, consist of multiple small, independent services that communicate with each other through APIs.

Deployment: Monolithic applications are typically deployed as a single unit, while microservices can be deployed and updated independently. This makes it easier to add new features, fix bugs, and scale individual components of the application as needed.

Scalability: Monolithic architectures can be difficult to scale, as the entire application must be replicated for each instance. Microservices architectures, on the other hand, can be scaled horizontally by adding additional instances of specific services as needed.

Maintenance: Monolithic applications can be more difficult to maintain and debug, as changes to one component of the application can have unintended effects on other components. Microservices architectures, by contrast, allow for more focused testing and debugging, as each service can be tested and updated independently.

Development: Developing and deploying new features in a monolithic architecture can be time-consuming, as the entire application must be built and tested. Microservices architectures, by contrast, allow for more rapid development and deployment of new features, as individual services can be developed and deployed independently.

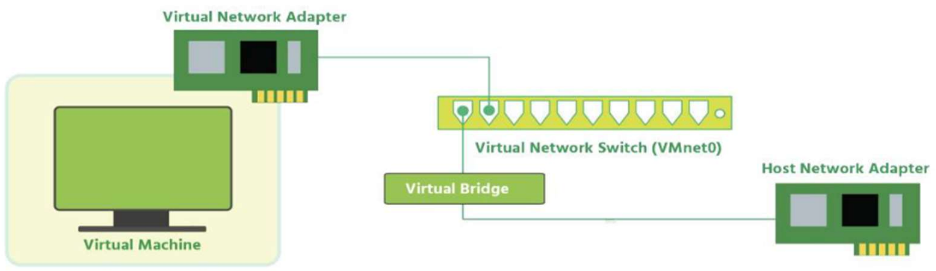
In summary, microservices and monolithic architectures offer different tradeoffs in terms of scalability, deployment, maintenance, and development. While monolithic architectures can be simpler to implement, they may not be as flexible or scalable as microservices architectures, which offer greater modularity, scalability, and flexibility. Ultimately, the choice between the two depends on the specific needs and requirements of the application in question.

Virtual Network

- Virtual networks dramatically increase the scope of physical networks
- Virtual networks can run in isolation along side or on top of identical physical networks
- Each network is unaffected by the events on another network

Bridge Network

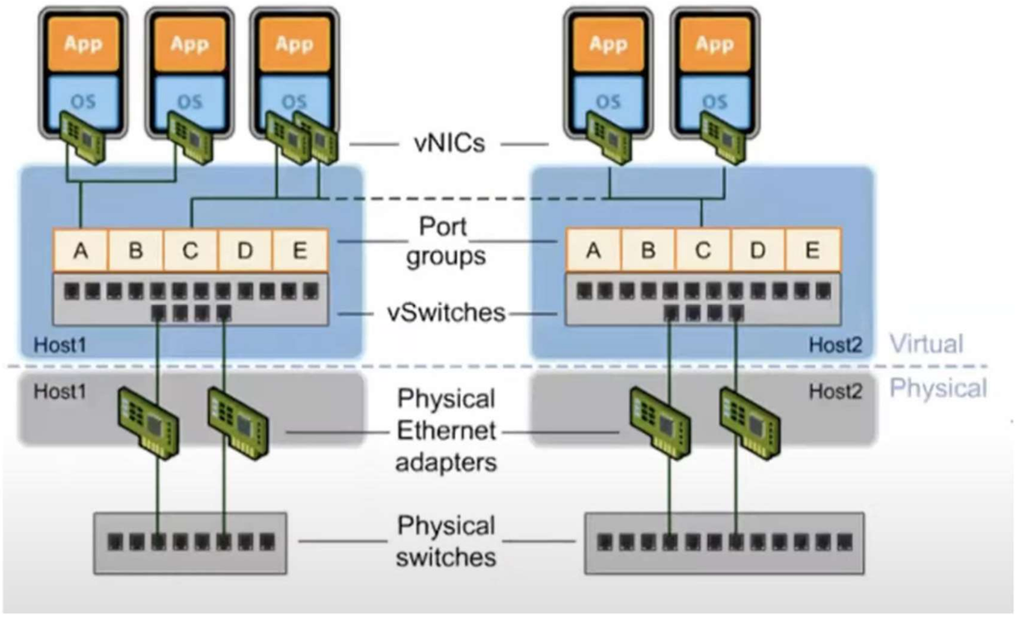
- A network type where both a virtual machine and the host that it is running on are connected to the same network
- With bridged networking, the virtual network adapter (vNIC) for the virtual machine connects to a physical NIC on the physical host system
- The host network adapter enables the VM to connect to the Local Area Network (LAN) that the host system uses

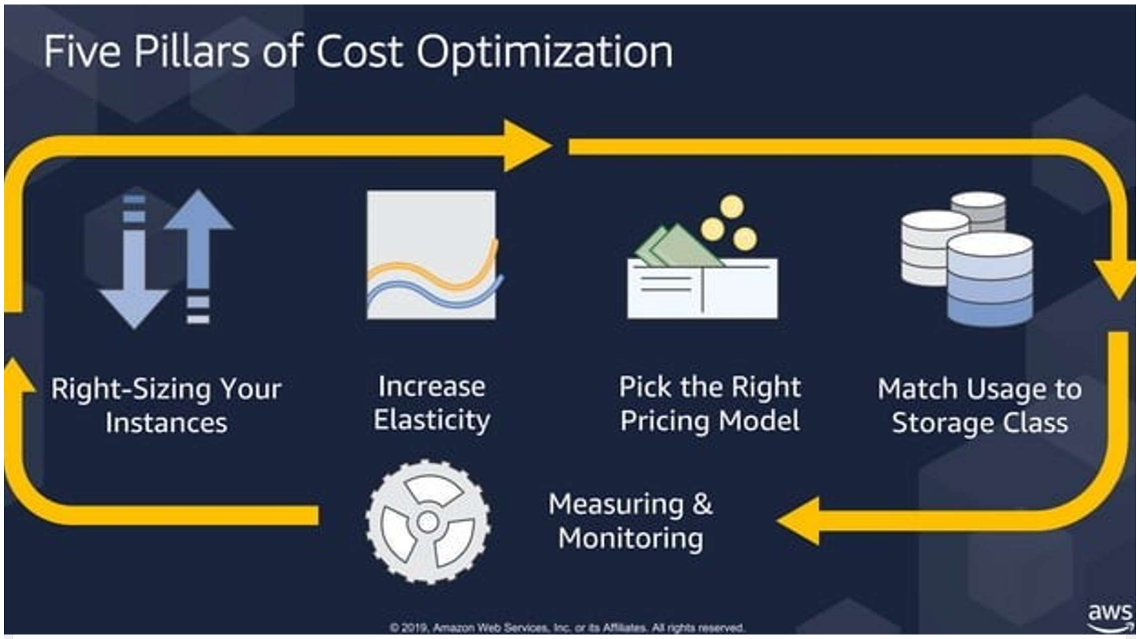


VMkerenel Adapter

- A VMkernel adapter is a port that is used by the hypervisor to attach a service to the network
- • Every VMkernel adapter has an IP address by which this service is accessible

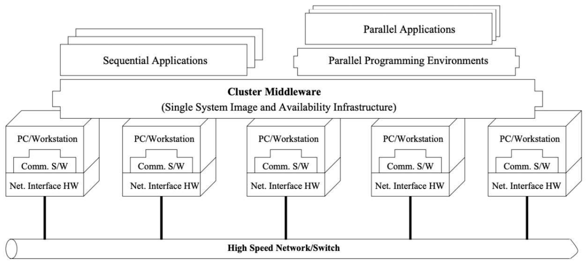
Overall Figure




Five Pillars of Cost Optimization
Right-Sizing Your Instances
Increase Elasticity
Pick the Right Pricing Model
Match Usage to Storage Class
Measuring & Monitoring
© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
aws

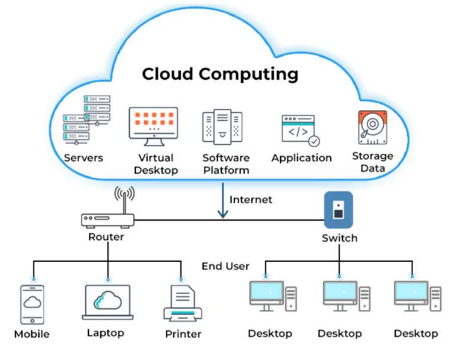
Cluster Computing

A cluster is a type of parallel or distributed processing system which consists of a collection of interconnected standalone computers working together as a **single integrated computing resource.** A computer node can be a single or multiprocessor system PCs workstations with memory IO facilities and an operating system A cluster generally refers to two or more computers nodes connected together in a single cabinet or be physically separated and connected via a LAN An inter connected LAN based cluster of computers can appear as a single system to users and applications Such a system can provide a cost effective way to gain features and benefits fast and reliable services that have historically been found only on more expensive proprietary shared memory systems.


Parallel Applications
Sequential Applications
Parallel Programming Environments
Cluster Middleware
(Single System Image and Availability Infrastructure)
PC/Workstation
Comm. S/W
Net. Interface HW
High Speed Network/Switch

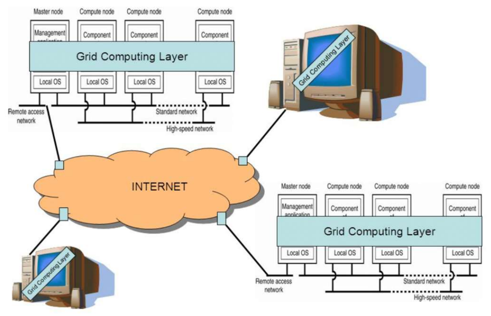
Cloud Computing

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction



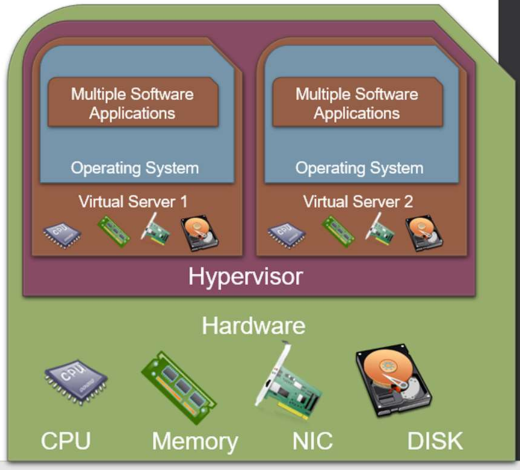
Grid Computing

Grid computing combines computers from multiple administrative domains to reach a common goal, to solve a single task, and may then disappear just as quickly. It is similar to the power grid. One of the main strategies of grid computing is to use middleware to divide and apportion pieces of a program among several computers. Grid computing involves computation in a distributed fashion, which may also involve the aggregation of large-scale cluster computing based systems. The size of a grid may vary from small a network of computer workstations within a corporation to large collaborations across many companies and networks

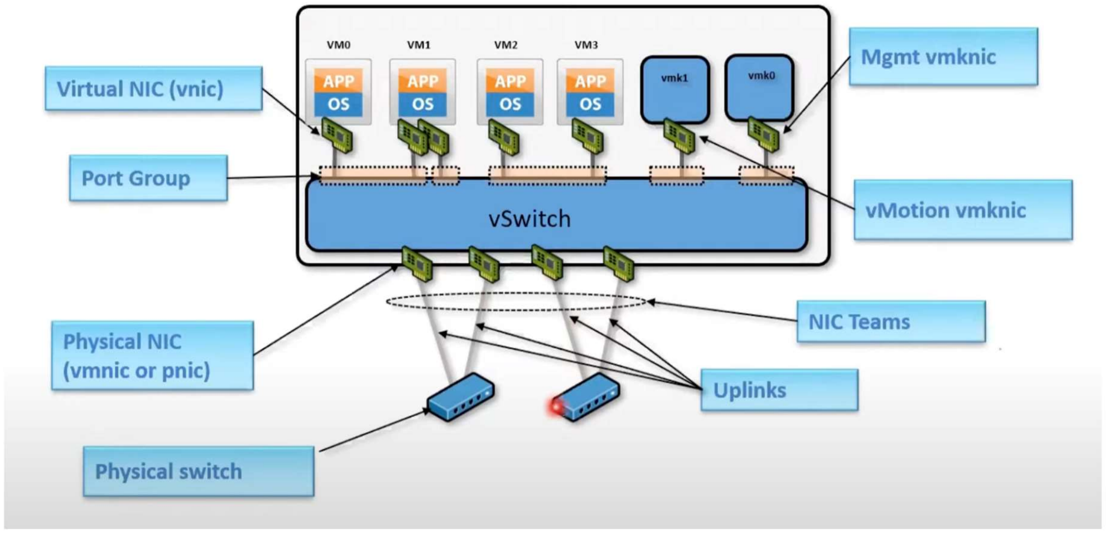


Server with virtualization

- Can run multiple OS simultaneously.
- Each OS can have different hardware configuration.
- Efficient utilization of hardware resources.
- Each virtual machine is independent.
- Save electricity, initial cost to buy servers, space etc.
- Easy to manage and monitor
  virtual machines centrally.

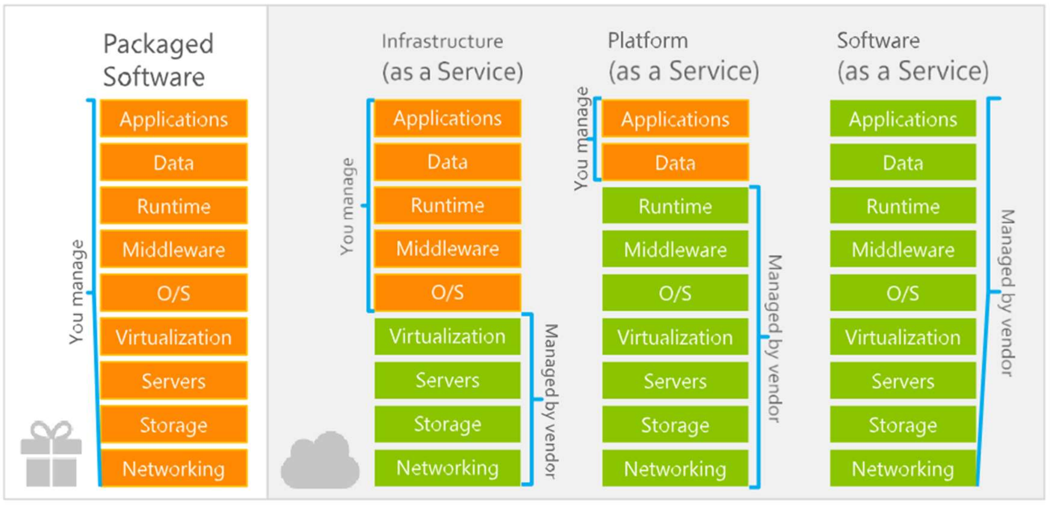


Virtual Network on VM



Cloud Services

1. **IAAS (Infrastructure as a service):** Outsource the elements of infrastructures like virtualization, storage, networking, load balancer.
2. **PAAS (Platform as a Service):** Core hosting operating systems and optional building block service that allows you to run your own applications.
3. **SAAS (Software as a service):** Consumed as a service only for the applications needed.


Packaged Software
Infrastructure (as a Service)
Platform (as a Service)
Software (as a Service)
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You manage
Managed by vendor