What is Distributed system

Definition: A *distributed system* is one in which **components** located at networked computers communicate and **coordinate** their actions only by passing **messages**. This definition leads to the following characteristics of distributed systems:

- Concurrency of components
- Lack of a global 'clock'
- Independent Memory
- Independent failures of components

Examples:

1. Local Area Network and Intranet
2. Database Management System
3. Automatic Teller Machine Network
4. Internet/World-Wide Web
5. Mobile and Ubiquitous Computing

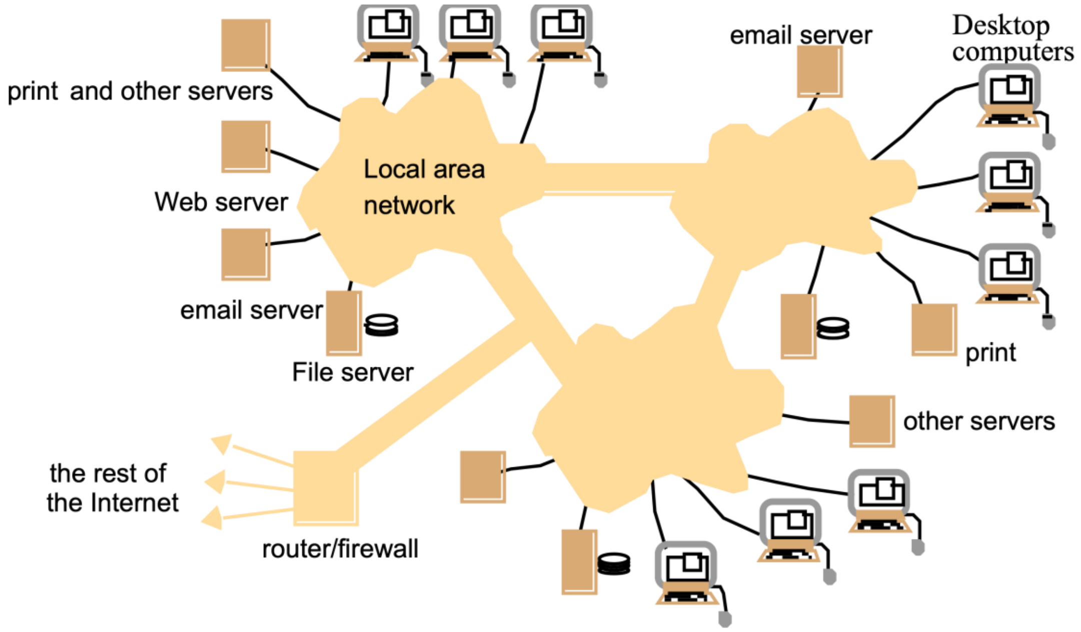
Characteristics of distributed system

- Multiple autonomous components
- Components are not shared by all users
- Resources may not be accessible
- Software runs in concurrent processes on different processors
- Multiple points of control
- Multiple points of failure

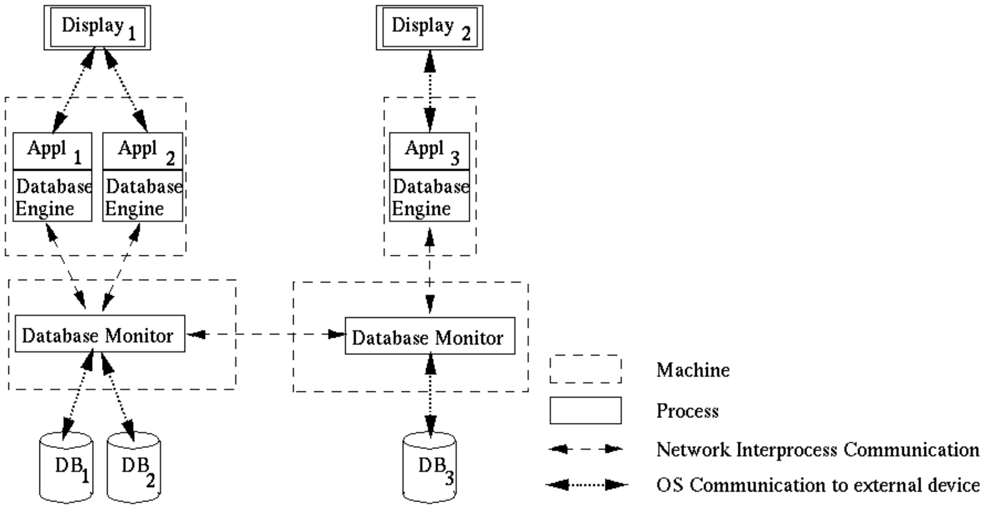
Characteristics of Centralized system

- Component shared by users all the time
- All resources accessible
- Software runs in a single process
- Single point of control
- Single point of failure

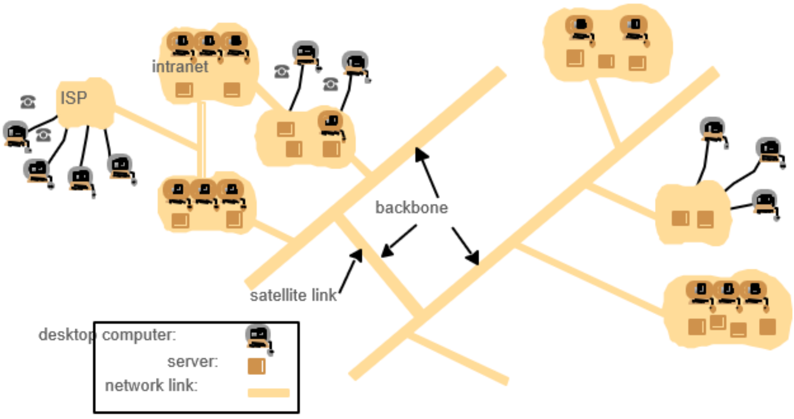
LAN diagram



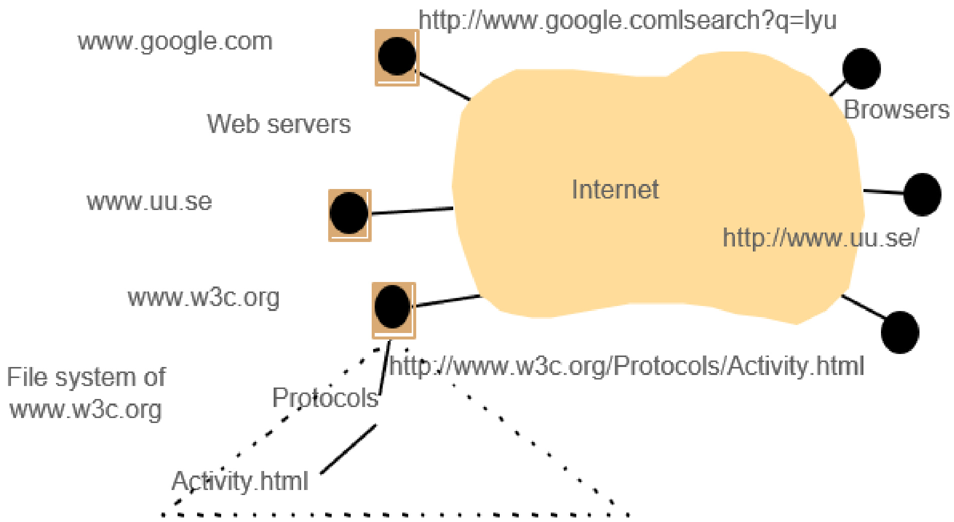
Database management diagram



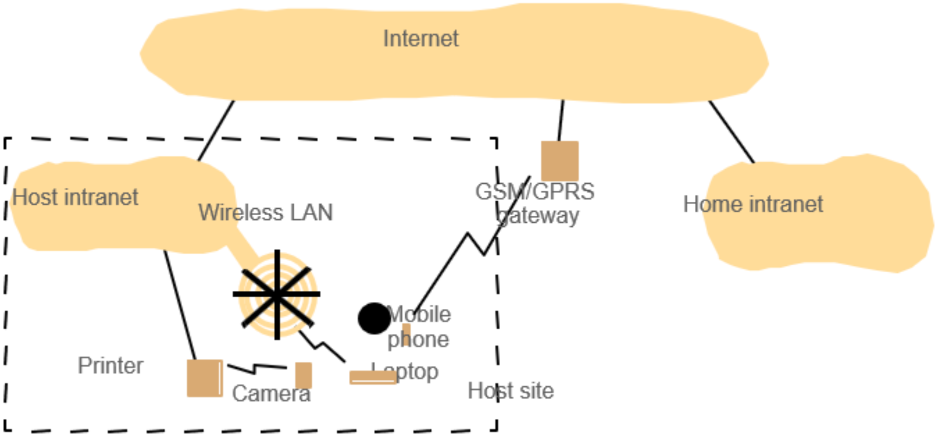
Internet Diagarm



Web Server and web browser


www.google.com
http://www.google.comlsearch?q=lyu
Browsers
Web servers
www.uu.se
Internet
http://www.uu.se/
www.w3c.org
http://www.w3c.org/Protocols/Activity.html
File system of www.w3c.org
Protocols
Activity.html

Mobile and ubiquitous computing



Common Characteristics

- **Heterogeneity**: I can access all the documents that are available on the Internet, even though the documents are located in different type of machines.
- **Openness**: I have credit cards from Hang Seng Bank and Wells Fargo Bank in U.S.A. and can use them at each others tellers. These banks, however, would never develop a common centralized teller system. It is because their systems are open and interoperable that I have this flexibility.
- **Security**: I want to purchase products in e-Commerce. I don't want other people to steal my credit card number.
- **Scalability**: Distributed systems, such as the Internet, grow each day to accommodate more users and to withstand higher load. (Hong Kong stock trading broker is on-line and you can open accounts and do on-line trading from home PC).
- **Failure Handling:** Two (distributed) account databases are managed by the bank to quickly recover from a break-down.
- **Concurrency**: Multiple database users can concurrently access and update data in a distributed database system. The database system preserves integrity against concurrent updates and users perceive the database as their own copy. They are, however, able to see each others changes after they have been completed.
- **Transparency**: When using a distributed system it appears to users as if it were centralized.

Specific issues for distributed systems:

- Naming
- Communication
- Software structure
- System architecture
- Workload allocation
- Consistency maintenance

System Architecture

- Client-Server
- Peer-to-Peer
- Services provided by multiple servers
- Proxy servers and caches
- Mobile code and mobile agents
- Network computers
- Thin clients and mobile devices

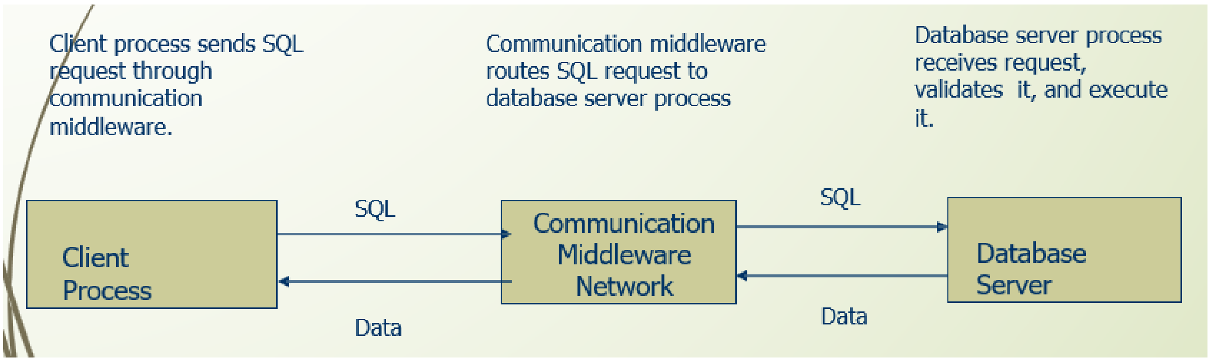
CLIENT SERVER COMPUTING

Client/Server is a term used to describe a computing model for the development of computerized systems. This model is based on the distribution of **functions** between processors: servers and clients. A client is any process that requests specific services from server processes. A server is a process that provides requested services for clients. Client and server processes can reside in the same computer or in different computers connected by a network.

Communication Middleware

It is any computer process(es) through which clients and servers communicate. The communication middleware, also known as **middleware** or the **communications layers,** is made up of several layers of software that aid the transmission of data and control information between clients and servers.

How Component Interacts


Client process sends SQL request through communication middleware.
Communication middleware routes SQL request to database server process
Database server process receives request, validates it, and execute it.
Client Process
SQL
Data
Communication Middleware Network
SQL
Data
Database Server

Client Server Principles

- Hardware Independence
- Software Independence
- Operating System
- Network System
- Applications
- Open access to services

Types of servers

1. Filee/print
2. Application
3. Database
4. Web
5. Mail
6. Hypervisor

File server

- **Stores** large number of **files** for an organisation
- Also controls **security**, such as **access rights** to files
- Finally, handles **backups** in case of accident or disaster

Print Server

- Manages **queue** of docs for printing
- So **shares load** of print requests across available devices
- Controls **access** to printers for users & routes docs to devices

Application server

- Installs, updates & controls access to **programs**
- Includes **applications** such as word processor or email
- Application runs through the main server, and client has the ability to execute it with support of frameworks.

Database Server

- Manages **security & updates** to large databases
- For example in a school, **security** means controlling access to staff
- So teachers update registers & office staff update financial data

Web Server

- **Stores all files** for a website
- Sends them when user's **browser** sends a **request** for the website
- Web server then sends the site's **files** using the **HTTP** protocol

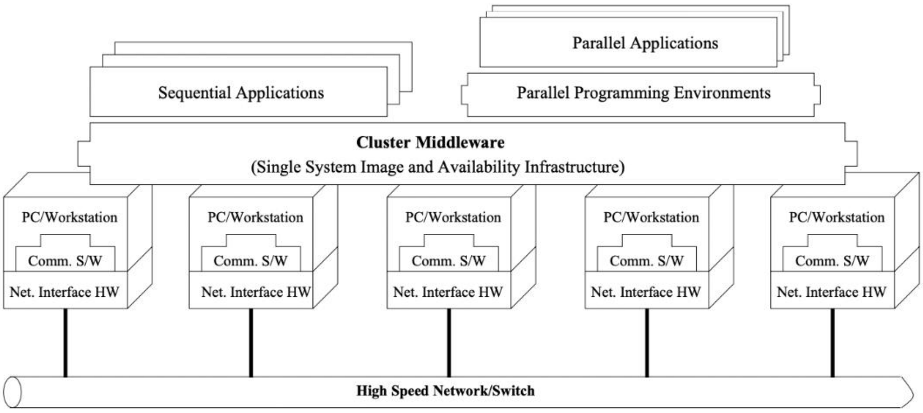
Mail Server

- **Stores & forwards** emails like an **electronic post office**
- Data traffic controlled by **protocols**, eg SMTP (send) & POP (receive)
- But email traffic is not immediate like instant messaging or 'chat'

CLUSTER COMPUTING

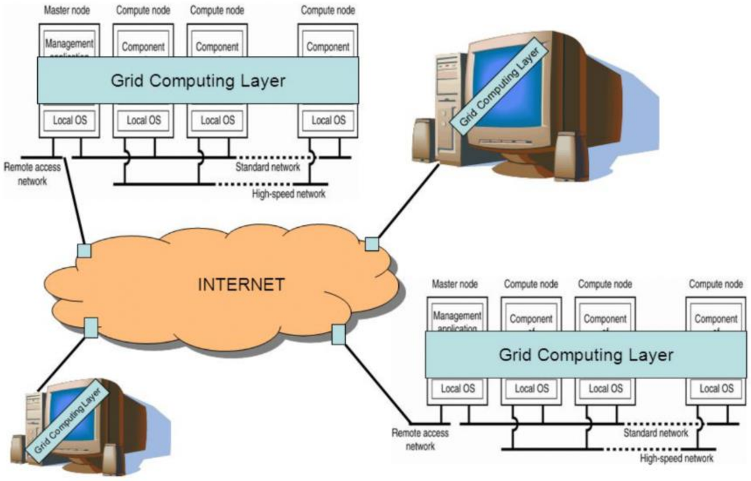
A cluster is a type of parallel or distributed processing system which consists of a collection of interconnected standalone computers working together as a **single integrated computing resource.** A computer node can be a single or multiprocessor system PCs workstations with memory IO facilities and 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.



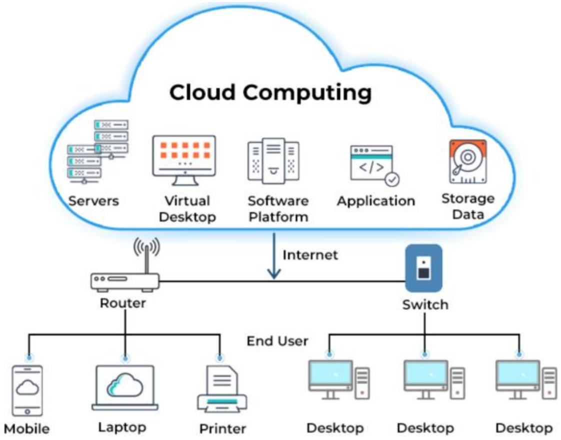
GRID COMPUTING

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



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



COMPARISON

S.No.	Parameter	Cluster	Grid	Cloud
1	Resource Handling	Centralized	Distributed	Both
2	Scalability	Complex	Yes	Yes
3	Reliability	No	Partial	Full
4	Network Type	Private	Private	Public/ Internet
5	Virtualization	Partial	Partial	Full
6	Business Model	No	No	Yes
7	Task size	Single large	Single large	Small, medium, large
8	Heterogeneity	No	Yes	Yes
9	Security	High	Medium-High	Low-High
10	Cost	Very High	High	Low

Types of cloud

1. **Public Cloud:** Share resources among the community of users. The infrastructure is made available to the general public or a large industry group and is owned by the organization selling cloud services.

2. **Private Cloud:** Services that are controlled and exclusive to certain users. The infrastructure is operated solely for an organization.

3. **Hybrid Cloud**: Ability to move workloads between private and public platforms. Composition of two or more Clouds (public, private, or community) as unique entities but bound by a standardised technology that enables data and application portability.

4. **Other types: e.g., Community/Federated Cloud:** The infrastructure is shared by several organizations and supports a community that has shared concerns.

Shared Resource and resource management:

- Cloud uses a shared pool of resources.
- Uses Internet techn. to offer **scalable** and **elastic** services.
- The term **"elastic computing**" refers to the ability of **dynamically** and **on-demand** acquiring computing resources and supporting a variable workload.
- Resources are metered and users are charged accordingly.
- It is more cost-effective due to **resource-multiplexing.**

Data storage

The data storage strategy can increase reliability, as well as security, and can lower communication costs.

Management

The maintenance and security are operated by service providers.

The service providers can operate more efficiently due to specialisation and centralisation.

Shared Resources includes

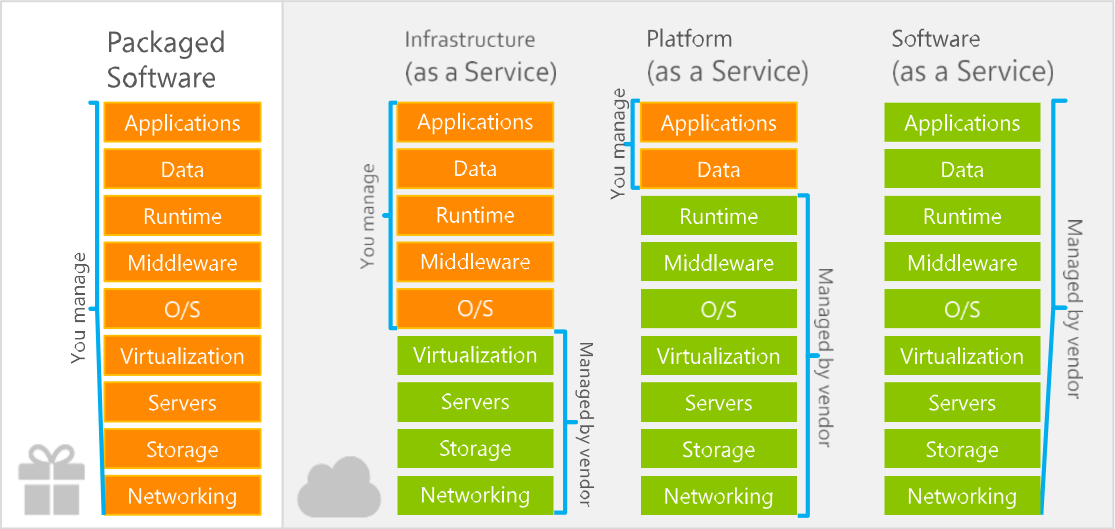
1. Resources, such as CPU , storage, network bandwidth, are **shared.**
2. When multiple applications share a system, their peak demands for resources are not synchronized thus, **multiplexing** leads to a higher resource utilization.
3. Resources can be **aggregated** to support data-intensive applications.
4. Data sharing facilitates **collaborative** activities. Many applications require multiple types of analysis of shared data sets and multiple decisions carried out by groups scattered around the globe.
5. Eliminates the **initial investment costs** for a private computing infrastructure and the maintenance and operation costs.
6. **Cost reduction**: concentration of resources creates the opportunity to pay as you go for computing.
7. **Elasticity**: the ability to accommodate workloads with very large peak-to-average ratios.
8. **User convenience**: virtualization allows users to operate in familiar environments rather than in idiosyncratic ones.

Challenges for Cloud Computing

1. **Availability of service**: what happens when the service provider cannot deliver?
2. **Data confidentiality** , a serious problem.
3. **Diversity of services**, data organization, user interfaces available at different service providers limit user mobility; once a customer is hooked to one provider it is hard to move to another.
4. **Data transfer bottleneck**; many applications are data-intensive.
5. **Performance unpredictability**, one of the consequences of resource sharing.
6. **Resource management**: It is a big challenge to manage different workloads running on large data centers. Are self-organization and self-management the solution?
7. **Security and confidentiality**: major concern for sensitive applications, e.g., healthcare applications.

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.



Cloud Activities

Service management and provisioning

- Virtualization.
- Service provisioning.
- Call center.
- Operations management.
- Systems management.
- QoS management.
- Billing and accounting, asset management.
- SLA management.
- Technical support and backups.

Security management including:

- ID and authentication.
- Certification and accreditation.
- intervention prevention.
- intervention detection.
- Virus protection.
- Cryptography.
- Physical security, incident response.
- Access control, audit and trails, and firewalls.

Customer services such as:

- Customer assistance and on-line help.

- Subscriptions.
- Business intelligence.
- Reporting.
- Customer preferences.
- Personalization.

Integration services including:

- Data management.
- Development.

TOTAL COST OF OWNERSHIP TCO

The total cost of ownership (TCO) is the purchase price of an asset plus the costs of operation. Assessing the total cost of ownership represents taking a bigger picture look at what the product is and what its value is over time.

The total cost of ownership is considered by companies and individuals when they are looking to buy assets and make investments in capital projects. For a business, the cost of purchase and the costs of operations and maintenance are often itemised separately on financial statements

When to use TCO?

TCO is useful whenever a company aims to acquire an asset or make a large investment. The metric could be relevant in situations such as. •Purchasing new computers and other tech devices.

- Renting a new office.
- Purchasing facilities for the company's headquarters.
- Hiring a new management system.

How to Calculate TCO?

- initial cost (I)
- operation cost (O)
- maintenance cost (M)
- downtime cost (D)
- production cost (P)
- remaining value (R)

Thus, the calculation will be: I + O + M + D + P – R = TCO

- The **initial cost** is the label price, that is, how much you will pay for the asset.
- The **maintenance cost,** in turn, involves the costs to ensure that the asset remains useful in the long term.
- The **remaining cost** is the asset's price in the long term, for example, in five years. This calculation focused on a possible devaluation

TCO of Keeping IT Infrastructure On-premise

There are two main considerations to make when assessing whether IT infrastructure should be purchased and installed in on-premise server rooms, or whether it should be set up "in the cloud" by partnering with a managed IT services provider.

1.What is the Total Cost of Ownership (TCO) of each approach?

2.How will each IT investment approach impact my organisation over the long term?

The TCO of on-premise IT infrastructure

There are a number of significant costs associated with installing and operating IT infrastructure inhouse:

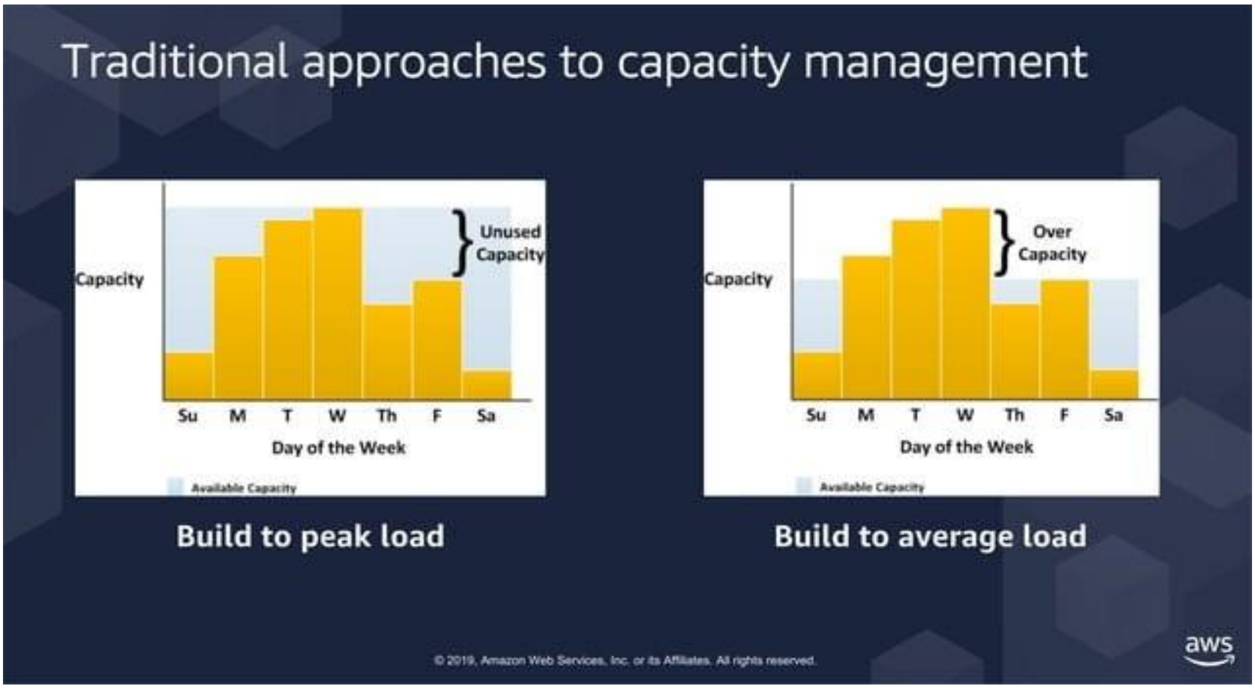
1. **Real estate costs** – you have to allocate climate-controlled, secure space to accommodate servers, storage, network infrastructure, air-conditioning units, raised floor and cabling.
2. **Infrastructure costs** – this includes the upfront costs of purchasing servers, storage, network infrastructure, air-conditioning and cabling.
3. **Hardware maintenance** –maintenance coverage for the first three years i.e. the warranty, is typically pre-paid at the time of purchase. At the end of the warranty, maintenance has to be paid again and usually at a higher rate (as the hardware is now older).
4. **Setup costs** – costs of IT staff to establish the environment and set up networking – including for remote users.
5. **Hardware and software** – IT resources required to support servers, storage, networks, and to perform updates, patches and fixes.
6. **Security & DR** – IT resources are responsible for data breaches and server failures, and IT bears the direct costs of backups and redundancy.

The TCO of managed IT services in a hosted private cloud

With cloud-based managed IT services, costs associated with Real estate, infrastructure, hardware maintenance, hardware and software currency, security are all replaced with a single monthly charge that matches with your ongoing resource consumption. Additionally, a hosted private cloud provider offers economies of scale that no on-premise alternative can come close to matching. These economies of scale include:

- ➢ Large data centres with space allocated to many private cloud tenants
- ➢ Security capabilities and costs amortised over many tenants.
- ➢ Wide use of virtualisation – especially on servers – ensuring optimal resource utilisation.
- ➢ Much greater buying power, leading to lower costs.
- ➢ Many processes are automated and standardised, leading to greater efficiencies

AWS

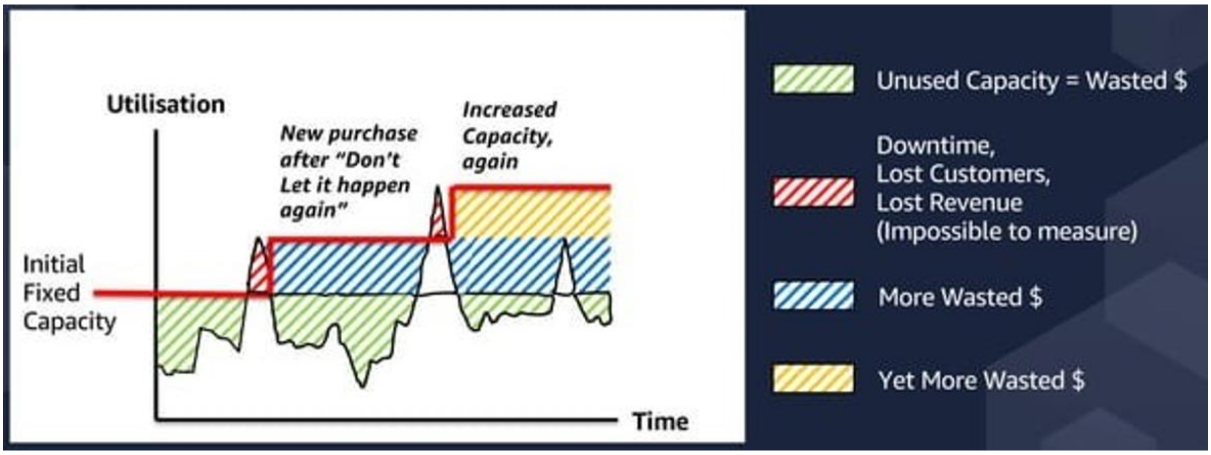


Traditional approaches to capacity management

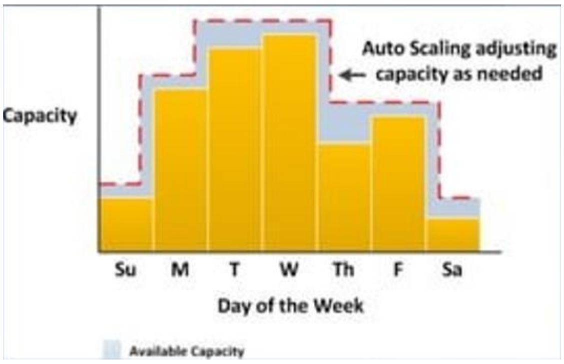
Build to peak load

Build to average load

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Auto Scaling



- React dynamically to change in load
- Schedule regular workloads
- Optimize your instance usage
- Reduce over-provisioning
- Complimentary service.

Five pillar of cost optimization

1. Measuring and monitoring
2. Right sizing your instances
3. Increase elasticity
4. Pick the right pricing model
5. Match usage to storage class

Monolithic Architecture

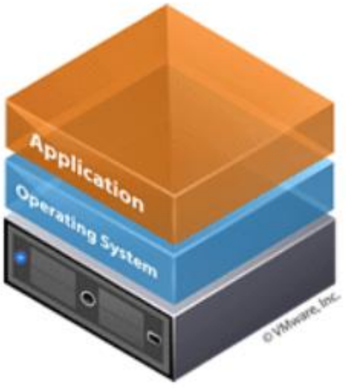
- Processes are tightly coupled
- Run as a single unit
- Risk factor is high as modules are dependent on other module
- Changing one module can cause failure
- Parallel execution not possible
- Large application donot use monolithic architecture

Microservices

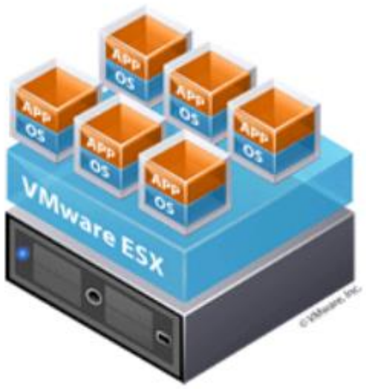
- Distributed architecture
- Application is break into independent components.
- No single unit application
- Each module have only one function
- Can modify single component without other modules to be affected

VIRTUALIZATION

- Server virtualization allows you to run multiple virtual machines on a single physical server.
- Desktop virtualization allows you to run multiple desktop machines on a single physical server, and distribute them.
- Application virtualization allows you to distribute multiple copies of an application from a single physical server.



Traditional Architecture



Virtual Architecture

Why virtualization

- Increased density
- Improves resource optimization but without sacrificing performance
- Partitioning
  - Run multiple operating systems on one physical machine
  - Share physical resources between virtual machines
- Portability
  - Entire virtual machine is saved as a file, so…
  - Move, copy, or export as easily as a file
- Security
  - Hardware is isolated from the operating system
  - Recovery as easily as restoring a file
- Agnostic
  - Migrate a virtual machine between similar, or different, physical servers

For Desktops

Business demanded:

- Cost savings
- Flexibility
- Mobility

End users demanded:

- Frequent refresh

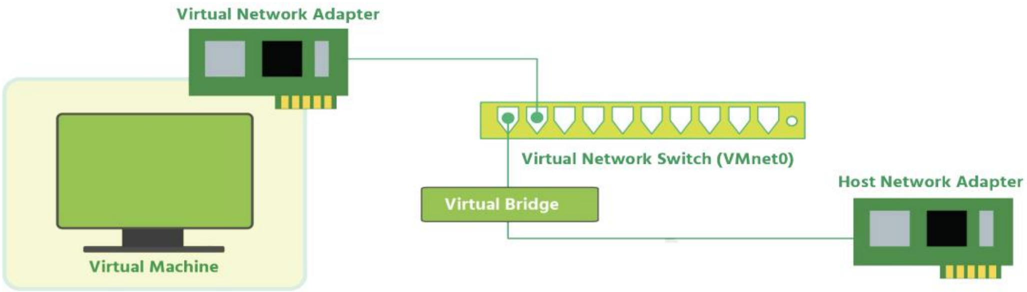
— More “power”
— Mobility
— BYOD (Bring your own device)
— Graphics

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

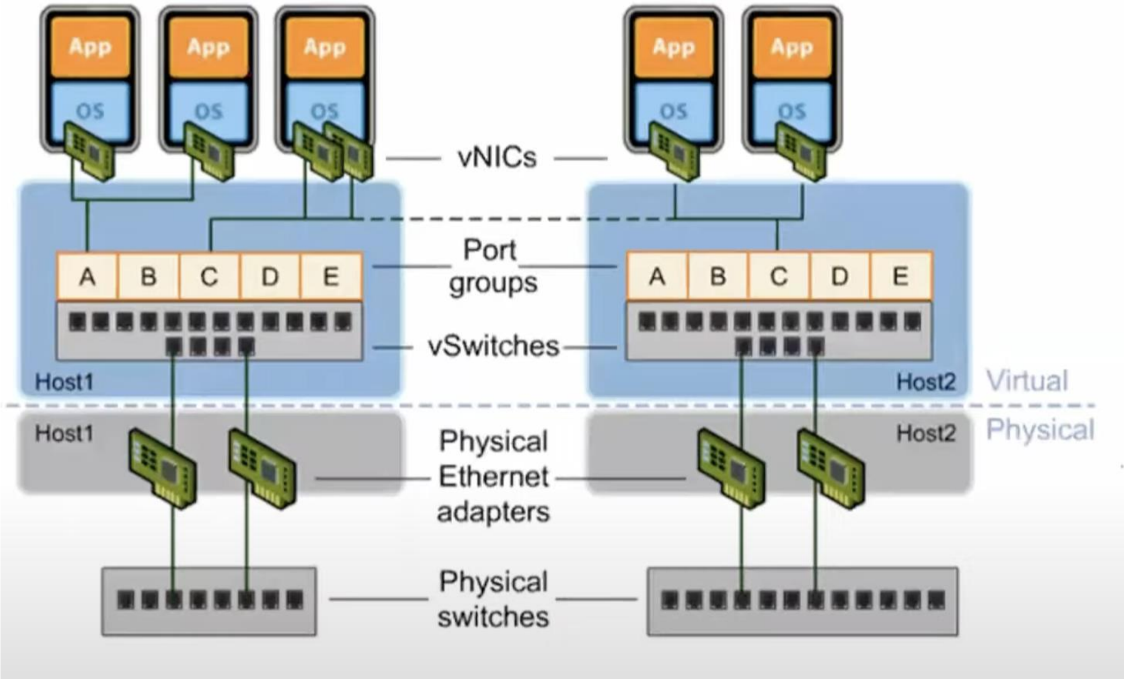
- A network type where both a virtual machine and the host that it is running on are connected to the same network
- With bridged networking, the virtual network adapter (vNIC) for the virtual machine connects to a physical NIC on the physical host system
- The host network adapter enables the VM to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



Hypervisors

Types:

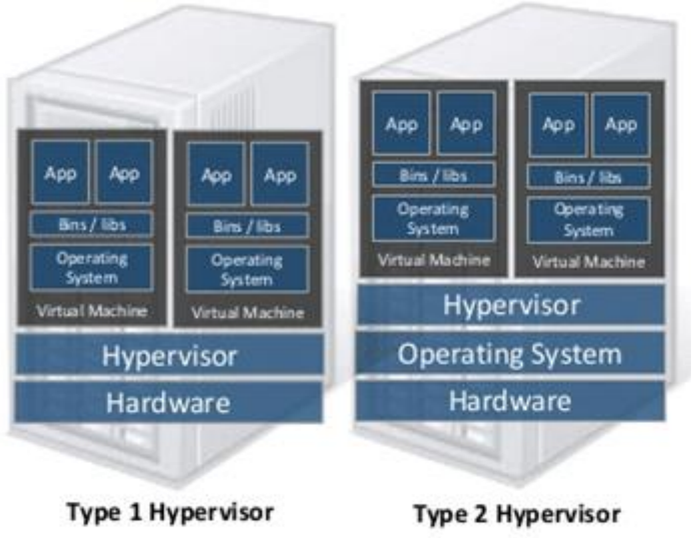
1. Hypervisor 1 Bare metal
2. Hypervisor 2

Hypervisor 1

- No OS layer
- More efficient than hypervisor 2

Hypervisor 2

Os layer included



Containers

A container is similar to an application, which runs as a process on top of the operating system (OS) and isolates with each other by running in its own **address space**. Nevertheless, more than a normal process, a container not only includes the application executable, but also packs together all the necessary software's that the application needs to run with, such as the libraries and the other dependencies.

Using containers is another way of packing applications in a much lighter weight and with a much faster delivery model. They are a fancy way of running multiple application processes on a single box, regardless of whether that box is a VM or a physical machine.

Containers are a solution to the problem of how to get software to run reliably when moved from one computing environment to another. This could be from a developer's laptop to a test environment, from a staging environment into production, and perhaps from a physical machine in a data center to a virtual machine in a private or public cloud.

VMs vs Containers

In traditional virtualization, a hypervisor virtualizes physical hardware. The result is that each virtual machine contains a guest OS, a virtual copy of the hardware that the OS requires to run, and an application and its associated libraries and dependencies.

Instead of virtualizing the underlying hardware, **containers virtualize the operating system (typically Linux)** so each individual container contains only the application and its libraries and dependencies. Containers are small, fast, and portable because unlike a virtual machine, containers do not need to include a guest OS in every instance and can, instead, simply leverage the features and resources of the host OS.

Just like virtual machines, containers allow developers **to improve CPU and memory utilization of physical machines.** Containers go even further, however, because they also enable microservice architectures, where application components can be deployed and **scaled more granularly**. This is an attractive alternative to having to scale up an entire monolithic application because a single component is struggling with load.

Advantages of Containers

Platform Independent

The biggest advantage of using Containerization is that the applications are platform-independent. A container will already contain everything that the application needs. It will come with various configuration dependencies and files. This will allow you to run your application on any computer you want. You can run applications on your physical server, local desktop or virtual server. Cloud users can also run their applications on private or public clouds. This offers great flexibility to organizations. Containers will also help you in speeding up the development process. You can easily switch from one cloud provider to another cloud provider. Also, you don’t need to worry about installing a different operating system in your system.

Efficient

You **don’t need any separate OS for running your applications**. Thus, containers will use fewer resources. VMs can take a lot of your computer resources**. You need more than 1 gigabyte of hard disk for running your virtual machine.** On the other hand, you only need a few megabytes for running your containerized application. Thus, you can run several containers on a single machine. Containers also have a high utilization level. Hence, containers are more efficient. It will help you in simplifying and reducing your regulatory compliance costs.

Effective resource sharing

You can run many containers on one server. Thus, they also use the **same pool of resources**. But, these applications will never communicate with each other. If one of your application crashes, then other applications will still keep running. They won’t face any technical issue. This will also help you in **decreasing the security risks**. Hackers can hack into your whole network by hacking into any application. But, if you are using containerization then hackers can’t use hacked applications for connecting with your network. Thus, it won’t have any effect on other applications.

Speed

Containers are lightweight when compared to VMs. You can start them in a fraction of seconds. If you are using **VM, then you need to boot an operating system first**. Also, you need to set up various things before running your VM. You can destroy or **create new containers in seconds**. Also, you can replicate applications in seconds. This will help allow your developers to work more effectively. It will also help you in improving your customer experience. Your developers can act quickly. This will help them in fixing bugs quickly.

Reproducibility

If you are using containers, then your application file systems will remain the same throughout the development phase. Version control will replace configuration management. This will help you in managing the different versions of your applications.This will allow your developers to work more quickly. It will also increase the flexibility and efficiency of your applications. Your IT team doesn't need to install VMs for testing their applications. Also, they don't need to debug your application separately for every platform.

Easy to operate

In normal virtualization, **you need various VMs for testing your applications**. These VMs consume a lot of your system resources. If you are using container technology, then you don't need to worry about installing any OS. Your application will be isolated from your operating system. Thus, you can run your applications on your normal systems. Your developers can quickly apply security patches and updates. This will also increase the productivity of your developers.

Improved productivity

There are many benefits of using container technology. A container will make sure that your application can run on any platform. You don't need to code separately for every platform. This will also remove any environmental inconsistency. Thus, the debugging and testing process will become much easier. Your developers don't need to devote hours to testing and debugging applications. All they need to do is test the application in their local system. Also, they can update your applications. You can destroy and create new containers in second. Thus, you can save a lot of your time. Tools like Docker also offer various features like version control. This will allow you to roll-out new updates.

ORCHESTRATION

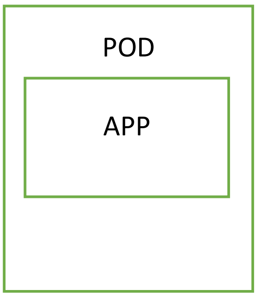
Container orchestration is the automation of much of the operational effort required to run containerized workloads and services. This includes a wide range of things software teams need to manage a container's lifecycle, including provisioning, deployment, scaling (up and down), networking, load balancing and more.

Advantages:

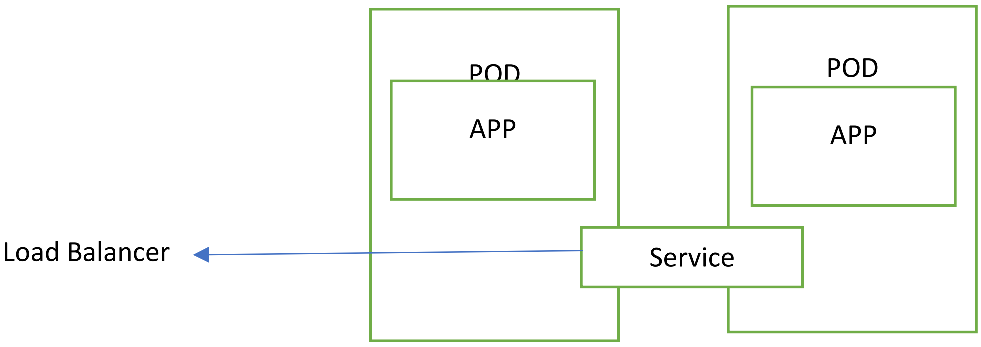
- No downtime
- high availability
- High Performance
- Scalability
- Automatic Backup solution

PODs

- Smallest unit
- 1 Pod runs 1 container at a time
- Pod can only run multiple container when there is a main container that is communicating with others
- If POD dies service dies as well



Replication



Kubernetes

Services

- Cluster IP
- NodePort
- LoadBalancer

YAML is a digestible data serialization language often used to create configuration files

Persistent Volume

- Persistent Volume is used for local storage for the container
- Persistent Volume can be used by claiming it

Advantages

- **Container deployment.** Kubernetes deploys a specified number of containers to a specified host and keeps them running in a desired state.

- **Rollouts.** A rollout is a change to a deployment. Kubernetes lets you initiate, pause, resume, or roll back rollouts.

- **Service discovery.** Kubernetes can automatically expose a container to the internet or to other containers using a DNS name or IP address.

- **Storage provisioning.** Developers can set Kubernetes to mount persistent local or cloud storage for your containers as needed.

- **Load balancing and scalability.** When traffic to a container spikes, Kubernetes can employ load balancing and scaling to distribute it across the network to ensure stability and performance. (It also saves developers the work of setting up a load balancer.)

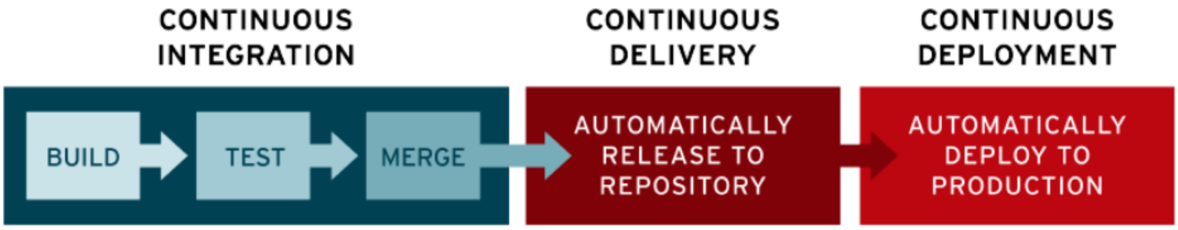
- **Self-healing for high availability.** When a container fails, Kubernetes can restart or replace it automatically. It can also take down containers that don't meet your health-check requirements.

CI/CD Continuous Integration / Continuous Delivery

A continuous integration and continuous deployment (CI/CD) pipeline is a series of steps that must be performed in order to deliver a new version of software. CI/CD pipelines are a practice focused on improving software delivery throughout the software development life cycle via automation.

A pipeline is a process that drives software development through a path of building, testing, and deploying code, also known as CI/CD. By automating the process, the objective is to minimize human error and maintain a consistent process for how software is released. Tools that are included in the pipeline could include compiling code, unit tests, code analysis, security, and binaries creation. For containerized environments, this pipeline would also include packaging the code into a container image to be deployed across a hybrid cloud.

Example: Jenkins



AWS Instance

EC2

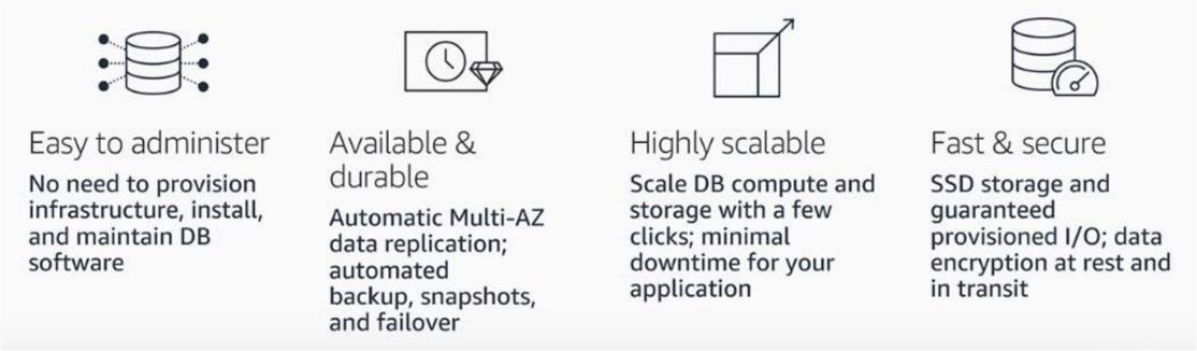
- Virtual computing environments, known as instances
- Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types
- Secure login information for your instances using key pairs
  2 formats
  .pem
  .ppk
- Storage volumes for temporary data that's deleted when you stop, hibernate, or terminate your instance, known as instance store volumes
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as Regions and Availability Zones
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups
- Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses
- 0.0.0.0/0 is for global traffic

AWS-RDS

Amazon RDS is a service which provides database connectivity through the Internet. RDS makes it very simple and easy to set -up a relational database in the cloud. Instead of concentrating on database features, you can concentrate more on the application to provide high availability, security, and compatibility. RDS is a fully managed RDBMS service.

- AWS gives all RDS services built in
- Platform as a Service
- Instance type requirement (infrastructure as a service only for this)
- Separate instance already there (choose according to requirement)

Amazon RDS managed service



Easy to administer

No need to provision infrastructure, install, and maintain DB software

Available & durable

Automatic Multi-AZ data replication; automated backup, snapshots, and failover

Highly scalable

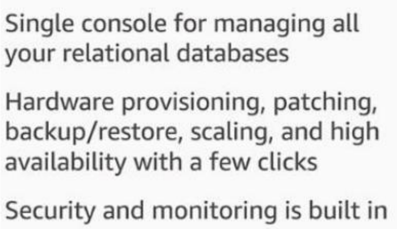
Scale DB compute and storage with a few clicks; minimal downtime for your application

Fast & secure

SSD storage and guaranteed provisioned I/O; data encryption at rest and in transit

Advantages

- **Ease of administration**

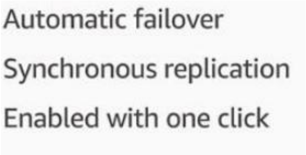


Single console for managing all your relational databases

Hardware provisioning, patching, backup/restore, scaling, and high availability with a few clicks

Security and monitoring is built in

- **Fault tolerance**

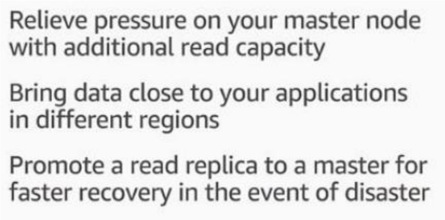


Automatic failover

Synchronous replication

Enabled with one click

- **Read Replicas**

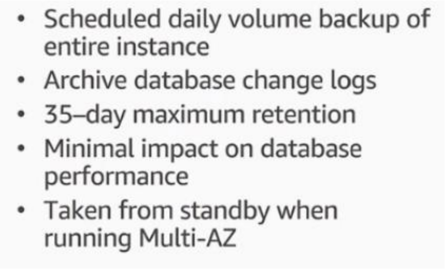


Relieve pressure on your master node with additional read capacity

Bring data close to your applications in different regions

Promote a read replica to a master for faster recovery in the event of disaster

- **Automatic Backup**

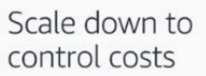


- Scheduled daily volume backup of entire instance
- Archive database change logs
- 35–day maximum retention
- Minimal impact on database performance
- Taken from standby when running Multi-AZ

- **Scale commute and storage ease**

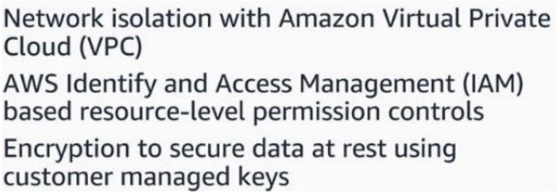
Scale compute to handle increased load

Scale storage for larger data sets



Scale down to control costs

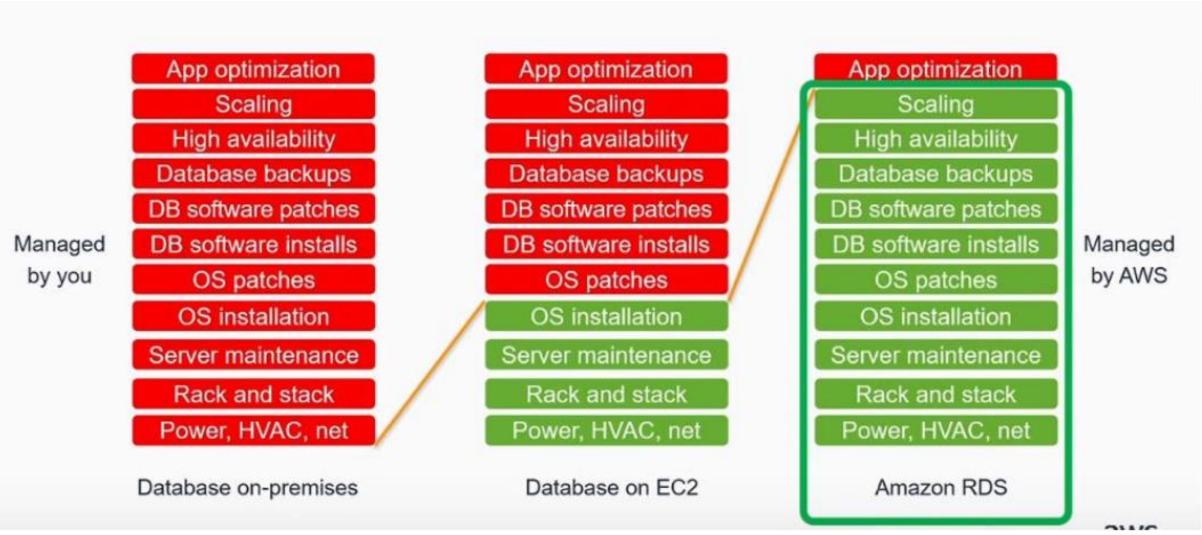
- **Security and Compliance**



Network isolation with Amazon Virtual Private Cloud (VPC)

AWS Identify and Access Management (IAM) based resource-level permission controls

Encryption to secure data at rest using customer managed keys



Database on-premises	Database on EC2	Amazon RDS
App optimization	App optimization	App optimization
Scaling	Scaling	Scaling
High availability	High availability	High availability
Database backups	Database backups	Database backups
DB software patches	DB software patches	DB software patches
DB software installs	DB software installs	DB software installs
OS patches	OS patches	OS patches
OS installation	OS installation	OS installation
Server maintenance	Server maintenance	Server maintenance
Rack and stack	Rack and stack	Rack and stack
Power, HVAC, net	Power, HVAC, net	Power, HVAC, net

Cloud Security and Solution

Misconfiguration

Misconfigurations of cloud security settings are a leading cause of cloud data breaches. Many organizations' cloud security posture management strategies are inadequate for protecting their cloud-based infrastructure.

Unauthorized Access

Unlike an organization's on-premises infrastructure, their cloud-based deployments are outside the network perimeter and directly accessible from the public Internet. While this is an asset for the accessibility of this infrastructure to employees and customers, it also makes it easier for an attacker to gain unauthorized access to an organization's cloud-based resources. Improperly-configured security or compromised credentials can enable an attacker to gain direct access, potentially without an organization's knowledge.

Insecure Interfaces/APIs

Cloud often provide a number of application programming interfaces (APIs) and interfaces for their customers. In general, these interfaces are well-documented in an attempt to make them easily-usable for customers.

Hijacking of Accounts

Many people have extremely weak password security, including password reuse and the use of weak passwords. This problem exacerbates the impact of phishing attacks and data breaches since it enables a single stolen password to be used on multiple different accounts.

Account hijacking is one of the more serious cloud security issues as organizations are increasingly reliant on cloud-based infrastructure and applications for core business functions. Additionally, in the cloud, organizations often lack the ability to identify and respond to these threats as effectively as for on-premises infrastructure.

Lack of Visibility

An organization's cloud-based resources are located outside of the corporate network and run on infrastructure that the company does not own. As a result, many traditional tools for achieving network visibility are not effective for cloud environments, and some organizations lack cloud-focused security tools. This can limit an organization's ability to monitor their cloud-based resources and protect them against attack.

External Sharing of Data

The cloud is designed to make data sharing easy. Many clouds provide the option to explicitly invite a collaborator via email or to share a link that enables anyone with the URL to access the shared resource.

While this easy data sharing is an asset, it can also be a major cloud security issue. The use of link-based sharing – a popular option since it is easier than explicitly inviting each intended collaborator – makes it difficult to control access to the shared resource.

Malicious Insiders

Insider threats are a major security issue for any organization. A malicious insider already has authorized access to an organization's network and some of the sensitive resources that it contains. Attempts to gain this level of access are what reveals most attackers to their target, making it hard for an unprepared organization to detect a malicious insider.

On the cloud, detection of a malicious insider is even more difficult. With cloud deployments, companies lack control over their underlying infrastructure, making many traditional security solutions less effective. This, along with the fact that cloud-based infrastructure is directly accessible from the public Internet and often suffers from security misconfigurations, makes it even more difficult to detect malicious insiders.

Cyberattacks

Cybercrime is a business, and cybercriminals select their targets based upon the expected profitability of their attacks. Cloud-based infrastructure is directly accessible from the public Internet, is often improperly secured, and contains a great deal of sensitive and valuable data. Additionally, the cloud is used by many different companies, meaning that a successful attack can likely be repeated many times with a high probability of success. As a result, organizations' cloud deployments are a common target of cyberattacks.

Denial of Service Attacks

The cloud is essential to many organizations' ability to do business. They use the cloud to store business-critical data and to run important internal and customer-facing applications.

This means that a successful Denial of Service (DoS) attack against cloud infrastructure is likely to have a major impact on a number of different companies. As a result, DoS attacks where the attacker demands a ransom to stop the attack pose a significant threat to an organization's cloud-based resources.

Data Loss/Leakage

Cloud-based environments make it easy to share the data stored within them. These environments are accessible directly from the public Internet and include the ability to share data easily with other parties via direct email invitations or by sharing a public link to the data.

The ease of data sharing in the cloud – while a major asset and key to collaboration in the cloud – creates serious concerns regarding data loss or leakage. In fact, 69% of organizations point to this as their greatest cloud security concern. Data sharing using public links or setting a cloud-based repository

to public makes it accessible to anyone with knowledge of the link, and tools exist specifically for searching the Internet for these unsecured cloud deployments.

Data Residence/Control

Most cloud providers have a number of geographically distributed data centers. This helps to improve the accessibility and performance of cloud-based resources and makes it easier for cloud to ensure that they are capable of maintaining service level agreements in the face of business-disrupting events such as natural disasters, power outages, etc.

Cloud security solutions

Organizations seeking cloud security solutions should consider the following criteria to solve the primary cloud security challenges of visibility and control over cloud data.

Visibility into cloud data — A complete view of cloud data requires direct access to the cloud service. Cloud security solutions accomplish this through an application programming interface (API) connection to the cloud service. With an API connection it is possible to view:

- What data is stored in the cloud.
- Who is using cloud data?

- The roles of users with access to cloud data.
- Who cloud users are sharing data with.
- Where cloud data is located.
- Where cloud data is being accessed and downloaded from, including from which device.

Control over cloud data — Once you have visibility into cloud data, apply the controls that best suit your organization. These controls include:

Data classification — Classify data on multiple levels, such as sensitive, regulated, or public, as it is created in the cloud. Once classified, data can be stopped from entering or leaving the cloud service.

Data Loss Prevention (DLP) — Implement a cloud DLP solution to protect data from unauthorized access and automatically disable access and transport of data when suspicious activity is detected.

Collaboration controls — Manage controls within the cloud service, such as downgrading file and folder permissions for specified users to editor or viewer, removing permissions, and revoking shared links.

Encryption — Cloud data encryption can be used to prevent unauthorized access to data, even if that data is ex-filtrated or stolen.

Access to cloud data and applications— As with in-house security, access control is a vital component of cloud security. Typical controls include:

User access control — Implement system and application access controls that ensure only authorized users access cloud data and applications. A Cloud Access Security Broker (CASB) can be used to enforce access controls

Device access control — Block access when a personal, unauthorized device tries to access cloud data.

Malicious behavior identification — Detect compromised accounts and insider threats with user behavior analytics (UBA) so that malicious data exfiltration does not occur.

Malware prevention — Prevent malware from entering cloud services using techniques such as file-scanning, application whitelisting, machine learning-based malware detection, and network traffic analysis.

Privileged access — Identify all possible forms of access that privileged accounts may have to your data and applications, and put in place controls to mitigate exposure.

Compliance — Existing compliance requirements and practices should be augmented to include data and applications residing in the cloud.

Risk assessment — Review and update risk assessments to include cloud services. Identify and address risk factors introduced by cloud environments and providers. Risk databases for cloud providers are available to expedite the assessment process.

Compliance Assessments — Review and update compliance assessments on cloud security standards.

Created By: Mr.Nobody….:-)