Cloud Computing

- Cloud computing is the delivery of on-demand computing services, from applications to storage and processing power, typically over the internet and on a pay-as-you-go basis.

Introduction to Cloud Computing

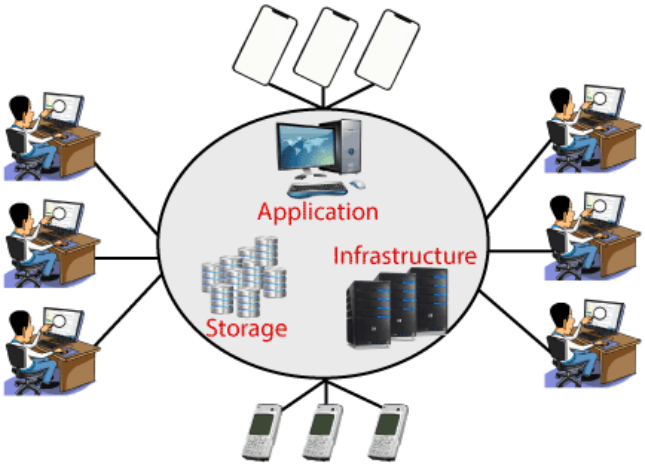


Fig. Cloud Computing

- Cloud Computing is the delivery of computing services such as servers, storage, databases, networking, software, analytics, intelligence, and more, over the Cloud (Internet).
- Cloud Computing provides an alternative to the on-premises datacenter.
- With an on-premises datacenter, we have to manage everything, such as purchasing and installing hardware, virtualization, installing the operating system, and any other required applications, setting up the network, configuring the firewall, and setting up storage for data.
- After doing all the set-up, we become responsible for maintaining it through its entire lifecycle.
- But if we choose Cloud Computing, a cloud vendor is responsible for the hardware purchase and maintenance.
- They also provide a wide variety of software and platform as a service.
- We can take any required services on rent.
- The cloud computing services will be charged based on usage.
- The cloud environment provides an easily accessible online portal that makes handy for the user to manage the compute, storage, network, and application resources.

Characteristics (Features) of Cloud Computing

The five essential characteristics of cloud computing:

1. **On-demand self-service:** A consumer can separately provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
2. **Broad network access:** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops and workstations).
3. **Resource pooling:** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned

according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state or datacenter). Examples of resources include storage, processing, memory and network bandwidth.

4. **Rapid elasticity:** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward matching with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
5. **Measured service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth and active user accounts). Resource usage can be monitored, controlled and reported, providing transparency for the provider and consumer.

Advantages of Cloud Computing

- **Cost:** It reduces the huge capital costs of buying hardware and software.
- **Speed:** Resources can be accessed in minutes, typically within a few clicks.
- **Scalability:** We can increase or decrease the requirement of resources according to the business requirements.
- **Productivity:** While using cloud computing, we put less operational effort. We do not need to apply patching, as well as no need to maintain hardware and software. So, in this way, the IT team can be more productive and focus on achieving business goals.
- **Reliability:** Backup and recovery of data are less expensive and very fast for business continuity.
- **Security:** Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

Disadvantages of Cloud Computing

- **Requires good speed internet with good bandwidth**: To access your cloud services, you need to have a good internet connection always with good bandwidth to upload or download files to/from the cloud
- **Downtime**: Since the cloud requires high internet speed and good bandwidth, there is always a possibility of service outage, which can result in business downtime. Today, no business can afford revenue or business loss due to downtime or slow down from an interruption in critical business processes.
- **Limited control of infrastructure**: Since you are not the owner of the infrastructure of the cloud, hence you don't have any control or have limited access to the cloud infra.
- **Restricted or limited flexibility**: The cloud provides a huge list of services, but consuming them comes with a lot of restrictions and limited flexibility for your applications or developments. Also, platform dependency or 'vendor lock-in' can sometimes make it difficult for you to migrate from one provider to another.
- **Ongoing costs**: Although you save your cost of spending on whole infrastructure and its management, on the cloud, you need to keep paying for services as long as you use them. But in traditional methods, you only need to invest once.
- **Security**: Security of data is a big concern for everyone. Since the public cloud utilizes the internet, your data may become vulnerable. In the case of a public cloud, it depends on the cloud provider to take care of your data. So, before opting for cloud services, it is required that you find a provider who follows maximum compliance policies for data security.

- **Vendor Lock-in**: Although the cloud service providers assure you that they will allow you to switch or migrate to any other service provider whenever you want, it is a very difficult process. You will find it complex to migrate all the cloud services from one service provider to another. During migration, you might end up facing compatibility, interoperability and support issues. To avoid these issues, many customers choose not to change the vendor.
- **Technical issues**: Even if you are a tech whiz, the technical issues can occur, and everything can't be resolved in-house. To avoid interruptions, you will need to contact your service provider for support. However, not every vendor provides 24/7 support to their clients.

Difference between Conventional Computing and Cloud Computing

Conventional Computing	Cloud Computing
In conventional computing environment more time is needed for installation, set up, and configuration.	Once the cloud computing environment is set up initially, you can gain access faster than conventional computing
Cost must be paid in advance	Pay-as-you-go
Cost is fixed	Cost is variable
Economic to scale for all organization	Economic to scale for large organization only
For Scaling manual effort is needed	Scaling can be elastic and automatic
Environment is mix of physical and virtualized	Usually environment is virtualized

History of Cloud Computing

- Before emerging the cloud computing, there was Client/Server computing which is basically a centralized storage in which all the software applications, all the data and all the controls are resided on the server side.
- If a single user wants to access specific data or run a program, he/she need to connect to the server and then gain appropriate access, and then he/she can do his/her business.
- Then after, distributed computing came into picture, where all the computers are networked together and share their resources when needed.
- On the basis of above computing, there was emerged of cloud computing concepts that later implemented.
- At around in 1961, John MacCharty suggested in a speech at MIT that computing can be sold like a utility, just like a water or electricity.
- It was a brilliant idea, but like all brilliant ideas, it was ahead of its time, as for the next few decades, despite interest in the model, the technology simply was not ready for it.
- But of course time has passed and the technology caught that idea and after few years we mentioned that:
    - In 1999, Salesforce.com started delivering of applications to users using a simple website. The applications were delivered to enterprises over the Internet, and this way the dream of computing sold as utility were true.
    - In 2002, Amazon started Amazon Web Services, providing services like storage, computation and even human intelligence. However, only starting with the launch of the Elastic Compute Cloud in 2006 a truly commercial service open to everybody existed.
    - In 2009, Google Apps also started to provide cloud computing enterprise applications.
    - Of course, all the big players are present in the cloud computing evolution, some were earlier and some were later. In 2009, Microsoft launched Windows Azure, and companies like Oracle and HP have all joined the game. This proves that today, cloud computing has become mainstream.

Cloud Orchestration

- Cloud Orchestration is a way to manage, co-ordinate, and provision all the components of a cloud platform automatically from a common interface.
- It orchestrates the physical as well as virtual resources of the cloud platform.
- Cloud orchestration is a must because cloud services scale up arbitrarily and dynamically, include fulfillment assurance and billing, and require workflows in various business and technical domains.
- Orchestration tools combine automated tasks by interconnecting the processes running across the heterogeneous platforms in multiple locations.
- Orchestration tools create declarative templates to convert the interconnected processes into a single workflow.
- The processes are so orchestrated that the new environment creation workflow is achieved with a single API call.
- Creation of these declarative templates, though complex and time consuming, is simplified by the orchestration tools.
- Cloud orchestration includes two types of models:
  - Single Cloud model
  - Multi-cloud model
- In Single cloud model, all the applications designed for a system run on the same IaaS platform (same cloud service provider).
- Applications, interconnected to create a single workflow, running on various cloud platforms for the same organization define the concept of multi-cloud model.
- IaaS requirement for some applications, though designed for same system, might vary. This results in availing services of multiple cloud service providers.
- For example, application with patient's sensitive medical data might reside in some IaaS, whereas the application for online OPD appointment booking might reside in another IaaS, but they are interconnected to form one system. This is called multi-cloud orchestration.
- Multi-cloud models provide high redundancy as compared to single IaaS deployments.
- This reduces the risk of down time.

Elasticity in Cloud

- Elasticity covers the ability to scale up but also the ability to scale down.
- The idea is that you can quickly provision new infrastructure to handle a high load of traffic.
- But what happens after that rush? If you leave all of these new instances running, your bill will skyrocket as you will be paying for unused resources.
- In the worst case scenario, these resources can even cancel out revenue from the sudden rush.
- An elastic system prevents this from happening. After a scaled up period, your infrastructure can scale back down, meaning you will only be paying for your usual resource usage and some extra for the high traffic period.
- The key is that this all happens automatically.
- When resource needs meet a certain threshold (usually measured by traffic), the system "knows" that it needs to de-provision a certain amount of infrastructure, and does so.
- With a couple hours of training, anyone can use the AWS web console to manually add or subtract instances.
- But it takes a true Solutions Architect to set up monitoring, account for provisioning time, and configure a system for maximum elasticity.

Cloud Service Options / Cloud Service Models / Cloud Computing Stack

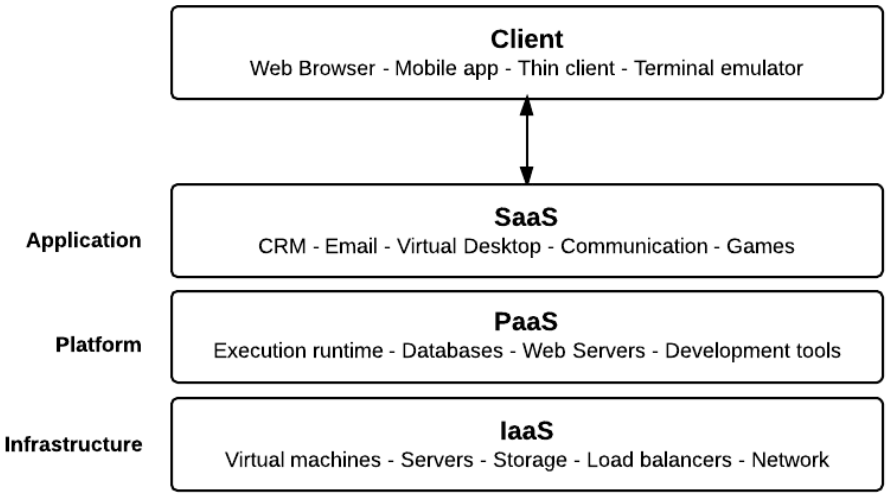


Fig. : Cloud Services

- Cloud computing is a model for enabling ubiquitous, 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.
- Although cloud computing has evolved over the time it has been majorly divided into three broad service categories:
  1. Infrastructure as a Service(IAAS),
  2. Platform as a Service (PAAS) and
  3. Software as a Service (SAAS)

1. Infrastructure as a Service (IAAS)

- Infrastructure as a Service (IAAS) is a form of cloud computing that provides virtualized computing resources over the internet.
- In an IAAS model, a third party provider hosts hardware, software, servers, storage and other infrastructure components on the behalf of its users.
- IAAS providers also host users' applications and handle tasks including system maintenance backup and resiliency planning.
- IAAS platforms offer highly scalable resources that can be adjusted on-demand which makes it a well-suited for workloads that are temporary, experimental or change unexpectedly.
- Other characteristics of IAAS environments include the automation of administrative tasks, dynamic scaling, desktop virtualization and policy based services.
- Technically, the IaaS market has a relatively low barrier of entry, but it may require substantial financial investment in order to build and support the cloud infrastructure.
- Mature open-source cloud management frameworks like OpenStack are available to everyone, and provide strong a software foundation for companies that want to build their private cloud or become a public cloud provider.

IAAS- Network:

- There are two major network services offered by public cloud service providers:

1. load balancing and
2. DNS (domain name systems).

- Load balancing provides a single point of access to multiple servers that run behind it. A load balancer is a network device that distributes network traffic among servers using specific load balancing algorithms.
- DNS is a hierarchical naming system for computers, or any other naming devices that use IP addressing for network identification – a DNS system associates domain names with IP addresses.

2. Platform as a Service (PAAS)

- Platform as a Service (PAAS) is a cloud computing model that delivers applications over the internet.
- In a PAAS model, a cloud provider delivers hardware and software tools, usually those needed for application development, to its users as a service.
- A PAAS provider hosts the hardware and software on its own infrastructure. As a result, PAAS frees users from having to install in-house hardware and software to develop or run a new application.
- PAAS doesn't replace a business' entire infrastructure but instead a business relies on PAAS providers for key services, such as Java development or application hosting.
- A PAAS provider, however, supports all the underlying computing and software, users only need to login and start using the platform-usually through a Web browser interface.
- PAAS providers then charge for that access on a per-use basis or on monthly basis.
- Some of the main characteristics of PAAS are :
  1) Scalability and auto-provisioning of the underlying infrastructure.
  2) Security and redundancy.
  3) Build and deployment tools for rapid application management and deployment.
  4) Integration with other infrastructure components such as web services, databases, and LDAP.
  5) Multi-tenancy, platform service that can be used by many concurrent users.
  6) Logging, reporting, and code instrumentation.
  7) Management interfaces and/or API.

3. Software as a Service (SAAS)

- Software as a Service (SAAS) is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet.
- SAAS has become increasingly prevalent delivery model as underlying technologies that support Web services and service- oriented architecture (SOA) mature and new development approaches, such as Ajax, become popular.
- SAAS is closely related to the ASP (Application service provider) and on demand computing software delivery models.
- IDC identifies two slightly different delivery models for SAAS which are
  1) the hosted application model and
  2) the software development model.
- Some of the core benefits of using SAAS model are:
  1) Easier administration.
  2) Automatic updates and patch management.
  3) Compatibility: all users will have the same version of software.
  4) Easier collaboration, for the same reason.
  5) Global accessibility.

Issues of SaaS

Permanent Internet connection

- Employees using SaaS software services must be permanently connected to the Internet.
- Working offline is no longer an option in this situation.
- We all know an Internet connection is not a problem anymore nowadays for those working in offices or home.
- Companies needing assurance that their employees always have a connection to their SaaS provider should consider redundant high speed Internet connections.
- Are you using mobile devices or travelling constantly? The best solution might be Software plus Service.

Data security

- When it comes to migrating traditional local software applications to a cloud based platform, data security may be a problem.
- When a computer and application is compromised the SaaS multi-tenant application supporting many customers could be exposed to the hackers.
- Any provider will promise that it will do the best in order for the data to be secure in any circumstances.
- But just to make sure, you should ask about their infrastructure and application security.

Data control

- Many businesses have no idea how their SaaS provider will secure their data or what backup procedures will be applied when needed.
- To avoid undesirable effects, before choosing a SaaS vendor, managers should research for providers with good reputations and that the vendor has backup solutions which are precisely described in the Service Level Agreement contract.

Data location

- This means being permanently aware where exactly in the world your data is located.
- Although the Federal Information Security Management Act in the USA requires customers to keep sensitive data within the country, in virtualized systems, data can move dynamically from one country to another.
- Ask about the laws for your customers data in respect to where they are located.

Cloud Deployment Models

Following are the four types of Cloud Deployment Models identified by NIST.

1. Private Cloud
2. Community Cloud
3. Public Cloud
4. Hybrid Cloud

1. Private Cloud

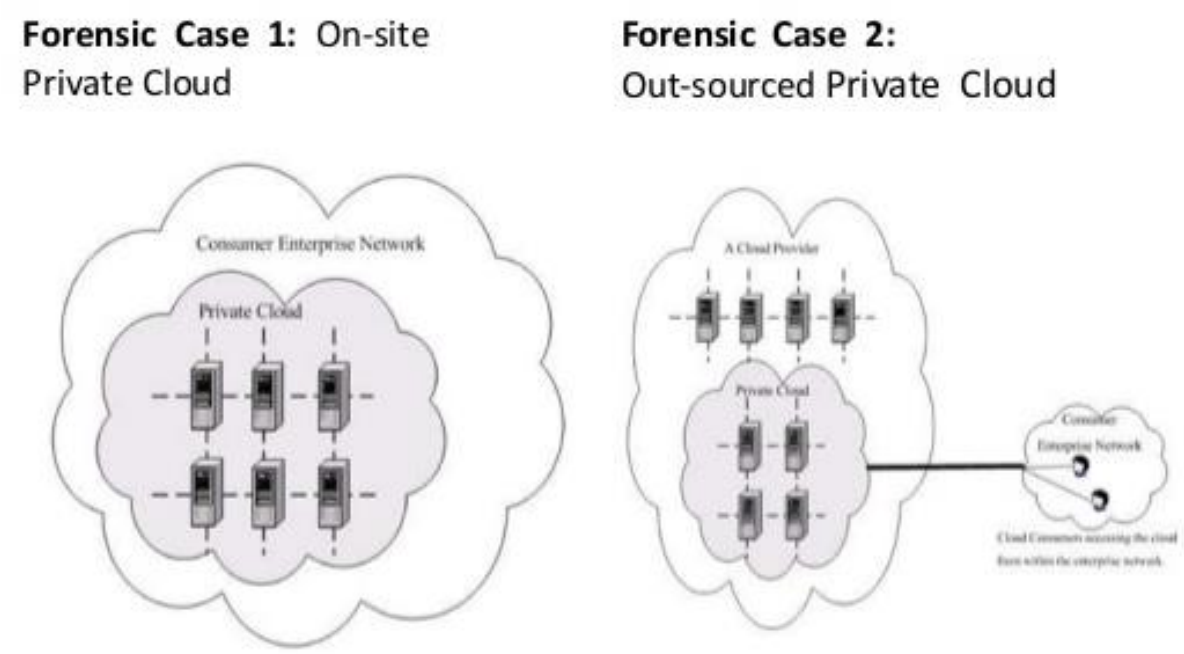


Fig.: Private Cloud

- The cloud infrastructure is operated solely for an organization.
- Contrary to popular belief, private cloud may exist off premises and can be managed by a third party. Thus, two private cloud scenarios exist, as follows:

On-site Private Cloud

- Applies to private clouds implemented at a customer's premises.

Outsourced Private Cloud

- Applies to private clouds where the server side is outsourced to a hosting company.

Examples of Private Cloud:

- Eucalyptus, Ubuntu Enterprise Cloud - UEC (powered by Eucalyptus), Amazon VPC (Virtual Private Cloud), VMware Cloud Infrastructure Suite, Microsoft ECI data center etc.

2. Community Cloud

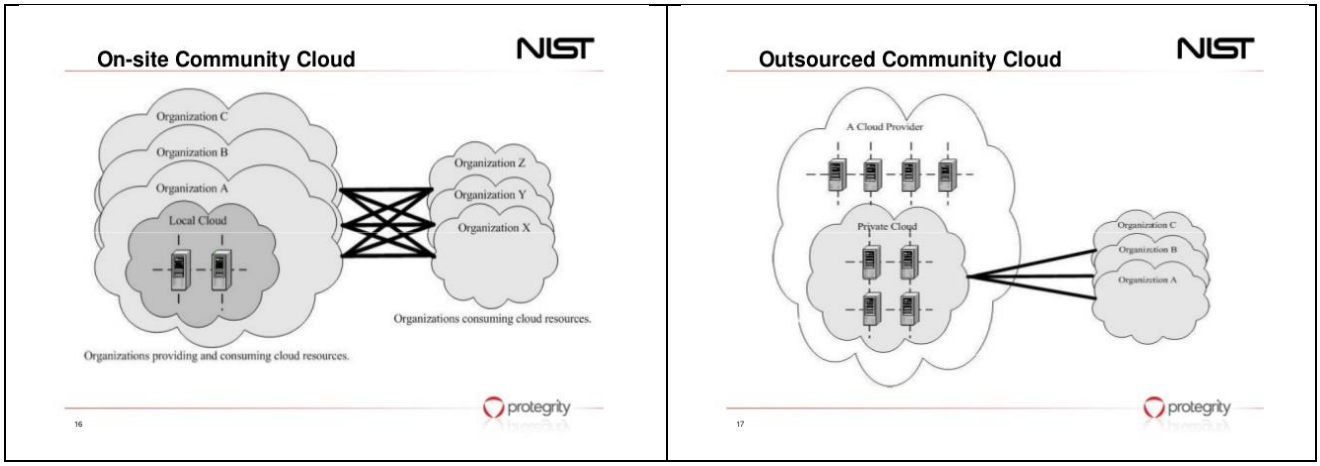


Fig. Community Cloud

- The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
- Government departments, universities, central banks etc. often find this type of cloud useful.
- Community cloud also has two possible scenarios:

On-site Community Cloud Scenario

- Applies to community clouds implemented on the premises of the customers composing a community cloud.

Outsourced Community Cloud

- Applies to community clouds where the server side is outsourced to a hosting company.

Examples of Community Cloud:

- Google Apps for Government, Microsoft Government Community Cloud, etc.

3. Public Cloud

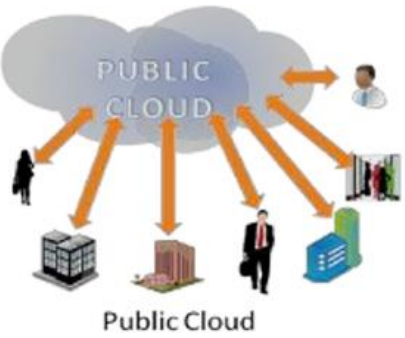


Fig. : Public Cloud

- The most ubiquitous, and almost a synonym for, cloud computing.
- The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.

Examples of Public Cloud:

- Google App Engine, Microsoft Windows Azure, IBM Smart Cloud, Amazon EC2, etc.

4. Hybrid Cloud

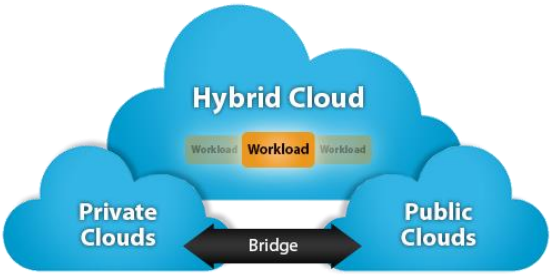


Fig. : Hybrid Cloud

- The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

Examples of Hybrid Cloud:

- Windows Azure (capable of Hybrid Cloud), VMware vCloud (Hybrid Cloud Services), etc.

Eucalyptus

- Eucalyptus is an open source software platform for implementing Infrastructure as a Service (IaaS) in a private or hybrid cloud computing environment.
- The Eucalyptus cloud platform pools together existing virtualized infrastructure to create cloud resources for infrastructure as a service, network as a service and storage as a service.
- The name Eucalyptus is an acronym for Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems.
- Eucalyptus was founded out of a research project in the Computer Science Department at the University of California, Santa Barbara, and became a for-profit business called Eucalyptus Systems in 2009.
- Eucalyptus Systems announced a formal agreement with Amazon Web Services (AWS) in March 2012, allowing administrators to move instances between a Eucalyptus private cloud and the Amazon Elastic Compute Cloud (EC2) to create a hybrid cloud.
- The partnership also allows Eucalyptus to work with Amazon's product teams to develop unique AWS-compatible features.

Eucalyptus features

- Supports both Linux and Windows virtual machines (VMs).
- Application program interface- (API) compatible with Amazon EC2 platform.
- Compatible with Amazon Web Services (AWS) and Simple Storage Service (S3).
- Works with multiple hypervisors including VMware, Xen and KVM.
- Can be installed and deployed from source code or DEB and RPM packages.
- Internal processes communications are secured through SOAP and WS-Security.
- Multiple clusters can be virtualized as a single cloud.
- Administrative features such as user and group management and reports.

Business Concerns in the Cloud

Security

- Due to the nature of cloud computing services and how they involve storing data without knowing its precise physical location, data security remains a concern for both prospective adopters of the technology and existing users.
- However, the security concerns associated with storing things in the cloud are more nuanced than merely not being able to see where data is stored. A number of data breaches involving cloud systems made the headlines in 2017, including the story of financial giant Deloitte having its cloud data compromised.
- These combined with the natural carefulness of trusting third parties with data makes information security a persistent challenge in cloud computing. However, with each breach comes enhanced security in cloud systems designed to ensure similar breaches never happen again. Improvements include the use of multi-factor authentication, implemented to ensure users are who they claim to be.
- Truth be told, security for most cloud providers is watertight, and breaches in the cloud are rare—when they do occur, though, they get all the headlines. To minimize risk, double-check that your cloud provider uses secure user identity management and access controls. It's also important to check which data security laws your cloud provider must follow. On the whole, cloud data security is as safe, if not safer, than on premise data security.

Outages

- Performance is a consistent challenge in cloud computing, particularly for businesses that rely on cloud providers to help them run mission-critical applications. When a business moves to the cloud it becomes dependent on the cloud provider, meaning that any outages suffered by the cloud provider also affect the business.
- The risk of outages in the cloud is not negligible—even the major players in cloud computing are susceptible. In February 2017, an AWS Amazon S3 outage caused disruptions for many websites and applications, and even sent them offline.
- There is a need, therefore, for some kind of site recovery solution for data held in cloud-based services. Disaster recovery as a service (DRaaS)—the replication and hosting of servers by a third party to provide failover in the event of a man-made or natural catastrophe—is a way companies can maintain business continuity even when disaster strikes.

Expertise

- The success of any movement towards cloud adoption comes down to the expertise at your disposal. The complexity of cloud technology and the sheer range of tools makes it difficult to keep up with the options available for all your use cases.
- Organizations need to strike a balance between having the right expertise and the cost of hiring dedicated cloud specialists. The optimum solution to this challenge is to work with a trusted cloud Managed Service Provider (MSP). Cloud MSPs have the manpower, tools and experience to manage multiple and complex customer environments simultaneously. The MSP takes complete responsibility for cloud processes and implementing them as the customer desires. This way, organizations can stay focused on their business goals.

Cost Management

- All the main cloud providers have quite detailed pricing plans for their services that explicitly define costs of processing and storage data in the cloud. The problem is that cost management is often an issue when using cloud services because of the sheer range of options available.
- Businesses often waste money on unused workloads or unnecessarily expensive storage, and 26 percent of respondents in this cloud survey cited cost management as a major challenge in the cloud. The solution is for organizations to monitor their cloud usage in detail and constantly optimize their choice of services, instances, and storage. You can monitor and optimize cloud implementation by using a cloud cost management tool such as CloudHealth or consulting a cloud cost expert.
- There are also some practical cost calculators available which clarify cloud costs, including Amazon's AWS Simple Monthly Calculator, and NetApp's calculators for both AWS and Azure cloud storage.

Governance

- Cloud governance, meaning the set of policies and methods used to ensure data security and privacy in the cloud, is a huge challenge. Confusion often arises about who takes responsibility for data stored in the cloud, who should be allowed use cloud resources without first consulting IT personnel, and how employees handle sensitive data.
- The only solution is for the IT department at your organization to adapt its existing governance and control processes to incorporate the cloud and ensure everyone is on the same page. This way, proper governance, compliance, and risk management can be enforced.

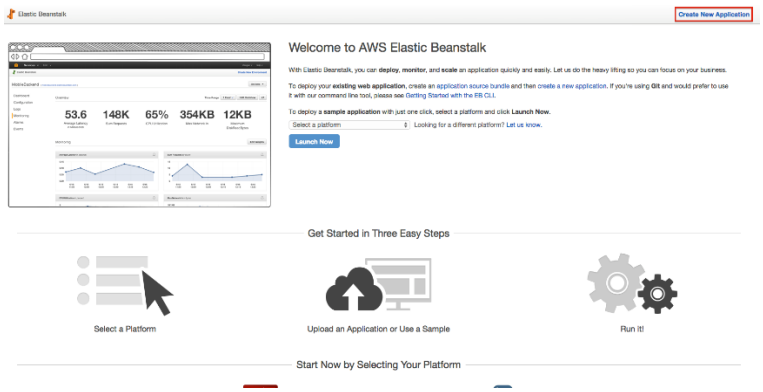
Cloud Optimization Strategy

- Finding the right strategy for cloud adoption is another important challenge. Many businesses moved to the cloud using a segmented approach in which isolated use cases, projects, and applications were migrated to cloud providers. The problem then for many companies is a lack of any holistic organization-wide cloud strategy.
- Finding the right strategy for cloud adoption comes back to the issue of cloud governance. With everyone on the same page thanks to robust cloud governance and clear policies, organizations can create a unified and optimized strategy for how they use the cloud.

Steps to Launch an Application with AWS Elastic Beanstalk

Step 1: Create a New Application

- Now that you're in the AWS Elastic Beanstalk dashboard, click on Create New Application to create and configure your application.

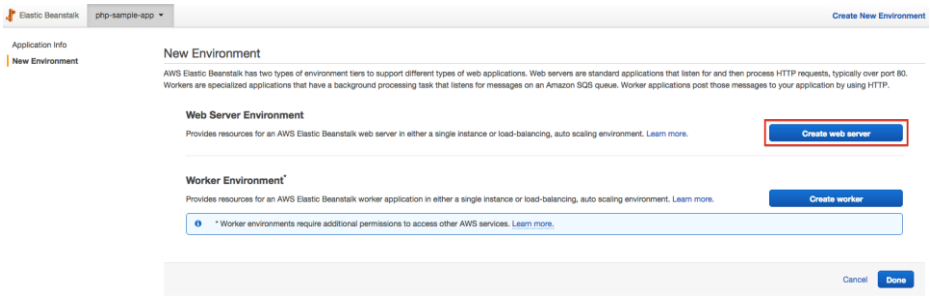


Step 2: Configure your Application

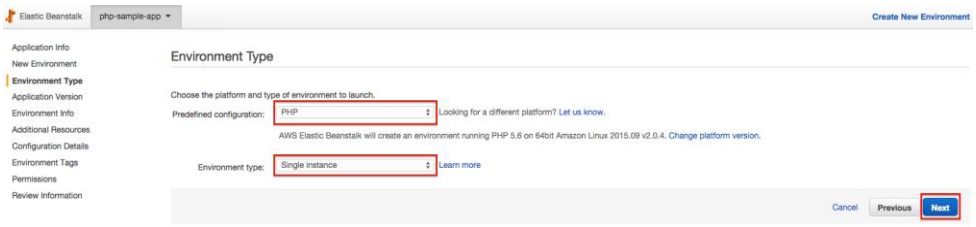
- Fill out the Application name with "Your-sample-app" and Description field with "Sample App". Click Next to continue.

Step 3: Configure your Environment

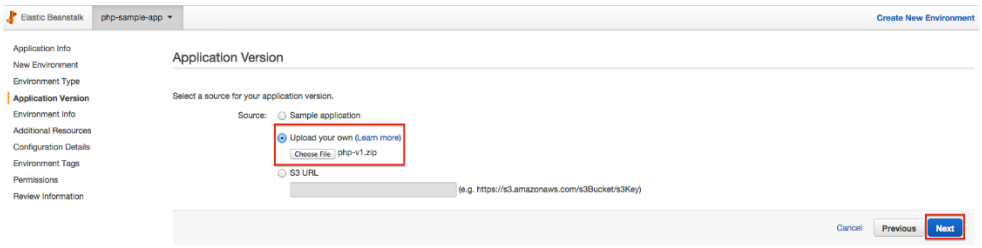
- For this tutorial, we will be creating a web server environment for our sample PHP application. Click on Create web server.



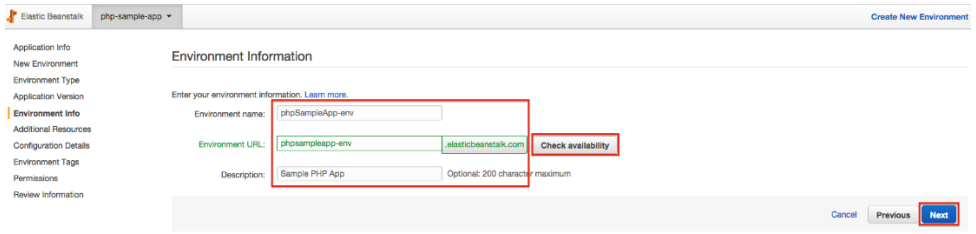
- Click on Select a platform next to Predefined configuration, then select "Your Plateform". Next, click on the drop-down menu next to Environment type, then select Single instance.



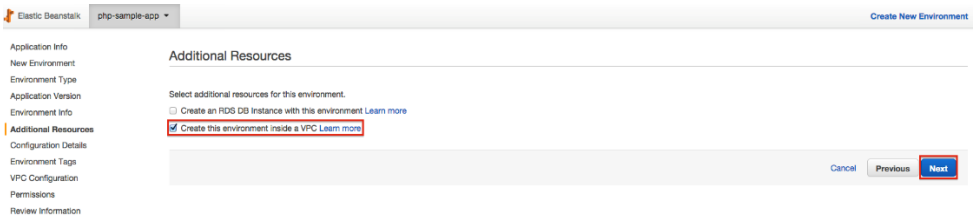
- Under Source, select the Upload your own option, then click Choose File to select the "Your-sample-app-v1.zip" file we downloaded earlier.



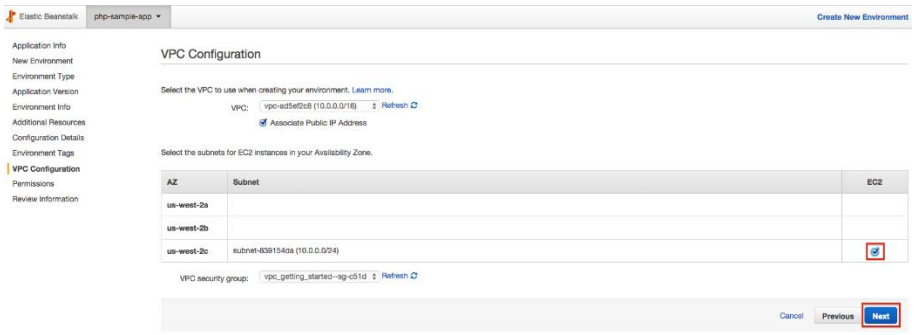
- Fill in the values for Environment name with "YourSampleApp-env". For Environment URL, fill in a globally unique value since this will be your public-facing URL; we will use "YourSampleApp-env" in this tutorial, so please choose something different from this one. Lastly, fill Description with "Your Sample App". For the Environment URL, make sure to click Check availability to make sure that the URL is not taken. Click Next to continue.



- Check the box next to Create this environment inside a VPC. Click Next to continue.



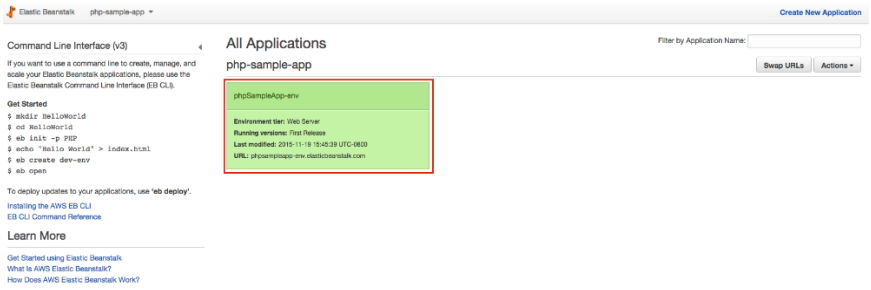
- On the Configuration Details step, you can set configuration options for the instances in your stack. Click Next.
- On the Environment Tags step, you can tag all the resources in your stack. Click Next.
- On the VPC Configuration step, select the first AZ listed by checking the box under the EC2 column. Your list of AZs may look different than the one shown as Regions can have different number of AZs. Click Next.



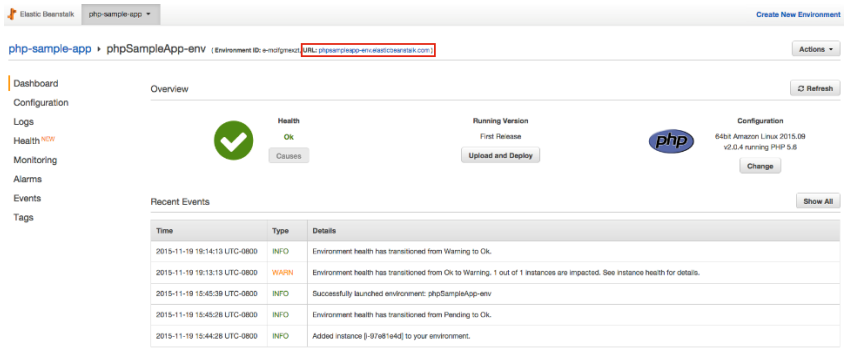
- At the Permissions step, leave everything to their default values, then click Next to continue. Then review your environment configuration on the next screen and then click Launch to deploy your application.

Step 4: Accessing your Elastic Beanstalk Application

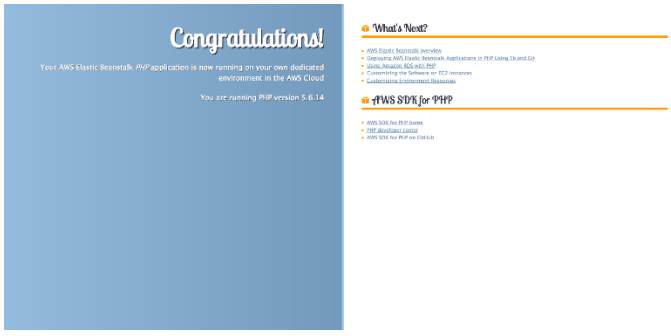
- Go back to the main Elastic Beanstalk dashboard page by clicking on Elastic Beanstalk. When your application successfully launched, your application's environment, "YourSampleApp-env", will show up as a green box. Click on "YourSampleApp-env", which is the green box.



- At the top of the page, you should see a URL field, with a value that contains the Environment URL you specified in step 3. Click on this URL field, and you should see a Congratulations page.



- Congratulations! You have successfully launched a sample PHP application using AWS Elastic Beanstalk.



Virtualization

- Virtualization is changing the mindset from physical to logical.

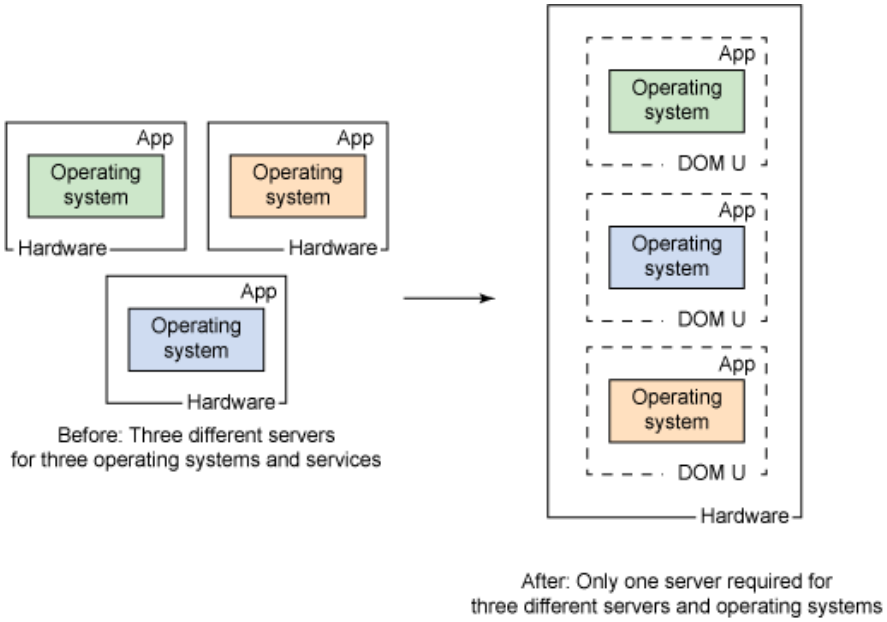


Fig. : Virtualization

- What virtualization means is creating more logical IT resources, called virtual systems, within one physical system. That's called system virtualization.
- It most commonly uses the hypervisor for managing the resources for every virtual system. The hypervisor is a software that can virtualize the hardware resources.

Benefits of Virtualization

- More flexible and efficient allocation of resources.
- Enhance development productivity.
- It lowers the cost of IT infrastructure.
- Remote access and rapid scalability.
- High availability and disaster recovery.
- Pay per use of the IT infrastructure on demand.
- Enables running multiple operating system.

Types of Virtualization

1. Application Virtualization:

- Application virtualization helps a user to have a remote access of an application from a server.
- The server stores all personal information and other characteristics of the application but can still run on a local workstation through internet.
- Example of this would be a user who needs to run two different versions of the same software.
- Technologies that use application virtualization are hosted applications and packaged applications.

2. Network Virtualization:

- The ability to run multiple virtual networks with each has a separate control and data plan.
- It co-exists together on top of one physical network.
- It can be managed by individual parties that potentially confidential to each other.

- Network virtualization provides a facility to create and provision virtual networks—logical switches, routers, firewalls, load balancer, Virtual Private Network (VPN), and workload security within days or even in weeks.

3. Desktop Virtualization:

- Desktop virtualization allows the users' OS to be remotely stored on a server in the data center.
- It allows the user to access their desktop virtually, from any location by different machine.
- Users who wants specific operating systems other than Windows Server will need to have a virtual desktop.
- Main benefits of desktop virtualization are user mobility, portability, and easy management of software installation, updates and patches.

4. Storage Virtualization:

- Storage virtualization is an array of servers that are managed by a virtual storage system.
- The servers aren't aware of exactly where their data is stored, and instead function more like worker bees in a hive.
- It makes managing storage from multiple sources to be managed and utilized as a single repository.
- Storage virtualization software maintains smooth operations, consistent performance and a continuous suite of advanced functions despite changes, break down and differences in the underlying equipment.

Full Virtualization

- Virtual machine simulates hardware to allow an unmodified guest OS to be run in isolation.
- There is two type of Full virtualizations in the enterprise market.
  1. Software assisted full virtualization
  2. Hardware-assisted full virtualization
- On both full virtualization types, guest operating system's source information will not be modified.

1. Software Assisted – Full Virtualization (BT – Binary Translation)

- It completely relies on binary translation to trap and virtualize the execution of sensitive, non-virtualizable instructions sets.
- It emulates the hardware using the software instruction sets.
- Due to binary translation, it often criticized for performance issue.
- Here is the list of software which will fall under software assisted (BT).
  - VMware workstation (32Bit guests)
  - Virtual PC
  - VirtualBox (32-bit guests)
  - VMware Server

2. Hardware-Assisted – Full Virtualization (VT)

- Hardware-assisted full virtualization eliminates the binary translation and it directly interrupts with hardware using the virtualization technology which has been integrated on X86 processors since 2005 (Intel VT-x and AMD-V).
- Guest OS's instructions might allow a virtual context execute privileged instructions directly on the processor, even though it is virtualized.
- Here is the list of enterprise software which supports hardware-assisted – Full virtualization which falls under hypervisor type 1 (Bare metal)

- VMware ESXi /ESX
- KVM
- Hyper-V
- Xen

- The following virtualization type of virtualization falls under hypervisor type 2 (Hosted).
  - VMware Workstation (64-bit guests only)
  - Virtual Box (64-bit guests only)
  - VMware Server (Retired)

Paravirtualization

- Paravirtualization works differently from the full virtualization.
- It doesn't need to simulate the hardware for the virtual machines.
- The hypervisor is installed on a physical server (host) and a guest OS is installed into the environment.
- Virtual guests aware that it has been virtualized, unlike the full virtualization (where the guest doesn't know that it has been virtualized) to take advantage of the functions.
- In this virtualization method, guest source codes will be modified with sensitive information to communicate with the host.
- Guest Operating systems require extensions to make API calls to the hypervisor.
- In full virtualization, guests will issue a hardware calls but in paravirtualization, guests will directly communicate with the host (hypervisor) using the drivers.
- Here is the lisf of products which supports paravirtualization.
  - Xen
  - IBM LPAR
  - Oracle VM for SPARC (LDOM)
  - Oracle VM for X86 (OVM)

Hybrid Virtualization (Hardware Virtualized with PV Drivers)

- In Hardware assisted full virtualization, Guest operating systems are unmodified and it involves many VM traps and thus high CPU overheads which limit the scalability.
- Paravirtualization is a complex method where guest kernel needs to be modified to inject the API.
- By considering these issues, engineers have come with hybrid paravirtualization.
- It's a combination of both Full & Paravirtualization. The virtual machine uses paravirtualization for specific hardware drivers (where there is a bottleneck with full virtualization, especially with I/O & memory intense workloads), and the host uses full virtualization for other features.
- The following products support hybrid virtualization.
  - Oracle VM for x86
  - Xen
  - VMware ESXi

OS level Virtualization

- Operating system-level virtualization is widely used.
- It also known as "containerization".
- Host Operating system kernel allows multiple user spaces also known as instance.
- In OS-level virtualization, unlike other virtualization technologies, there will be very little or no overhead since its uses the host operating system kernel for execution.

- Oracle Solaris zone is one of the famous containers in the enterprise market.
- Here is the list of other containers.
  - Linux LCX
  - Docker
  - AIX WPAR

Virtual computing

- Virtual computing refers to the use of a remote computer from a local computer where the actual computer user is located.
- For example, a user at a home computer could log in to a remote office computer (via the Internet or a network) to perform job tasks.
- Once logged in via special software, the remote computer can be used as though it were at the user's location, allowing the user to perform tasks via the keyboard, mouse, or other tools.

Virtual Machine

- A virtual machine (VM) is an operating system (OS) or application environment that is installed on software, which reproduces dedicated hardware. The end user has the same experience on a virtual machine as they would have on dedicated hardware.

Virtual Machine Conversions in VMM (Virtual Machine Migration)

- When you use cloud computing, you are accessing pooled resources using a technique called virtualization.
- Virtualization assigns a logical name for a physical resource and then provides a pointer to that physical resource when a request is made.
- Virtualization provides a means to manage resources efficiently because the mapping of virtual resources to physical resources can be both dynamic and facile.
- Virtualization is dynamic in that the mapping can be assigned based on rapidly changing conditions, and it is facile because changes to a mapping assignment can be nearly instantaneous.
- These are among the different types of virtualization that are characteristic of cloud computing:
  - **Access**: A client can request access to a cloud service from any location.
  - **Application**: A cloud has multiple application instances and directs requests to an instance based on conditions.
  - **CPU**: Computers can be partitioned into a set of virtual machines with each machine being assigned a workload. Alternatively, systems can be virtualized through load-balancing technologies.
  - **Storage**: Data is stored across storage devices and often replicated for redundancy. To enable these characteristics, resources must be highly configurable and flexible.
- You can define the features in software and hardware that enable this flexibility as conforming to one or more of the following mobility patterns:
  - P2V: Physical to Virtual
  - V2V: Virtual to Virtual
  - V2P: Virtual to Physical
  - P2P: Physical to Physical
  - D2C: Datacenter to Cloud
  - C2C: Cloud to Cloud
  - C2D: Cloud to Datacenter
  - D2D: Datacenter to Datacenter

Virtual Machine Types

1. General Purpose

- This family includes the M1 and M3 VM types.
- These types provide a balance of CPU, memory, and network resources, which makes them a good choice for many applications.
- The VM types in this family range in size from one virtual CPU with two GB of RAM to eight virtual CPUs with 30 GB of RAM.
- The balance of resources makes them ideal for running small and mid-size databases, more memory-hungry data processing tasks, caching fleets, and backend servers.
- M1 types offer smaller instance sizes with moderate CPU performance.
- M3 types offer larger number of virtual CPUs that provide higher performance.
- It is recommended to use M3 instances if you need general-purpose instances with demanding CPU requirements.

2. Compute Optimized

- This family includes the C1 and CC2 instance types, and is geared towards applications that benefit from high compute power.
- Compute-optimized VM types have a higher ratio of virtual CPUs to memory than other families but share the NCs (Node Controllers) with non-optimized ones.
- It is recommended to use these type if you are running any CPU-bound scale-out applications.
- CC2 instances provide high core count (32 virtual CPUs) and support for cluster networking.
- C1 instances are available in smaller sizes and are ideal for scaled-out applications at massive scale.

3. Memory Optimized

- This family includes the CR1 and M2 VM types and is designed for memory-intensive applications.
- It is recommended to use these VM types for performance-sensitive database, where your application is memory-bound.
- CR1 VM types provide more memory and faster CPU than do M2 types.
- CR1 instances also support cluster networking for bandwidth intensive applications.
- M2 types are available in smaller sizes, and are an excellent option for many memory-bound applications.

4. Micro

- This Micro family contains the T1 VM type.
- The T1 micro provides a small amount of consistent CPU resources and allows you to increase CPU capacity in short bursts when additional cycles are available.
- It is recommended to use this type of VM for lower throughput applications like a proxy server or administrative applications, or for low-traffic websites that occasionally require additional compute cycles. It is not recommended for applications that require sustained CPU performance.

Load Balancing

- In computing, load balancing improves the distribution of workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units, or disk drives.

Need of load balancing in cloud computing

(i) High Performing applications

- Cloud load balancing techniques, unlike their traditional on premise counterparts, are less expensive and simple to implement. Enterprises can make their client applications work faster and deliver better performances, that too at potentially lower costs.

(ii) Increased scalability

- Cloud balancing takes help of cloud's scalability and agility to maintain website traffic. By using efficient load balancers, you can easily match up the increased user traffic and distribute it among various servers or network devices. It is especially important for ecommerce websites, who deals with thousands of website visitors every second. During sale or other promotional offers they need such effective load balancers to distribute workloads.

(iii) Ability to handle sudden traffic spikes

- A normally running University site can completely go down during any result declaration. This is because too many requests can arrive at the same time. If they are using cloud load balancers, they do not need to worry about such traffic surges. No matter how large the request is, it can be wisely distributed among different servers for generating maximum results in less response time.

(iv) Business continuity with complete flexibility

- The basic objective of using a load balancer is to save or protect a website from sudden outages. When the workload is distributed among various servers or network units, even if one node fails the burden can be shifted to another active node.
- Thus, with increased redundancy, scalability and other features load balancing easily handles website or application traffic.

Network resources that can be load balanced

- Servers
- Routing mechanism

Hypervisors

- It is the part of the private cloud that manages the virtual machines, i.e. it is the part (program) that enables multiple operating systems to share the same hardware.
- Each operating system could use all the hardware (processor, memory, etc.) if no other operating system is on. That is the maximum hardware available to one operating system in the cloud.
- Nevertheless, the hypervisor is what controls and allocates what portion of hardware resources each operating system should get, in order every one of them to get what they need and not to disrupt each other.

There are two types of hypervisors

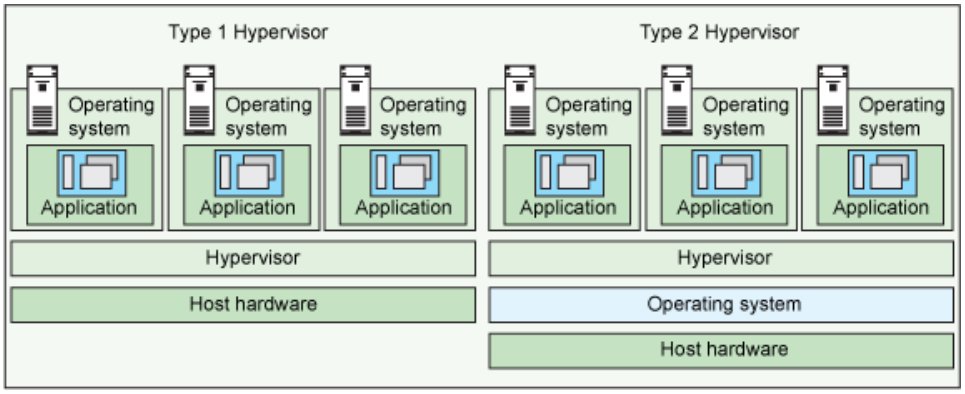


Fig. : Types of Hypervisors

- **Type 1 hypervisor**: hypervisors run directly on the system hardware – A "bare metal" embedded hypervisor. Examples are:
  1) VMware ESX and ESXi
  2) Microsoft Hyper-V
  3) Citrix XenServer
  4) Oracle VM
- **Type 2 hypervisor**: hypervisors run on a host operating system that provides virtualization services, such as I/O device support and memory management. Examples are:
  1) VMware Workstation/Fusion/Player
  2) Microsoft Virtual PC
  3) Oracle VM VirtualBox
  4) Red Hat Enterprise Virtualization

Machine Imaging

- Machine imaging is a process that is used to achieve the goal of system portability, provision, and deploy systems in the cloud through capturing the state of systems using a system image.
- A system image makes a copy or a clone of the entire computer system inside a single file.
- The image is made by using a program called system imaging program and can be used later to restore a system image.
- For example Amazon Machine Image (AMI) is a system image that is used in the cloud computing.
- The Amazon Web Services uses AMI to store copies of a virtual machine.
- An AMI is a file system image that contains an operating system, all device drivers, and any applications and state information that the working virtual machine would have.
- The AMI files are encrypted and compressed for security purpose and stored in Amazon S3 (Simple Storage System) buckets as a set of 10MB chunks.
- Machine imaging is mostly run on virtualization platform due to this it is also called as Virtual Appliances and running virtual machines are called instances.
- Because many users share clouds, the cloud helps you track information about images, such as ownership, history, and so on.
- The IBM SmartCloud Enterprise knows what organization you belong to when you log in.
- You can choose whether to keep images private, exclusively for your own use, or to share with other users in your organization.
- If you are an independent software vendor, you can also add your images to the public catalog.

Cloud Marketplace Overview

- A cloud marketplace is an online storefront operated by a cloud service provider.
- A cloud marketplace provides customers with access to software applications and services that are built on, integrate with or complement the cloud provider's offerings.
- A marketplace typically provides customers with native cloud applications and approved apps created by third-party developers.
- Applications from third-party developers not only help the cloud provider fill niche gaps in its portfolio and meet the needs of more customers, but they also provide the customer with peace of mind by knowing that all purchases from the vendor's marketplace will integrate with each other smoothly.

Examples of cloud marketplaces

- AWS Marketplace - helps customers find, buy and use software and services that run in the Amazon Elastic Compute Cloud (EC2).
- Oracle Marketplace - offers a comprehensive list of apps for sales, service, marketing, talent management and human capital management.
- Microsoft Windows Azure Marketplace - an online market for buying and selling Software as a Service (SaaS) applications and research datasets.
- Salesforce.com's AppExchange - provides business apps for sales representatives and customer relationship management (CRM).

Comparison of Cloud Providers

	Amazon Web Service	**Azure**	**Rackspace**
Introduction	Amazon Web Services (AWS) is a collection of remote computing services (also called web services) that together make up a cloud computing platform, offered over the Internet by Amazon.com.	Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed datacenters.	Rackspace is a managed cloud computing provider offering high percentage availability of applications based on RAID10.
Distinguishing Features	Rich set of services and integrated monitoring tools; competitive pricing model.	Easy-to-use administration tool, especially for Windows admins.	Easy to use control panel, especially for non-system administrators.
Virtualization	Xen hypervisor	Microsoft Hyper-V	Opensource (Xen, Kvm) and VMware
Base OS	Linux (+QEMU) and Windows	Windows and Linux	Ubuntu
Pricing model	Pay-as-you-go, then subscription	Pay-as-you-go	Pay-as-you-go
Major products	Elastic block store, IP addresses, virtual private cloud, cloud watch, Cloud Front, clusters etc.	Server Failover Clustering, Network Load Balancing, SNMP Services, Storage Manager for SANs, Windows Internet Name Service, Disaster Recovery to Azure, Azure Caching and Azure Redis Cache.	Managed cloud, block storage, monitoring

	Amazon Web Service	**Azure**	**Rackspace**
CDN Features	Origin-Pull, Purge, Gzip compression, Persistent connections, Caching headers, Custom CNAMEs, Control Panel & stats, Access Logs.	Robust security, Lower latencies, Massively scalable, Capacity on demand.	Rackspace provide CDN services through a partnership with Akamai's service.
Access interface	Web-based, API, console	Web interface	Web-based control panel
Preventive measures	Moderate	Basic	Basic
Reactive measures	Moderate	Basic	Basic
Reliability	Good	Average	Good
Scalability	Good	Good	Good
Support	Good and chargeable	Good	Excellent
Availability (%)	99.95	99.95	99.99
Server Performance (Over a period)	Good	Excellent and consistent	Average
Tools/ framework	Amazon machine image (AMI), Java, PHP, Python, Ruby	PHP, ASP.NET, Node.js, Python	-
Database RDS	MySQL, MsSQL, Oracle	Microsoft SQL Database	MySQL

AWS History

- The AWS platform was launched in July 2002.
- In its early stages, the platform consisted of only a few disparate tools and services.
- Then in late 2003, the AWS concept was publicly reformulated when Chris Pinkham and Benjamin Black presented a paper describing a vision for Amazon's retail computing infrastructure that was completely standardized, completely automated, and would rely extensively on web services for services such as storage and would draw on internal work already underway.
- Near the end of their paper, they mentioned the possibility of selling access to virtual servers as a service, proposing the company could generate revenue from the new infrastructure investment.
- In November 2004, the first AWS service launched for public usage: Simple Queue Service (SQS).
- Thereafter Pinkham and lead developer Christopher Brown developed the Amazon EC2 service, with a team in Cape Town, South Africa.
- Amazon Web Services was officially re-launched on March 14, 2006, combining the three initial service offerings of Amazon S3 cloud storage, SQS, and EC2.
- The AWS platform finally provided an integrated suite of core online services, as Chris Pinkham and Benjamin Black had proposed back in 2003, as a service offered to other developers, web sites, client-side applications, and companies.
- Andy Jassy, AWS founder and vice president in 2006, said at the time that Amazon S3 (one of the first and most scalable elements of AWS) helps free developers from worrying about where they are going to store data, whether it will be safe and secure, if it will be available when they need it, the costs associated with server maintenance, or whether they have enough storage available.
- Amazon S3 enables developers to focus on innovating with data, rather than figuring out how to store it.
- In 2016 Jassy was promoted to CEO of the division.
- Reflecting the success of AWS, his annual compensation in 2017 hit nearly $36 million.
- In 2014, AWS launched its partner network entitled APN (AWS Partner Network) which is focused on helping AWS-based companies grow and scale the success of their business with close collaboration and best practices.
- To support industry-wide training and skills standardization, AWS began offering a certification program for computer engineers, on April 30, 2013, to highlight expertise in cloud computing.
- In January 2015, Amazon Web Services acquired Annapurna Labs, an Israel-based microelectronics company reputedly for US$350–370M.
- James Hamilton, an AWS engineer, wrote a retrospective article in 2016 to highlight the ten-year history of the online service from 2006 to 2016. As an early fan and outspoken proponent of the technology, he had joined the AWS engineering team in 2008.
- In January 2018, Amazon launched an auto scaling service on AWS.
- In November 2018, AWS announced customized ARM cores for use in its servers.
- Also in November 2018, AWS is developing ground stations to communicate with customer's satellites.

AWS Infrastructure

- Amazon Web Services (AWS) is a global public cloud provider, and as such, it has to have a global network of infrastructure to run and manage its many growing cloud services that support customers around the world.
- Now we'll take a look at the components that make up the AWS Global Infrastructure.
  1) Availability Zones (AZs)

2) Regions
3) Edge Locations
4) Regional Edge Caches

- If you are deploying services on AWS, you'll want to have a clear understanding of each of these components, how they are linked, and how you can use them within your solution to YOUR maximum benefit. Let's take a closer look.

1) Availability Zones (AZ)

- AZs are essentially the physical data centers of AWS. This is where the actual compute, storage, network, and database resources are hosted that we as consumers provision within our Virtual Private Clouds (VPCs).
- A common misconception is that a single availability zone is equal to a single data center. This is not the case. In fact, it's likely that multiple data centers located close together form a single availability zone.
- Each AZ will always have at least one other AZ that is geographically located within the same area, usually a city, linked by highly resilient and very low latency private fiber optic connections. However, each AZ will be isolated from the others using separate power and network connectivity that minimizes impact to other AZs should a single AZ fail.
- These low latency links between AZs are used by many AWS services to replicate data for high availability and resilience purposes.
- Multiple AZs within a region allows you to create highly available and resilient applications and services.
- By architecting your solutions to utilize resources across more than one AZ ensures that minimal or no impact will occur to your infrastructure should an AZ experience a failure, which does happen.
- Anyone can deploy resources in the cloud, but architecting them in a way that ensures your infrastructure remains stable, available, and resilient when faced with a disaster is a different matter.
- Making use of at least two AZs in a region helps you maintain high availability of your infrastructure and it's always a recommended best practice.

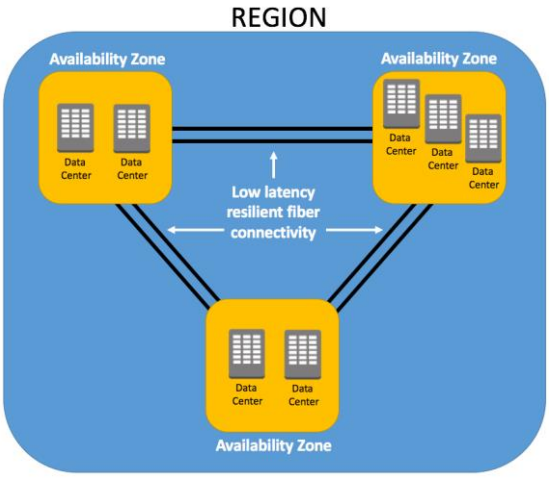


Fig. : Availability Zone and Region

2) Regions

- Region is a collection of availability zones that are geographically located close to one other.
- This is generally indicated by AZs within the same city. AWS has deployed them across the globe to allow its worldwide customer base to take advantage of low latency connections.
- Each Region will act independently of the others, and each will contain at least two Availability Zones.
- Example: if an organization based in London was serving customers throughout Europe, there would be no logical sense to deploy services in the Sydney Region simply due to the latency response times for its

customers. Instead, the company would select the region most appropriate for them and their customer base, which may be the London, Frankfurt, or Ireland Region.

- Having global regions also allows for compliance with regulations, laws, and governance relating to data storage (at rest and in transit).
- Example: you may be required to keep all data within a specific location, such as Europe. Having multiple regions within this location allows an organization to meet this requirement.
- Similarly to how utilizing multiple AZs within a region creates a level of high availability, the same can be applied to utilizing multiple regions.
- You may want to use multiple regions if you are a global organization serving customers in different countries that have specific laws and governance about the use of data.
- In this case, you could even connect different VPCs together in different regions.
- The number of regions is increasing year after year as AWS works to keep up with the demand for cloud computing services.
- In July 2017, there are currently 16 Regions and 43 Availability Zones, with 4 Regions and 11 AZs planned.

3) Edge Locations

- Edge Locations are AWS sites deployed in major cities and highly populated areas across the globe. They far outnumber the number of availability zones available.
- While Edge Locations are not used to deploy your main infrastructures such as EC2 instances, EBS storage, VPCs, or RDS resources like AZs, they are used by AWS services such as AWS CloudFront and AWS Lambda@Edge (currently in Preview) to cache data and reduce latency for end user access by using the Edge Locations as a global Content Delivery Network (CDN).
- As a result, Edge Locations are primarily used by end users who are accessing and using your services.
- For example, you may have your website hosted on EC2 instances and S3 (your origin) within the Ohio region with a configured CloudFront distribution associated. When a user accesses your website from Europe, they would be re-directed to their closest Edge Location (in Europe) where cached data could be read on your website, significantly reducing latency.

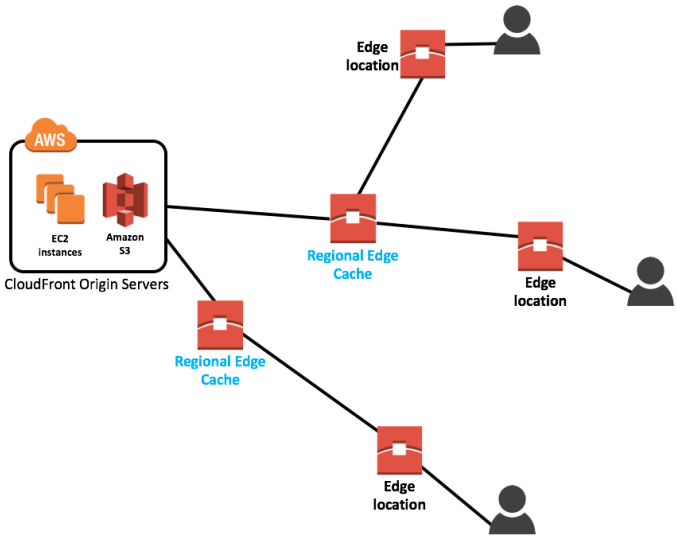


Fig. : Edge Location and Regional Edge Cache

4) Regional Edge Cache

- In November 2016, AWS announced a new type of Edge Location, called a Regional Edge Cache.
- These sit between your CloudFront Origin servers and the Edge Locations.

- A Regional Edge Cache has a larger cache-width than each of the individual Edge Locations, and because data expires from the cache at the Edge Locations, the data is retained at the Regional Edge Caches.
- Therefore, when data is requested at the Edge Location that is no longer available, the Edge Location can retrieve the cached data from the Regional Edge Cache instead of the Origin servers, which would have a higher latency.

Pods, Aggregation, Silos

- Workloads support a certain no. of user when the workload reaches the limit of largest virtual machine instance possible, a copy or clone of the instance is required. A group of users within a particular instance is called a pod.
- Sizing limitation of pod need to be considered when building large cloud-based application. Pods are aggregated into pools within IaaS region or site called an availability zone.
- When the computing infrastructure isolates user clouds from one another, so that interoperating is impossible this creates an information silo, or simply silo.

AWS Services

- This AWS services list covers the huge catalog of services offered by Amazon Web Services (AWS). These services range from the core compute products like EC2 to newer releases like AWS Deepracer for machine learning.
- There are currently 190 unique services provided by AWS which divided into 24 categories which are listed below:
  - Analytics
  - Application Integration
  - AR & VR
  - AWS Cost Management
  - Blockchain
  - Business Applications
  - Compute
  - Customer Engagement
  - Database
  - Developer Tools
  - End User Computing
  - Game Tech
  - Internet of Things
  - Machine Learning
  - Management & Governance
  - Media Services
  - Migration & Transfer
  - Mobile
  - Networking & Content Delivery
  - Robotics
  - Satellite
  - Security, Identity, & Compliance
  - Storage
  - Quantum Technologies

AWS Ecosystem

- In general a cloud ecosystem is a complex system of interdependent components that all work together to enable cloud services. In cloud computing, the ecosystem consists of hardware and software as well as cloud customers, cloud engineers, consultants, integrators and partners.
- Amazon Web Services (AWS) is the market leader in IaaS (Infrastructure-as-a-Service) and PaaS (Platform-as-a-Service) for cloud ecosystems, which can be combined to create a scalable cloud application without worrying about delays related to infrastructure provisioning (compute, storage, and network) and management.
- With AWS you can select the specific solutions you need, and only pay for exactly what you use, resulting in lower capital expenditure and faster time to value without sacrificing application performance or user experience.
- New and existing companies can build their digital infrastructure partially or entirely in the cloud with AWS, making the on premise data center a thing of the past.
- The AWS cloud ensures infrastructure reliability, compliance with security standards, and the ability to instantly grow or shrink your infrastructure to meet your needs and maximize your budget, all without upfront investment in equipment.

Basic Understanding APIs

- Amazon API Gateway is an AWS service for creating, publishing, maintaining, monitoring, and securing REST, HTTP, and WebSocket APIs at any scale. API developers can create APIs that access AWS or other web services, as well as data stored in the AWS Cloud. As an API Gateway API developer, you can create APIs for use in your own client applications. Or you can make your APIs available to third-party app developers. For more information, see Who uses API Gateway?
- API Gateway creates RESTful APIs that:
  - Are HTTP-based.
  - Enable stateless client-server communication.
  - Implement standard HTTP methods such as GET, POST, PUT, PATCH, and DELETE.

AWS Programming Interfaces

API Gateway

- API Gateway is an AWS service that supports the following:
  - Creating, deploying, and managing a RESTful application programming interface (API) to expose backend HTTP endpoints, AWS Lambda functions, or other AWS services.
  - Creating, deploying, and managing a WebSocket API to expose AWS Lambda functions or other AWS services.
  - Invoking exposed API methods through the frontend HTTP and WebSocket endpoints.

API Gateway REST API

- A collection of HTTP resources and methods that are integrated with backend HTTP endpoints, Lambda functions, or other AWS services. You can deploy this collection in one or more stages. Typically, API resources are organized in a resource tree according to the application logic. Each API resource can expose one or more API methods that have unique HTTP verbs supported by API Gateway.

API Gateway HTTP API

- A collection of routes and methods that are integrated with backend HTTP endpoints or Lambda functions. You can deploy this collection in one or more stages. Each route can expose one or more API methods that have unique HTTP verbs supported by API Gateway.

API Gateway WebSocket API

- A collection of WebSocket routes and route keys that are integrated with backend HTTP endpoints, Lambda functions, or other AWS services. You can deploy this collection in one or more stages. API methods are invoked through frontend WebSocket connections that you can associate with a registered custom domain name.

API Deployment

- A point-in-time snapshot of your API Gateway API. To be available for clients to use, the deployment must be associated with one or more API stages.

API Developer

- Your AWS account that owns an API Gateway deployment (for example, a service provider that also supports programmatic access).

API Endpoint

- A hostname for an API in API Gateway that is deployed to a specific Region. The hostname is of the form {api-id}.execute-api.{region}.amazonaws.com. The following types of API endpoints are supported:
  - Edge-optimized API endpoint
    - The default hostname of an API Gateway API that is deployed to the specified Region while using a CloudFront distribution to facilitate client access typically from across AWS Regions. API requests are routed to the nearest CloudFront Point of Presence (POP), which typically improves connection time for geographically diverse clients.
  - Private API endpoint
    - An API endpoint that is exposed through interface VPC endpoints and allows a client to securely access private API resources inside a VPC. Private APIs are isolated from the public internet, and they can only be accessed using VPC endpoints for API Gateway that have been granted access.
  - Regional API endpoint
    - The host name of an API that is deployed to the specified Region and intended to serve clients, such as EC2 instances, in the same AWS Region. API requests are targeted directly to the Region-specific API Gateway API without going through any CloudFront distribution. For in-Region requests, a Regional endpoint bypasses the unnecessary round trip to a CloudFront distribution.

API Key

- An alphanumeric string that API Gateway uses to identify an app developer who uses your REST or WebSocket API. API Gateway can generate API keys on your behalf, or you can import them from a CSV file. You can use API keys together with Lambda authorizers or usage plans to control access to your APIs.

WebSocket Connection

- API Gateway maintains a persistent connection between clients and API Gateway itself. There is no persistent connection between API Gateway and backend integrations such as Lambda functions. Backend services are invoked as needed, based on the content of messages received from clients.

Web Services

- You can choose from a couple of different schools of thought for how web services should be delivered.
- The older approach, **SOAP** (short for Simple Object Access Protocol), had widespread industry support, complete with a comprehensive set of standards.
- Those standards were too comprehensive, unfortunately. The people designing SOAP set it up to be extremely flexible —it can communicate across the web, e-mail, and private networks.
- To ensure security and manageability, a number of supporting standards that integrate with SOAP were also defined.
- SOAP is based on a document encoding standard known as Extensible Markup Language (XML, for short), and the SOAP service is defined in such a way that users can then leverage XML no matter what the underlying communication network is.
- For this system to work, though, the data transferred by SOAP (commonly referred to as the payload) also needs to be in XML format.
- Notice a pattern here? The push to be comprehensive and flexible (or, to be all things to all people) plus the XML payload requirement meant that SOAP ended up being quite complex, making it a lot of work to use properly.
- As you might guess, many IT people found SOAP daunting and, consequently, resisted using it.

- About a decade ago, a doctoral student defined another web services approach as part of his thesis: **REST**, or Representational State Transfer.
- REST, which is far less comprehensive than SOAP, aspires to solve fewer problems.
- It doesn't address some aspects of SOAP that seemed important but that, in retrospect, made it more complex to use — security, for example.
- The most important aspect of REST is that it's designed to integrate with standard web protocols so that REST services can be called with standard web verbs and URLs.
- For example, a valid REST call looks like this: http://search.examplecompany.com/CompanyDirectory/EmployeeInfo?empname=BernardGolden
- That's all it takes to make a query to the REST service of examplecompany to see my personnel information.
- The HTTP verb that accompanies this request is GET, asking for information to be returned.
- To delete information, you use the verb DELETE.
- To insert my information, you use the verb POST.
- To update my information, you use the verb PUT.
- For the POST and PUT actions, additional information would accompany the empname and be separated by an ampersand (&) to indicate another argument to be used by the service.
- REST imposes no particular formatting requirements on the service payloads. In this respect, it differs from SOAP, which requires XML.
- For simple interactions, a string of bytes is all you need for the payload. For more complex interactions (say, in addition to returning my employee information, I want to place a request for the employee information of all employees whose names start with G), the encoding convention JSON is used.
- As you might expect, REST's simpler use model, its alignment with standard web protocols and verbs, and its less restrictive payload formatting made it catch on with developers like a house on fire.
- AWS originally launched with SOAP support for interactions with its API, but it has steadily deprecated its SOAP interface in favor of REST.

AWS URL Naming

- You can access your bucket using the Amazon S3 console. Using the console UI, you can perform almost all bucket operations without having to write any code.
- If you access a bucket programmatically, note that Amazon S3 supports RESTful architecture in which your buckets and objects are resources, each with a resource URI that uniquely identifies the resource.
- Amazon S3 supports both virtual-hosted–style and path-style URLs to access a bucket.
- In a **virtual-hosted–style URL**, the bucket name is part of the domain name in the URL. For example:
  - http://bucket.s3.amazonaws.com
  - http://bucket.s3-aws-region.amazonaws.com
- In a virtual-hosted–style URL, you can use either of these endpoints.
- If you make a request to the http://bucket.s3.amazonaws.com endpoint, the DNS has sufficient information to route your request directly to the Region where your bucket resides.
- In a **path-style URL**, the bucket name is not part of the domain (unless you use a Region-specific endpoint). For example:
  - US East (N. Virginia) Region endpoint, http://s3.amazonaws.com/bucket
  - Region-specific endpoint, http://s3-aws-region.amazonaws.com/bucket
- In a path-style URL, the endpoint you use must match the Region in which the bucket resides.

- For example, if your bucket is in the South America (São Paulo) Region, you must use the http://s3-sa-east-1.amazonaws.com/bucket endpoint. If your bucket is in the US East (N. Virginia) Region, you must use the http://s3.amazonaws.com/bucket endpoint.

Matching Interfaces and Services

- In the simplest case, the service interfaces are identical apart from name and category (inbound or outbound), that is, if the outbound service interface and all interface objects, are referenced by this service interface, are copies of the corresponding objects of an inbound service interfaces.
- If, however, the consumer only wants to call one operation of the inbound service interface, for example, it is not necessary to create the other inbound service interface operations in the outbound service interface as well.
- Simply put, the operations and corresponding outbound service interface data structures can be a subset of the operations and corresponding data structures of the inbound interface referenced.
- The service interface editor provides a check for the service interface pairs to determine the compatibility of the inbound and outbound service interface.
- This check is performed in multiple steps to determine compatibility (starting service interfaces, across operations and down to data types).
- The following section describes the steps to be able to estimate for a service interface assignment whether two service interfaces match.

Matching Service Interfaces

- Two service interfaces match each other if the following conditions are fulfilled:
  - One service interface is of the outbound category and the other service interface if of the inbound category. Neither of the service interfaces can be abstract.
  - Both of the service interfaces have the same interface pattern.
  - There is a matching operation in the inbound service interface for each of the operations in the outbound service interface.

Matching Operations

- An inbound service interface operation matches an outbound service interface operation (and the other way around) if the following conditions are met:
  - Both operations must have the name mode (asynchronous or synchronous).
  - Both operations must have the same Operation Pattern.
  - The message type for the request, which must be referenced by each operation, must have the same name and same XML Namespace. The names of the operations may differ. The same applies for the response with synchronous communication.
  - If the inbound service interface operation references a fault message type, the outbound service interface operation must also reference a fault message type with the same name and XML Namespace.
  - The data types of the message types, which the outbound service interface for the request message references (and, if necessary, for the response and fault message) must be compatible with the corresponding inbound service interface data types.

Matching Data Types

- The check whether the corresponding data types are compatible with each other is sufficient until the comparison of the Facets of an XSD type.

- The data types are compared using the same method as other objects: The structures are compatible if they contain the same fields (elements and attributes) and if these fields have compatible types, frequencies, details, and default values.
- There are however a few restraints, for example the target structure can contain attributes or elements that do not appear in the outbound structure, but if these are not required and where the frequency is optional or prohibited (attributes) or minOccurs=0 (elements).
  - The data structures compared must both be correct. For example, not all correct facets are skipped or considered in the compatibility check.
  - Some XSD schema language elements that can appear in a reference to an external message in the data structure are not supported. Therefore, the elements redefine and any, for example, as well as the attributes blockDefault, finalDefault, and substitutionGroup.
  - The comparison of structures is, for example, restricted to the following:
    - The details white Space and pattern are not checked
    - If the facet pattern is used for the outbound structure field, all the other details are not checked.
    - If the order of sub elements if different between the outbound and target field, a warning is displayed.

Elastic Block Store

- Amazon Elastic Block Store is an AWS block storage system that is best used for storing persistent data.
- Often incorrectly referred to as Elastic Block Storage, Amazon EBS provides highly available block level storage volumes for use with Amazon Elastic Compute Cloud (EC2) instances.
- An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure.
- To begin, create an EBS volume (General Purpose, Provisioned IOPS or Magnetic), pick a size for it (up to a terabyte of data) and attach that to any one of your EC2 instances.
- An EBS volume can only be attached to one instance at a time but if you need to have multiple copies of the volume, you can take a snapshot and create another volume from that snapshot and attach it to another drive.
- A snapshot file is equivalent to a backup of whatever the EBS volume looks like at the time. For every snapshot you create, you can make an identical EC2 instance. This will allow you to publish identical content on multiple servers.
- Amazon EBS is ideal if you're doing any substantial work with EC2, you want to keep data persistently on a file system, and you want to keep that data around even after you shut down your EC2 instance.
- EC2 instances have local storage that you can use as long as you're running the instance, but as soon as you shut down the instance you lose the data that was on there.
- If you want to save anything, you need to save it on Amazon EBS. Because EC2 is like having a local drive on the machine, you can access and read the EBS volumes anytime once you attach the file to an EC2 instance.

Amazon Simple Storage Service (S3)

- Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.
- Amazon S3 is intentionally built with a minimal feature set that focuses on simplicity and robustness.
- Following are some of advantages of the Amazon S3 service:

- **Create Buckets** – Create and name a bucket that stores data. Buckets are the fundamental container in Amazon S3 for data storage.
- **Store data in Buckets** – Store an infinite amount of data in a bucket. Upload as many objects as you like into an Amazon S3 bucket. Each object can contain up to 5 TB of data. Each object is stored and retrieved using a unique developer-assigned key.
- **Download data** – Download your data any time you like or allow others to do the same.
- **Permissions** – Grant or deny access to others who want to upload or download data into your Amazon S3 bucket.
- **Standard interfaces** – Use standards-based REST and SOAP interfaces designed to work with any Internet-development toolkit.

Amazon S3 Application Programming Interfaces (API)

- The Amazon S3 architecture is designed to be programming language-neutral, using their supported interfaces to store and retrieve objects.
- Amazon S3 provides a REST and a SOAP interface.
- They are similar, but there are some differences. For example, in the REST interface, metadata is returned in HTTP headers. Because we only support HTTP requests of up to 4 KB (not including the body), the amount of metadata you can supply is restricted.

The REST Interface

- The REST API is an HTTP interface to Amazon S3.
- Using REST, you use standard HTTP requests to create, fetch, and delete buckets and objects.
- You can use any toolkit that supports HTTP to use the REST API.
- You can even use a browser to fetch objects, as long as they are anonymously readable.
- The REST API uses the standard HTTP headers and status codes, so that standard browsers and toolkits work as expected.
- In some areas, they have added functionality to HTTP (for example, we added headers to support access control).

The SOAP Interface

- SOAP support over HTTP is deprecated, but it is still available over HTTPS.
- New Amazon S3 features will not be supported for SOAP.
- The SOAP API provides a SOAP 1.1 interface using document literal encoding.
- The most common way to use SOAP is to download the WSDL, and use a SOAP toolkit such as Apache Axis or Microsoft .NET to create bindings, and then write code that uses the bindings to call Amazon S3.

Operations we can execute through API

- Login into Amazon S3.
- Uploading.
- Retrieving.
- Deleting etc.

Amazon Glacier (Now Amazon S3 Glacier) - Content Delivery Platforms

- Amazon Glacier is an extremely low-cost storage service that provides secure, durable, and flexible storage for data backup and archival.
- With Amazon Glacier, customers can reliably store their data for as little as $0.004 per gigabyte per month.

- Amazon Glacier enables customers to offload the administrative burdens of operating and scaling storage to AWS, so that they don't have to worry about capacity planning, hardware provisioning, data replication, hardware failure detection and repair, or time-consuming hardware migrations.
- Amazon Glacier enables any business or organization to easily and cost effectively retain data for months, years, or decades.
- With Amazon Glacier, customers can now cost effectively retain more of their data for future analysis or reference, and they can focus on their business rather than operating and maintaining their storage infrastructure.
- Customers seeking compliance storage can deploy compliance controls using Vault Lock to meet regulatory and compliance archiving requirements.

Benefits of Glacier Storage Service.

1. Retrievals as Quick as 1-5 Minutes

- Amazon Glacier provides three retrieval options to fit your use case. Expedited retrievals typically return data in 1-5 minutes, and are great for Active Archive use cases. Standard retrievals typically complete between 3-5 hours' work, and work well for less time-sensitive needs like backup data, media editing, or long-term analytics. Bulk retrievals are the lowest-cost retrieval option, returning large amounts of data within 5-12 hours.

2. Unmatched Durability & Scalability

- Amazon Glacier runs on the world's largest global cloud infrastructure, and was designed for 99.999999999% of durability. Data is automatically distributed across a minimum of three physical Availability Zones that are geographically separated within an AWS Region, and Amazon Glacier can also automatically replicate data to any other AWS Region.

3. Most Comprehensive Security & Compliance Capabilities

- Amazon Glacier offers sophisticated integration with AWS CloudTrail to log, monitor and retain storage API call activities for auditing, and supports three different forms of encryption. Amazon Glacier also supports security standards and compliance certifications including SEC Rule 17a-4, PCI-DSS, HIPAA/HITECH, FedRAMP, EU Data Protection Directive, and FISMA, and Amazon Glacier Vault Lock enables WORM storage capabilities, helping satisfy compliance requirements for virtually every regulatory agency around the globe.

4. Low Cost

- Amazon Glacier is designed to be the lowest cost AWS object storage class, allowing you to archive large amounts of data at a very low cost. This makes it feasible to retain all the data you want for use cases like data lakes, analytics, IoT, machine learning, compliance, and media asset archiving. You pay only for what you need, with no minimum commitments or up-front fees.

5. Most Supported Platform with the Largest Ecosystem

- In addition to integration with most AWS services, the Amazon object storage ecosystem includes tens of thousands of consulting, systems integrator and independent software vendor partners, with more joining every month. And the AWS Marketplace offers 35 categories and more than 3,500 software listings from over 1,100 ISVs that are pre-configured to deploy on the AWS Cloud. AWS Partner Network partners have adapted their services and software to work with Amazon S3 and Amazon Glacier for solutions like Backup & Recovery, Archiving, and Disaster Recovery. No other cloud provider has more partners with solutions that are pre-integrated to work with their service.

6. Query in Place

- Amazon Glacier is the only cloud archive storage service that allows you to query data in place and retrieve only the subset of data you need from within an archive. Amazon Glacier Select helps you reduce the total cost of ownership by extending your data lake into cost-effective archive storage.

Identity Management and Access Management (IAM)

- Identity and access management (IAM) is a framework for business processes that facilitates the management of electronic or digital identities.
- The framework includes the organizational policies for managing digital identity as well as the technologies needed to support identity management.
- With IAM technologies, IT managers can control user access to critical information within their organizations.
- Identity and access management products offer role-based access control, which lets system administrators regulate access to systems or networks based on the roles of individual users within the enterprise.
- In this context, access is the ability of an individual user to perform a specific task, such as view, create or modify a file.
- Roles are defined according to job competency, authority and responsibility within the enterprise.
- Systems used for identity and access management include single sign-on systems, multifactor authentication and access management.
- These technologies also provide the ability to securely store identity and profile data as well as data governance functions to ensure that only data that is necessary and relevant is shared.
- These products can be deployed on premises, provided by a third party vendor via a cloud-based subscription model or deployed in a hybrid cloud.

How Does IAM Work?

- The IAM workflow includes the following six elements:
  1. A principal is an entity that can perform actions on an AWS resource. A user, a role or an application can be a principal.
  2. Authentication is the process of confirming the identity of the principal trying to access an AWS product. The principal must provide its credentials or required keys for authentication.
  3. Request: A principal sends a request to AWS specifying the action and which resource should perform it.
  4. Authorization: By default, all resources are denied. IAM authorizes a request only if all parts of the request are allowed by a matching policy. After authenticating and authorizing the request, AWS approves the action.
  5. Actions are used to view, create, edit or delete a resource.
  6. Resources: A set of actions can be performed on a resource related to your AWS account.

Identities (Users, Groups, and Roles)

- IAM identities, which you create to provide authentication for people and processes in your AWS account.
- IAM groups, which are collections of IAM users that you can manage as a unit.
- Identities represent the user, and can be authenticated and then authorized to perform actions in AWS.
- Each of these can be associated with one or more policies to determine what actions a user, role, or member of a group can do with which AWS resources and under what conditions.

The AWS Account Root User

- When you first create an Amazon Web Services (AWS) account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account.

- This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account.

IAM Users

- An IAM user is an entity that you create in AWS.
- The IAM user represents the person or service who uses the IAM user to interact with AWS.
- A primary use for IAM users is to give people the ability to sign in to the AWS Management Console for interactive tasks and to make programmatic requests to AWS services using the API or CLI.
- A user in AWS consists of a name, a password to sign into the AWS Management Console, and up to two access keys that can be used with the API or CLI.
- When you create an IAM user, you grant it permissions by making it a member of a group that has appropriate permission policies attached (recommended), or by directly attaching policies to the user.
- You can also clone the permissions of an existing IAM user, which automatically makes the new user a member of the same groups and attaches all the same policies.

IAM Groups

- An IAM group is a collection of IAM users.
- You can use groups to specify permissions for a collection of users, which can make those permissions easier to manage for those users.
- For example, you could have a group called Admins and give that group the types of permissions that administrators typically need.
- Any user in that group automatically has the permissions that are assigned to the group. If a new user joins your organization and should have administrator privileges, you can assign the appropriate permissions by adding the user to that group.
- Similarly, if a person changes jobs in your organization, instead of editing that user's permissions, you can remove him or her from the old groups and add him or her to the appropriate new groups.
- Note that a group is not truly an identity because it cannot be identified as a Principal in a resource-based or trust policy. It is only a way to attach policies to multiple users at one time.

IAM Roles

- An IAM role is very similar to a user, in that it is an identity with permission policies that determine what the identity can and cannot do in AWS.
- However, a role does not have any credentials (password or access keys) associated with it.
- Instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it.
- An IAM user can assume a role to temporarily take on different permissions for a specific task.
- A role can be assigned to a federated user who signs in by using an external identity provider instead of IAM.
- AWS uses details passed by the identity provider to determine which role is mapped to the federated user.

Temporary Credentials

- Temporary credentials are primarily used with IAM roles, but there are also other uses.
- You can request temporary credentials that have a more restricted set of permissions than your standard IAM user.
- This prevents you from accidentally performing tasks that are not permitted by the more restricted credentials.

- A benefit of temporary credentials is that they expire automatically after a set period of time.
- You have control over the duration that the credentials are valid.

Security Policies

- You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources.
- A policy is an object in AWS that, when associated with an identity or resource, defines their permissions.
- AWS evaluates these policies when an IAM principal (user or role) makes a request.
- Permissions in the policies determine whether the request is allowed or denied.
- Most policies are stored in AWS as JSON documents.
- AWS supports six types of policies.
- IAM policies define permissions for an action regardless of the method that you use to perform the operation.
- For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API.
- When you create an IAM user, you can choose to allow console or programmatic access. If console access is allowed, the IAM user can sign in to the console using a user name and password. Or if programmatic access is allowed, the user can use access keys to work with the CLI or API.

Policy Types

- **Identity-based policies** – Attach managed and inline policies to IAM identities (users, groups to which users belong, or roles). Identity-based policies grant permissions to an identity.
- **Resource-based policies** – Attach inline policies to resources. The most common examples of resource-based policies are Amazon S3 bucket policies and IAM role trust policies. Resource-based policies grant permissions to the principal that is specified in the policy. Principals can be in the same account as the resource or in other accounts.
- **Permissions boundaries** – Use a managed policy as the permissions boundary for an IAM entity (user or role). That policy defines the maximum permissions that the identity-based policies can grant to an entity, but does not grant permissions. Permissions boundaries do not define the maximum permissions that a resource-based policy can grant to an entity.
- **Organizations SCPs** – Use an AWS Organizations service control policy (SCP) to define the maximum permissions for account members of an organization or organizational unit (OU). SCPs limit permissions that identity-based policies or resource-based policies grant to entities (users or roles) within the account, but do not grant permissions.
- **Access control lists (ACLs)** – Use ACLs to control which principals in other accounts can access the resource to which the ACL is attached. ACLs are similar to resource-based policies, although they are the only policy type that does not use the JSON policy document structure. ACLs are cross-account permissions policies that grant permissions to the specified principal. ACLs cannot grant permissions to entities within the same account.
- **Session policies** – Pass advanced session policies when you use the AWS CLI or AWS API to assume a role or a federated user. Session policies limit the permissions that the role or user's identity-based policies grant to the session. Session policies limit permissions for a created session, but do not grant permissions. For more information, see Session Policies.

IAM Abilities/Features

- **Shared access to the AWS account**. The main feature of IAM is that it allows you to create separate usernames and passwords for individual users or resources and delegate access.
- **Granular permissions**. Restrictions can be applied to requests. For example, you can allow the user to download information, but deny the user the ability to update information through the policies.
- **Multifactor authentication (MFA)**. IAM supports MFA, in which users provide their username and password plus a one-time password from their phone—a randomly generated number used as an additional authentication factor.
- **Identity Federation**. If the user is already authenticated, such as through a Facebook or Google account, IAM can be made to trust that authentication method and then allow access based on it. This can also be used to allow users to maintain just one password for both on-premises and cloud environment work.
- **Free to use**. There is no additional charge for IAM security. There is no additional charge for creating additional users, groups or policies.
- **PCI DSS compliance**. The Payment Card Industry Data Security Standard is an information security standard for organizations that handle branded credit cards from the major card schemes. IAM complies with this standard.
- **Password policy**. The IAM password policy allows you to reset a password or rotate passwords remotely. You can also set rules, such as how a user should pick a password or how many attempts a user may make to provide a password before being denied access.

IAM Limitations

- Names of all IAM identities and IAM resources can be alphanumeric. They can include common characters such as plus (+), equal (=), comma (,), period (.), at (@), underscore (_), and hyphen (-).
- Names of IAM identities (users, roles, and groups) must be unique within the AWS account. So you can't have two groups named DEVELOPERS and developers in your AWS account.
- AWS account ID aliases must be unique across AWS products in your account. It cannot be a 12 digit number.
- You cannot create more than 100 groups in an AWS account.
- You cannot create more than 5000 users in an AWS account. AWS recommends the use of temporary security credentials for adding a large number of users in an AWS account.
- You cannot create more than 500 roles in an AWS account.
- An IAM user cannot be a member of more than 10 groups.
- An IAM user cannot be assigned more than 2 access keys.
- An AWS account cannot have more than 1000 customer managed policies.
- You cannot attach more than 10 managed policies to each IAM entity (user, groups, or roles).
- You cannot store more than 20 server certificates in an AWS account.
- You cannot have more than 100 SAML providers in an AWS account.
- A policy name should not exceed 128 characters.
- An alias for an AWS account ID should be between 3 and 63 characters.
- A username and role name should not exceed 64 characters.
- A group name should not exceed 128 characters.

AWS Physical and Environmental Security

- AWS data centers are state of the art, utilizing innovative architectural and engineering approaches.

- Amazon has many years of experience in designing, constructing, and operating large-scale data centers.
- This experience has been applied to the AWS platform and infrastructure.
- AWS data centers are housed in facilities that are not branded as AWS facilities.
- Physical access is strictly controlled both at the perimeter and at building ingress points by professional security staff utilizing video surveillance, intrusion detection systems, and other electronic means.
- Authorized staff must pass two-factor authentication a minimum of two times to access data center floors. All visitors are required to present identification and are signed in and continually escorted by authorized staff.
- AWS only provides data center access and information to employees and contractors who have a legitimate business need for such privileges.
- When an employee no longer has a business need for these privileges, his or her access is immediately revoked, even if they continue to be an employee of Amazon or Amazon Web Services.
- All physical access to data centers by AWS employees is logged and audited routinely.

Fire Detection and Suppression

- Automatic fire detection and suppression equipment has been installed to reduce risk.
- The fire detection system utilizes smoke detection sensors in all data center environments, mechanical and electrical infrastructure spaces, chiller rooms and generator equipment rooms.
- These areas are protected by either wet-pipe, double interlocked pre-action, or gaseous sprinkler systems.

Power

- The data center electrical power systems are designed to be fully redundant and maintainable without impact to operations, 24 hours a day, and seven days a week.
- Uninterruptible Power Supply (UPS) units provide back-up power in the event of an electrical failure for critical and essential loads in the facility.
- Data centers use generators to provide back-up power for the entire facility.

Climate and Temperature

- Climate control is required to maintain a constant operating temperature for servers and other hardware, which prevents overheating and reduces the possibility of service outages.
- Data centers are conditioned to maintain atmospheric conditions at optimal levels.
- Personnel and systems monitor and control temperature and humidity at appropriate levels.

Management

- AWS monitors electrical, mechanical, and life support systems and equipment so that any issues are immediately identified.
- Preventative maintenance is performed to maintain the continued operability of equipment.

Storage Device Decommissioning

- When a storage device has reached the end of its useful life, AWS procedures include a decommissioning process that is designed to prevent customer data from being exposed to unauthorized individuals.
- AWS uses the techniques detailed in NIST 800-88 (“Guidelines for Media Sanitization”) as part of the decommissioning process.

AWS Compliance Initiatives

- AWS Compliance enables customers to understand the robust controls in place at AWS to maintain security and data protection in the cloud.
- As systems are built on top of AWS cloud infrastructure, compliance responsibilities are shared.
- By tying together governance-focused, audit friendly service features with applicable compliance or audit standards, AWS Compliance enablers build on traditional programs; helping customers to establish and operate in an AWS security control environment.
- The IT infrastructure that AWS provides to its customers is designed and managed in alignment with security best practices and a variety of IT security standards, including:
  - SOC 1/SSAE 16/ISAE 3402 (formerly SAS 70)
  - SOC 2
  - SOC 3
  - FISMA, DIACAP, and FedRAMP
  - DOD CSM Levels 1-5
  - PCI DSS Level 1
  - ISO 9001 / ISO 27001 / ISO 27017 / ISO 27018
  - ITAR
  - FIPS 140-2
  - MTCS Level 3
  - HITRUST
- In addition, the flexibility and control that the AWS platform provides allows customers to deploy solutions that meet several industry-specific standards, including:
  - Criminal Justice Information Services (CJIS)
  - Cloud Security Alliance (CSA)
  - Family Educational Rights and Privacy Act (FERPA)
  - Health Insurance Portability and Accountability Act (HIPAA)
  - Motion Picture Association of America (MPAA)
- AWS provides a wide range of information regarding its IT control environment to customers through white papers, reports, certifications, accreditations, and other third party attestations.

Understanding Public/Private Keys

- Amazon AWS uses keys to encrypt and decrypt login information.
- At the basic level, a sender uses a public key to encrypt data, which its receiver then decrypts using another private key. These two keys, public and private, are known as a key pair.
- You need a key pair to be able to connect to your instances. The way this works on Linux and Windows instances is different.
- First, when you launch a new instance, you assign a key pair to it. Then, when you log in to it, you use the private key.
- The difference between Linux and Windows instances is that Linux instances do not have a password already set and you must use the key pair to log in to Linux instances.
- On the other hand, on Windows instances, you need the key pair to decrypt the administrator password. Using the decrypted password, you can use RDP and then connect to your Windows instance.
- Amazon EC2 stores only the public key, and you can either generate it inside Amazon EC2 or you can import it.

- Since the private key is not stored by Amazon, it's advisable to store it in a secure place as anyone who has this private key can log in on your behalf.

AWS API Security

- API Gateway supports multiple mechanisms of access control, including metering or tracking API uses by clients using API keys.
- The standard AWS IAM roles and policies offer flexible and robust access controls that can be applied to an entire API set or individual methods.
- Custom authorizers and Amazon Cognito user pools provide customizable authorization and authentication solutions.

A. Control Access to an API with IAM Permissions

- You control access to Amazon API Gateway with IAM permissions by controlling access to the following two API Gateway component processes:
  - To create, deploy, and manage an API in API Gateway, you must grant the API developer permissions to perform the required actions supported by the API management component of API Gateway.
  - To call a deployed API or to refresh the API caching, you must grant the API caller permissions to perform required IAM actions supported by the API execution component of API Gateway.

B. Use API Gateway Custom Authorizers

- An Amazon API Gateway custom authorizer is a Lambda function that you provide to control access to your API methods.
- A custom authorizer uses bearer token authentication strategies, such as OAuth or SAML. It can also use information described by headers, paths, query strings, stage variables, or context variables request parameters.
- When a client calls your API, API Gateway verifies whether a custom authorizer is configured for the API method. If so, API Gateway calls the Lambda function.
- In this call, API Gateway supplies the authorization token that is extracted from a specified request header for the token-based authorizer, or passes in the incoming request parameters as the input (for example, the event parameter) to the request parameters-based authorizer function.
- You can implement various authorization strategies, such as JSON Web Token (JWT) verification and OAuth provider callout.
- You can also implement a custom scheme based on incoming request parameter values, to return IAM policies that authorize the request. If the returned policy is invalid or the permissions are denied, the API call does not succeed.

C. Use Amazon Cognito User Pools

- In addition to using IAM roles and policies or custom authorizers, you can use an Amazon Cognito user pool to control who can access your API in Amazon API Gateway.
- To use an Amazon Cognito user pool with your API, you must first create an authorizer of the COGNITO_USER_POOLS type and then configure an API method to use that authorizer.
- After the API is deployed, the client must first sign the user in to the user pool, obtain an identity or access token for the user, and then call the API method with one of the tokens, which are typically set to the request's Authorization header.
- The API call succeeds only if the required token is supplied and the supplied token is valid, otherwise, the client isn't authorized to make the call because the client did not have credentials that could be authorized.

D. Use Client-Side SSL Certificates for Authentication by the Backend

- You can use API Gateway to generate an SSL certificate and use its public key in the backend to verify that HTTP requests to your backend system are from API Gateway.
- This allows your HTTP backend to control and accept only requests originating from Amazon API Gateway, even if the backend is publicly accessible.
- The SSL certificates that are generated by API Gateway are self-signed and only the public key of a certificate is visible in the API Gateway console or through the APIs.

E. Create and Use API Gateway Usage Plans

- After you create, test, and deploy your APIs, you can use API Gateway usage plans to extend them as product offerings for your customers.
- You can provide usage plans to allow specified customers to access selected APIs at agreed-upon request rates and quotas that can meet their business requirements and budget constraints.

AWS Security, Identity, & Compliance services

Category	Use cases	AWS service
Identity & access management	Securely manage access to services and resources	AWS Identity & Access Management
	Cloud single-sign-on (SSO) service	AWS Single Sign-On
	Identity management for your apps	Amazon Cognito
	Managed Microsoft Active Directory	AWS Directory Service
	Simple, secure service to share AWS resources	AWS Resource Access Manager
Detective controls	Unified security and compliance center	AWS Security Hub
	Managed threat detection service	Amazon GuardDuty
	Analyze application security	Amazon Inspector
	Investigate potential security issues	Amazon Detective
Infrastructure protection	DDoS protection	AWS Shield
	Filter malicious web traffic	AWS Web Application Firewall (WAF)
	Central management of firewall rules	AWS Firewall Manager
Data protection	Discover and protect your sensitive data at scale	Amazon Macie
	Key storage and management	AWS Key Management Service (KMS)
	Hardware based key storage for regulatory compliance	AWS CloudHSM
	Provision, manage, and deploy public and private SSL/TLS certificates	AWS Certificate Manager
	Rotate, manage, and retrieve secrets	AWS Secrets Manager
Compliance	No cost, self-service portal for on-demand access to AWS' compliance reports	AWS Artifact

Dark Web

- The dark web is a general term for the seedier corners of the web, where people can interact online without worrying about the watchful eye of the authorities.
- Usually, these sites are guarded by encryption mechanisms such as Tor that allow users to visit them anonymously.
- But there are also sites that don't rely on Tor, such as password-protected forums where hackers trade secrets and stolen credit card numbers, that can also be considered part of the dark web.

- People use the dark web for a variety of purposes: buying and selling drugs, discussing hacking techniques and selling hacking services and so forth.
- It's important to remember that the technologies used to facilitate "dark web" activities aren't inherently good or bad.
- The same technologies used by drug dealers to hide their identity can also be used by authorized informers to securely pass information to government agencies.

Elastic Cloud Compute (EC2)

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.
- It is designed to make web-scale cloud computing easier for developers.
- Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction.
- It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

Advantages of EC2

- In less than 10 minutes you can rent a slice of Amazon's vast cloud network and put those computing resources to work on anything from data science to bitcoin mining.
- EC2 offers a number of benefits and advantages over alternatives. Most notably:

Affordability

- EC2 allows you to take advantage of Amazon's enormous scale.
- You can pay a very low rate for the resources you use. The smallest EC2 instance can be rented for as little as $.0058 per hour which works out to about $4.18 per month. Of course, instances with more resources are more expensive but this gives you a sense of how affordable EC2 instances are.
- With EC2 instances, you're only paying for what you use in terms of compute hours and bandwidth so there's little wasted expense.

Ease of use

- Amazon's goal with EC2 was to make accessing compute resources low friction and, by and large, they've succeeded.
- Launching an instance is simply a matter of logging into the AWS Console, selecting your operating system, instance type, and storage options.
- At most, it's a 10 minute process and there aren't any major technical barriers preventing anyone from spinning up an instance, though it may take some technical knowledge to leverage those resources after launch.

Scalability

- You can easily add EC2 instances as needed, creating your own private cloud of computer resources that perfectly matches your needs.
- Here at Pagely a common configuration is an EC2 instance to run a WordPress app, an instance to run RDS (a database service), and an EBS so that data can easily be moved and shared between instances as they're added.
- AWS offers built-in, rules-based auto scaling so that you can automatically turn instances on or off based on demand.
- This helps you ensure that you're never wasting resources but you also have enough resources available to do the job.

Integration

- Perhaps the biggest advantage of EC2, and something no competing solution can claim, is its native integration with the vast ecosystem of AWS services.

- Currently there are over 170 services. No other cloud network can claim the breadth, depth, and flexibility AWS can.

EC2 Image Builder

- EC2 Image Builder simplifies the creation, maintenance, validation, sharing, and deployment of Linux or Windows Server images for use with Amazon EC2 and on-premises.
- Keeping server images up-to-date can be time consuming, resource intensive, and error-prone.
- Currently, customers either manually update and snapshot VMs or have teams that build automation scripts to maintain images.
- Image Builder significantly reduces the effort of keeping images up-to-date and secure by providing a simple graphical interface, built-in automation, and AWS-provided security settings.
- With Image Builder, there are no manual steps for updating an image nor do you have to build your own automation pipeline.
- Image Builder is offered at no cost, other than the cost of the underlying AWS resources used to create, store, and share the images.

Auto Scaling

- AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost.
- Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes.
- The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas.
- AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them.
- If you're already using Amazon EC2 Auto Scaling to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services.
- With AWS Auto Scaling, your applications always have the right resources at the right time.
- It's easy to get started with AWS Auto Scaling using the AWS Management Console, Command Line Interface (CLI), or SDK.
- AWS Auto Scaling is available at no additional charge. You pay only for the AWS resources needed to run your applications and Amazon CloudWatch monitoring fees.

Elastic Load Balancing

- Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses.
- It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.
- Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant.
    - Application Load Balancers,
    - Network Load Balancers, and
    - Classic Load Balancers.

Application Load Balancer

- Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including micro services and containers.
- Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.

Network Load Balancer

- Network Load Balancer is best suited for load balancing of TCP traffic where extreme performance is required.
- Operating at the connection level (Layer 4), Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies.
- Network Load Balancer is also optimized to handle sudden and volatile traffic patterns.

Classic Load Balancer

- Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level.
- Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

Benefits of Elastic Load Balancing for reducing workload

Highly Available

- Elastic Load Balancing automatically distributes incoming traffic across multiple targets – Amazon EC2 instances, containers, and IP addresses – in multiple Availability Zones and ensures only healthy targets receive traffic. Elastic Load Balancing can also load balance across a Region, routing traffic to healthy targets in different Availability Zones.

Secure

- Elastic Load Balancing works with Amazon Virtual Private Cloud (VPC) to provide robust security features, including integrated certificate management and SSL decryption. Together, they give you the flexibility to centrally manage SSL settings and offload CPU intensive workloads from your applications.

Elastic

- Elastic Load Balancing is capable of handling rapid changes in network traffic patterns. Additionally, deep integration with Auto Scaling ensures sufficient application capacity to meet varying levels of application load without requiring manual intervention.

Flexible

- Elastic Load Balancing also allows you to use IP addresses to route requests to application targets. This offers you flexibility in how you virtualize your application targets, allowing you to host more applications on the same instance. This also enables these applications to have individual security groups and use the same network port to further simplify inter-application communication in microservices based architecture.

Robust Monitoring and Auditing

- Elastic Load Balancing allows you to monitor your applications and their performance in real time with Amazon CloudWatch metrics, logging, and request tracing. This improves visibility into the behavior of

your applications, uncovering issues and identifying performance bottlenecks in your application stack at the granularity of an individual request.

Hybrid Load Balancing

- Elastic Load Balancing offers ability to load balance across AWS and on-premises resources using the same load balancer. This makes it easy for you to migrate, burst, or failover on-premises applications to the cloud.

AMIs

- An Amazon Machine Image (AMI) provides the information required to launch an instance.
- You must specify an AMI when you launch an instance.
- You can launch multiple instances from a single AMI when you need multiple instances with the same configuration.
- You can use different AMIs to launch instances when you need instances with different configurations.
- An AMI includes the following:
  - One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
  - Launch permissions that control which AWS accounts can use the AMI to launch instances.
  - A block device mapping that specifies the volumes to attach to the instance when it's launched.

Using an AMI

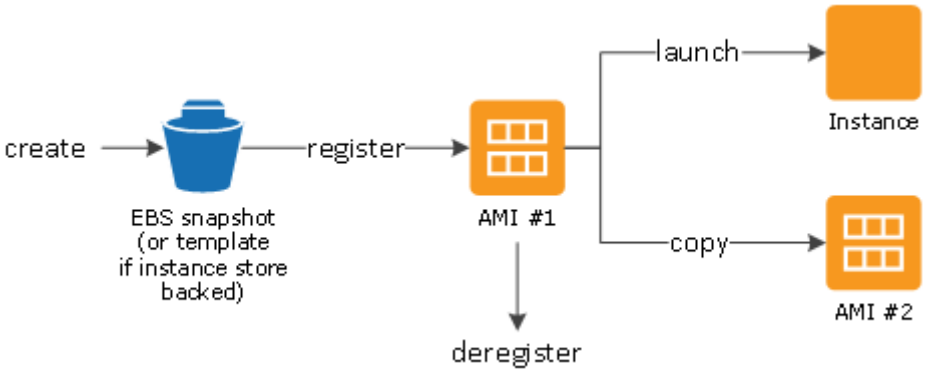


Fig. : The AMI lifecycle (create, register, launch, copy, and deregister)

- The following diagram summarizes the AMI lifecycle.
- After you create and register an AMI, you can use it to launch new instances. (You can also launch instances from an AMI if the AMI owner grants you launch permissions.)
- You can copy an AMI within the same Region or to different Regions.
- When you no longer require an AMI, you can deregister it.
- You can search for an AMI that meets the criteria for your instance.
- You can search for AMIs provided by AWS or AMIs provided by the community.
- After you launch an instance from an AMI, you can connect to it.
- When you are connected to an instance, you can use it just like you use any other server.
- For information about launching, connecting, and using your instance, see Amazon EC2 instances.

Multi Tenancy

- In cloud computing, multi tenancy means that multiple customers of a cloud vendor are using the same computing resources.
- Despite the fact that they share resources, cloud customers aren't aware of each other, and their data is kept totally separate.
- Multi tenancy is a crucial component of cloud computing; without it, cloud services would be far less practical.
- Multitenant architecture is a feature in many types of public cloud computing, including IaaS, PaaS, SaaS, containers, and server less computing.
- To understand multi tenancy, think of how banking works.
- Multiple people can store their money in one bank, and their assets are completely separate even though they're stored in the same place.
- Customers of the bank don't interact with each other, don't have access to other customers' money, and aren't even aware of each other.
- Similarly, in public cloud computing, customers of the cloud vendor use the same infrastructure – the same servers, typically – while still keeping their data and their business logic separate and secure.
- The classic definition of multi tenancy was a single software instance that served multiple users, or tenants.
- However, in modern cloud computing, the term has taken on a broader meaning, referring to shared cloud infrastructure instead of just a shared software instance.

Cataloging the Marketplace

- AWS Marketplace is a curated digital catalog customers can use to find, buy, deploy, and manage third-party software, data, and services that customers need to build solutions and run their businesses.
- AWS Marketplace includes thousands of software listings from popular categories such as security, networking, storage, machine learning, business intelligence, database, and DevOps.
- AWS Marketplace also simplifies software licensing and procurement with flexible pricing options and multiple deployment methods.
- In addition, AWS Marketplace includes data products available from AWS Data Exchange.
- Customers can quickly launch preconfigured software with just a few clicks, and choose software solutions in Amazon Machine Images (AMIs), software as a service (SaaS), and other formats.
- You can browse and subscribe to data products.
- Flexible pricing options include free trial, hourly, monthly, annual, multi-year, and BYOL, and get billed from one source.
- AWS handles billing and payments, and charges appear on customers' AWS bill.
- You can use AWS Marketplace as a buyer (subscriber), seller (provider), or both.
- Anyone with an AWS account can use AWS Marketplace as a buyer, and can register to become a seller.
- A seller can be an independent software vendor (ISV), value-added reseller, or individual who has something to offer that works with AWS products and services.
- Every software product on AWS Marketplace has been through a curation process.
- On the product page, there can be one or more offerings for the product.
- When the seller submits a product in AWS Marketplace, they define the price of the product and the terms and conditions of use.

- When a consumer subscribes to a product offering, they agree to the pricing and terms and conditions set for the offer.
- The product can be free to use or it can have an associated charge.
- The charge becomes part of your AWS bill, and after you pay, AWS Marketplace pays the seller.
- Products can take many forms. For example, a product can be offered as an Amazon Machine Image (AMI) that is instantiated using your AWS account.
- The product can also be configured to use AWS CloudFormation templates for delivery to the consumer.
- The product can also be software as a service (SaaS) offerings from an ISV, web ACL, set of rules, or conditions for AWS WAF.
- Software products can be purchased at the listed price using the ISV's standard end user license agreement (EULA) or offered with customer pricing and EULA.
- Products can also be purchased under a contract with specified time or usage boundaries.
- After the product subscriptions are in place, the consumer can copy the product to their AWS Service Catalog to manage how the product is accessed and used in the consumer's organization.

Selling On the Marketplace.

- As a seller, go to the AWS Marketplace Management Portal to register.
- If you're providing a data product or you're charging for use of your software product, you must also provide tax and banking information as part of your registration.
- When you register, you create a profile for your company or for yourself that is discoverable on AWS Marketplace.
- You also use the AWS Marketplace Management Portal to create and manage product pages for your products.
- Eligible partners can programmatically list AWS Marketplace products outside of AWS Marketplace.
- For information about becoming an eligible partner, contact your AWS Marketplace business development partner.

Virtual private clouds

- A **virtual private cloud (VPC)** is an on-demand configurable pool of shared computing resources allocated within a public cloud environment, providing a certain level of isolation between the different organizations.
- You already know that there are three major types of clouds: Public, Private and Hybrid. Now, there's a newer player in the game: Virtual Private Clouds.
- What makes these different from public and private clouds, and what is the benefit? Is it just a fancy name for public cloud, or is it a private one?
- VPC are related to the public cloud, but they are not the same. Instead of sharing resources and space in a public infrastructure, you get a changeable allotment of resources to configure.
- There is a certain level of isolation between you and other users, via a private IP subnet and virtual communication construct (such as a VLAN) on a per user basis.
- This ensures a secure method of remotely accessing your cloud resources. This isolation within a public cloud lends the name "virtual private" because you are essentially operating a private cloud within a public cloud.
- That also doesn't mean Virtual Private Clouds and private clouds are the same.
- Private clouds are entirely dedicated to your organization, and that includes the hardware.
- Virtual Private clouds do not have the same hardware dedication; it just creates a more secure environment on public infrastructure.
- Think of it as operating like a VPN: You use them to send messages over the public internet in a secure way as if you had your own personal network, but it's not the same as actually having your own.
- What's the benefit to this? Wouldn't it just be easier to have a private cloud? Not necessarily. Private clouds are expensive to operate, and because the hardware as well as the resources required to run it belong to you alone, there is no one to share that cost with.
- Virtual Private Clouds give you the best of both worlds: A private cloud for security and compliance purposes, reduced infrastructure costs that come with public clouds. The allotment of resources is yours to use, so there is no worry about running out or having to share with others. You simply are sharing the infrastructure.
- Virtual Private Clouds are commonly used with Infrastructure as a Service (IaaS) providers.
- Because the shared resources (CPU, RAM, etc.) are not always the responsibility of the hardware provider, it is possible to have different infrastructure and VPC providers.
- However, having the same VPC and infrastructure provider can help cut down on the confusion and communication process between you and your vendor.

Amazon Route 53 Announces Private DNS within Amazon VPC

- You can now use Amazon Route 53, AWS's highly available and scalable DNS service, to easily manage your internal domain names with the same simplicity, security, and cost effectiveness that Route 53 already provides for external DNS names.
- You can use the Route 53 Private DNS feature to manage authoritative DNS within your Virtual Private Clouds (VPCs), so you can use custom domain names for your internal AWS resources without exposing DNS data to the public Internet.
- You can use Route 53 Private DNS to manage internal DNS hostnames for resources like application servers, database servers, and web servers.

- Route 53 will only respond to queries for these names when the queries originate from within the VPC(s) that you authorize.
- Using custom internal DNS names (rather than IP addresses or AWS-provided names such as ec2-10-1-2-3.us-west-2.compute.amazonaws.com) has a variety of benefits, for example, being able to flip from one database to another just by changing the mapping of a domain name such as internal.example.com to point to a new IP address.
- Route 53 also supports split-view DNS, so you can configure public and private hosted zones to return different external and internal IP addresses for the same domain names.

Relational Database Service

- Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud.
- It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups.
- It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.
- Amazon RDS is available on several database instance types optimized for memory, performance or I/O and provides you with six familiar database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server.
- You can use the AWS Database Migration Service to easily migrate or replicate your existing databases to Amazon RDS.

Advantages/Benefits of Relational Database Service

(i) Easy to Administer

- Amazon RDS makes it easy to go from project conception to deployment. Use the AWS Management Console, the AWS RDS Command-Line Interface, or simple API calls to access the capabilities of a production-ready relational database in minutes. No need for infrastructure provisioning, and no need for installing and maintaining database software.

(ii) Highly Scalable

- We can scale our database's compute and storage resources with only a few mouse clicks or an API call, often with no downtime. Many Amazon RDS engine types allow you to launch one or more Read Replicas to offload read traffic from your primary database instance.

(iii) Available and Durable

- Amazon RDS runs on the same highly reliable infrastructure used by other Amazon Web Services. When you provision a Multi-AZ DB Instance, Amazon RDS synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Amazon RDS has many other features that enhance reliability for critical production databases, including automated backups, database snapshots, and automatic host replacement.

(iv) Fast

- Amazon RDS supports the most demanding database applications. You can choose between two SSD-backed storage options: one optimized for high-performance OLTP applications, and the other for cost-effective general-purpose use. In addition, Amazon Aurora provides performance on par with commercial databases at 1/10th the cost.

(v) Secure

- Amazon RDS makes it easy to control network access to your database. Amazon RDS also lets you run your database instances in Amazon Virtual Private Cloud (Amazon VPC), which enables you to isolate your database instances and to connect to your existing IT infrastructure through an industry-standard encrypted IPsec VPN. Many Amazon RDS engine types offer encryption at rest and encryption in transit.

(vi) Inexpensive

- You pay very low rates and only for the resources you actually consume. In addition, you benefit from the option of On-Demand pricing with no up-front or long-term commitments, or even lower hourly rates via Reserved Instance pricing.

DynamoDB

- Amazon DynamoDB -- also known as Dynamo Database or DDB -- is a fully managed NoSQL database service provided by Amazon Web Services. DynamoDB is known for low latencies and scalability.
- According to AWS, DynamoDB makes it simple and cost-effective to store and retrieve any amount of data, as well as serve any level of request traffic.
- All data items are stored on solid-state drives, which provide high I/O performance and can more efficiently handle high-scale requests.
- An AWS user interacts with the service by using the AWS Management Console or a DynamoDB API.
- DynamoDB uses a NoSQL database model, which is nonrelational, allowing documents, graphs and columnar among its data models.
- A user stores data in DynamoDB tables, then interacts with it via GET and PUT queries, which are read and write operations, respectively.
- DynamoDB supports basic CRUD operations and conditional operations. Each DynamoDB query is executed by a primary key identified by the user, which uniquely identifies each item.

Scalability, Availability and Durability

- DynamoDB enforces replication across three availability zones for high availability, durability and read consistency.
- A user can also opt for cross-region replication, which creates a backup copy of a DynamoDB table in one or more global geographic locations.
- The DynamoDB scan API provides two consistency options when reading DynamoDB data:
  - Eventually consistent reads
  - Strongly consistent reads
- The former, which is the AWS default setting, maximizes throughput at the potential expense of not having a read reflect the latest write or update. The latter reflects all writes and updates.
- There are no DynamoDB limits on data storage per user, nor a maximum throughput per table.

Security

- Amazon DynamoDB offers Fine-Grained Access Control (FGAC) for an administrator to protect data in a table.
- The admin or table owner can specify who can access which items or attributes in a table and what actions that person can perform.
- FGAC is based on the AWS Identity and Access Management service, which manages credentials and permissions.

- As with other AWS products, the cloud provider recommends a policy of least privilege when granting access to items and attributes.
- An admin can view usage metrics for DynamoDB with Amazon CloudWatch.

Additional DynamoDB Features

- The DynamoDB Triggers feature integrates with AWS Lambda to allow a developer to code actions based on updates to items in a DynamoDB table, such as sending a notification or connecting a table to another data source.
- The developer associates a Lambda function, which stores the logic code, with the stream on a DynamoDB table.
- AWS Lambda then reads updates to a table from a stream and executes the function.
- The DynamoDB Streams feature provides a 24-hour chronological sequence of updates to items in a table.
- An admin can access the stream via an API call to take action based on updates, such as synchronizing information with another data store. An admin enables DynamoDB Streams on a per-table basis.

Advantages of DynamoDB

Performance at scale

- DynamoDB supports some of the world's largest scale applications by providing consistent, single-digit millisecond response times at any scale.
- You can build applications with virtually unlimited throughput and storage.
- DynamoDB global tables replicate your data across multiple AWS Regions to give you fast, local access to data for your globally distributed applications.
- For use cases that require even faster access with microsecond latency, DynamoDB Accelerator (DAX) provides a fully managed in-memory cache.

No servers to manage

- DynamoDB is server less with no servers to provision, patch, or manage and no software to install, maintain, or operate.
- DynamoDB automatically scales tables up and down to adjust for capacity and maintain performance.
- Availability and fault tolerance are built in, eliminating the need to architect your applications for these capabilities.
- DynamoDB provides both provisioned and on-demand capacity modes so that you can optimize costs by specifying capacity per workload, or paying for only the resources you consume.

Enterprise ready

- DynamoDB supports ACID transactions to enable you to build business-critical applications at scale.
- DynamoDB encrypts all data by default and provides fine-grained identity and access control on all your tables.
- You can create full backups of hundreds of terabytes of data instantly with no performance impact to your tables, and recover to any point in time in the preceding 35 days with no downtime.
- DynamoDB is also backed by a service level agreement for guaranteed availability.

ElastiCache

- ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud.

- It provides a high-performance, scalable, and cost-effective caching solution, while removing the complexity associated with deploying and managing a distributed cache environment.
- With ElastiCache, you can quickly deploy your cache environment, without having to provision hardware or install software.
- You can choose from Memcached or Redis protocol-compliant cache engine software, and let ElastiCache perform software upgrades and patch management for you.
- For enhanced security, ElastiCache can be run in the Amazon Virtual Private Cloud (Amazon VPC) environment, giving you complete control over network access to your clusters.
- With just a few clicks in the AWS Management Console, you can add or remove resources such as nodes, clusters, or read replicas to your ElastiCache environment to meet your business needs and application requirements.
- Existing applications that use Memcached or Redis can use ElastiCache with almost no modification.
- Your applications simply need to know the host names and port numbers of the ElastiCache nodes that you have deployed.
- The ElastiCache Auto Discovery feature for Memcached lets your applications identify all of the nodes in a cache cluster and connect to them, rather than having to maintain a list of available host names and port numbers.
- In this way, your applications are effectively insulated from changes to node membership in a cluster.
- ElastiCache has multiple features to enhance reliability for critical production deployments:
  - Automatic detection and recovery from cache node failures.
  - Multi-AZ with Automatic Failover of a failed primary cluster to a read replica in Redis clusters that support replication (called replication groups in the ElastiCache API and AWS CLI.
  - Flexible Availability Zone placement of nodes and clusters.
  - Integration with other AWS services such as Amazon EC2, Amazon CloudWatch, AWS CloudTrail, and Amazon SNS to provide a secure, high-performance, managed in-memory caching solution.

ElastiCache Nodes

- A node is the smallest building block of an ElastiCache deployment.
- A node can exist in isolation from or in some relationship to other nodes.
- A node is a fixed-size chunk of secure, network-attached RAM.
- Each node runs an instance of the engine and version that was chosen when you created your cluster.
- If necessary, you can scale the nodes in a cluster up or down to a different instance type.
- Every node within a cluster is the same instance type and runs the same cache engine.
- Each cache node has its own Domain Name Service (DNS) name and port.
- Multiple types of cache nodes are supported, each with varying amounts of associated memory.
- You can purchase nodes on a pay-as-you-go basis, where you only pay for your use of a node.
- Or you can purchase reserved nodes at a much-reduced hourly rate.
- If your usage rate is high, purchasing reserved nodes can save you money.

ElastiCache for Redis Shards

- A Redis shard (called a node group in the API and CLI) is a grouping of one to six related nodes.
- A Redis (cluster mode disabled) cluster always has one shard.
- A Redis (cluster mode enabled) cluster can have 1–90 shards.
- A multiple node shard implements replication by having one read/write primary node and 1–5 replica nodes.

ElastiCache for Redis Clusters

- A Redis cluster is a logical grouping of one or more ElastiCache for Redis Shards.
- Data is partitioned across the shards in a Redis (cluster mode enabled) cluster.
- Many ElastiCache operations are targeted at clusters:
  - Creating a cluster
  - Modifying a cluster
  - Taking snapshots of a cluster (all versions of Redis)
  - Deleting a cluster
  - Viewing the elements in a cluster
  - Adding or removing cost allocation tags to and from a cluster

Redshift

- Perhaps one of the most exciting outcomes of the public cloud was addressing the shortcomings of traditional enterprise data warehouse (EDW) storage and processing. The fast provisioning, commodity costs, infinite scale, and pay-as-you-grow pricing of public cloud are a natural fit for EDW needs, providing even the smallest of users the ability to now get valuable answers to business intelligence (BI) questions.
- **Amazon Redshift** is one such system built to address EDW needs, and it boasts low costs, an easy SQL-based access model, easy integration to other Amazon Web Services (AWS) solutions, and most importantly, high query performance.
- Amazon Redshift gets its name from the astronomical phenomenon noticed by Hubble, which explained the expansion of the universe. By adopting the Amazon Redshift moniker, AWS wanted to relay to customers that the service was built to handle the perpetual expansion of their data.

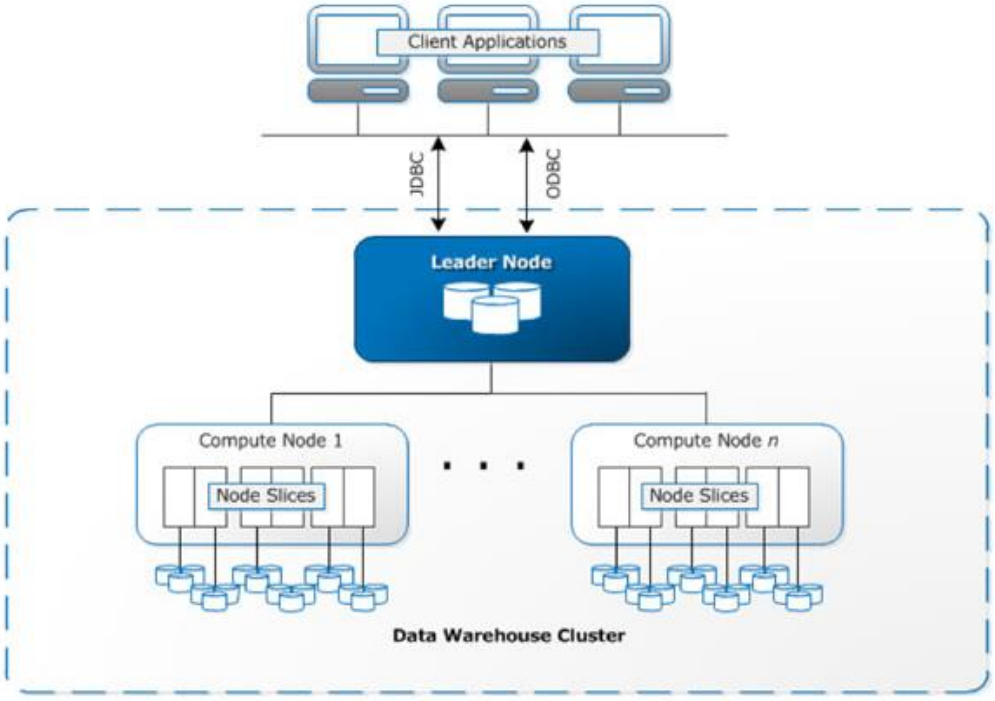


Fig. : Amazon Redshift Architecture

- An Amazon Redshift cluster consists of one leader node (which clients submit queries to) and one or more follower (or “compute”) nodes, which actually perform the queries on locally stored data.
- By allowing for unlimited expansion of follower nodes, Amazon Redshift ensures that customers can continue to grow their cluster as their data needs grow.

- Customers can start with a "cluster" as small as a single node (acting as both leader and follower), and for the smallest supported instance type (a DW2), that could be as low cost as $0.25/hour or about $180/month. By using "Reservations" (paying an up-front fee in exchange for a lower hourly running cost) for the underlying instances, Amazon Redshift can cost as little as $1,000/TB/year — upwards of one-fifth to one-tenth of the cost of a traditional EDW.
- Because Amazon Redshift provides native Open Database Connectivity (ODBC) and Database Connectivity (JDBC) connectivity (in addition to PostgresSQL driver support), most third-party BI tools (like Tableu, Qlikview, and MicroStrategy) work right out of the box. Amazon Redshift also uses the ubiquitous Structured Query Language (SQL) language for queries, ensuring that your current resources can quickly and easily become productive with the technology.
- Amazon Redshift was custom designed from the ParAccel engine — an analytic database which used columnar storage and parallel processing to achieve very fast I/O.
- Columns of data in Amazon Redshift are stored physically adjacent on disk, meaning that queries and scans on those columns (common in online analytical processing [OLAP] queries) run very fast.
- Additionally, Amazon Redshift uses 10GB Ethernet interconnects, and specialized EC2 instances (with between three and 24 spindles per node) to achieve high throughput and low latency.
- For even faster queries, Amazon Redshift allows customers to use column-level compression to both greatly reduce the amount of data that needs stored, and reduce the amount of disk I/O.
- Amazon Redshift, like many of AWS's most popular services, is also fully managed, meaning that low-level, time-consuming administrative tasks like OS patching, backups, replacing failed hardware, and software upgrades are handled automatically and transparently.
- With Amazon Redshift, users simply provision a cluster, load it with their data, and begin executing queries. All data is continuously, incrementally, automatically backed up in the highly durable S3, and enabling disaster recovery across regions can be accomplished with just a few clicks.
- Spinning a cluster up can be as simple as a few mouse clicks, and as fast as a few minutes.
- A very exciting aspect of Amazon Redshift, and something that is not possible in traditional EDWs, is the ability to easily scale a provisioned cluster up and down.
- In Amazon Redshift, this scaling is transparent to the customer—when a resize is requested, data is copied in parallel from the source cluster (which continues to function in read-only mode) to a new cluster, and once all data is live migrated, DNS is flipped to the new cluster and the old cluster is de-provisioned.
- This allows customers to easily scale up and down, and each scaling event nicely re-stripes the data across the new cluster for a balanced workload.
- Amazon Redshift offers mature, native, and tunable security. Clusters can be deployed into a Virtual Private Cloud (VPC), and encryption of data is supported via hardware accelerated AES-256 (for data at rest) and SSL (for data on the wire).
- Compliance teams will be pleased to learn that users can manage their own encryption keys via AWS's Hardware Security Module (HSM) service, and that Amazon Redshift provides a full audit trail of all SQL connection attempts, queries, and modifications of the cluster.

Advantages of Amazon Redshift

Exceptionally fast

- Redshift is very fast when it comes to loading data and querying it for analytical and reporting purposes.
- Redshift has Massively Parallel Processing (MPP) Architecture which allows you to load data at blazing fast speed.
- In addition, using this architecture, Redshift distributes and parallelize your queries across multiple nodes.

- Redshift gives you an option to use Dense Compute nodes which are SSD based data warehouses. Using this you can run most complex queries in very less time.

High Performance

- As discussed in the previous point, Redshift gains high performance using massive parallelism, efficient data compression, query optimization, and distribution.
- MPP enables Redshift to parallelize data loading, backup and restore operation. Furthermore, queries that you execute get distributed across multiple nodes.
- Redshift is a columnar storage database, which is optimized for huge and repetitive type of data. Using columnar storage, reduces the I/O operations on disk drastically, improving performance as a result.
- Redshift gives you an option to define column-based encoding for data compression. If not specified by the user, redshift automatically assigns compression encoding.
- Data compression helps in reducing memory footprint and significantly improves the I/O speed.

Horizontally Scalable

- Scalability is a very crucial point for any Data warehousing solution and Redshift does pretty well job in that.
- Redshift is horizontally scalable. Whenever you need to increase the storage or need it to run faster, just add more nodes using AWS console or Cluster API and it will upscale immediately.
- During this process, your existing cluster will remain available for read operations so your application stays uninterrupted.
- During the scaling operation, Redshift moves data parallel between compute nodes of old and new clusters. Therefore enabling the transition to complete smoothly and as quickly as possible.

Massive Storage capacity

- As expected from a Data warehousing solution, Redshift provides massive storage capacity.
- A basic setup can give you a petabyte range of data storage.
- In addition, Redshift gives you an option to choose Dense Storage type of compute nodes which can provide large storage space using Hard Disk Drives for a very low price.
- You can further increase the storage by adding more nodes to your cluster and it can go well beyond petabyte of data range.

Attractive and transparent pricing

- Pricing is a very strong point in favor of Redshift, it is considerably cheaper than alternatives or an on premise solution. Redshift has 2 pricing models, pay as you go and reserved instance.
- Hence this gives you the flexibility to categorize this expense as an operational expense or capital expense.
- If your use case requires more data storage, then with 3 years reserved instance Dense Storage plan, effective price per terabyte per year can be as low as $935.
- Comparing this to traditional on premise storage, which roughly costs around $19k-$25k per terabyte, Redshift is significantly cheaper.

SQL interface

- Redshift Query Engine is based on ParAccel which has the same interface as PostgreSQL If you are already familiar with SQL, you don't need to learn a lot of new techs to start using query module of Redshift.
- Since Redshift uses SQL, it works with existing Postgres JDBC/ODBC drivers, readily connecting to most of the Business Intelligence tools.

AWS ecosystem

- Many businesses are running their infrastructure on AWS already, EC2 for servers, S3 for long-term storage, RDS for database and this number is constantly increasing.
- Redshift works very well if the rest of your infra is already on AWS and you get the benefit of data locality and cost of data transport is comparatively low.
- For a lot of businesses, S3 has become the de-facto destination for cloud storage.
- Since Redshift is virtually co-located with S3 and it can access formatted data on S3 with single COPY command.
- When loading or dumping data on S3, Redshift uses Massive Parallel Processing which can move data at a very fast speed.

Security

- Amazon Redshift comes packed with various security features.
- There are options like VPC for network isolation, various ways to handle access control, data encryption etc.
- Data encryption option is available at multiple places in Redshift.
- To encrypt data stored in your cluster you can enable cluster encryption at the time of launching the cluster.
- Also, to encrypt data in transit, you can enable SSL encryption.
- When loading data from S3, redshift allows you to use either server-side encryption or client-side encryption.
- Finally, at the time of loading data, S3 or Redshift copy command handles the decryption respectively.
- Amazon Redshift clusters can be launched inside your infrastructure Virtual Private Cloud (VPC).
- Hence you can define VPC security groups to restrict inbound or outbound access to your redshift clusters.
- Using the robust Access Control system of AWS, you can grant privilege to specific users or maintain access on specific database level.
- Additionally, you can even define users and groups to have access to specific data in tables.

Amazon Redshift Limitations

Doesn't enforce uniqueness

- There is no way in redshift to enforce uniqueness on inserted data.
- Hence, if you have a distributed system and it writes data on Redshift, you will have to handle the uniqueness yourself either on the application layer or by using some method of data de-duplication.

Only S3, DynamoDB and Amazon EMR support for parallel upload

- If your data is in Amazon S3 or relational DynamoDB or on Amazon EMR, Redshift can load it using Massively Parallel Processing which is very fast.
- But for all other sources, parallel loading is not supported.
- You will either have to use JDBC inserts or some scripts to load data into Redshift.
- Alternatively, you can use an ETL solution like Hevo which can load your data into Redshift parallel from 100s of sources.

Requires a good understanding of Sort and Distribution keys

- Sort keys and Distribution keys decide how data is stored and indexed across all Redshift nodes.
- Therefore, you need to have a solid understanding of these concepts and you need to properly set them on your tables for optimal performance.

- There can be only one distribution key for a table and that cannot be changed later on, which means you have to think carefully and anticipate future workloads before deciding Distribution key.

Can't be used as live app database

- While Redshift is very fast when running queries on a huge amount of data or running reporting and analytics, but it is not fast enough for live web apps.
- So you will have to pull data into a caching layer or a vanilla Postgres instance to serve redshift data to web apps.

Data on Cloud

- Though it is a good thing for most of the people, in some use cases it could be a point of concern.
- So if you are concerned with the privacy of data or your data has extremely sensitive content, you may not be comfortable putting it on the cloud.

High performance AWS Networking.

- High performance AWS Networking is nothing but use of various network services provided by AWS for better performance.
- AWS Networking include following services:
  1. Private DNS Servers
     - The Private DNS are name servers that reflect your domain name rather than our default ones.
     - Having private nameservers could be useful if you intend to resell hosting services or want to brand your business.
     - Also, when using Private DNS, if a domain name is migrated to another server, there is no need to change any nameservers and the domain names will automatically point to the new location.
  2. Virtual Private Clouds (Explain Earlier)
  3. Cloud Models (Explain Earlier) etc.

Big Data Analytics

- Big data analytics is the often complex process of examining large and varied data sets, or big data, to uncover information -- such as hidden patterns, unknown correlations, market trends and customer preferences -- that can help organizations make informed business decisions.

AWS Analytics Services

Amazon Athena

- Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL.
- Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.
- Athena is easy to use. Simply point to your data in Amazon S3, define the schema, and start querying using standard SQL.
- Most results are delivered within seconds. With Athena, there's no need for complex extract, transform, and load (ETL) jobs to prepare your data for analysis.
- This makes it easy for anyone with SQL skills to quickly analyze large-scale datasets.
- Athena is out-of-the-box integrated with AWS Glue Data Catalog, allowing you to create a unified metadata repository across various services, crawl data sources to discover schemas and populate your Catalog with new and modified table and partition definitions, and maintain schema versioning.
- You can also use Glue's fully-managed ETL capabilities to transform data or convert it into columnar formats to optimize cost and improve performance.

Amazon EMR

- Amazon EMR provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances.
- You can also run other popular distributed frameworks such as Apache Spark, HBase, Presto, and Flink in Amazon EMR, and interact with data in other AWS data stores such as Amazon S3 and Amazon DynamoDB.
- EMR Notebooks, based on the popular Jupyter Notebook, provide a development and collaboration environment for ad hoc querying and exploratory analysis.
- Amazon EMR securely and reliably handles a broad set of big data use cases, including log analysis, web indexing, data transformations (ETL), machine learning, financial analysis, scientific simulation, and bioinformatics.

Amazon CloudSearch

- Amazon CloudSearch is a managed service in the AWS Cloud that makes it simple and cost-effective to set up, manage, and scale a search solution for your website or application.
- Amazon CloudSearch supports 34 languages and popular search features such as highlighting, autocomplete, and geospatial search.

Amazon Elasticsearch Service

- Amazon Elasticsearch Service makes it easy to deploy, secure, operate, and scale Elasticsearch to search, analyze, and visualize data in real-time.
- With Amazon Elasticsearch Service, you get easy-to-use APIs and real-time analytics capabilities to power use-cases such as log analytics, full-text search, application monitoring, and clickstream analytics, with enterprise-grade availability, scalability, and security.

- The service offers integrations with open-source tools like Kibana and Logstash for data ingestion and visualization.
- It also integrates seamlessly with other AWS services such as Amazon Virtual Private Cloud (Amazon VPC), AWS Key Management Service (AWS KMS), Amazon Kinesis Data Firehose, AWS Lambda, AWS Identity and Access Management (IAM), Amazon Cognito, and Amazon CloudWatch, so that you can go from raw data to actionable insights quickly.

Amazon Kinesis

- Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information.
- Amazon Kinesis offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application.
- With Amazon Kinesis, you can ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications.
- Amazon Kinesis enables you to process and analyze data as it arrives and respond instantly instead of having to wait until all your data is collected before the processing can begin.
- Amazon Kinesis currently offers four services: Kinesis Data Firehose, Kinesis Data Analytics, Kinesis Data Streams, and Kinesis Video Streams.

Amazon Kinesis Data Firehose

- Amazon Kinesis Firehose is the easiest way to reliably load streaming data into data stores and analytics tools.
- It can capture, transform, and load streaming data into Amazon S3, Amazon Redshift, Amazon Elasticsearch Service, and Splunk, enabling near real-time analytics with existing business intelligence tools and dashboards you're already using today.
- It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration.
- It can also batch, compress, transform, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security.
- You can easily create a Firehose delivery stream from the AWS Management Console, configure it with a few clicks, and start sending data to the stream from hundreds of thousands of data sources to be loaded continuously to AWS—all in just a few minutes.
- You can also configure your delivery stream to automatically convert the incoming data to columnar formats like Apache Parquet and Apache ORC, before the data is delivered to Amazon S3, for cost-effective storage and analytics.

Amazon Kinesis Data Analytics

- Amazon Kinesis Data Analytics is the easiest way to analyze streaming data, gain actionable insights, and respond to your business and customer needs in real time.
- Amazon Kinesis Data Analytics reduces the complexity of building, managing, and integrating streaming applications with other AWS services.
- SQL users can easily query streaming data or build entire streaming applications using templates and an interactive SQL editor.
- Java developers can quickly build sophisticated streaming applications using open source Java libraries and AWS integrations to transform and analyze data in real-time.
- Amazon Kinesis Data Analytics takes care of everything required to run your queries continuously and scales automatically to match the volume and throughput rate of your incoming data.

Amazon Kinesis Data Streams

- Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service.
- KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events.
- The data collected is available in milliseconds to enable real-time analytics use cases such as real-time dashboards, real-time anomaly detection, dynamic pricing, and more.

Amazon Kinesis Video Streams

- Amazon Kinesis Video Streams makes it easy to securely stream video from connected devices to AWS for analytics, machine learning (ML), playback, and other processing.
- Kinesis Video Streams automatically provisions and elastically scales all the infrastructure needed to ingest streaming video data from millions of devices.
- It also durably stores, encrypts, and indexes video data in your streams, and allows you to access your data through easy-to-use APIs.
- Kinesis Video Streams enables you to playback video for live and on-demand viewing, and quickly build applications that take advantage of computer vision and video analytics through integration with Amazon Recognition Video, and libraries for ML frameworks such as Apache MxNet, TensorFlow, and OpenCV.

Amazon Redshift

- Amazon Redshift is a fast, scalable data warehouse that makes it simple and cost-effective to analyze all your data across your data warehouse and data lake.
- Redshift delivers ten times faster performance than other data warehouses by using machine learning, massively parallel query execution, and columnar storage on high-performance disk.
- You can setup and deploy a new data warehouse in minutes, and run queries across petabytes of data in your Redshift data warehouse, and exabytes of data in your data lake built on Amazon S3.
- You can start small for just $0.25 per hour and scale to $250 per terabyte per year, less than one-tenth the cost of other solutions.

Amazon QuickSight

- Amazon QuickSight is a fast, cloud-powered business intelligence (BI) service that makes it easy for you to deliver insights to everyone in your organization.
- QuickSight lets you create and publish interactive dashboards that can be accessed from browsers or mobile devices.
- You can embed dashboards into your applications, providing your customers with powerful self-service analytics.
- QuickSight easily scales to tens of thousands of users without any software to install, servers to deploy, or infrastructure to manage.

AWS Data Pipeline

- AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premises data sources, at specified intervals.
- With AWS Data Pipeline, you can regularly access your data where it's stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon RDS, Amazon DynamoDB, and Amazon EMR.

- AWS Data Pipeline helps you easily create complex data processing workloads that are fault tolerant, repeatable, and highly available.
- You don't have to worry about ensuring resource availability, managing inter-task dependencies, retrying transient failures or timeouts in individual tasks, or creating a failure notification system.
- AWS Data Pipeline also allows you to move and process data that was previously locked up in on-premises data silos.

AWS Glue

- AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics.
- You can create and run an ETL job with a few clicks in the AWS Management Console.
- You simply point AWS Glue to your data stored on AWS, and AWS Glue discovers your data and stores the associated metadata (e.g. table definition and schema) in the AWS Glue Data Catalog.
- Once cataloged, your data is immediately searchable, queryable, and available for ETL.

AWS Lake Formation

- AWS Lake Formation is a service that makes it easy to set up a secure data lake in days.
- A data lake is a centralized, curated, and secured repository that stores all your data, both in its original form and prepared for analysis.
- A data lake enables you to break down data silos and combine different types of analytics to gain insights and guide better business decisions.
- However, setting up and managing data lakes today involves a lot of manual, complicated, and time-consuming tasks.
- This work includes loading data from diverse sources, monitoring those data flows, setting up partitions, turning on encryption and managing keys, defining transformation jobs and monitoring their operation, re-organizing data into a columnar format, configuring access control settings, deduplicating redundant data, matching linked records, granting access to data sets, and auditing access over time.
- Creating a data lake with Lake Formation is as simple as defining where your data resides and what data access and security policies you want to apply.
- Lake Formation then collects and catalogs data from databases and object storage, moves the data into your new Amazon S3 data lake, cleans and classifies data using machine learning algorithms, and secures access to your sensitive data.
- Your users can then access a centralized catalog of data which describes available data sets and their appropriate usage.
- Your users then leverage these data sets with their choice of analytics and machine learning services, like Amazon EMR for Apache Spark, Amazon Redshift, Amazon Athena, Amazon SageMaker, and Amazon QuickSight.

Amazon Managed Streaming for Kafka (MSK)

- Amazon Managed Streaming for Kafka (MSK) is a fully managed service that makes it easy for you to build and run applications that use Apache Kafka to process streaming data.
- Apache Kafka is an open-source platform for building real-time streaming data pipelines and applications.
- With Amazon MSK, you can use Apache Kafka APIs to populate data lakes, stream changes to and from databases, and power machine learning and analytics applications.
- Apache Kafka clusters are challenging to setup, scale, and manage in production.

- When you run Apache Kafka on your own, you need to provision servers, configure Apache Kafka manually, replace servers when they fail, orchestrate server patches and upgrades, architect the cluster for high availability, ensure data is durably stored and secured, setup monitoring and alarms, and carefully plan scaling events to support load changes.
- Amazon Managed Streaming for Kafka makes it easy for you to build and run production applications on Apache Kafka without needing Apache Kafka infrastructure management expertise.
- That means you spend less time managing infrastructure and more time building applications.
- With a few clicks in the Amazon MSK console you can create highly available Apache Kafka clusters with settings and configuration based on Apache Kafka's deployment best practices.
- Amazon MSK automatically provisions and runs your Apache Kafka clusters.
- Amazon MSK continuously monitors cluster health and automatically replaces unhealthy nodes with no downtime to your application.
- In addition, Amazon MSK secures your Apache Kafka cluster by encrypting data at rest.

Application Services

Tracking Software Licenses with AWS Service Catalog and AWS Step Functions

- Enterprises have many business requirements for tracking how software product licenses are used in their organization for financial, governance, and compliance reasons.
- By tracking license usage, organizations can stay within budget, track expenditures, and avoid unplanned true-up bills from their vendors' true-up processes.
- The goal is to track the usage licenses as resources are deployed.
- In this post, you learn how to use AWS Service Catalog to deploy services and applications while tracking the licenses being consumed by end users, and how to prevent license overruns on AWS.
- This solution uses the following AWS services. Most of the resources are set up for you with an AWS CloudFormation stack:
  - AWS Service Catalog
  - AWS Lambda
  - AWS Step Functions
  - AWS CloudFormation
  - Amazon DynamoDB
  - Amazon SES

Secure Serverless Development Using AWS Service Catalog

- Serverless computing allows you to build and run applications and services without having to manage servers.
- AWS Service Catalog allows you to create and manage catalogs of services that are approved for use on AWS.
- Combining Serverless and Service Catalog together is a great way to safely allow developers to create products and services in the cloud.
- In this post, I demonstrate how to combine the controls of Service Catalog with AWS Lambda and Amazon API Gateway and allow your developers to build a Serverless application without full AWS access.

How to secure infrequently used EC2 instances with AWS Systems Manager

- Many organizations have predictable spikes in the usage of their applications and services.
- For example, retailers see large spikes in usage during Black Friday or Cyber Monday.

- The beauty of Amazon Elastic Compute Cloud (Amazon EC2) is that it allows customers to quickly scale up their compute power to meet these demands.
- However, some customers might require more time-consuming setup for their software running on EC2 instances.
- Instead of creating and terminating instances to meet demand, these customers turn off instances and then turn them on again when they are needed.
- Eventually the patches on those instances become out of date, and they require updates.

How Cloudticity Automates Security Patches for Linux and Windows using Amazon EC2 Systems Manager and AWS Step Functions

- As a provider of HIPAA-compliant solutions using AWS, Cloudticity always has security as the base of everything we do.
- HIPAA breaches would be an end-of-life event for most of our customers.
- Having been born in the cloud with automation in our DNA, Cloudticity embeds automation into all levels of infrastructure management including security, monitoring, and continuous compliance.
- As mandated by the HIPAA Security Rule (45 CFR Part 160 and Subparts A and C of Part 164), patches at the operating system and application level are required to prevent security vulnerabilities.
- As a result, patches are a major component of infrastructure management.
- Cloudticity strives to provide consistent and reliable services to all of our customers.
- As such, we needed to create a custom patching solution that supports both Linux and Windows.
- The minimum requirements for such a solution were to read from a manifest file that contains instance names and a list of knowledge base articles (KBs) or security packages to apply to each instance.
- Below is a simplified, high-level process overview.

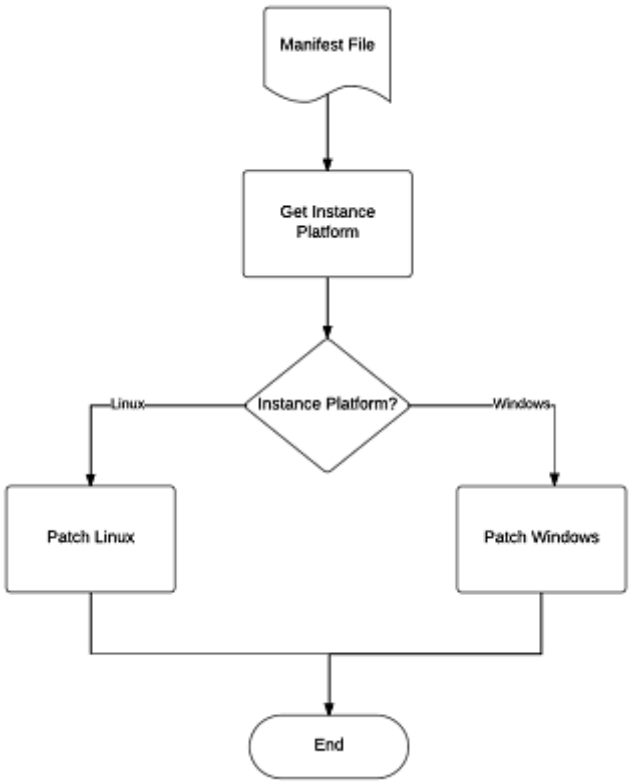


Fig. : High-Level Process Overview

- There were a few guidelines to be considered when designing the solution:
  - Each customer has a defined maintenance window that patches can be completed within. As such, the solution must be able to perform the updates within the specified maintenance window.
  - The solution must be able to provide patches to one or many instances and finish within the maintenance window.
  - The solution should use as many AWS services as possible to reduce time-to-market and take advantage of the built-in scaling that many AWS services provide.
  - Code reusability is essential.

Cloud Security

- A number of security threats are associated with cloud data services: not only traditional security threats, such as network eavesdropping, illegal invasion, and denial of service attacks, but also specific cloud computing threats, such as side channel attacks, virtualization vulnerabilities, and abuse of cloud services.
- The following security requirements limit the threats if we achieve that requirement than we can say our data is safe on cloud.
- **Identity management**
  - Every enterprise will have its own identity management system to control access to information and computing resources.
  - Cloud providers either integrate the customer's identity management system into their own infrastructure, using federation or SSO technology, or a biometric-based identification system, or provide an identity management system of their own.
  - CloudID, for instance, provides privacy-preserving cloud-based and cross-enterprise biometric identification.
  - It links the confidential information of the users to their biometrics and stores it in an encrypted fashion.
  - Making use of a searchable encryption technique, biometric identification is performed in encrypted domain to make sure that the cloud provider or potential attackers do not gain access to any sensitive data or even the contents of the individual queries.
- **Physical security**
  - Cloud service providers physically secure the IT hardware (servers, routers, cables etc.) against unauthorized access, interference, theft, fires, floods etc. and ensure that essential supplies (such as electricity) are sufficiently robust to minimize the possibility of disruption.
  - This is normally achieved by serving cloud applications from 'world-class' (i.e. professionally specified, designed, constructed, managed, monitored and maintained) data centers.
- **Personnel security**
  - Various information security concerns relating to the IT and other professionals associated with cloud services are typically handled through pre-, para- and post-employment activities such as security screening potential recruits, security awareness and training programs, proactive.
- **Privacy**
  - Providers ensure that all critical data (credit card numbers, for example) are masked or encrypted and that only authorized users have access to data in its entirety. Moreover, digital identities and credentials must be protected as should any data that the provider collects or produces about customer activity in the cloud.
- **Confidentiality**
  - Data confidentiality is the property that data contents are not made available or disclosed to illegal users.

- Outsourced data is stored in a cloud and out of the owners' direct control. Only authorized users can access the sensitive data while others, including CSPs, should not gain any information of the data.
- Meanwhile, data owners expect to fully utilize cloud data services, e.g., data search, data computation, and data sharing, without the leakage of the data contents to CSPs or other adversaries.

- **Access controllability**
  - Access controllability means that a data owner can perform the selective restriction of access to her or his data outsourced to cloud.
  - Legal users can be authorized by the owner to access the data, while others cannot access it without permissions.
  - Further, it is desirable to enforce fine-grained access control to the outsourced data, i.e., different users should be granted different access privileges with regard to different data pieces.
  - The access authorization must be controlled only by the owner in untrusted cloud environments.
- **Integrity**
  - Data integrity demands maintaining and assuring the accuracy and completeness of data.
  - A data owner always expects that her or his data in a cloud can be stored correctly and trustworthily.
  - It means that the data should not be illegally tampered, improperly modified, deliberately deleted, or maliciously fabricated.
  - If any undesirable operations corrupt or delete the data, the owner should be able to detect the corruption or loss.
  - Further, when a portion of the outsourced data is corrupted or lost, it can still be retrieved by the data users.

CloudWatch

- Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS.
- You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources.
- Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate.
- You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health.
- You can use these insights to react and keep your application running smoothly.

CloudFormation

- AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment.
- CloudFormation allows you to use a simple text file to model and provision, in an automated and secure manner, all the resources needed for your applications across all regions and accounts.
- This file serves as the single source of truth for your cloud environment.
- AWS CloudFormation is available at no additional charge, and you pay only for the AWS resources needed to run your applications.

Advantage of Cloud Formation

Model it all

- AWS CloudFormation allows you to model your entire infrastructure in a text file. This template becomes the single source of truth for your infrastructure. This helps you to standardize infrastructure components used across your organization, enabling configuration compliance and faster troubleshooting.

Automate and deploy

- AWS CloudFormation provisions your resources in a safe, repeatable manner, allowing you to build and rebuild your infrastructure and applications, without having to perform manual actions or write custom scripts. CloudFormation takes care of determining the right operations to perform when managing your stack, and rolls back changes automatically if errors are detected.

It's just code

- Codifying your infrastructure allows you to treat your infrastructure as just code. You can author it with any code editor, check it into a version control system, and review the files with team members before deploying into production.

CloudTrail

- AWS CloudTrail is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account.
- Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail.
- Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.
- CloudTrail is enabled on your AWS account when you create it.
- When activity occurs in your AWS account, that activity is recorded in a CloudTrail event.
- You can easily view recent events in the CloudTrail console by going to Event history.
- For an ongoing record of activity and events in your AWS account, create a trail.
- Visibility into your AWS account activity is a key aspect of security and operational best practices.
- You can use CloudTrail to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure.
- You can identify who or what took which action, what resources were acted upon, when the event occurred, and other details to help you analyze and respond to activity in your AWS account.
- Optionally, you can enable AWS CloudTrail Insights on a trail to help you identify and respond to unusual activity.
- You can integrate CloudTrail into applications using the API, automate trail creation for your organization, check the status of trails you create, and control how users view CloudTrail events.

Working of CloudTrail

- You can create two types of trails for an AWS account:

  A trail that applies to all regions
  - When you create a trail that applies to all regions, CloudTrail records events in each region and delivers the CloudTrail event log files to an S3 bucket that you specify.
  - If a region is added after you create a trail that applies to all regions that new region is automatically included, and events in that region are logged.
  - This is the default option when you create a trail in the CloudTrail console.

A trail that applies to one region

- When you create a trail that applies to one region, CloudTrail records the events in that region only.
- It then delivers the CloudTrail event log files to an Amazon S3 bucket that you specify.
- If you create additional single trails, you can have those trails deliver CloudTrail event log files to the same Amazon S3 bucket or to separate buckets.
- This is the default option when you create a trail using the AWS CLI or the CloudTrail API.

- Beginning on April 12, 2019, trails will be viewable only in the AWS Regions where they log events.
- If you create a trail that logs events in all AWS Regions, it will appear in the console in all AWS Regions.
- If you create a trail that only logs events in a single AWS Region, you can view and manage it only in that AWS Region.
- If you have created an organization in AWS Organizations, you can also create a trail that will log all events for all AWS accounts in that organization.
- This is referred to as an organization trail. Organization trails can apply to all AWS Regions or one Region.
- Organization trails must be created in the master account, and when specified as applying to an organization, are automatically applied to all member accounts in the organization.
- Member accounts will be able to see the organization trail, but cannot modify or delete it.
- By default, member accounts will not have access to the log files for the organization trail in the Amazon S3 bucket.
- You can change the configuration of a trail after you create it, including whether it logs events in one region or all regions.
- You can also change whether it logs data or CloudTrail Insights events.
- Changing whether a trail logs events in one region or in all regions affects which events are logged.
- By default, CloudTrail event log files are encrypted using Amazon S3 server-side encryption (SSE).
- You can also choose to encrypt your log files with an AWS Key Management Service (AWS KMS) key.
- You can store your log files in your bucket for as long as you want.
- You can also define Amazon S3 lifecycle rules to archive or delete log files automatically.
- If you want notifications about log file delivery and validation, you can set up Amazon SNS notifications.
- CloudTrail typically delivers log files within 15 minutes of account activity.
- In addition, CloudTrail publishes log files multiple times an hour, about every five minutes.
- These log files contain API calls from services in the account that support CloudTrail.

Benefits of CloudTrail

Simplified compliance

- With AWS CloudTrail, simplify your compliance audits by automatically recording and storing event logs for actions made within your AWS account.
- Integration with Amazon CloudWatch Logs provides a convenient way to search through log data, identify out-of-compliance events, accelerate incident investigations, and expedite responses to auditor requests.

Security analysis and troubleshooting

- With AWS CloudTrail, you can discover and troubleshoot security and operational issues by capturing a comprehensive history of changes that occurred in your AWS account within a specified period of time.

Visibility into user and resource activity

- AWS CloudTrail increases visibility into your user and resource activity by recording AWS Management Console actions and API calls.

- You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.

Security automation

- AWS CloudTrail allows you track and automatically respond to account activity threatening the security of your AWS resources.
- With Amazon CloudWatch Events integration, you can define workflows that execute when events that can result in security vulnerabilities are detected.
- For example, you can create a workflow to add a specific policy to an Amazon S3 bucket when CloudTrail logs an API call that makes that bucket public.

OpsWorks

- AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet.
- Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers.
- OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your Amazon EC2 instances or on-premises compute environments.
- OpsWorks has three offerings, AWS Opsworks for Chef Automate, AWS OpsWorks for Puppet Enterprise, and AWS OpsWorks Stacks.

AWS OpsWorks for Chef Automate

- AWS OpsWorks for Chef Automate is a fully managed configuration management service that hosts Chef Automate, a suite of automation tools from Chef for configuration management, compliance and security, and continuous deployment.
- OpsWorks also maintains your Chef server by automatically patching, updating, and backing up your server.
- OpsWorks eliminates the need to operate your own configuration management systems or worry about maintaining its infrastructure.
- OpsWorks gives you access to all of the Chef Automate features, such as configuration and compliance management, which you manage through the Chef Console or command line tools like Knife.
- It also works seamlessly with your existing Chef cookbooks.
- Choose AWS OpsWorks for Chef Automate if you are an existing Chef user.

AWS OpsWorks for Puppet Enterprise

- AWS OpsWorks for Puppet Enterprise is a fully managed configuration management service that hosts Puppet Enterprise, a set of automation tools from Puppet for infrastructure and application management.
- OpsWorks also maintains your Puppet master server by automatically patching, updating, and backing up your server.
- OpsWorks eliminates the need to operate your own configuration management systems or worry about maintaining its infrastructure.
- OpsWorks gives you access to all of the Puppet Enterprise features, which you manage through the Puppet console.
- It also works seamlessly with your existing Puppet code.
- Choose AWS OpsWorks for Puppet Enterprise if you are an existing Puppet user.

AWS OpsWorks Stacks

- AWS OpsWorks Stacks is an application and server management service. With OpsWorks Stacks, you can model your application as a stack containing different layers, such as load balancing, database, and application server.
- Within each layer, you can provision Amazon EC2 instances, enable automatic scaling, and configure your instances with Chef recipes using Chef Solo.
- This allows you to automate tasks such as installing packages and programming languages or frameworks, configuring software, and more.
- Choose AWS OpsWorks Stacks if you need a solution for application modeling and management.

OpenID Connect (OIDC)

- IAM OIDC identity providers are entities in IAM that describe an external identity provider (IdP) service that supports the OpenID Connect (OIDC) standard, such as Google or Salesforce.
- You use an IAM OIDC identity provider when you want to establish trust between an OIDC-compatible IdP and your AWS account.
- This is useful when creating a mobile app or web application that requires access to AWS resources, but you don't want to create custom sign-in code or manage your own user identities.
- You can create and manage an IAM OIDC identity provider using the AWS Management Console, the AWS Command Line Interface, the Tools for Windows PowerShell, or the IAM API.
- When you create an OpenID Connect (OIDC) identity provider in IAM, you must supply a thumbprint.
- IAM requires the thumbprint for the root certificate authority (CA) that signed the certificate used by the external identity provider (IdP).
- The thumbprint is a signature for the CA's certificate that was used to issue the certificate for the OIDC-compatible IdP.
- When you create an IAM OIDC identity provider, you are trusting identities authenticated by that IdP to have access to your AWS account.
- By supplying the CA's certificate thumbprint, you trust any certificate issued by that CA with the same DNS name as the one registered.
- This eliminates the need to update trusts in each account when you renew the IdP's signing certificate.
- You can create an IAM OIDC identity provider with the AWS Command Line Interface, the Tools for Windows PowerShell, or the IAM API.
- When you use these methods, you must obtain the thumbprint manually and supply it to AWS.
- When you create an OIDC identity provider with the IAM console, the console attempts to fetch the thumbprint for you.
- We recommend that you also obtain the thumbprint for your OIDC IdP manually and verify that the console fetched the correct thumbprint.

Managing Costs, Utilization and Tracking

- The cloud allows you to trade capital expenses (such as data centers and physical servers) for variable expenses, and only pay for IT as you consume it.
- And, because of the economies of scale, the variable expenses are much lower than what you would pay to do it yourself.
- Whether you were born in the cloud, or you are just starting your migration journey to the cloud, AWS has a set of solutions to help you manage and optimize your spend.
- During this unprecedented time, many businesses and organizations are facing disruption to their operations, budgets, and revenue.
- AWS has a set of solutions to help you with cost management and optimization.
- This includes services, tools, and resources to organize and track cost and usage data, enhance control through consolidated billing and access permission, enable better planning through budgeting and forecasts, and further lower cost with resources and pricing optimizations.

AWS Cost Management Solutions

Organize and Report Cost and Usage Based on User-Defined Methods

- You need complete, near real-time visibility of your cost and usage information to make informed decisions.
- AWS equips you with tools to organize your resources based on your needs, visualize and analyze cost and usage data in a single pane of glass, and accurately chargeback to appropriate entities (e.g. department, project, and product).
- Rather than centrally policing the cost, you can provide real-time cost data that makes sense to your engineering, application, and business teams.
- The detailed, allocable cost data allows teams to have the visibility and details to be accountable of their own spend.

Billing with Built-in Control

- Business and organization leaders need a simple and easy way to access AWS billing information, including a spend summary, a breakdown of all service costs incurred by accounts across the organization, along with discounts and credits.
- Customer can choose to consolidate your bills and take advantage of higher volume discounts based on aggregated usage across your bills.
- Leaders also need to set appropriate guardrails in place so you can maintain control over cost, governance, and security.
- AWS helps organizations balance freedom and control by enabling the governance of granular user permission.

Improved Planning with Flexible Forecasting and Budgeting

- Businesses and organizations need to plan and set expectations around cloud costs for your projects, applications, and more.
- The emergence of the cloud allowed teams to acquire and deprecate resources on an ongoing basis, without relying on teams to approve, procure and install infrastructure.
- However, this flexibility requires organizations to adapt to the new, dynamic forecasting and budgeting process.

- AWS provides forecasts based on your cost and usage history and allows you to set budget threshold and alerts, so you can stay informed whenever cost and usage is forecasted to, or exceeds the threshold limit.
- You can also set reservation utilization and/or coverage targets for your Reserved Instances and Savings Plans and monitor how they are progressing towards your target.

Optimize Costs with Resource and Pricing Recommendations

- With AWS, customers can take control of your cost and continuously optimize your spend.
- There are a variety of AWS pricing models and resources you can choose from to meet requirements for both performance and cost efficiency, and adjust as needed.
- When evaluating AWS services for your architectural and business needs, you will have the flexibility to choose from a variety of elements, such as operating systems, instance types, availability zones, and purchase options.
- AWS offers resources optimization recommendations to simplify the evaluation process so you can efficiently select the cost-optimized resources.
- We also provide recommendations around pricing models (up to 72% with Reserved Instances and Savings Plans and up to 90% with Spot Instances) based on your utilization patterns, so you can further drive down your cost without compromising workload performance.

Monitor, Track, and Analyze Your AWS Costs & Usage

- Appropriate management, tracking and measurement are fundamental in achieving the full benefits of cost optimization.

Amazon CloudWatch

- Amazon CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers.

AWS Trusted Advisor

- AWS Trusted Advisor is an online tool that provides you real time guidance to help you provision your resources following AWS best practices.

AWS Cost Explorer

- AWS Cost Explorer has an easy-to-use interface that lets you visualize, understand, and manage your AWS costs and usage over time.

Bottom Line Impact

- As AWS provide large range of service and we can utilize it for our business on pay as you go basis so it will save our cost and time.
- Due to that company can reduce their cost and increase revenue by focusing on core work and other service management is done by cloud providers.
- It will create bottom line impact for organization.

Geographic Concerns

- The AWS Global Cloud Infrastructure is the most secure, extensive, and reliable cloud platform, offering over 175 fully featured services from data centers globally.

- Whether you need to deploy your application workloads across the globe in a single click, or you want to build and deploy specific applications closer to your end-users with single-digit millisecond latency, AWS provides you the cloud infrastructure where and when you need it.
- With millions of active customers and tens of thousands of partners globally, AWS has the largest and most dynamic ecosystem.
- Customers across virtually every industry and of every size, including start-ups, enterprises, and public sector organizations, are running every imaginable use case on AWS.

Failure plans / Disaster Recovery (DR)

- Our data is the most precious asset that we have and protecting it is our top priority.
- Creating backups of our data to an off shore data center, so that in the event of an on premise failure we can switch over to our backup, is a prime focus for business continuity.
- As AWS says, 'Disaster recovery is a continual process of analysis and improvement, as business and systems evolve. For each business service, customers need to establish an acceptable recovery point and time, and then build an appropriate DR solution.'
- Backup and DR on Cloud reduces costs by half as compared to maintaining your own redundant data centers. And if you think about it, it's really not that surprising.
- Imagine the kind of cost you would entail in buying and maintaining servers and data centers, providing secure and stable connectivity and not to mention keeping them secure.
- You would also be underutilizing severs; and in times of unpredictable traffic rise it would be strenuous to set up new ones. To all these cloud provides a seamless transition reducing cost dramatically.

4 Standard Approaches of Backup and Disaster Recovery Using Amazon Cloud

1. Backup and Recovery

- To recover your data in the event of any disaster, you must first have your data periodically backed up from your system to AWS.
- Backing up of data can be done through various mechanisms and your choice will be based on the RPO (Recovery Point Objective- So if your disaster struck at 2 pm and your RPO is 1 hr, your Backup & DR will restore all data till 1 pm.) that will suit your business needs.
- AWS offers AWS Direct connect and Import Export services that allow for faster backup.
- For example, if you have a frequently changing database like say a stock market, then you will need a very high RPO. However if your data is mostly static with a low frequency of changes, you can opt for periodic incremental backup.
- Once your backup mechanisms are activated you can pre-configure AMIs (operating systems & application software).
- Now when a disaster strikes, EC2 (Elastic Compute Capacity) instances in the Cloud using EBS (Elastic Block Store) coupled with AMIs can access your data from the S3 (Simple Storage Service) buckets to revive your system and keep it going.

2. Pilot Light Approach

- The name pilot light comes from the gas heater analogy. Just as in a heater you have a small flame that is always on, and can quickly ignite the entire furnace; a similar approach can be thought of about your data system.
- In the preparatory phase your on premise database server mirrors data to data volumes on AWS. The database server on cloud is always activated for frequent or continuous incremental backup.

- This core area is the pilot from our gas heater analogy. The application and caching server replica environments are created on cloud and kept in standby mode as very few changes take place over time.
- These AMIs can be updated periodically. This is the entire furnace from our example. If the on premise system fails, then the application and caching servers get activated; further users are rerouted using elastic IP addresses to the ad hoc environment on cloud. Your Recovery takes just a few minutes.

3. Warm Standby Approach

- This Technique is the next level of the pilot light, reducing recovery time to almost zero.
- Your application and caching servers are set up and always activated based on your business critical activities but only a minimum sized fleet of EC2 instances are dedicated.
- The backup system is not capable of handling production load, but can be used for testing, quality assurance and other internal uses.
- In the event of a disaster, when your on premise data center fails, two things happen.
- Firstly multiple EC2 instances are dedicated (vertical and horizontal scaling) to bring your application and caching environment up to production load. ELB and Auto Scaling (for distributing traffic) are used to ease scaling up.
- Secondly using Amazon Route 53 user traffic is rerouted instantly using elastic IP addresses and there is instant recovery of your system with almost zero down time.

4. Multi-Site Approach

- Well this is the optimum technique in backup and DR and is the next step after warm standby.
- All activities in the preparatory stage are similar to a warm standby; except that AWS backup on Cloud is also used to handle some portions of the user traffic using Route 53.
- When a disaster strikes, the rest of the traffic that was pointing to the on premise servers are rerouted to AWS and using auto scaling techniques multiple EC2 instances are deployed to handle full production capacity.
- You can further increase the availability of your multi-site solution by designing Multi-AZ architectures.

Examining Logs

- It is necessary to examine the log files in order to locate an error code or other indication of the issue that your cluster experienced.
- It may take some investigative work to determine what happened.
- Hadoop runs the work of the jobs in task attempts on various nodes in the cluster.
- Amazon EMR can initiate speculative task attempts, terminating the other task attempts that do not complete first.
- This generates significant activity that is logged to the controller, stderr and syslog log files as it happens.
- In addition, multiple tasks attempts are running simultaneously, but a log file can only display results linearly.
- Start by checking the bootstrap action logs for errors or unexpected configuration changes during the launch of the cluster.
- From there, look in the step logs to identify Hadoop jobs launched as part of a step with errors.
- Examine the Hadoop job logs to identify the failed task attempts.
- The task attempt log will contain details about what caused a task attempt to fail.

Book

1. *Cloud Computing Bible, Barrie Sosinsky, John Wiley & Sons, ISBN-13: 978-0470903568.*
2. *Mastering AWS Security, Albert Anthony, Packt Publishing Ltd., ISBN 978-1-78829-372-3.*
3. *Amazon Web Services for Dummies, Bernard Golden, For Dummies, ISBN-13: 978- 1118571835.*

Websites

1. *www.aws.amazon.com*
2. *www.docs.aws.amazon.com*
3. *www.bluepiit.com*
4. *www.inforisktoday.com*
5. *www.techno-pulse.com*
6. *www.exelanz.com*
7. *www.ibm.com*
8. *www.iarjset.com/upload/2017/july-17/IARJSET%2018.pdf*
9. *www.searchservervirtualization.techtarget.com*
10. *www.docs.eucalyptus.com*
11. *www.cloudacademy.com*
12. *www.searchaws.techtarget.com*
13. *www.searchsecurity.techtarget.com*
14. *www.en.wikipedia.org/wiki/Cloud_computing_security*
15. *www.znetlive.com*
16. *www.en.wikipedia.org/wiki/Virtual_private_cloud*
17. *www.resource.onlinetech.com*
18. *www.globalknowledge.com*
19. *www.blog.blazeclan.com/4-approaches-backup-disaster-recovery-explained-amazon-cloud*
20. *www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-about-the-cloud*
21. *www.javatpoint.com/introduction-to-cloud-computing*
22. *www.javatpoint.com/history-of-cloud-computing*
23. *www.allcloud.io/blog/6-cloud-computing-concerns-facing-2018*
24. www.searchitchannel.techtarget.com/definition/cloud-marketplace
25. www.en.wikipedia.org/wiki/Amazon_Web_Services
26. www.msystechnologies.com/blog/cloud-orchestration-everything-you-want-to-know
27. www.linuxacademy.com/blog/linux-academy/elasticity-cloud-computing
28. www.searchitchannel.techtarget.com/definition/Eucalyptus
29. www.geeksforgeeks.org/virtualization-cloud-computing-types
30. www.cloudsearch.blogspot.com
31. www.simplilearn.com/tutorials/aws-tutorial/aws-iam
32. www.d1.awsstatic.com/whitepapers/aws-security-whitepaper.pdf
33. www.resources.intenseschool.com/amazon-aws-understanding-ec2-key-pairs-and-how-they-are-used-for-windows-and-linux-instances/
34. www.pagely.com/blog/amazon-ec2/
35. www.cloudflare.com/learning/cloud/what-is-multitenancy/
36. www.hevodata.com/blog/amazon-redshift-pros-and-cons/