

Hand Book

For

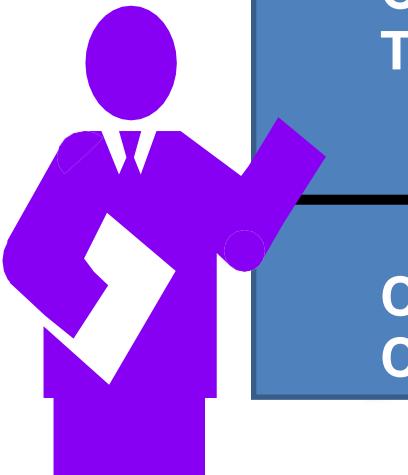
AWS Admin

By HARI



MODULE#1 -Introduction to Cloud Computing

Agenda



What is Cloud Computing ?
Why we go for Cloud Computing?
History and Origins of Cloud Computing
Characteristics of Cloud Computing
Types of cloud services
 Software as a Service SaaS
 Platform as a Service PaaS
 Infrastructure as a Service IaaS
Cloud implementation types
Conclusion

What is Cloud Service?

Services and Solutions that are delivered and consumed in real time over internet are Cloud Services

-When you store your photos online, use webmail or social networking site, you will use “Cloud Computing” Service

What is Cloud Computing?

Cloud Computing is a delivery model of Computing services over the internet

-It enables real time development, development and delivery of broad range of products, services and solutions

“Cloud computing is a style of computing where massively scalable IT-related capabilities are provided as a service across the Internet to multiple external customers”

What is Cloud Computing?

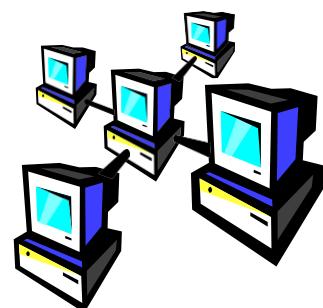
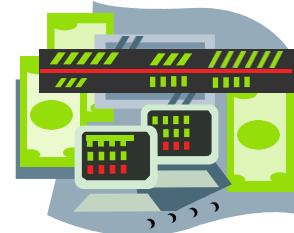


Computing and software resources that are delivered on demand, as service..

APPLICATIONS



STORAGE SERVICES



(DATABASE)

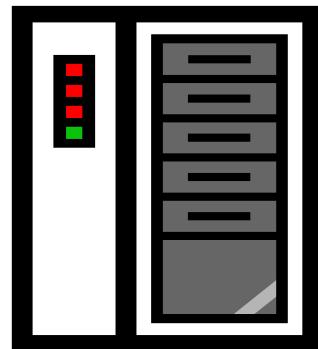
COMPUTER

AWS Administration

-Your Best Choice



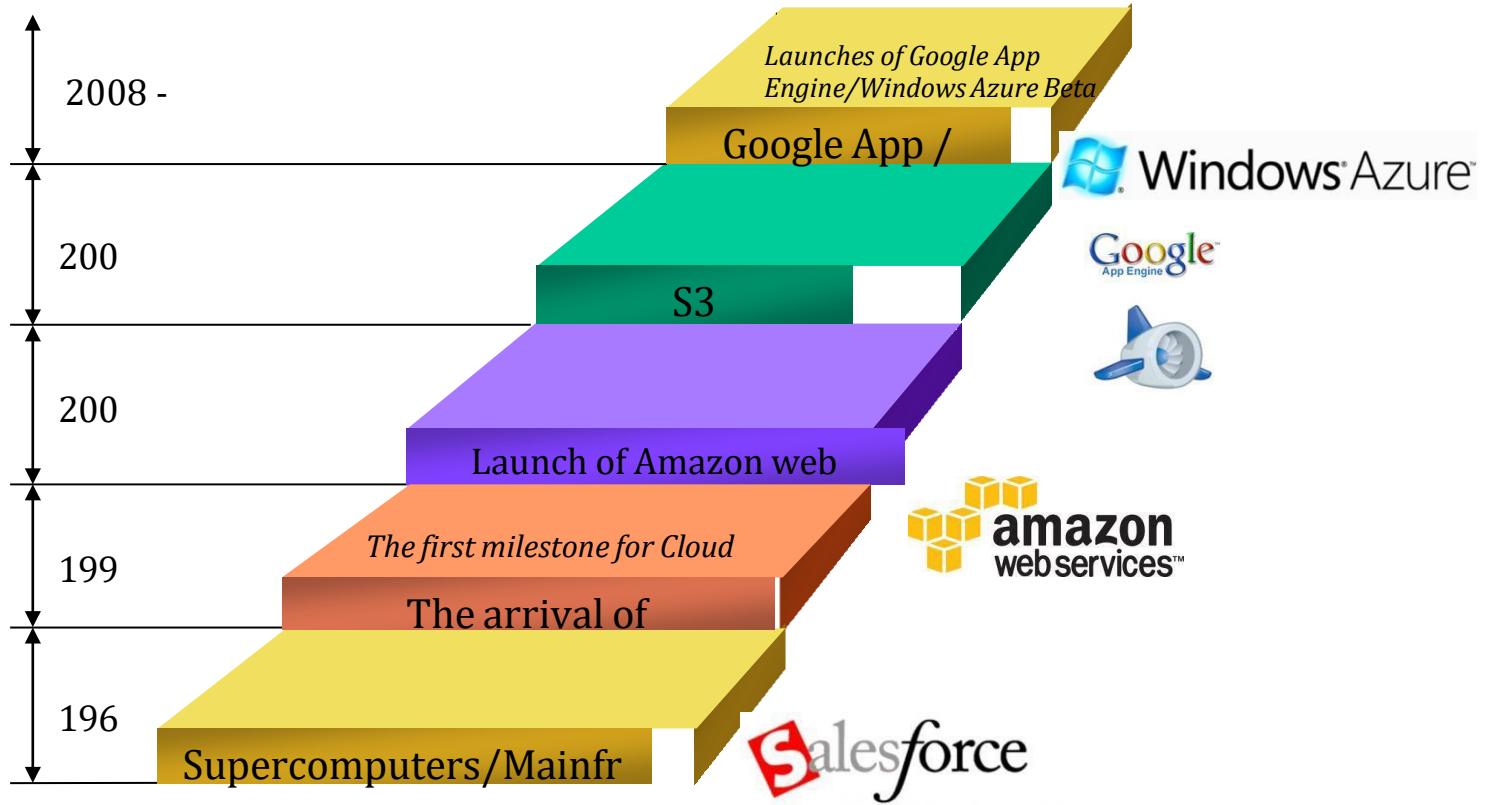
NETWORK



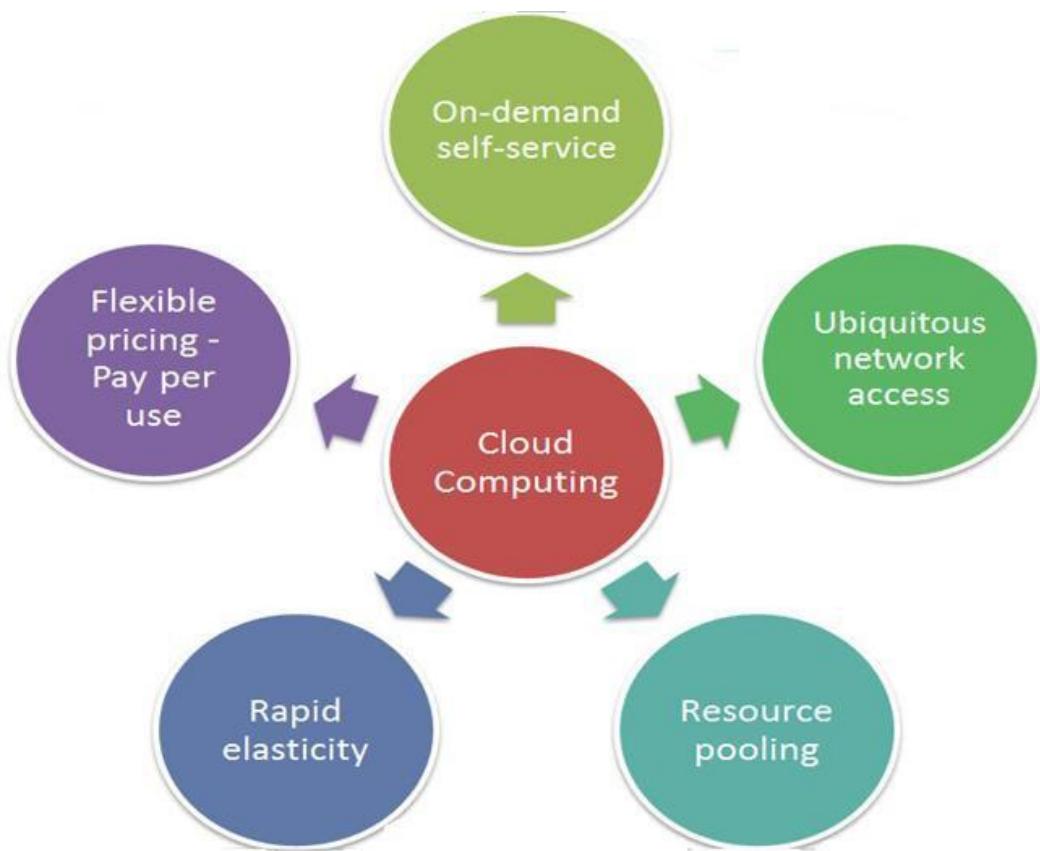
Why we go for Cloud Computing?

- Lower Computing Cost
- Improved Performance
- Reduced Software Cost
- Instant Software Updates
- Unlimited Storage Capacity
- Increased Data Reliability
- Device Independence and the “always on!, anywhere and any place”
- Free From Maintenance and the “no-need-to-know”

History and Origins of Cloud Computing



Characteristics of Cloud Computing



Characteristics of Cloud Computing

- On-demand self-service
- Ubiquitous network access
- Resource pooling (advanced virtualization)
- Rapid elasticity
- Flexible pricing - Pay per use



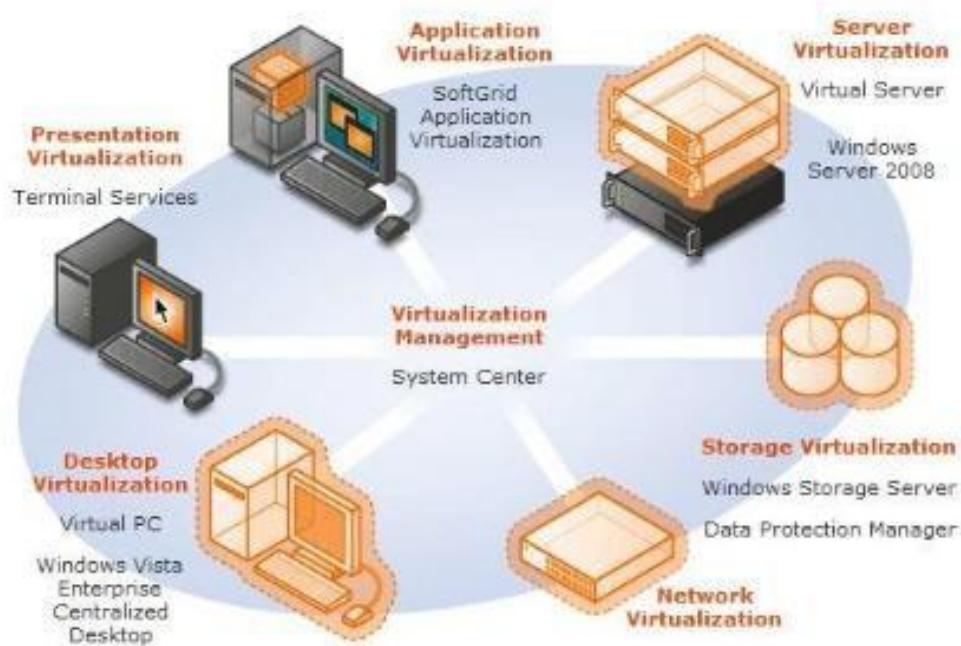
Characteristics of Cloud Computing

- On-demand self-service
- Ubiquitous network access
- Resource pooling (advanced virtualization)
- Rapid elasticity
- Flexible pricing - Pay per use



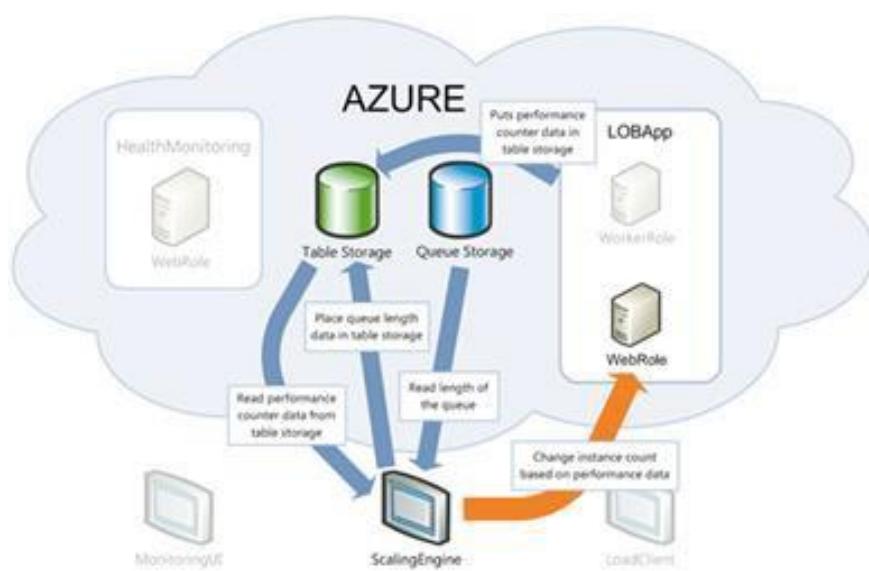
Characteristics of Cloud Computing

- On-demand self-service
- Ubiquitous network access
- Resource pooling (advanced virtualization)
- Rapid elasticity
- Flexible pricing - Pay per use



Characteristics of Cloud Computing

- On-demand self-service
- Ubiquitous network access
- Resource pooling (advanced virtualization)
- Rapid elasticity
- Flexible pricing - Pay per use

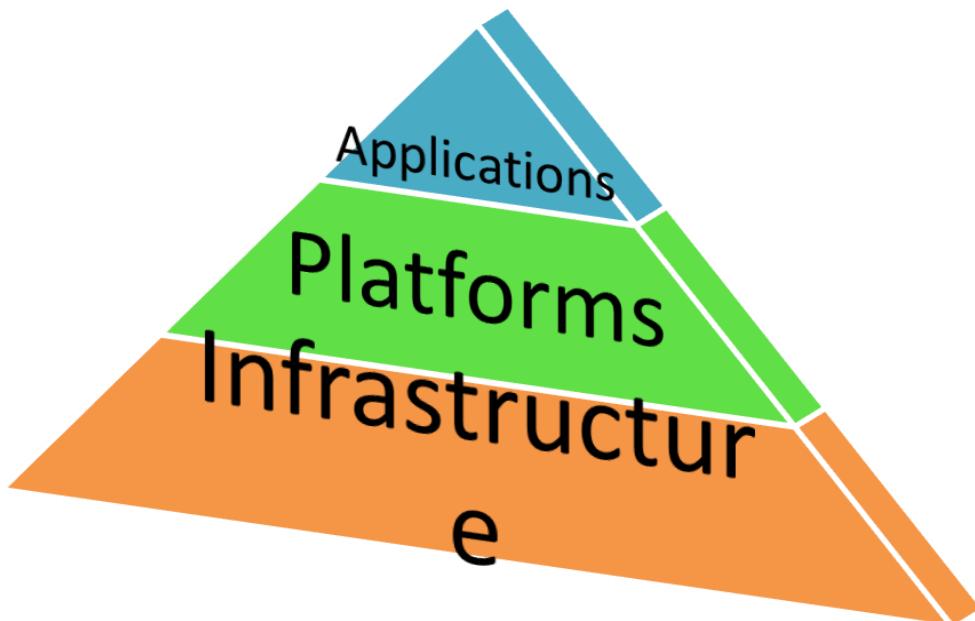


Characteristics of Cloud Computing

- On-demand self-service
- Ubiquitous network access
- Resource pooling (advanced virtualization)
- Rapid elasticity
- Flexible pricing - Pay per use



Cloud Service Layers



Cloud Service Layers

Software as a Service (SaaS)

- SaaS is a software delivery methodology that provides licensed multi-tenant access to software and its functions remotely as a Web-based service.

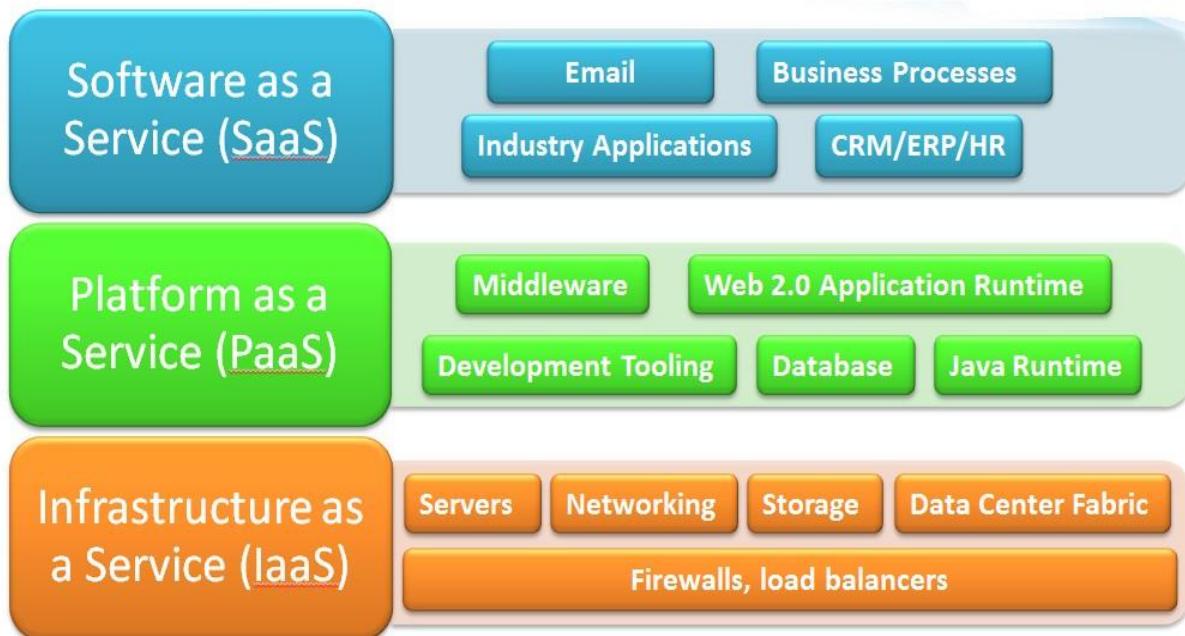
Platform as a Service (PaaS)

- PaaS provides all of the facilities required to support the complete life cycle of building and delivering web applications and services entirely from the Internet.

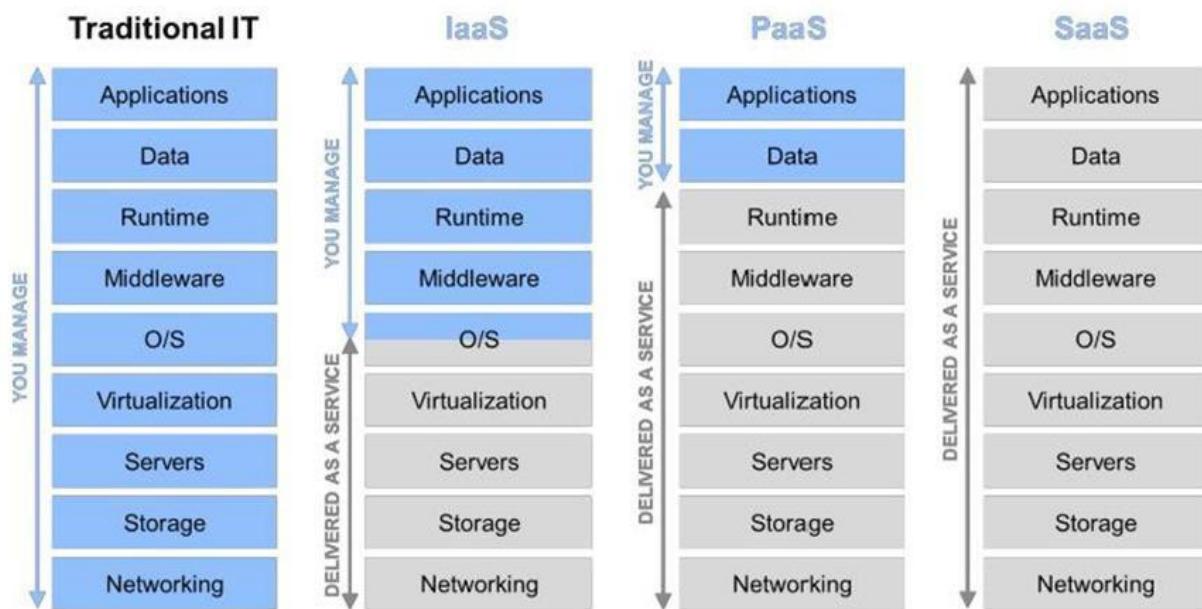
Infrastructure as a Service (IaaS)

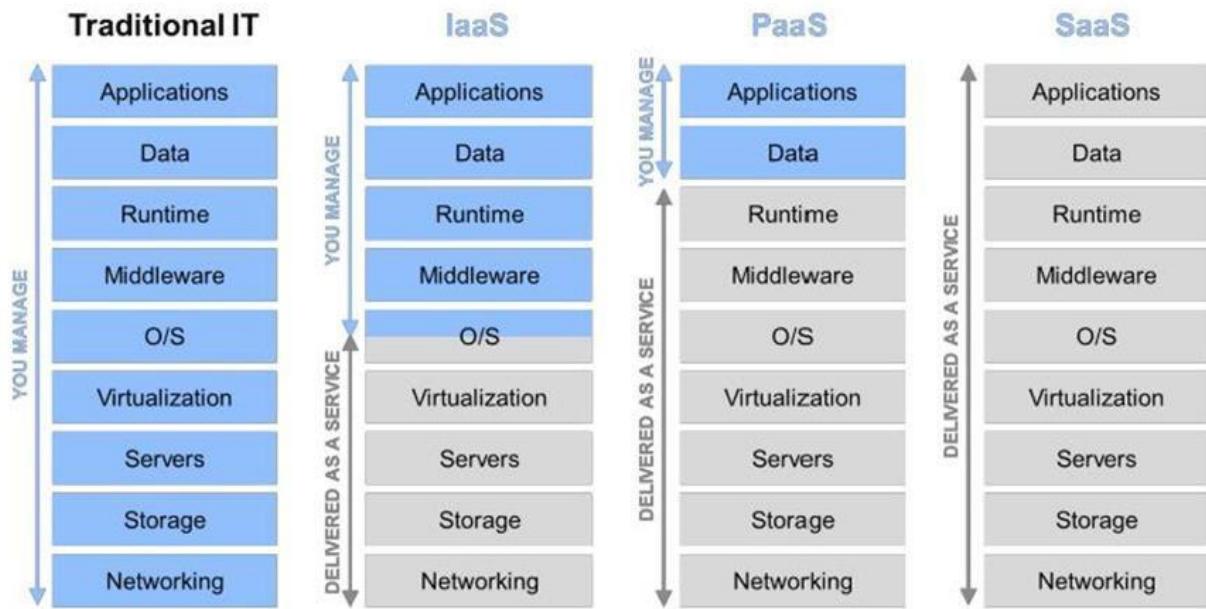
- IaaS is the delivery of technology infrastructure as an on demand scalable service.

Cloud Service Layers

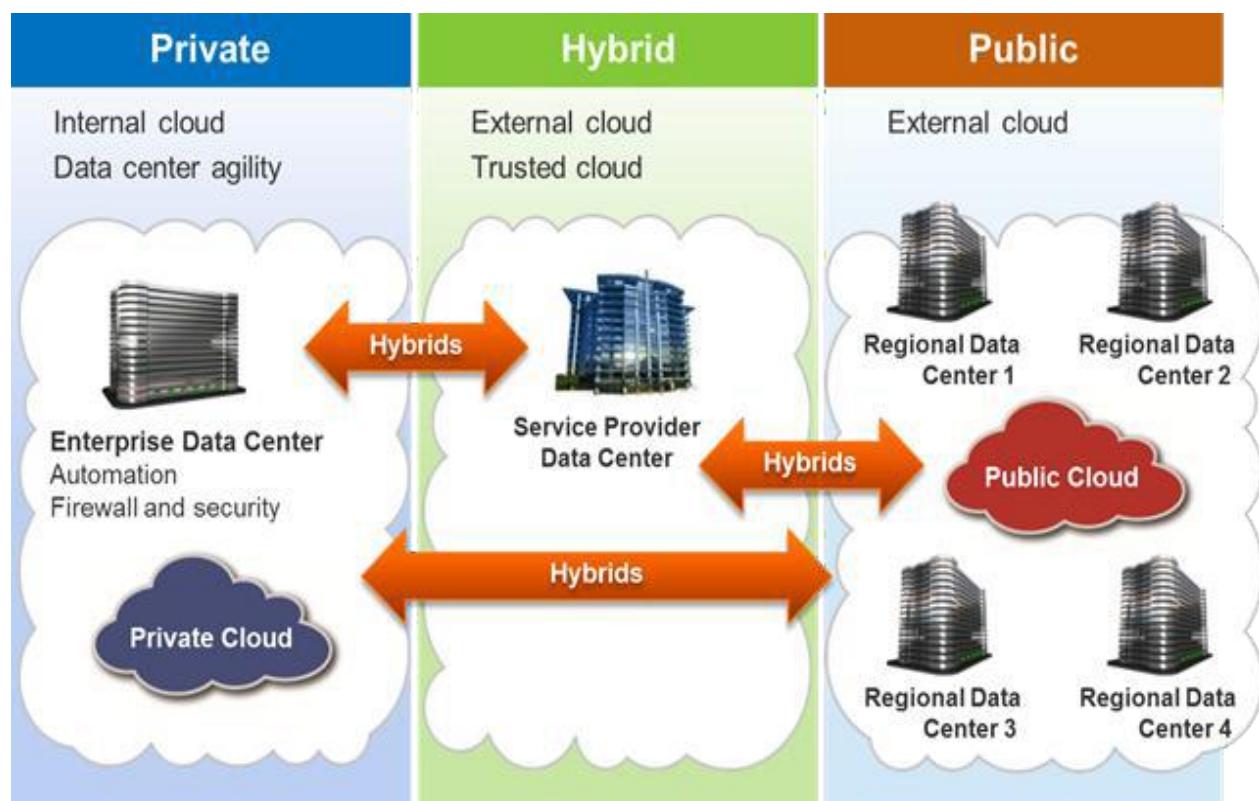


Cloud Service Model -Comparison



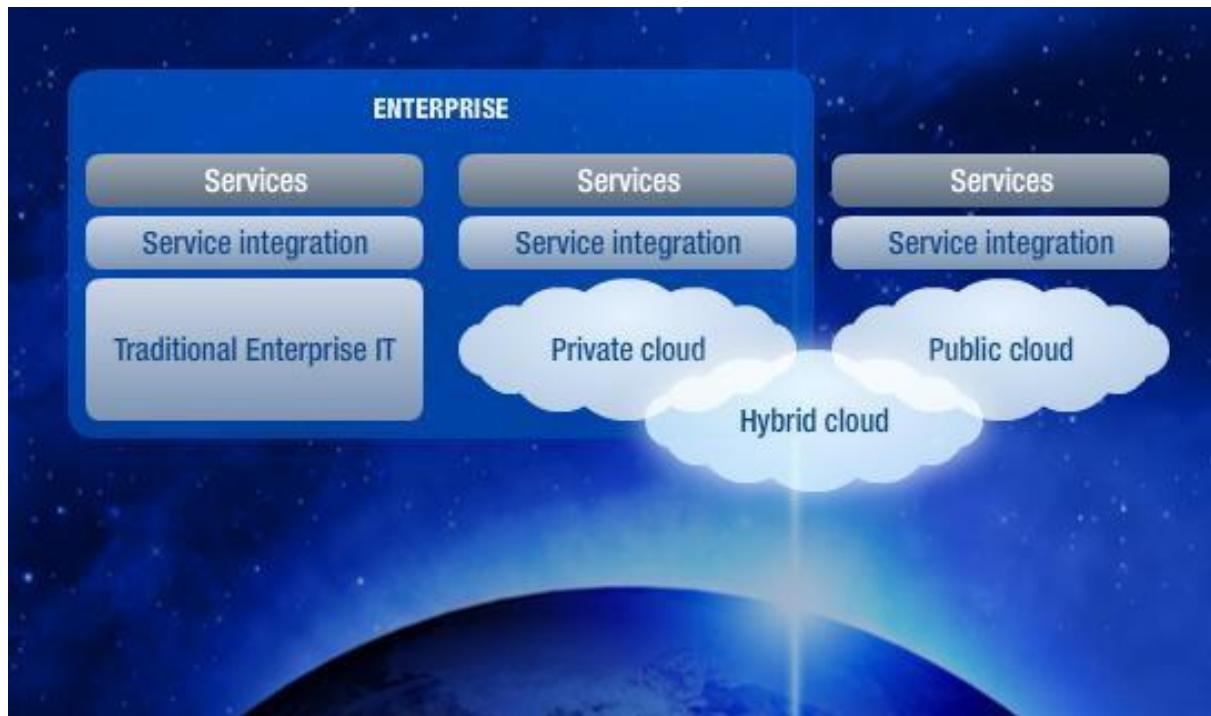


Cloud implementation types



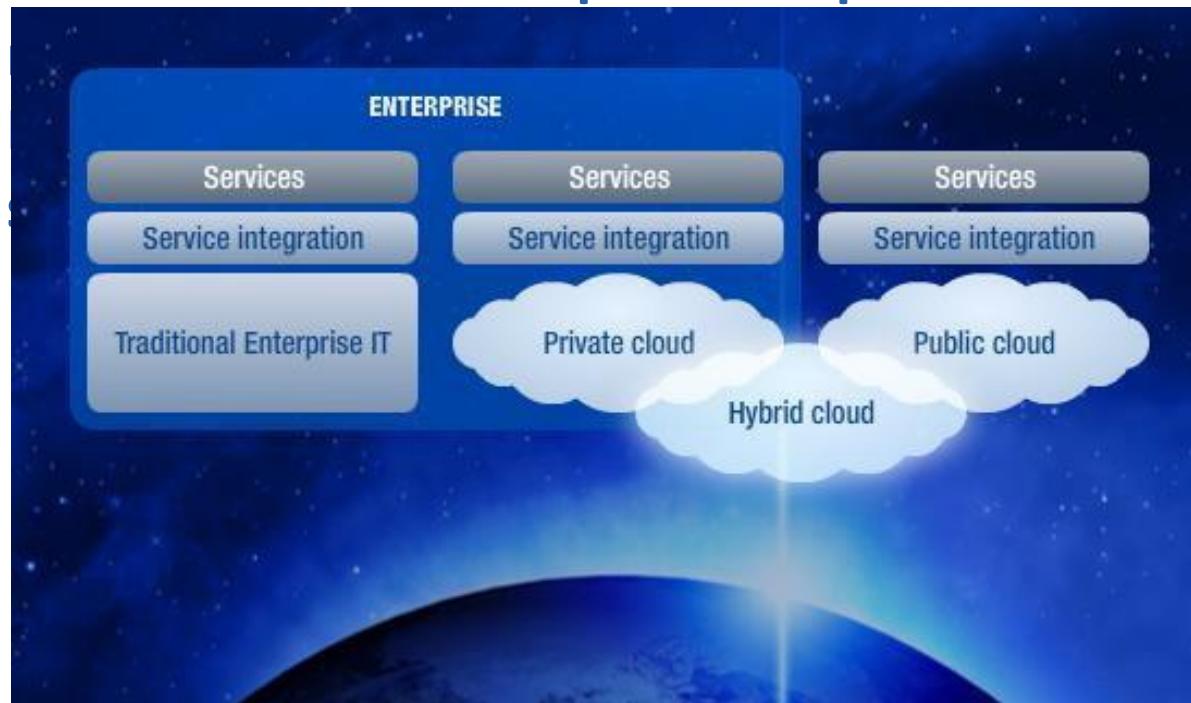
Public Cloud

- Owned and managed by service provider
- Made available to the general public or a large industry group



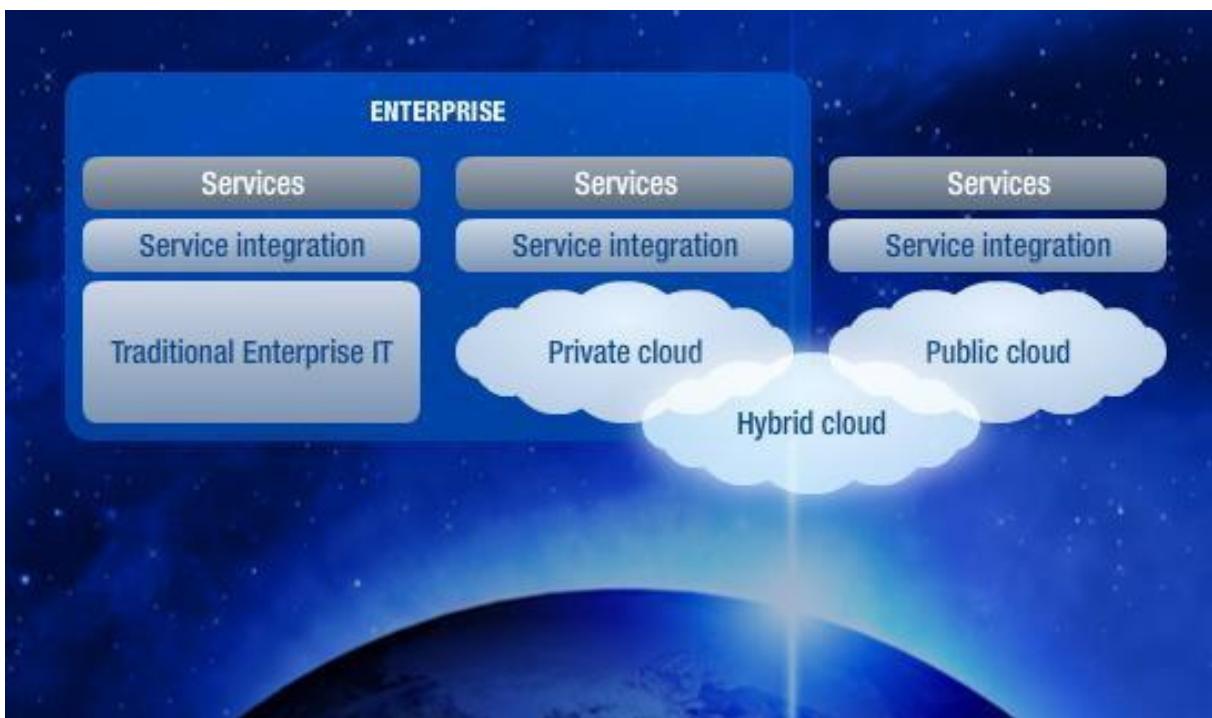
Private Cloud

- Operated solely for an organization
- May be managed by the organization or a third party
- Limits access to enterprise and partner



Hybrid Cloud

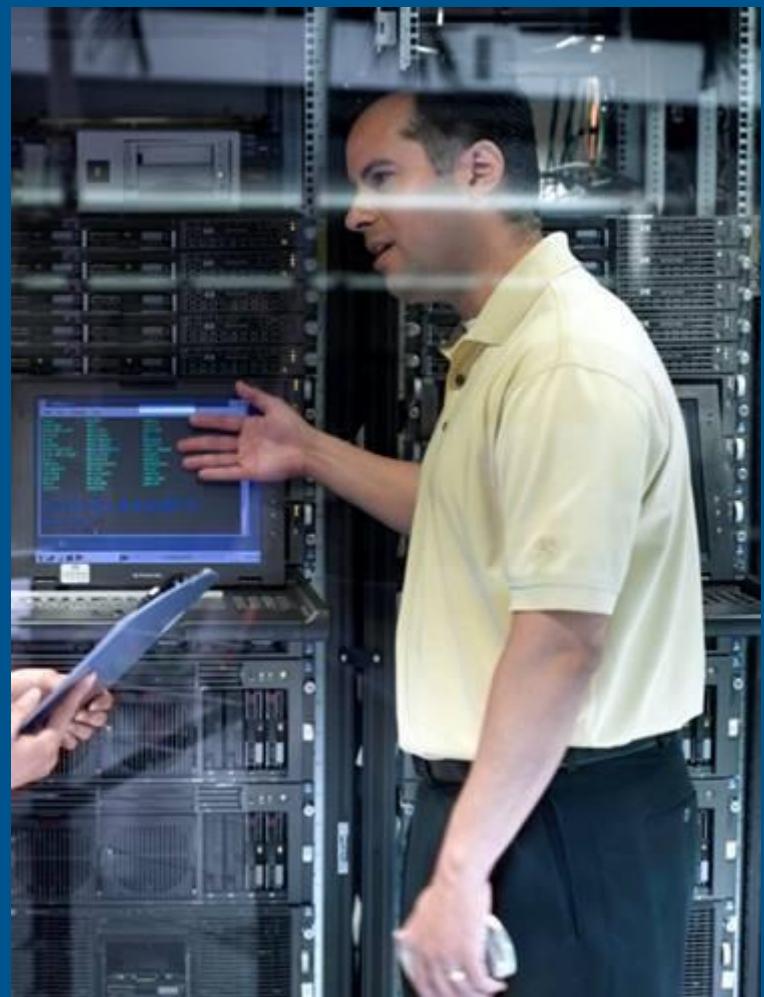
- Composition of two or more clouds (private, community, or public) bound together by standardized or proprietary technology that enables data and application portability



Conclusion

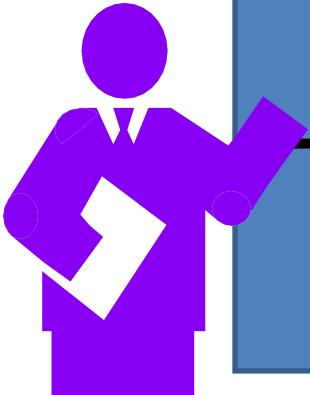
Cloud Computing is the fastest growing part of network based computing. It provides tremendous benefits to customers of all sizes: simple users, developers, enterprises and all types of organizations.

Lab Activity



MODULE#2 -Introduction to AWS

Agenda



- What is AWS ?**
- Benefits of using AWS?**
- History and Origins of AWS**
- AWS Components**
- AWS Services**
- AWS Regions**
- AWS Availability Zones**
- AWS Edge Locations**
- AWS Certifications**



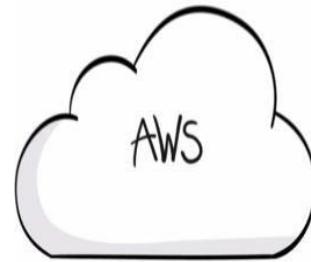
What is AWS?

Amazon Web Services(AWS) are a collection of remote services(Also called as web service) offered by the amazon.com over the internet build and run an application.

Amazon Web Services (AWS) - robust, scalable and affordable infrastructure for cloud computing.

AWS provide compute, storage and database service quickly provisioning the IT needs

What is AWS ?

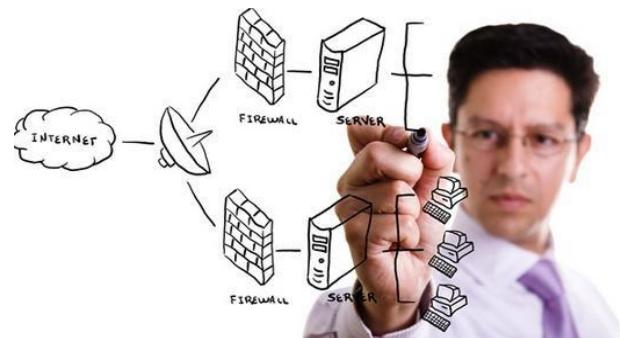


AWS is a set of cloud computing services

AWS is flexibility, availability, and scalability

**AWS is Elasticity: scale up or scale down as needed,
We can get resources instantly**

AWS is fully on demand



Benefits of using AWS ?

- ***Pay-per use model***
you are only charged for disk space, CPU time and bandwidth that you use.
- ***Instant scalability***
Your Service automatically scales on AWS stack.
- ***Reliable/Redundant***
Infrequent outages (so far).
Data is redundant in the cloud.
All services have built-in security
- ***Security***
AWS delivers a scalable cloud-computing platform that provides customers with end-to-end security and end-to-end privacy.
- ***Most services accessed via simple REST/SOAP API***
Libraries are available in all major languages.
Minimal learning curve.

Service Level Agreement (SLA)

SLA between 99.99 and 100% availability

Amazon S3 maintains a durability of 99.9999%

- ***Availability***

Availability Zones exist on isolated fault lines, flood plains, and electrical grids to substantially reduce the chance of simultaneous failure

- ***Support***

AWS provides 24/7 support in the real-time operational status of all services around the globe



History and Origins of AWS

Grew out of Amazon's need to rapidly provision and configure machines of standard configurations for its own business.

Early 2000s – Both private and shared data centers began using virtualization to perform “server consolidation”

2003 – Internal memo by Chris Pinkham describing an “infrastructure service for the world.”

2006 – S3 first deployed in the spring, EC2 in the fall

2008 – Elastic Block Store available.

- Amazon EC2 Now Offers Windows Server 2008

2009 – Relational Database Service

- Announcing Amazon CloudFront Streaming

- AWS Launches the Northern California Region

- Amazon EC2 Instances Now Can Boot from Amazon EBS

- Introducing Amazon EBS Shared Snapshots

2012 – DynamoDB

- Announcing Windows Server 2012 on AWS

2013 – Amazon DynamoDB Console now available in AWS GovCloud (US)

- New Management Console and support for more AWS features

- Announcing the next generation of Amazon EC2 High I/O instance

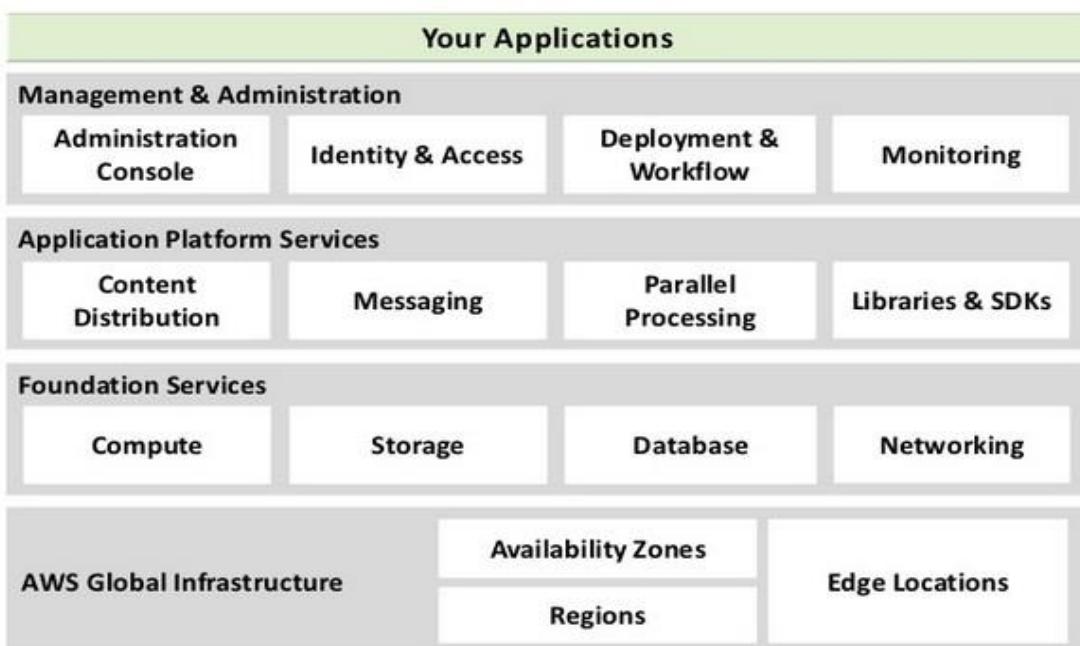
2014 – Amazon Glacier is Now Available in the AWS GovCloud (US) Region

- Amazon CloudWatch Logs - Cross Account Subscriptions

- Amazon EC2 Container Service Available in US West (N. California)

Various Components of AWS

AWS Computing Platform



The Big Picture of AWS

Application				Deployment & Management
Identity & Access AWS IAM	Administration Interface Management Console	Monitoring Amazon CloudWatch	Deployment & Automation AWS Elastic Beanstalk AWS CloudFormation	
Search Amazon CloudSearch	Content Delivery Amazon CloudFront	Parallel Processing Amazon Elastic MapReduce	Workflow Amazon SWF	Application Services
Queuing Amazon SQS	Notifications Amazon SNS	Email Amazon SES	Libraries & SDKs Java, PHP, Python, Ruby & .NET	
Compute Amazon EC2 Auto Scaling	Storage Amazon S3 Amazon EBS Amazon Storage Gateway Amazon Glacier	Databases Amazon RDS Amazon DynamoDB Amazon SimpleDB Amazon ElastiCache	Networking Amazon VPC Elastic Load Balancing Amazon Route 53 AWS Direct Connect	Foundation Services
AWS Global Infrastructure				

AWS Services

Amazon provides the various types of services as below

EC2 – Virtual Private Servers

- Similar to standard VPS's and are called instances
- Available in a variety of sizes (613MB to 64GB of RAM)
- All major operating systems supported

S3 – Cloud Storage

- Highly scalable (some companies have PBs of data)
- Highly available – data is stored in multiple data centers

RDS – Relational Database

- MySQL, SQL and Oracle databases
- Variety of sizes
- High availability available for an extra cost
- Read replication
- Scheduled backups

Route 53 – DNS

- High availability
- Works well with other AWS services
- Fast and secure
- Pay per zone and million queries

CloudFront – Content Delivery Network

- Uses a number of global edge locations
- Fast
- Pay per GB of data transfer (prices vary on the region)

Glacier – Data Archiving

- Store data for a prolong period of time (years)
- Very cheap
- Data retrieval takes a few hours

DynamoDB – NoSQL Database

- High availability
- Data stored on SSDs for speed
- Pay for number of read/writes per second

ElastiCache - Memcache

- **Cache database results**
- **Available in a variety of sizes**
- **Pay per hours**

AWS Services

Compute

-  EC2
Virtual Servers in the Cloud
-  Lambda
Run Code in Response to Events
-  EC2 Container Service
Run and Manage Docker Containers

Storage & Content Delivery

-  S3
Scalable Storage in the Cloud
-  Elastic File System PREVIEW
Fully Managed File System for EC2
-  Storage Gateway
Integrates On-Premises IT Environments with Cloud Storage
-  Glacier
Archive Storage in the Cloud
-  CloudFront
Global Content Delivery Network

Database

-  RDS
MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
-  DynamoDB
Predictable and Scalable NoSQL Data Store
-  ElastiCache
In-Memory Cache
-  Redshift
Managed Petabyte-Scale Data Warehouse Service

Networking

-  VPC
Isolated Cloud Resources
-  Direct Connect
Dedicated Network Connection to AWS
-  Route 53
Scalable DNS and Domain Name Registration

Administration & Security

-  Directory Service
Managed Directories in the Cloud
-  Identity & Access Management
Access Control and Key Management
-  Trusted Advisor
AWS Cloud Optimization Expert
-  CloudTrail
User Activity and Change Tracking
-  Config
Resource Configurations and Inventory
-  CloudWatch
Resource and Application Monitoring
-  Service Catalog
Personalized Catalog of AWS Resources

Deployment & Management

-  Elastic Beanstalk
AWS Application Container
-  OpsWorks
DevOps Application Management Service
-  CloudFormation
Templated AWS Resource Creation
-  CodeDeploy
Automated Deployments
-  CodeCommit
Managed Git Repositories
-  CodePipeline
Continuous Delivery

Analytics

-  EMR
Managed Hadoop Framework
-  Kinesis
Real-time Processing of Streaming Big Data
-  Data Pipeline
Orchestration for Data-Driven Workflows

Application Services

-  SQS
Message Queue Service
-  SWF
Workflow Service for Coordinating Application Components
-  AppStream
Low Latency Application Streaming
-  Elastic Transcoder
Easy-to-use Scalable Media Transcoding
-  SES
Email Sending Service
-  CloudSearch
Managed Search Service
-  API Gateway
Build, Deploy and Manage APIs

Mobile Services

-  Cognito
User Identity and App Data Synchronization
-  Device Farm
Test Android, Fire OS, and iOS apps on real devices in the Cloud
-  Mobile Analytics
Collect, View and Export App Analytics
-  SNS
Push Notification Service

Enterprise Applications

-  WorkSpaces
Desktops in the Cloud
-  WorkDocs
Secure Enterprise Storage and Sharing Service
-  WorkMail PREVIEW
Secure Email and Calendaring Service

AWS Regions

AWS Regions are completely isolated from each other and are in different parts of the world and AWS Regions is

- A collection of data centers (Availability Zones or “AZ”)**
- Each region has a set number of AZs**
- All AZs in a region connected by high-bandwidth**
- Cost vary from Region to Region**
- Default Region in US East**



AWS Regions

- Datacenters spread across the globe
- There are 9 Regions
 - US-East (N.Virginia)
 - US-West-1 (N.California)
 - EU (Ireland)
 - APAC-1 (Singapore)
 - APAC-2 (Tokyo)
 - APAC-3 (Sydney)
 - GovCloud
 - US-West-2 (Oregon)
 - South America (Sao Paulo)

AWS Availability Zones

- Subset of a Region
- Physically isolated & independent infrastructure
- **Note: There is an extra region called the AWS GovCloud region, but this can only be used by government agencies**
- Low latency
- Every Region has a minimum of 2 AZs



AWS Edge Locations

Edge locations are the important part of the AWS infrastructure. There are currently 40 edge locations. They are located in most the major cities around the world and are used by CloudFront (CDN) and route53(DNS) to distribute content nearer to the end user

AWS Certifications

Associate Level

AWS Certified Solutions
Architect – Associate

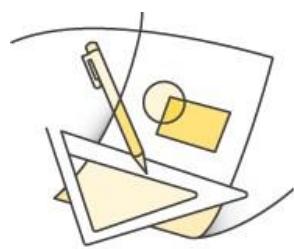
AWS Certified
Developer – Associate

AWS Certified SysOps
Administrator – Associate

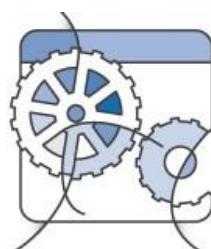
Professional Level

AWS Certified Solutions
Architect – Professional

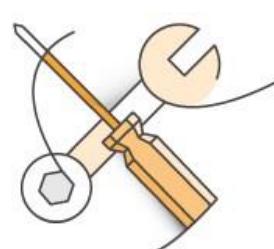
AWS Certified DevOps Engineer—Professional



Solutions Architect



Developer



SysOps Administrator

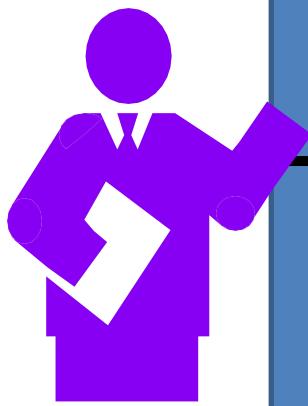
Lab activity





MODULE#3 -Elastic Cloud Computing EC2

Agenda



Introduction to AWS EC2
Benefits of using AWS EC2?
Amazon EC2 use cases
AWS EC2 Pricing Model
AWS Instance type
Amazon Machine Image [AMI]
Selecting EC2 instance
AWS Regions and Availability zones
Launching Amazon EC2 Instance
Step by Step

Introduction to Amazon EC2

Amazon EC2, which is also known as Amazon Elastic Compute Cloud, provides resizable computing capacity in the Amazon Web Services (AWS)

It is a 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.

EC2 is the core service of AWS cloud platform and launched in 2006

These are the virtual servers, also called as an instances we can use these instances pay per use basis

Benefits of using AWS EC2?

Easier and Faster - Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.



Elastic and Scalable – Quickly add and subtract resources to applications to meet customer demand and manage costs. Avoid provisioning resources up-front for projects with variable consumption rates or short lifetimes.



High Availability – Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.



Cost-Effective – Consume only the amount of compute, storage and other IT resources needed. No long-term commitment, minimum spend or up-front investment is required.



Amazon EC2 Typical Use Cases

- Application Hosting
- Content Delivery
- E-Commerce
- Media Hosting
- Web Hosting

EC2 Pricing Model

- Free Usage Tier
- On-Demand Instances
 - Start and stop instances whenever you like, costs are rounded up to the nearest hour. (Worst price)
- Reserved Instances
 - Pay up front for one/three years in advance. (Best price)
 - Unused instances can be sold on a secondary market.
- Spot Instances
 - Specify the price you are willing to pay, and instances get started and stopped without any warning as the marked changes. (Kind of like Condor!)
 - Dedicated Instances
- Charge for instance

AWS Instance type

- Mapped to the H/W specification
- Classified based on the CPU, memory and storage
- Choice of instance types
 - Standard
 - Medium
 - Micro
 - High-Memory
 - High-CPU
 - High I/O
 - Cluster Compute
 - Cluster GPU
- Instance type influences the performance & cost

Model	vCPU	CPU Credits / hour	Mem (GiB)	Storage (GB)
t2.micro	1	6	1	EBS Only
t2.small	1	12	2	EBS Only
t2.medium	2	24	4	EBS Only

Use Cases

Development environments, build servers, code repositories, low-traffic web applications, early product experiments, small databases.

Model	vCPU	Mem (GiB)	SSD Storage (GB)
m3.medium	1	3.75	1 x 4
m3.large	2	7.5	1 x 32
m3.xlarge	4	15	2 x 40
m3.2xlarge	8	30	2 x 80

Use Cases

Small and mid-size databases, data processing tasks that require additional memory, caching fleets, and for running backend servers for SAP, Microsoft SharePoint, and other enterprise applications.

Model	vCPU	Mem (GiB)	SSD Storage (GB)
c3.large	2	3.75	2 x 16
c3.xlarge	4	7.5	2 x 40
c3.2xlarge	8	15	2 x 80
c3.4xlarge	16	30	2 x 160
c3.8xlarge	32	60	2 x 320

Use Cases

High performance front-end fleets, web-servers, on-demand batch processing, distributed analytics, high performance science and engineering applications, ad serving, batch processing, MMO gaming, video encoding, and distributed analytics.

Model	vCPU	Mem (GiB)	SSD Storage (GB)
r3.large	2	15.25	1 x 32
r3.xlarge	4	30.5	1 x 80
r3.2xlarge	8	61	1 x 160
r3.4xlarge	16	122	1 x 320
r3.8xlarge	32	244	2 x 320

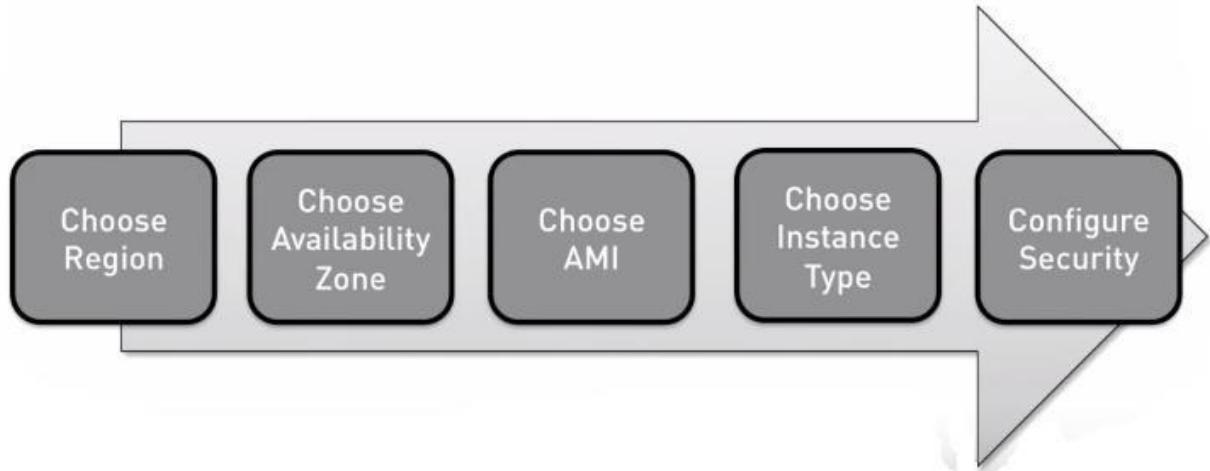
Use Cases

We recommend memory-optimized instances for high performance databases, distributed memory caches, in-memory analytics, genome assembly and analysis, larger deployments of SAP, Microsoft SharePoint, and other enterprise applications.

Amazon Machine Image [AMI]

- Server template / VM image
- Comes with pre-installed OS and optional S/W
- AMI can be launched to create Instances
- A variety of pre-built AMIs in the catalog
- Existing AMIs can be customized and saved (Bundling)
- Independent of the configuration
- Every AMI is uniquely identified

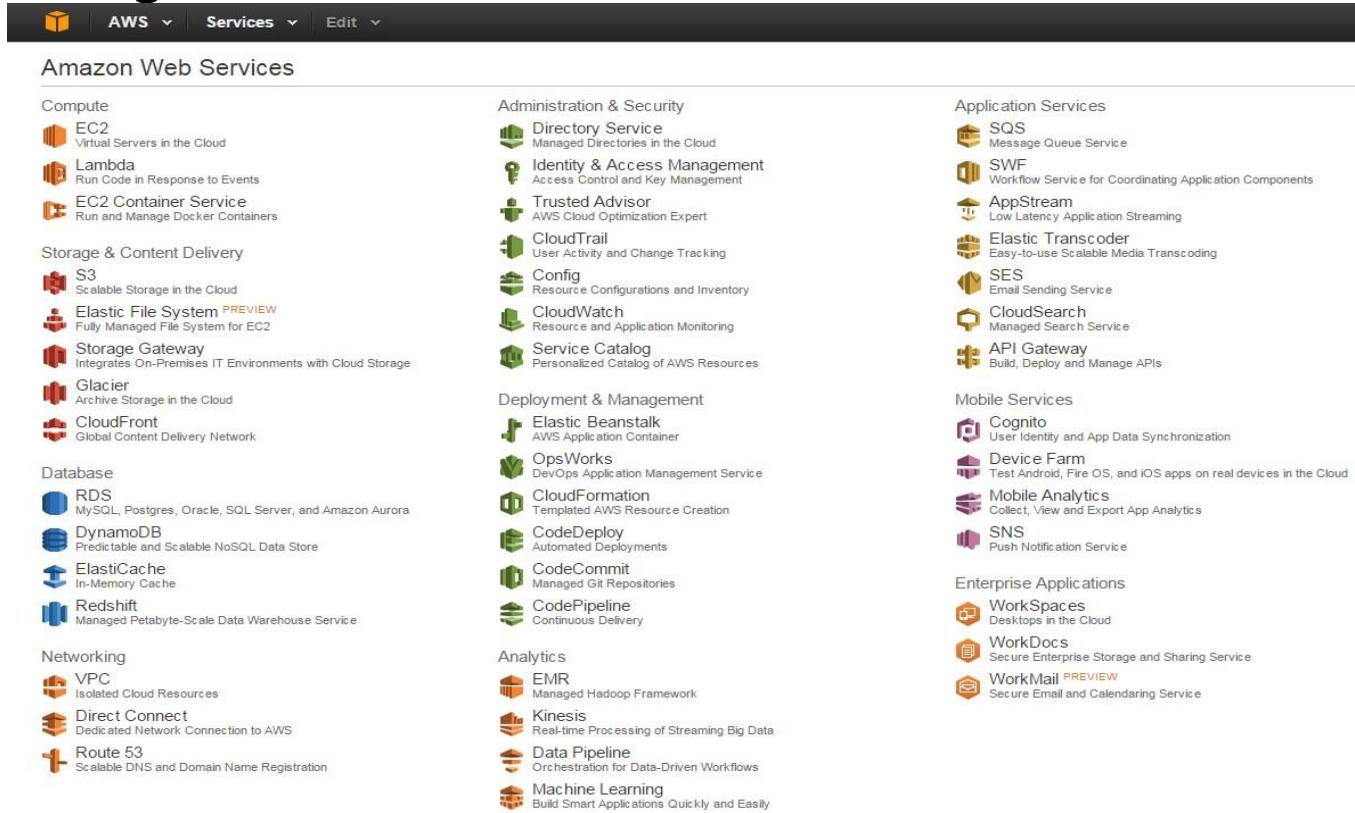
Launching an Amazon EC2 Instance Step by Step



Launching an Amazon EC2 Instance Step by Step

- Sign up for AWS at <http://aws.amazon.com>
- Apply the service credit you received by email.
- Create and download a Key-Pair, save it in your home directory.
- Create a VM via the AWS Console
- Connect to your newly-created VM like this:
 - `ssh -i my-aws-keypair.pem ec2-user@ip-address-of-vm`

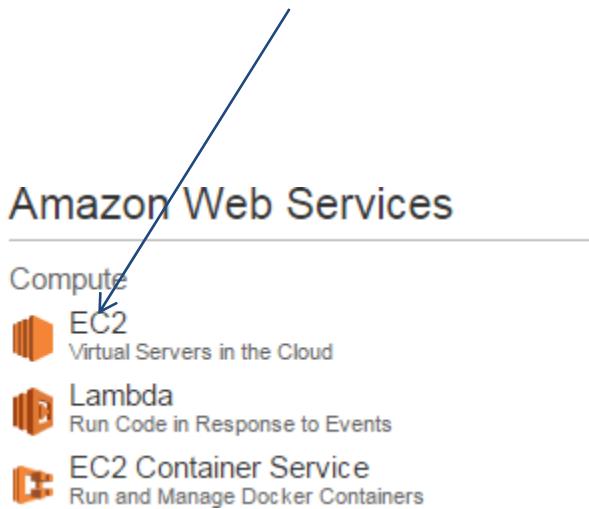
Login to AWS Console



The screenshot shows the AWS Management Console homepage. At the top, there's a navigation bar with icons for Home, AWS, Services, and Edit. Below the navigation bar, the title "Amazon Web Services" is displayed. The page is organized into several sections:

- Compute** (highlighted):
 - EC2: Virtual Servers in the Cloud
 - Lambda: Run Code in Response to Events
 - EC2 Container Service: Run and Manage Docker Containers
- Storage & Content Delivery**:
 - S3: Scalable Storage in the Cloud
 - Elastic File System PREVIEW: Fully Managed File System for EC2
 - Storage Gateway: Integrates On-Premises IT Environments with Cloud Storage
 - Glacier: Archive Storage in the Cloud
 - CloudFront: Global Content Delivery Network
- Database**:
 - RDS: MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
 - DynamoDB: Predictable and Scalable NoSQL Data Store
 - ElastiCache: In-Memory Cache
 - Redshift: Managed Petabyte-Scale Data Warehouse Service
- Networking**:
 - VPC: Isolated Cloud Resources
 - Direct Connect: Dedicated Network Connection to AWS
 - Route 53: Scalable DNS and Domain Name Registration
- Administration & Security**:
 - Directory Service: Managed Directories in the Cloud
 - Identity & Access Management: Access Control and Key Management
 - Trusted Advisor: AWS Cloud Optimization Expert
 - CloudTrail: User Activity and Change Tracking
 - Config: Resource Configurations and Inventory
 - CloudWatch: Resource and Application Monitoring
 - Service Catalog: Personalized Catalog of AWS Resources
- Deployment & Management**:
 - Elastic Beanstalk: AWS Application Container
 - OpsWorks: DevOps Application Management Service
 - CloudFormation: Templated AWS Resource Creation
 - CodeDeploy: Automated Deployments
 - CodeCommit: Managed Git Repositories
 - CodePipeline: Continuous Delivery
- Analytics**:
 - EMR: Managed Hadoop Framework
 - Kinesis: Real-time Processing of Streaming Big Data
 - Data Pipeline: Orchestration for Data-Driven Workflows
 - Machine Learning: Build Smart Applications Quickly and Easily
- Application Services**:
 - SQS: Message Queue Service
 - SWF: Workflow Service for Coordinating Application Components
 - AppStream: Low Latency Application Streaming
 - Elastic Transcoder: Easy-to-use Scalable Media Transcoding
 - SES: Email Sending Service
 - CloudSearch: Managed Search Service
 - API Gateway: Build, Deploy and Manage APIs
- Mobile Services**:
 - Cognito: User Identity and App Data Synchronization
 - Device Farm: Test Android, Fire OS, and iOS apps on real devices in the Cloud
 - Mobile Analytics: Collect, View and Export App Analytics
 - SNS: Push Notification Service
- Enterprise Applications**:
 - WorkSpaces: Desktops in the Cloud
 - WorkDocs: Secure Enterprise Storage and Sharing Service
 - WorkMail PREVIEW: Secure Email and Calendering Service

Click on EC2 from Compute services from AWS Console



The screenshot shows the "Compute" section of the AWS Management Console. The "EC2" service is highlighted with a blue arrow pointing to it. The services listed are:

- EC2: Virtual Servers in the Cloud
- Lambda: Run Code in Response to Events
- EC2 Container Service: Run and Manage Docker Containers

You can see the EC2 dashboard and all EC2 instance

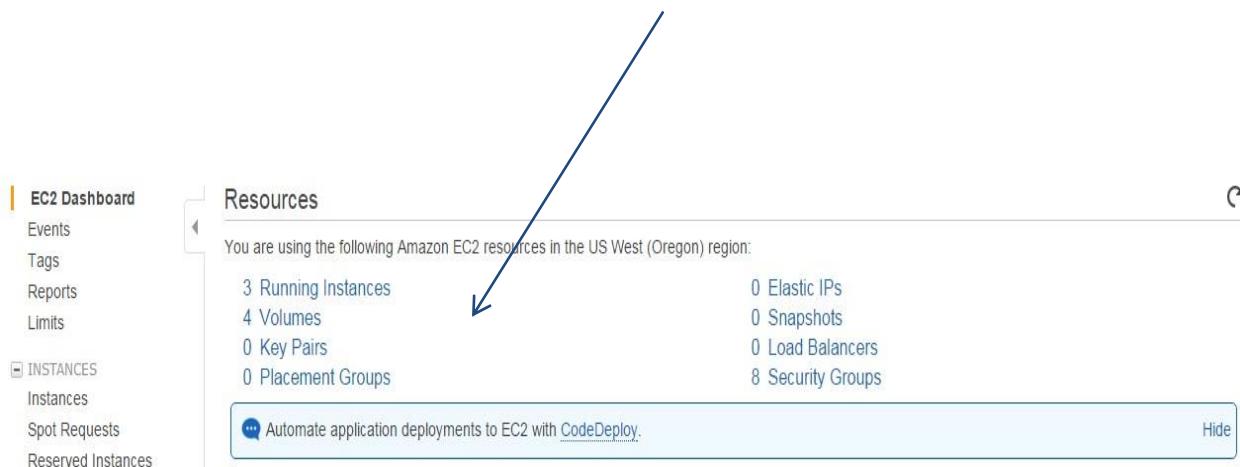


The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with links for EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with sub-links for Instances, Spot Requests, and Reserved Instances), and a Hide button. The main content area is titled "Resources" and displays the following summary:

- You are using the following Amazon EC2 resources in the US West (Oregon) region:
- 3 Running Instances
- 4 Volumes
- 0 Key Pairs
- 0 Placement Groups
- 0 Elastic IPs
- 0 Snapshots
- 0 Load Balancers
- 8 Security Groups

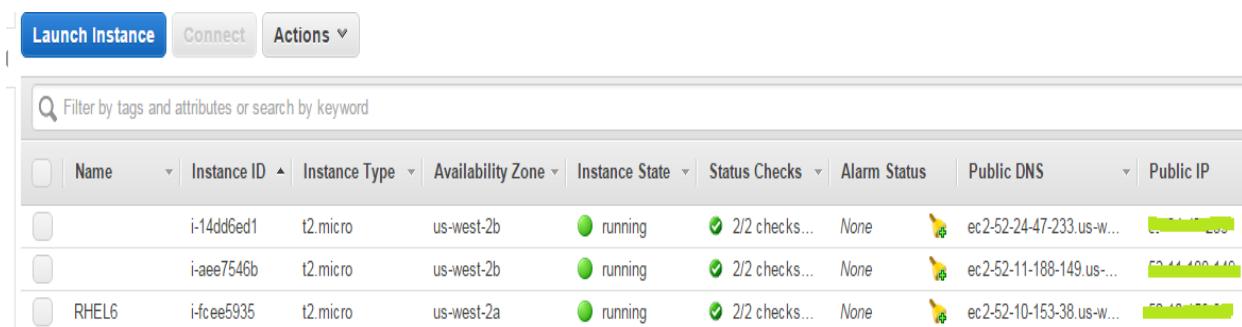
At the bottom, there's a call-to-action button: "Automate application deployments to EC2 with CodeDeploy." A "Hide" link is located at the far right.

Click on running instances to see the existing running instances



This screenshot is identical to the one above it, showing the AWS EC2 Dashboard with the same resource summary and sidebar. A blue arrow originates from the text "Click on running instances to see the existing running instances" and points directly to the "3 Running Instances" count in the resource summary table.

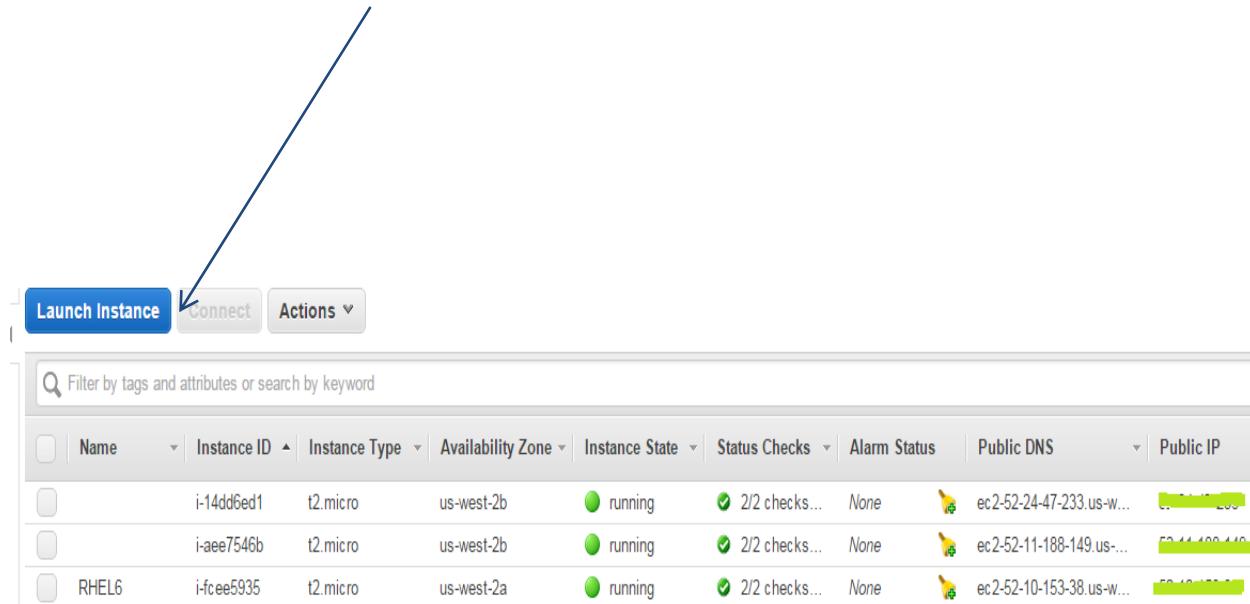
Running instances



The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for "Launch Instance", "Connect", and "Actions". Below that is a search bar with the placeholder "Filter by tags and attributes or search by keyword". The main table lists three running instances:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
	i-14ddfed1	t2.micro	us-west-2b	running	2/2 checks...	None	ec2-52-24-47-233.us-w...	128.199.24.47.233
	i-aee7546b	t2.micro	us-west-2b	running	2/2 checks...	None	ec2-52-11-188-149.us-...	128.11.188.149
RHEL6	i-fcee5935	t2.micro	us-west-2a	running	2/2 checks...	None	ec2-52-10-153-38.us-w...	128.10.153.38

Click on Launch Instance to create New Instance



The screenshot shows the AWS CloudWatch Metrics console. At the top, there are three buttons: 'Launch Instance' (highlighted with a blue arrow), 'Connect', and 'Actions'. Below the buttons is a search bar with the placeholder 'Filter by tags and attributes or search by keyword'. A table follows, displaying three EC2 instances:

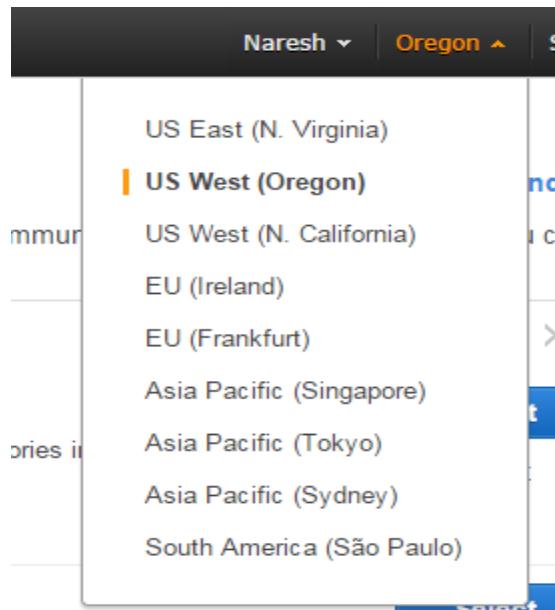
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
<input type="checkbox"/>	i-14dd6ed1	t2.micro	us-west-2b	running	2/2 checks...	None	Normal	ec2-52-24-47-233.us-w...	52.24.47.233
<input type="checkbox"/>	i-ae7546b	t2.micro	us-west-2b	running	2/2 checks...	None	Normal	ec2-52-11-188-149.us-w...	52.11.188.149
<input type="checkbox"/>	RHEL6	i-fcee5935	t2.micro	us-west-2a	running	2/2 checks...	None	ec2-52-10-153-38.us-w...	52.10.153.38

AWS Regions

- Datacenters spread across the globe
- There are 9 Regions
 - US-East (N.Virginia)
 - US-West-1 (N.California)
 - EU (Ireland)
 - APAC-1 (Singapore)
 - APAC-2 (Tokyo)
 - APAC-3 (Sydney)
 - GovCloud
 - US-West-2 (Oregon)
 - South America (Sao Paulo)

Note: There is an extra region called the AWS GovCloud region, but this can only be used by government agencies

AWS Regions



AWS Availability Zones

- Subset of a Region
- Physically isolated & independent infrastructure
- High speed connectivity
- Low latency
- Every Region has a minimum of 2 AZs





Select the respective OS Image to install

Step 1: Choose an Amazon Machine Image (AMI)

[Cancel and Exit](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

K < 1 to 22 of 22 AMIs >

Category	Image Name	Description	Action
My AMIs	Amazon Linux AMI 2015.03 (HVM), SSD Volume Type - ami-e7527ed7	The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.	Select
AWS Marketplace	Amazon Linux	Amazon Linux AMI 2015.03 (HVM), SSD Volume Type - ami-e7527ed7	64-bit
Community AMIs	Red Hat Enterprise Linux 7.1 (HVM), SSD Volume Type - ami-4dbf9e7d	Red Hat Enterprise Linux version 7.1 (HVM), EBS General Purpose (SSD) Volume Type	Select
Free tier only ⓘ	Red Hat	Red Hat Enterprise Linux 7.1 (HVM), SSD Volume Type - ami-4dbf9e7d	64-bit
Windows	Microsoft Windows Server 2012 R2 Base - ami-c3b3b1f3	Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]	Select

Select the Instance type

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of resources to choose from. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: [All instance types](#) [Current generation](#) [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GB) ⓘ
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of lower prices, or use reserved instances to save money.

Number of instances	<input type="text" value="1"/>	
Purchasing option	<input checked="" type="checkbox"/> Request Spot Instances	
Network	vpc-f17e1f94 (172.31.0.0/16) (default)	<input type="button" value="Create new VPC"/>
Subnet	No preference (default subnet in any Availability Zone)	<input type="button" value="Create new subnet"/>
Auto-assign Public IP	Use subnet setting (Enable)	
IAM role	None	<input type="button" value="Create new IAM role"/>
Shutdown behavior	Stop	
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply.	
Tenancy	Shared tenancy (multi-tenant hardware)	Additional charges will apply for dedicated tenancy.

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/sda1	snap-6bbf7736	<input type="text" value="10"/>	General Purpose (SSD)	30 / 3000	<input checked="" type="checkbox"/>	Not Encrypted
<input type="button" value="Add New Volume"/>							

Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	<input type="text"/> <input type="button" value="X"/>
<input type="button" value="Create Tag"/> (Up to 10 tags maximum)	



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: Create a new security group

Select an existing security group

Security group name:

Description:

Type <i>i</i>	Protocol <i>i</i>	Port Range <i>i</i>	Source <i>i</i>
SSH	TCP	22	Anywhere ▾ 0.0.0.0/0

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

⚠ Improve your instances' security. Your security group, launch-wizard-8, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.

You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details

Red Hat Enterprise Linux 7.1 (HVM), SSD Volume Type - ami-4dbf9e7d

Free tier eligible Red Hat Enterprise Linux version 7.1 (HVM), EBS General Purpose (SSD) Volume Type
Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Select an existing key pair or create a new key pair



A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.

[Cancel](#)

[Launch Instances](#)



Launch Status

✓ Your instances are now launching

The following instance launches have been initiated: [i-e93eb32c](#) [View launch log](#)

💬 Get notified of estimated charges

[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key	
	i-14dd6ed1	t2.micro	us-west-2b	● running	● 2/2 checks...	None	●	ec2-52-24-47-233.us-w...	52.24.47.233	New	
	i-aeef7546b	t2.micro	us-west-2b	● running	● 2/2 checks...	None	●	ec2-52-11-188-149.us-...	52.11.188.149		
	RHEL6	i-fceef5935	t2.micro	us-west-2a	● running	● 2/2 checks...	None	●	ec2-52-10-153-38.us-w...	52.10.153.38	none
		i-e93eb32c	t2.micro	us-west-2b	● pending	● Initializing	None	●	ec2-52-11-82-27.us-we...	52.11.82.27	

Launch Status

✓ Your instances are now launching

The following instance launches have been initiated: [i-e93eb32c](#) [View launch log](#)

💬 Get notified of estimated charges

[Create billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

When NOT to user EC2 ?

- Physical Access to the system
- Non supporting Operating system
- Multicast/Manipulation of L2 Networking
- Disks attached to multiple VM's
- You don't want to do system administration



Lab Activity



By
Kmreddi



MODULE#4 -Identity and Access Management[IAM]

Agenda



Introduction to Identity and Access management (IAM)
Understanding IAM console
Creating and managing security group
Creating and managing users
Managing the user passwords and security Key ID
Creating and managing roles
Understanding and managing policy
Understanding multi factor authentication
User login process



➤ What is IAM?

IAM stands for Identity and Access Management

IAM is a web service that enables you to manage users and group permissions in AWS

It is targeted at organizations with multiple users or systems that use AWS products such as Amazon Elastic Compute Cloud, Amazon Relational Database Service, and the AWS Management Console



➤ Why we go for IAM?

To avoid a security and logistical headache

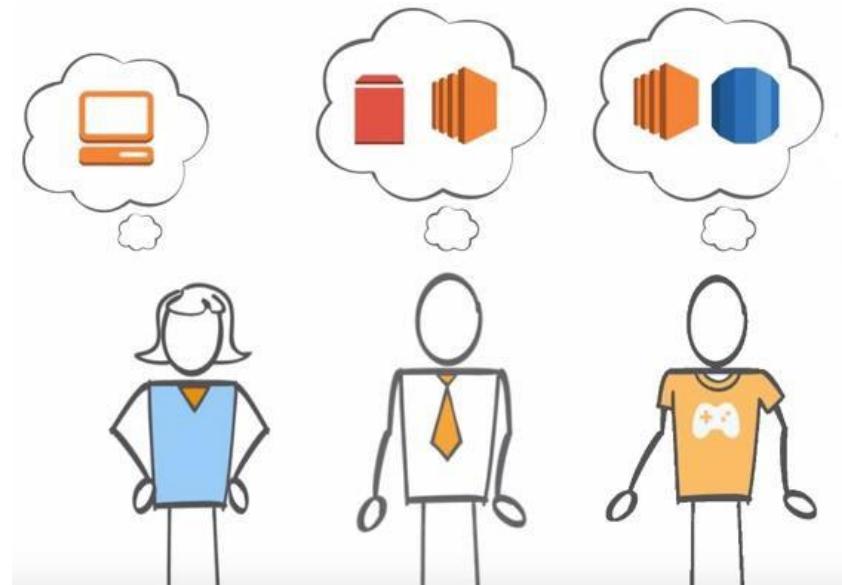
When you create an AWS account, it has permissions to do anything and everything with all the resources

IAM Allows you to limit access as needed and gives you the peace of mind that approved people are accessing the right resources in the desired manner

➤ Why we go for IAM?

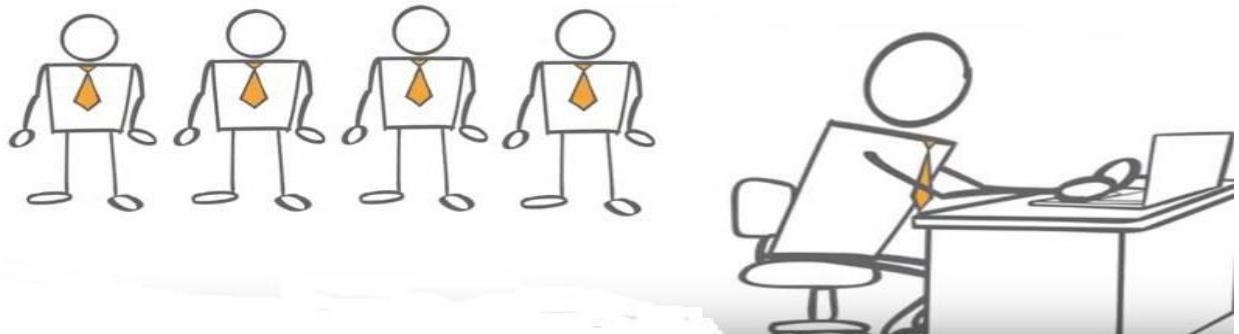


IAM will allow us to create multiple users with individual security credentials and permissions, with this IAM, each user is allowed to do only what they need to do



➤ Why we go for IAM?

Each user in the AWS account must have a unique set of credentials to access the console





➤ What is IAM?

Different types of users have different set of permissions



Administrators need to access all AWS resources



Amazon
EC2



Amazon
RDS



Amazon
S3

Developers need only access on Amazon Elastic Compute Cloud (EC2)



Amazon
EC2



Amazon
RDS



Amazon
S3



➤ What is IAM ?

We can use IAM to create a unique user for each employee and define their permissions

Adele
(Administrator)



Bob
(Systems Operations)



Dave
(Developer)



➤ What is a Group ?

A group is a collection of IAM users

After you set permissions on a group, those permissions are set to all users in the group

Even if we create user, we need to use groups to set permissions.
We need to manage access for number of groups instead of managing access for every individual user.



➤ Administrator Access

Provides full access to AWS services and resources.



➤ Creating group

Select IAM from AWS console

The screenshot shows the AWS Services dashboard. At the top, there are links for Compute (EC2, Lambda, EC2 Container Service), Administration & Security (Directory Service, Identity & Access Management, Trusted Advisor), and other services like CloudWatch Metrics and CloudWatch Logs. The 'Identity & Access Management' service is highlighted with a blue border.

➤ Creating Group

The screenshot shows the AWS IAM Groups page. On the left, there's a sidebar with links for Dashboard, Details, Groups, Users, Roles, Policies, Identity Providers, Account Settings, Credential Report, and Encryption Keys. A note at the top right says: "The Password Policy page has been renamed to Account Settings. Click Account Settings to find your account's password configuration options." The main area displays the "Welcome to Identity and Access Management" message, the IAM Resources section (Users: 1, Groups: 1, Roles: 0, Identity Providers: 0), and the Security Status section with a checklist:

- Delete your root access keys
- Activate MFA on your root account
- Create individual IAM users
- Use groups to assign permissions
- Apply an IAM password policy



➤ Creating Group

Screenshot of the AWS Groups page:

- Top navigation bar: AWS Services Edit
- Left sidebar menu:
 - Dashboard
 - Details
 - Groups** (selected)
 - Users
 - Roles
 - Policies
 - Identity Providers
 - Account Settings
 - Credential Report
- Main content area:
 - Search bar: Search
 - Table:
 - Group Name: Supportgroup (checkbox selected)
 - Users: 0

➤ Creating Group

Create New Group Wizard

Step 1: Group Name

Step 2: Attach Policy

Step 3: Review

Set Group Name

Specify a group name. Group names can be edited any time.

Group Name: awsadmin

Example: Developers or ProjectAlpha
Maximum 128 characters



➤ Attach the policy

Attach Policy

Select up to two policies to attach to the group.

Filter: Policy Type ▾		Search
	Policy Name ▾	Attached Entities ▾
<input type="checkbox"/>	AmazonSNSSRole	1
<input type="checkbox"/>	AdministratorAccess	0
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	0
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	0

➤ Review the group

AWS Services Edit ▾

Create New Group Wizard

Step 1: Group Name

Step 2: Attach Policy

Step 3: Review

Review

Review the following information, then click **Create Group** to proceed.

Group Name	awsadmin	Edit Group Name
Policies		Edit Policies



Group created

The screenshot shows the AWS Groups management interface. On the left, a sidebar menu includes options like Dashboard, Details, Groups (which is selected and highlighted in orange), Users, Roles, and Policies. The main content area has a "Create New Group" button and a "Group Actions" dropdown. Below these are search and filter fields. A table lists groups with columns for Group Name and Users. The "awsadmin" group is selected, indicated by a green border around its row. The "Supportgroup" group is listed below it with 0 users.

Group Name	Users
awsadmin	0
Supportgroup	0

➤ Changing group name

Select group->Group actions->Edit group

This screenshot shows the same AWS Groups interface as above, but with a different focus. The "Edit Group Name" option is highlighted in the "Group Actions" dropdown menu. The main table shows the "awsadmin" group selected (indicated by a checked checkbox) and the "Supportgroup" group. The "Edit Group Name" option is part of a larger context menu that also includes "Add Users to Group", "Delete Group", and "Remove Users from Group".

Group Name	Users
awsadmin	0
Supportgroup	0



➤ Deleting group

Select group->Group actions->Delete group

The screenshot shows the AWS Groups management interface. On the left, a sidebar lists navigation options: Dashboard, Details, Groups (which is selected and highlighted in orange), Users, Roles, Policies, and Identity Providers. The main content area displays a table of groups. A context menu is open over the first group row, which contains the following options: Create New Group, Group Actions (with a dropdown arrow), Add Users to Group, Delete Group (which is highlighted in orange), Edit Group Name, and Remove Users from Group. The group table shows two entries: 'awsadmin' (selected) and 'Supportgroup'. To the right of the table, a column labeled 'Users' shows '0' for both groups.

➤ Adding users to group

Select group->Group actions->Add users to Group

This screenshot is identical to the one above, showing the AWS Groups management interface. The sidebar is the same, showing 'Groups' is selected. The main content area shows a table with two groups: 'awsadmin' and 'Supportgroup'. A context menu is open over the 'awsadmin' group, with the 'Add Users to Group' option highlighted in orange. The other options in the menu are 'Delete Group', 'Edit Group Name', and 'Remove Users from Group'. The 'Users' column indicates '0' users for both groups.



➤ Select users to add group

AWS Services Edit

Add Users to Group

Select users to add to the group **awsadmin**

<input type="checkbox"/>	User Name ▾	Groups	Password
<input checked="" type="checkbox"/>	suvenit	0	
<input checked="" type="checkbox"/>	kvreddi	0	
<input type="checkbox"/>	Naresh	0	

Dashboard

Details

Groups

Users

Roles

Policies

Identity Providers

Create New Group

Group Actions ▾

Search

Group Name ▾

Users

2

awsadmin

Supportgroup

0

AWS Services Edit

Dashboard

Details

Groups

Users

Roles

Policies

Identity Providers

Create New Users

User Actions ▾

Search

User Name ▾

Groups

kvreddi

0

Naresh

0



➤ Creating user

Screenshot of the AWS IAM 'Create User' page.

The top navigation bar shows: AWS Services Edit.

The main area has a title 'Create User' and a sub-section 'Enter User Names:' with five input fields. The first field contains 'suvenit'. Below the fields is a note: 'Maximum 64 characters each'.

1.	suvenit
2.	
3.	
4.	
5.	

➤ User key ID

Screenshot of the AWS IAM 'Create User' page showing successful creation.

The top navigation bar shows: AWS Services Edit.

The main area has a title 'Create User' and a message: '✓ Your 1 User(s) have been created successfully. This is the last time these User security credentials will be available for download. You can manage and recreate these credentials any time.' Below this is a 'Hide User Security Credentials' link and a card displaying user details.

User suvenit

Access Key ID: AKIAI7PIVVU32Ezs4X4Q
Secret Access Key: P1W2YQuZpopbBbccctsOlk03iBUESjWL6nNgrpLCb



➤ Deleting user

The screenshot shows the AWS IAM 'Users' page. On the left, there's a sidebar with options: Dashboard, Details, Groups, Users (which is selected and highlighted in orange), Roles, Policies, Identity Providers, and Account Settings. The main area has a 'Create New Users' button and a 'User Actions' dropdown. The 'User Actions' dropdown menu is open, showing several options: Add User to Groups, Delete User (which is highlighted in orange), Manage Access Keys, Manage Password, Manage Signing Certificates, Manage MFA Device, and Remove User from Groups.

➤ Managing user password

This screenshot is similar to the previous one, showing the AWS IAM 'Users' page. The sidebar and main interface are identical, but the 'User Actions' dropdown menu is different. In this version, the 'Manage Password' option is highlighted in orange, indicating it is the selected action for the selected user 'suvenit'.



AWS Services Edit

Manage Password

Users who will be using the AWS Management Console require a password. Select from the options below to manage the password for user suvenit.

- Assign an auto-generated password
- Assign a custom password

Require user to create a new password at next sign-in

AWS Services Edit

Manage Password

Your password has been created successfully.

This is the last time these User security credentials will be available for download.

You can manage and recreate these credentials any time.

[▼ Hide User Security Credentials](#)

suvenit
Password: #dvzsC{VprCq

AWS Services Edit

Dashboard

Details

Groups

Users

Roles

Policies

Identity Providers

Account Settings

Create New Users

User Actions ▾

Add User to Groups

Delete User

Manage Access Keys

Manage Password

Manage Signing Certificates

Manage MFA Device

Remove User from Groups

AWS Services Edit

Manage Password

Users who will be using the AWS Management Console require a password. Select from the options below to manage the password for user suvenit.

- Keep existing password
- Replace existing password with new auto-generated password
- Replace existing password with new custom password
- Remove existing password

Require user to create a new password at next sign-in



➤ Managing access Keys

The screenshot shows the AWS IAM 'Users' page. The 'User Actions' dropdown is open, showing options: 'Add User to Groups', 'Delete User', 'Manage Access Keys' (which is highlighted), 'Manage Password', 'Manage Signing Certificates', 'Manage MFA Device', and 'Remove User from Groups'. Below this, a modal window titled 'Manage Access Keys' is displayed. It contains a message: 'Use access keys to make secure REST or Query protocol requests to any AWS service API.' A table lists two access keys:

Access Key ID	Created	Last Used	Last Used Service	Last Used Region	Status
AKIAI7PIVVU32EZS4X4Q	2015-08-19 17:18 EDT	N/A	N/A	N/A	Active (Make Inactive Delete)
AKIAIMUB3YSFP2BOJMCA	2015-08-19 17:25 EDT	N/A	N/A	N/A	Active (Make Inactive Delete)

Note: For your protection, you should never share your secret keys with anyone. In addition, industry best practice recommends frequent key rotation.

➤ Learn more about Access Keys

Cancel **Create Access Key**



AWS Services Edit

Dashboard

Details

Groups

Users

Roles

Policies

Identity Providers

Account Settings

Create New Users

User Actions

Search



User Name



Naresh

Groups

Password



suvenit

0



suvenit

AWS Services Edit

Dashboard

Details

Groups

Users

Roles

Policies

Identity Providers

Account Settings

Credential Report

Encryption Keys

IAM > Users > suvenit

Summary

User ARN:

arn:aws:iam::179583788394:user/suvenit

Has Password:

Yes

Groups (for this user):

0

Path:

/

Creation Time:

2015-08-19 17:18 EDT

Groups

Permissions

Managed Policies

There are no managed policies attached to this user.

Attach Policy

Inline Policies

Security Credentials



AWS Services Edit Naresh Global Support

Dashboard Details Groups **Users** Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

Security Credentials

Access Keys

Use access keys to make secure REST or Query protocol requests to any AWS service API. For your protection, you should never share your secret keys with anyone. In addition, industry best practice recommends frequent key rotation. Learn more about Access Keys

Create Access Key

Access Key ID	Created	Last Used	Last Used Service	Last Used Region	Status	Actions
AKIAI7PIVU32EZA4X4Q	2015-08-19 17:18 EDT	N/A	N/A	N/A	Active	Make Inactive Delete
AKIAIMUB3YSFP2BOJMCA	2015-08-19 17:25 EDT	N/A	N/A	N/A	Active	Make Inactive Delete

Sign-In Credentials

User Name suvenit [Manage Password](#)

Password Yes

Last Used no_information

Multi-Factor Authentication Device [Manage MFA Device](#)

Signing Certificates None [Manage Signing Certificates](#)

➤ User default password settings

AWS Services Edit

Dashboard Details Groups Users Roles Policies Identity Providers **Account Settings** Credential Report Encryption Keys

more information about password policies, go to [Managing Passwords in Using IAM](#).

Modify your existing password policy below.

Minimum password length:

Require at least one uppercase letter [i](#)

Require at least one lowercase letter [i](#)

Require at least one number [i](#)

Require at least one non-alphanumeric character [i](#)

Allow users to change their own password [i](#)

Enable password expiration [i](#)

Password expiration period (in days):

Prevent password reuse [i](#)

Number of passwords to remember:

Password expiration requires administrator reset [i](#)

Apply password policy **Delete password policy**

➤ Understanding roles



AWS Services Edit

Dashboard Details Groups Users Roles Policies

Create New Role Role Actions

Search

Role Name

No records found.

➤ Creating roles

AWS Services Edit

Create Role Step 1: Set Role Name Step 2: Select Role Type Step 3: Establish Trust Step 4: Attach Policy Step 5: Review

Set Role Name

Enter a role name. You cannot edit the role name after the role is created.

Role Name sysadmin

Maximum 64 characters. Use alphanumeric and '+-,@-' characters

AWS Services Edit Naresh Global

Create Role Step 1: Set Role Name Step 2: Select Role Type Step 3: Establish Trust Step 4: Attach Policy Step 5: Review

Select Role Type

AWS Service Roles

Amazon EC2 Allows EC2 instances to call AWS services on your behalf. Select

AWS Directory Service Allows AWS Directory Service to manage access for existing directory users and groups to AWS services. Select

AWS Lambda Allows Lambda Function to call AWS services on your behalf. Select

AWS Config Allows AWS Config to call AWS services and collect resource configurations on your behalf. Select

AWS SWF Allows SWF workflows to invoke Lambda functions on your behalf. Select

➤ Attaching policy to role



AWS Services Edit

Create Role

Step 1: Set Role Name

Step 2: Select Role Type

Step 3: Establish Trust

Step 4: Attach Policy

Step 5: Review

Attach Policy

Select up to two policies to attach to the role.

Filter: Policy Type		Search
	Policy Name	Attached
<input type="checkbox"/>	AmazonSNSRole	1
<input type="checkbox"/>	AdministratorAccess	0

AWS Services Edit

Create Role

Step 1: Set Role Name

Step 2: Select Role Type

Step 3: Establish Trust

Step 4: Attach Policy

Step 5: Review

Review

Review the following role information. To edit the role, click an edit link, or click **Create Role** to finish.

Role Name	sysadmin	Edit Role Name
Role ARN	arn:aws:iam::179583788394:role/sysadmin	
Trusted Entities	The identity provider(s) ec2.amazonaws.com	
Policies	arn:aws:iam::aws:policy/service-role/AmazonSNSRole	Edit Policies

AWS Services Edit

Dashboard

Details

Groups

Users

Roles

Policies

Create New Role

Role Actions

Search

<input type="checkbox"/>	Role Name
<input type="checkbox"/>	sysadmin



AWS Services Edit

Dashboard Details Groups Users Roles

IAM > Roles > sysadmin

Summary The role Amazon Resource Name (ARN) is a globally-unique identifier for roles which includes your AWS Account ID and role name

Role ARN arn:aws:iam::179583788394:role/sysadmin
Instance Profile arn:aws:iam::179583788394:instance-profile/sysadmin
Path /
Creation Time 2015-08-19 18:04 EDT

AWS Services Edit

Dashboard Details Groups Users Roles

Create New Role Role Actions ▾

Search

Role Name ▾
 sysadmin

➤ Deleting role

Attach Policy

Policy Name	Actions
AmazonSNSRole	Show Policy Detach Policy Simulate Policy



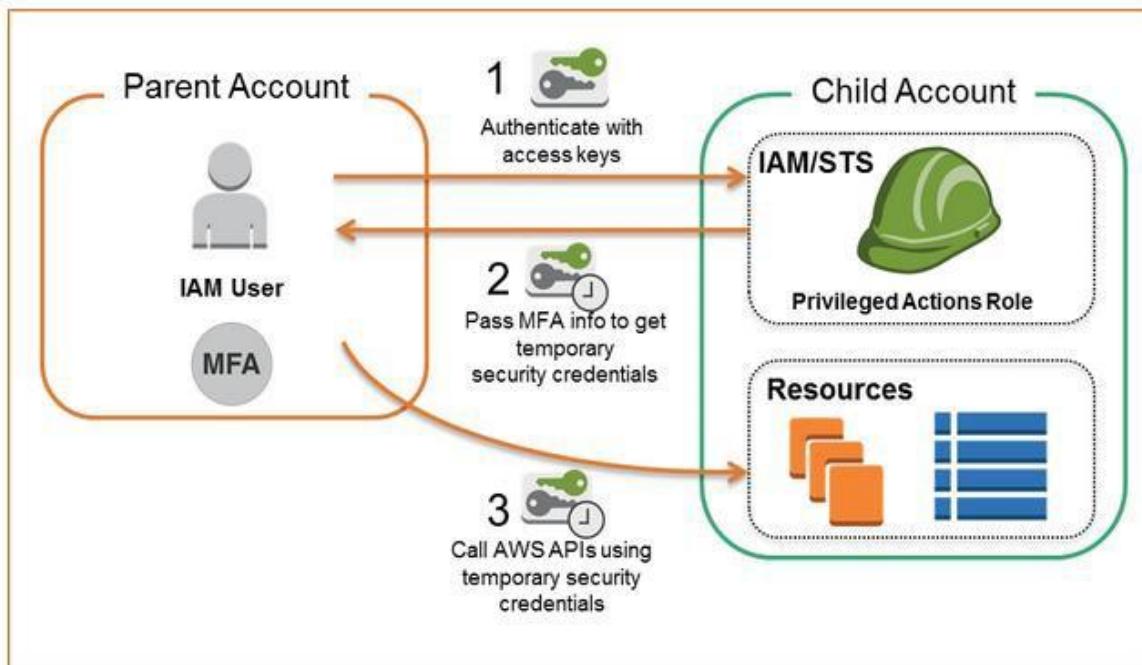
➤ Multi Factor Authentication [MFA]

Multi-Factor Authentication, or MFA.
MFA provides additional security by
requiring users to use a password and an
authentication code from an external device



MFA is especially recommended for the AWS root accounts and account with administrator permissions since they have access to all your AWS resources

➤ Login process





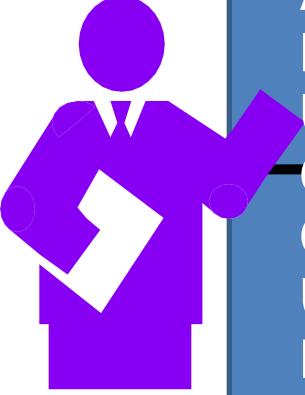
Lab activity





MODULE5# -Simple Storage Solution[S3]

Agenda



- Understanding Storage in AWS**
- Different types of Storages in AWS**
- What is Amazon S3**
- Functions and concepts of S3**
- Advantages and disadvantage of S3**
- Requirement for S3**
- Understanding pricing of S3**
- Creating and managing S3 Buckets**
- Creating and managing Objects in S3**
- Uploading Objects to S3**
- Deleting Buckets**
- Deleting objects**



Storages available in AWS?

Storage is an important role AWS and find the below storages available in AWS and uses

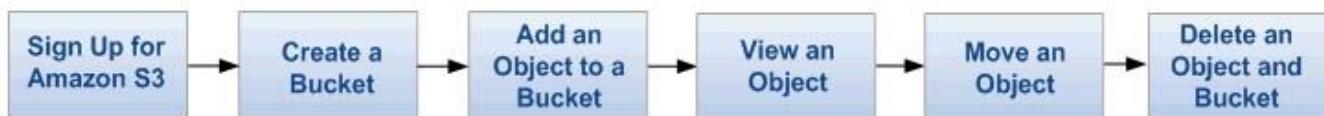
	Amazon S3	Scalable storage in the cloud
	Amazon Glacier	Low-cost archive storage in the cloud
	Amazon EBS	Persistent block storage volumes for Amazon EC2 virtual machines
	Amazon EC2 Instance Storage	Temporary block storage volumes for Amazon EC2 virtual machines
	AWS Import/Export	Large volume data transfer
	AWS Storage Gateway	Integrates on-premises IT environments with cloud storage
	Amazon CloudFront	Global content delivery network (CDN)
	Amazon SQS	Message queue service
	Amazon RDS	Managed relational database server for MySQL, Oracle, and Microsoft SQL Server
	Amazon DynamoDB	Fast, predictable, highly-scalable NoSQL data store
	Amazon ElastiCache	In-memory caching service
	Amazon Redshift	Fast, powerful, full-managed, petabyte-scale data warehouse service
	Databases on Amazon EC2	Self-managed database on an Amazon EC2 instance



What is AWS S3?

Amazon Simple Storage Service (Amazon S3) is storage for the Internet. We can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web.

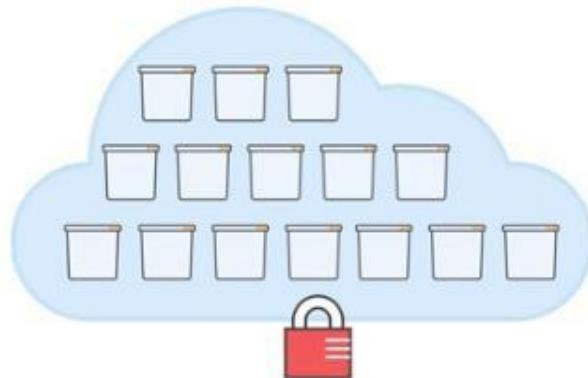
We can accomplish these tasks using the AWS Management Console, which is a simple and intuitive web interface.



S3 is designed for

- Remote data storage
- Low cost, pay-as-you go
- No up-front costs
- High-availability
- High bandwidth

Uses Amazon's own network



Features

- 99.99999999% durability (on a given year)
- 99.99% availability (on a given year)
- designed to support concurrent data failure in two physical facilities.

March 2006: Amazon launched Simple Storage Service (S3)



Functions & concepts of S3

- Allows unlimited storage of objects(files) containing of 1 byte to 5 gigabytes each.
- Objects consist of the raw object data and metadata.
- Objects are stored and retrieved using a developer-assigned key.
- Data are kept secured from unauthorised access through authentication mechanism.
- Objects can be made available to public by the http or bittorrent protocol.

- All objects are stored in buckets.
- A bucket is simply a container for objects. It is used to partition the namespace of objects at the highest level.
- Buckets are similar to Internet domain names. They are accessed via **bucketname.s3.amazonaws.com**.
- Each developer account has a limit of 100 buckets.



- A key is the unique identifier for an object within a bucket.
- A bucket and a key together uniquely identify each object in S3. Every object can be addressed through bucket and key combination.
- For example, if your bucket name is **mybucket** and key is **myhomepage.html**, the URL for the object will be <http://mybucket.s3.amazonaws.com/myhome page.html>

Advantages of S3?

- Scalability. The amount of storage & bandwidth you need can scale as you like without any configuration changes needed.
- Availability, speed, throughput, capacity, and robustness is not affected even if you gain 10,000 users overnight.
- Unlimited storage. You pay as you go.
- Inexpensive and no capital outlay. Great for startups!
- Data is accessible from any location.
- Since it is based on the Amazon infrastructure, it is probably more reliable than other cheap data storage providers.



Use cases

- Asset storage and CDN
- Data storage
- Static site
- Backups
- Mobile storage backend
- File distribution

Pricing

- Charges for using S3 is based on the location of your buckets.
- You are billed according to storage(average), data transfer in and out and the number of requests per month.
- There is no minimum fee to use S3, you pay for only what you use.
- You can view your current charges incurred almost immediately on the S3 portal.



What is Bucket?

A bucket is the basic storage unit in Amazon S3. It is a single-level container (no hierarchy supported), and it's based on key-object associations.

Details

- Sub-folders are supported through specific Content-Type headers and a “substring mechanism”
- Upload and download are easy
- Renaming of folders; navigation of complex hierarchy can be problematic.

Bucket is a

- Collection of objects
- Globally unique id
- a-z A-Z 0-9 . –
- Max 100 buckets/user
- No limit on number of objects

Buckets

Equivalent of directories

Single, common namespace across S3

But bucketNames can include “/”, eg

mgateway/backups/presentations

Objects

Equivalent of files

Up to 5Gb in size

Identified by key (== filename)



Best practices on naming

- DNS compatible
- FQDN ○ Allows for vhost ○ watch out for SSL:
no dots :-(

Objects

- Blob
- Don't care about file formats
- Metadata can be added (like mimetype)
- Maximum 5 TB/object

How to Access AWS S3 storage?

Accessible using simple HTTP URLs

`http://s3.amazonaws.com/bucket/key`

`http://bucket.s3.amazonaws.com/key`

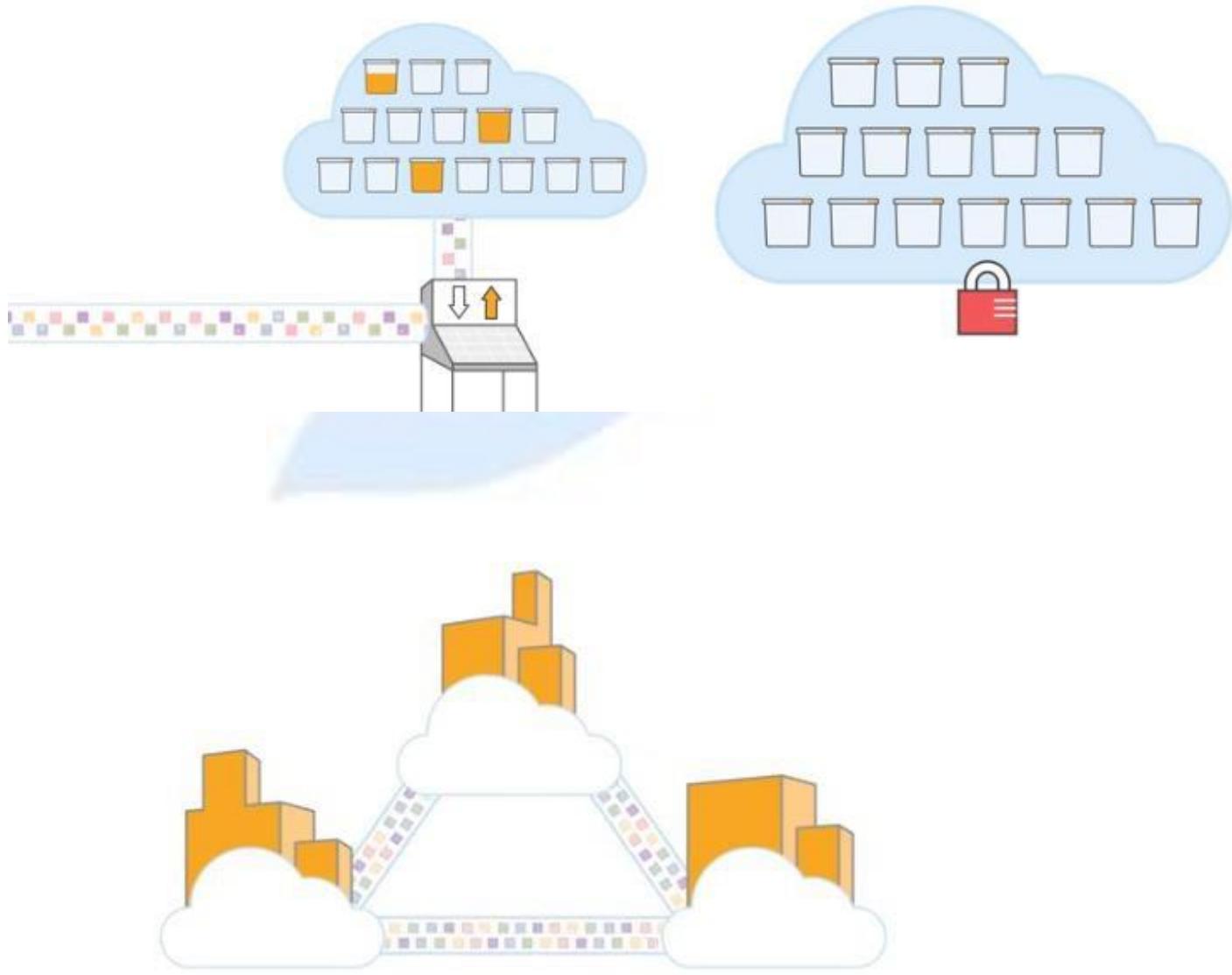
`http://bucket/key`

where *bucket* is a DNS CNAME record pointing to
`s3.amazonaws.com`)

- Use Amazon AWS Management Console
- Use Cloudberry Explorer

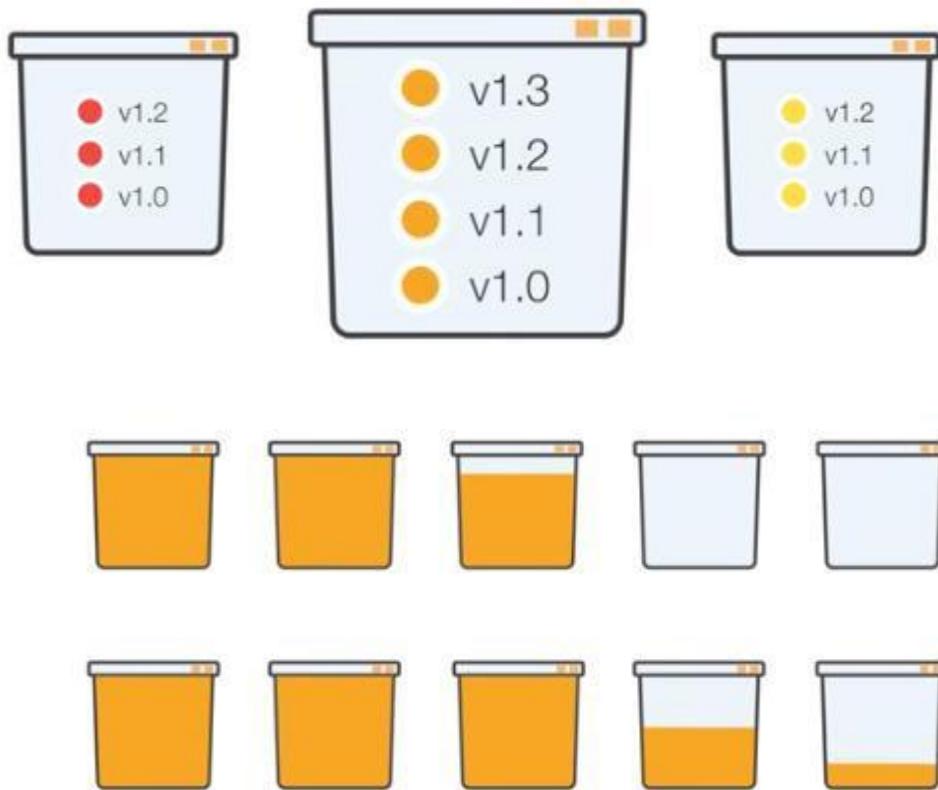


- S3 allows you to specify an Access Control List for every object in the database
- You can set permissions for the owner, for authenticated user, for specific users (e-mail and Amazon ID) and for everybody.
- It's even possible to create public URLs that expire at a given date





It maintains the versions of Objects stored in S3 and recover in case of data loss



S3 Range of Classes

S3 Standard class for frequently accessed data





We can also setup auto policy to migrate data from one class to another class like standards to Glacier etc..





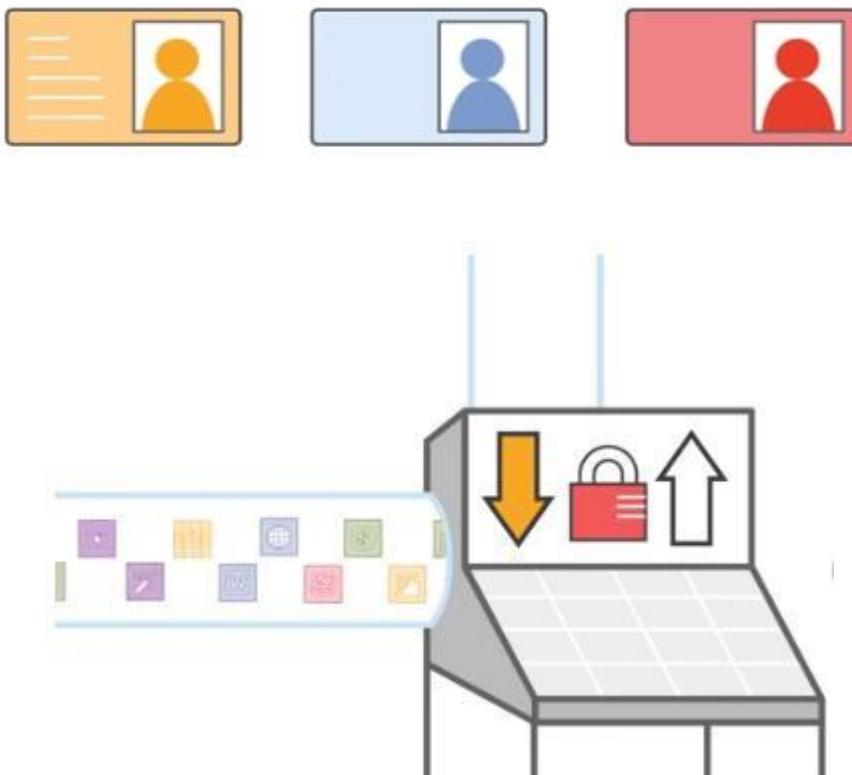
Security in S3

Access Control List

Bucket Policy

Key Authentication

S3 also offer SSL encryption for data upload and download





To sign up for Amazon S3

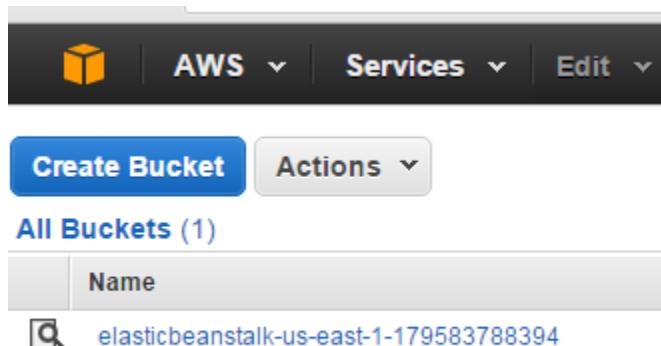
1. Go to <http://aws.amazon.com/s3> and click **Sign Up**.
2. Follow the on-screen instructions.

Once we login to AWS console click on S3 from Storage and content delivery

Storage & Content Delivery

-  **S3**
Scalable Storage in the Cloud
-  **CloudFront**
Global Content Delivery Network
-  **Elastic File System** PREVIEW
Fully Managed File System for EC2
-  **Glacier**
Archive Storage in the Cloud
-  **Import/Export Snowball**
Large Scale Data Transport
-  **Storage Gateway**
Hybrid Storage Integration

We are in S3 dashboard.



Name
 elasticbeanstalk-us-east-1-179583788394

We are ready to create a bucket using the AWS Management Console. Every object in Amazon S3 is stored in a bucket. Before we can store data in Amazon S3, We must create a bucket.

Note: You are not charged for creating a bucket; you are charged only for storing objects in the bucket and for transferring objects in and out of the bucket.



To create a bucket

A bucket is a logical unit of storage in Amazon Web Services (AWS) object storage service, Simple Storage Solution S3. Buckets are used to store objects, which consist of data and metadata that describes the data.

1. Sign into the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3>.
2. Click **Create Bucket**.

The screenshot shows a modal dialog box titled "Create a Bucket - Select a Bucket Name and Region". At the top right is a "Cancel" button. The main content area contains a descriptive text about buckets and their regions, followed by input fields for "Bucket Name" (a text input box) and "Region" (a dropdown menu set to "Oregon"). At the bottom are three buttons: "Set Up Logging >" (disabled), "Create" (highlighted in blue), and "Cancel".

3. In the **Create a Bucket** dialog box, in the **Bucket Name** box, enter a bucket name.

The bucket name you choose must be unique across all existing bucket names in Amazon S3.

Note: After we create a bucket, we cannot change its name. In addition, the bucket name is visible in the URL that points to the objects stored in the bucket. Ensure that the bucket name we choose is appropriate.

4. In the **Region** box, select a region. For this exercise, select **Oregon** from the drop-down list. We can choose a region to optimize latency, minimize costs, or address regulatory requirements. Objects stored in a region never leave that region unless you explicitly transfer them to another region.



Create a Bucket - Select a Bucket Name and Region

Cancel

A bucket is a container for objects stored in Amazon S3. When creating a bucket, you can choose a Region to optimize for latency, minimize costs, or address regulatory requirements. For more information regarding bucket naming conventions, please visit the [Amazon S3 documentation](#).

Bucket Name: suvenit

Region: Select a Region

US Standard

Oregon

Northern California

Ireland

Singapore

Tokyo

Sydney

Sao Paulo

Frankfurt

Seoul

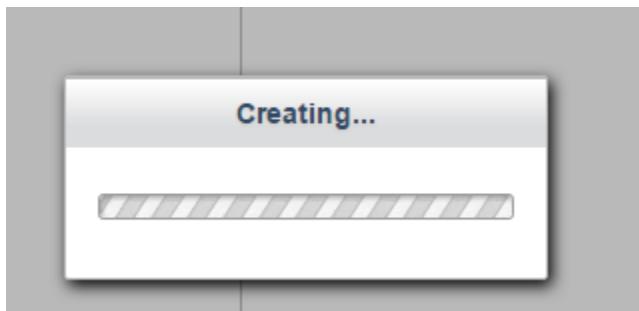
[Set Up Logging >](#)

Create

[Cancel](#)

5. Click **Create**.

When Amazon S3 successfully creates your bucket, the console displays your empty bucket in the **Buckets** panel.





Bucket “suvenit” is created

The screenshot shows the AWS S3 console. At the top, there's a navigation bar with the AWS logo, 'AWS', 'Services', and 'Edit' dropdowns. Below that is a blue button labeled 'Create Bucket'. To its right is a 'Actions' dropdown. Underneath, it says 'All Buckets (1)'. A table follows, with columns for 'Name' and 'Actions'. It lists two buckets: 'elasticbeanstalk-us-east-1-179583788394' and 'suvenit'. The 'suvenit' row has a magnifying glass icon next to it and is highlighted with a green box.

We can create, delete and manage the buckets by action button.

The screenshot shows the AWS S3 console with the same layout as the previous one. A red arrow points from the explanatory text to the 'Actions' dropdown menu, which is now open. The menu contains several options: 'Create Bucket...', 'Delete Bucket', 'Empty Bucket', 'Refresh', 'Paste Into', 'Manage CloudTrail Logs...', and 'Properties'. The background shows the same list of buckets as the first screenshot.



Once we click on Bucket properties, we can see all below properties.

Bucket: suvenit

Bucket: suvenit
Region: Oregon
Creation Date: Mon Jan 18 17:55:25 GMT-500 2016
Owner: suvenit

▼ Permissions

You can control access to the bucket and its contents using access policies. Learn more.

Grantee: suvenit

List Upload/Delete View Permissions Edit Permissions

x

Add more permissions

Add bucket policy

Add CORS Configuration

Save

Cancel

► Static Website Hosting

► Logging

► Events

► Versioning

► Lifecycle

► Cross-Region Replication

► Tags

► Requester Pays

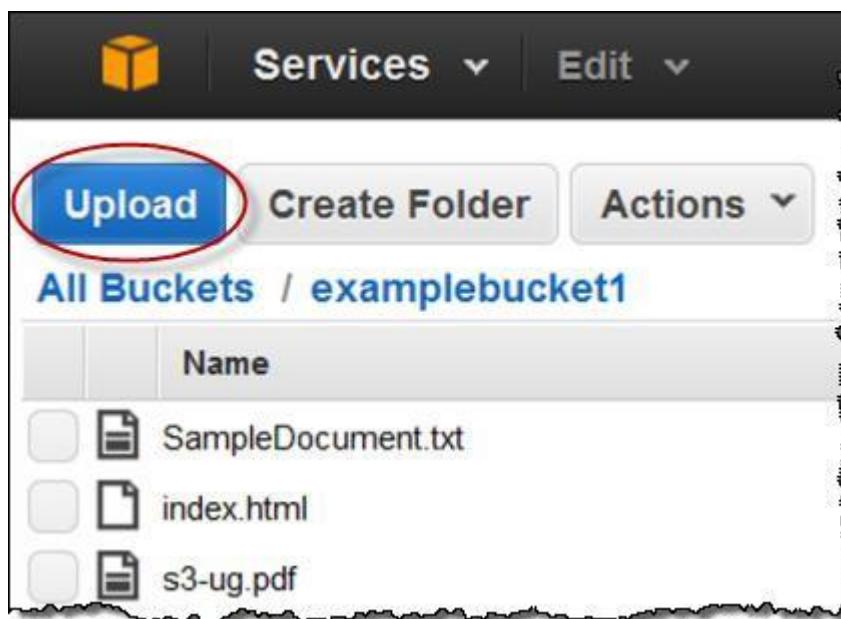


Add an Object to a Bucket.

An object can be any kind of file: a text file, a photo, a video and so forth. When you add a file to Amazon S3, you have the option of including metadata with the file and setting permissions to control access to the file.

To upload an object

1. In the [Amazon S3 console](#), click the name of bucket that you want to upload an object to and then click **Upload**.



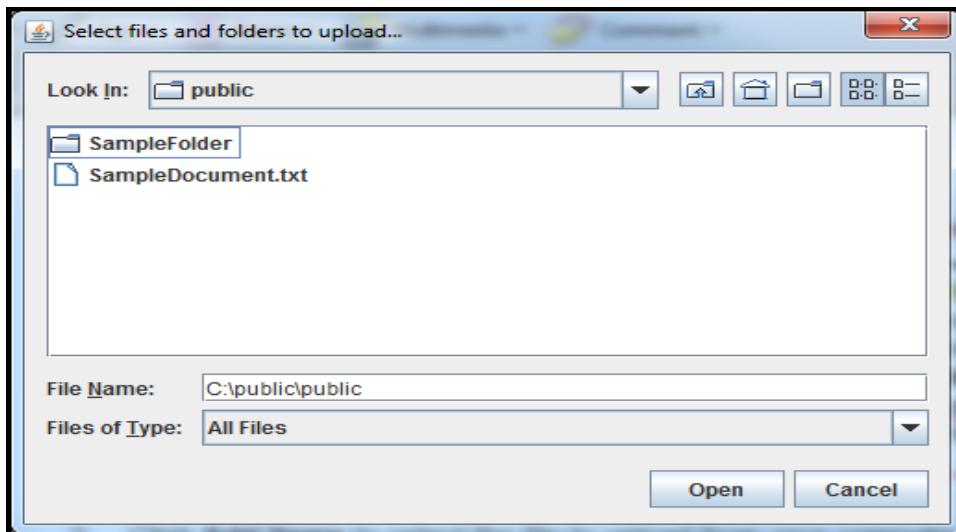
2. In the **Upload - Select Files** wizard, if you want to upload an entire folder, you must click **Enable Enhanced Uploader** to install the necessary Java applet. You only need to do this once per console session.



3. Click **Add Files**.

A file selection dialog box opens:

- If you enabled the advanced uploader in step 2, you see a Java dialog box titled **Select files and folders to upload**, as shown.
- If not, you see the **File Upload** dialog box associated with your operating system.





4. Select the file that you want to upload and then click **Open**.
5. Click **Start Upload**.

You can watch the progress of the upload from within the **Transfer** panel.

6. Click on action button to manage an objects

The screenshot shows the AWS S3 console interface. A context menu is open over a file named "12079239_598642230276565_6081589537051098074_n(2).jpg" in the "suvenit" bucket. The menu includes options like Open, Download, Create Folder..., Upload, Make Public, Rename, Delete, Initiate Restore, Cut, Copy, Paste, and Properties.

7. Click on properties to see the object properties and url to access.

Object: 12079239_598642230276565_6081589537051098074_n(2).jpg ×

Bucket: suvenit
Name: 12079239_598642230276565_6081589537051098074_n(2).jpg
Link: [https://s3-us-west-2.amazonaws.com/suvenit/12079239_598642230276565_6081589537051098074_n+\(2\).jpg](https://s3-us-west-2.amazonaws.com/suvenit/12079239_598642230276565_6081589537051098074_n+(2).jpg)
Size: 39494
Last Modified: Mon Jan 18 18:17:27 GMT-500 2016
Owner: suvenit
ETag: a31f9f02d9bbb944848a46d646d07ca0
Expiry Date: None
Expiration Rule: N/A

▼ Details

Storage Class: Standard Standard - Infrequent Access Reduced Redundancy

Server Side Encryption: None AES-256

Save Cancel

▶ Permissions

▶ Metadata



Click on Details to change the Storage class

▼ Details

Storage Class: Standard Standard - Infrequent Access Reduced Redundancy

Server Side Encryption: None AES-256

Save

Cancel

Click on permission to see and to change permissions.

▼ Permissions

You can control access to the bucket and its contents using access policies. [Learn more](#).

Grantee: suvenit

Open/Download

View Permissions

Edit Permissions

x

+ Add more permissions

Save

Cancel



MODULE6# -Virtual Private Cloud

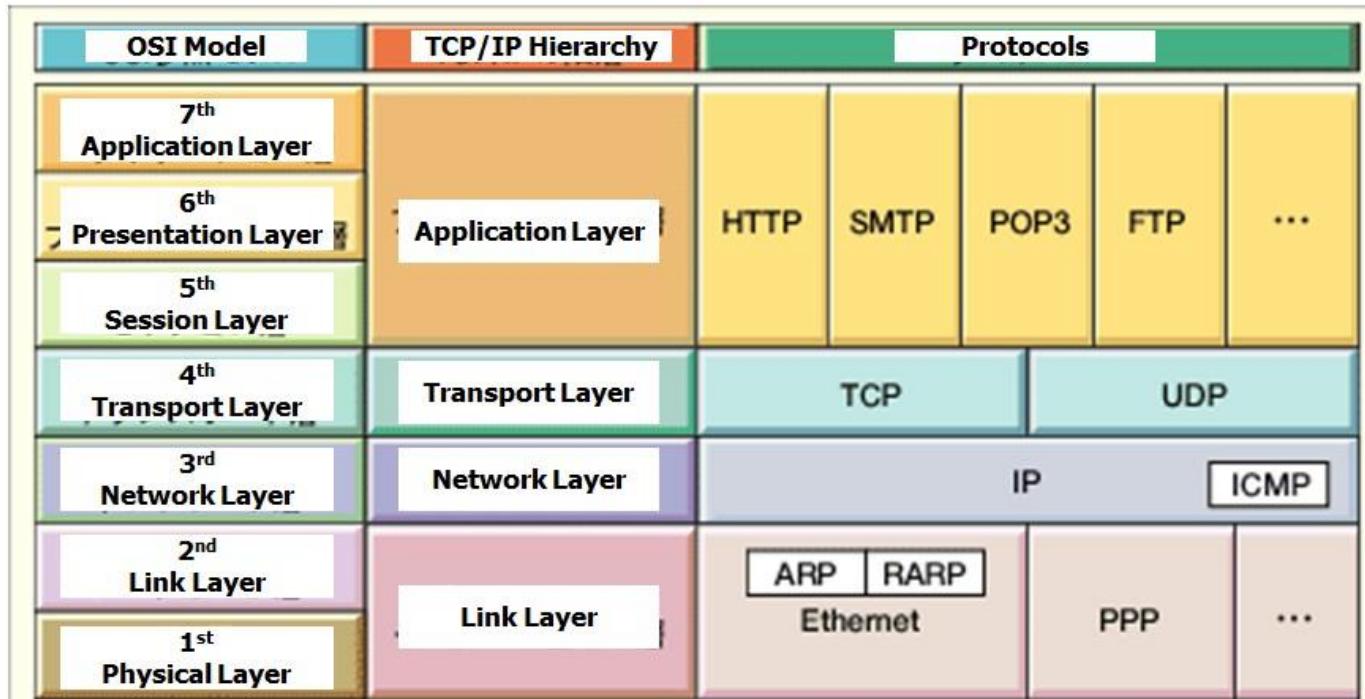
Agenda



- Understanding Networking(TCP/IP, IP Address etc)**
- Understanding network in AWS**
- Subnet and AWS Communications**
- Internet gateway & Routing tables**
- Security groups and ACL's**
- CloudFront, Route53, VPN's, VPG's and direct connect**
- Load Balancing**
- Auto Scaling**



Understanding the TCP/IP model



- Link Layer : includes device driver and network interface card
Network Layer : handles the movement of packets, i.e. Routing
Transport Layer : provides a reliable flow of data between two hosts
Application Layer: handles the details of the particular application

IP

Responsible for end to end transmission, Sends data in individual packets, Maximum size of packet is determined by the networks
Fragmented if too large
Unreliable Packets might be lost, corrupted, duplicated, delivered out of order



IP addresses

4 bytes

e.g. 163.1.125.98

Each device normally gets one (or more)

In theory there are about 4 billion available

Routing

How does a device know where to send a packet?

All devices need to know what IP addresses are on directly attached networks If the destination is on a local network, send it directly there

If the destination address isn't local

Most non-router devices just send everything to a single local router
Routers needs to know which network corresponds to each possible IP address

Allocation of addresses

Controlled centrally by ICANN

-Fairly strict rules on further delegation to avoid wastage

Have to demonstrate actual need for them

Organizations that got in early have bigger allocations than they really need

IP packets

Source and destination addresses

Protocol number

1 = ICMP, 6 = TCP, 17 = UDP

Various options

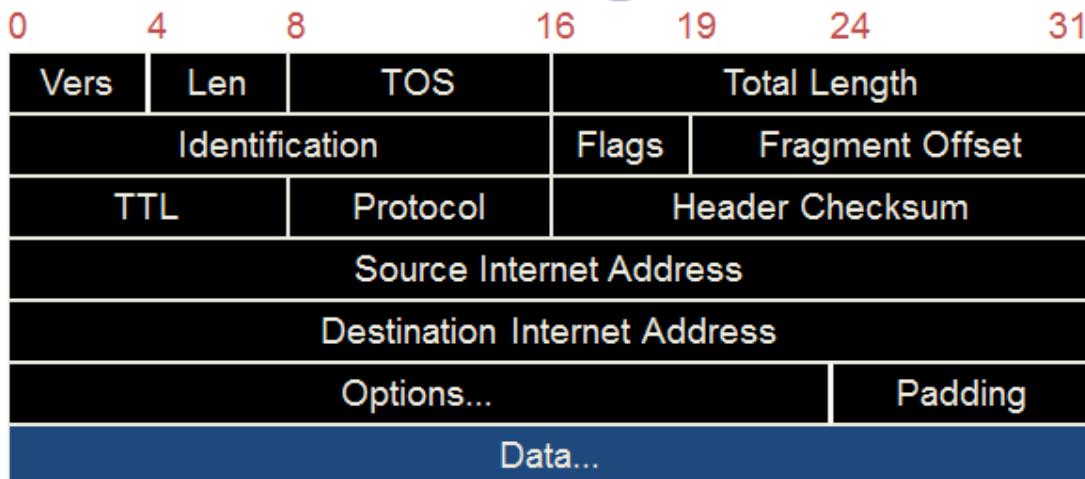
e.g. to control fragmentation

Time to live (TTL)

Prevent routing loops



IP Datagram



Field Purpose

Vers IP version number

Len Length of IP header (4 octet units)

TOS Type of Service

T. Length Length of entire datagram (octets)

Ident. IP datagram ID (for frag/reassembly)

Flags Don't/More fragments

Frag Off Fragment Offset

Field Purpose

TTL Time To Live - Max # of hops

Protocol Higher level protocol (1=ICMP,
6=TCP, 17=UDP)

Checksum Checksum for the IP header

Source IA Originator's Internet Address

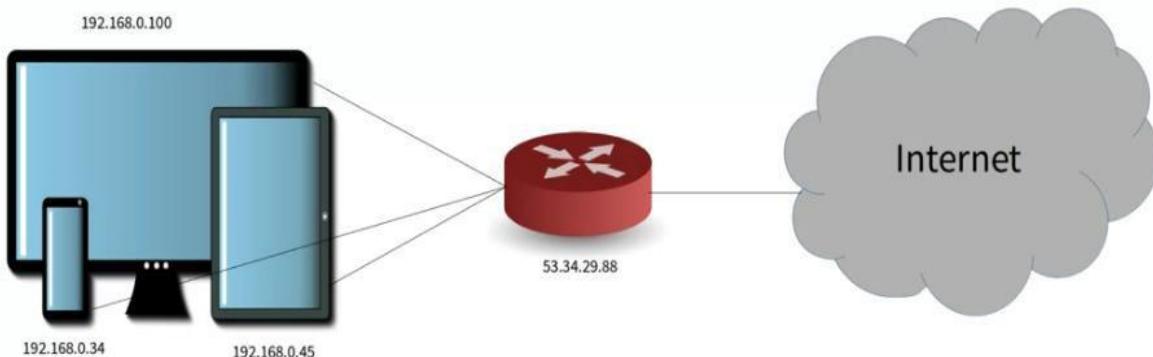
Dest. IA Final Destination Internet Address

Options Source route, time stamp, etc.

Data... Higher level protocol data



NAT Translation



Reserved NAT Ranges:

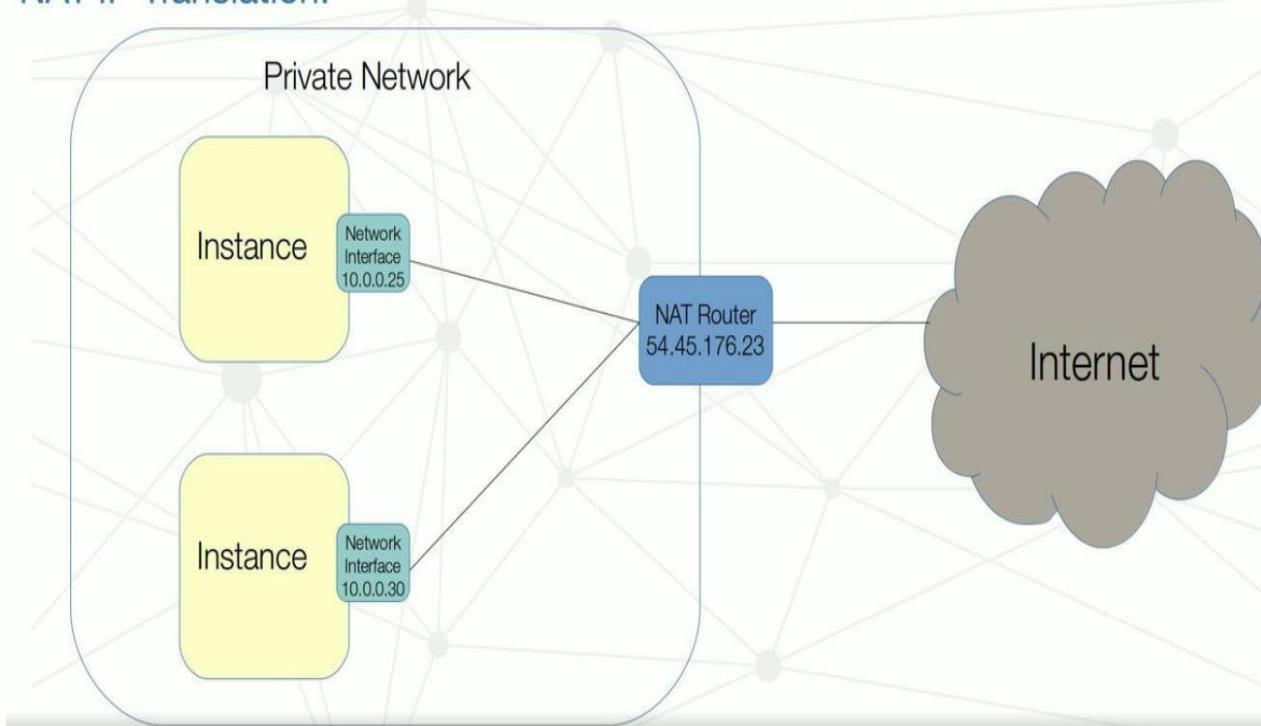
Start:	End:	Number of addresses:
10.0.0.0	10.255.255.255	16,777,216
172.16.0.0	172.31.255.255	1,048,576
192.168.0.0	192.168.255.255	65,536

EC2- Classic VS. VPC Networks

<u>EC2-Classic</u>	<u>VPC</u>
part of AWS network	discrete networks
Instance bound to group	apply new group to running instance
instance and group must be from the same region	able to attach group from any region



NAT IP Translation:



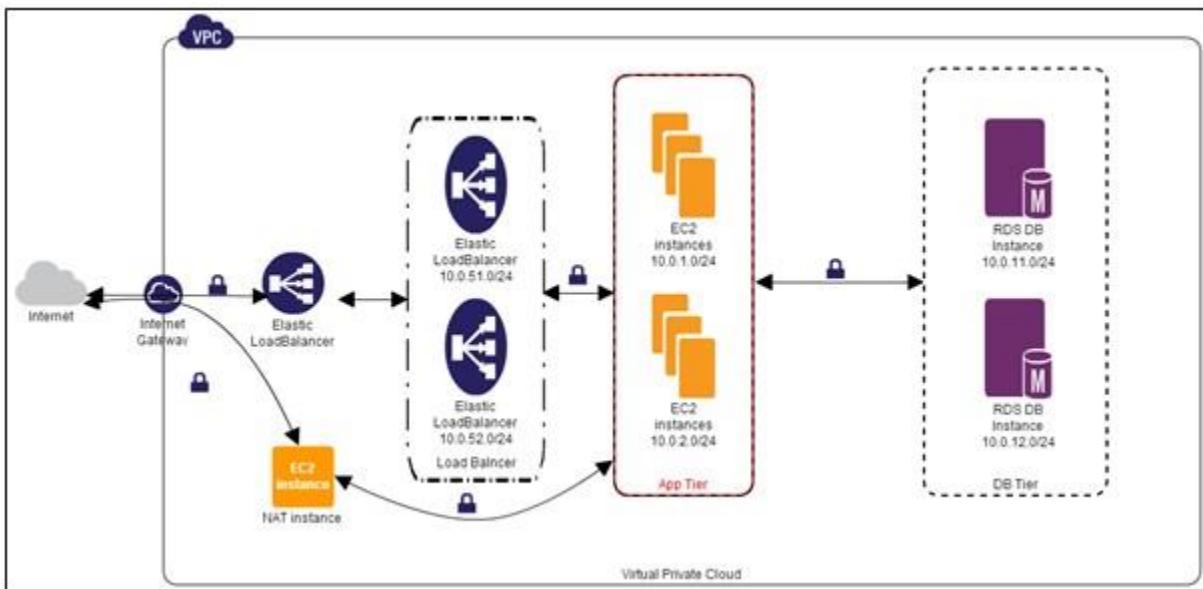
Reserved NAT Ranges:

Start	End	No. of addresses
10.0.0.0	10.255.255.255	16777216
172.16.0.0	172.31.255.255	1048576
192.168.0.0	192.168.255.255	65536



VPC Best Configuration Practices

A *virtual private cloud* (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud. We can launch your AWS resources, such as Amazon EC2 instances, into your VPC. We can configure your VPC; we can select its IP address range, create subnets, and configure route tables, network gateways, and security settings.



1. Create VPC

Log in to the AWS console.

Navigate to Services->VPC->Your VPCs.

Click **Create VPC**.

When you create a VPC, you specify a set of IP addresses in the form of a Classless Inter-Domain Routing (CIDR) block (for example, 10.0.0.0/16). For more information about CIDR notation and what "/16" means, see [Classless Inter-Domain Routing](#).

You can assign a single CIDR block to a VPC. The allowed block size is between a /28 netmask and /16 netmask. In other words, the VPC can contain from 16 to 65,536 IP addresses.



You cannot change a VPC's size after creating it. If your VPC is too small for your needs, you'll need to terminate all of the instances in the VPC, delete it, and then create a new, larger VPC.

To create your VPC, go to the Create VPC dialog box, specify the following VPC details and then click **-Yes, Create!!**.

CIDR Block: Specify the CIDR block for your VPC. I prefer 10.0.0.0/16.

Tenancy: Default tenancy: This is for running instances on shared hardware and is free of charge.

Dedicated Tenancy: This is for running your instances on single-tenant hardware. A \$2 fee applies for each hour in which any dedicated instance is running in a region.

Create VPC Cancel

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. Please use the Classless Inter-Domain Routing (CIDR) block format to specify your VPC's contiguous IP address range, for example, 10.0.0.0/16. Please note that you can create a VPC no larger than /16.

CIDR Block: (e.g. 10.0.0.0/16)

Tenancy:

Cancel Yes, Create

2. Create Subnets

In the navigation pane click on **-Subnets!!**.

Click **-Create Subnet!!**.

Before we create a subnet, let's understand the best practices for creating them.

You should create subnets across multiple availability zones, with each subnet residing within a single zone. Creating subnets in and launching instances across multiple availability zones will ensure a high-availability environment.

When creating separate subnets for ELB, EC2 and RDS instances, each tier should have at least 2 subnets across availability zones.



For this example, we created subnets using zones us-east1b and us-east-1d. These subnets are called -private subnets because the instances we launch are not accessible from the Internet. In other words, these instances don't have a public IP unless you assign an EIP.

App Tier: 10.0.1.0/24(zone-b), 10.0.2.0/24(zone-d)

ELB: 10.0.51.0/24(zone-b), 10.0.52.0/24(zone-d)

Database (RDS): 10.0.11.0/24(zone-b), 10.0.12.0/24(zone-d)

Always choose the same availability zones for all tiers. For example, if you choose two zones for high availability and use us-east-1a and us-east1b, then maintain those same 1a and 1b zones for all tiers. This will minimize data transfer charges because data transfers between instances within the same availability zone are free.

Create Subnet

Please use the CIDR format to specify your subnet's IP address block (e.g., 10.0.0.0/24). Please note that block sizes must be between a /16 netmask and /28 netmask. Also, please note that a subnet can be the same size as your VPC.

VPC:

Availability Zone:

CIDR Block: (e.g. 10.0.0.0/24)

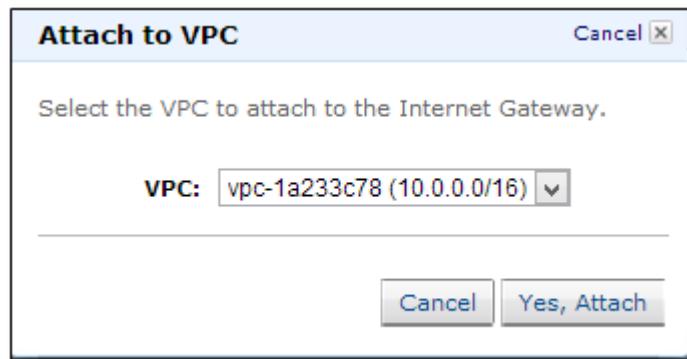
3. Create Internet Gateway

By default, instances that are launched into a VPC can't communicate with the Internet. However, you can enable Internet access by attaching an Internet gateway to the VPC.

Go to Internet Gateways in the navigation pane and click **Create Internet Gateway**.



Now attach the gateway to a VPC by right clicking on –VPC|| and selecting –**Attach to VPC||**.



4. Create Route Tables

A route table contains a set of rules, called routes, that determine where network traffic is directed.

Each subnet in your VPC must be associated with a route table that will control that subnet's routing. You can associate multiple subnets with a single route table; however, you can only associate a subnet with one route table.

Creating a VPC automatically creates a main route table which, by default, enables the instances in your VPC to communicate with one other.

Go to Route Tables in the navigation pane and click on –**Create Route Table||**.



Create Route Table Cancel

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

VPC: **vpc-1a233c78 (10.0.0.0/16)**

Cancel Yes, Create

As a best practice create separate route tables for each tier. This will provide more control in maintaining the security of each subnet.

Now associate the subnets to the route tables.

Click on one route table and go to the Associations tab.

Select the subnet and click **-Associate**.

1 Route Table selected.

Route Table: rtb-7fc2dc1d

Associations

Subnet	Actions
subnet-ca037fe2 (10.0.1.0/24)	Associate

The following subnets have not been associated with any route tables and are therefore using the Main table routes:

- subnet-ca037fe2 (10.0.1.0/24)

Associate each tier's subnets separately to the dedicated route table.

Create 3 new route tables:

1. **ELB Route table**—Associate 10.0.51.0/24 and 10.0.52.0/24.
2. **APP route table**—Associate 10.0.1.0/24 and 10.0.2.0/24.
3. **RDS route table**—Associate 10.0.11.0/24 and 10.0.12.0/24.

Do not associate any subnets with the main route table.

Now navigate to the main route table to add a route to allow Internet traffic to the VPC.

Go to Routes and specify the following values:



Destination: 0.0.0.0/0

Target: Select -Internet Gateway- from the dropdown menu.

A screenshot of the AWS Route Table configuration interface. At the top, it says "1 Route Table selected" and "Route Table: rtb-1fc7d97d". Below this are tabs for "Routes", "Associations", "Route Propagation", and "Tags", with "Routes" being the active tab. The main area shows a table with columns: Destination, Target, Status, Propagated, and Actions. There is one row: Destination 10.0.0.0/16, Target local, Status active, Propagated No, and Actions with "Remove" and "Add" buttons.

Destination	Target	Status	Propagated	Actions
10.0.0.0/16	local	active	No	<button>Remove</button>
0.0.0.0	igw-6b0f9c09			<button>Add</button>

5. Create Security Groups

This process is similar to creating an SG (Security Group) in classic EC2.

Create separate security groups for ELB, APP, DB (RDS) and NAT instances.

A screenshot of the "Create Security Group" dialog box. It has fields for "Name" (APP_SG01), "Description" (App Security Group), and "VPC" (vpc-1a233c78). At the bottom are "Cancel" and "Yes, Create" buttons.

Create Security Group

Name: APP_SG01

Description: App Security Group

VPC: vpc-1a233c78

Cancel Yes, Create

1. APP_SG01
2. NAT_SG01
3. ELB_SG01
4. DB_SG01

Allow Inbound rules for ELB, DB and APP to suit your needs. We'll address NAT security group rules later in this post.



6. Create NAT instance

Instances launched into a private subnet in a VPC cannot communicate with the Internet unless you assign a public IP or EIP to the instance. However, assigning a public IP to an instance will allow everyone to initiate inbound Internet traffic.

Using a Network Address Translation (NAT) instance in your VPC enables instances in the private subnet to initiate outbound Internet traffic.

Create a subnet with netmask 10.0.0.0/24 for NAT instance. [Refer to section #2 of this post]. We call this subnet a -public subnet|| and the others -private subnets||. While, technically, there is no difference between public or private subnet, for clarity we call publicly accessible instances public subnets.

Associate this subnet to the main route table. You can also create separate route tables to associate to the subnet. If you do create a separate route table, don't forget to add a route that will allow Internet traffic into the subnet. [Refer to section #4 of this post].

Now navigate to Services->EC2->Launch Instance

In the Launch Wizard select -Community AMIs|| and search for -ami-vpc-natl|. — Select the first AMI from the results list to launch the instance into the VPC created in section #1. Choose the subnet 10.0.0.0/24 and then check the -Assign public IP|| box. You can also assign an EIP, if needed. On the Configure Security Group page, choose -Select an existing security group|| and select the NAT_SG security group that you created earlier.

The screenshot shows the 'Launch Wizard' configuration interface. It includes fields for 'Number of instances' (set to 1), 'Purchasing option' (checkbox for Request Spot Instances), 'Network' (selected network is 'vpc-1a233c78 (10.0.0.0/16)'), 'Subnet' (selected subnet is 'subnet-cc7d01e4(10.0.0.0/24) | us-east-1b'), and 'Public IP' (checkbox for automatically assigning a public IP). There are also buttons for 'Create new VPC' and 'Create new subnet'.

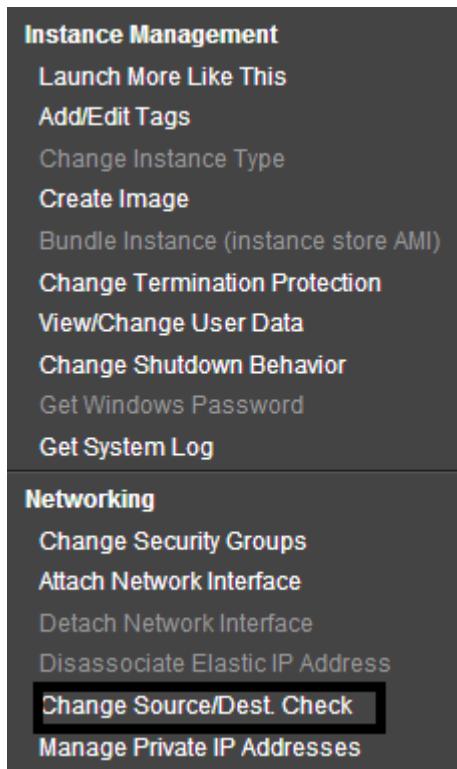
Number of instances	1
Purchasing option	<input type="checkbox"/> Request Spot Instances
Network	vpc-1a233c78 (10.0.0.0/16)
Subnet	subnet-cc7d01e4(10.0.0.0/24) us-east-1b 251 IP Addresses available
Public IP	<input checked="" type="checkbox"/> Automatically assign a public IP address to your instances

For this example, we created a micro server.



Choose a NAT instance type based on your intended workload. If your application only occasionally needs to connect to the Internet and doesn't require high network bandwidth, then a micro instance will suffice. If your application talks to the Internet continuously and requires better bandwidth, then start with m1.medium instances. You may need to upgrade the NAT instance to m1.large because network I/O varies between instance types.

Now, deselect the **-Source/Destination** check box, right click on the NAT instance, select **-Change Source/Dest. Check**, and click on **-Disable**.



The NAT instance must be able to send and receive traffic from sources or destinations other than itself, so you'll need to deselect the **-source/destination** check boxes.

Now navigate to Security Groups to add rules for inbound traffic.

Go to the Inbound tab for NAT_SG01. These rules will allow app servers to talk to the NAT instance on the 80 and 443 ports.

1. Select **-HTTP** from the Create a new rule list. In the Source box, specify the IP address range of your private subnet (App server subnets) and then click **-Add Rule**.



2. Select **-HTTPS** from the Create a new rule list. In the Source box, specify the IP address range of your private subnet, and then click **-Add Rule**.

Click **-Apply Rule Changes**.

Now navigate to Route Tables and select the private subnets 10.0.1.0/24 and 10.0.2.0/24.

On the Routes tab, specify 0.0.0.0/0 in the Destination box, specify the instance ID of the NAT instance in the Target box, and then click **-Add**.

The screenshot shows the AWS Route Table configuration interface. At the top, it says "1 Route Table selected" and "Route Table: rtb-7fc2dc1d". Below this are tabs for "Routes", "Associations", "Route Propagation", and "Tags". The "Routes" tab is selected. A table lists routes with columns: Destination, Target, Status, Propagated, and Actions. One row is shown: Destination 10.0.0.0/16, Target local, Status active, Propagated No, Actions Remove. Below this is a form to add a new route: Destination 0.0.0.0, Target i-66911146, Status (dropdown), Propagated No, Actions Add. The "Add" button is highlighted.

If you don't need an additional instance for NAT, you can minimize cost by assigning a public IP to the instance that needs Internet access. That will allow the instance to access the Internet directly.

7. Create App Servers

Now go to Services->EC2 ->Launch Instance.

On the Configure Instance Details page, from the Network list choose the VPC that you created previously and select your app server subnet (10.0.1.0/24, 10.0.2.0/24) from the Subnet list.

Optional: Select the **-Public IP** check box to request that your app instance receive a public IP address. This is required when you don't have a NAT instance, but your instance requires Internet access.

On the Configure Security Group page, select the option **-Select an existing security group** and then select the APP_SG01 security group that you created previously. Click **-Review and Launch**.

Now log in to the server and check to see whether or not you can access the Internet.

```
$ ping google.com
```

You now might ask, **-How can I access from my desktop an instance that was created in a private subnet and has no assigned public IP?** The answer is that you can't. To do so, you'll need a bastion box in the public subnet. You can use a NAT instance as a bastion server (also known as a jump box).



Log in to the bastion (NAT) server first. You can access any instance from this server that was created in a private subnet.

For more details, see [here](#).

8. Create RDS

Navigate to Services->RDS

Go to Subnet Groups in the navigation pane and click -**Create DB Subnet Group**.

Select the VPC ID from the drop down menu.

Select -**Availability Zone** and choose the Subnet IDs of 10.0.11.0/24 and 10.0.12.0/24. Then click -**Add**.

Click -**Yes, Create** to create the subnet group.

Create DB Subnet Group

To create a new Subnet Group give it a name, description, and select an existing VPC below. Once you select an existing VPC, you will be able to add subnets related to that VPC.

Name:	MYDB_SUBGROUP01
Description:	MYDB_SUBGROUP01
VPC ID:	vpc-1a233c78

Add Subnet(s) to this Subnet Group. You may add subnets one at a time below or add all the subnets related to this VPC. You may make additions/edits after this group is created.

Availability Zone:	Subnet ID:	Availability Zone	Subnet ID	CIDR Block	Action
us-east-1d	subnet-2c90cd6a	us-east-1d	subnet-2c90cd6a	10.0.12.0/24	Remove
	Add	us-east-1b	subnet-057d012d	10.0.11.0/24	Remove

Cancel **Yes, Create**

Creating an Options Group and a Parameters Group is similar to doing so in classic EC2.

Launch an RDS instance within the subnet group created above.

In the Additional Config window, select the VPC and DB Subnet Groups created previously.



Additional Config

Provide the optional additional configuration details below.

Database Name: (e.g. mydb)

Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

Database Port:

Choose a VPC:

DB Subnet Group:

Publicly Accessible: Yes No

To make sure that your RDS instance is launched in subnets 10.0.11.0/24 and 10.0.12.0/24, select the -mydb-subgroup01|| subnet group.

All other steps for creating an RDS are as usual.

9. Create ELB

Now it's time to create the load balancer. The load balancer will be the frontend and will be accessible from the Internet, which means that the ELB will be launched in public subnets 10.0.51.0/24 and 10.0.52.0/24.

At this point the two subnets can't access the Internet. To make them public subnets, update the route table that these subnets are associated to.

Navigate to Services->VPC->Route Tables

Select the ELB route table.

On the Routes tab, specify 0.0.0.0/0 in the Destination box, select the Internet gateway in the Target box, and then click -Add||.

Navigate to Services-> EC2-> Load Balancers

Click -**Create Load Balancer||.**

In the Launch Wizard, select -Create LB inside|| as your VPC ID.

Do not select -**Create an internal load balancer||.**

Click -**Continue||**

In Add EC2 Instances select the subnets where you want the load balanced instances to be. Select 10.0.51.0/24 and 10.0.52.0/24.



Create a New Load Balancer

Cancel X

Define Load Balancer Configure Health Check Add EC2 Instances Review

You will need to select a Subnet for each Availability Zone where you wish to have load balanced instances. A Virtual Network Interface will be placed inside the Subnet and allow traffic to be routed into that Availability Zone. Only one subnet per Availability Zone may be selected.

VPC: vpc-1a233c78

Available Subnets

	Subnet ID	Subnet CIDR	Availability Zones
	subnet-ca037fe2	10.0.1.0/24	us-east-1b
	subnet-2c90cd6a	10.0.12.0/24	us-east-1d
	subnet-cc7d01e4	10.0.0.0/24	us-east-1b
	subnet-057d012d	10.0.11.0/24	us-east-1b

Selected Subnets*

	Subnet ID	Subnet CIDR	Availability Zones
	subnet-067d012e	10.0.51.0/24	us-east-1b
	subnet-2093ce66	10.0.52.0/24	us-east-1d

Back Continue * Required field

In the next window select "Choose from your existing security group" and then select the ELB_SG01 security group that you created previously. Click **-Continue**.

In the next window select the App servers. Click **-Continue**.

Review the details and click **-Create**.

Make sure that you've enabled the APP_SG01 inbound ports (80/443) to ELB_SG01 so that the ELB can route traffic to backend app servers. Also make sure that ELB_SG01 HTTP and HTTPS ports are publicly accessible (0.0.0.0/0).

