



AWS CERTIFIED SOLUTION ARCHITECT ASSOCIATE

Study Guide V1.0

Abdul Jaseem. VP

Technical Consulting Engineer at Cisco
CCIE Collab 59174, CCNP Collab, CCNA,
DevNet, vmware DCV, AWS SAA, CKA

30/August/2020

[LinkedIn](#); [YouTube](#)

Contents

Introduction to Virtualization and Cloud Computing	7
What is Amazon Web Services (AWS)?.....	8
History of AWS.....	9
AWS Global Infrastructure.....	10
1. Regions	10
2. Availability Zones (AZ).....	10
3. AWS Edge Locations:	10
AWS Foundation Services	11
1. Compute.....	11
2. Networking.....	11
3. Storage.....	11
AWS Platform Services	11
1. Databases (Example. Pre-installed SQL).....	11
2. Analytics.....	11
3. Deployment.....	11
4. Mobile Services.....	11
AWS Management Console	12
1. Command Line Interface (CLI)	12
2. RESTFUL API.....	12
3. Web Based Console	12
Create an AWS Account	13
Amazon EC2 Instance.....	15
What is Amazon Elastic Compute Cloud EC2?.....	15
How to Deploy Amazon Linux AMI EC2?.....	16
How to Access Amazon Linux AMI EC2 Instance? Standalone SSH Client.....	20
How to Access Amazon Linux AMI EC2 Instance? Browser-based SSH connection.....	22
How to Deploy Microsoft Windows Server 2019 Base AMI EC2?	23
How to Access Windows Server 2019 AMI EC2 Instance RDP?	26
EC2 Instance Screenshot	27
Elastic Network Interface (ENI)	28
[LAB] Create Secondary ENI and Add to Windows EC2	29
Elastic IP Address.....	30
[LAB] Create Elastic IP and Assign to Windows EC2.....	31
Life Cycle of an EC2 Instance.....	33
EC2 Instance Machine Types	34
Meta Data & User Data of EC2 Instances	35
Storage Options for EC2 Instances	37
1. Instance Store Backed.....	37
2. AWS Elastic Block Store (EBS).....	38
Types of EBS.....	40
[LAB] Managing EBS	41
EC2 Placement Group	43
Creating Custom AMI	44
EC2 Pricing Model.....	46
On-Demand / Pay as you Go	46
Reserved	46
Savings Plans	46
Spot.....	46
Dedicated Host.....	46

Tracking your AWS Free Tier usage	47
AWS Pricing Calculator.....	48
AWS VPC (Virtual Private Cloud)	49
Components of VPC (Overview)	50
IP Classless Inter-Domain Routing CIDR Block	51
Subnets	52
Routing in VPC	53
[LAB] VPC Configurations	54
Step 1: Creating VPC	54
Step 2: Enable Public DNS Hostnames	56
Step 3: Creating Subnets.....	57
Step 4: Enable Auto Assign Public IP for Public Subnet.....	59
Step 5: Add Internet Gateway and Associate to Public Subnet	60
Step 6: Route Table Configuration.....	61
Step 7: Associate Route Table with Subnets	62
Step 8: Launch EC2 Instances in the VPC	64
Security Groups.....	66
[LAB] Create or Select Security Group while Launching EC2.....	67
[LAB] Modify Existing Security Group.....	68
[LAB] Allow ICMP from Internet to a Windows EC2 Instance.....	70
[LAB] Assign new Security Group to a Running EC2 Instance.....	72
Network Access Control Lists (NACLs)	74
[LAB] Modify NACLs to Block ICMP Traffic at the Subnet Level	75
[LAB] Create new NACL and Associate to Subnet.....	76
ASSIGNMENT 1	77
[LAB] Manually Host Sample HTML Website in AWS Windows Server 2019 EC2	77
[LAB] Get a free Domain Name.....	81
[LAB] PowerShell Script to Host Sample HTML Website in AWS Windows Server 2019 EC2	83
[LAB] Hosted PowerShell Script to Host HTML Website in AWS Windows Server 2019 EC2	87
[LAB] HTML Web Hosting Automation in AWS Windows Server 2019 EC2 Using User Data.....	88
ASSIGMENT 2	89
[LAB] Manually Host Sample PHP Website in AWS Linux EC2	89
[LAB] Bash Script to Host Sample PHP Website in AWS Linux EC2	90
[LAB] Hosted Bash Script to Host Sample PHP Website in AWS Linux EC2.....	91
[LAB] PHP Web Hosting Automation in AWS Linux EC2 Using User Data	92
EC2 Launch Template.....	93
AWS Simple Storage Service (S3)	96
What is AWS Simple Storage Service (S3).....	96
S3 Object Versioning.....	97
Storage Classes.....	98
S3 Life Cycle Management.....	100
[LAB] AWS Simple Storage Service S3 Configuration.....	101
[LAB] S3 Bucket and Object Permissions Method 1	104
[LAB] S3 Bucket and Object Permissions Method 2	105
[LAB] Revoke S3 Object Permission	107
[LAB] Configuring S3 Life Cycle Management	108
[LAB] S3 Versioning.....	109
[LAB] Host a Web Site in S3	110
AWS Snow Family	112
AWS Snowcone	112
AWS Snowball	112

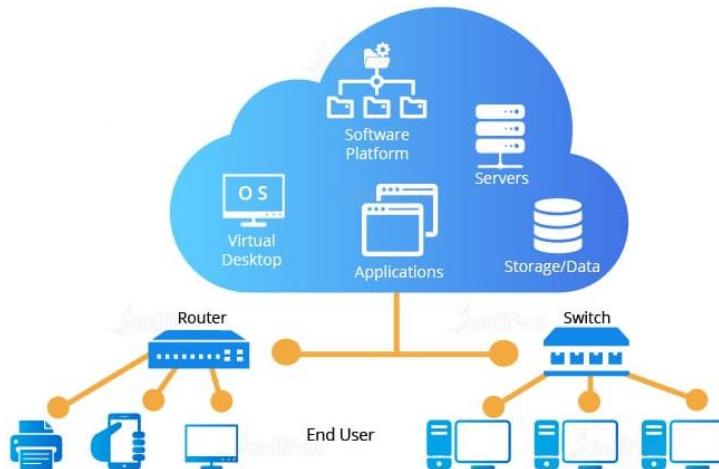
AWS Snowmobile	113
AWS Shared Responsibility Model	114
Identity and Access Management (IAM)	115
[LAB] IAM Lab - Customize the IAM User Login URL (Account Alias)	116
[LAB] Configure IAM User.....	118
[LAB] Cerate IAM Group	120
[LAB] Create IAM Policy	121
[LAB] IAM Role Configuration	123
[LAB] Inline Policy for IAM User	126
[LAB] Inline Policy for IAM Group	127
AWS Organizations	128
[LAB] Creating a Child Account and Provide t2.micro Policy	129
AWS CLI	132
Install AWS CLI on Windows	132
Install AWS CLI on Linux	133
AWS Access Key and Secret Key Authentication	134
Launch EC2 Instance Using AWS CLI.....	136
AWS Python API / SDK - BOTO3	137
Install BOTO3.....	137
Start / Stop EC2 Instance with Python Boto3 Module	138
Deploy EC2 using Python Boto3 API	139
Components of VPC Overview (Detailed)	140
NAT Gateway	141
[LAB] Provide Internet Access to Private Subnet Instances using NAT Gateway.....	143
NAT Instance	146
[LAB] How to Deploy NAT Instance	147
VPC Peering	149
[LAB] VPC Peering Cross Region.....	150
Step 1: Create a VPC in Singapore ap-southeast-1	150
Step 2: Creating 4 Subnets (2 Private and 2 Public)	150
Step 3: Auto-assign public IPv4 address for Public Subnets	151
Step 4: Create 2 Route Tables (Public and Private) in Singapore VPC.....	151
Step 5: Launch a Windows EC2 in Private Subnet	152
Step 6: Create a Peering Connection from Mumbai to Singapore	152
Step 7: Update Mumbai Public and Private Routables	153
Step 8: Update Singapore Private Route Table.....	153
Step 9: Access Singapore EC2 instance from Mumbai Instance	154
VPC IPSec Virtual Private Network (VPN)	155
[LAB] VPC IPSec VPN Configuration.....	156
Step 1: Create Customer gateway	156
Step 2: Create VPN Gateway & Attach to VPC.....	156
Step 3: Create Site to Site VPN Connection	157
Step 4: Download the Vendor Specific Configuration.....	158
VPC Direct Connect	159
AWS Transit Gateway (TGW).....	160
[LAB] Transit Gateway.....	161
Step 1: Create 3 VPCs, 3 Subnets, edit 3 default route table	161
Step 2: Launch Amazon Linux EC2 on 3 VPC's Private Subnets	162
Step 3: Creating Transit Gateway	162
Step 4: Attaching VPCs to the Transit Gateway.....	163
Step 5: Update Each VPC Route Table with Transit Gateway	164

Step 6: Login to Public Windows EC2 and Access other Linux EC2s.....	164
AWS Relational Database Service.....	166
[LAB] RDS Configuration.....	167
DynamoDB (AWS NoSQL).....	170
[LAB] Dynamo DB Configurations.....	171
Amazon CloudWatch.....	172
[LAB] Configure EC2 Alarm for CPU Utilization	173
Elastic Load Balances (ELB)	176
[LAB] Elastic Load Balances.....	177
VPC Endpoints.....	182
[LAB] Get Access to S3 from Private Subnet EC2 Instance	183
VPC Interface Endpoints.....	185
SNS - Simple Notification Service	186
[LAB] SNS Configuration Notification email for ELB Number of Connections.....	187
SQS - Simple Queuing Service.....	191
Simple Email Service (SES).....	192
[LAB] Simple Email Service.....	192
Auto Scaling	195
[LAB] Auto Scaling Configuration.....	197
[LAB] Test Auto Scaling Lab: Unhealthy Instance	200
[LAB] Test Auto Scaling Lab: CPU Utilization Scale Out.....	201
[LAB] Test Auto Scaling Lab: CPU Utilization Scale In	203
Route 53.....	204
Routing Policies.....	204
[LAB] Route 53.....	205
Domain Registration.....	205
Hosted Zone.....	205
AWS CloudTrail.....	207
[LAB] Cloud Trail	208
AWS Lambda.....	209
[LAB] AWS Lambda Example	210
Step 1: Create a Policy with below JSON	210
Step 2: IAM Role for Lambda	211
Step 3: Test Lambda functions.....	214
Step 4: Create CloudWatch Rule to Trigger Lambda Functions	215
AWS CloudFront	216
[LAB] AWS CloudFront Distribution for a load balanced Web Site	217
AWS Cloud Formation	219
[LAB] AWS Could Formation - Exporting Existing Infa to Template (CloudFormer)	220
[LAB] AWS Could Formation - Deploy from Template	226
AWS Could Formation - Designer	227
Elastic Beanstalk.....	228
[LAB] Elastic Beanstalk.....	229
Other AWS Services (Theory Only)	231
AWS Cognito	232
AWS Directory Services	232
Single Sign-On (SSO).....	233
Storage Gateway	233
API Gateway	233
AWS OpsWorks	233
RedShift.....	234



Kinesis.....	234
Elastic File System (EFS).....	234
FXs	234
Terra Forms.....	234
About the Author:.....	235

Introduction to Virtualization and Cloud Computing



- **What is Internet:** A global computer network using standardized communication protocols (e.g. UDP, TCP/IP) providing information and communication facilities.
- **Local Network:** Private network in LAN (Local Area Network)
- **Virtualization:** Run multiple OSs on a host machine (Type 1: BareMetal, Uses Hypervisor OS (e.g. ESXi) and Type 2: Application running on another base OS (e.g. vmware workstation)).
- **Virtual Machine:** Software representation of virtual computer as set of files! Easy to move, independent of hardware, Effective utilization of resources. We can do virtual networking between VMs.
- **Data Center:** Data centers are simply centralized locations where computing and networking equipment is concentrated for the purpose of collecting, storing, processing, distributing or allowing access to large amounts of data.
- **What is Cloud?** There is now Cloud, it is someone else computer accessible over Internet! Virtual Machine running on a cloud server is the most widely used way of hosting any applications online.
- **Cloud Computing:** On demand delivery of IT resources via the Internet, Instead of setting up Physical DCs, we can access Compute, Network, Storage on demand basis from Cloud providers. Use cases: Data backup, DRS, Email, Virtual Desktop, Software Development, Testing, Web Apps, Online Gaming, IoT, etc. Worldwide availability.



AWS Reference: [What is Cloud Computing?](#)

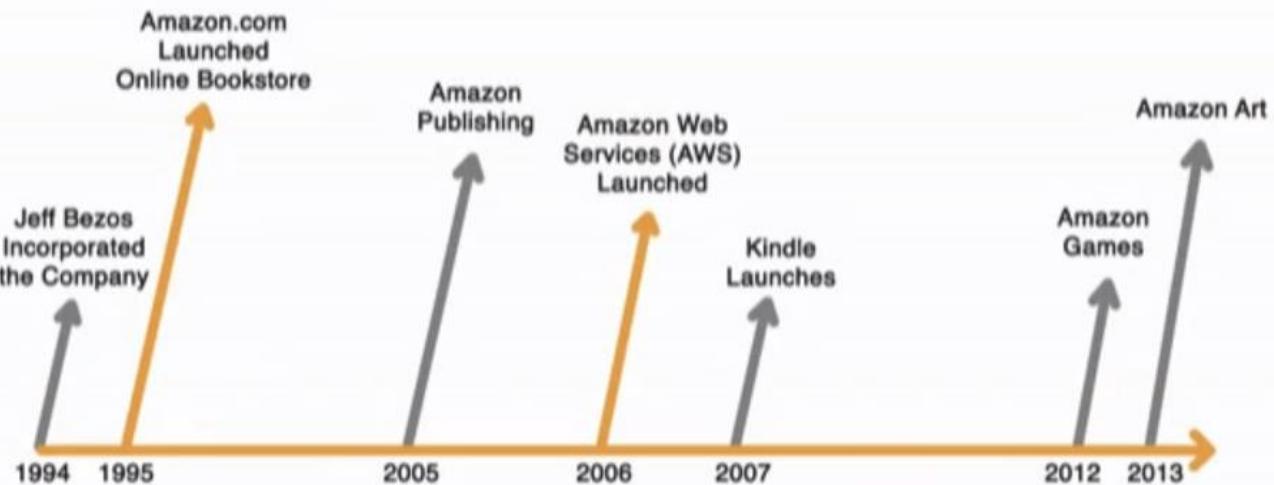
What is Amazon Web Services (AWS)?



- Applications like YouTube, Gmail, Facebook are software that are hosted on cloud. Those are often called Software as a Services (SaaS).
- Amazon Web Services is a public cloud provider, a gigantic pool of configurable resources (Servers, Storage, Networks, etc.) spanned across multiple Data Centers in the globe where you can deploy your infrastructure/applications. Hence it is also called Infrastructure as a Services (IaaS).
- IaaS – Entire Infrastructure provided to you as a fully managed service
- Rapid provisioning and release of the resources.
- Resources are elastic in nature; you can scale up and scale down the resources. (Scaling: Ability to change an implementation to respond to changing traffic patterns.). e.g. During Black Friday sale, services may require more compute power.
- You can build could apps/ software (SaaS) using AWS (IaaS). Services can be used On-demand and Pay as you Go fashion (Like our Electricity bill, pay for the usage).
- We can gradually move services to Cloud (On-Ramp), some of the services are moved to cloud, often called Hybrid Cloud.
- AWS follows strict regulations that allows Banking, Research sectors to move their infrastructure to cloud.
- The collective money that we pay to Amazon, enables them to maintain the Data Centers all over the world.
- Examples AWS Solutions: amazon.com, Netflix, etc. uses AWS in the backend.

AWS Reference: [What is AWS?](#)

History of AWS



- 1994 – Jeff Bezos Started the company
- 1995 – amazon.com launched as an online bookstore
- 2006 – AWS Launched (Elastic Compute Cloud - EC2, S3)
- 2009 – Relational Database Services – RDS
- 2011 – Elastic Beanstalk
- 2012 – Glacier, DynamoDB, RedShift
- 2016 – Lambda

Over the years AWS has been diligent about expanding their offerings, making sure the features and products available inside the cloud that allow us to take enormous advantage of the resources available to us.

AWS Global Infrastructure

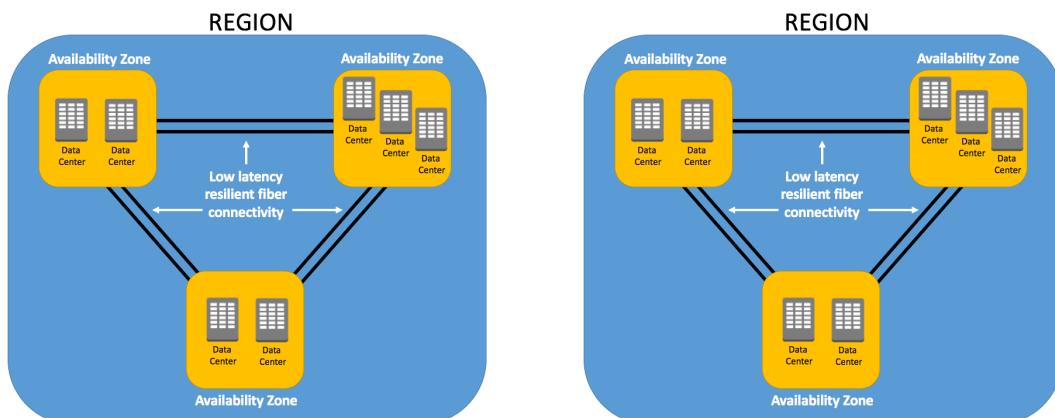
AWS infrastructure is geographical deployed in different Data Centers.

1. Regions

- Any geographic location where AWS has a **Cluster of Data Center** is called Region.
- AWS infrastructure Regions meet the highest levels of security, compliance, and data protection.
- 24 Launched and 3 announced regions (at the time of this article)

2. Availability Zones (AZ)

- Each **group data centers** called an Availability Zone (AZ). Each AZ has independent power, cooling, and physical security and is connected via redundant, ultra-low-latency networks.
- Inside a Region we can find multiple Availability Zone (1a, 1b, 1c), each **Availability Zone comprised of multiple Data Centers**.
- Inside an Availability Zone, our data/ infrastructure is split-up among multiple DCs. This gives built in reliability and redundancy.
- Also enables to push services close to your customer to reduce the latency.



- AZ's are physically separated by a meaningful distance, many kilometers, from any other AZ, although all are within 100 km (60 miles) of each other.
- Amazon recommends deploying apps/ solutions in at least 2 AZs
- 77 Availability Zones (at the time of this article)

3. AWS Edge Locations:

- Used to **cache your data in order to get faster response time** for customers. Like Content Delivery Network (CDN). e.g. Video Streaming, Netflix, Amazon Prime.
- CloudFront CDN available here
- 205 Edge Locations (at the time of this article)

AWS Foundation Services

1. Compute

- EC2, Lambda, Elastic Beanstalk

2. Networking

- Load-balancing, Route53, VPC, Direct Connect

3. Storage

- S3, Block Storage, Glacier

AWS Platform Services

More like Software as a Service (SaaS). The AWS Global Cloud Infrastructure is the most secure, extensive, and reliable cloud platform, offering over **175 fully featured services** from data centers globally.

1. Databases (Example. Pre-installed SQL)

- DynamoDB, RDS, Redshift

2. Analytics

- Kinesis, EMR, Data Pipeline

3. Deployment

- Elastic Beanstalk (Platform as a Services - PaaS), Code Deploy

4. Mobile Services

- Cognito, SNS

PLATFORM SERVICES



AWS Reference: [Regions and Availability Zones](#)

AWS Management Console

Three ways to connect AWS.

1. Command Line Interface (CLI)

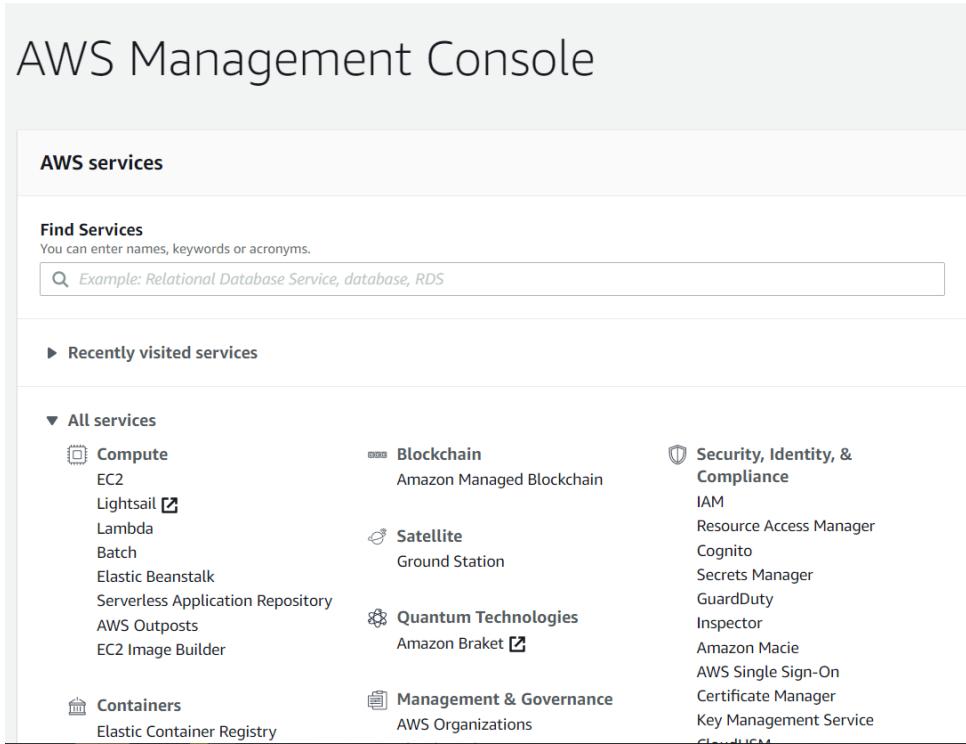
- Utility that you can download from AWS website.
- Integrate with your Shell (cmd, bash, etc.) and write AWS commands.

2. RESTFUL API

- HTTP Calls to AWS.
- Makes easier for automation.

3. Web Based Console

- Easiest way to connect AWS via Web browser, even AWS has mobile version!



The screenshot shows the AWS Management Console homepage. At the top, there's a search bar labeled "Find Services" with the placeholder "You can enter names, keywords or acronyms." Below it is a "Recently visited services" section. The main navigation area is titled "AWS services" and contains several categories:

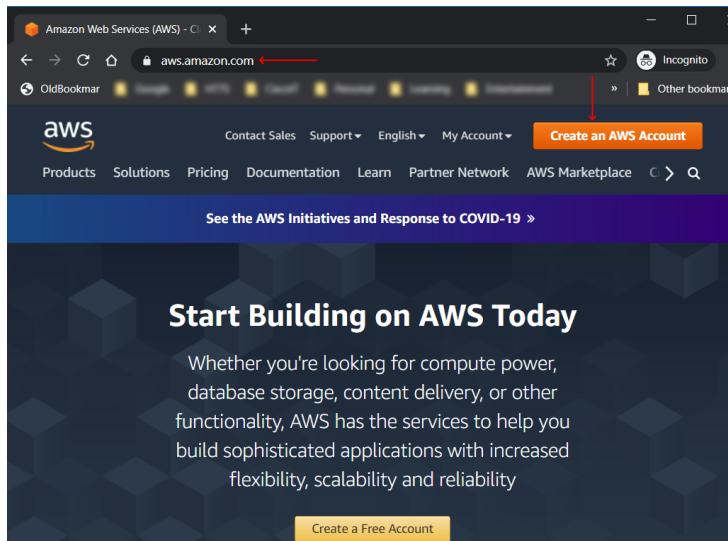
- All services** (expanded):
 - Compute** (includes EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, EC2 Image Builder)
 - Containers** (includes Elastic Container Registry)
 - Blockchain** (includes Amazon Managed Blockchain)
 - Satellite** (includes Ground Station)
 - Quantum Technologies** (includes Amazon Braket)
 - Management & Governance** (includes AWS Organizations)
 - Security, Identity, & Compliance** (includes IAM, Resource Access Manager, Cognito, Secrets Manager, GuardDuty, Inspector, Amazon Macie, AWS Single Sign-On, Certificate Manager, Key Management Service)

Create an AWS Account

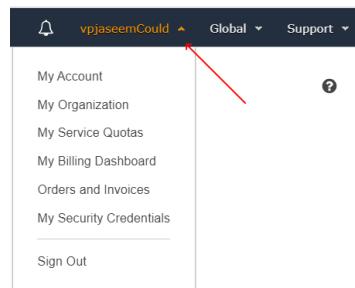
AWS Accounts Include 12 Months of Free Tier Access, including use of Amazon EC2, Amazon S3, and Amazon DynamoDB. Visit aws.amazon.com/free for more details. You need a Credit / Debit card to sign up. AWS will charge you RS.2/- and refund it back to the same card.

Create an AWS Account

- Go to aws.amazon.com >> Create an AWS Account



- Follow the Signup process. You can choose any friendly name as **AWS account name**. That will be displayed once you login (My AWS account name is 'vpjaseemCloud', this can be edited anytime)



Select a Support Plan

AWS offers a selection of support plans to meet your needs. Choose the support plan that best aligns with your AWS usage. [Learn more](#)

Basic Plan	Developer Plan	Business Plan
Free	From \$29/month	From \$100/month
<ul style="list-style-type: none"> Included with all accounts 24x7 self-service access to AWS resources For account and billing issues only Access to Personal Health Dashboard & Trusted Advisor 	<ul style="list-style-type: none"> For early adoption, testing and development Email access to AWS Support during business hours 1 primary contact can open an unlimited number of support cases 12-hour response time for nonproduction systems 	<ul style="list-style-type: none"> For production workloads & business-critical dependencies 24/7 chat, phone, and email access to AWS Support Unlimited contacts can open an unlimited number of support cases 1-hour response time for production systems

Need Enterprise level support?
Contact your account manager for additional information on running business and mission critical-workloads on AWS (starting at \$15,000/month). [Learn more](#).

AWS Management Console

AWS services

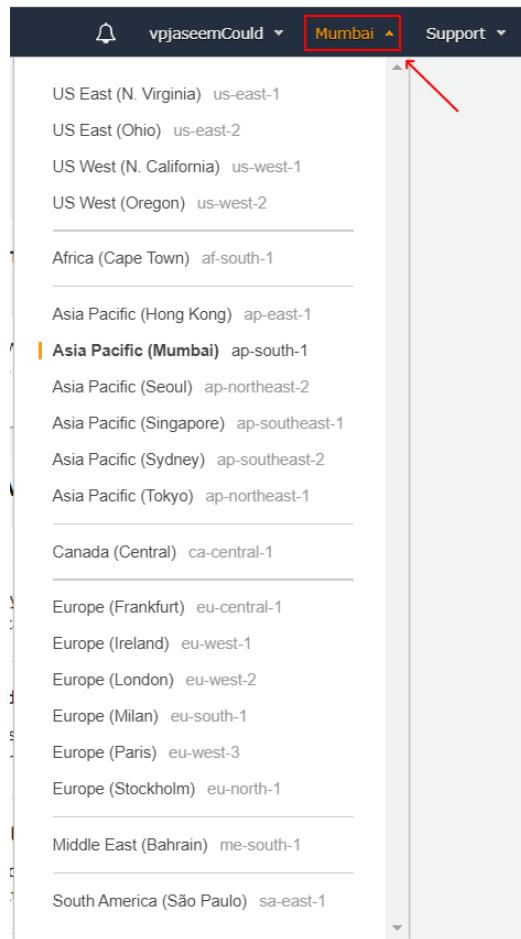
Find Services
You can enter names, keywords or acronyms.

Recently visited services

All services

Compute	Blockchain	Security, Identity, & Compliance
EC2	Amazon Managed Blockchain	IAM
Lightsail	Satellite	Resource Access Manager
Lambda	Ground Station	Cognito
Batch		Secrets Manager
Elastic Beanstalk		GuardDuty
Serverless Application Repository		Inspector
AWS Outposts		Amazon Macie
EC2 Image Builder		AWS Single Sign-On
 		Certificate Manager
Containers	Management & Governance	Key Management Service
Elastic Container Registry	AWS Organizations	

- Once you click the Region Name, you can see all available regions as shown below.



- You can pin some essential services as quick link in the top bar

To customize one-click navigation shortcuts simply drag your services to and from the menu bar above.

The screenshot shows the AWS Management Console top bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and pinned services: EC2, VPC, and EFS. Below the top bar is a sidebar with a list of services. A red arrow points from the text 'To customize one-click navigation shortcuts simply drag your services to and from the menu bar above.' to the pinned services area. Another red arrow points from the text to the list of services in the sidebar.

AWS AppSync	EFS	Service Catalog
AWS Auto Scaling	Elastic Beanstalk	Simple Email Service
AWS Backup	Elastic Container Registry	Simple Notification Service
AWS Budgets	Elastic Container Service	Simple Queue Service
AWS Chatbot	Elastic Kubernetes Service	Step Functions
AWS Cloud Map	Elastic Transcoder	Storage Gateway
AWS Compute Optimizer	ElastiCache	SWF
AWS Lambda		

Amazon EC2 Instance



What is Amazon Elastic Compute Cloud EC2?

- Amazon Elastic Compute Cloud (Amazon EC2) is an IaaS offering that provides Custom Virtual Machines in the cloud. It's a fully managed infrastructure.
- Fast provisioning and Supports Elastic Scaling – Dynamically add instances to meet specific traffic requirements.
- The scope of EC2 is AZ
- **Amazon Machine Images (AMIs):** Preconfigured Operating System templates with and additional software installed. We can build our own custom images as well.
- **Instance Types:** Various configurations of CPU, memory, storage, and networking capacity for your instances.
- Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)
- Metadata, known as tags, that you can create and assign to your Amazon EC2 resources. This is helpful to identify the instances easily.
- Flexible payment options: Pay as You Go (Pay only for the machine in a running state, stop machine to save cost. Hourly pricing (partial hour count as full)), Reserved, Scheduled

AWS Reference: [What is Amazon EC2?](#)

How to Deploy Amazon Linux AMI EC2?

Now let's see how to deploy a 'Amazon Linux AMI' EC2 instance. There are 4 things to consider while deploying an EC2 instance.

1. Region and Availability Zone
2. Choose an Image (AMI) to launch from
3. Machine Type (CPU, HDD, RAM, etc.)
4. Configure Network, Storage, Security Groups

The screenshot shows the AWS EC2 Dashboard. On the left sidebar under 'INSTANCES', 'Instances' is selected. In the main content area, there is a 'Launch instance' section with a large orange 'Launch instance' button. A red arrow points to this button.

Go to EC2 Dashboard >> Launch instance >>

Launch instance >>

The screenshot shows the 'Choose AMI' step of the EC2 wizard. It lists several AMI options, with 'Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-00ab7932c8e82a7b5' selected and highlighted by a red box. A red arrow points to the 'Select' button next to it.

Free tier only >> Select 'Amazon Linux AMI' >>

The screenshot shows the 'Choose Instance Type' step of the EC2 wizard. It lists various instance types, with 't2.micro' selected and highlighted by a red box. A red arrow points to the 'Select' button next to it.

General purpose t2.micro >> Next: Configure Instance Details

AWS Reference: [EC2 Instance Types](#)

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 the lower pricing, assign an access management role to the instance, and more.

Number of instances	1	Launch into Auto Scaling Group
Purchasing option	<input type="checkbox"/> Request Spot Instances	
Network	vpc-fa6ccaa91 (default)	<input type="button"/> Create new VPC
Subnet	No preference (default subnet in any Availability Zone)	<input type="button"/> Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)	
Placement group	<input type="checkbox"/> Add instance to placement group	
Capacity Reservation	Open	<input type="button"/> Create new Capacity Reservation
IAM role	None	<input type="button"/> Create new IAM role
Shutdown behavior	Stop	
Stop - Hibernate behavior	<input type="checkbox"/> Enable hibernation as an additional stop behavior	

Buttons: Cancel, Previous, Review and Launch, Next: Add Storage

- Configure instance details to suit your requirements.
- Select all default options for now, we will cover every component in future sessions.

>> Next: Add Storage

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.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-08e7474af8cb4de9a	8	General Purpose S	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Buttons: Cancel, Previous, Review and Launch, Next: Add Tags

- An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer.
- You can attach additional EBS (Elastic Block Storage) 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.

>> Next: Add Tags

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	amazon-linux	<input type="checkbox"/>	<input type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Buttons: Cancel, Previous, Review and Launch, Next: Configure Security Group

- Tags used to identify the EC2 instances as quickly as possible

>> Next: Configure Security Group

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 your instance, add 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: aws-linux-security-group
Description: aws-linux-security-group

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom	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.

Cancel **Previous** **Review and Launch**

- Select Create a **new security group >>** provide a **Security group name**
- **Security Group:** A security group is a set of firewall rules that control the traffic for your instance
- You can add rules to **allow specific traffic to reach your instance**, Make sure you have SSH (22) allowed from all IP Address >> **Review and Launch**

Step 7: Review Instance Launch

AMI Details

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-00ab7932c8e82a7b5

Free tier eligible

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.

Root Device Type: ebs Virtualization type: hvm

Instance Type

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

Security Groups

Instance Details

Storage

Tags

Edit AMI **Edit instance type** **Edit security groups** **Edit instance details** **Edit storage** **Edit tags**

Cancel **Previous** **Launch**

- We can review all the settings here. Quickly verify those.

>> Launch

- 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.

Select an existing key pair or create a new key pair X

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](#).

Create a new key pair **Key pair name** **aws-linux** **Download Key Pair**

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location**. You will not be able to download the file again after it's created.

Cancel **Launch Instances**

- Select 'Create a new key pair' and enter a name for that.
- You can download the 'Private Key'. Keep that in a safe place

>> Launch Instances



Launch Status

Your instances are now launching

The following instance launches have been initiated: i-0cc7bad28d9c7af5c Hide launch log

Creating security groups	Successful (sg-06bea6f5dbecc8a1c)
Authorizing inbound rules	Successful
Initiating launches	Successful
Launch initiation complete	

Get notified of estimated charges

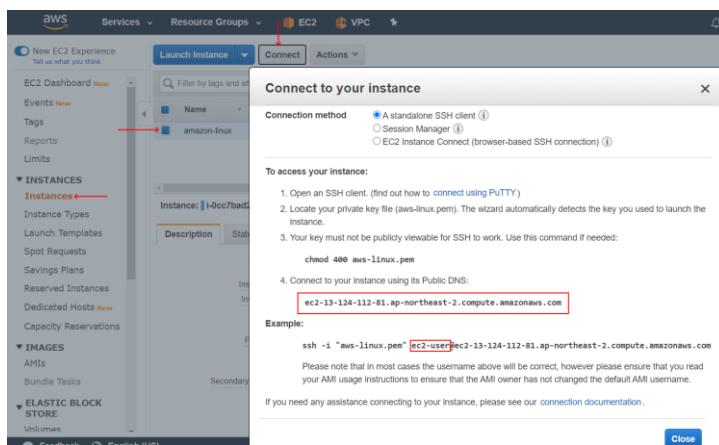
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like EC2 Dashboard, Instances (which is selected and highlighted in red), Images, and Elastic Block Storage. The main content area displays a table of instances. One instance is listed: 'amazon-linux' (Instance ID: i-0cc7bad28d9c7af5c, Instance Type: t2.micro, Availability Zone: ap-northeast-2a, Status: running). Below the table, detailed information for this instance is shown in a card format, including Public DNS, IPv4 and IPv6 Public IPs, Instance State, Instance Type, Finding, Opt-in to AWS Compute Optimizer, Private DNS, and Availability zone.

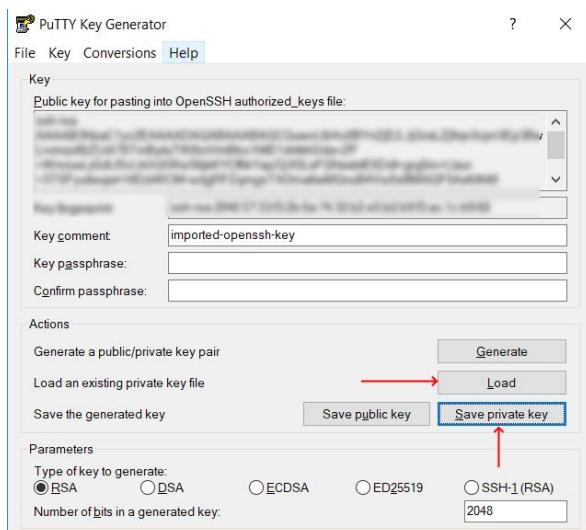
- Once the launching is completed, you can click the 'View Instances' button on the same page to see the newly created EC2 instance
- You could see all the details (Status, DNS, Public IP, Type, etc.) in this page

How to Access Amazon Linux AMI EC2 Instance? Standalone SSH Client

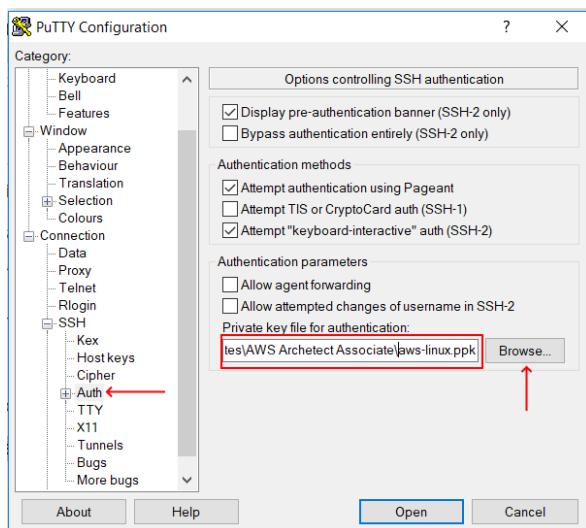
- Linux machines build from AMI can be accessed from anywhere over internet. You should need the .PEM file that downloaded after the creation of an EC2 instance.
- Windows uses Putty Gen to convert the .PEM file to .PPK file, and this file used to connect Linux over SSH. We do not use any password!



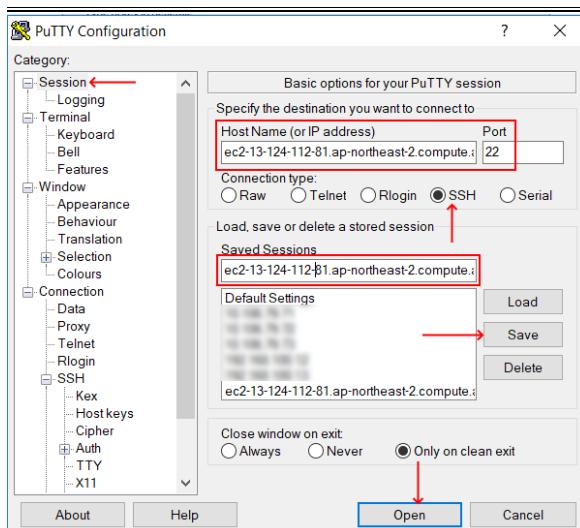
- Select the EC2 instance >> Connect >> Note the Public DNS (You can also use the Public IP of the EC2)
- Also note the username here is ec2-user



- Open PuttyGen and Load the .PEM file downloaded before. Click Save Private Key
- Keep the private key in a safe place



- Open Putty >> Auth >> Browse the Private Key (.PPK file created in previous step)



- Session >> Enter the FQDN or Public IP of the EC2 instance >> Select SSH >> Optionally you can Save the session >> Click Open

```

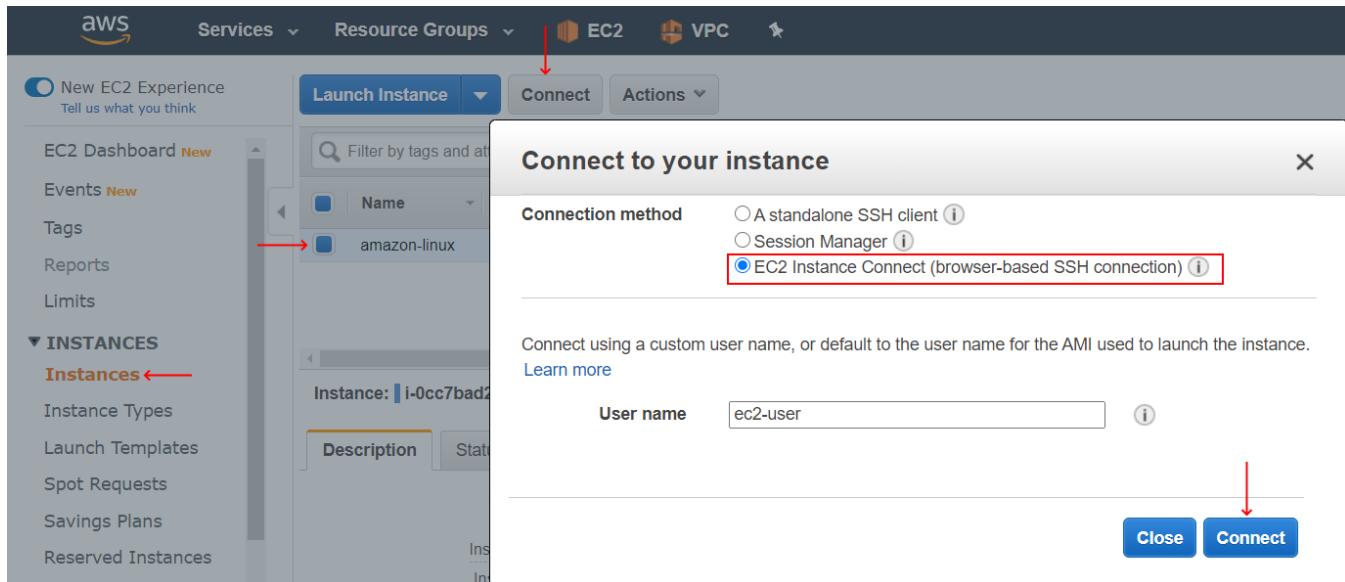
[ec2-user@ip-172-31-2-252:~]
[ec2-user@ip-172-31-2-252:~] login as: ec2-user
[ec2-user@ip-172-31-2-252:~] Authenticating with public key "imported-openssh-key"
[ec2-user@ip-172-31-2-252:~] _|_ ) 
[ec2-user@ip-172-31-2-252:~] _| ( /  Amazon Linux AMI
[ec2-user@ip-172-31-2-252:~] __\_\_|_
[ec2-user@ip-172-31-2-252:~] https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-2-252:~] 5 package(s) needed for security, out of 7 available
[ec2-user@ip-172-31-2-252:~] Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-2-252:~] $ [ec2-user@ip-172-31-2-252:~] $
[ec2-user@ip-172-31-2-252:~] $ [ec2-user@ip-172-31-2-252:~] $
[ec2-user@ip-172-31-2-252:~] $

```

That's it, you are successfully connected!

How to Access Amazon Linux AMI EC2 Instance? Browser-based SSH connection

- Select the EC2 instance >> Connect >> EC2 Instance Connect (browser-based SSH connection) >> Connect



- This will open Linux CLI on new browser window. Please note that, some time, it fail

How to Deploy Microsoft Windows Server 2019 Base AMI EC2?

Now let's see how to deploy a 'Microsoft Windows Server 2019 AMI' EC2 instance. There are 4 things to consider while deploying an EC2 instance.

1. Region and Availability Zone
2. Choose an Image (AMI) to launch from
3. Machine Type (CPU, HDD, RAM, etc.)
4. Configure Network, Storage, Security Groups

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under 'INSTANCES', 'Instances' is selected. In the main content area, there is a large 'Launch instance' button with a red arrow pointing to it. Above the button, a note says 'To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.' Below the button, it says 'Note: Your instances will launch in the Asia Pacific (Seoul) Region'.

Go to EC2 Dashboard >> Launch instance >>

Launch instance >>

The screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' screen. A search bar at the top contains 'Microsoft Windows Server 2019'. Below it, a note says 'AWS Launch Wizard for SQL Server offers an easy way to size, configure, and deploy Microsoft SQL Server Always On availability groups. Use AWS Launch Wizard for this launch?'. Under 'Quick Start (2)', there are two options: 'Microsoft Windows Server 2019 Base - ami-08ea95913b2505d3' and 'Microsoft Windows Server 2019 Base with Containers - ami-0bddd2d9c2991146'. Both have a 'Select' button. A red arrow points to the 'Free tier only' checkbox on the left. Another red arrow points to the 'Select' button for the first AMI.

Free tier only >> Select 'Microsoft Windows Server 2019 AMI' >>

The screenshot shows the 'Step 2: Choose an Instance Type' screen. A note at the top says '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 CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.' Below is a table of instance types. The 't2.micro' row is highlighted with a red border. At the bottom, there are 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Instance Details' buttons. A red arrow points to the 'Next: Configure Instance Details' button.

General purpose t2.micro >> Next: Configure Instance Details

AWS Reference: [EC2 Instance Types](#)

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 the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option: Request Spot Instances

Network: vpc-fa6ccca91 (default) [Create new VPC](#)

Subnet: No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: [Add instance to placement group](#)

Capacity Reservation: Open [Create new Capacity Reservation](#)

IAM role: None [Create new IAM role](#)

Shutdown behavior: Stop [Edit](#)

Stop - Hibernate behavior: Enable hibernation as an additional stop behavior

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

- Configure instance details to suit your requirements.
- Select all default options for now, we will cover every component in future sessions.

>> Next: Add Storage

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.](#)

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-027232d7f9e13fb8	30	General Purpose	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

- Configure instance storage here.
- Default will get 30GB C:/ Drive Volume

>> Next: Add Tags

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes
Name	windows-server-2019			<input checked="" type="checkbox"/>	<input type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

- Tags used to identify the EC2 instances as quickly as possible

>> Next: Configure Security Group

Step 6: Configure Security Group

A security group is a set of firewall rules that control 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 your instance, add 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: windows-server-security-group

Description: windows-server-security-group

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	Custom	0.0.0.0/0 (e.g. SSH for Admin Desktop)

[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.

[Cancel](#) [Previous](#) [Review and Launch](#)

- Select Create a new security group >> provide a Security group name
- Make sure you have RDP (Port 3389) allowed from all IP Address
- >> Review and Launch

Step 7: Review Instance Launch

AMI Details: Microsoft Windows Server 2019 Base - ami-0f8ea95913b2505d3

Instance Type: t2.micro (1 vCPU, 1 GB Memory)

Security Groups: (Edit security groups)

Instance Details: (Edit instance details)

Storage: (Edit storage)

Tags: (Edit tags)

Launch

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](#).

Create a new key pair: Key pair name: aws-windows

Download Key Pair

You have to download the **private key file (*.pem file)** before you can continue. [Store it in a secure and accessible location](#). You will not be able to download the file again after it's created.

Cancel Launch Instances

- We can review all the settings here. Quickly verify those.

>> Launch

- Select 'Create a new key pair' and enter a name for that.
- You can download the 'Private Key'. Keep that in a safe place
- You can also use previous key (aws-linux) if you like to proceed without generating new key

>> Launch Instances

- Once the launching is completed, you can click the 'View Instances' button on the same page to see the newly created EC2 instance
- You could see all the details (Status, DNS, Public IP, Type, etc.) in this page

New EC2 Experience Tell us what you think

EC2 Dashboard New

Events New

Tags

Reports

Limits

INSTANCES Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts New

Capacity Reservations

IMAGES AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
windows-server-2019	i-05e8a5928f15e32e5	t2.micro	ap-northeast-2a	running	Initializ
amazon-linux	i-0cc7bad28d9c7af5c	t2.micro	ap-northeast-2a	running	2/2 chec

Instance: i-05e8a5928f15e32e5 (windows-server-2019) Public DNS: ec2-3-34-131-94.ap-northeast-2.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID: i-05e8a5928f15e32e5 Public DNS (IPv4): ec2-3-34-131-94.ap-northeast-2.compute.amazonaws.com

Instance state: running IPv4 Public IP: 3.34.131.94

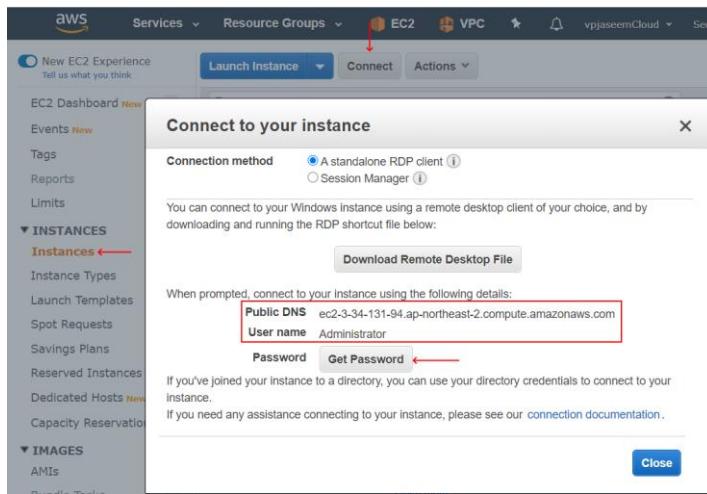
Instance type: t2.micro Instance state: running IPv6 IPs: -

Finding Opt-in to AWS Compute Optimizer for recommendations. Learn more

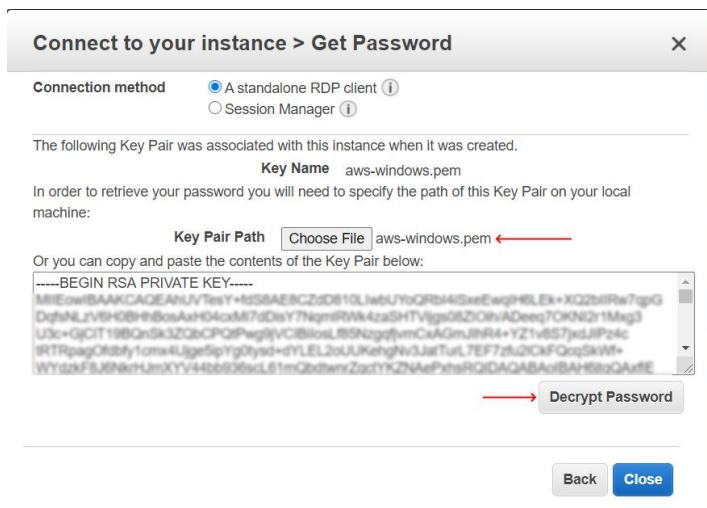
Private DNS: ip-172-31-15-235.ap-northeast-2 Availability zone: ap-northeast-2a

How to Access Windows Server 2019 AMI EC2 Instance RDP?

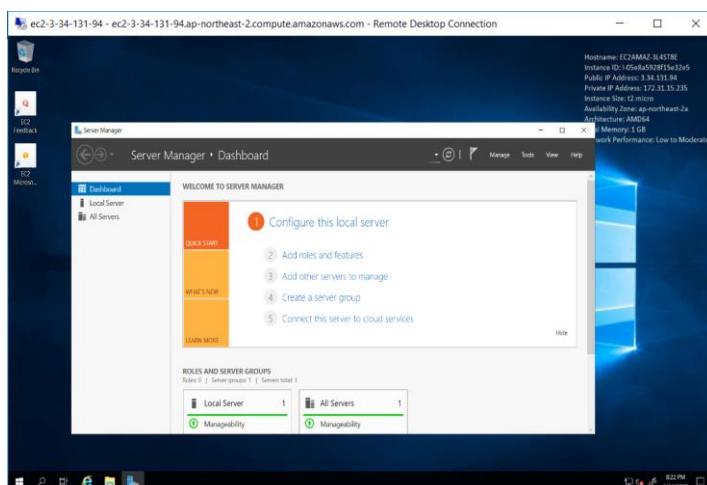
- Windows Server build from AMI can be accessed from anywhere over internet. You should need the .PEM file that downloaded after the creation of an EC2 instance.



- Select the EC2 instance > Connect > Note the Public DNS (You can also use the Public IP of the EC2)
- Also note the username here is Administrator
- Click on the Get Password button



- Choose the .PEM file you downloaded at the last stage of EC2 creation process
- Click Decrypt Password button to get the password



- Once you get the password, use Windows Remote Desktop Connection (DRP) to login to Windows Server 2019

EC2 Instance Screenshot

The screenshot shows the AWS EC2 Instances page. At the top, there's a navigation bar with 'Resource Groups', 'EC2', and 'VPC'. Below that is a search bar labeled 'Filter by tags and attributes or search...'. A list of instances is shown, with 'windows-server-2019' selected and highlighted with a red arrow. An 'Actions' button is clicked, opening a dropdown menu. The menu items are: Connect, Get Windows Password, Create Template From Instance, Launch More Like This, Instance State, Instance Settings (highlighted with a red arrow), Image, Networking, CloudWatch Monitoring, Add/Edit Tags, Attach to Auto Scaling Group, Attach/Replace IAM Role, Change Instance Type, Change Termination Protection, View/Change User Data, Change Shutdown Behavior, Change T2/T3 Unlimited, Get System Log, Get Instance Screenshot (highlighted with a red arrow), Modify Instance Placement, and Modify Capacity Reservation Settings.

The screenshot shows a modal dialog titled 'Get instance screenshot'. It displays a screenshot of a Windows desktop. The desktop is mostly blue, with a message at the top left: 'Press Ctrl+Alt+Delete to unlock.' In the center, it shows the time as '6:46' and the date as 'Tuesday, June 16'. In the bottom right corner, there's a small icon of a monitor. At the very bottom of the dialog, there's a 'Close' button.

Elastic Network Interface (ENI)



- An elastic network interface is a logical networking component (in a *VPC) that represents a virtual network card. Virtualized NIC for EC2s.
- Primary ENI of an EC2 instance can not be detached.
- ENI contains below,
 - A primary private IPv4 address from the IPv4 address range of your VPC
 - One or more secondary private IPv4 addresses from the IPv4 address range of your VPC
 - One public IPv4 address (Dynamic Public IP)
 - One Elastic IP address (IPv4) per private IPv4 address (Static Public IP, manually created)
 - One or more IPv6 addresses
 - Security group
 - A MAC addresses
 - A source/destination check flag
 - A description

AWS Reference: [Elastic network interfaces](#)

*VPC - Virtual Private Cloud, this will be discussed later.

[LAB] Create Secondary ENI and Add to Windows EC2

Note: It is recommended to do ENI lab after learning VPC Configurations.

The screenshot illustrates the process of creating a secondary Network Interface (ENI) and attaching it to a Windows EC2 instance. The steps are as follows:

- Create Network Interface:** In the AWS EC2 service, a new Network Interface named "secondary-eni-for-windows" is created. It is assigned to a specific subnet (subnet-0bc97ee219e9fc411) and VPC (vpc-054e17ec). The IPv4 Private IP is auto-assigned.
- Elastic Fabric Adapter:** A security group named "sg-0815e72d86c4561ba" is selected for the new ENI.
- Tags:** A tag "Name: secondary-eni-for-windows" is added to the ENI.
- Launch Instance:** A Windows Server 2019 instance (ami-03b171651ed4ee2c) is selected for attachment.
- Attach Network Interface:** The "Attach" button is clicked to attach the secondary ENI to the instance.
- EC2 Instance Details:** The newly attached ENI (eth0) is listed under the "Network interfaces" section of the instance details page.
- Windows Control Panel:** On the Windows instance, the "Network Connections" screen shows two network adapters: "Ethernet" (Network 2: AWS PV Network Device #0) and "Ethernet 3" (AWS PV Network Device #1), which corresponds to the secondary ENI.

Elastic IP Address



- An Elastic IP address is a static public IPv4 address, which is reachable from the internet.
- If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet.
- Elastic IP has to be assigned manually.
- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
- The Public IP is dynamic in nature, each time you reboot the instance, it changes. Elastic Public IP is static assignment.
- While your instance is running, you are not charged for one Elastic IP address associated with the instance. If the Elastic IP is not active (Instance is not running, detached from instance), there will be an hourly charge.

AWS Reference: [Elastic IP addresses](#)

[LAB] Create Elastic IP and Assign to Windows EC2

Note: It is recommended to do ENI lab after learning VPC Configurations, also release elastic IP once you complete the labs, elastic IPs are chargeable if not associated to any instance.

The screenshots illustrate the step-by-step process of creating an Elastic IP and associating it with a Windows EC2 instance:

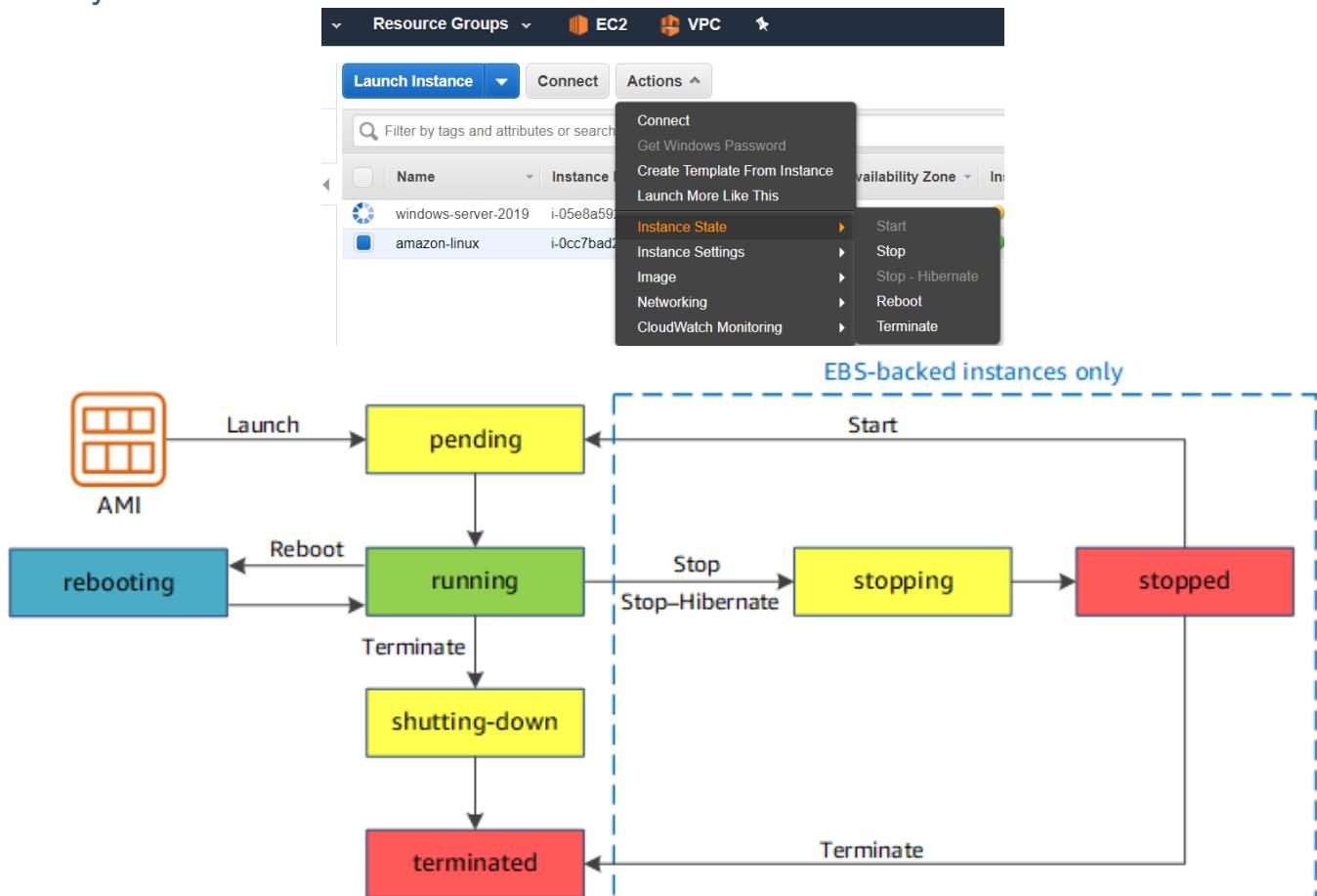
- Screenshot 1: AWS EC2 - Elastic IP addresses**
Shows the 'Elastic IP addresses' page with a red arrow pointing to the 'Allocate Elastic IP address' button.
- Screenshot 2: Allocate Elastic IP address**
Shows the 'Allocate Elastic IP address' dialog with a red arrow pointing to the 'Allocate' button.
- Screenshot 3: Elastic IP address allocated successfully**
Shows the success message 'Elastic IP address allocated successfully.' with a red arrow pointing to the 'Associate this Elastic IP address' button.
- Screenshot 4: Associate Elastic IP address**
Shows the 'Associate Elastic IP address' dialog with a red arrow pointing to the 'Associate' button.
- Screenshot 5: Launch Instance**
Shows the 'Launch Instance' page with a red arrow pointing to the 'public-windows-server-01' instance.
- Screenshot 6: Instance Details**
Shows the instance details for 'public-windows-server-01' with a red arrow pointing to the 'Elastic IP: 15.207.132.129' field.
- Screenshot 7: AWS EC2 - Elastic IP addresses**
Shows the 'Elastic IP addresses' page with a red arrow pointing to the 'Associate Elastic IP address' button.

The screenshot shows the AWS EC2 console with the 'Elastic IP addresses' section selected. A green banner at the top indicates that an elastic IP address has been successfully associated with an instance. The main table lists one assigned IP address:

Name	Type
15.207.132.129	Public IP

Below the table, there is a summary card for the instance 15.207.132.129, which includes tabs for 'Summary' and 'Tags'.

Life Cycle of an EC2 Instance



Instance state	Description
pending	The instance is preparing to enter the running state. An instance enters the pending state when it launches for the first time, or when it is started after being in the stopped state.
running	The instance is running and ready for use.
stopping	The instance is preparing to be stopped or stop-hibernated.
stopped	The instance is shut down and cannot be used. The instance can be started at any time.
shutting-down	The instance is preparing to be terminated.
terminated	The instance has been permanently deleted and cannot be started.

- Shutting down from the OS is equivalent to Stop option in AWS

AWS Reference: [Instance lifecycle](#)

EC2 Instance Machine Types

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

Family	Instance sizes	Use Cases
General Purpose	A1, T3, T3a, T2, M6g, M5, M5a, M5n, M4	<ul style="list-style-type: none"> • Low traffic web applications • Small & Medium size database
Compute Optimized	C6g, C5, C5a, C5n, C4	<ul style="list-style-type: none"> • High performance front-end • Video encoding
Memory Optimized	R6g, R5, R5a, R5n, R4, X1e, X1, High Memory, z1d	<ul style="list-style-type: none"> • High performance Databases • Distributed memory caches
Storage Optimized	I3, I3en, D2, H1	<ul style="list-style-type: none"> • Data warehousing • Log / Data processing applications
Accelerated Computing (GPU Optimized)	P3, P2, Inf1, G4, G3, F1	<ul style="list-style-type: none"> • 3D Application streaming • Machine learning

AWS reference: [Amazon EC2 Instance Types](#)

Meta Data & User Data of EC2 Instances

- In memory key value pairs that are accessible from the EC2 instance
- **Instance Meta Data:** Filled by AWS when you spin-up your EC2. It is the data about your instance. Can be used to configure and manage running instances via Scripts (e.g. Python Automated tasks)
- To view the meta data keys of a Linux EC2, use [curl http://169.254.169.254/latest/meta-data](http://169.254.169.254/latest/meta-data) then embed the 'key' name at the last of the URL as shown below to get the value. e.g.

<http://169.254.169.254/latest/meta-data/instance-type>

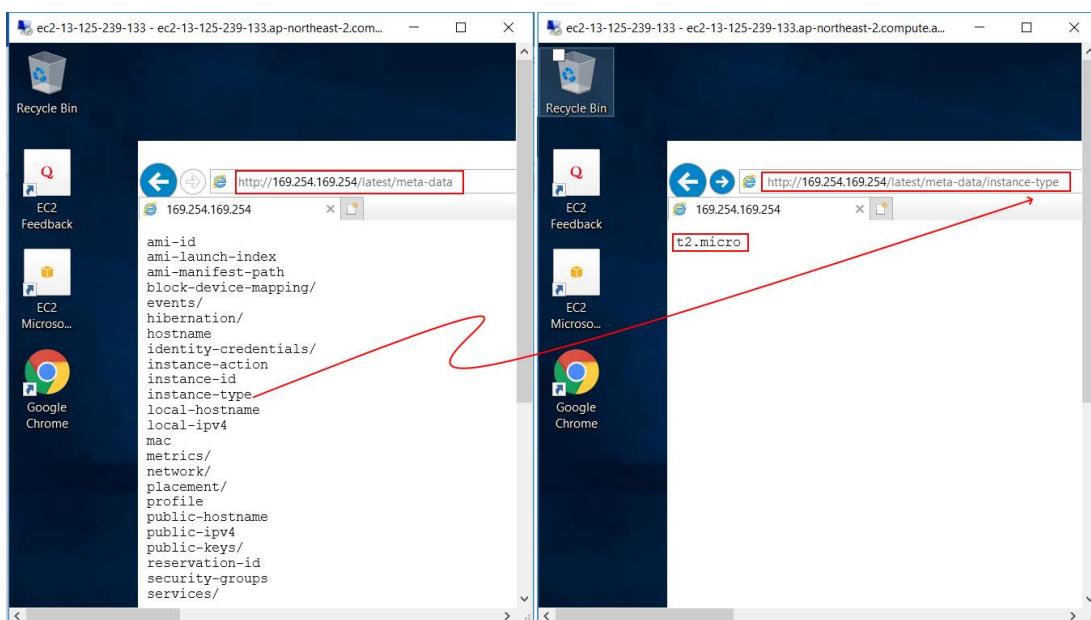
```

ec2-user@ip-172-31-2-252:~$ curl http://169.254.169.254/latest/meta-data
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hibernation/
hostname
identity-credentials/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
managed-ssh-keys/
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
[ec2-user@ip-172-31-2-252 ~]$ curl http://169.254.169.254/latest/meta-data/instance-type
t2.micro
[ec2-user@ip-172-31-2-252 ~]$ 
[ec2-user@ip-172-31-2-252 ~]$ 

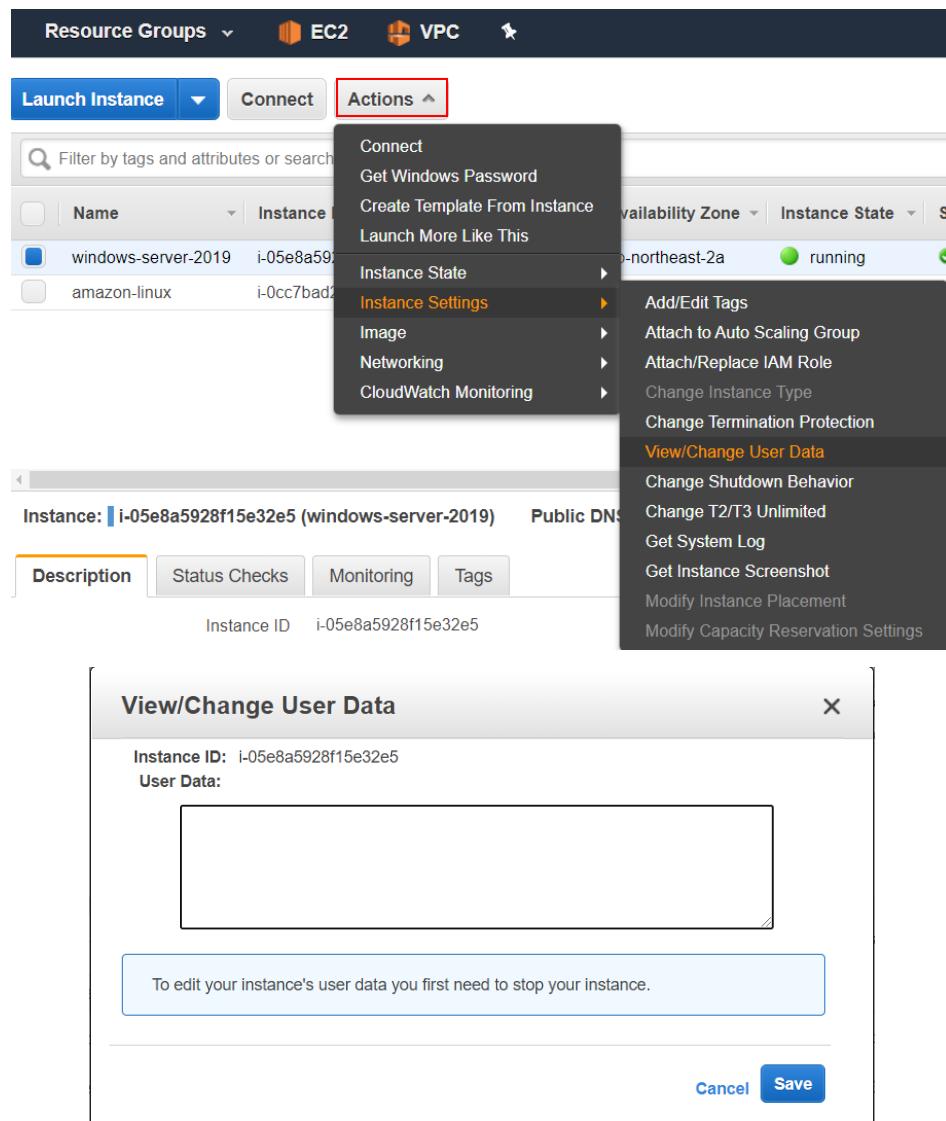
```

- To view the metadata of a Windows EC2, use browse <http://169.254.169.254/latest/meta-data> then embed the 'key' name at the last of the URL as shown below to get the value. e.g.

<http://169.254.169.254/latest/meta-data/instance-type>



- **Instance User Data:** Can be passed to the instance at launch, used to run startup scripts, can be used to perform common automated tasks
- It can be Linux bash scripts or Windows batch / PowerShell scripts
- Use this URL to obtain User Data <http://169.254.169.254/latest/user-data>



Note: Lab for User date is covered in the Web App hosting section.

Storage Options for EC2 Instances

Now let's take a look at the Root Volume configurations of an EC2 Instance. When we start an EC2 instance, the fundamental things that you have to choose what sort of storage you want? specifically the root volume storage where your OS is located.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links for New EC2 Experience, Services (selected), Resource Groups, EC2 (selected), VPC, and various reports and limits. The main area shows a list of instances. A red arrow points to the 'Instances' link in the sidebar. Another red box highlights the 'Description' tab in the instance details panel. The instance details panel shows the following information for the selected instance (i-05e8a5928f15e32e5):

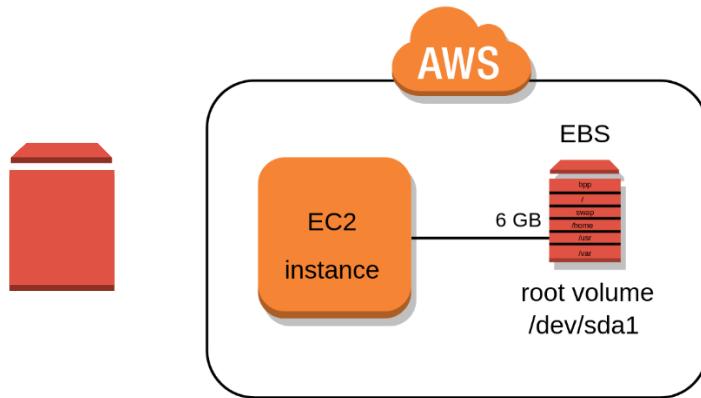
	Value
Instance ID	i-05e8a5928f15e32e5
Public DNS (IPv4)	ec2-3-34-131-94.ap-northeast-2.compute.amazonaws.com
Instance state	running
Instance type	t2.micro
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS	ip-172-31-15-235.ap-northeast-2.compute.internal
Private IPs	172.31.15.235
Public IP	3.34.131.94
IPv6 IPs	-
Elastic IPs	-
Availability zone	ap-northeast-2a
Security groups	windows-server-security-group, view inbound rules, view outbound rules
Scheduled events	No scheduled events
AMI ID	Windows_Server-2019-English-Full-BASE-2020.06.10 (ami-0f8ea5913b2505d3)
Platform details	Windows
Usage operation	RunInstances:0002
Source/dest. check	True
T2/T3 Unlimited	Disabled
EBS-optimized	False
Root device type	ebs
Root device	/dev/sda1
Block devices	/dev/sda1

Where is the root volume stored in an EC2? We have two options available,

1. Instance Store Backed

- It is a Physical Drive that is attached to the host computer that hosts the EC2 VM
- When you stop / terminate the EC2, the Instance Store loses its data, it evaporates
- Data on Instance Store is non-persistent
- Max size 5GB
- Can't be stopped

2. AWS Elastic Block Store (EBS)



- We don't see any configuration menu / button for EBS, this is a storage mechanism tied closely to EC2 instance
- We can call it as Virtual Hard Disk in the cloud (like .VHD file in vmware)
- We could create EBS at the time of creation of EC2 or independently
- Based on SSD (1GB to 16TB capacity) or Magnetic drive (1GB to 1TB capacity)
- We can take snapshot of EBS and store in *S3
- Pay for what you provision, IOPS (Input Output Per Second) also charged. Also, charges are based on region.
- Default behavior is to delete the EBS volume along with the EC2 while terminating, but you can override that to keep data on EBS store.
- Data on Instance Store is persistent, Can be stopped and modified the configurations inside

While selecting the AMI, we can see the root device type.

Step 1: Choose an Amazon Machine Image (AMI)

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.

Search by Systems Manager parameter

AMIs	Description	Select	Root device type	Virtualization type	ENI Enabled
amzn-ami-hvm-2018.03.0.20200206.0-x86_64-ebs	ami-063635011144132da Amazon Linux AMI 2018.03.0.20200206.0 x86_64 HVM ebs	Select	ebs	hvm	Yes
amzn-ami-hvm-2018.03.0.20200206.0-x86_64-gp2	ami-0ecd78c22823e02ef Amazon Linux AMI 2018.03.0.20200206.0 x86_64 HVM gp2	Select	ebs	hvm	Yes
amzn-ami-hvm-2018.03.0.20200206.0-x86_64-s3	ami-00824ec7312925146 Amazon Linux AMI 2018.03.0.20200206.0 x86_64 HVM s3	Select	instance-store	hvm	Yes

The following results for "amzn-ami-hvm-2018.03.0.20200206" were found in other catalogs:

- 4141 results in AWS Marketplace

AWS Marketplace provides partner software that is pre-configured to run on AWS

AWS EBS	AWS S3*
Cloud disk drive	Cloud based storage
Block storage file system based	Object Store
Redundancy - Availability Zone	Redundancy - Region
No access via Internet*	Access via Internet*

The screenshot shows the AWS EBS Volumes page. On the left, there's a navigation sidebar with sections like IMAGES, NETWORK & SECURITY, LOAD BALANCING, and AUTO SCALING. Under the EBS section, 'Volumes' is highlighted with a red arrow. The main area displays a table of volumes:

Name	Volume ID	Size	Type	IOPS	Snapshot	Created	Availability Zone	Status	Alarm Status
windows-server	vol-045a009e65e91519c	30 GiB	gp2	100	snap-0a72c5f7...	June 15, 2020 at 1:08:24 AM UTC+5:30	ap-northeast-2a	in-use	None
amazon-linux	vol-0bb4e99...	8 GiB	gp2	100	snap-08e7474...	June 14, 2020 at 1:08:24 AM UTC+5:30	ap-northeast-2a	in-use	None

Below the table, a modal window is open for the volume 'windows-server-2019'. It shows detailed information including Volume ID, Snapshot, Availability Zone, Encryption, and various metrics like Size, Created, State, and IOPS. A red box highlights the 'Attachment information' section, which shows the instance ID and device path.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a navigation sidebar with sections like EC2 Dashboard, Instances, Images, and Elastic Block Store. 'Elastic Block Store' is highlighted with a red arrow. The main area has three main sections: 'Resources', 'Launch instance', and 'Service health'.

Resources: Displays the following counts: Running instances (0), Elastic IPs (0), Dedicated Hosts (0), Snapshots (0), Volumes (0), Load balancers (0), Key pairs (0), Security groups (1), and Placement groups (0).

Launch instance: A large button labeled 'Launch instance' with a dropdown menu. Below it, a note says 'Note: Your instances will launch in the Asia Pacific (Singapore) Region'.

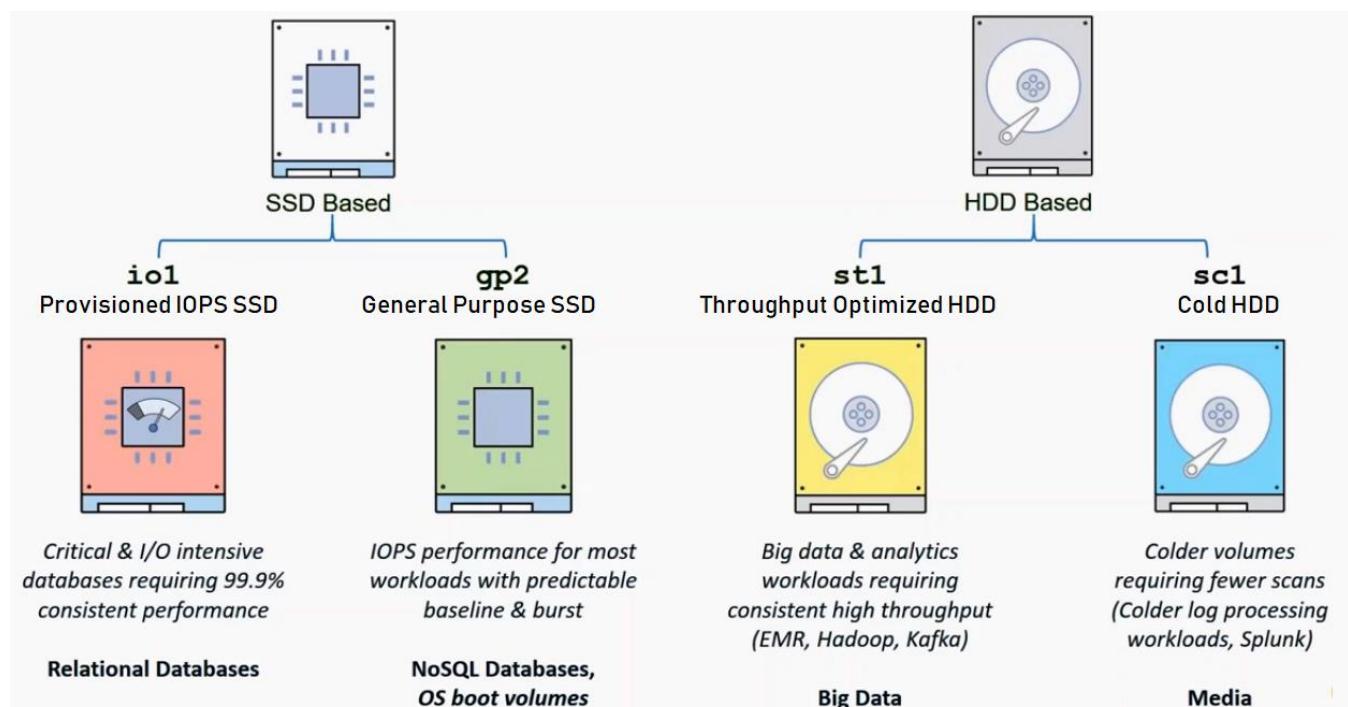
Service health: Shows the status of the service: Region (Asia Pacific (Singapore)) and Status (This service is operating normally).

You can access / create new EBS drive from the Volumes options.

*S3: Simple Storage Service: This will be covered later

- Block storage for EC2 instance (root volume), high performance and low latency.
- We can attach and detach EBS to any one instance at a time
- Scope EBS is AZ
- To move EBS between AZs, take a snapshot and move it in to different AZs
- Fault tolerance is built in to EBS, AWS will create replicated copy in same AZ called Secondary volume. It is not accessible but automatically used in case of primary volume failure.
- We can alter the performance of EBS (IOPs, Latency, Throughput)
- **IOPS:** Input Output (Read/Write) per second
- **Latency:** Time in milli seconds between IO submission and complete. Consider in Disk level.
- **Throughput:** Read/Write transfer rate (MB/s); IOPS x Data Rate
- Which one to select? IOPS Optimized is preferred for Mobile Data (small data in transaction). Movie Streaming basically takes Throughput optimized volumes.
- We can increase the Disk Size but not decrease
- To reduce the size, create a snapshot of bigger volume and create a new small volume based on the snapshot.
- We can change the volume types (e.g. io1 to st1) but not all the combinations.
- We can increase or decrease IOPS

Types of EBS



AWS Reference: [Types of EBS](#)

[LAB] Managing EBS

- Snapshots are the image of an EBS, it can be used to backup an EBS and create new from it

The left screenshot shows the EC2 Dashboard with the 'Volumes' section selected. A context menu is open over a volume named 'windows-with-python', with 'Create Snapshot' highlighted.

The right screenshot shows the 'Create Snapshot' wizard. It displays the selected volume ('vol-05eec8ad7fd8aa3efc') and a description ('windows-volume'). The 'Create Snapshot' button is at the bottom right.

- We can modify the volume type whenever required

The left sidebar shows the 'Elastic Block Store' section with 'Volumes' selected. A context menu is open over a volume named 'windows-with-python', with 'Actions' selected.

The right side shows the 'Modify Volume' dialog. The 'Volume ID' is 'vol-05eec8ad7fd8aa3efc'. The 'Volume Type' dropdown is set to 'General Purpose SSD (gp2)'. Other options shown are 'Provisioned IOPS SSD (io1)' and 'Magnetic (standard)'. The 'Modify' button is at the bottom right.

- From the snapshot, we can create new volume

The screenshot shows the AWS EC2 Dashboard with the 'Solutions Architect Associate' badge in the top right corner. The left sidebar navigation includes 'Events', 'Tags', 'Limits', 'Instances' (with sub-options like 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', and 'Capacity Reservations'), 'Images' (with 'AMIs'), and 'Elastic Block Store' (with 'Volumes' and 'Snapshots'). A red arrow points to the 'Snapshots' link. The main content area displays a table for a selected snapshot:

Description		Permissions	Tags
Snapshot ID	snap-03159b63844de9404		Progress 100%
Status	completed		Capacity 30 GiB
Volume	vol-05eec8ad7fdcaa3efc		Encryption Not Encrypted
Started	August 2, 2020 at		KMS Key ID

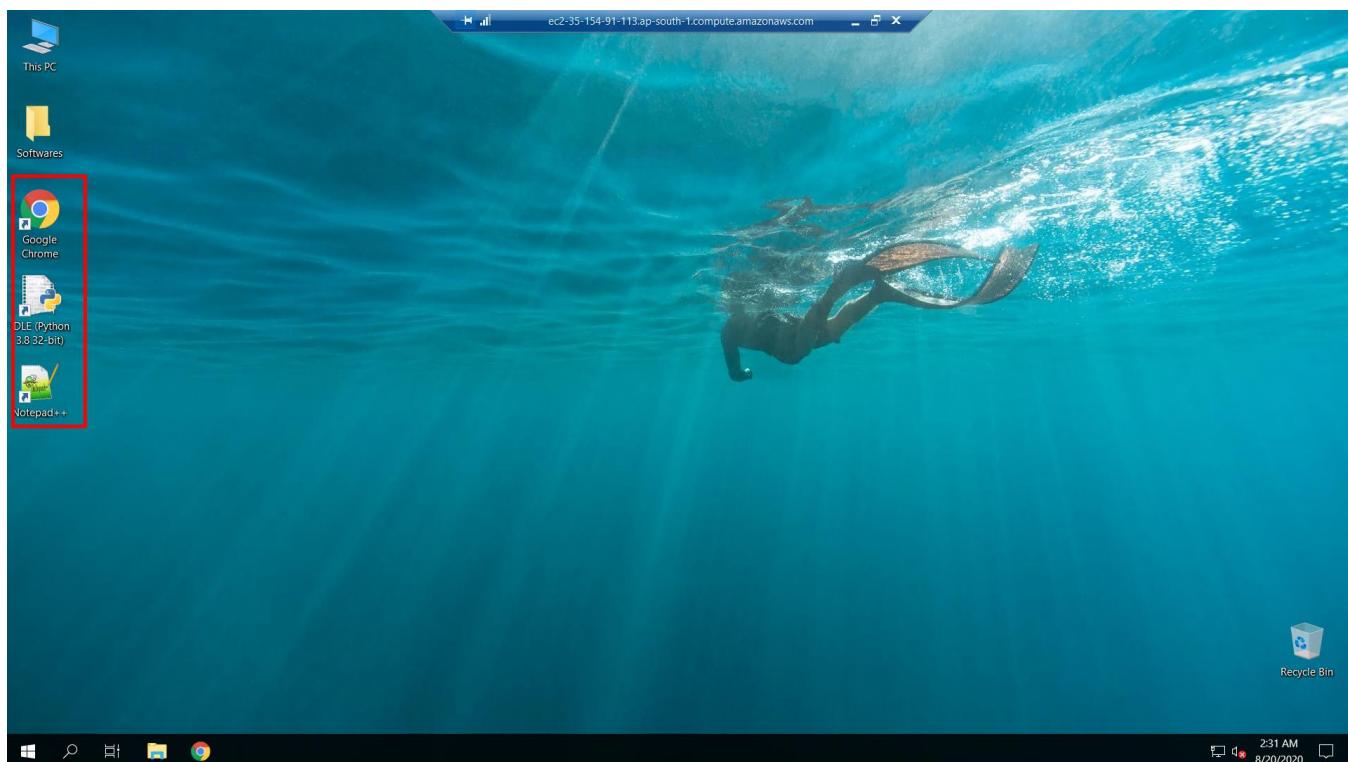
A context menu is open over the snapshot table, with a red arrow pointing to the 'Actions' button. The menu options are: Delete, Create Volume, Manage Fast Snapshot Restore, Create Image, Copy, Modify Permissions, and Add/Edit Tags.

EC2 Placement Group

- When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures.
- You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload.
 - **Cluster** – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.
 - **Partition** – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads like Hadoop.
 - **Spread** – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

Creating Custom AMI

- AMI provides information required to launch an EC2 instance in cloud. AMI contains the root volume for the instance to keep the OS files.
- Once we have an instance, we can install software, patches, etc. and build another image from it. This will save some time in future. You will have the software pre-installed on new image.
- Select the instance >> Actions >> Image >> Create Image
- You can build new EC2 instances from this image in future.



The screenshots illustrate the steps to create a custom AMI from an existing EC2 instance. In the first screenshot, the 'Create Image' option is selected in the 'Actions' menu. In the second screenshot, the 'Create Image' dialog is open, showing the configuration details: Instance ID (i-03b171651edf4ee2c), Image name (win-server-2019-Chrome-Python-N++), and Image description (Windows Server 2019 with pre-installed Chrome, Python). The 'Create Image' button is highlighted with a red arrow.

The figure consists of three screenshots of the AWS EC2 console, arranged vertically.

Screenshot 1: Instances - Create Image

This screenshot shows the 'Create Image' dialog box. The 'Image name' field is set to 'win-2019-Chrome-Python-NotepadPP' and the 'Image description' field is set to 'win-2019-Chrome-Python-NotepadPP'. The 'Create' button is highlighted with a red box.

Screenshot 2: EC2 Image Builder - Details

This screenshot shows the details of the newly created AMI, 'ami-0e37302f2516ce5df'. It includes fields for AMI ID, AMI Name, Owner, Source, Status, and State Reason.

Screenshot 3: Choose AMI - Step 1

This screenshot shows the 'Step 1: Choose an Amazon Machine Image (AMI)' step of the instance creation wizard. The 'My AMIs' section is selected, showing the 'win-2019-Chrome-Python-NotepadPP' AMI. The 'Select' button is highlighted with a red box.

EC2 Pricing Model

On-Demand / Pay as you Go

- With On-Demand instances, you pay for compute capacity by the hour or the second depending on which instances you run.
- No longer-term commitments or upfront payments are needed.

Reserved

- Reserved Instances provide you with a significant discount (up to 75%) compared to On-Demand instance pricing.
- Cheap and long-term lock (1 to 3 years)

Savings Plans

- Savings Plans is a flexible pricing model that provides savings of up to 72% on your AWS compute usage.
- For a known burst period (e.g. Black Friday sale)

Spot

- Amazon EC2 Spot instances allow you to request spare Amazon EC2 computing capacity for up to 90% off the On-Demand price
- Other AWS users sell their Reserved or Savings Plans if they are not using it, its like bidding

Dedicated Host

- A Dedicated Host is a physical EC2 server dedicated for your use.

AWS Reference: [Amazon EC2 pricing](#)

Tracking your AWS Free Tier usage

- You can track your AWS Free Tier usage to help you stay under the AWS Free Tier limits.
- AWS automatically provides alerts through AWS Budgets to notify you by email when you exceed 85 percent of your AWS Free Tier limits for each service.

The screenshot shows the AWS Preferences page. On the left sidebar, 'Billing' is highlighted with a red arrow. In the main content area, there's a section titled 'Receive Free Tier Usage Alerts' with a checked checkbox. Below it, an 'Email Address' input field contains 'jaseemci@gmail.com'. A blue 'Save preferences' button is at the bottom. Red arrows point from the top of the 'Billing' sidebar and from the right side of the 'Receive Free Tier Usage Alerts' section towards the 'Save preferences' button.

AWS Free Tier limit alert

no-reply-aws@amazon.com Jul 21, 2020, 5:50 PM to me



AWS Free Tier usage limit alerting via AWS Budgets 07/21/2020

Dear AWS Customer,

Your AWS account 618927232701 has exceeded 85% of the usage limit for one or more AWS Free Tier-eligible services for the month of July.

AWS Free Tier Usage as of 07/21/2020 AWS Free Tier Usage Limit

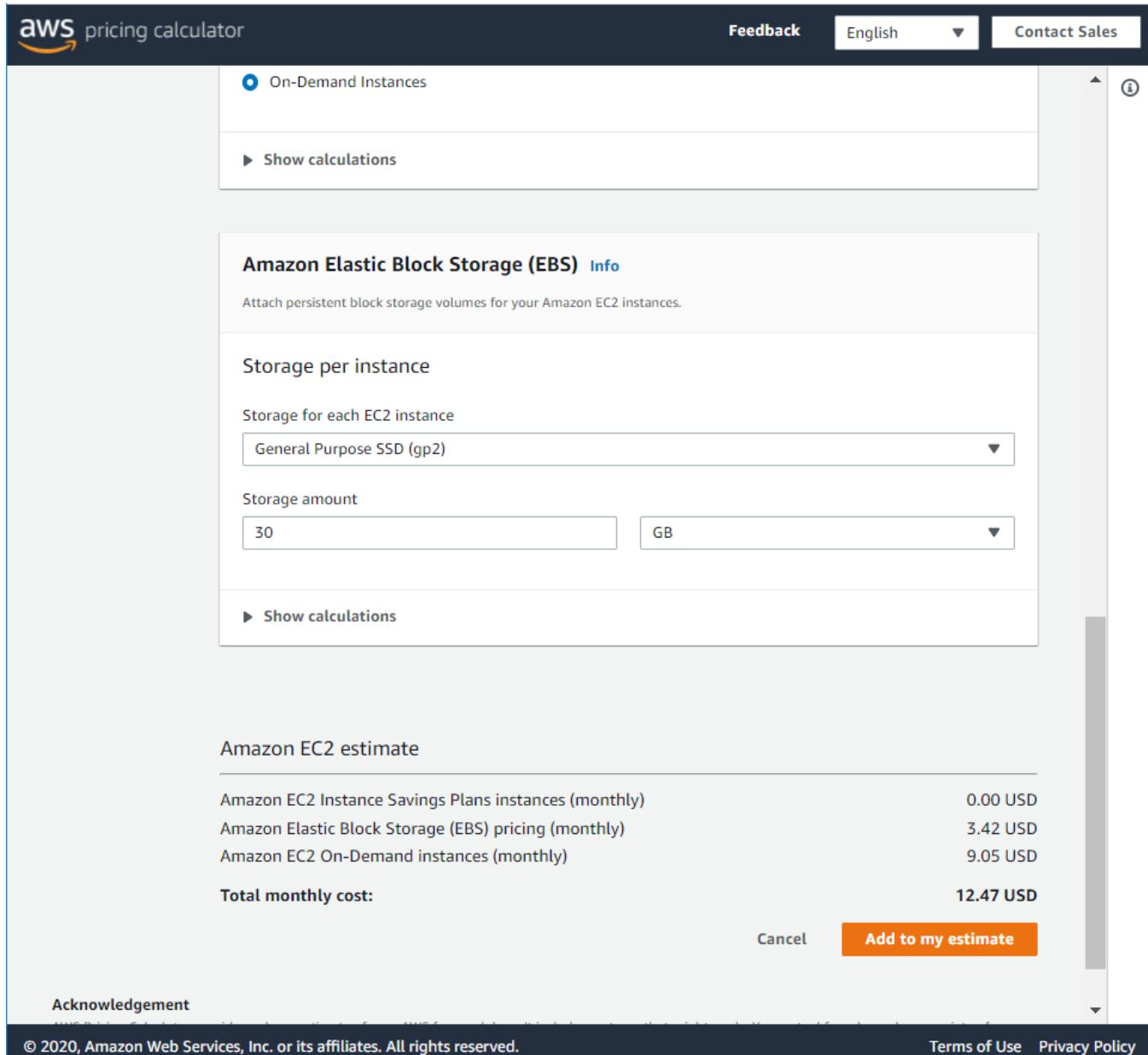
640.0 Hrs

750 hours of Amazon EC2 Microsoft Windows Server t2.micro instance usage

To learn more about your AWS Free Tier usage, please access the [AWS Billing & Cost Management Dashboard](#). You can find more information on AWS Free Tier [here](#).

AWS Pricing Calculator

- [AWS Pricing Calculator](#) gives estimate that fits your unique business or personal needs with AWS products and services.



The screenshot shows the AWS Pricing Calculator interface. At the top, there's a navigation bar with the AWS logo, 'pricing calculator', 'Feedback', 'English' dropdown, and 'Contact Sales'. A sidebar on the left has a radio button selected for 'On-Demand Instances' and a link to 'Show calculations'. The main content area starts with a section for 'Amazon Elastic Block Storage (EBS) [Info](#)'. It says 'Attach persistent block storage volumes for your Amazon EC2 instances.' Below this is a 'Storage per instance' section. Under 'Storage for each EC2 instance', a dropdown menu is set to 'General Purpose SSD (gp2)'. Under 'Storage amount', a text input field contains '30' and a dropdown menu next to it is set to 'GB'. At the bottom of this section is a 'Show calculations' button. Further down, there's a section for 'Amazon EC2 estimate' with a table of monthly costs:

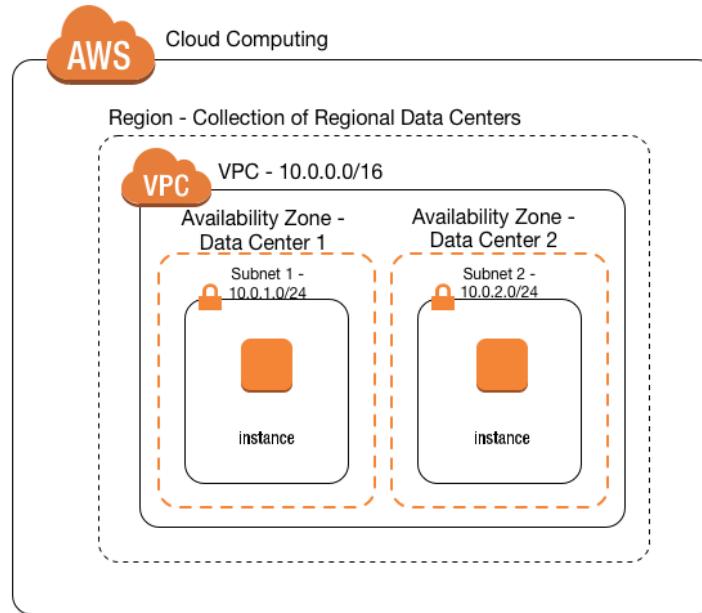
Amazon EC2 Instance Savings Plans instances (monthly)	0.00 USD
Amazon Elastic Block Storage (EBS) pricing (monthly)	3.42 USD
Amazon EC2 On-Demand instances (monthly)	9.05 USD
Total monthly cost:	12.47 USD

At the bottom right of the calculator are 'Cancel' and 'Add to my estimate' buttons. The footer contains an 'Acknowledgement' link, copyright notice ('© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.'), and links for 'Terms of Use' and 'Privacy Policy'.

AWS VPC (Virtual Private Cloud)

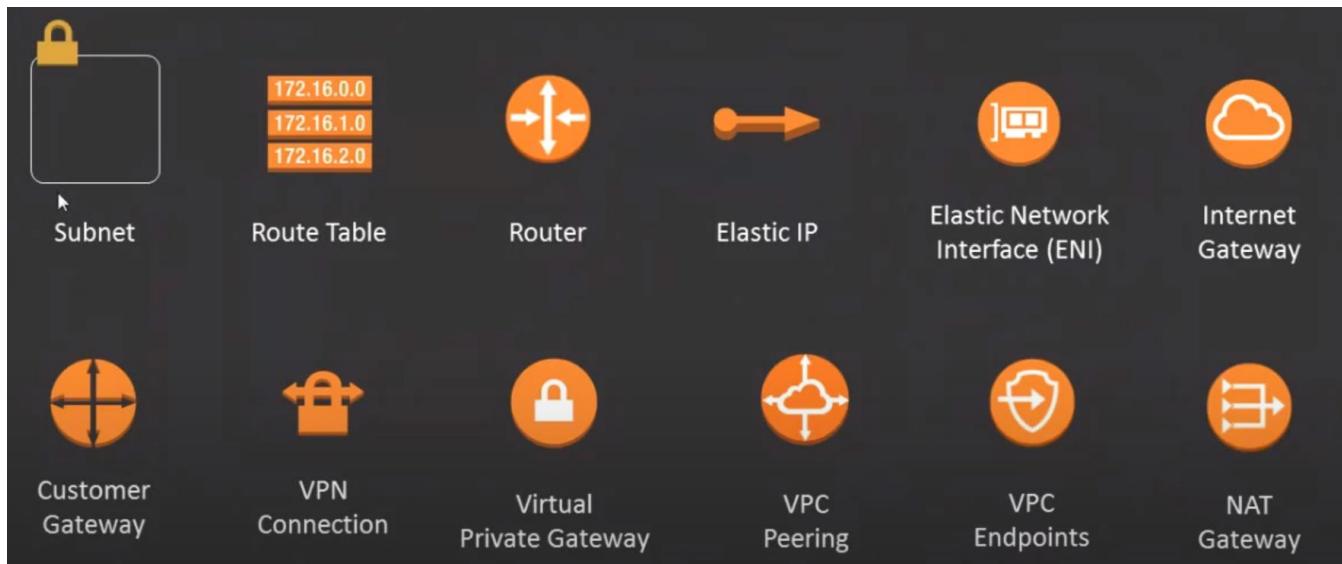


- Isolated user defined network inside AWS (like VLAN in switch)
- Logically isolated subnets
- Reserve IP Address ranges and assign to resources
- Setup Route Tables, Firewalls, Gateways
- Use Network ACL, IAM to control access



- Remotely connect on-prem DC network with VPN or Direct Connect
- Multiple VPCs can be peered across Region
- Subnet IPs are in Private range, once we assign Internet Gateway, we call it as Public Subnet
- VPC allocates a part of the cloud as your own network, we can control, manage IP, manage security, etc.
- When AWS launched initially the entire AWS cloud was a flat network.
- Forced to apply instance level security since the network is not managed / maintained by us
- We can connect to the VPC network and then access EC2s with private IPs
- We create Subnets in VPC so that we can provide access level
- Virtual Private Gateway (1.25Gbps) used to connect on-prem DC
- AWS has created Default VPCs in every Region of your account.

Components of VPC (Overview)



- **Subnet:** We create Subnet inside VPC (later we can name it as Public or Private). VPC spans across Region and Subnet spans within AZ.
- **Route Table:** Routing table holds information about where traffic to be sent. Router works based on the information in the route table. Within the VPC we use static routes. When it comes to VPN and Direct Connect, we go for Dynamic routing.
- **Router:** Router goes ahead and looks at the route table and asks the route table that how do I go to the other system / device.
- **Elastic IP:** Static public IP, when we start EC2 in default VPC, dynamically we get public IP to the EC2 instance along with the private IP. If we restart EC2, the public IP may change. When we use Elastic IP, the public IP will be statically assigned to the EC2 until you release.
- **Elastic Network Interface (ENI):** Virtual network card for EC2 instance. When we assign IP (Elastic or Dynamic), those IPs always get assigned to the ENI and the ENI gets assigned to EC2. We can always re-attach the ENI from one EC2 to another EC2 instance.
- **Internet Gateway (IGW):** Gateway that allows you to connect to internet
- **Customer Gateway:** The router in on-prem data center for VPN
- **VPN Connection:** IPSec VPN connection between Customer Gateway and Virtual Private Gateway
- **Virtual Private Gateway:** Router in AWS side for VPN
- **VPC Peering:** To interconnect 2 VPCs together, 1st VPC sends a request and the other VPC accepts it.
- **VPC Endpoints:** Connectivity that allows you to talk to certain AWS services privately from VPC. For example, EC2 communicating to S3 goes over via internet, VPC Endpoint gives direct connection to S3 privately. It doesn't go out of AWS network.
- **NAT Gateway:** Provides internet access to devices in private subnet of PVC. NAT Gateway sits in the Public subnet. Only outbound connection and response to that connections are allowed, no new inbound connections from internet.

IP Classless Inter-Domain Routing CIDR Block

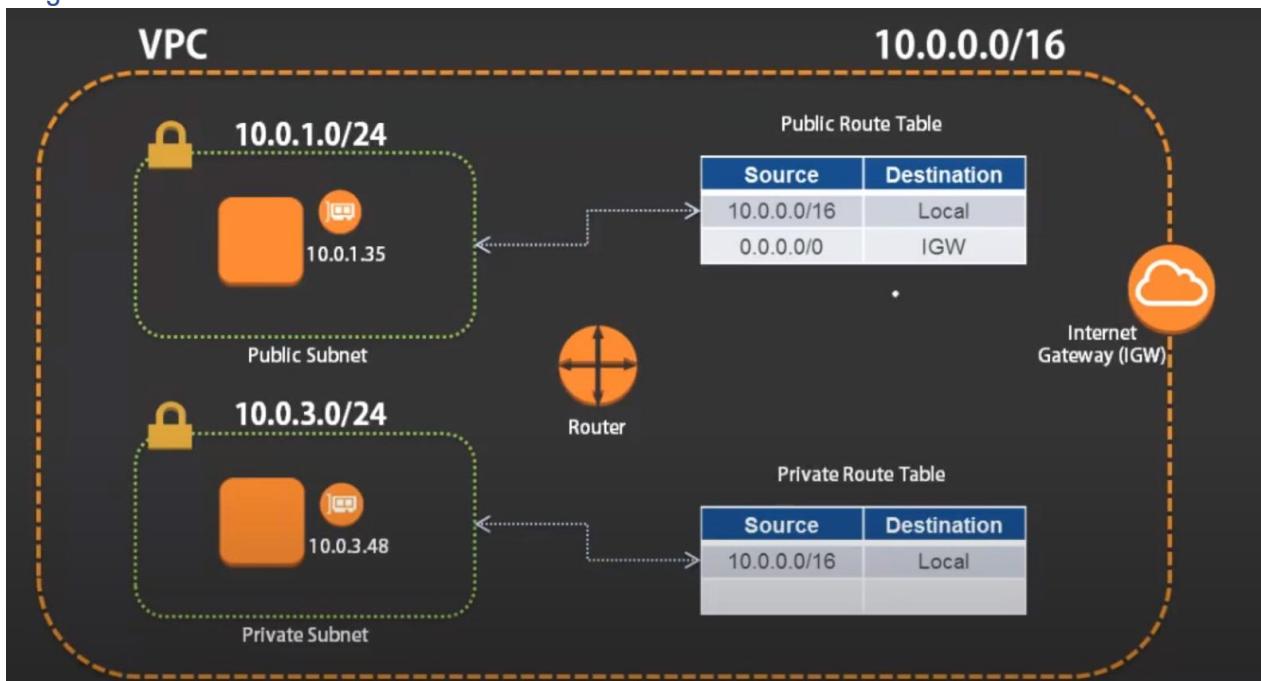
- Notation that shows exactly how many IPs are there in a network, how large the network is, and additional information about the network. E.g. 10.0.0.0/16 (Network Prefix IP Addressing Method)
- Range: 10.0.0.0 to 10.0.255.255
- When we create VPC, we need to specify CIDR range of that VPC. We can add additional CIDR range to VPC, but existing one can't be deleted.
- 10.0.1.2/32 - Represents one IP using CIDR notation.
- Subnet calculator: <http://jodies.de/ipcalc>
- VPC capacity is between /28 (14 +2 IPs) to /16 (65534+2 IPs)
- We can bring our public IPs to AWS (Should talk to provider and AWS)
- IP Address Reservations:**
 - .0 - Network Address
 - .1 - Reserved by AWS for the internal VPC Router
 - .2 - Reserved by AWS for the internal DNS Server
 - .3 - Reserved by AWS for future use
 - .255 - Broadcast Address. Broadcast is not supported with in VPC.

Subnets



- Create subnets to divide the larger chunks of IPs
- Subnets are the Sub Networks of VPC, they have to fall in the IP range (CIDR) of VPC
- E.g.: CIDR 10.0.0.0/16
 - Subnet 1: 10.0.1.0/24 (10.0.1.0 to 10.0.1.255)
 - Subnet 2: 10.0.2.0/24 (10.0.2.0 to 10.0.2.255)
 -
 - Subnet 255: 10.0.255.0/24 (10.0.1.0 to 10.0.255.255)

Routing in VPC



- When we create Subnets in VPC, it is just a smaller chunk of VPC CIDR, there is nothing called Public or Private.
- The moment we associate Route Table on subnet, based on the Route Table information, we call the subnet as Public or Private.
- Route Table is the one that decides whether the subnet is Public or Private.
- Apart from the Public Route Table, your EC2 instance should have a Dynamic Public IP or Elastic IP

[LAB] VPC Configurations

Step 1: Creating VPC

The screenshot shows the AWS Services menu with the VPC icon highlighted in red. Other visible services include EC2, S3, IAM, Billing, and EC2 Image Builder.

- Open AWS Console >> Services >> VPC

The screenshot shows the VPC Dashboard with the 'Launch VPC Wizard' button highlighted in red. It displays resources by region, including VPCs, Subnets, Route Tables, Internet Gateways, and Security Groups.

(Optional)

- We have an option to use Launch VPC Wizard to create a VPC, however that is too easy and you might not understand what's happening behind the scene.

The screenshot shows the 'Create VPC' page. A red arrow points to the table where the default VPC is listed with the ID 'vpc-8fd2cde7'. The table also shows the state 'available' and the IPv4 CIDR '172.31.0.0/16'.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
vpc-8fd2cde7	vpc-8fd2cde7	available	172.31.0.0/16	-

- Go to Your VPCs, you can see the default VPC created by AWS here. Every region will have default VPCs created.
- We can also see the IP CIDR block as well
- Go ahead and hit Create VPC button

VPCs > Create VPC

Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an IPv6 CIDR block with the VPC.

Name tag: ap-south-1-vpc

IPv4 CIDR block: 172.16.0.0/16

IPv6 CIDR block: No IPv6 CIDR Block
 Amazon provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy: Default

* Required

Cancel **Create**

- Go to Your VPCs, you can see the default VPC created by AWS here. Every region will have default VPCs created.
- We can also see the IP CIDR block as well
- Go ahead and hit Create VPC button

VPCs > Create VPC

Create VPC

The following VPC was created:

VPC ID: vpc-00c3b6efcf0ee9add

Close

- Now we have successfully created VPC

New VPC Experience
Tell us what you think

VPC Dashboard New

Filter by VPC:
Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs **←**

- Subnets
- Route Tables
- Internet Gateways New
- Egress Only Internet Gateways
- DHCP Options Sets New
- Elastic IPs New
- Managed Prefix Lists New
- Endpoints
- Endpoint Services
- NAT Gateways
- Peering Connections

Create VPC Actions **Newly created VPC**

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
ap-south-1-vpc	vpc-00c3b6efcf0ee9add	available	172.16.0.0/16	-
aws-default-vpc	vpc-8fd2cde7	available	172.31.0.0/16	-

VPC: vpc-8fd2cde7

Description	CIDR Blocks	Flow Logs	Tags
VPC ID: vpc-8fd2cde7 State: available IPv4 CIDR: 172.31.0.0/16 IPv6 Pool: - Network ACL: acl-592cea32 DHCP options set: dopt-a21de7c9 Owner: 618927232701	Tenancy: default Default VPC: Yes IPv6 CIDR: - DNS resolution: Enabled DNS hostnames: Enabled Route table: rtb-9e7		

- In the VPC services page, you can see the newly created VPC and AWS Default VPC.

Note: There won't be any name tag for default VPC, it is recommended to provide a name tag for the default VPC.

Step 2: Enable Public DNS Hostnames

- When you create VPC, it doesn't give any Public DNS hostnames to EC2 instances by default.
- To enable this, select the VPC > Actions > Edit DNS Hostnames > Yes > Save

The screenshot shows two side-by-side views of the AWS VPC console. On the left, the 'Your VPCs' list is shown with a dropdown menu open over the 'ap-south-1' VPC entry. The menu options are: Delete VPC, Edit CIDRs, Create Default VPC, Create flow log, Edit DHCP options set, Edit DNS resolution, **Edit DNS hostnames**, and Add/Edit Tags. A red arrow points from the text 'To enable this, select the VPC > Actions > Edit DNS Hostnames > Yes > Save' to the 'Edit DNS hostnames' option in the menu. On the right, the 'Edit DNS hostnames' configuration page is displayed. It shows the VPC ID 'vpc-00c3b6efcf0ee9add' and the 'DNS hostnames' checkbox is checked with the value 'enable'. A red arrow points from the same text above to this checkbox. The 'Save' button is visible at the bottom right of the form.

Step 3: Creating Subnets

New VPC Experience
Tell us what you think

VPC Dashboard New
Filter by VPC:
Select a VPC

VIRTUAL PRIVATE CLOUD
Your VPCs
Subnets ←
Route Tables
Internet Gateways New
Egress Only Internet Gateways
DHCP Options Sets New
Elastic IPs New
Managed Prefix Lists New
Endpoints
Endpoint Services
NAT Gateways
Peering Connections

Feedback English (US) Privacy Policy Terms of Use

- Go to the Subnet options in VPC

Dashboard >> You can observe the default Subnets created by AWS itself.

Note: It is recommended to provide a friendly name to the default subnets.

New VPC Experience
Tell us what you think

VPC Dashboard New
Filter by VPC:
Select a VPC

VIRTUAL PRIVATE CLOUD
Your VPCs
Subnets ←
Route Tables
Internet Gateways New
Egress Only Internet Gateways
DHCP Options Sets New
Elastic IPs New
Managed Prefix Lists New
Endpoints
Endpoint Services
NAT Gateways
Peering Connections

I have given friendly names to default Subnets

Subnet: subnet-9f2d29f7

Description	Flow Logs	Route Table	Network ACL	Tags
Subnet ID: subnet-9f2d29f7				State: available
VPC: vpc-8fd2cde7				IPv4 CIDR: 172.3
Available IPv4 Addresses: 4091				IPv6 CIDR: -
Availability Zone: ap-				Route Table: rtb-9e

Feedback English (US) Privacy Policy Terms of Use

- I have given some friendly names to the default Subnets.

- Go ahead and click Create Subnet button

- In our lab practice we are creating 4 subnets (2 Private in 2 AZs and 2 Public in 2 AZs)

- ap-south-1-public-subnet-1a
- ap-south-1-public-subnet-1b
- ap-south-1-private-subnet-1a
- ap-south-1-private-subnet-1b

Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag	ap-south-1-public-subnet-1a
VPC*	vpc-00c3b6efcf0ee9add
Availability Zone	ap-south-1a
VPC CIDRs	CIDR Status Status Reason
172.16.0.0/16 associated	
IPv4 CIDR block*	172.16.1.0/24

* Required

Create

Feedback English (US) Privacy Policy Terms of Use

I have configured below subnets,

SUBNET	CIDR
ap-south-1-public-subnet-1a	172.16.11.0/24
ap-south-1-public-subnet-1b	172.16.12.0/24
ap-south-1-private-subnet-1a	172.16.21.0/24
ap-south-1-private-subnet-1b	172.16.22.0/24

- We can apply a filter in the Subnets and see all the Subnets that we created.

The screenshot shows the AWS VPC Subnet list interface. A red arrow points to the search bar at the top, which contains the text "VPC vpc-00c3b6efcf0ee9add". The search results table displays four subnets under the "ap-south-1" VPC:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4
ap-south-1-public-subnet-1a	subnet-05bd2cc86c1e09ec5	available	vpc-00c3b6efcf0ee9add ap-south-1-vpc	172.16.11.0/24	251
ap-south-1-public-subnet-1b	subnet-09f347fe337e174e9	available	vpc-00c3b6efcf0ee9add ap-south-1-vpc	172.16.12.0/24	251
ap-south-1-private-subnet-1b	subnet-0acb68796db03fe43	available	vpc-00c3b6efcf0ee9add ap-south-1-vpc	172.16.22.0/24	251
ap-south-1-private-subnet-1a	subnet-0f483584b239c2f5c	available	vpc-00c3b6efcf0ee9add ap-south-1-vpc	172.16.21.0/24	251

Step 4: Enable Auto Assign Public IP for Public Subnet

- Now we have Public and Private Subnets, let's enable Auto Assign Public IP settings on the two Public subnets.
- Select the Public Subnet >> Actions >> Modify auto-assign IP settings >> Enable auto-assign public IPv4 address.
- EC2 instances on the Public subnet will be automatically assigned with Public IP.

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under 'Subnets', there is a list of subnets:

Name	Subnet ID	State	VPC
ap-south-1-public-subnet-1a	subnet-05bd2cc86c1e09ec5	available	vpc-00c3b6efcf0ee9add
ap-south-1-public-subnet-1b	subnet-09347fe337e174e9	available	vpc-00c3b6efcf0ee9add
ap-south-1-private-subnet-1b	subnet-0acb68796db03fe43	available	vpc-00c3b6efcf0ee9add
ap-south-1-private-subnet-1a	subnet-0f483584b239c2f5c	available	vpc-00c3b6efcf0ee9add

In the main pane, details for the selected subnet ('ap-south-1-public-subnet-1a') are shown:

- VPC: vpc-00c3b6efcf0ee9add
- IPv4 CIDR: 172.16.11.0/24
- Available IPv4 Addresses: 250
- Availability Zone: ap-south-1a (aps1-az1)
- Network ACL: acl-0882d1c755ca03a42
- Auto-assign public IPv4 address: No
- Outpost ID: -
- Owner: 618927232701

The screenshot shows the 'Modify auto-assign IP settings' dialog for the selected subnet. The 'Auto-assign IPv4' checkbox is checked, and the 'Enable auto-assign public IPv4 address' checkbox is also checked. A red arrow points to the 'Save' button.

The screenshot shows the AWS VPC Dashboard. The subnet 'ap-south-1-public-subnet-1a' is selected. In the main pane, the 'Auto-assign public IPv4 address' setting has been changed to 'Yes'. Other details remain the same as in the previous screenshot.

Step 5: Add Internet Gateway and Associate to Public Subnet

The screenshot shows the AWS VPC console with the 'Internet Gateways' section selected. A red arrow points from the 'Internet' link in the left sidebar to the 'Internet Gateways' section. The main pane displays a table with one row for the default internet gateway, named 'aws-default-igw' with ID 'igw-b3e840db'. The 'Create internet gateway' button is visible at the top right of the table.

- Go to Internet Gateways >> Create internet gateway

Note: You can see the default Internet Gateway here.

- This will provide connectivity to internet from VPC

The screenshot shows the 'Create internet gateway' wizard. The first step, 'Internet gateway settings', has a 'Name tag' input field containing 'ap-south-1-igw'. A red arrow points to this input field. The 'Create internet gateway' button is at the bottom right of the form.

- Provide a Name tag >> Create internet gateway

The screenshot shows the AWS VPC console with the 'Internet Gateways' section selected. A red arrow points from the 'Actions' dropdown menu to the 'Attach to a VPC' button in a green success message box. Below it, the table lists the new gateway 'igw-054a82ce1feb26815 / ap-south-1-igw'.

The screenshot shows the 'Attach to VPC' wizard. The first step, 'Attach to VPC (igw-054a82ce1feb26815)', has a 'Available VPCs' search bar containing 'vpc-00c3b6efcf0ee9add'. A red arrow points to this search bar. The 'Attach internet gateway' button is at the bottom right.

Step 6: Route Table Configuration

- For every VPC we create, AWS will add a specific Route Table for that VPC. You can see Main = Yes in those Route tables.
- We basically need 2 Route Tables now, either you can create 2 new or edit existing one and add another one. (ap-south-1-public-route-table, ap-south-1-private-route-table)

The screenshot shows the AWS VPC Route Tables page. On the left sidebar, under 'Route Tables', 'Route Tables' is selected. In the main content area, there is a table with two rows:

Name	Route Table ID	Explicit subnet associations
ap-south-1-public-route-table	rtb-0975ced770c1f30c9	-
aws-default-route-table	rtb-9e7a2aff5	-

The screenshot shows the 'Edit routes' dialog for the 'ap-south-1-public-route-table'. The table has one existing route:

Destination	Target	Status	Propagated
172.16.0.0/16	local	active	No

A new route is being added:

Destination	Target
0.0.0.0	igw-054a82ce1feb26815

Buttons at the bottom include 'Add route', 'Cancel', and 'Save routes'.

The screenshot shows the 'Create route table' dialog. It includes fields for 'Name tag' (set to 'ap-south-1-private-route-table') and 'VPC' (set to 'vpc-00c3b6efcf0ee9add'). Buttons at the bottom include 'Cancel' and 'Create'.

- Rename the Route Table that was created automatically for ap-south-1-vpc
- We are going to attach the Internet Gateway app-south-1-igw to this route table. Now this route table becomes public route table.
- Edit routes >>

- Add an internet route (0.0.0.0/0) and point to the app-south-1-igw

- Now add one more Route Table ap-south-1-private-route-table
- Do not associate Internet gateway to this route table so that this becomes always private.

- Let's try understand how one is Public and other one is Private with below table.

ap-south-1-public-route-table		ap-south-1-private-route-table	
Destination	Target	Destination	Target
172.16.0.0/16	Local	172.16.0.0/16	Local
0.0.0.0/0	igw-054a82ce1feb26815	-	-

Step 7: Associate Route Table with Subnets

- Select the ap-south-1-public-route-table >> Subnet Associations >> Edit subnet associations

- Associate ap-south-1-public-subnet-1a and ap-south-1-public-subnet-1b to the ap-south-1-public-route-table
- Now this particular subnet is actually a Public Subnet

- Similarly, associate ap-south-1-private-subnet-1a and ap-south-1-private-subnet-1b to the ap-south-1-private-route-table
- Now this particular subnet is actually a Private Subnet

-
- Let's try understand how the entire routing has been configured for the VPC using below table.

TYPE	SUBNET	CIDR	ROUTE TABLE	ROUTES	
				Destination	Target
Public subnet	ap-south-1-public-subnet-1a	172.16.11.0/24	ap-south-1-public-route-table	172.16.0.0/16	Local
	ap-south-1-public-subnet-1b	172.16.12.0/24		0.0.0.0/0	igw-054a82ce1feb26815
Private subnet	ap-south-1-private-subnet-1a	172.16.21.0/24	ap-south-1-private-route-table	172.16.0.0/16	Local
	ap-south-1-private-subnet-1b	172.16.22.0/24		-	-

Step 8: Launch EC2 Instances in the VPC

- Now launch an EC2 instance on your VPC with in Public Subnet, make sure you have a security group that allows access the EC2 from internet. Try to access the EC2 instance

The screenshot shows two side-by-side AWS EC2 instance creation interfaces.

Left Panel (Instance Creation):

- Step 3: Configure Instance Details**: Shows the configuration for launching an instance. The "Network" dropdown is set to "vpc-00c3b6fcf0ee9add | ap-south-1-vpc". The "Subnet" dropdown is set to "subnet-05bd2cc86c1e09ec5 | ap-south-1-public-subr". The "Auto-assign Public IP" dropdown is set to "Use subnet setting (Enable)". These three fields are highlighted with red boxes.
- Right Panel (Security Group Configuration):**
 - Step 6: Configure Security Group**: Shows creating a new security group named "public-windows-security-group". The "Type" dropdown is set to "RDP", "Protocol" to "TCP", and "Port Range" to "3389". The "Source" dropdown is set to "0.0.0.0/0". This row is highlighted with a red box.
 - Warning Message**: A warning message states: "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."

Bottom Panels:

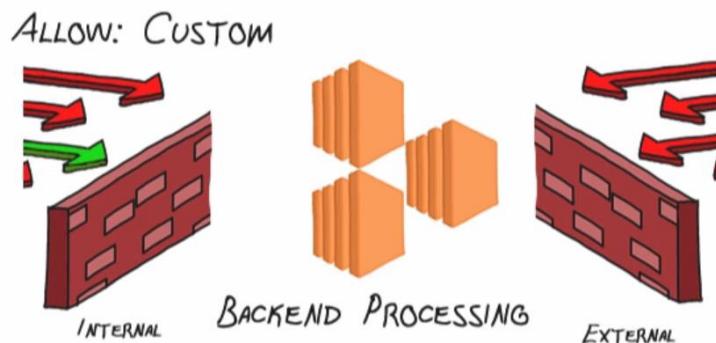
- EC2 Dashboard (Left):** Shows the instance status as "Running". The instance ID is "i-0a7a9785a55d872", type is "t2.micro", and availability zone is "ap-south-1a".
- Windows Taskbar (Right):** Shows the Windows Start menu and a Network Connection window for "Internet (Ethernet) - AWS PV Network Device #0". The window displays network connection details, including the public IP address "10.209.157.183" and private IP address "172.16.11.224".

- Now launch an EC2 instance on your VPC with in Private Subnet, you won't get any Public IP top access it.
- Even if you enable auto assign public IP, and allow RDP access to all network, still the instance won't be accessible since it is in private subnet (without an internet gateway).
- The only way to access the EC2 instance is via another EC2 that is in Public subnet and then RDP to private subnet EC2 via private IP address.

The composite screenshot illustrates the AWS EC2 instance creation process and its resulting configuration across four panels:

- Step 3: Configure Instance Details**: Shows the configuration of the instance type (t2.micro), network (vpc-00c3b6efcf0ee9add | ap-south-1-vpc), subnet (subnet-0f4483584b239c2f5c | ap-south-1-private-subnet), and security group (private-windows-security-group). The "Auto-assign Public IP" dropdown is set to "Disable".
- Step 6: Configure Security Group**: Shows the creation of a new security group named "private-windows-security-group". A rule is added to allow RDP traffic (TCP port 3389) from the private subnet (172.16.0.0/16).
- EC2 Dashboard - Instances**: Shows the newly launched instance "private-windows-server-2019" (Instance ID: i-0b35d6c62b7d36553) in the "running" state. It has a Private IP of 172.16.21.18 and a Private DNS of ip-172-16-21-18.ap-south-1.compute.internal. It is associated with the "private-windows-security-group".
- Remote Desktop Connection**: A screenshot of a Windows Remote Desktop session connected to the instance. The status bar shows the Public DNS of the instance (ip-172-16-21-18.ap-south-1.compute.internal) and the Private IP (172.16.21.18). The desktop environment is visible, showing the Windows Start menu and system tray.

Security Groups



- Virtual firewalls for EC2 instances applied at instance NIC level.
- Created under VPC and applied at EC2
- Block / allow inbound and outbound traffic for an EC2 instances.
- One Security Group can be assigned to multiple EC2 instances, but one EC2 will have only one security group.
- Some other services (RDS, Redshift) also uses Security Groups to protect network attack.
- Each of the EC2 instances are protected by Security Group
- It is only protecting the individual resources assigned to an EC2
- If 2 instances are communicating in same Security group, traffic from one E2 goes out to the security group and enters to the same security group, then other EC2.
- Default is Allow all outbound and Deny all inbound.
- Response to the Security Group accepted automatically, no need to specify inbound rule for accepting response traffic.

[LAB] Create or Select Security Group while Launching EC2

- You might have already seen, the way of assigning Security Group during the creation of EC2 instance.
- You can either create a new security group and define allow rules or select existing
- At this time we only add inbound rules, all outbound traffics allowed default.

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 your instance, add 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: launch-wizard-1

Description: launch-wizard-1 created 2020-07-04T12:51:46.045+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom	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.

Inbound rules for sg-03bc17a19529ccc88 (Selected security groups: sg-03bc17a19529ccc88)

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	0.0.0.0/0	

Cancel Previous Review and Launch

Feedback English (US) Privacy Policy Terms of Use

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 your instance, add 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 ID	Name	Description	Actions
sg-09559047168ed7b1a	default	default VPC security group	Copy to new
sg-055dcbb0d47c7168	private-windows-security-group	private-windows-security-group	Copy to new
sg-03bc17a19529ccc88	public-windows-security-group	public-windows-security-group	Copy to new

Review and Launch

Feedback English (US) Privacy Policy Terms of Use

[LAB] Modify Existing Security Group

- Modify inbound and outbound rules of Security group is available in EC2 Dashboard >> Security Groups

The screenshot shows the AWS EC2 Security Groups dashboard. On the left, a list of services like EC2, VPC, S3, etc., is visible. Under the 'NETWORK & SECURITY' section, 'Security Groups' is selected. In the main pane, a table lists four security groups. The first one, 'sg-03bc17a19529ccc88 - public-windows-security-group', is selected and expanded. It shows an 'Inbound rules' tab with a single rule: Type RDP, Protocol TCP, Port range 3389, Source 0.0.0.0/0. The 'Outbound rules' tab is also visible. On the right, another identical view of the same security group is shown, with a red arrow pointing to the 'Security group name' dropdown in the top table row.

- You can also create new security group from this dashboard.

The screenshot shows the 'Create security group' wizard. The first step, 'Basic details', has 'Security group name' set to 'MyWebServerGroup'. The second step, 'Inbound rules', shows a table with a single rule: Type Custom TCP, Protocol TCP, Port range 0, Source Custom, Destination 0.0.0.0/0. An 'Add rule' button is highlighted with a red arrow. The third step, 'Outbound rules', shows a similar table with a single rule: Type Custom TCP, Protocol TCP, Port range 0, Source Custom, Destination 0.0.0.0/0. An 'Add rule' button is also highlighted with a red arrow. The final step, 'Tags - optional', shows a note about tags and an 'Add new tag' button. At the bottom right is a large orange 'Create security group' button.



[LAB] Allow ICMP from Internet to a Windows EC2 Instance

- When you spin up an EC2 in public subnet, the default Security Group blocks the ICMP traffic to the instance.
- In this lab, we modify the existing security group to allow ICMP (ping) traffic.

The screenshot shows the AWS EC2 Instances dashboard. It lists two instances: 'private-windows-server-2019' and 'public-windows-server-2019'. Both instances are running and have their Public DNS and Private DNS details listed. The 'Security groups' section for the public instance is highlighted with a red box, showing it is associated with the 'public-windows-security-group'.

The screenshot shows the AWS Security Groups page for the 'sg-03bc17a19529cc88 - public-windows-security-group'. It displays the 'Details' section and the 'Inbound rules' tab, which contains a single RDP rule. A red arrow points to the 'Edit inbound rules' button.

The screenshot shows the 'Edit inbound rules' dialog for the security group. It displays two inbound rules: one for RDP (TCP port 3389) and another for ICMP (All ICMP - IPv4). A red box highlights the ICMP rule, and a red arrow points to the 'Add rule' button at the bottom.

- You can either select the Security group from the EC2 instance or navigate to Security Group from the EC2 dashboard.
- Select the Security Groups >> Inbound Rules >> Edit inbound rules >> Add Rule (Inbound rule 2 will be created)

The screenshot shows the 'Edit inbound rules' dialog on a Windows instance. It displays an 'Inbound rule 1' for RDP (TCP port 3389) and an 'Add rule' button. A red arrow points to the 'Add rule' button.

The screenshot shows the Windows Defender Firewall with Advanced Security interface. It displays the 'Inbound Rules' list, which includes various sharing and connection rules. A red box highlights the 'File and Printer Sharing (Echo Request - ICMPv4-In)' rule, and a red arrow points to the 'Save rules' button at the bottom right.

Since we are working on a Windows EC2, make sure either disable Windows native firewall or enable ICMP.

sg-03bc17a19529ccc88 - public-windows-security-group

[Details](#) | **Inbound rules** | [Outbound rules](#) | [Tags](#)

Inbound rules					Edit inbound rules
Type	Protocol	Port range	Source	Description - optional	
RDP	TCP	3389	0.0.0.0/0	-	

```
cmd Command Prompt
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\abvp>ping 13.126.185.9 ←
Pinging 13.126.185.9 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 13.126.185.9:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\abvp>
```

sg-03bc17a19529ccc88 - public-windows-security-group

[Details](#) | **Inbound rules** | [Outbound rules](#) | [Tags](#)

Inbound rules					Edit inbound rules
Type	Protocol	Port range	Source	Description - optional	
RDP	TCP	3389	0.0.0.0/0	-	
All ICMP - IPv4	ICMP	All	0.0.0.0/0	-	

```
cmd Command Prompt
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\abvp>ping 13.126.185.9 ←
Pinging 13.126.185.9 with 32 bytes of data:
Reply from 13.126.185.9: bytes=32 time=84ms TTL=112
Reply from 13.126.185.9: bytes=32 time=88ms TTL=112
Reply from 13.126.185.9: bytes=32 time=86ms TTL=112
Reply from 13.126.185.9: bytes=32 time=91ms TTL=112

Ping statistics for 13.126.185.9:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 84ms, Maximum = 91ms, Average = 87ms
C:\Users\abvp>
```

[LAB] Assign new Security Group to a Running EC2 Instance

- So far, we have seen modifying security group rules, what if we need to change the security group of a running EC2. That can be done by NIC level.
- Select the EC2 >> Network eth0 >> Interface ID >> Actions >> Change Security Groups

The screenshot shows the AWS EC2 service dashboard. On the left, under 'INSTANCES', there is a list of instances, one of which is selected and highlighted with a red arrow. In the center, a detailed view of a specific network interface (eth0) is shown. On the right, a context menu is open over the interface ID, with the 'Change Security Groups' option highlighted and also indicated by a red arrow.

The screenshot shows the 'Change Security Groups' dialog box. At the top, it displays the 'Network Interface' (eni-06a6f9cd6d4023bdd). Below that, a search bar shows 'sg-03bc17a19529ccc88'. A list of security groups is displayed, with the first item, 'sg-03bc17a19...', selected and highlighted with a red box. At the bottom of the dialog, there is a note saying '* Required' and two buttons: 'Cancel' and 'Save'.



Network Access Control Lists (NACLs)

- Virtual firewalls for subnets, rules defined in CIDR range
- Created inside VPC and associated to Subnets
- We can define NACLs for the ports range as well
- Rules applied in order (1st rule applied always)
- One NACL can be allied to many subnets but one subnet will have only one NACL
- By default all inbound and all outbound traffic allowed

[LAB] Modify NACLs to Block ICMP Traffic at the Subnet Level

- In the previous lab we have enabled ICMP traffic to our Windows EC2, you can ping the Windows machine from internet.
- Network ACLs allow all traffic (inbound and outbound) by default, in this lab, let's block the ICMP so the subnet (where our EC2 is running) using NACLs.
- You can access NACL from VPC dashboard >> Security >> Network ACLs or from EC2 >> Subnet ID >> NACL ID

The figure consists of three screenshots of the AWS console:

- Screenshot 1: EC2 Instances Overview**
Shows a list of EC2 instances. One instance, "public-windows-server-2019", is selected and highlighted with a red box. Its details pane shows it is associated with a subnet: "subnet-05bd2cc8c1e09ec5 (ap-south-1-public-subnet-1a)".
- Screenshot 2: VPC Subnets Overview**
Shows a list of subnets. One subnet, "subnet-05bd2cc8c1e09ec5 (ap-south-1-public-subnet-1a)", is selected and highlighted with a red box. This subnet is associated with a VPC: "vpc-00c3b6efcf0ee9add (ap-south-1-vpc)".
- Screenshot 3: Network ACL Management**
Shows the "Edit inbound rules" screen for a Network ACL named "ad-0882d1c755ca03a42".

Rule #	Type	Protocol	Port Range / ICMP Type	Source	Allow / Deny
100	All Traffic	All	All	0.0.0.0	ALLOW
1	All ICMP - IPv4	ICMP (1)	All	0.0.0.0	DENY

A red arrow points from the "Edit inbound rules" button in Screenshot 3 back to the "Inbound Rules" tab in Screenshot 2, indicating the flow of configuration.

[LAB] Create new NACL and Associate to Subnet

- If you are creating a new NACL, you need to associate to the required subnets.
- When you create a new NACL, all the traffic blocked by default

New VPC Experience
Tell us what you think

Lists New
Endpoints
Endpoint Services
NAT Gateways
Peering Connections

SECURITY
Network ACLs (selected)
Security Groups New

VIRTUAL PRIVATE NETWORK (VPN)
Customer Gateways
Virtual Private Gateways
Site-to-Site VPN Connections
Client VPN Endpoints

TRANSIT GATEWAYS
Transit Gateways
Transit Gateway

Feedback English (US) Privacy Policy Terms of Use

New VPC Experience
Tell us what you think

Managed Prefix Lists New
Endpoints
Endpoint Services
NAT Gateways
Peering Connections

SECURITY
Network ACLs (selected)
Security Groups New

VIRTUAL PRIVATE NETWORK (VPN)
Customer Gateways
Virtual Private Gateways
Site-to-Site VPN Connections
Client VPN Endpoints

TRANSIT GATEWAYS
Transit Gateways
Transit Gateway Attachments
Transit Gateway Route Tables

Feedback English (US) Privacy Policy Terms of Use

New VPC Experience
Tell us what you think

VIRTUAL PRIVATE CLOUD
Your VPCs
Subnets
Route Tables
Internet Gateways New
Egress Only Internet Gateways New
DHCP Options Sets New
Elastic IPs New
Managed Prefix Lists New
Endpoints
Endpoint Services
NAT Gateways
Peering Connections

SECURITY
Network ACLs (selected)
Security Groups New

VIRTUAL PRIVATE NETWORK (VPN)
Customer Gateways
Virtual Private Gateways
Site-to-Site VPN Connections
Client VPN Endpoints

TRANSIT GATEWAYS
Transit Gateways

Feedback English (US) Privacy Policy Terms of Use

Network ACLs > Create network ACL

Create network ACL

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

Name tag: app-south-1-vpc-icmp-block-acl

VPC: vpc-00c3b6efcf0ee9add

* Required Cancel Create

Network ACL: ad-0c85af129f70fe72

Edit inbound rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
1	All ICMP - IPv4	ICMP (1)	ALL	0.0.0.0/0	DENY

Add Rule

* Required Cancel Save

Network ACLs > Edit subnet associations

Edit subnet associations

Network ACL ID: ad-0c85af129f70fe72 (app-south-1-vpc-icmp-block-acl)

Subnets: subnet-05bd2cc06c1e09ec5

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-0483584b239c25c ap-south-1-private-subnet-1a	172.16.21.0/24	-
subnet-05bd2cc06c1e09ec5 ap-south-1-public-subnet-1a	172.16.11.0/24	-
subnet-0ac868796d03f643 ap-south-1-private-subnet-1b	172.16.22.0/24	-
subnet-09f347e337e174e9 ap-south-1-public-subnet-1b	172.16.12.0/24	-

* Required Cancel Edit

ASSIGNMENT 1

[LAB] Manually Host Sample HTML Website in AWS Windows Server 2019 EC2

- Let see how to host a sample website in a Windows EC2 and access via public IP and DNS name.
- Launch a new Windows 2019 EC2 instance in Public Subnet

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 the lower pricing, assign an access management role to the instance, and more.

Number of Instances: 1 Launch into Auto Scaling Group:

Purchasing option: Request Spot Instances

Network: vpc-0c3b6efcf0ee9add | ap-south-1-vpc
Subnet: subnet-05b02cc86c1e09ac5 | ap-south-1-public-sub
250 IP Addresses available

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open Create new Capacity Reservation

Domain join directory: No directory Create new directory

IAM role: None Create new IAM role

Shutdown behavior: Stop
Stop - Hibernate: Enable hibernation as an additional stop behavior

Enable termination protection: Protect against accidental termination

Monitoring: CloudWatch detailed monitoring
Additional charges apply

T2/T3 Unlimited: Enable
Additional charges may apply

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	windows-web-server-1	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Add Storage

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 your instance, add 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: web-servers-security-group

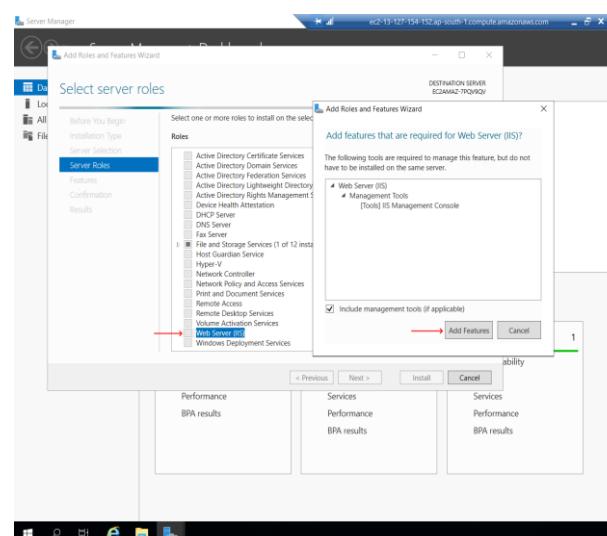
Description: web-servers-security-group

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	Custom <input style="border: none; background-color: transparent; font-size: small;" type="button" value="..."/>	e.g. SSH for Admin Desktop
(HTTP)	TCP	80	Custom <input style="border: none; background-color: transparent; font-size: small;" type="button" value="..."/>	e.g. SSH for Admin Desktop

Add Rule

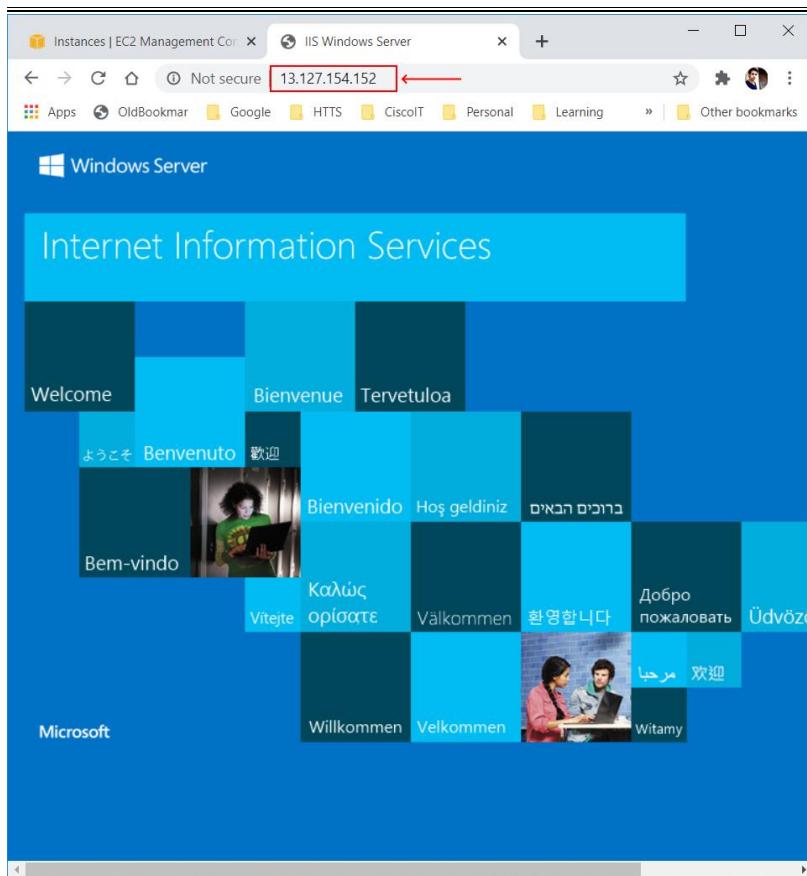
Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

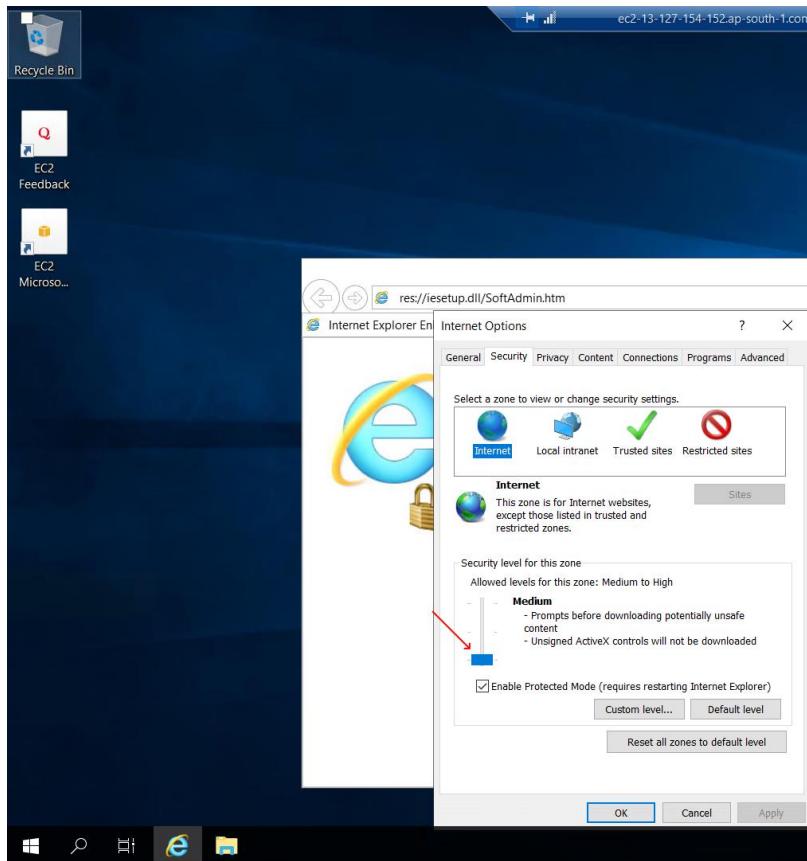


- Make sure your security group allows HTTP port 80 and RDP port 3389

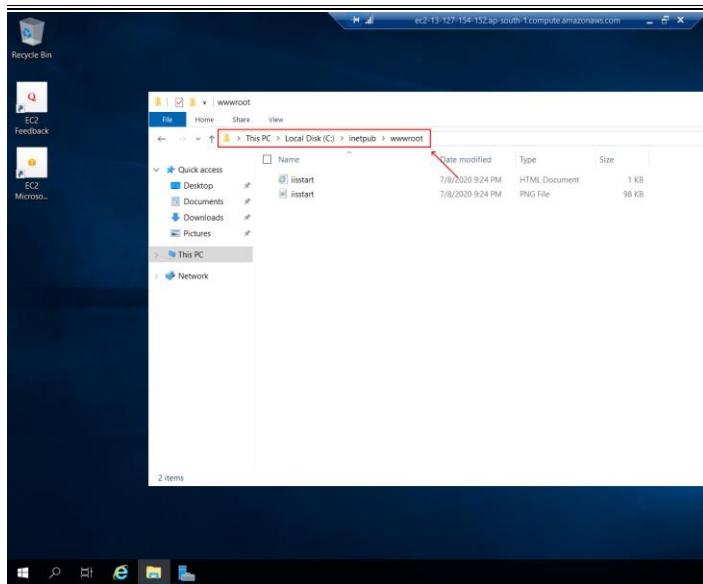
- Once the instance is ready, install Web Server IIS Role there.



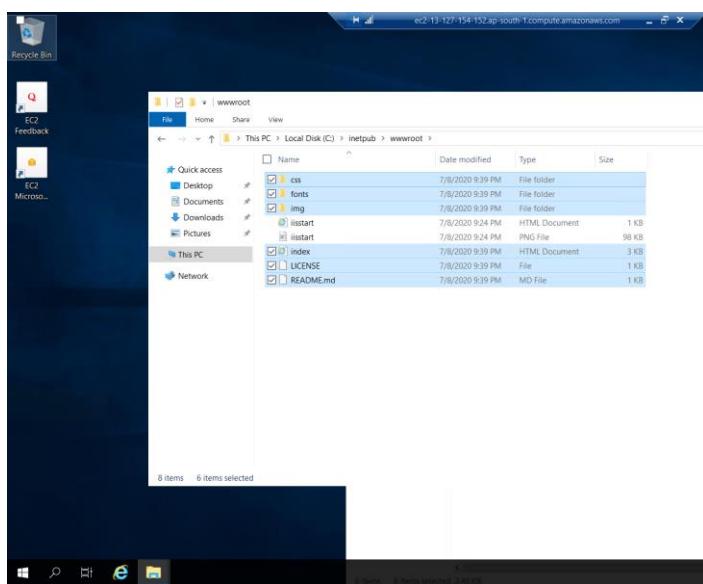
- Now try to access the public IP of the instance from a web browser, you will be able to see IIS default page.
- Now let's host our own website instead of the IIS default.



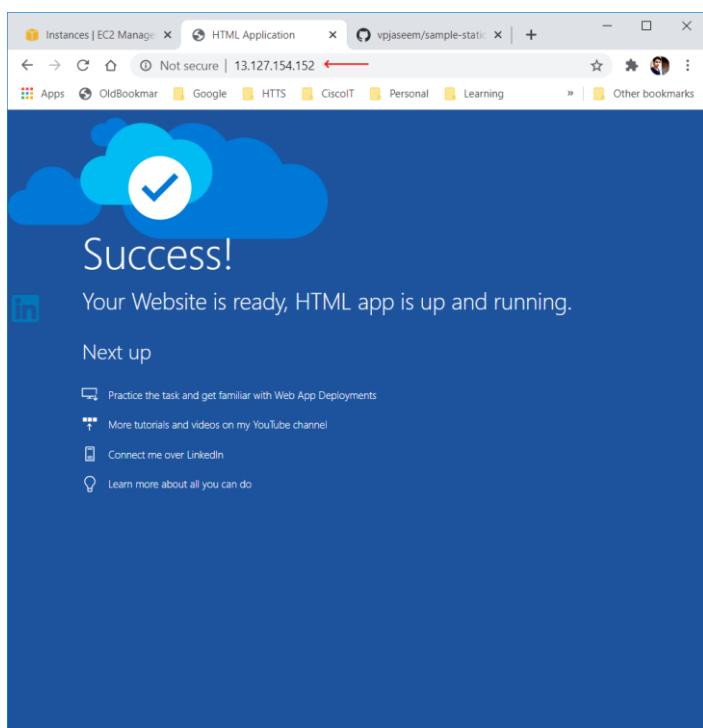
- On your Windows Server, turn off the IE Security Configurations.
- This is just to download a file, not a part of web server configuration



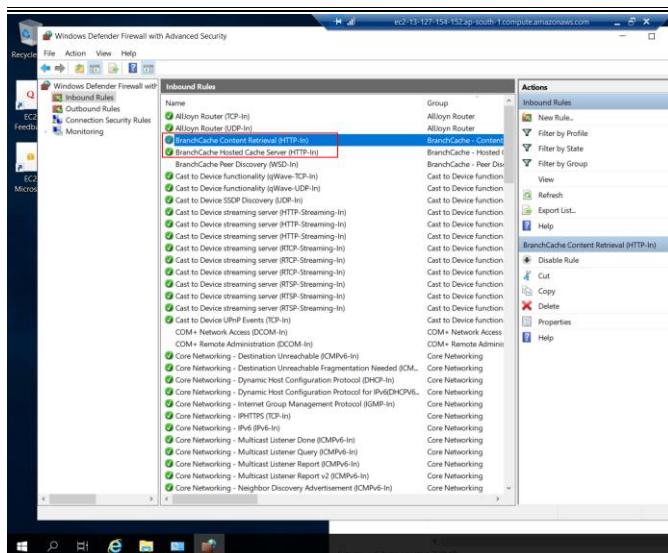
- Go to **C:\inetpub\wwwroot**
- I have created html website, those files can be downloaded from this link: [sample-html-web-app](#)



- Download those files and extract the content to **C:\inetpub\wwwroot**



- Now try to access again the public IP or AWS FQDN of the instance from a web browser, you will be able to see your new website.
- We can access it via public domain name as well.
- I have a domain Godaddy name **ajclassroom.co.in** where I have added CNAME (public FQDN) of Windows EC2



My Domains / Domain Settings

DNS Management

ajclassroom.co.in

Last updated 8/12/2020 7:58 AM

Type	Name	Value	TTL
NS	@	ns45.domaincontrol.com	1 Hour
NS	@	ns46.domaincontrol.com	1 Hour
SOA	@	Primary nameserver: ns4...	1 Hour
CNAME	www	ec2-15-207-85-8.ap-sou...	1 Hour

Records

ADD

HTML Application

Not secure | www.ajclassroom.co.in

AJ Labs

Success!

Your Website is ready, HTML app is up and running.

Next up

- Practice the task and get familiar with Web App Deployments
- More tutorials and videos on my YouTube channel
- Connect me on LinkedIn | GitHub
- This is an HTML Website

- While accessing via FQDN, please make sure you have HTTP port allowed in Windows Firewall
- Note: You need to allow on Windows Firewall as well as AWS Security Group

Advanced Features

Make sure your website is safe and always online. [Learn more](#)

DNSSEC
Create an unbreakable chain that stops hackers from hijacking your website and stealing your data.

Secondary DNS
Protect your website's uptime and availability from power outages and internet routing problems.

Add Premium DNS No, thanks

Forwarding

DOMAIN http://www.ajclassroom.co.in ADD

SUBDOMAIN Not set up

Manage Templates Export Zone File (Unix) Import Zone File DNSSEC Host names

Copyright © 1999 - 2020 GoDaddy Operating Company, LLC. All Rights Reserved. [Privacy Policy](#) Do not sell my personal information

[LAB] Get a free Domain Name

- Go to [CloudNS](#) and signup for a FREE account.

The screenshot shows the CloudNS website interface. On the left, there's a comparison table for different DNS plans:

FREE DNS	PREMIUM DNS	DDOS PROTECTED DNS	GEODNS
Free Forever	Starting from \$2.95/month	Starting from \$5.95/month	Starting from \$9.95/month
<ul style="list-style-type: none"> 4 Unicast DNS servers 1 DNS zone 50 DNS Records 500K DNS queries per month 1 Mail forward Dynamic DNS Web redirects 24/7 Live chat support 	<ul style="list-style-type: none"> All free and extended features +4 Anycast Premium Servers More zones, records, queries and mail forwards Unlimited queries per month DNS Failover & DNSSEC Free zones migration 24/7 Live chat support 1,000% Uptime SLA Starting from \$2.95/month 	<ul style="list-style-type: none"> All free and extended features +4 Anycast Protected Servers More zones, records and mail forwards Unlimited queries per month DNS Failover & DNSSEC Free zones migration 24/7 Live chat support 10,000% Uptime SLA Starting from \$5.95/month 	<ul style="list-style-type: none"> 4 Anycast DNS servers 1 DNS zone 50 DNS Records 500K DNS queries per month 1 Mail forward Dynamic DNS Web redirects 24/7 Live chat support Free zones migration 24/7 Live chat support 10,000% Uptime SLA Starting from \$9.95/month

Buttons at the bottom include "SIGN UP FOR FREE", "COMPARE PLANS", and "Online - Live Chat".

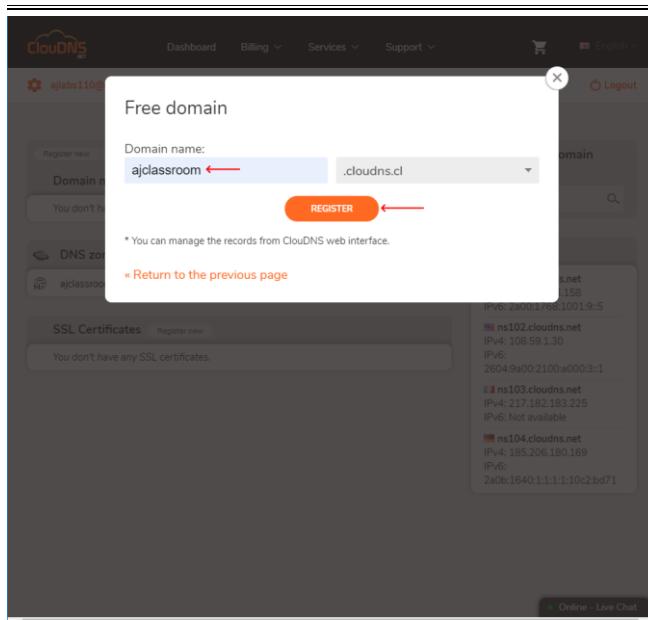
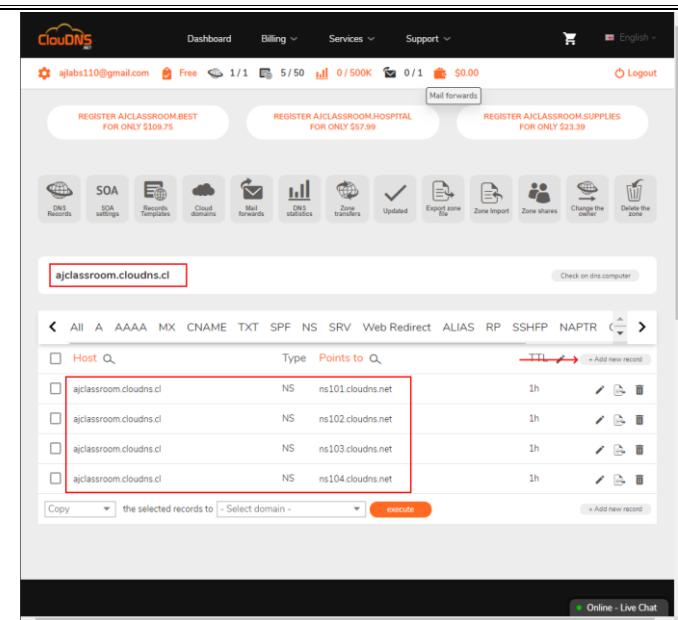
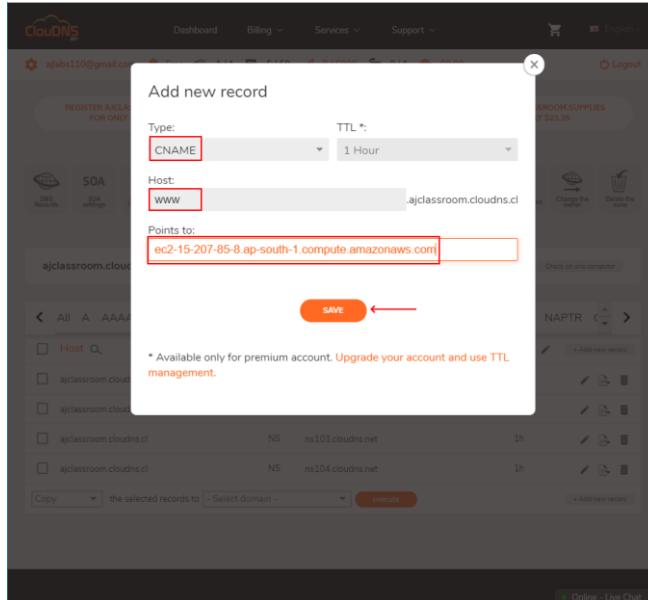
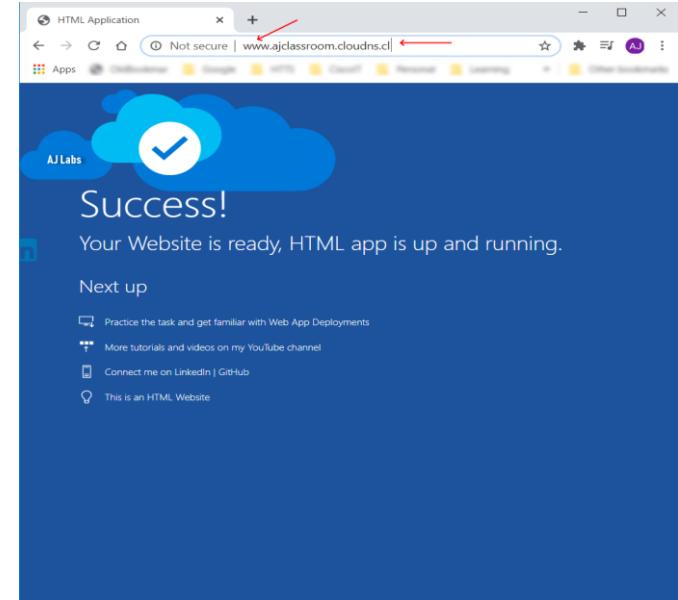
The right side shows a "Sign Up" modal window where "Abdul Jaseem" has entered their email (ajlabs110@gmail.com) and password. A checkbox for "I agree with the Terms of Service." is checked, and a "SIGN UP" button is present. Below the modal, a link "Click here, if you already have an account" is visible.

At the bottom of the page, there are "SIGN UP FOR FREE", "COMPARE PLANS", and "Online - Live Chat" buttons.

The dashboard shows the user's account information (ajlabs110@gmail.com) and various management sections:

- Domain names:** Shows no registered or transferred domain names.
- DNS zones:** Shows no registered DNS zones. A button "Add new" is available.
- SSL Certificates:** Shows no SSL certificates.
- DNS servers:** Lists four servers with their IP addresses and types (IPv4/IPv6).
- Master zone:** Primary DNS - Records can be managed only from our interface.
- Slave/Backup zone:** Secondary DNS - Records can be managed only at your master server.
- Master reverse zone:** Master IPv4 or IPv6 reverse zone.
- Slave reverse zone:** Slave IPv4 or IPv6 reverse zone.
- Primary ENUM zone:** Primary zone for E.164 numbers.
- Secondary ENUM zone:** Secondary zone for E.164 numbers.
- Parked zone:** Simple web page with contact form, title and description.
- Free zone:** DNS zone with free domain name. Records can be managed only from our interface.

Each section has a "DNS" icon and a "Edit" button. At the bottom, there are "Online - Live Chat" and "Logout" buttons.

[LAB] PowerShell Script to Host Sample HTML Website in AWS Windows Server 2019 EC2

- Whatever steps we have performed above can be automated using PowerShell scripts. In this lab, let's do the same process using PowerShell.
- Launch another new Windows 2019 EC2 instance in Public Subnet

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 pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: Request Spot Instances

Network: vpc-00c3b6efcf0ee9add | ap-south-1-vpc

Subnet: subnet-05bd2cc08c1e09ec5 | ap-south-1-public-sub
250 IP Addresses available

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory

IAM role: None

Shutdown behavior: Stop Enable hibernation as an additional stop behavior

Stop - Hibernation behavior: Protect against accidental termination

Enable termination protection: Monitoring: Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance Additional charges may apply when launching Dedicated instances.

T2/T3 Unlimited: Enable Additional charges may apply

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances (1)	Volumes (1)
Name	windows-web-server-1	<input type="checkbox"/>	<input type="checkbox"/>

Add another tag (Up to 50 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 your instance, add 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: web-servers-security-group

Description: web-servers-security-group

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	Custom <input type="button" value="..."/> 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom <input type="button" value="..."/> 0.0.0.0/0	e.g. SSH for Admin Desktop

- We need to follow 3 steps,
 - Add Windows Feature IIS
 - Download and place the sample web app files to wwwroot folder
 - Allow HTTP port 80 in Windows firewall

I have 3 PowerShell scripts available in the this link: [PowerShell Scripts](#)

Download those and execute one by one

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

vpjaseem / windows-startup-scripts

Code Issues Pull requests Actions Projects Wiki Security

Branch: master

vpjaseem committed 6935127 8 minutes ago 8 commits

- 1-add-iis-feature.ps1 Add files via upload
- 2-download-web-app-extract.ps1 Add files via upload
- 3-allow-http-inbound.ps1 Add files via upload
- README.md Create README.md
- git-installer.inf Add files via upload

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

vpjaseem / windows-startup-scripts

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security

master windows-startup-scripts / 1-add-iis-feature.ps1

vpjaseem Add files via upload Latest commit 393c0d0 2 days ago History

1 contributor

2 lines (2 sloc) 79 Bytes Raw Blame

```
1 Add-WindowsFeature -Name web-server
2 Install-WindowsFeature web-Nginx-Console
```

vpjaseem Add files via upload Latest commit 334ea8d 2 days ago History

1 contributor

28 lines (22 sloc) 968 Bytes Raw Blame

```
1 #Download the Web App file
2 cd C:\inetpub\wwwroot
3 $url = "https://github.com/vpjaseem/sample-static-web-app/archive/master.zip"
4 $output = "C:\inetpub\wwwroot\master.zip"
5 $start_time = Get-Date
6
7 Invoke-WebRequest -Uri $url -OutFile $output
8 Write-Output "Time taken: $((Get-Date).Subtract($start_time).Seconds) seconds"
9
10 #Unzip the Web App master zip file
11 Add-Type -AssemblyName System.IO.Compression.FileSystem
12 Function Unzip
13 {
14     param([string]$zipfile, [string]$outpath)
15
16     [System.IO.Compression.ZipFile]::ExtractToDirectory($zipfile, $outpath)
17 }
18
19 Unzip "C:\inetpub\wwwroot\master.zip" "C:\inetpub\wwwroot\master"
20
21 #Moving Web App contents to wwwroot of IIS
22 cd C:\inetpub\wwwroot\master\sample-static\web-app-master
23 Move-Item -Path '\*' -Destination C:\inetpub\wwwroot\
24 cd C:\inetpub\wwwroot\
25
26 #Removing unwanted folders
27 Remove-Item -LiteralPath "C:\inetpub\wwwroot\master\" -Force -Recurse
28 Remove-Item "C:\inetpub\wwwroot\master.zip"
```

- I have 3 PowerShell scripts available in the this link: [PowerShell Scripts](#)

Script 1:

- This is used to add Windows Feature IIS

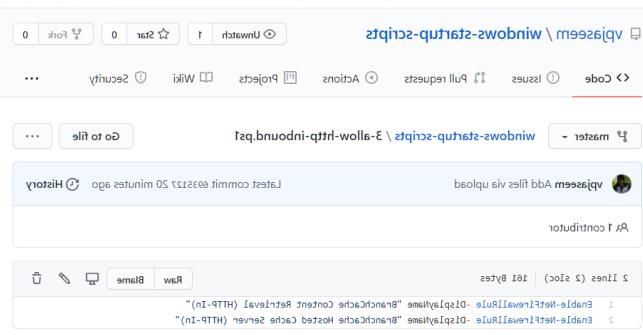
Script 2: Performs below tasks

- Download the Web App files
- Unzip the file content
- Place the contents to wwwroot folder of IIS
- Delete the unwanted files

Learn Git and GitHub Without Any Code!

Using the Hello World guide, you'll learn a lot about GitHub, write comments, and open a pull request.

[Read the guide](#)



Script 3:

- Enable HTTP port on Windows firewall

```
Add-WindowsFeature -Name web-server
Install-WindowsFeature web-Mgmt-Console
```

```
#Download the Web App file
cd C:\inetpub\wwwroot
$url = "https://github.com/vpjaseem/sample-static-html-web-app/archive/master.zip"
$output = "C:\inetpub\wwwroot\master.zip"
$start_time = Get-Date

Invoke-WebRequest -Uri $url -OutFile $output
Write-Output "Time taken: $((Get-Date).Subtract($start_time).Seconds) second(s)"

#Uzip the Web App master zip file
Add-Type -AssemblyName System.IO.Compression.FileSystem
function Unzip
{
    param([string]$zipfile, [string]$outpath)

    [System.IO.Compression.ZipFile]::ExtractToDirectory($zipfile, $outpath)
}

Unzip "C:\inetpub\wwwroot\master.zip" "C:\inetpub\wwwroot\master"

#Moving Web App contents to wwwroot of IIS
cd C:\inetpub\wwwroot\master\sample-static-html-web-app-master
Move-Item -Path .\* -Destination C:\inetpub\wwwroot\
cd C:\inetpub\wwwroot\

#Removing unwanted folders
Remove-Item -LiteralPath "C:\inetpub\wwwroot\master\" -Force -Recurse
```

```
Enable-NetFirewallRule -DisplayName "BranchCache Content Retrieval (HTTP-In)"
Enable-NetFirewallRule -DisplayName "BranchCache Hosted Cache Server (HTTP-In)"
```

Administrator: Windows PowerShell ISE

```

1 Add-WindowsFeature -Name web-server
2 Install-WindowsFeature web-Mgmt-Console
3
4 #Download the Web App file
5 cd C:\inetpub\wwwroot
6 $url = "https://github.com/vpjaseem/sample-static-web-app/archive/master.zip"
7 $output = "C:\inetpub\wwwroot\master.zip"
8 $start_time = Get-Date
9
10 Invoke-WebRequest -Uri $url -Outfile $output
11 Write-Output "Time taken: $(Get-Date).Subtract($start_time).Seconds) second(s)"
12
13 #Unzip the Web App master zip file
14 Add-Type -AssemblyName System.IO.Compression.FileSystem
15 function Unzip {
16     param([string]$zipfile, [string]$outpath)
17     [System.IO.Compression.ZipFile]::ExtractToDirectory($zipfile, $outpath)
18 }
19
20 Unzip "C:\inetpub\wwwroot\master.zip" "C:\inetpub\wwwroot\master"
21
22 #Moving Web App contents to wwwroot of IIS
23 cd C:\inetpub\wwwroot\master\sample-static-web-app-master
24 Move-Item -Path .\* -Destination C:\inetpub\wwwroot\
25
26 cd C:\inetpub\wwwroot\
27
28 #Removing unneeded folders
29
30

```

Collecting data...
6%

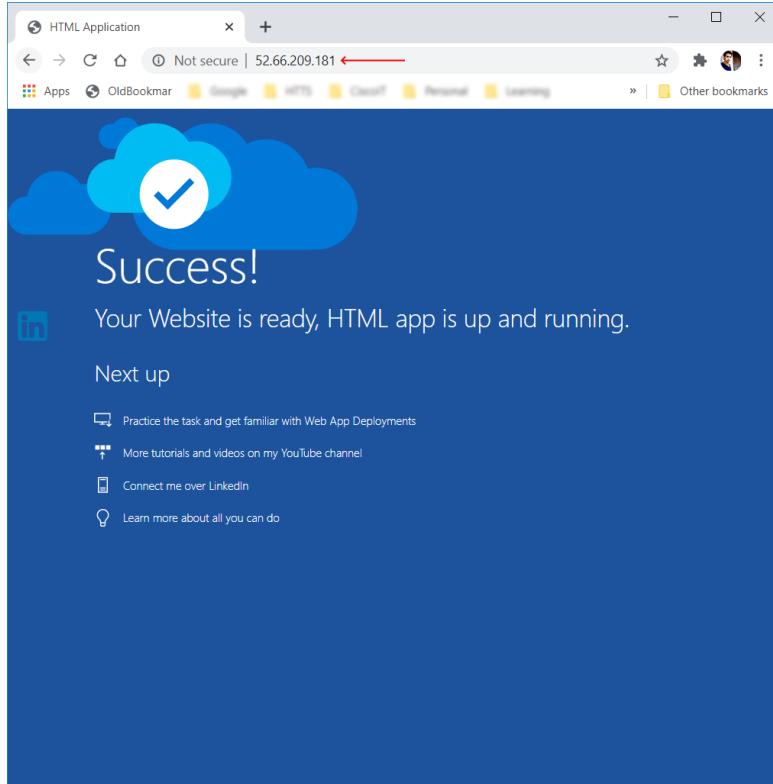
```

Remove-Item C:\inetpub\wwwroot\master\*.zip
Enable-NetFirewallRule -DisplayName "BranchCache Content Retrieval (HTTP-In)"
Enable-NetFirewallRule -DisplayName "BranchCache Hosted Cache Server (HTTP-In)"

```

Running script / selection. Press Ctrl+Break to stop. Press Ctrl+B to break into debugger.

- Run all the 3 scripts using Windows PowerShell (one by one or all at once)



- Your Website is ready!

[LAB] Hosted PowerShell Script to Host HTML Website in AWS Windows Server 2019 EC2

- We know that all the 2 scripts used for deploying simple website is available on GitHub
- Instead of copying and executing it, you can create a master PowerShell script that can call multiple other scripts hosted in any cloud (GitHub, GoogleDrive, Dropbox, AWS S3, etc.)

```
ielx ((New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/vpjaseem/windows-startup-scripts/master/1-add-iis-feature.ps1'))  
ielx ((New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/vpjaseem/windows-startup-scripts/master/2-download-web-app-extract.ps1'))  
ielx ((New-Object System.Net.WebClient).DownloadString('https://github.com/vpjaseem/windows-startup-scripts/raw/master/3-allow-http-inbound.ps1'))
```

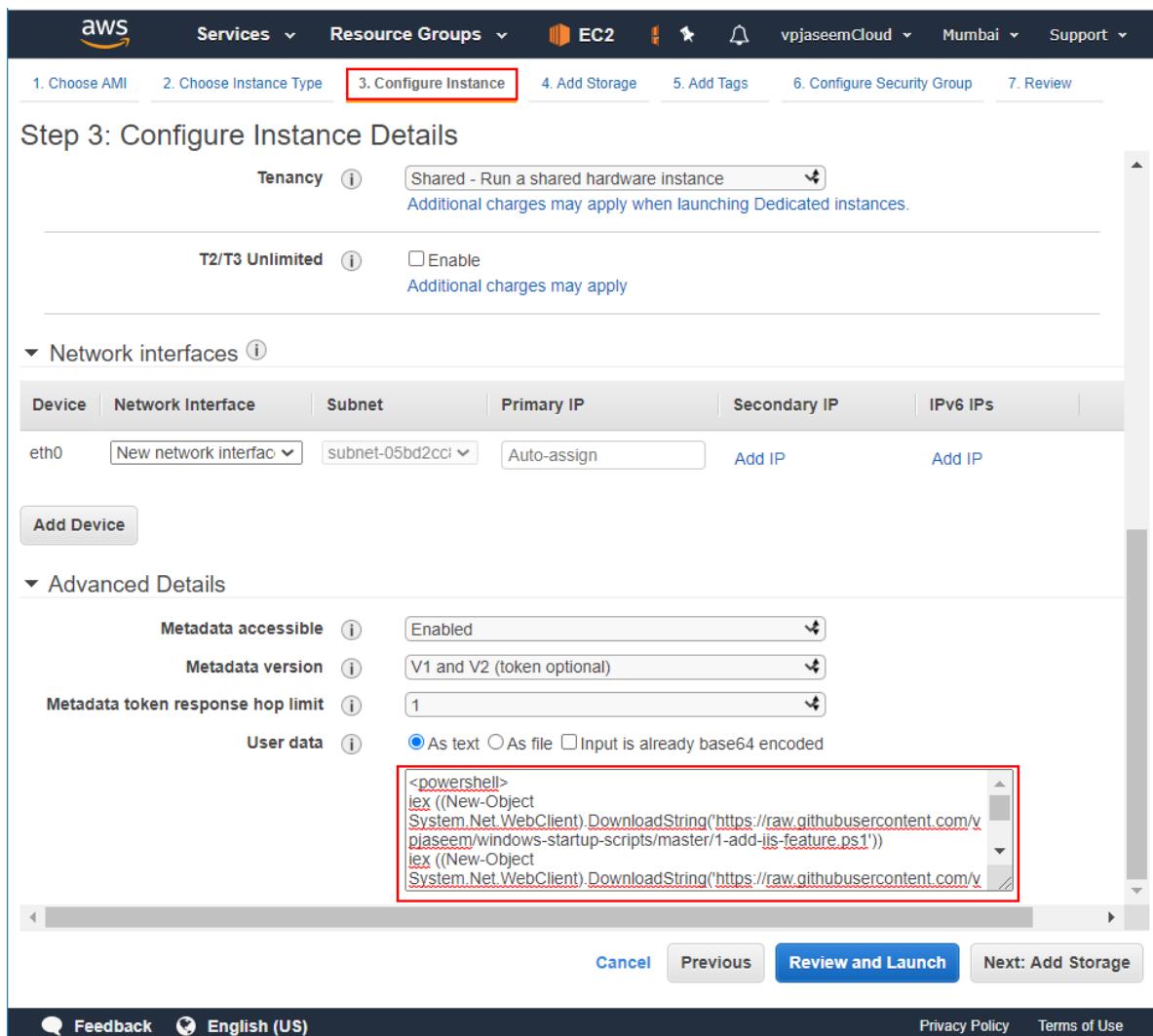
- In this case, all the 3 Scripts are hosted in GitHub cloud, using the above Script we are calling one by one.
- This method will make the script modification easy.
- Launch a new Windows Server 2019 EC2 instance in public subnet and allow port 80 on security group.
Then just run above script to get your website deployed.

[LAB] HTML Web Hosting Automation in AWS Windows Server 2019 EC2 Using User Data

- User Data is used to pass some startup scripts to EC2, when the EC2 spun up, it will execute those commands automatically.
- Our master PowerShell script can be supplied to a Windows Server 2019 EC2 instance via User Data, it will be executed automatically, and your web app deployed in minutes without you even logging in to windows.

```
<powershell>
iex ((New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/vpjaseem/windows-
startup-scripts/master/1-add-iis-feature.ps1'))
iex ((New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/vpjaseem/windows-
startup-scripts/master/2-download-web-app-extract.ps1'))
iex ((New-Object System.Net.WebClient).DownloadString('https://github.com/vpjaseem/windows-startup-
scripts/raw/master/3-allow-http-inbound.ps1'))
</powershell>
```

- You can pass any kind of PowerShell script in this way.
- Launch the EC2 like below User data, make sure you allow port 80 in security group.



ASSIGNMENT 2

[LAB] Manually Host Sample PHP Website in AWS Linux EC2

- Launch Amazon Linux AMI in public subnet and allow port 80 in security group

Step 1: Choose an Amazon Machine Image (AMI)

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-026e9920fd9d9473 (64-bit Arm) Select 64-bit (x86) 64-bit (Arm)
 SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type - ami-0bb01356763b34253 (64-bit Arm)
 Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
 Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-08706cb5f5822d09 Select 64-bit (x86)
 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.
 Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
 Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-03b6a287edcc0c1253 (64-bit Arm)
 Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type - ami-03b6a287edcc0c1253 (64-bit Arm) Select 64-bit (x86) 64-bit (Arm)
 Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
 Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
 Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-02d55cb47e83a959a0 (64-bit x86)
 Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type - ami-02d55cb47e83a959a0 (64-bit x86) Select 64-bit (x86) 64-bit (Arm)
 Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
 Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
 Amazon Linux 2 (HVM), SSD Volume Type - ami-073262d310b8e057 (64-bit x86)
 Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned. Select 64-bit (x86) 64-bit (Arm)
 Amazon Linux 2 (HVM), SSD Volume Type - ami-0a03ca9988cfe0c1 (64-bit Arm)

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more about tagging your Amazon EC2 resources.](#)

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	linux-web-server-1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- SSH to the Linux and run below commands,
 - Install Apache HTTPD Web Server
 - Clone the web app files from GitHub to home/var/www/html/ directory

```
sudo yum update -y
sudo yum install httpd -y
sudo yum install php -y
sudo service httpd start
sudo chkconfig httpd on
sudo yum install git -y

cd /$home/var/www/html/
sudo git clone https://github.com/vpjaseem/sample-static-php-web-app.git
sudo cp -r /var/www/html/sample-static-php-web-app/* /var/www/html/
sudo service httpd restart
```

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 the lower pricing, assign an access management role to the instance, and more.

Number of Instances 1 Launch into Auto Scaling Group

Purchasing option Request Spot instances

Network vpc-003b6efc0ee9add1 ap-south-1-vpc Create new VPC
 Subnet subnet-055b02cc86c1e09e4c5 ap-south-1-public-sub Create new subnet
 Auto-assign Public IP Use subnet setting (Enable)

Placement group Add instance to placement group Create new Capacity Reservation

IAM role None Create new IAM role

Shutdown behavior Stop Enable hibernation as an additional stop behavior
 Stop - Hibernate behavior Protect against accidental termination
 Enable termination protection Enable CloudWatch detailed monitoring Additional charges apply

Monitoring Shared - Run a shared hardware instance Additional charges may apply when launching Dedicated instances

T2/T3 Unlimited Enable Additional charges may apply

File systems Add file system Create new file system

Cancel Previous Review and Launch Next: Add Storage

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

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 your instance, add 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:				
<input type="radio"/> Create a new security group	<input checked="" type="radio"/> Select an existing security group			
Security Group ID	Name	Description	Actions	
sg-0df0fc4796752b96	aws-rds-mysql-security-group	aws-rds-mysql-security-group	<input type="checkbox"/> Copy to new	
sg-09558047165e07b1a	default	default VPC security group	<input type="checkbox"/> Copy to new	
sg-055dc0bf047c7188	private-windows-security-group	private-windows-security-group	<input type="checkbox"/> Copy to new	
sg-03bc17a19529cc88	public-windows-security-group	public-windows-security-group	<input type="checkbox"/> Copy to new	
sg-0348e96684b59cf2a	web-servers-security-group	web-servers-security-group	<input type="checkbox"/> Copy to new	

Inbound rules for sg-0348e96684b59cf2a (Selected security groups: sg-0348e96684b59cf2a)

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
SSH	TCP	22	0.0.0.0/0	<input checked="" type="checkbox"/>
RDP	TCP	3389	0.0.0.0/0	

Cancel Previous Review and Launch

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

[LAB] Bash Script to Host Sample PHP Website in AWS Linux EC2

- The same above Linux commands can be grouped into bash script and execute once to deploy the web server.
- Save this as <FILE-NAME>.sh (*web-deploy.sh*)

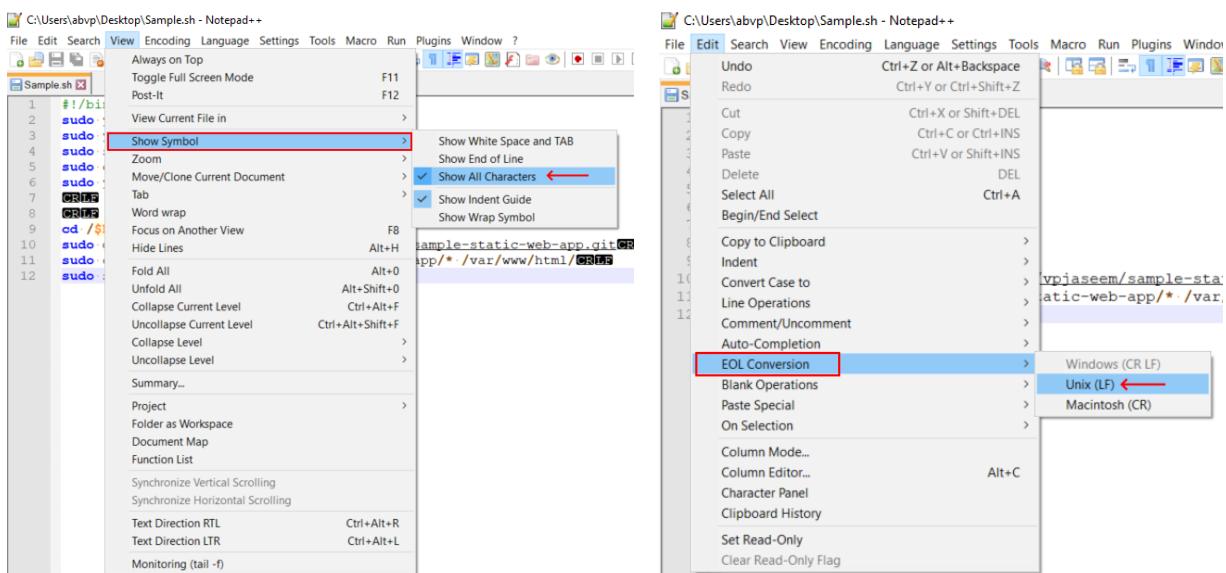
```
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo service httpd start
sudo chkconfig httpd on
sudo yum install git -y

cd /$home/var/www/html/
sudo git clone https://github.com/vpjaseem/sample-static-web-app.git
sudo cp -r /var/www/html/sample-static-web-app/* /var/www/html/
sudo service httpd restart
```

- Grant permission using *chmod +x web-deploy.sh* then execute *./web-deploy.sh* or *sudo ./web-deploy.sh*
- The same script can also be passed as User Data to automate the process at the time of EC2 instance launching.

Note:

If you are using Windows Text editors for bash script, the new line will be CR+LF (Carriage Return + Line Feed) but Linux understands LF only.



[LAB] Hosted Bash Script to Host Sample PHP Website in AWS Linux EC2

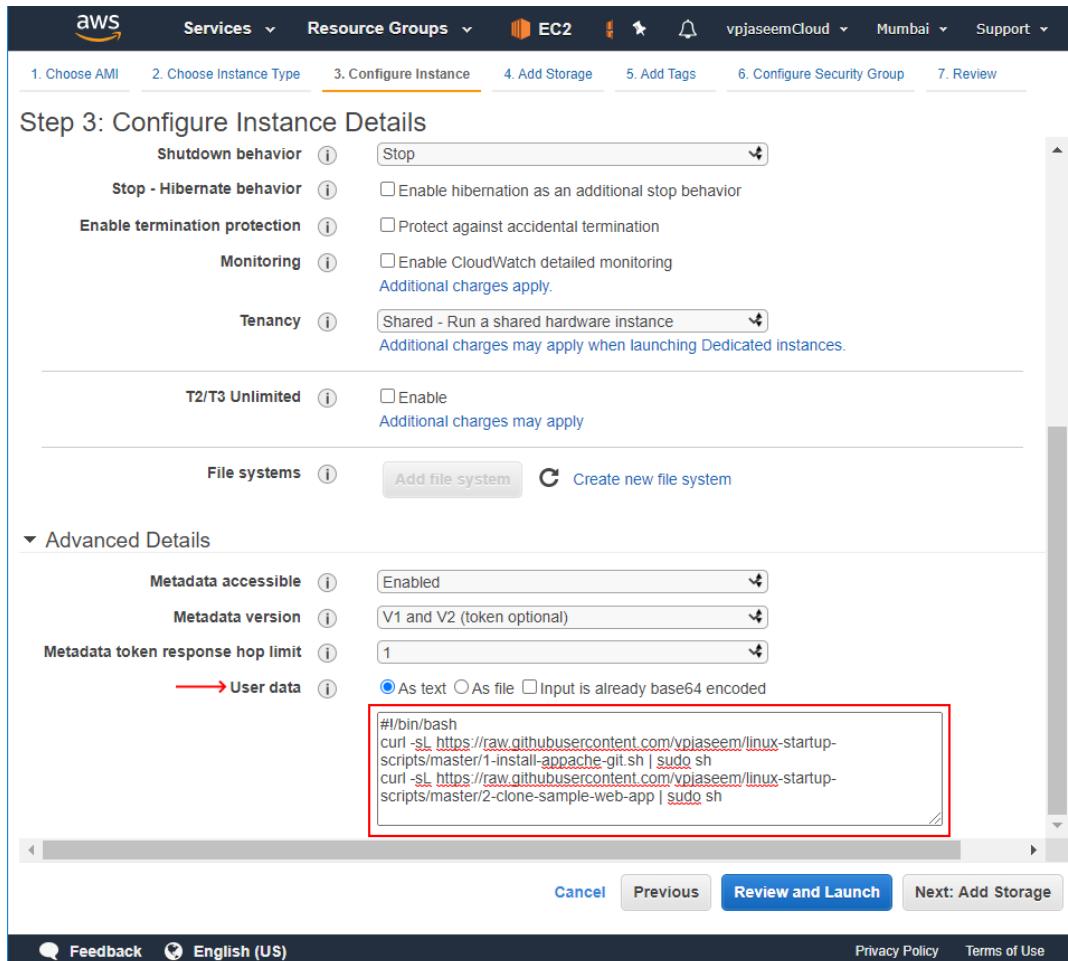
- Alternately, we can host the same bas script in GitHub (or GoogleDrive, DropBox, S3, etc.) and use another bash script to call all these.
- Below is the master bash script that calls the other 2 bash scripts to install httpd, PHP and clone web app files.

```
#!/bin/bash
curl -sL https://raw.githubusercontent.com/vpjaseem/linux-startup-scripts/master/1-install-apache-php-git.sh | sudo sh
curl -sL https://raw.githubusercontent.com/vpjaseem/linux-startup-scripts/master/3-clone-sample-php-web-app.sh | sudo sh
```

[LAB] PHP Web Hosting Automation in AWS Linux EC2 Using User Data

- Above Script can be passed to User Data so that the complete process of Web app deployment will be automated end to end.

```
#!/bin/bash
curl -sL https://raw.githubusercontent.com/vpjaseem/linux-startup-scripts/master/1-install-apache-php-git.sh | sudo sh
curl -sL https://raw.githubusercontent.com/vpjaseem/linux-startup-scripts/master/3-clone-sample-php-web-app.sh | sudo sh
```



- You can put this all to a Launch Template and deploy web app in 1 click.

EC2 Launch Template

- Launch templates used to automate the launch configurations
- We create a template by pre-configuring all the necessary steps required for a particular EC2 instance

The screenshot displays the AWS EC2 Launch Templates interface across five panels:

- Panel 1: Launch Templates List**
Shows a list of existing launch templates. A red arrow points to the "Create launch template" button.
- Panel 2: Create launch template - Step 1**
Shows fields for "Launch template name" (set to "windows-web-server") and "Template version description" (also set to "windows-web-server"). A red arrow points to the "Launch template name" field.
- Panel 3: Create launch template - Step 2**
Shows "Network settings" (Virtual Private Cloud selected) and "Security groups" (selected group: "web-servers-security-group sg-0348e96684b59cf2a"). A red arrow points to the "Virtual Private Cloud (VPC)" radio button.
- Panel 4: Create launch template - Step 3**
Shows "Storage (volumes)" (Volume 1 (AMI Root) selected) and "Resource tags". A red arrow points to the "Volume 1 (AMI Root)" selection.
- Panel 5: Create launch template - Step 4**
Shows "Actions" (highlighted with a red box) and "Launch instance from template". A red arrow points to the "Actions" button.

In the bottom right corner of each panel, there are "Privacy Policy" and "Terms of Use" links.

Network settings

Networking platform [Info](#)

- Virtual Private Cloud (VPC)** Launch into a virtual network in your own logically isolated area within the AWS Cloud
- EC2-Classic** Launch into a single flat network that you share with other customers

Subnet [Info](#)

subnet-05bd2cc86c1e09ec5 VPC: vpc-00c3befcf0ee9add Owner: 618927232701 Availability zone: ap-south-1a IP addresses available: 247

[Create new subnet](#)

Security groups [Info](#)

Select security groups

web-servers-security-group sg-0348e96684b59cf2a X

VPC: vpc-00c3befcf0ee9add

Storage (volumes) [Info](#)

Volume 1 (AMI Root) (30 GiB, EBS, General purpose SSD (gp2))

[Add new volume](#)

Resource tags [Info](#)

Currently no tags are specified and therefore the instance will launch with the default tag settings. Edit your tags if you would like to override the default settings.

[Add tag](#)
50 remaining (Up to 50 tags maximum)

Network interfaces [Info](#)

Currently no network interface details are specified and therefore the instance will launch with the template default network interface settings.

[Add network interface](#)

Advanced details [Info](#)

[Cancel](#) **Launch instance from template**



AWS Simple Storage Service (S3)

What is AWS Simple Storage Service (S3)



- Amazon Simple Storage Service is storage for the Internet, A cloud storage (like Google Drive, Dropbox, etc.)
- Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.
- To upload your data to Amazon S3, you must first create an S3 bucket in one of the AWS Regions. Cannot be nested (1 bucket inside another bucket)
- We can create 100 buckets, each bucket can store unlimited data
- Buckets are the top-level containers for data, bucket name has to be globally unique.
- It's an object-based storage, whatever file we upload to S3, we get a web link to access it via HTTP(s). Individual file size between 0 and 5TB
- Also available via REST or SOAP
- Built for unstructured binary data (basically files, large files), it is not a database.
- Objects can be versioned (same file upload twice, AWS will keep both)
- To access any object, bucket+object+version(optional) –
<https://bucket.s3.amazonaws.com/objectname.mp4>
- Highly scalable, Durable and Unlimited.
- Secure with client side or AWS Server side encryption
- Object and bucket level logs
- We can store Application data, photos, videos, documents, AMI, a complete website in S3
- Cost of S3 is relatively cheap, pay for only what you use, uploading data to S3 is free, download is chargeable (hourly charge)
- Access Control for the Bucket and Object to secure S3 (ACL - Access Control List: User to Permission mapping, IAM - Identity and Access Management policies)
- Bucket and Object policies can be defined with Access Policy Language (JSON Structure)
- Encrypt at REST with AES-256 or encrypt before sending.

AWS Reference: [What is Amazon S3?](#)

S3 Object Versioning

- Allows you to retain one or more versions of individual object (same file upload after some edits, S3 will keep latest as well as old)
- Versioning is enabled at the bucket level
- Existing objects have 'Null' version, new objects assigned with new version ID
- We can configure life cycle policy to automatically delete unwanted versions
- <https://bucket.s3.amazonaws.com/objectname.mp4?versionId=XXXXXXXX>

Storage Classes

Each object in Amazon S3 has a storage class associated with it. For example, if you list the objects in an S3 bucket, the console shows the storage class for all the objects in the list.

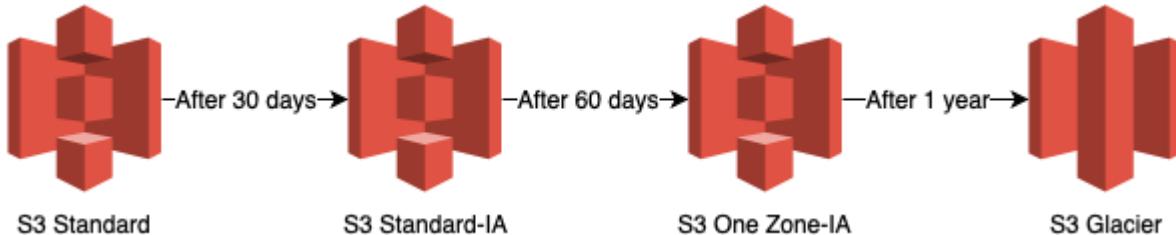
Viewing 1 to 3			
<input type="checkbox"/> Name ↑	Last modified ↑	Size ↑	Storage class ↑
<input type="checkbox"/> notice.pdf	Jul 13, 2016 7:19:13 PM GMT-0700	175.5 KB	Standard
<input type="checkbox"/> screen-shot.png	Apr 2, 2018 6:47:22 PM GMT-0700	109.8 KB	Standard-IA
<input type="checkbox"/> screen-shot3.png	Apr 2, 2018 7:08:32 PM GMT-0700	109.8 KB	Standard-IA

Class	Features
S3 Standard:	<ul style="list-style-type: none"> Frequently accessed data 99.99999999% durability 99.99% availability, >= 3 AZs
S3 Standard-Infreq Access (IA)	<ul style="list-style-type: none"> Long-lived, infrequently accessed data 99.99999999% durability 99.99% availability, >= 3 AZs 30 days minimum storage duration 128 KB minimum billable object size Per GB retrieval fees apply.
S3 Intelligent-Tiering	<ul style="list-style-type: none"> Long-lived data with changing or unknown access patterns 99.99999999% durability 99.99% availability, >= 3 AZs 30 days minimum storage duration Monitoring and automation fees per object apply. No retrieval fees.
S3 One Zone-IA	<ul style="list-style-type: none"> Long-lived, infrequently accessed, non-critical data 99.99999999% durability 99.5% availability, 1 AZs 30 days minimum storage duration 128 KB minimum billable object size Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.

S3 Glacier	<ul style="list-style-type: none"> • Long-term data archiving with retrieval times ranging from minutes to hours • 99.999999999% durability • 99.99% availability, • >=3 AZs • 90 days minimum storage duration • 40 KB minimum billable object size • Per GB retrieval fees apply. • Glacier basically a different entity, to access data from Glacier, you need to restore the data to S3 and then use it
S3 Glacier Deep Archive	<ul style="list-style-type: none"> • Archiving rarely accessed data with a default retrieval time of 12 hours • 99.999999999% durability • 99.99% availability, • >=3 AZs • 180 days minimum storage duration • 40 KB minimum billable object size • Per GB retrieval fees apply. You must first restore archived objects before you can access them.
RRS – Reduce Redundancy (Not recommended)	<ul style="list-style-type: none"> • Frequently accessed, non-critical data • 99.99% durability • 99.99% availability, • >=3 AZs

AWS Reference: [Amazon S3 Storage Classes](#)

S3 Life Cycle Management



- Versioning and storage classes are tied together using Life Cycle Management.
- Allows us to specify rules / policies to automatically manage the life cycle of object
- It happens in Object level or Bucket level
- An S3 Lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects.
- Transition actions**—Define when objects transition to another storage class. For example, you might choose to transition objects to the S3 Standard-IA storage class 30 days after you created them, or archive objects to the S3 Glacier storage class one year after creating them.
- Expiration actions**—Define when objects expire. Amazon S3 deletes expired objects on your behalf.

```

<LifecycleConfiguration>
  <Rule>
    <ID>sample-rule</ID>
    <Prefix>documents/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>

```

[LAB] AWS Simple Storage Service S3 Configuration

Now let's create a bucket in S3 and upload some files

The screenshot shows the AWS Services menu. A red arrow points from the 'Services' dropdown to the 'S3' service icon. The 'Storage' section is expanded, and a red arrow points to the 'S3' icon.

- History
- S3
- Console Home
- Billing
- EC2
- EC2 Image Builder
- VPC
- Compute
 - EC2
 - Lightsail
 - Lambda
 - Batch
 - Elastic Beanstalk
 - Serverless Application Repository
 - AWS Outposts
 - EC2 Image Builder
- Storage
 - S3**
 - EFS
 - FSx
 - S3 Glacier
 - Storage Gateway
 - AWS Backup
- Database
 - RDS
- Blockchain
 - Amazon Managed Blockchain
- Satellite
 - Ground Station
- Quantum Technologies
 - Amazon Braket
- Management & Governance
 - AWS Organizations
 - CloudWatch
 - AWS Auto Scaling
 - CloudFormation
 - CloudTrail
 - Config
 - OpsWorks
 - Service Catalog
 - Systems Manager

- In AWS Console >> Services >> S3

The screenshot shows the Amazon S3 buckets page. A red arrow points to the '+ Create bucket' button.

- Amazon S3
- Buckets
- Batch operations
- Access analyzer for S3
- Block public access (account settings)
- Feature spotlight

>> Create Bucket

The screenshot shows the 'Create bucket' wizard, step 1: Name and region. A red arrow points to the 'Bucket name' field, which contains 'my-cloud-bucket01'. Another red arrow points to the 'Region' dropdown, which is set to 'Asia Pacific (Singapore)'. A red arrow also points to the 'Create' button at the bottom left.

- ① Name and region
- ② Configure options
- ③ Set permissions
- ④ Review

Create bucket

Name and region

Bucket name

Region

Copy settings from an existing bucket

You have no buckets

>> Bucket Name (This has to be unique) >>

Region (Select) >> Create

Note: We can do many customizations here (Config options, permissions, etc.), but we will deal with those later.

The screenshot shows the Amazon S3 buckets page. A red arrow points to the 'my-cloud-bucket01' bucket entry in the list.

- Amazon S3
- Buckets
- Batch operations
- Access analyzer for S3
- Block public access (account settings)
- Feature spotlight

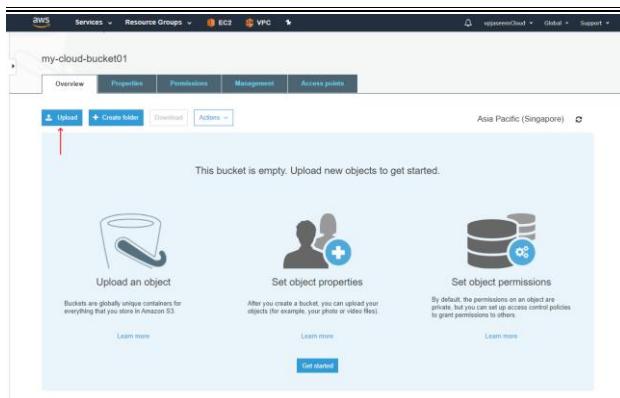
My buckets

Search for buckets

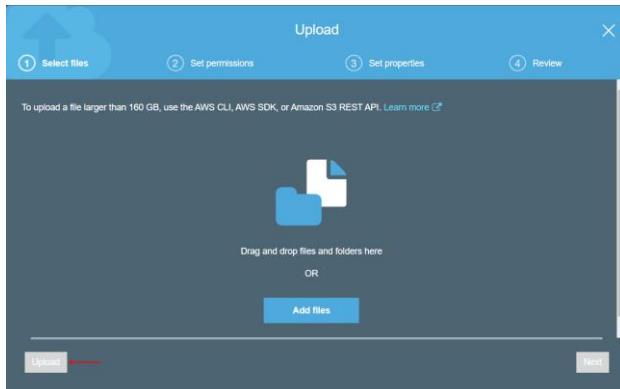
+ Create bucket

Bucket	Last modified	Size	Type
my-cloud-bucket01	Just now	0 B	Bucket

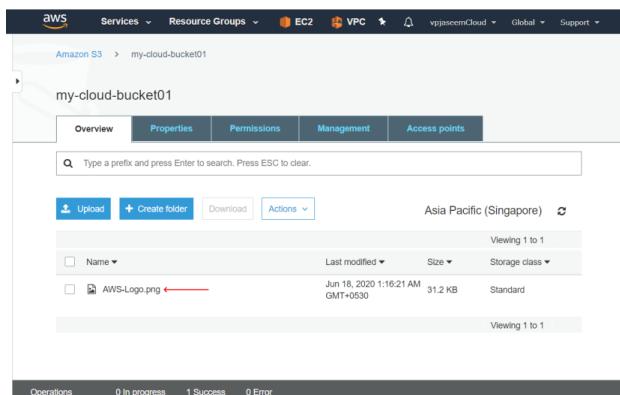
You can see the bucket that you created. Go ahead and open it.



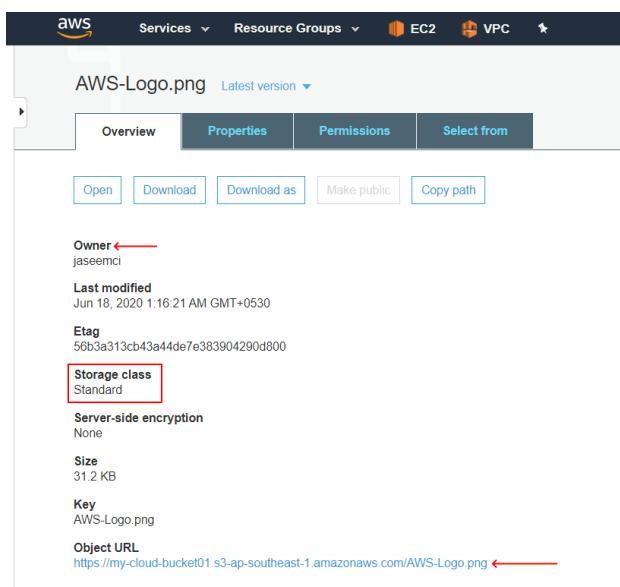
>> Upload button to upload new files (This is just like any other cloud storage e.g. Google Drive)



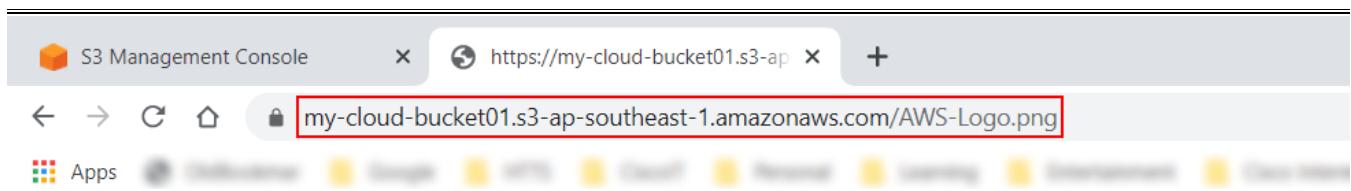
While uploading the file, you can do much more customization (Storage Class, Permissions, Properties, etc.), but we will deal with those later.



I have uploaded a file called 'AWS-Logo.png'



We can see the Storage Class, Public URL to access the object here.
You can only use 'Open' button to access the file.
We don't get access to Object URL until you make it as public.



The screenshot shows a browser window with the title "S3 Management Console". The address bar contains the URL "https://my-cloud-bucket01.s3-ap-southeast-1.amazonaws.com/AWS-Logo.png". A red box highlights this URL. Below the address bar, there's a navigation bar with icons for back, forward, home, and search. The main content area displays an XML error response:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>943FB5D17FEC2C13</RequestId>
  <HostId>k0g06+wcq+1o+7PUqVk1QTodcje4CYWNh/5qtgzb3empU40bhKA1qgSsMLxJ/alJIlwD32CYmSk=</HostId>
</Error>
```

I tried to access the Object URL and above is the error that I got. This is common. How to get anonymous access to the Object URL? Will see in next topic.

[LAB] S3 Bucket and Object Permissions Method 1

If we directly try to access an object URL, we may get the ‘Access Denied’ error. Well, how to set permissions and public access to S3 object?

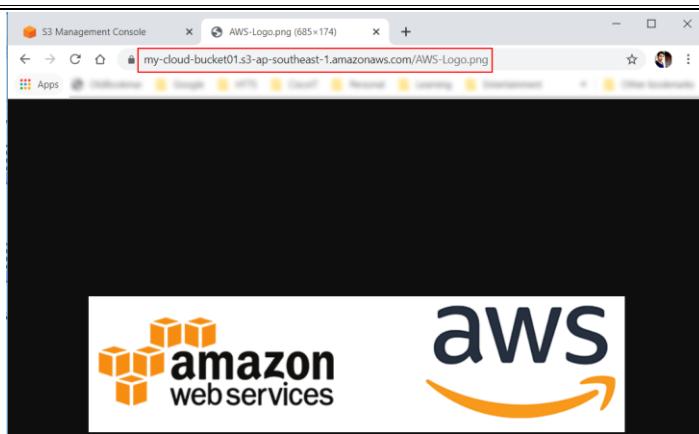
- Go to Bucket >> Permissions >> Block Public Access >> Edit >> Uncheck Block *all* public access
- Now we can enable public access to the objects in the bucket

[LAB] S3 Bucket and Object Permissions Method 2

The screenshot shows the AWS Management Console with the S3 service selected. In the left sidebar, under 'Amazon S3', the 'Buckets' link is highlighted with a red arrow. In the main content area, there's a search bar and a 'Create bucket' button. Below these, a list of buckets includes 'my-cloud-bucket01', which is also highlighted with a red arrow. To the right, a modal window titled 'Edit block public access settings for selected buckets' is open. It displays a single checkbox labeled 'Block all public access'. This checkbox is checked, and a red box highlights it. Below the checkbox, there's explanatory text and three other optional settings. At the bottom of the modal are 'Cancel' and 'Save' buttons, with a red arrow pointing to the 'Save' button.

The screenshot shows the 'my-cloud-bucket01' bucket details page. The 'Properties' tab is active. A file named 'AWS-Logo.png' is listed in the file list. A red arrow points from the file name to the file thumbnail. The file details show it was last modified on Jun 18, 2020 at 1:16:21 AM GMT+0530, has a size of 31.2 KB, and is stored in the Standard storage class. The region is set to Asia Pacific (Singapore).

The screenshot shows the object details for 'AWS-Logo.png'. The 'Properties' tab is active. The object metadata includes: Owner (jaseemci), Last modified (Jun 18, 2020 1:16:21 AM GMT+0530), Etag (56b3a313cb43a44de7e383904290d800), Storage class (Standard), Server-side encryption (None), Size (31.2 KB), Key (AWS-Logo.png), and Object URL (<https://my-cloud-bucket01.s3-ap-southeast-1.amazonaws.com/AWS-Logo.png>). A red arrow points from the 'Make public' button in the top navigation bar to the 'Permissions' tab.



[LAB] Revoke S3 Object Permission

The screenshot shows the AWS S3 console with the path: Amazon S3 > my-cloud-bucket01 > AWS-Logo.png. The 'Permissions' tab is selected. A modal window titled 'Everyone' is displayed, containing the following content:

This object has public access
Everyone has access to one or all of the following: read this object, read and write permissions.

Access to the object

- Read object

Access to this object's ACL

- Read object permissions
- Write object permissions

Operations: 0 In progress, 6 Success, 0 Error

Feedback, **English (US)**, © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. **Privacy Policy**, **Terms of Use**

Group	Read object	Read
Everyone	Yes	-

>> Select the Object >> Permissions >> Everyone >> Uncheck the Read Object

[LAB] Configuring S3 Life Cycle Management

- To manage your objects so that they are stored cost effectively throughout their life cycle.
- Life Cycle configuration is a set of rules that define actions that AWS S3 applies to group of objects.
- Transition Action:** Define when object transition to another class
- Expiration Action:** Define when object expires, S3 delete the expired object on your behalf

The screenshot shows the AWS S3 console for the 'my-cloud-bucket01' bucket. The 'Lifecycle' tab is selected under the 'Management' tab. A red arrow points to the '+ Add lifecycle rule' button. Below the tabs, it says 'There is no lifecycle rule applied to this bucket. Here is how to get started.' Three icons are shown: 'Use lifecycle rules to manage your objects', 'Automate transition to tiered storage', and 'Expire your objects'.

Step 1: Name and scope

Enter a rule name: my-cloud-bucket01-life-cycle
Choose a rule scope: Apply to all objects in the bucket
Next

Step 2: Transitions

storage class. Learn more or see Amazon S3 pricing
Current version Previous versions
For current versions of objects + Add transition
Object creation Days after creation
Transition to Intelligent-Tiering after 30 X
Transition to One Zone-IA after 60 X
Transition to Glacier after 90 X
⚠️ Transitioning small objects to Glacier or Glacier Deep Archive will increase costs.
I acknowledge that this lifecycle rule will increase the one-time lifecycle request cost if it transitions small objects.
Previous Next

Step 3: Expiration

Configure expiration
Current version Previous versions
Expire current version of object After 455 days from object creation
Clean up expired object delete markers
You cannot enable clean up expired object delete markers if you enable Expiration.
Clean up incomplete multipart uploads
Previous Next

Step 4: Review

Name: my-cloud-bucket01-life-cycle
Scope: Whole bucket
Transitions
For current version of objects
Transition to Intelligent-Tiering after 30 days
Transition to One Zone-IA after 60 days
Transition to Glacier after 90 days
Expiration
Expire after 455 days
⚠️ This rule applies to all objects in the bucket.
I acknowledge that this lifecycle rule will apply to all objects in the bucket.
Save Previous

Final Step: Confirmation

Amazon S3 > my-cloud-bucket01
my-cloud-bucket01
Overview Properties Permissions Management Access points
Lifecycle Replication Analytics Metrics Inventory
+ Add lifecycle rule Edit Delete Actions
Lifecycle rule Applied to Actions for current version(s)
my-cloud-bucket01-life-cycle Whole bucket Intelligent-Tiering / One Zone-IA / Glacier / Expire
Feedback English (US) © 2006–2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

[LAB] S3 Versioning

- Versioning allows to upload same file name multiple times, S3 will keep all the files with version ID.
- Versioning is a means of keeping multiple variants of an object in the same bucket.
- You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket.

The screenshots illustrate the AWS S3 console interface for managing versioning:

- Screenshot 1: Bucket Properties**
Shows the 'Properties' tab selected in the navigation bar. Under 'Versioning', it is set to 'Enabled'. Other features like 'Object-level logging' and 'Default encryption' are also shown.
- Screenshot 2: Object Details**
Shows the 'Latest version' of 'AWS-Logo.png'. Three versions are listed: Aug 25, 2020 (Latest version), Aug 25, 2020 (Aug 25, 2020 11:01:26 PM GMT+0530), and Jun 18, 2020 (Jun 18, 2020 1:16:21 AM GMT+0530). Each version has standard storage class and other metadata.
- Screenshot 3: Detailed View of Latest Version**
Shows a detailed view of the latest version (Aug 25, 2020). It includes fields like Last modified, Etag, Storage class, Server-side encryption, Expiration date, Expiration rule, Size, Key, and Object URL. A red arrow points from the 'Latest version' in Screenshot 2 to this detailed view.

AWS Reference: [S3 Versioning](#)

[LAB] Host a Web Site in S3

The screenshot shows the AWS S3 console for the bucket 'my-cloud-bucket01'. The 'Block public access' section is expanded, showing various options like 'Block all public access' and 'Block public access to buckets and objects granted through new access control lists (ACLs)'. The 'Static website hosting' section is also expanded, showing the endpoint 'http://my-cloud-bucket01.s3-website-ap-southeast-1.amazonaws.com'. The 'Default encryption' section is shown as disabled.

Block public access (bucket settings)

- Block all public access (Off)
 - Block public access to buckets and objects granted through new access control lists (ACLs)
 - Block public access to buckets and objects granted through any access control lists (ACLs) (Off)
 - Block public access to buckets and objects granted through new public bucket or access point policies
 - Block public and cross-account access to buckets and objects through any public bucket or access point policies (Off)

Operations: 0 In progress, 7 Success, 0 Error

Feedback, **English (US)**

Static website hosting

Endpoint: http://my-cloud-bucket01.s3-website-ap-southeast-1.amazonaws.com

Use this bucket to host a website (Learn more)

Index document: index.html

Error document: error.html

Redirection rules (optional):

Redirect requests (Learn more)

Disable website hosting

Bucket hosting

Object-level logging

Record object-level API activity using the CloudTrail data events feature (additional cost).

Default encryption

Automatically encrypt objects when stored in Amazon S3

Advanced settings

Operations: 0 In progress, 7 Success, 0 Error

Feedback, **English (US)**

403 Forbidden

Not secure | my-cloud-bucket01.s3-website-ap-southeast-1.amazonaws.com

Code: AccessDenied
Message: Access Denied
RequestId: 53DFD769E39C3754
HostId: m9JFZwrymvlWRnDwbZGrv2hj8sMGdpI1cdr9GzNDrVM9NGR6W/cZpIGQIMji48tv3AAIN49ng=

The screenshot shows a GitHub repository named 'sample-static-html-web-app'. The 'Code' tab is selected, showing the repository structure with files like 'css', 'fonts', 'img', 'README.md', and 'index.html'. A red arrow points to the 'Download ZIP' button. The repository has 1 star and 0 forks.

Code, **Issues**, **Pull requests**, **Actions**, **Projects**, **Wiki**, **...**

Code (dropdown), master (branch), Go to file, Add file, Download ZIP (highlighted with a red arrow)

Clone with HTTPS (dropdown), Use SSH

Use Git or checkout with SVN using the web URL.
https://github.com/vpjaseem/sample-sta

README.md, Add files via upload, 16 days ago

index.html

sample-static-html-web-app

About: No description, website, or topics provided.

Readme

Releases: No releases published. Create a new release

Packages: No packages published. Publish your first package

Languages: CSS 55.3%

Upload

1 Select files 2 Set permissions 3 Set properties 4 Review

12 Files Size: 823.3 KB Target path: my-cloud-bucket01

To upload a file larger than 160 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. Learn more ↗

+ Add more files

- css 1 Objects - 2.9 KB
- fonts 1 Objects - 789.2 KB
- img 8 Objects - 28.8 KB
- index.html - 2.4 KB
- README.md 29.0 B

Upload Next

Operations 0 In progress 7 Success 0 Error

Feedback English (US) © 2008 - 2020 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

sample-static-html-web-app-master sample-static-html-web-app-master

Name	Date modified	Type	Size
css	21-Jul-20 8:30 AM	File folder	
fonts	21-Jul-20 8:30 AM	File folder	
img	21-Jul-20 8:30 AM	File folder	
index.html	21-Jul-20 8:30 AM	Chrome HTML Do...	
README.md	21-Jul-20 8:30 AM	MD File	

Upload

1 Select files 2 Set permissions 3 Set properties 4 Review

12 Files Size: 823.3 KB Target path: my-cloud-bucket01

Manage users

User ID Objects Object permissions

jaseemc(Owner) Read Read Write

Access for other AWS account + Add account

Account Objects Object permissions

Manage public permissions

Grant public read access to this object(s)

⚠ This object(s) has public read access.
Everyone in the world will have read access to this object(s).

Upload Previous Next

Operations 0 In progress 7 Success 0 Error

Feedback English (US) © 2008 - 2020 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Selection: 2 Objects, 3 Folders Total size: 823.3 KB Total objects: 12

css/ 1 Objects - 2.9 KB

fonts/ 1 Objects - 789.2 KB

img/ 8 Objects - 28.8 KB

README.md - 29.0 B

index.html - 2.4 KB

Everyone will have access to one or all of the following: read this object, read and write permissions.

Cancel Make public

HTML Application

Not secure | my-cloud-bucket01.s3-website-ap-southeast-1.amazonaws.com

AJ Labs

Success!

Your Website is ready, HTML app is up and running.

Next up

- Practice the task and get familiar with Web App Deployments
- More tutorials and videos on my YouTube channel
- Connect me on LinkedIn | GitHub
- This is an HTML Website

AWS Snow Family

- Deploying AWS managed hardware and software to locations outside AWS Regions and even beyond AWS Outposts.
- These services help physically transport up to exabytes of data into and out of AWS.
- Used to carry data locally from outside to AWS DC.

AWS Snowcone



- Smallest member of the AWS Snow Family of edge computing and data transfer devices.
- Snowcone is portable, rugged, and secure. You can use Snowcone to collect, process, and move data to AWS, either offline by shipping the device, or online with AWS DataSync.

AWS Snowball



- AWS Snowball is a data migration and edge computing device that comes in two device options: Compute Optimized and Storage Optimized.

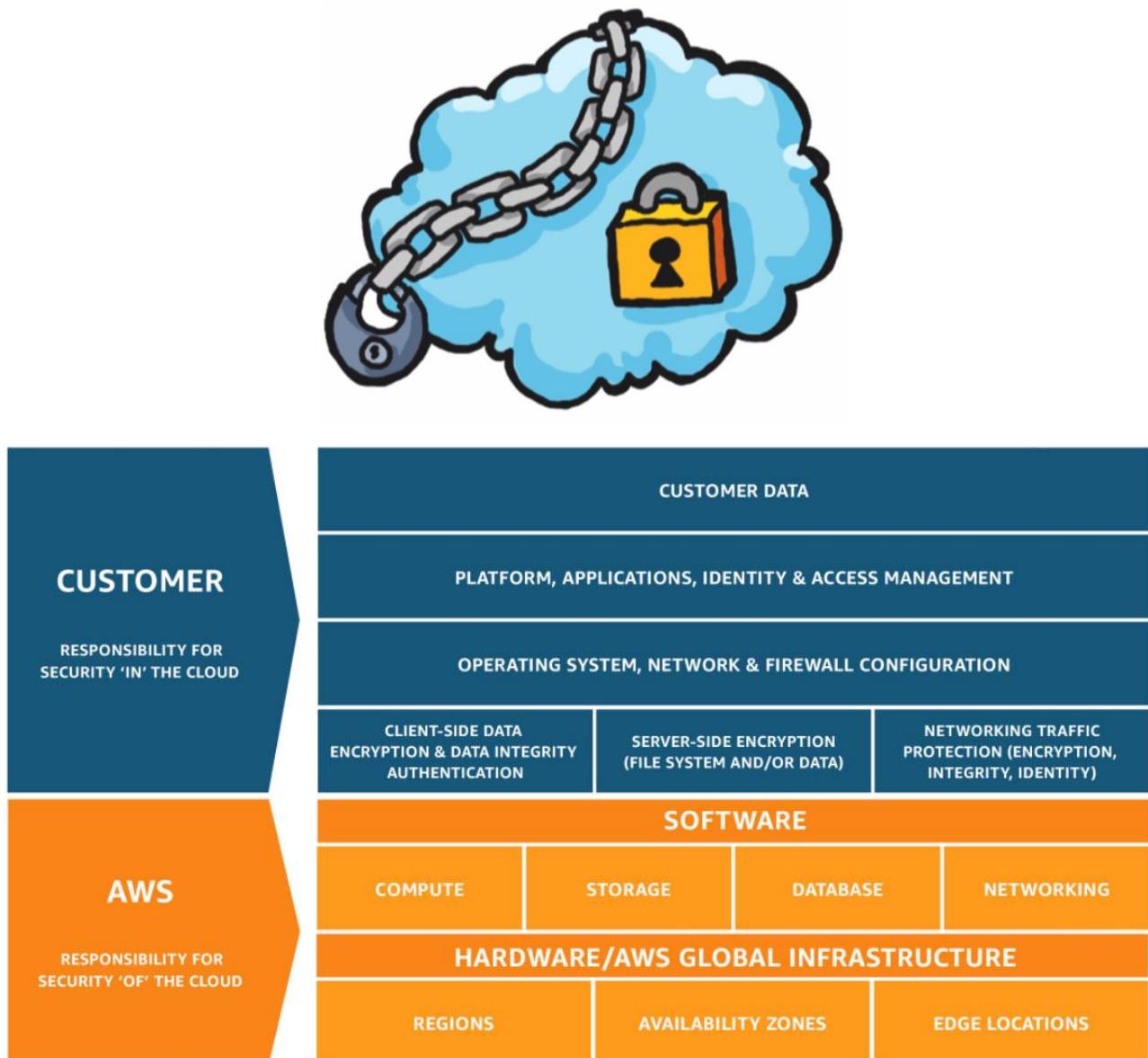
AWS Snowmobile



- AWS Snowmobile moves up to 100 PB of data in a 45-foot long ruggedized shipping container and is ideal for multi-petabyte or Exabyte-scale digital media migrations and data center shutdowns.
- A Snowmobile arrives at the customer site and appears as a network-attached data store for more secure, high-speed data transfer.
- After data is transferred to Snowmobile, it is driven back to an AWS Region where the data is loaded into Amazon S3.

AWS Reference: [Snow Family overview](#)

AWS Shared Responsibility Model

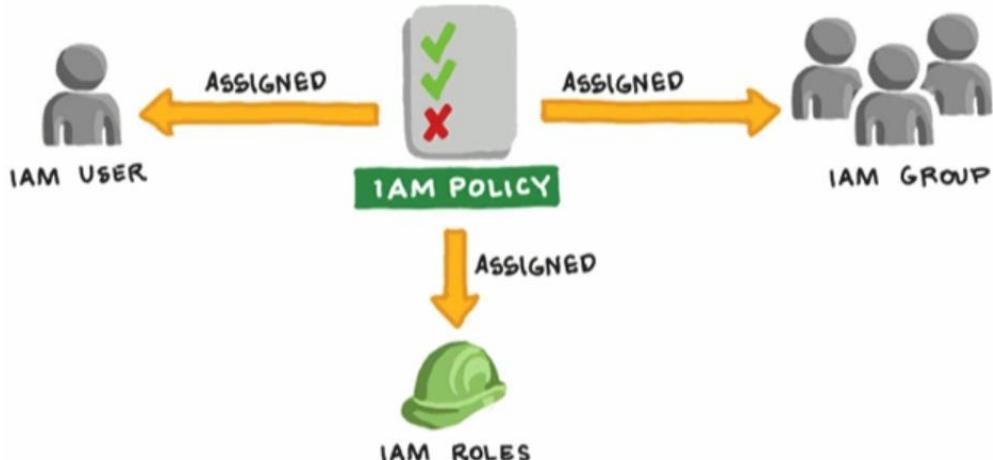


- Who responsible for what?
- What AWS is responsible for securing and what we responsible for securing our resources
- AWS is responsible for Physical Security, CPU, Memory, Storage, Networking resources, Hypervisor, etc.
- We are responsible for Firewall, Security Groups, System updates
- AWS DCs are undisclosed locations, we can't identify it, 24/7 security guard, 2FA for entry,
- Automated Change Control, Bastion Server to configure any devices, Abnormal activity detection
- Regulatory compliance: <https://aws.amazon.com/compliance/programs/>

Identity and Access Management (IAM)



- We do not use the 'root' user to perform day today activities.
- We create separate users with respective privileges.
- IAM allow us to create Users, assign permission to users, Groups and Roles, assign permission to groups and Roles.



- Provide a separate URL for IAM users to login.
- Also allows Multi Factor Authentication (MFA)
- IAM gives 2 key pairs (Access Key and Secret Key) to access AWS via API
- IAM policies are applied to User / Group / Roles and policies are written in JSON format.
- AWS provides Policy Generator (JSON format generator)
- Resources are defined with ARN (Amazon Resource Name)
- There are three different policies,
- **Managed Policies:** AWS provides for us (e.g. S3 Administrator)
- **Custom Policies:** We can create
- **Inline Policies:** They are custom policies designed and tied with resources, if we delete the resource, policy goes away.

[LAB] IAM Lab - Customize the IAM User Login URL (Account Alias)

The screenshot shows the AWS IAM Dashboard. In the top right corner, there is a red box around the sign-in link: <https://618927232701.signin.aws.amazon.com/console>. To the right of the link, there is a "Customize" button with a red arrow pointing to it.

- All the IAM users will use a separate URL to login to the AWS console.
- https://ACCOUNT_ID.signin.aws.com/console
- We can customize the account ID part to something else.
- Click the Customize button

The screenshot shows the "Create Account Alias" dialog box. It has two input fields: "Account" (containing "vpjaseemcloud") and "Alias". Below the fields are "Cancel" and "Yes, Create" buttons. A red arrow points to the "Alias" field.

- Enter an Account Alias name

The screenshot shows the AWS IAM Dashboard again. The sign-in link in the top right corner now reads: <https://vpjaseemcloud.signin.aws.amazon.com/console>, which is highlighted with a red box. The "Customize" button is also visible.

- My custom login URL is
<https://vpjaseemcloud.signin.aws.amazon.com/console>

signin.aws.amazon.com

signin.aws.amazon.com

vpjaseemcloud.signin.aws.amazon.com/console

Sign in

Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user

User within an account that performs daily tasks. [Learn more](#)

Root user email address

username@example.com

Next

New to AWS? [———](#)

[Create a new AWS account](#)

Sign in

Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user

User within an account that performs daily tasks. [Learn more](#)

Account ID (12 digits) or account alias

———

Next

New to AWS? [———](#)

[Create a new AWS account](#)

Sign in as IAM user

Account ID (12 digits) or account alias

———

IAM user name

———

Password

———

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

[LAB] Configure IAM User

The screenshot shows the AWS IAM 'Add user' wizard. In Step 1: Set user details, the 'User name' field is filled with 'anashira'. Below it, under 'Access type', the 'Programmatic access' checkbox is selected, and its description is visible. A red arrow points to the 'User name' field.

The screenshot shows the AWS IAM 'Add user' wizard. In Step 2: Set AWS access type, the 'Access type' section is shown with two checkboxes selected: 'Programmatic access' and 'AWS Management Console access'. A red arrow points to the 'Programmatic access' checkbox.

The screenshot shows the AWS IAM 'Add user' wizard. In Step 3: Set permissions, the 'Set permissions' section is shown with three options: 'Add user to group', 'Copy permissions from existing user', and 'Attach existing policies directly'. A red arrow points to the 'Create group' button.

The screenshot shows the AWS IAM 'Add user' wizard. In Step 4: Add tags (optional), the 'Add tags (optional)' section is shown with a single tag: 'Name: ami-aws-administrators'. A red arrow points to the 'Value (optional)' field.

The screenshot shows the AWS IAM 'Add user' wizard. In Step 5: Review, there is a warning message: 'This user has no permissions'. The 'User details' section shows 'User name: anashira' and 'AWS access type: Programmatic access and AWS Management Console access'. A red arrow points to the 'Create user' button.

The screenshot shows the AWS IAM 'Add user' wizard. In Step 6: Success, a success message is displayed: 'You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.' Below it, a table shows user credentials: 'User: anashira', 'Access key ID: AKIAZAGXQC26R4QOXQFQ', 'Secret access key: *****', and a 'Show' link. A red arrow points to the 'Show' link.

The screenshot shows a modal dialog from the AWS Management Console titled 'Add user'. The 'Success' section contains a message: 'You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.' Below this, it says 'Users with AWS Management Console access can sign-in at: <https://vpjaseemcloud.signin.aws.amazon.com/console>'. A red arrow points to the 'Download.csv' button. The main table lists one user, 'anashira', with columns: User, Access key ID, Secret access key, and Email login info. The 'Secret access key' column for 'anashira' is highlighted with a red box. At the bottom right of the modal is a 'Close' button.

User	Access key ID	Secret access key	Email login info
anashira	AKIAZAGXQC26R4QOXQFQ	IAwC7Gtss5IOJe18l0TR4+U5yl+VAHDM/QgeWUJq	Send email Hide

- We just created an IAM user
- Secret Access key is used to sign in to AWS via REST API or CLI (we will cover that later)

[LAB] Create IAM Group

Create New Group Wizard

Step 1 : Group Name

Group Name: aws-admin-users

Step 2 : Attach Policy

Step 3 : Review

Set Group Name

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

Group Name: aws-admin-users

Example: Developers or ProjectAlpha
Maximum 128 characters

Next Step

Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

Step 1 : Group Name

Step 2 : Attach Policy

Step 3 : Review

Create New Group Wizard

Review

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

Group Name: aws-admin-users

Policies

Create Group

Identity and Access Management (IAM)

Groups

Add Users to Group

Add Users to Group

Add Users

[LAB] Create IAM Policy

The screenshot shows the AWS IAM Policies page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and a 'Policies' section highlighted with a red arrow. The main area lists various AWS managed policies like 'AccessAnalyzerServiceRole', 'AdministratorAccess', etc. A red arrow points from the 'Create policy' button at the top to the 'Policy actions' dropdown.

This screenshot shows the 'Create policy' visual editor. It's set to the 'EC2 (All actions)' service. Under 'Actions', 'All EC2 actions (ec2:*)' is selected. A red arrow points from the 'Review policy' button at the bottom right to the 'Review policy' button at the bottom of the main screen.

This screenshot shows the 'Create policy' visual editor. Under 'Resources', 'All resources' is selected. A red arrow points from the 'Review policy' button at the bottom right to the 'Review policy' button at the bottom of the main screen.

This screenshot shows the 'Create policy' JSON editor. It displays a policy document with a red arrow pointing from the 'Review policy' button at the bottom right to the 'Review policy' button at the bottom of the main screen.

This screenshot shows the 'Review policy' step. It includes fields for 'Name' (aws-admins-iam-policy), 'Description' (aws-admins-iam-policy), and a 'Summary' table. A red arrow points from the 'Create policy' button at the bottom right to the 'Create policy' button at the bottom of the main screen.

This screenshot shows the IAM Policies page. It highlights the 'Attach' button in the 'Policy actions' dropdown for the 'aws-admins-iam-policy'. A red arrow points from the 'Create policy' button at the bottom right to the 'Create policy' button at the bottom of the main screen.

Attach policy

Attach the policy to users, groups, or roles in your account

Filter: Filter ▾ Q Search Showing 2 results

Name	Type
anashira	User
<input checked="" type="checkbox"/> aws-admin-users	Group

Cancel → Attach policy

Sign in as IAM user

Account ID (12 digits) or account alias: vpjaseemcloud

IAM user name: anashira

Password:

Sign in

Sign in using root user email | Forgot password? | LEARN MORE

English ▾ Terms of Use Privacy Policy © 1996-2020, Amazon Web Services, Inc. or its affiliates.

IAM user accessing EC2

New EC2 Experience Tell us what you think

EC2 Dashboard Events Tags Reports Limits Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations Images AMIs Bundle Tasks Elastic Block Store Volumes Snapshots Lifecycle Manager

Feedback English (US)

Services Resource Groups EC2 VP1 vpjaseemcloud Global Support

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Running instances	1
Elastic IPs	0
Dedicated Hosts	0
Snapshots	0
Volumes	2
Load balancers	-
Key pairs	1
Security groups	4
Placement groups	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch Instance

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

Privacy Policy Terms of Use

IAM user accessing S3

Amazon S3 S3 buckets Discover the console

Buckets + Create bucket Edit public access settings Empty Delete Buckets 0 Regions

Error Access Denied

Block public access (account settings)

Feature spotlight

Bucket name Access Region Date

Feedback English (US)

Privacy Policy Terms of Use

[LAB] IAM Role Configuration

- IAM roles are a secure way to grant permissions to entities that you trust.
- If we want our EC2 resource to access S3 bucket (without user ID, Password, Secret key)
- Securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting your credentials from other users.
- Commonly used for API or CLI access to AWS.
- For this lab, either you need to have AWS CLI installed on your Windows EC2 or you should have an Amazon Linux EC2 (AWS CLI is pre-installed). I'm going with Windows Installer.
- Follow the [link](#) to download setup file for Windows and install it on your Windows EC2. The installation is out of the scope of this document.

```

Administrator: Command Prompt
13.126.185.9

C:\Users\Administrator>aws --version
aws-cli/2.0.28 Python/3.7.7 Windows/10 botocore/2.0.0dev32

C:\Users\Administrator>aws s3 ls ←
Unable to locate credentials. You can configure credentials by running "aws configure".
C:\Users\Administrator>

```

Hostname: EC2AMAZ-4N65INS
Instance ID: i-0a7a89785fa55d872
Public IP Address: 13.126.185.9
Private IP Address: 172.16.11.224
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate

- I have used an AWS CLI command (aws s3 ls) to list the S3 bucket details, you can see that there is an error message 'Unable to locate credentials' because we did not configure anything.
- To provide access to S3 from this machine, without a username and password, we use IAM roles.

Identity and Access Management (IAM)

- Dashboard
- Access management
- Groups
- Users
- Roles** ←
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analyzers
- Settings

Credential report

- Organization activity
- Service control policies (SCPs)

Search IAM

AWS account ID: G18927232701

Create role

Select type of trusted entity

- AWS service** EC2, Lambda and others
- Another AWS account Belonging to you or 3rd party
- Web identity Cognito or any OpenID provider
- SAML 2.0 Your provider

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

- EC2** Allows EC2 instances to call AWS services on your behalf.
- Lambda Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeGuru	ElastiCache	Kinesis	RoboMaker
AWS Backup	CodeStar Notifications	Elastic Beanstalk	Lake Formation	S3
AWS Chatbot	Comprehend	Elastic Container Service	Lambda	SMS
AWS Support	Config	Elastic Transcoder	Lex	SNS
Amplify	Connect	Elastic Load Balancing	License Manager	SWF
AppStream 2.0	DMS	Forecast	Machine Learning	SageMaker
AppSync	Data Lifecycle Manager	GameLift	Macie	Security Hub
Application Auto Scaling	Data Pipeline	Global Accelerator	Managed Blockchain	Service Catalog
Amazon Connect	DataSync	Glue	MediaConvert	Step Functions

Required

Next: Permissions

The composite screenshot illustrates the process of creating an IAM role and attaching it to an EC2 instance.

Step 1: Create IAM Role - Attach permissions policies

In the AWS IAM console, a user is creating a new role named "s3-full-access-role-for-ec2". They have selected the "AmazonS3FullAccess" policy, which is highlighted with a red arrow. The "Create policy" button is visible at the top left.

Step 2: Create IAM Role - Add tags (optional)

The user adds a tag "Name: s3-full-access-role-for-ec2" to the role. A red arrow points to the "Value (optional)" field.

Step 3: Create IAM Role - Review

The user reviews the role configuration, including the role name "s3-full-access-role-for-ec2" and the attached policy "AmazonS3FullAccess". A red arrow points to the "Create role" button.

Step 4: EC2 Instances - Attach/Replace IAM Role

In the AWS EC2 Instances page, the user selects the instance "public-windows-server-2019" and opens the Actions menu. The "Attach/Replace IAM Role" option is highlighted with a red arrow. The instance details show it is currently using the "aws-windows" AMI and has the "aws-windows" key pair.

Step 5: EC2 Instances - Attach/Replace IAM Role (Completed)

The user has completed the "Attach/Replace IAM Role" process for the instance "public-windows-server-2019". The instance now shows the "s3-full-access-role-for-ec2" IAM role attached, indicated by a red arrow pointing to the "IAM role" field in the instance details.

- Now the Windows EC2 can access S3 (AWS CLI commands) from Windows CMD

Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.1282]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>aws s3 ls
2020-06-18 15:51:07 my-cloud-bucket01
C:\Users\Administrator>

Hostname: EC2AMAZ-4N65INS
Instance ID: i-0a7a89785fa55d872
Public IP Address: 13.126.185.9
Private IP Address: 172.16.11.224
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate

[LAB] Inline Policy for IAM User

- These are the policies directly tied-up with Users, once you delete the user / group, the policy will get deleted.

The screenshot shows the AWS IAM User Summary page for a user named 'anashira'. The 'Permissions' tab is selected, showing one policy applied: 'aws-admins-iam-pol...' (Managed policy from group aws-admin-users). A red box highlights the 'Add inline policy' button.

The screenshot shows the 'Create policy' page. The 'Visual editor' tab is selected. Under 'Service', 'S3' is chosen. Under 'Actions', 'All S3 actions (s3:*)' is selected. A red box highlights the 'Resources' section, which is set to 'All resources'. At the bottom right, a red arrow points to the 'Review policy' button.

The screenshot shows the 'Review policy' page. The policy name is 'aws-s3-lam-inline-policy'. The 'Summary' section shows 'Allow (1 of 233 services) Show remaining 232' with 'S3 Full access All resources'. A red box highlights the 'Create policy' button at the bottom right.

The screenshot shows the AWS IAM User Summary page for 'anashira' again. The 'Permissions' tab now shows two policies applied: 'aws-s3-lam-inline-policy' (inline policy) and 'aws-admins-iam-pol...' (Managed policy from group aws-admin-users).

The screenshot shows the AWS Amazon S3 console. The 'Buckets' section lists a single bucket: 'my-cloud-bucket01'. A red box highlights the 'Amazon S3' link in the top navigation bar.

[LAB] Inline Policy for IAM Group

- Group specific policies can be assigned by this way, once you delete the group, the policy will get deleted.

Identity and Access Management (IAM)

Dashboard

Access management

Groups ←

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCPs)

AWS account ID: 618927232701

Search IAM

IAM > Groups > aws-admin-users

Summary

Group ARN: arn:aws:iam:618927232701:group/aws-admin-users

Users (in this group): 1

Path: /

Creation Time: 2020-07-05 01:22 UTC+0530

Users Permissions Access Advisor

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Attach Policy

Policy Name	Actions
aws-admins-lam-policy	Show Policy Detach Policy Simulate Policy

Inline Policies

There are no inline policies to show. To create one, click here

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Manage Group Permissions

Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

Policy Generator

Use the policy generator to create your own set of permissions.

Policy Generator Select

Custom Policy

Cancel

Manage Group Permissions

Edit Permissions

The policy generator enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see Overview of Policies in Using AWS Identity and Access Management.

Effect Allow Deny

AWS Service Amazon S3

Actions All Actions Selected

Amazon Resource Name (ARN) arn:aws:s3:::bucket-name

Add Conditions (optional)

Add Statement

Cancel Previous Next Step

Manage Group Permissions

Edit Permissions

The policy generator enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more information about creating policies, see Overview of Policies in Using AWS Identity and Access Management.

Effect Allow Deny

AWS Service AWS AppSync

Actions --Select Actions--

Amazon Resource Name (ARN)

Add Conditions (optional)

Add Statement

Effect	Action	Resource
Allow	s3:*	*

Cancel Previous Next Step

Manage Group Permissions

Review Policy

Customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in the Using IAM guide. To test the effects of this policy before applying your changes, use the IAM Policy Simulator.

Policy Name aws-s3-group-inline-policy

```

1 "Version": "2012-10-17",
2 "Statement": [
3     {
4         "Sid": "Stmt1593953082000",
5         "Effect": "Allow",
6         "Action": [
7             "s3:*"
8         ],
9         "Resource": [
10            "*"
11        ]
12    }
13 ]
14 }
15 }
```

Use autoformatting for policy editing

Cancel Validate Policy Apply Policy

Identity and Access Management (IAM)

Dashboard

Access management

Groups ←

Users

Roles

Policies

Identity providers

Account settings

Access reports

Access analyzer

Archive rules

Analyzers

Settings

Credential report

Organization activity

Service control policies (SCPs)

AWS account ID: 618927232701

Search IAM

IAM > Groups > aws-admin-users

Summary

Group ARN: arn:aws:iam:618927232701:group/aws-admin-users

Users (in this group): 1

Path: /

Creation Time: 2020-07-05 01:22 UTC+0530

Users Permissions Access Advisor

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Attach Policy

Policy Name	Actions
aws-admins-lam-policy	Show Policy Detach Policy Simulate Policy

Inline Policies

This view shows all inline policies that are embedded in this group.

Create Group Policy

Policy Name	Actions
aws-s3-group-inline-policy	Show Policy Edit Policy Remove Policy Simulate Policy

127

AWS Organizations

- Master and Child account concept
- Consolidated billing of multiple accounts
- Service Control Policy (Specific roles for child account)
- AWS Organizations enables you to centrally apply policy-based controls across multiple accounts in the AWS Cloud.
- You can consolidate all your AWS accounts into an organization, and arrange all AWS accounts into distinct organizational units.
- Invite existing AWS accounts to join your organization or create new accounts based on your needs.

[LAB] Creating a Child Account and Provide t2.micro Policy

The screenshots illustrate the process of creating a child AWS account and granting it a specific policy.

Step 1: Create a New AWS Account

In the AWS Organizations console, under the "Accounts" tab, a modal window titled "Check your email to finish verifying your master account" is displayed. It contains instructions to verify the master account email (jaseemo@gmail.com) and invite other accounts. Below this, a table lists existing accounts: jaseemo@gmail.com (Joined on 8/15/20).

Step 2: Add an Existing AWS Account

In the same AWS Organizations console, a modal window titled "How do you want to add an account to your organization?" offers two options: "Invite account" and "Create account". The "Create account" button is highlighted with a red arrow.

Step 3: Verify Email Address

The AWS Organizations console shows a success message: "Your email address has been verified. You can now invite existing AWS accounts to join your organization." The account ajabs110@gmail.com is listed as created on 8/15/20.

Step 4: Create an Organizational Unit

A modal window titled "Create organizational unit" is shown. It asks for the "Name of organizational unit" (AWS Developer Accounts) and has a "Create organizational unit" button highlighted with a red arrow. A sidebar on the right shows the account details for jaseemo@gmail.com.

Step 5: Move an Account to an Organizational Unit

The AWS Organizations console displays the "Root" organizational unit. The account ajabs110@gmail.com is selected and highlighted with a red arrow. A "Move" button is visible at the bottom left of the account list.

Step 6: Move Account Confirmation

A modal window titled "Move 1 account" asks to choose the target organizational unit (OU). The "Root" OU is selected and highlighted with a red arrow. A "Move" button is highlighted with a red arrow.

AWS Organizations

Organize accounts (selected)

Policies

Root

Service control policies (Enabled)

Tag policies (Enabled)

AI services opt-out policies (Enabled)

Backup policies (Enabled)

Service control policies (Enabled)

Tag policies (Enabled)

AI services opt-out policies (Enabled)

Backup policies (Enabled)

Policy type

Service control policies (Enabled)

Tag policies (Disabled)

AI services opt-out policies (Disabled)

Backup policies (Disabled)

Feedback English (US)

AWS Organizations

Policies > **Service control policies**

Service control policies (SCPs) offer central control over the maximum available permissions for all accounts in your organization, allowing you to ensure your accounts stay within your organization's access control guidelines.

Create policy **Delete policy**

Name	Type	Description
FullAWSAccess	Service c...	Allows access to every operation

Policy name * **ec2-instance-t2.micro-only**

Description **ec2-instance-t2.micro-only**

Policy *

```

1: {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RequireMicroInstanceType",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "StringNotEquals": {
          "ec2:InstanceType": "t2.micro"
        }
      }
    }
  ]
}

```

Statement not selected
Choose a policy statement on the right

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RequireMicroInstanceType",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "StringNotEquals": {
          "ec2:InstanceType": "t2.micro"
        }
      }
    }
  ]
}
```

The screenshot displays two adjacent AWS browser windows. The left window shows the AWS Organizations console under the 'Organize accounts' tab. It lists a single organizational unit named 'Root' containing an account named 'vpjasseemCloud'. The right window shows the 'Launch Status' page for a launch named 'AWS Developer A...'. A red arrow points from the 'AWS Developer A...' label in the left window to the same label in the right window. Another red arrow points from the 'Attach' button in the left window to the 'Attach' button in the right window. A third red arrow points from the 'Launch Failed' status indicator in the right window back to the 'AWS Developer A...' label in the left window. The right window's status panel contains the following information:

Launch Failed	
You are not authorized to perform this operation. Encoded authorization failure message: CQRGr2YOA7P4KUrP02mPvAvWlBlu7SnlSgS6ExOpJZ6CIEB5_2n9z_0RNXP4ZB-aNETQ0kIS9eUDnSFZPVwCBK3A-9MspPi0-nHtByPAKplKM61WHVmApjk2Tj04K934NQS4kHTD0fVSvnmcK2fBa92_K_wak-hzEyBVZ_LpXK7acUBvx24WKLrDyBi-3zPZNfc_1_CjSj03uCH4v9NsAa6VKBPKoN44SCr-p0auM4ltlyxx-TFKPPOO1mB-GaMACauVTXuq0etqTqTKPkm7IRHPGuEgchIKSWORRac7namAH_bwNTRGByEo-VjylrVFC1hpWlRef6c86tKzgcBTXmk-2WbsEjuId_7my8lU97qjblC2g01Isqno_6HlkGDF238qyT3McnyY9AOrInczEnD95qehlJSZfzwplVnfGpJc1DUNg-bPzTAirkJqqwdomyUduJxdx5fb5u4BPFP_LVjpwtsqXRytqW3Bg1gb7s4OKbMYz2wE6n7tMU4bw59heNH2HNPodtTINRLKAfWo60PQXWwQfHxE4VVwdtpCXCfHrLlJa3Ql26wv9T_IQ9sZE04PDm4fjXpR8dx6frtSbYT0aISRVif2P0jutv4VlnkQpJ95-QQAZkrSVYuZLds_jwGrv0986NuX77dxP4bIVcCAZ-H-GSejGwnIOL7KLBHlYacv6RtaCkwfRTEJaGqQQ8mGMSbcGzqjI0XKNA9gZghVZXJuCuIJSS7ym_JAAtsgz2HDNmBd	
Creating security groups	Successful (sg-0ed16cbe24ad936af)
Authorizing inbound rules	Successful
Initiating launches	Failure Retry

At the bottom of the right window, there are buttons for 'Cancel', 'Back to Review Screen', and 'Retry Failed Tasks'.

AWS Reference: [What is AWS Organizations](#)

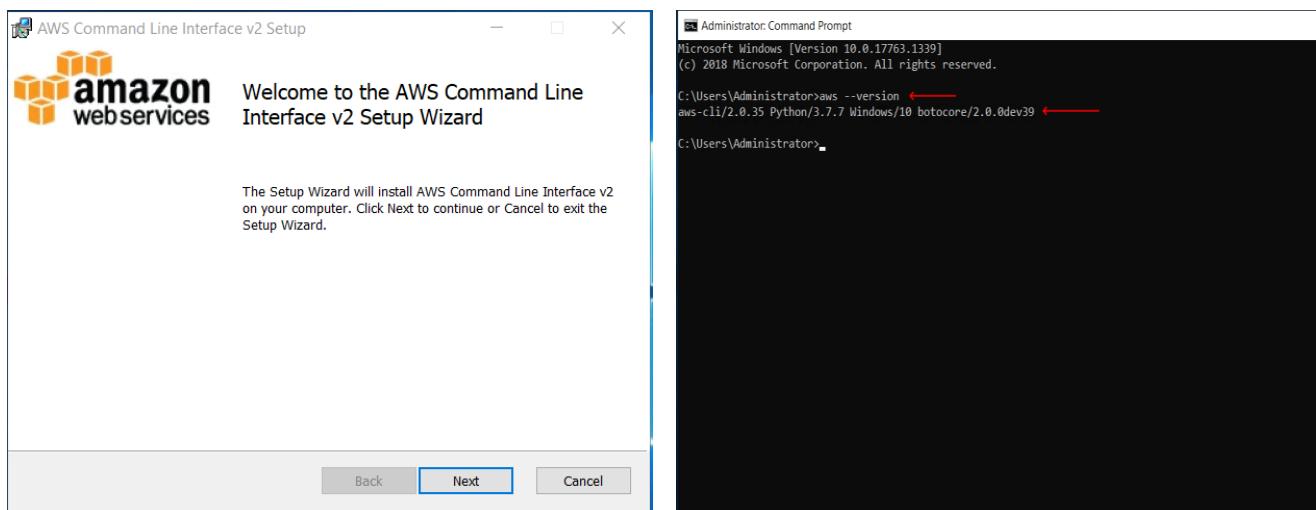
AWS CLI

- The AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell.
- With minimal configuration, the AWS CLI enables you to start running commands that implement functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your favorite terminal program.

Install AWS CLI on Windows

- Download AWS CLI Setup file from this link: <https://awscli.amazonaws.com/AWSCLIV2.msi>
- Go ahead and install the program (On your PC or AWS EC2)

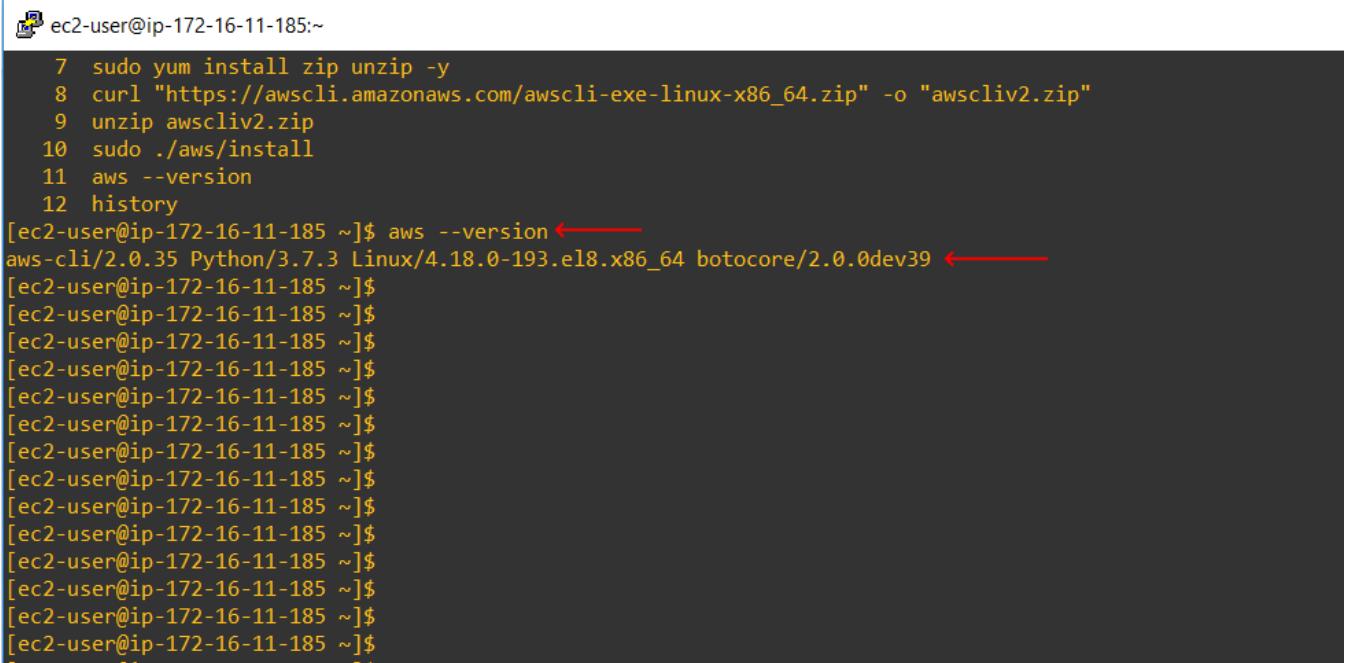
AWS Reference: [Installing the AWS CLI version 2 on Windows](#)



Install AWS CLI on Linux

- The Amazon Linux AMI comes with pre-installed AWS CLI, If you have any other Linux distribution, follow below steps.

```
sudo yum install zip unzip -y
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
aws --version
```



A screenshot of a terminal window titled 'ec2-user@ip-172-16-11-185:~'. The terminal shows the command history and the output of the 'aws --version' command. The output indicates that the AWS CLI version is 2.0.35, Python is 3.7.3, the operating system is Linux/4.18.0-193.el8.x86_64, and botocore is 2.0.0dev39.

```
[ec2-user@ip-172-16-11-185 ~]$ sudo yum install zip unzip -y
[ec2-user@ip-172-16-11-185 ~]$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
[ec2-user@ip-172-16-11-185 ~]$ unzip awscliv2.zip
[ec2-user@ip-172-16-11-185 ~]$ sudo ./aws/install
[ec2-user@ip-172-16-11-185 ~]$ aws --version
aws-cli/2.0.35 Python/3.7.3 Linux/4.18.0-193.el8.x86_64 botocore/2.0.0dev39
[ec2-user@ip-172-16-11-185 ~]$ [ec2-user@ip-172-16-11-185 ~]$
```

AWS Access Key and Secret Key Authentication

- Access key and Secret key are used to access AWS via CLI (programmatic access), these are the credentials used to authenticate AWS from CLI.
- Go to IAM and create a user, make sure you copy Access key and Secret key.

The screenshot shows the AWS IAM 'Add user' wizard across five steps:

- Step 1: Set user details**: User name is set to "aws-api-user".
- Step 2: Set permissions**: The "AdministratorAccess" policy is selected under "Attach existing policies directly".
- Step 3: Set permissions boundary**: No boundary is selected.
- Step 4: Review**: Summary of user creation: User name: aws-api-user, AWS access type: Programmatic access and AWS Management Console access, Console password type: Custom, Require password reset: Yes, Permissions boundary: Not set.
- Step 5: Success**: Confirmation message: "You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time." The user "aws-api-user" is listed with their Access key ID and Secret access key.

- Login to AWS CLI and enter below,

```
[ec2-user@ip-172-16-11-185 ~]$ aws configure
AWS Access Key ID [None]: AKIAZAGXQCXXXXXXXXHW
AWS Secret Access Key [None]: kfNVFpkynXXXXXXXXXXD9pGtCqX1HuoNb9CPTaI
Default region name [None]: ap-south-1
Default output format [None]:
```

- These data will be saved in '/home/ec2-user/.aws'
- Now the AWS CLI authenticated with our AWS account

```
[ec2-user@ip-172-16-11-185 ~]$  
[ec2-user@ip-172-16-11-185 ~]$  
[ec2-user@ip-172-16-11-185 ~]$ pwd ←  
/home/ec2-user  
[ec2-user@ip-172-16-11-185 ~]$ ls -ltr -a  
total 32536  
-rw-r--r--. 1 ec2-user ec2-user 312 Aug 30 2019 .bashrc  
-rw-r--r--. 1 ec2-user ec2-user 141 Aug 30 2019 .bash_profile  
-rw-r--r--. 1 ec2-user ec2-user 18 Aug 30 2019 .bash_logout  
drwxr-xr-x. 3 ec2-user ec2-user 78 Jul 28 21:07 aws ←  
drwxr-xr-x. 3 root root 22 Jul 29 19:54 ..  
drwx----- 2 ec2-user ec2-user 29 Jul 29 19:54 .ssh  
-rw-rw-r--. 1 ec2-user ec2-user 33303407 Jul 29 19:59 awscliv2.zip  
drwxrwxr-x. 2 ec2-user ec2-user 39 Jul 29 20:26 .aws ←  
drwx----- 5 ec2-user ec2-user 117 Jul 29 20:26 .  
[ec2-user@ip-172-16-11-185 ~]$ cd .aws/  
[ec2-user@ip-172-16-11-185 .aws]$ ls  
config credentials ←  
[ec2-user@ip-172-16-11-185 .aws]$ cat credentials ←  
[default]  
aws_access_key_id = AKIAZAGXQC2H0000CJ2W0000  
aws_secret_access_key = kfNVFpkynJZbYvJyCNnD9pCPTaI  
[ec2-user@ip-172-16-11-185 .aws]$ █
```

Launch EC2 Instance Using AWS CLI

- You can use the AWS Command Line Interface (AWS CLI) to launch, list, and terminate Amazon Elastic Compute Cloud (Amazon EC2) instances.

1. Launch EC2 Instance

```
aws ec2 run-instances --image-id ami-03dbf9550d4620230 --count 1 --instance-type t2.micro -  
-key-name aws-windows --security-group-ids sg-03bc17a19529ccc88 --subnet-id subnet-  
05bd2cc86c1e09ec5
```

```
ec2-user@ip-172-16-11-185:~/aws  
[ec2-user@ip-172-16-11-185 .aws]$  
[ec2-user@ip-172-16-11-185 .aws]$  
[ec2-user@ip-172-16-11-185 .aws]$ aws ec2 run-instances --image-id ami-03dbf9550d4620230 --count 1 --instance-type t2.micro --key-name aws-windows --security-group-ids sg-03bc17a19529ccc88 --subnet-id subnet-05bd2cc86c1e09ec5  
{  
    "Groups": [],  
    "Instances": [  
        {  
            "AmiLaunchIndex": 0,  
            "ImageId": "ami-03dbf9550d4620230",  
            "InstanceId": "i-0a998993c7e1e334b",  
            "InstanceType": "t2.micro",  
            "KeyName": "aws-windows",  
            "LaunchTime": "2020-07-29T20:55:30+00:00",  
            "Monitoring": {  
                "State": "disabled"  
            },  
            "Placement": {  
                "AvailabilityZone": "ap-south-1a",  
                "GroupName": "",  
                "Tenancy": "default"  
            },  
            "Platform": "Windows",  
            "PrivateDnsName": "ip-172-16-11-35.ap-south-1.compute.internal",  
            "PrivateIpAddress": "172.16.11.35",  
            "ProductCodes": [],  
            "PublicDnsName": "",  
            "State": {  
                "Code": 0,  
                "Name": "pending"  
            },  
            "StateTransitionReason": "",  
            "SubnetId": "subnet-05bd2cc86c1e09ec5",  
            "VpcId": "vpc-00c3b6efcf0ee9add",  
            "...skipping..."  
        }  
    ]  
}
```

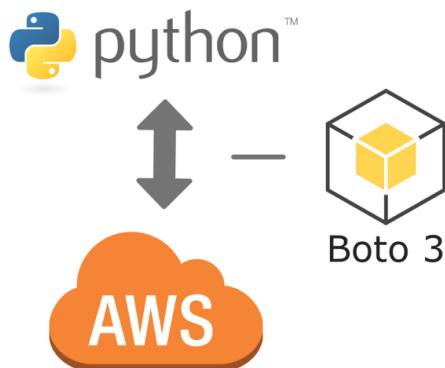
2. Stop EC2 Instance

```
aws ec2 stop-instances --instance-ids i-0a998993c7e1e334b
```

3. Terminate EC2 Instance

```
aws ec2 terminate-instances --instance-ids i-0a998993c7e1e334b
```

AWS Python API / SDK - BOTO3



- Boto is the Amazon Web Services (AWS) SDK for Python.
- It enables Python developers to create, configure, and manage AWS services, such as EC2 and S3.
- Boto provides an easy to use, object-oriented API, as well as low-level access to AWS services.

Boto3 Reference: [Boto3 Documentation](#)

Install BOTO3

- We use the popular Python Package Installer (PIP) for installing Boto3 (pip install boto3)

```

Administrator: Command Prompt
C:\Users\Administrator>
C:\Users\Administrator>
C:\Users\Administrator>
C:\Users\Administrator>pip install boto3
Collecting boto3
  Downloading boto3-1.14.31-py2.py3-none-any.whl (129 kB)
    ██████████| 129 kB 3.2 MB/s
Collecting s3transfer<0.4.0,>=0.3.0
  Downloading s3transfer-0.3.3-py2.py3-none-any.whl (69 kB)
    ██████████| 69 kB 870 kB/s
Collecting jmespath<1.0.0,>=0.7.1
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting botocore<1.18.0,>=1.17.31
  Downloading botocore-1.17.31-py2.py3-none-any.whl (6.4 MB)
    ██████████| 6.4 MB 2.2 MB/s
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
    ██████████| 227 kB 2.2 MB/s
Collecting docutils<0.16,>=0.10
  Downloading docutils-0.15.2-py3-none-any.whl (547 kB)
    ██████████| 547 kB 6.4 MB/s
Collecting urllib3<1.26,>=1.20; python_version != "3.4"
  Downloading urllib3-1.25.10-py2.py3-none-any.whl (127 kB)
    ██████████| 127 kB 1.3 MB/s
Collecting six>=1.5
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: six, python-dateutil, docutils, urllib3, jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.14.31 botocore-1.17.31 docutils-0.15.2 jmespath-0.10.0 python-dateutil-2.8.1 s3transfer-0.3.3 six-1.15.0 urllib3-1.25.10
  
```

Start / Stop EC2 Instance with Python Boto3 Module

- Run below Python Script to Start an EC2 Instance.

```
import boto3
region = 'ap-south-1'
instances = ['i-09587b4cb446b0072']
ec2 = boto3.client('ec2', region_name=region)
ec2.start_instances(InstanceIds=instances)
print('Started your instances: ' + str(instances))
```

- Run below Python Script to Stop an EC2 Instance.

```
import boto3
region = 'ap-south-1'
instances = ['i-09587b4cb446b0072']
ec2 = boto3.client('ec2', region_name=region)
ec2.stop_instances(InstanceIds=instances)
print('Stopped your instances: ' + str(instances))
```

The screenshot shows two windows. The left window is a code editor with the file 'stop-ec2.py' open. It contains the Python script provided above. A red arrow points to the first line 'import boto3'. The right window is a terminal titled 'Python 3.8.5 Shell'. It shows the command 'Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32' and the output of running the script: 'Type "help", "copyright", "credits" or "license()" for more information.' followed by '>>> ===== RESTART: C:/Users/Administrator/Desktop/stop-ec2.py =====' and 'Stopped your instances: ['i-09587b4cb446b0072']'.

```
stop-ec2.py - C:/Users/Administrator/Desktop/stop-ec2.py (3.8.5)
File Edit Format Run Options Window Help
import boto3
region = 'ap-south-1'
instances = ['i-09587b4cb446b0072']
ec2 = boto3.client('ec2', region_name=region)
ec2.stop_instances(InstanceIds=instances)
print('Stopped your instances: ' + str(instances))

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/Administrator/Desktop/stop-ec2.py =====
Stopped your instances: ['i-09587b4cb446b0072']
```

Deploy EC2 using Python Boto3 API

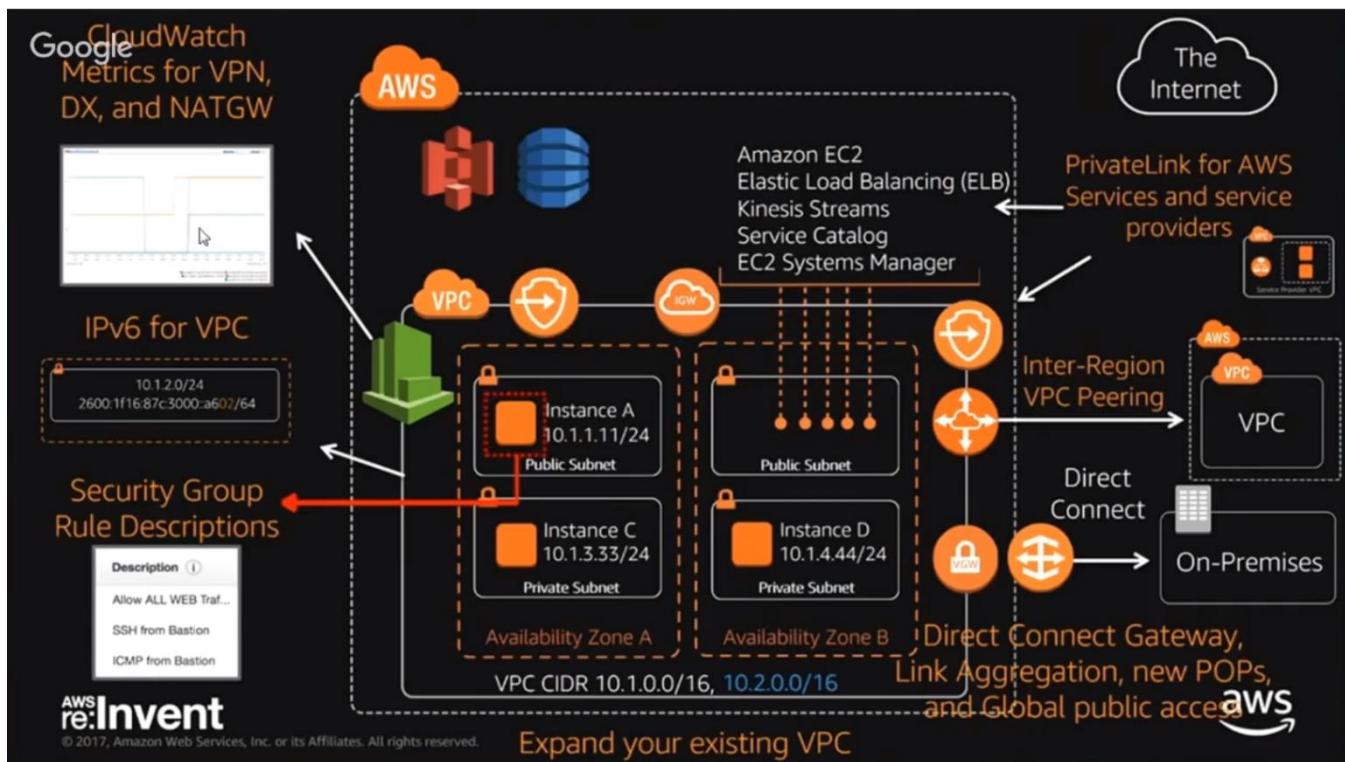
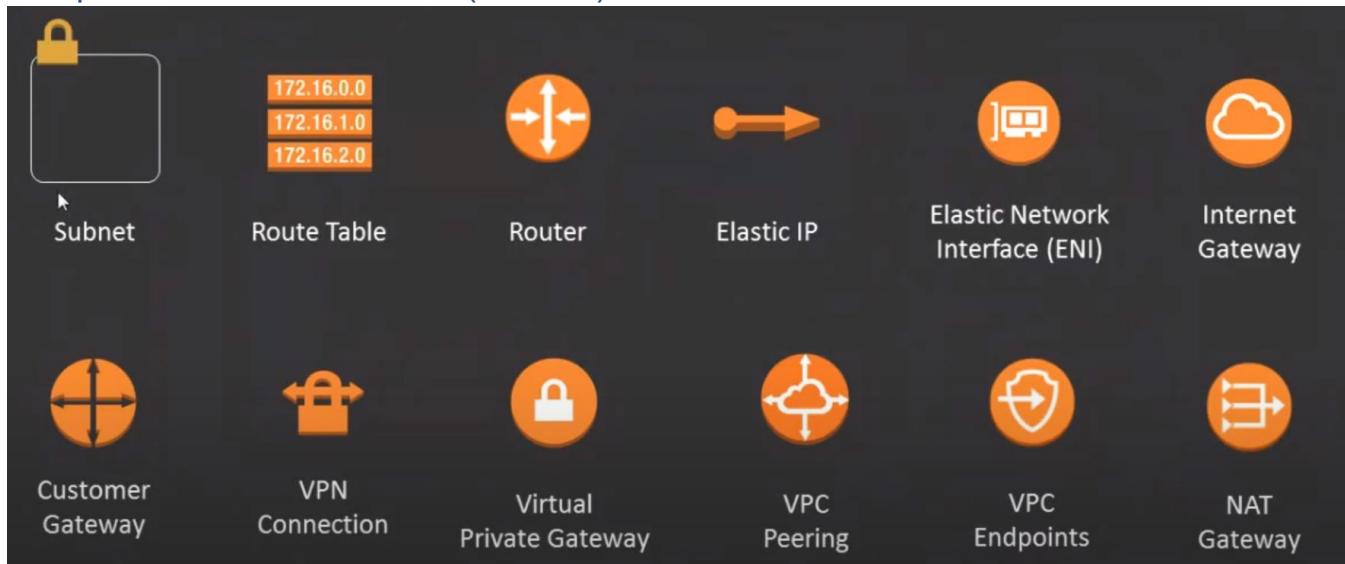
- This is the beauty of Automation, Boto3 has developed such a way that we can deploy an EC2 just in seconds.

```
import boto3
ec2 = boto3.resource('ec2', region_name='ap-south-1')

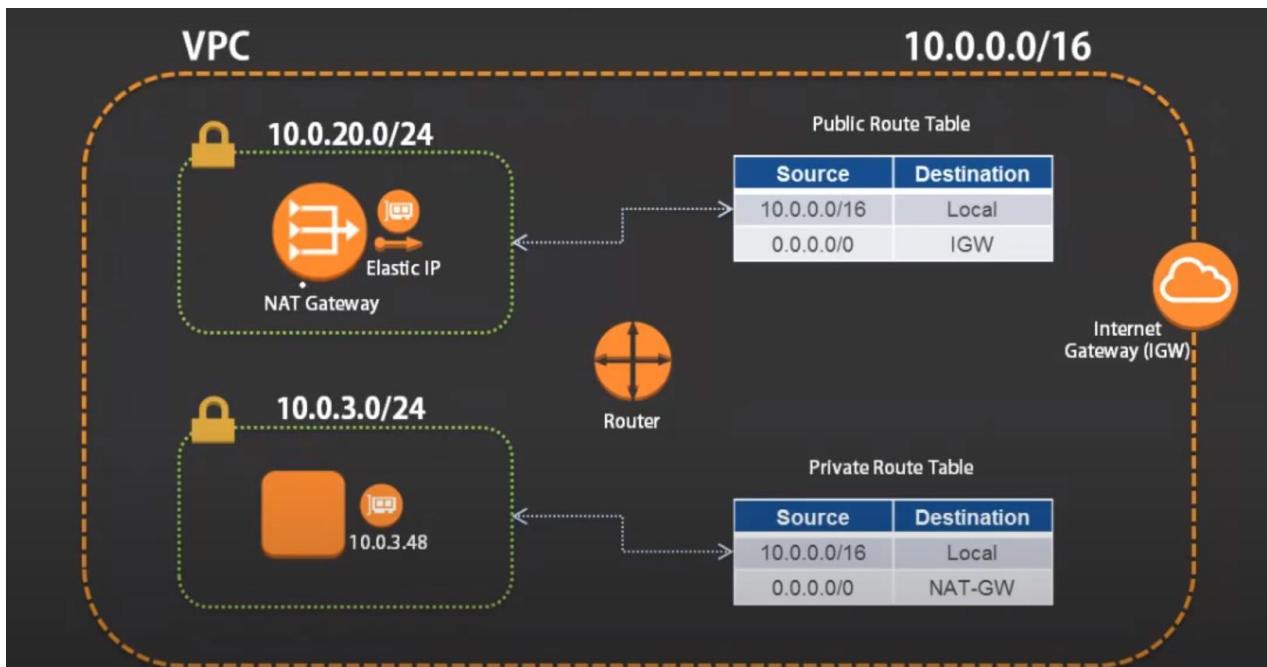
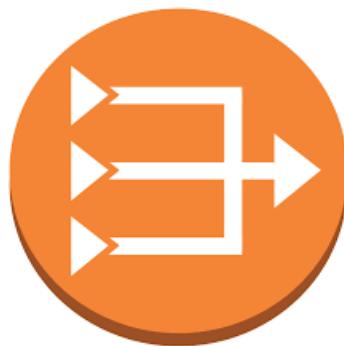
# create a new EC2 instance
instances = ec2.create_instances(
    ImageId='ami-03dbf9550d4620230',
    MinCount=1,
    MaxCount=1,
    InstanceType='t2.micro',
    KeyName='aws-windows',
    NetworkInterfaces=[{
        'SubnetId': 'subnet-05bd2cc86c1e09ec5',
        'DeviceIndex': 0,
        'AssociatePublicIpAddress': True,
        'Groups': ['sg-03bc17a19529ccc88'],
    }]
)
print('Successfully launched EC2 Instance')
```

The screenshot shows two windows side-by-side. The left window is a code editor titled "create-new.py - C:\Users\Administrator\Desktop\create-new.py (3.8.5)". It contains the Python script provided above. The right window is a "Python 3.8.5 Shell" window. It shows the command-line interface with the Python interpreter. The output from the script execution is visible in the shell window, showing the message "Successfully launched EC2 Instance".

Components of VPC Overview (Detailed)

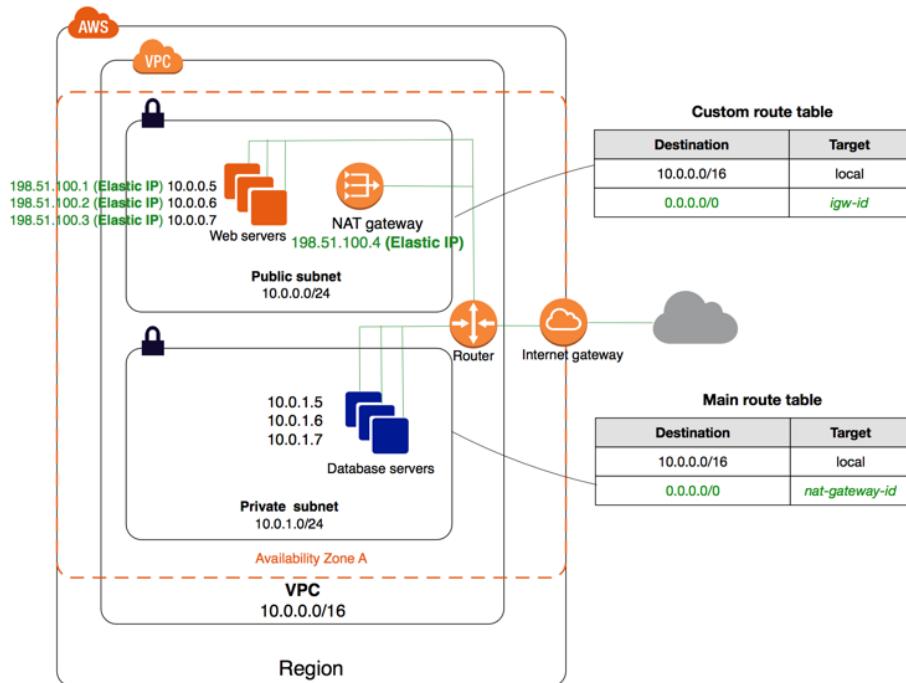


NAT Gateway



- An AWS managed Network Address Translation (NAT) gateway to enable instances in Private Subnet to connect to internet or other AWS services, but prevent internet from initiating a connection to those instances.
- Allows your EC2 instances to perform updates/ patches etc. still being inside private subnet.
- Works only for IPV4. We use Egress-only Internet Gateway for IPV6
- Private Subnet checks Route Table and sees NAT-GW, the NAT-GW is in another subnet that has internet access via IGW and also a Public IP. (Private EC2 >> Route Table >> NAT Gateway >> IGW)
- Completely managed by AWS
- So far, we have configured EC2 instances in Public Subnet, what if we configure an EC2 on private subnet? How do we access them? Whether the EC2 gets internet?
- You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.
- To access EC2 instances in Private Subnet, you need to have either VPN connection to VPC or access an EC2 in Public Subnet and from there connect to private subnet instances.
- You must also specify an Elastic IP address to associate with the NAT gateway when you create it.

- After you've created a NAT gateway, you must update the route table associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway. This enables instances in your private subnets to communicate with the internet.



AWS Reference: [NAT Gateway](#)

- A NAT gateway supports 5 Gbps of bandwidth and automatically scales up to 45 Gbps.
- You can associate exactly one Elastic IP address with a NAT gateway. You cannot disassociate an Elastic IP address from a NAT gateway after it's created. To use a different Elastic IP address for your NAT gateway, you must create a new NAT gateway with the required address, update your route tables, and then delete the existing NAT gateway if it's no longer required.
- You cannot associate a security group with a NAT gateway. You can use security groups for your instances in the private subnets to control the traffic to and from those instances.
- You can use a network ACL to control the traffic to and from the subnet in which the NAT gateway is located. The network ACL applies to the NAT gateway's traffic. A NAT gateway uses ports 1024–65535.

[LAB] Provide Internet Access to Private Subnet Instances using NAT Gateway

- Launch a Windows EC2 instance in Private Subnet, allow RDP from VPC CIDR (not from internet)

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 the lower pricing, assign an access management role to the instance, and more.

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 your instance, add 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.

Security Group ID	Name	Description	Actions
sg-04ff0fc4796752095	aws-rds-mysql-security-group	aws-rds-mysql-security-group	Copy to new
sg-09558047168e67b1a	default	default VPC security group	Copy to new
sg-055dcbf0d47c71f88	private-windows-security-group	private-windows-security-group	Copy to new
sg-03bc17a19529cc68	public-windows-security-group	public-windows-security-group	Copy to new
sg-0348e96684b59c2a	web-servers-security-group	web-servers-security-group	Copy to new

Inbound rules for sg-055dcbf0d47c71f88 (Selected security groups: sg-055dcbf0d47c71f88)

Type	Protocol	Port Range	Source	Description
RDP	TCP	3389	172.16.0.0/16	

- RDP to the EC2 in Public Subnet and then RDP to EC2 in Private Subnet, then try to access internet. Yes, we don't have internet connectivity.

EC2 in Public Subnet

Hostname: windows-server19-01
Instance ID: i-0a7a89785fa55d872
Public IP Address: 13.126.185.9
Private IP Address: 172.16.11.224
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate

EC2 in Private Subnet

172.16.21.29 - Remote Desktop Connection

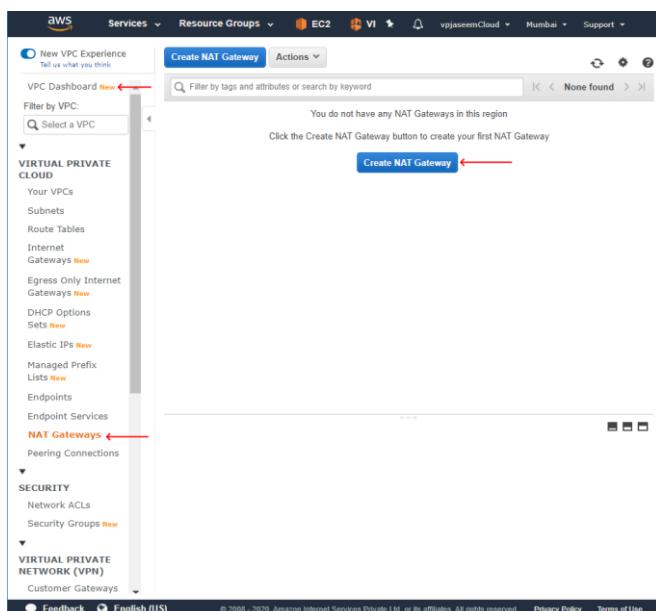
http://www.google.com/ → Can't reach this page

- Make sure the web address is correct
- Search for this site on Bing
- Refresh the page

Fix connection problems

Windows Taskbar: 5:25 PM 7/11/2020

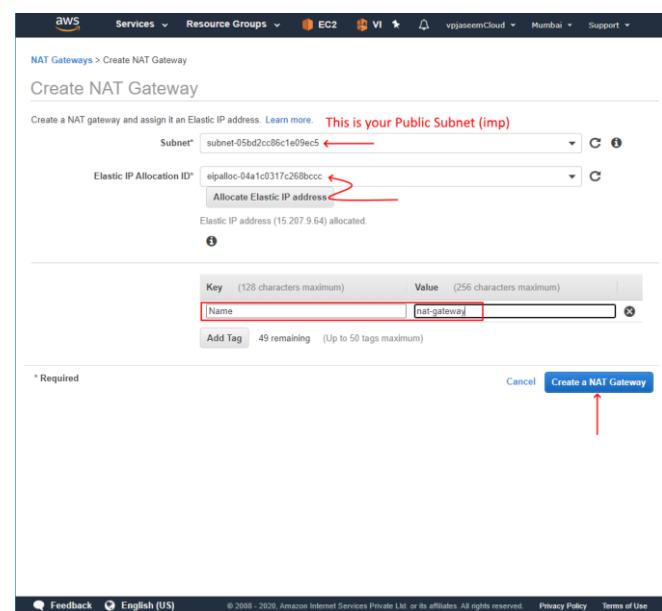
System Tray: 5:25 PM 7/11/2020



Create NAT Gateway

Click the Create NAT Gateway button to create your first NAT Gateway

Create NAT Gateway



Create a NAT gateway and assign it an Elastic IP address. Learn more. **This is your Public Subnet (imp)**

Subnet: subnet-05bd2cc06c1e09ec5

Elastic IP Allocation ID: eipalloc-04a1c0317c268bcc

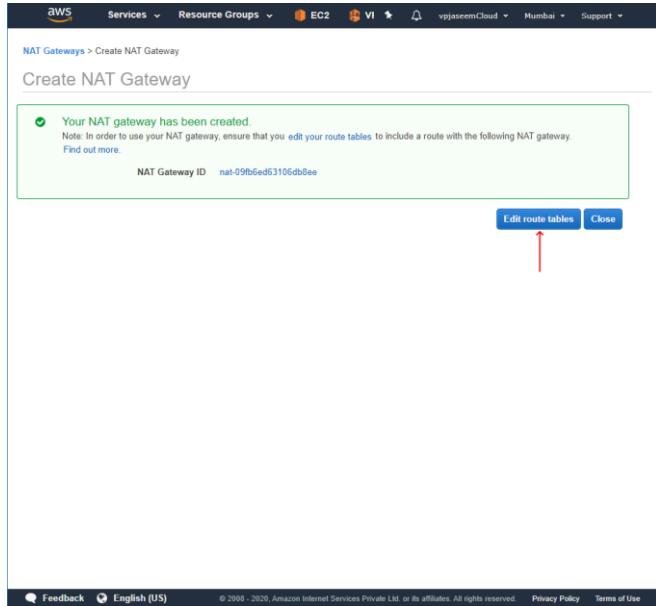
Allocate Elastic IP address

Elastic IP address (15.207.9.64) allocated.

Name: nat-gateway

Add Tag

Create a NAT Gateway

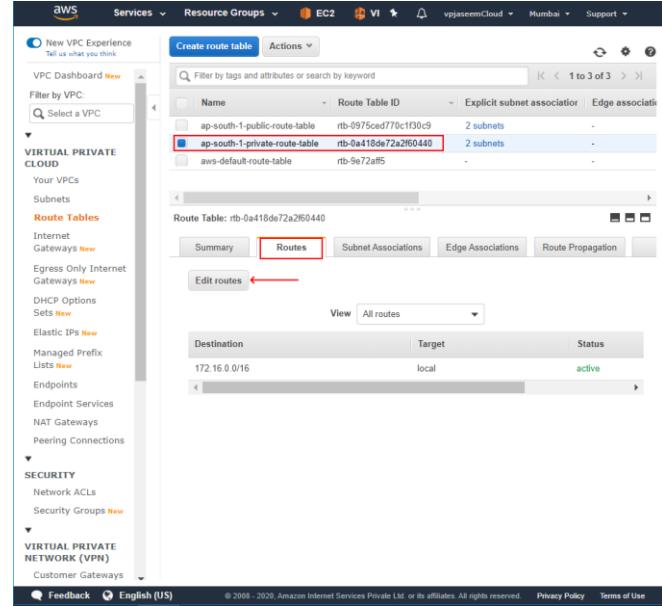


Your NAT gateway has been created.

Note: In order to use your NAT gateway, ensure that you edit your route tables to include a route with the following NAT gateway. [Find out more.](#)

NAT Gateway ID: nat-09b6ed63106db0ee

Edit route tables



Route Table: rtb-0a418de72a260440

Routes

Destination	Target	Status	Propagated
172.16.0.0/16	local	active	No
0.0.0.0/0	nat	active	No

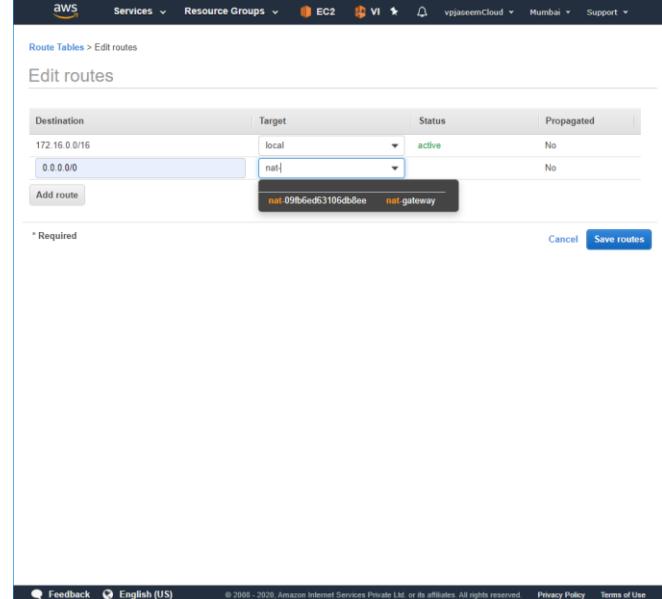
Add route

*** Required**

Destination: 0.0.0.0/0

Target: nat

Cancel Save routes



Route Table: rtb-0a418de72a260440

Routes

Destination	Target	Status	Propagated
172.16.0.0/16	local	active	No
0.0.0.0/0	nat	active	No

Add route

*** Required**

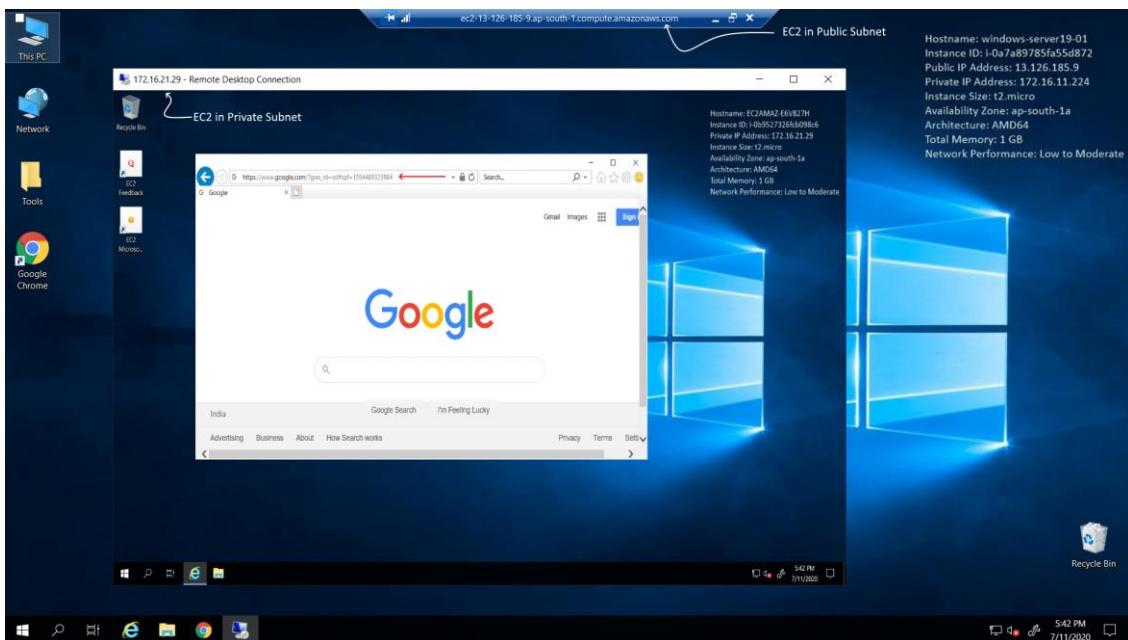
Destination: 0.0.0.0/0

Target: nat-gateway

Cancel Save routes

The screenshot shows the AWS VPC Dashboard. In the left sidebar, under 'Route Tables', 'Internet Gateways New' is selected. In the main pane, the 'Routes' tab is active for the 'ap-south-1-private-route-table'. A route entry for '0.0.0.0/0' with a target of 'nat-09fb5ed63106db8ee' is highlighted with a red box.

Note: NAT Gateways are chargeable, hence delete immediately once the lab is completed



The screenshot shows the AWS Elastic IP addresses page. Under the 'Actions' dropdown, the 'Release Elastic IP address' option is highlighted with a red box. Below the table, a summary for the IP '15.207.9.64' is shown.

Name	Allocation ID
-	epalloc-04a1c0317c26

- NAT Gateway is assigned with an Elastic IP address, when an Elastic IP is not being used, it is chargeable, hence release the Elastic IP once lab is done.
- Deleting NAT Gateway won't release elastic IP, we need to release it manually.

NAT Instance

- It is an EC2 instance, same functionality what NAT Gateway does, but is not AWS managed
- It is an Amazon customized Linux AMI with preconfigured NAT rules.
- AMIs include the string **amzn-ami-vpc-nat** in their names, so that you can identify them in the Amazon EC2 console or search for them using the AWS CLI.
- NAT instance should be launched in Public Subnet.
- Since t2.micro NAT instance is free, it is a better alternative for NAT Gateway.

[LAB] How to Deploy NAT Instance

Step 1: Choose an Amazon Machine Image (AMI)

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 (0)
My AMIs (0)
AWS Marketplace (4)
Community AMIs (40)

Operating system
Architecture

Feedback English (US) Privacy Policy Terms of Use

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 the lower pricing, assign an access management role to the instance, and more.

Number of instances 1 Launch into Auto Scaling Group

Purchasing option Request Spot instances

Network vpc-00c3b6efcf0ee5add9c1 ap-south-1-vpc
Subnet subnet-99bd2cc86c1af9e51 ap-south-1-public-sub
Auto-assign Public IP Use subnet setting (Enable)

Placement group Add instance to placement group
Capacity Reservation Open Create new Capacity Reservation

IAM role None Create new IAM role

Shutdown behavior Stop Stop - Hibernate behavior
Enable termination protection Protect against accidental termination
Monitoring Enable CloudWatch detailed monitoring Additional charges apply

Tenancy Shared - Run a shared hardware instance Additional charges may apply

File systems Add file system Create new file system

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

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 your instance, add 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: nat-instance-security-group
Description: nat-instance-security-group

Type	Protocol	Port Range	Source	Description
All traffic	All	0 - 65535	Custom	172.16.0.0/16 All traffic from VPC CIDR

Add Rule

Cancel Previous Review and Launch

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Create route table

Tell us what you think
VPC Dashboard New
Filter by VPC Select a VPC

VIRTUAL PRIVATE CLOUD Your VPCs Subnets Route Tables
Internet Gateways New Egress Only Internet Gateways New DHCP Options Sets New Elastic IPs New Managed Prefix Lists New Endpoints Endpoint Services NAT Gateways Peering Connections
SECURITY Network ACLs Security Groups New
VIRTUAL PRIVATE NETWORK (VPN) Customer Gateways

Route Table: rtb-0a418de72a260440

Summary Routes Subnet Associations Edge Associations Route Propagation
Edit routes

Destination	Target	Target	Status
172.16.0.0/16	local	local	active
0.0.0.0/0			No

View All routes

Destination Target Status
172.16.0.0/16 local active

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Edit routes

Route Tables Edit routes

Destination	Target	Status	Propagated
172.16.0.0/16	local	active	No
0.0.0.0/0		No	

Add route
* Required

Destination Target Status
172.16.0.0/16 local active
0.0.0.0/0
private-windows-server-2019
public-windows-server-2019
nat-instance
0b574c8e4f0b88d4a

Save routes

Cancel Save routes

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Launch instance

New EC2 Experience Tell us what you think
EC2 Dashboard Events Tags Limits
Instances Instances Instances Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations Images AMIs
Elastic Block Store Volumes Snapshots Lifecycle Manager
Network & Security Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

Connect Get Windows Password Create Template From Instance Launch More Like This Instance State Instance Settings Image Networking CloudWatch Monitoring Change Security Groups Attach Network Interface Detach Network Interface Disassociate Elastic IP Address Change Source/Dest. Check Manage IP Addresses

Instance: i-0b574c8e4f0b88d4a (nat-instance) Public DNS: ec2-13-232-29-47.ap-south-1.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID: i-0b574c8e4f0b88d4a Public DNS (IPv4): ec2-13-232-29-47.ap-south-1.compute.amazonaws.com
Instance state: running IPv4 Public IP: 13.232.29.47
Network Interfaces

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

EC2 Dashboard

Instances

Enable Source/Destination Check

Are you sure that you would like to disable Source/Destination Check for the instance with the following details:

Instance: i-0b574c8e4f0b88d4a (nat-instance)
Network Interface: eni-0cd94a68bc8725fd
Status: Enabled

Cancel **Yes, Disable**

Instance Details

Instance: i-0b574c8e4f0b88d4a (nat-instance) Public DNS: ec2-13-232-29-47.ap-south-1.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID: i-0b574c8e4f0b88d4a Public DNS (IPv4): ec2-13-232-29-47.ap-south-1.compute.amazonaws.com

Instance state: running IPv4 Public IP: 13.232.29.47

EC2 in Public Subnet

Hostname: windows-server-19-01
Instance ID: i-0a7a89785fa55d872
Public IP Address: 13.126.185.9
Private IP Address: 172.16.11.224
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate

EC2 in Private Subnet

This PC Network Tools Google Chrome

172.16.21.29 - Remote Desktop Connection

YouTube IN https://www.youtube.com/?g_i�&tah=w1

Recommended

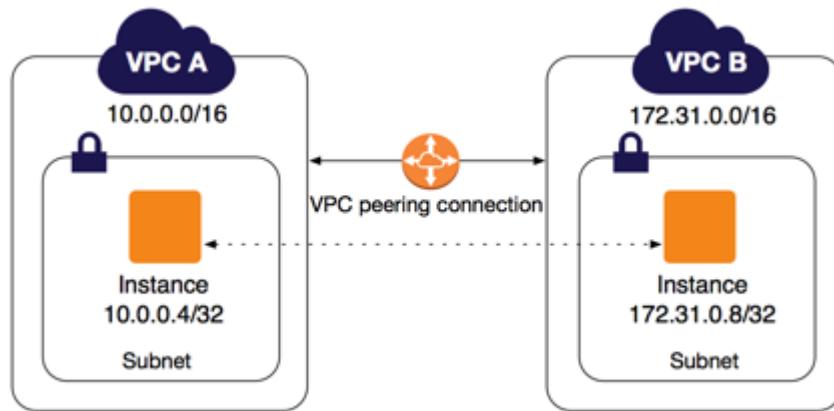
Full Song: KHAIRYAT (BONUS TRACK) CHHICHHORE | Sushant, Shraddha | Pritam, Amitabh B|Arijit Singh

South Actors On Hindi Cinema | Parvathy | Vijay Sethupathi | Vijay Deverakonda

Comedy Nights with Kapil - Deepika, SRK & Rohit Shetty - Full Episode Colors TV 6.3M views • 3 months ago

7:31PM 7/11/2020

VPC Peering



- We interconnect 2 VPCs together (e.g. when one company acquire another one)
- Without VPC, any given AWS resources need to travel though internet and talk to other AWS resource.
- Initially VPC peering works within the region but as of today we can do cross region VPC peering.
- There shouldn't be any overlapping CIDR block, no transitive peering (A peer to B and B peer to C, that doesn't mean A can talk to C)
- IGW access across VPC is not possible, no NAT routing between VPCs.

[LAB] VPC Peering Cross Region

- We already have a VPC with CIDR range 172.16.0.0/16 belongs to ap-south-1 region, go ahead and create another new VPC with CIDR 172.18.0.0/16 on another region (e.g. Singapore ap-southeast-1)

Step 1: Create a VPC in Singapore ap-southeast-1

The screenshot shows the AWS VPC dashboard. On the left sidebar, under 'Your VPCs', there is a red arrow pointing to the 'Create VPC' button. In the main pane, a modal window titled 'Create VPC' is open. Inside, the 'Name tag' field contains 'ap-southeast-1-vpc'. The 'IPv4 CIDR block' dropdown is set to '172.18.0.0/16'. A red arrow points to this dropdown. Below it, there's a radio button group for 'IPv6 CIDR block': 'No IPv6 CIDR Block' (selected), 'Amazon provided IPv6 CIDR block', and 'IPv6 CIDR owned by me'. The 'Tenancy' dropdown is set to 'Default'. At the bottom right of the modal is a 'Create' button.

The screenshot shows the AWS VPC dashboard. On the left sidebar, under 'Your VPCs', there is a red arrow pointing to the newly created VPC 'ap-southeast-1-vpc'. In the main pane, the VPC details are shown: VPC ID '0d771d2b3165b7688', State 'available', IPv4 CIDR '172.18.0.0/16', and DNS resolution 'Enabled'. A red arrow points to the 'IPv4 CIDR' field. Below the VPC details, the 'DNS hostnames' section is also highlighted with a red arrow.

Step 2: Creating 4 Subnets (2 Private and 2 Public)

The screenshot shows the AWS VPC dashboard. On the left sidebar, under 'Subnets', there is a red arrow pointing to the 'Create subnet' button. In the main pane, a modal window titled 'Create subnet' is open. Inside, the 'Name' dropdown is set to 'default-subnet-ap-southeast-1c'. The 'VPC' dropdown is set to 'ap-southeast-1-vpc'. The 'IPv4 CIDR' dropdown is set to '172.31.0.0/20'. A red arrow points to the 'IPv4 CIDR' dropdown. Below it, the 'Available IPv4 Addresses' section is highlighted with a red arrow.

- Enable Public DNS Hostnames assignments

- I have given some friendly names to the default Subnets.
- Go ahead and click Create Subnet button
- In our lab practice we are creating 4 subnets (2 Private in 2 AZs and 2 Public in 2 AZs)
 - ap-southeast-1-public-subnet-1a
 - ap-southeast-1-public-subnet-1b
 - ap-southeast-1-private-subnet-1a
 - ap-southeast-1-private-subnet-1b

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag	ap-southeast-1-public-subnet-1a
VPC*	vpc-0d771d2b3165b7688
Availability Zone	ap-southeast-1a
VPC CIDRs	CIDR Status Status Reason
	172.18.0.0/16 associated
IPv4 CIDR block*	172.18.11.0/24

* Required Cancel Create

I have configured below subnets,

SUBNET	CIDR
ap-southeast-1-public-subnet-1a	172.18.11.0/24
ap-southeast-1-public-subnet-1b	172.18.12.0/24
ap-southeast-1-private-subnet-1a	172.18.21.0/24
ap-southeast-1-private-subnet-1b	172.18.22.0/24

Step 3: Auto-assign public IPv4 address for Public Subnets

VIRTUAL PRIVATE CLOUD
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Managed Prefix Lists
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
SECURITY
Network ACLs
Security Groups

New VPC Experience Tell us what you think

VPC Dashboard New Filter by VPC: Select a VPC

Name	Subnet ID	State	VPC
ap-southeast-1-private-subnet-1b	subnet-02285251eb427da78	available	vpc-0d771d2b3165b7688
ap-southeast-1-public-subnet-1a	subnet-0307181516e8527c2a	available	vpc-0d771d2b3165b7688
ap-southeast-1-public-subnet-1b	subnet-0b293898dad213d55	available	vpc-0d771d2b3165b7688
ap-southeast-1-private-subnet-1a	subnet-0ca91c50fd7908ce	available	vpc-0d771d2b3165b7688

Feedback English (US) Privacy Policy Terms of Use

New VPC Experience Tell us what you think

VPC Dashboard New Filter by VPC: Select a VPC

VIRTUAL PRIVATE CLOUD Your VPCs Subnets Route Tables Internet Gateways Egress Only Internet Gateways DHCP Options Sets Elastic IPs Managed Prefix Lists Endpoints Endpoint Services NAT Gateways Peering Connections SECURITY Network ACLs Security Groups

Create subnet Actions

Delete subnet Create flow log Modify auto-assign IP settings Edit network ACL association Edit route table association Share subnet Add/Edit Tags

Subnet ID	VPC	IPv4 CIDR	Available IPv4 Addresses	Availability Zone	Network ACL	Auto-assign public IPv4 address	Route Table	Default subnet	Owner
subnet-03b7181516e8527c2a	vpc-0d771d2b3165b7688	172.18.11.0/24	251	ap-southeast-1a (apse1-a2z)	ac-046139659f1732058	Yes	rtb-01aea4a9f5	No	6189272327

Feedback English (US) Privacy Policy Terms of Use

Step 4: Create 2 Route Tables (Public and Private) in Singapore VPC

VIRTUAL PRIVATE CLOUD Your VPCs Subnets Route Tables Internet Gateways Egress Only Internet Gateways DHCP Options Sets Elastic IPs Managed Prefix Lists Endpoints Endpoint Services NAT Gateways Peering Connections SECURITY Network ACLs Security Groups

New VPC Experience Tell us what you think

VPC Dashboard New Filter by VPC: Select a VPC

Name	Route Table ID	Explicit subnet association	Edge association
ap-southeast-1-private-route-table	rtb-012a1ab1ed7b20bd3	subnet-0c451c5fd7908ce	-
ap-southeast-1-public-route-table	rtb-01eeaa9521e425f1	-	-
aws-default-route-table	rtb-af9ef0c9	-	-

Route Table: rtb-012a1ab1ed7b20bd3

Summary Routes Subnet Associations Edge Associations Route Propagation

Edit subnet associations

Subnet ID IPv4 CIDR IPv6 CIDR

subnet-0c451c5fd7908ce | ap-southeast-1-private-subnet-1a 172.18.21.0/24 -

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- I just interested in one Route Table (Private) because I'm going to launch my instance in Singapore Private Subnet.
- Make sure in the Private Subnet is attached to the Private Route Table.

Step 5: Launch a Windows EC2 in Private Subnet

Step 6: Create a Peering Connection from Mumbai to Singapore

The screenshot shows two related pages from the AWS VPC service.

Left Panel: Shows the 'Accept VPC Peering Connection Request' dialog. It asks if you want to accept a VPC peering connection request (pcx-0fb69306fac467d6). It displays details about the requester and accepter accounts, VPC IDs, regions, and CIDRs. The 'Yes, Accept' button is highlighted.

Right Panel: Shows the 'Create Peering Connection' page. It lists the peering connection 'peer-to-singapore-ap-southeast-1-vpc' with status 'Active'. A red arrow points from the 'Accept' button in the dialog to the 'Active' status in the list.

Step 7: Update Mumbai Public and Private Routables

The screenshot shows the 'Edit routes' dialog for the 'ap-south-1-public-route-table'.

Route Table Summary: Shows three routes: 172.16.0.0/16 to 'local' (active), 0.0.0.0/0 to 'igw-054a82ce1feb26815' (active), and 172.16.0.0/16 to 'pcx-0fb69306fac467d6' (status 'No').

Edit Routes Dialog: A new route is being added with the following details:

Destination	Target	Status
172.16.0.0/16	pcx-0fb69306fac467d6	No

The screenshot shows the 'Edit routes' dialog for the 'ap-south-1-private-route-table'.

Route Table Summary: Shows three routes: 172.16.0.0/16 to 'local' (active), 0.0.0.0/0 to 'igw-054a82ce1feb26815' (active), and 172.16.0.0/16 to 'pcx-0fb69306fac467d6' (status 'No').

Edit Routes Dialog: A new route is being added with the following details:

Destination	Target	Status
172.16.0.0/16	pcx-0fb69306fac467d6	No

Step 8: Update Singapore Private Route Table

The screenshot shows the 'Edit routes' dialog for the 'ap-southeast-1-private-route-table'.

Route Table Summary: Shows three routes: 172.16.0.0/16 to 'local' (active), 0.0.0.0/0 to 'igw-054a82ce1feb26815' (active), and 172.16.0.0/16 to 'pcx-0fb69306fac467d6' (status 'No').

Edit Routes Dialog: A new route is being added with the following details:

Destination	Target	Status
172.16.0.0/16	pcx-0fb69306fac467d6	No

- In Singapore Private Route table, add the CIDR of Mumbai VPC and point to Peering Connection.

Step 9: Access Singapore EC2 instance from Mumbai Instance

The screenshot shows the AWS EC2 Instances page. A red arrow points to the instance ID 'i-091acf62d3d94e339' in the list. Another red arrow points to the 'Private IPs' field, which contains '172.18.21.29'. The instance details panel on the right shows the following information:

Instance state	running
Instance type	t2.micro
Finding	Opt-in to AWS Compute Optimizer for recommendations.
Private DNS	ip-172-18-21-29.ap-southeast-1.compute.internal
Availability zone	ap-southeast-1a
Security groups	private-wind-security-group, view inbound rules, view outbound rules, No scheduled events

The screenshot shows a Windows desktop environment. A window titled '172.18.21.29 - Remote Desktop Connection Instance in Singapore Private Subnet' is open. The desktop background is blue. In the top right corner of the desktop, there is a status bar with the following information:

Hostname: EC2AMAZ-KNES3UT
Instance ID: i-091acf62d3d94e339
Private IP Address: 172.18.21.29
Instance Size: t2.micro
Availability Zone: ap-southeast-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate

VPC IPSec Virtual Private Network (VPN)

- Create secure private link between AWS VPC and on-prem data center.
- Access VCP resources with its Private IP address from the on-prem data center.
- AWS VPN is comprised of two services: AWS Site-to-Site VPN and AWS Client VPN. Together, they deliver a highly-available, managed, and elastic cloud VPN solution to protect your network traffic.
- Customer Gateway: Logical representation of on-premise end of VPN
- AWS VPN gateway: AWS end of VPN connection

AWS Reference: [Site to Site VPN](#)

[LAB] VPC IPSec VPN Configuration

Step 1: Create Customer gateway

The screenshot shows the AWS VPC console. In the left sidebar under 'VIRTUAL PRIVATE NETWORK (VPN)', the 'Customer Gateways' section is highlighted with a red arrow. At the top center, there is a 'Create Customer Gateway' button.

The screenshot shows the 'Create Customer Gateway' configuration dialog. It includes fields for 'Name' (set to 'bangalore-onprem-dc'), 'Routing' (set to 'Static'), 'IP Address' (set to '157.46.135.150 Public IP of on-prem Router'), 'Certificate ARN' (set to 'Select Certificate ARN'), and 'Device' (set to 'Optional'). A red arrow points from the 'Name' field back to the 'Create Customer Gateway' button at the bottom right.

Step 2: Create VPN Gateway & Attach to VPC

The screenshot shows the AWS VPC console. In the left sidebar under 'VIRTUAL PRIVATE NETWORK (VPN)', the 'Virtual Private Gateways' section is highlighted with a red arrow. At the top center, there is a 'Create Virtual Private Gateway' button.

The screenshot shows the 'Create Virtual Private Gateway' configuration dialog. It includes fields for 'Name tag' (set to 'vpn-ap-southeast-1'), 'ASN' (set to 'Amazon default ASN'), and other optional fields. A red arrow points from the 'Name tag' field back to the 'Create Virtual Private Gateway' button at the bottom right.

Virtual Private Gateways > Attach to VPC

Select the VPC to attach to the virtual private gateway.

Virtual Private Gateway ID: vgw-00bd971df92a2341e

VPC:

- vpc-05424481cbc9bcf64 ap-south-1-vpc-3
- vpc-07aa365cc4021796 ap-south-1-vpc-4
- vpc-02662665a8fc5cef ap-south-1-vpc-2
- vpc-003bfef0ee9add ap-south-1-vpc-1**
- vpc-8fd2cde7 aws-default-vpc

Yes, Attach

Step 3: Create Site to Site VPN Connection

VPN Connections > Create VPN Connection

Create VPN Connection

You do not have any VPN Connections in this region.

Click the Create VPN Connection button to create your first VPN Connection.

Name tag: vpn-connection-ap-south-1

Target Gateway Type: Virtual Private Gateway Transit Gateway

Virtual Private Gateway: vgw-00bd971df92a2341e

Customer Gateway: Existing

Customer Gateway ID: cgw-02a56e7ba2a1fc0b8

Routing Options: Dynamic (requires BGP) Static

Static IP Prefixes:

IP Prefixes	Source	State
192.168.1.0/24	-	-

Add Another Rule

The screenshot shows two side-by-side views of the AWS VPC console. The left view displays the 'Create VPN Connection' page with a search bar and a table listing a single connection named 'vpn-connect...'. The right view shows the 'Tunnel Details' tab of the same connection, displaying two tunnels (Tunnel 1 and Tunnel 2) with their respective IP ranges and statuses. Arrows point from the 'Tunnel Details' tab on the left to the 'Tunnel Details' tab on the right, and from the 'Static Routes' section on the right back to the 'Static Routes' tab on the left.

Step 4: Download the Vendor Specific Configuration

The screenshot shows the 'Download Configuration' dialog box overlaid on the VPC console. The dialog box allows selecting a vendor (Cisco Systems, Inc.), platform (Cisco ASR 1000), and software version (IOS 12.4+). It also lists 'Tunnel 2 Options' including Phase 1 and Phase 2 encryption algorithms, DH group numbers, and IKE parameters. A red arrow points from the 'Download' button in the dialog box to the 'Download Configuration' button in the main VPC console interface.

The screenshot shows a Notepad++ window with the file path 'C:\Users\abvp\Downloads\vpn-0e2d50064d07100f3.txt'. The file contains a Cisco IOS configuration script for an ISAKMP profile. Key parameters visible include:

```

40    encryption aes 128
41    authentication pre-share
42    group 2
43    lifetime 28800
44    hash sha
45    exit
46
47 ! The ISAKMP keyring stores the Pre Shared Key used to authenticate the
48 ! tunnel endpoints.
49 !
50 crypto keyring keyring-vpn-0e2d50064d07100f3-0
51 local-address 157.46.135.150
52 pre-shared-key address 3.7.34.246 key kk86J_v_oMtJtsNA_Oj.RQPYNm85cmJF
53 exit
54
55 ! An ISAKMP profile is used to associate the keyring with the particular
56 ! endpoint.
57 !
58 crypto isakmp profile isakmp-vpn-0e2d50064d07100f3-0
59 local-address 157.46.135.150
60 match identity address 3.7.34.246
61 keyring keyring-vpn-0e2d50064d07100f3-0
62 exit
63
64 ! #2: IPSec Configuration
65 !
66 ! The IPSec transform set defines the encryption, authentication, and IPSec
67 ! mode parameters.
68 ! Category "VPN" connections in the GovCloud region have a minimum requirement of AES12:
69 ! Please note, you may use these additionally supported IPSec parameters for encryption.
70 ! NOTE: If you customized tunnel options when creating or modifying your VPN connection.
71 !
72 ! Higher parameters are only available for VPNs of category "VPN," and not for "VPN-Cla:
73 !

```

Norm length: 13,690 lines: 313 Ln:192 Col:20 Sel:0|0 Unix (LF) UTF-8 INS

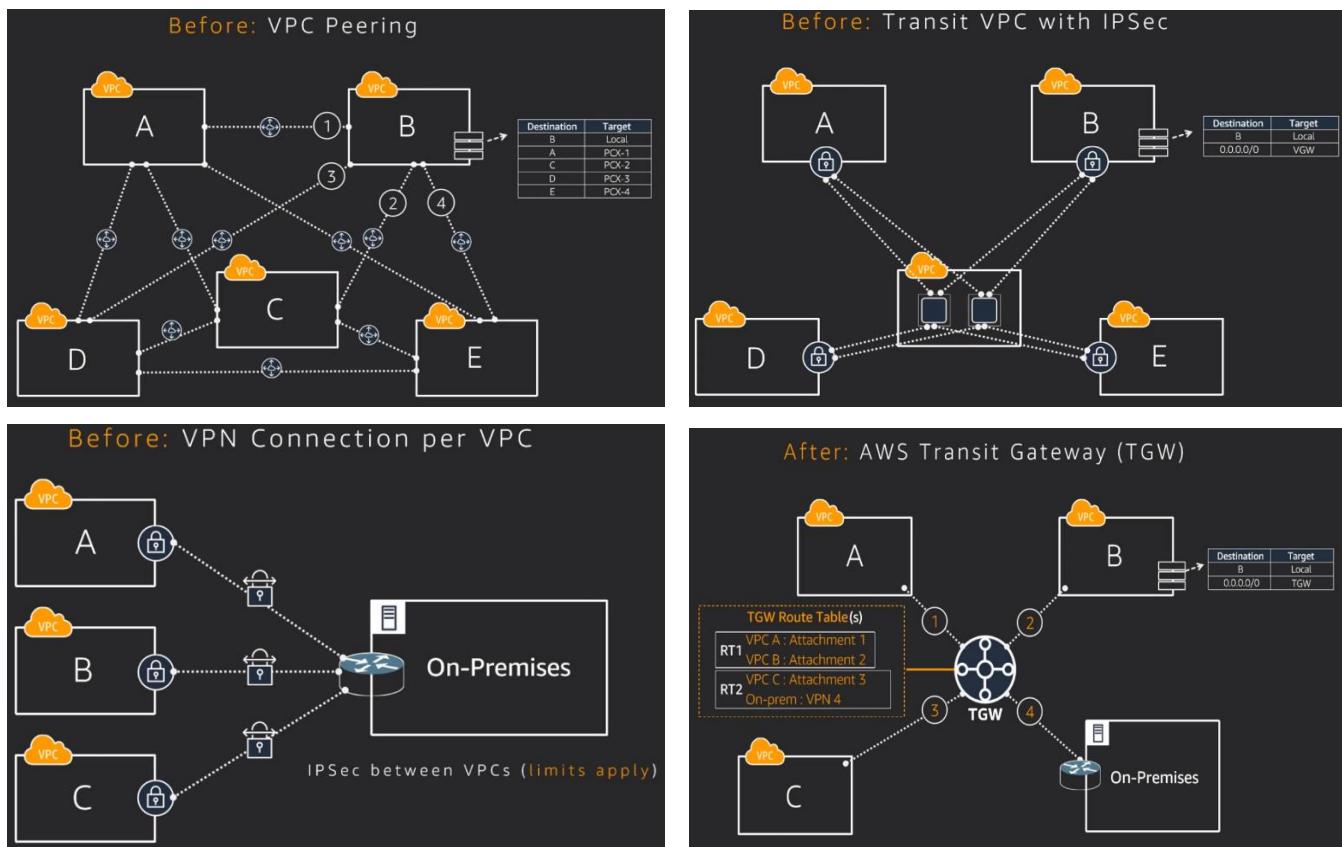
VPC Direct Connect

- Dedicated network connection from on-prem DC to AWS, better speed, and bandwidth than VPN.
- Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

AWS Reference: [AWS Direct Connect](#)

AWS Transit Gateway (TGW)

- A Transit Gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same account or across accounts.
- When we have multiple VPCs, peering becomes challenging, we end up building a complete peer full mesh.
- TGW scope is Region
- Consider we have n VPCs, then, Total number of Peering connections = $\frac{n(n-1)}{2}$
- Limit for Static Routes per VPC route table = 100; and Peering connection per VPC = 125



- Transit gateway works as Hub and Spoke model, all the VPCs are attached to the Transit Gateway
- On-Prem DC can also be attached to Transit Gateway
- We can define rules Transit Gateway regarding the routes
- **Attachment**: Connection between VPC and Transit Gateway (VPN to TGW). Attach one Subnet per AZ in VPC
- **Association**: Route table used to route traffic coming from an attachment. One attachment uses one Route table.
- **Propagation**: Route table where the attachment's routes are installed.

[LAB] Transit Gateway

- Let's create 4 VPCs in same region (ap-south-1), one has Public and Private Subnets, the others have only 1 private subnet.
- Assuming we have 1 VPC already and a Windows EC2 instance running on it with Public IP.

Step 1: Create 3 VPCs, 3 Subnets, edit 3 default route table

The screenshot shows the AWS VPC Dashboard. On the left, under 'Your VPCs', four VPCs are listed: ap-south-1-vpc-1, ap-south-1-vpc-2, ap-south-1-vpc-3, and ap-south-1-vpc-4. Each VPC has multiple subnets associated with it. On the right, a separate window shows the 'Create subnet' screen where three new subnets are being created for VPC ap-south-1-vpc-1, with the message 'Newly created' at the bottom.

VPC Name	VPC CIDR	Subnet CIDR	Name of Subnet	Route Table
ap-south-1-vpc-1	172.16.0.0/16	172.16.11.0/24	ap-south-1-public-subnet-1a	ap-south-1-public-route-table
		172.16.12.0/24	ap-south-1-public-subnet-1b	
		172.16.21.0/24	ap-south-1-private-subnet-1a	
		172.16.22.0/24	ap-south-1-private-subnet-1b	ap-south-1-private-route-table
ap-south-1-vpc-2	172.22.0.0/16	172.22.21.0/24	vpc2-ap-south-1-private-subnet-1a	vpc2-ap-south-1-private-rt
ap-south-1-vpc-3	172.23.0.0/16	172.23.21.0/24	vpc3-ap-south-1-private-subnet-1a	vpc3-ap-south-1-private-rt
ap-south-1-vpc-4	172.24.0.0/16	172.24.21.0/24	vpc4-ap-south-1-private-subnet-1a	vpc4-ap-south-1-private-rt

The screenshot shows the AWS Route Tables page. It lists five route tables: ap-south-1-private-route-table, ap-south-1-public-route-table, vpc2-ap-south-1-private-rt, vpc3-ap-south-1-private-rt, and vpc4-ap-south-1-private-rt. Each route table is associated with two subnets. A red box highlights the newly created route tables, and a message 'Newly created Route Tables' is displayed at the bottom.

Step 2: Launch Amazon Linux EC2 on 3 VPC's Private Subnets

- Deploy Amazon Linux on each of the 3 VPCs, make sure to allow all traffic in Security Group

VPC Name	EC2 Instance Name
ap-south-1-vpc-2	aws-linux-ap-south-1-vpc-2
ap-south-1-vpc-3	aws-linux-ap-south-1-vpc-3
ap-south-1-vpc-4	aws-linux-ap-south-1-vpc-4

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Events, Tags, Limits, Instances, Images, AMIs, Elastic Block Store, Network & Security, and more. The main area displays a table of running instances. One instance, 'windows-with-python', is highlighted with a red box. The table columns include Name, Instance ID, Instance Type, Availability Zone, and Instance State. Below the table, there's a detailed view of one instance, showing its configuration: instance state (running), instance type (t2.micro), private DNS (ip-172-22-21-252.ap-south-1.compute.internal), private IPs (172.22.21.252), security groups (default, view inbound rules, view outbound rules), and scheduled events (No scheduled events). At the bottom, it shows the VPC ID (vpc-02662665a8fc5caf) and AMI ID (amzn2-ami-hvm-2.0.20200722.0-vml.4dvv7/ami).

Step 3: Creating Transit Gateway

The screenshot shows the AWS VPC Services page. The sidebar includes sections for New VPC Experience, Direct Connect Services, NAT Gateways, Peering Connections, SECURITY (Network ACLs, Security Groups), and VIRTUAL PRIVATE NETWORK (VPN) (Customer Gateways, Virtual Private Gateways, Site-to-Site VPN Connections, Client VPN Endpoints). A red arrow points to the 'TRANSIT GATEWAYS' section. Under TRANSIT GATEWAYS, there are links for Transit Gateway Attachments, Transit Gateway Route Tables, and Network Manager.

The screenshot shows the 'Create Transit Gateway' wizard. It has two tabs: 'Create Transit Gateway' (selected) and 'Configure sharing options for cross account'. The 'Create Transit Gateway' tab contains fields for 'Name tag' (transit-gateway-ap-south-1) and 'Description' (transit-gateway-ap-south-1). Below these are configuration options: 'Amazon side ASN' (64512), 'DNS support' (enable), 'VPN ECMP support' (enable), 'Default route table association' (enable), and 'Default route table propagation' (enable). The 'Configure sharing options for cross account' tab has a single option 'Auto accept shared attachments' (enable). At the bottom right, there are 'Cancel' and 'Create Transit Gateway' buttons, with the latter being highlighted by a red arrow.

Step 4: Attaching VPCs to the Transit Gateway

Create Transit Gateway Attachment

Select a Transit Gateway and the type of attachment you would like to create.

Transit Gateway ID* **tgw-03753d901cdc2c250**

Attachment type VPC VPN Peering Connection

VPC Attachment

Attachment name tag **attach-ap-south-1-vpc1**

DNS support enable

IPv6 support enable

VPC ID* **vpc-00c3b6efcf0ee9add**

Subnet IDs* **subnet-0483584b239c2f5c** **subnet-0acb68796db03fe43**

Availability Zone Subnet ID

- ap-south-1a **subnet-9183584b239c2f5c (ap-south-1-private-subnet-1a)**
- ap-south-1b **subnet-0acb68796db03fe43 (ap-south-1-private-subnet-1b)**
- ap-south-1c No subnet available

* Required

Create attachment

Name	Transit Gateway attachment ID	Transit Gateway ID	Resource type
attach-ap-south-1-vpc1	tgw-attach-092351175035931	tgw-03753d901cdc2c250	VPC
attach-ap-south-1-vpc2	tgw-attach-0a4646bb0203d9f1ee	tgw-03753d901cdc2c250	VPC
attach-ap-south-1-vpc3	tgw-attach-07feef9dd614069	tgw-03753d901cdc2c250	VPC
attach-ap-south-1-vpc4	tgw-attach-0aaecc02f09b0956c	tgw-03753d901cdc2c250	VPC

Create Transit Gateway Route Table

Transit Gateway route table ID **tgw-rtb-0ba356f38052feada**

Transit Gateway ID **tgw-03753d901cdc2c250**

State **available**

Create association

Attachment ID	Resource type	Resource ID
tgw-attach-0a4646bb0203d9f1ee	VPC	vpc-02662665a8fc5cef
tgw-attach-07feef9dd614069	VPC	vpc-05424481cbc9f964
tgw-attach-092351175035931	VPC	vpc-00c3b6efcf0ee9add
tgw-attach-0aaecc02f09b0956c	VPC	vpc-07faa365cc402179b

Name	Transit Gateway route table ID	Transit Gateway ID	State	Default ass.
tgw-rtb-0ba356f38052feada	tgw-03753d901cdc2c250	tgw-03753d901cdc2c250	available	Yes

Create Transit Gateway Route Table

Transit Gateway route table ID **tgw-rtb-0ba356f38052feada**

Transit Gateway ID **tgw-03753d901cdc2c250**

State **available**

Create static route

CIDR	Attachment
172.16.0.0/16	tgw-attach-092351175035931 vpc-00c3b6efcf0ee9add
172.22.0.0/16	tgw-attach-0a4646bb0203d9f1ee vpc-02662665a8fc5cef
172.23.0.0/16	tgw-attach-07feef9dd614069 vpc-05424481cbc9f964
172.24.0.0/16	tgw-attach-0aaecc02f09b0956c vpc-07faa365cc402179b

Step 5: Update Each VPC Route Table with Transit Gateway

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under 'Your VPCs', there are links for Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options, Sets, Elastic IPs, Managed Prefix Lists, Endpoints, Endpoint Services, NAT Gateways, and Peering Connections. The main pane displays four VPCs:

Name	VPC ID	State	IPv4 CIDR
ap-south-1-vpc-1	vpc-00c3b6efcf0ee9add	available	172.16.0.0/16
ap-south-1-vpc-2	vpc-02662665a8fc5cef	available	172.22.0.0/16
ap-south-1-vpc-3	vpc-0542f4481cbc9f464	available	172.23.0.0/16
ap-south-1-vpc-4	vpc-07faa365cc402179b	available	172.24.0.0/16

The screenshot shows the AWS Route Tables page. Under 'Route Tables', it lists several entries, with 'ap-south-1-public-route-table' highlighted with a red arrow. Below this, the 'Edit routes' section is shown for the 'Route Table: rb-0975cad770c1f30c9'. The 'Routes' tab is selected, and the 'Edit routes' button is highlighted with a red arrow. The table shows routes for various destinations:

Destination	Target	Status
172.16.0.0/16	local	active
pl-78a54011 (com.amazonaws.ap-south-1.s3)	vpc-e0391ef9f56b717511	active
52.219.62.0/23, 3.5.212.0/23, 3.5.208.0/22,		
52.219.64.0/22		
0.0.0.0/0	igw-054a02ce1fb20615	active
172.18.0.0/16	pxc-0ff69306fac467d6	active
172.0.0.0/8	tgw-03753d961cdc2c250 transit-gateway-ap-south-1	active

- My VPC CIDR blocks are 172.16.0.0/16, 172.22.0.0/16, 172.23.0.0/16, 172.24.0.0/16
- To match all these CIDR, let me select something like this, 172.0.0.0/8

The screenshot shows the 'Edit routes' section for the same route table. A new route is being added for '172.0.0.0/8' with a target of 'tgw-03753d961cdc2c250 transit-gateway-ap-south-1'. The 'Save routes' button is highlighted with a red arrow. A note at the bottom right says 'Note: Must do this on all Route Tables'.

Step 6: Login to Public Windows EC2 and Access other Linux EC2s

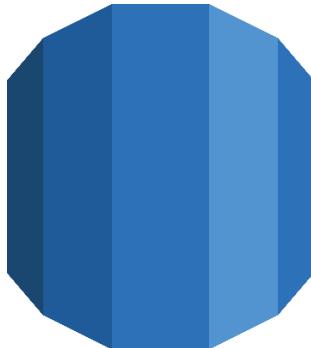
The screenshot shows four EC2 instances connected via the AWS Management Console:

- EC2 in ap-south-1-vpc-1 (Public)**: A Windows instance.
- EC2 in ap-south-1-vpc-2**: An Amazon Linux 2 AMI instance.
- EC2 in ap-south-1-vpc-3**: An Amazon Linux 2 AMI instance.
- EC2 in ap-south-1-vpc-4**: An Amazon Linux 2 AMI instance.

Each instance has a terminal window showing ping statistics between the others. For example, the Windows instance is pinging the Linux instances, and vice versa. The ping results show low latency and high throughput.



AWS Relational Database Service



- RDS is the way to store structured, queriable, indexable data inside the cloud
- Unstructured Data stored in EBS or S3 whereas Structured Data stored in databases (Redshift, RDS, Dynamo DB) SQL Style
- These services are called DBaaS (Database As A Service)
- We can choose any Database engine from PostgreSQL, MySQL, ORACLE, Microsoft SQL, MariaDB, Amazon Aurora)
- We can resize any time if required
- Auto backup is enabled by default, 0 to 35 days data backup and point in time recovery.
- Geo replicated
- Also have manual database snapshot to S3 feature available
- RDS Encryptions (Encryption at rest), uses AWS KMS (Key Management Store)
- Also, DB engine will have its own encryption.

[LAB] RDS Configuration

AWS Services Resource Groups Support

VPC > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info ←

Name cannot be edited after creation.

Description Info

VPC Info ↓

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
MySQL/Aurora	TCP	3306	Custom ←	Allow access to AWS RDS from VPC EC2s ←
				<input type="text" value="172.16.0.0/16"/> ←

Add rule

Outbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Destination <small>Info</small>	Description - optional <small>Info</small>
All traffic	All	All	Custom ←	<input type="text" value="0.0.0.0/0"/> ←

Add rule

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag

You can add up to 50 more tag

↓ Create security group

AWS Services Resource Groups EC2 Mumbai

RDS EFS Governance Security, Identity, Compliance

EC2 FSx CloudWatch IAM Resource Access Mgt

IAM S3 Glacier AWS Auto Scaling Cognito Secrets Manager

S3 Storage Gateway CloudFormation CloudTrail GuardDuty

VPC AWS Backup CloudWatch Metrics Inspector Amazon Macie

Database DynamoDB AWS AppConfig AWS Single Sign-On Certificate Manager

RDS ElastiCache Neptune Trusted Advisor Key Management Service

Amazon Redshift Amazon QLDB AWS Well-Architected Tool CloudHSM Directory Service

Amazon DocumentDB Amazon Keyspaces Personal Health Dashboard WAF & Shield AWS Firewall Manager

Migration & Transfer AWS Migration Hub AWS Compute Optimizer Artifact

Application Discovery Service Database Migration Service Server Migration Service AWS Lambda Security Hub Detective

Media Services Elastic Transcoder Kinesis Video Streams Mobile AWS Amplify

Amazon RDS

Dashboard Databases Query Editor Performance Insights Snapshots Automated backups Reserved instances Proxies

Create DB Subnet Group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name Info ←

Description Info

VPC Info ←

Add subnets

Availability Zones Choose the Availability Zones that include the subnets you want to add.

Subnets Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

↓ Feedback English (US) Privacy Policy Terms of Use

Amazon RDS

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose different VPC identifier after your subnet group has been created.
ap-south-1-vpc (vpc-00c3b6efcf0ee9add)

Add subnets
Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
ap-south-1a ap-south-1b

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-0f483584b239c2f5c (172.16.21.0/24)
subnet-0acb68796db03fe43 (172.16.22.0/24)

Subnets selected (2)
Availability zone Subnet ID CIDR block
ap-south-1a subnet-0f483584b239c2f5c 172.16.21.0/24
ap-south-1b subnet-0acb68796db03fe43 172.16.22.0/24

Create

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Amazon RDS

Databases
Create database

Databases
Group resources Modify Actions Restore from S3
Create database

DB identifier

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

RDS > Create Database

Choose a database creation method info
Standard Create You set all of the configuration options, including ones for availability, security, backups, and maintenance.
Easy Create Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options
Engine type Info
Amazon Aurora MySQL MariaDB
PostgreSQL Oracle Microsoft SQL Server
Edition MySQL Community
Version Info MySQL 8.0.16

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

dWS

Templates
Choose a sample template to meet your use case.
Production Dev/Test Free tier
The Free tier is intended for development use outside of a production environment.

Settings
DB instance identifier info Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
aws-rds-mysql-database-1
The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings
Master username info Type a login ID for the master user of your DB instance.
admin
1 to 16 alphanumeric characters. First character must be a letter
Auto generate a password Amazon RDS can generate a password for you, or you can specify your own password
Master password info Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), " (double quote) and @ (at sign).
Confirm password info Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), " (double quote) and @ (at sign).

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws Services Resource Groups EC2 VPC S3 vjaseemCloud Mumbai Support

DB instance size
DB instance class info Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.
Standard classes (includes m classes)
Memory Optimized classes (includes r and x classes)
Burstable classes (includes t classes)
db.t2.micro 1 vCPUs 1 GiB RAM Not EBS Optimized
Include previous generation classes

Storage
Storage type info General Purpose (SSD)
Allocated storage 20 GiB
(Minimum: 20 GiB, Maximum: 16384 GiB) Higher allocated storage may improve IOPS performance.

Storage autoscaling info Provides dynamic scaling support for your database's storage based on your application's needs.
Enable storage autoscaling Enabling this feature will allow the storage to increase once the specified threshold is exceeded.
Maximum storage threshold info Charges will apply when your database autoscales to the specified threshold
1000 GiB
Minimum: 21 GiB, Maximum: 16384 GiB

Availability & durability
Amazon RDS automatically creates a backup of your database every day.

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

aws Services Resource Groups EC2 VPC S3 vjaseemCloud Mumbai Support

Connectivity
Virtual private cloud (VPC) info VPC that defines the virtual networking environment for this DB instance.
ap-south-1-vpc (vpc-00c3b6efcf0ee9add)
Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

Additional connectivity configuration
Subnet group info DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.
aws-rds-db-subnet-group
Publicly accessible info Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.
No ROS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)
Choose existing Choose existing VPC security groups
Existing VPC security groups
aws-rds-mysql-security-group
Create new Create new VPC security group
Allow inbound to TCP port 3306
Availability Zone info ap-south-1a
Database port info TCP/IP port that the database will use for application connections.
3306

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Database authentication

Database authentication options: [Info](#)

- Password authentication: Authenticates using database passwords.
- AWS IAM database authentication**: Authenticates using the database password and user credentials through AWS IAM users and roles.
- Password and Kerberos authentication**: Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

Additional configuration

Database options, backup enabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

Learn more about AWS Free Tier. [\[?\]](#)

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the Amazon RDS Pricing page. [\[?\]](#)

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

[Cancel](#) [Create database](#)

[Feedback](#) [English \(US\)](#)

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

This PC

Network

Tools

Google Chrome

MySQL Workbench

File Edit View Database Server Tools Scripting Help

Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Name or IP address of the server host - and TCP/IP port.

Port:

Username: Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

Default Schema:

Test Connection Cancel [OK](#)

This means that MySQL is not installed or is not running. [Rescan servers](#)

13.126.185.9

Hostname: windows-server19-01
Instance ID: i-0a7a89785fa55d872
Public IP Address: 13.126.185.9
Private IP Address: 172.16.11.224
Availability Zone: ap-south-1a
Architecture: AMD64
Virtual Memory: 1 GB
Network Performance: Low to Moderate

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SQL File 1* sample_table - Table

Table Name: Schema: sample_db

Charset/Collation: utf8mb4 Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AZ	G	Default Expression
employeeid	INT(11)									
name	VARCHAR(45)									
age	INT(11)									

Administration Schemas Information

Table: sample_table

Columns: employeeid int(11) PK, name varchar(45), age int(11)

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SQL File 1* sample_table - Table

Table Name: Schema: sample_db

Charset/Collation: utf8mb4 Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AZ	G	Default Expression
employeeid	INT(11)									
name	VARCHAR(45)									
age	INT(11)									

Administration Schemas Information

Table: sample_table

Columns: employeeid int(11) PK, name varchar(45), age int(11)

Result Grid | Filter Rows:
employeeid name age
101 Abdul 29
102 Arav 26
103 null null

Output

13.126.185.9

7:41 PM 7/6/2020

DynamoDB (AWS NoSQL)



- NoSQL database getting popular day by day.

	SQL	NoSQL
Data Storage	Rows and Columns	Key-Value
Schemas	Fixed	Dynamic
Querying	Using SQL	Focused on collection of documents
Scalability	Vertical (Add more CPU/ RMA)	Horizontal (Add more machines)

- Top level Organized into TABLES
- Table contain ITEMS
- ITEMS contain ATTRIBUTES (Key-value pair)
- Every TABLE requires a PRIMARY KEY
- Optionally include a SORT KEY and up to five LOCAL SECONDARY INDEX

 **Table: Employee**

Item:	Item:
EmpID: 0913	EmpID: 1498
First: Ben	Name: Amy Smith
Last: Finkel	Age: 38

- Charged for Read and Write capacity throughput.
- Performing operations against AWS DynamoDB is either Management Console or REST API or Program Codes or SDKs.
- On demand auto scaling based on the application requirement

[LAB] Dynamo DB Configurations

DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB allows you to create a database table that can store and retrieve any amount of data, and serve any level of request traffic. [More info](#)

Create table

aws-dynamodb-table1

Items

Scan: [Table] aws-dynamodb-table1: employeeid ▾ Viewing 0 to 0 items

An item consists of one or more attributes. Each attribute consists of a name, a data type, and a value. When you read or write an item, the only attributes that are required are those that make up the primary key. [More info](#)

Create item

Tree ▾ Item {3} employeeId Number : 101 name String : Abdul age Number : 29

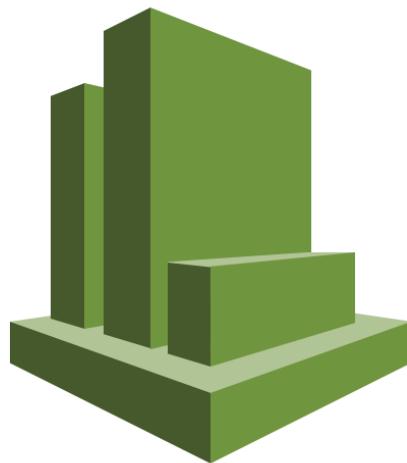
Save

aws-dynamodb-table1

Scan: [Table] aws-dynamodb-table1: employeeid ▾ Viewing 1 to 1 items

employeeid	age	name
101	29	Abdul

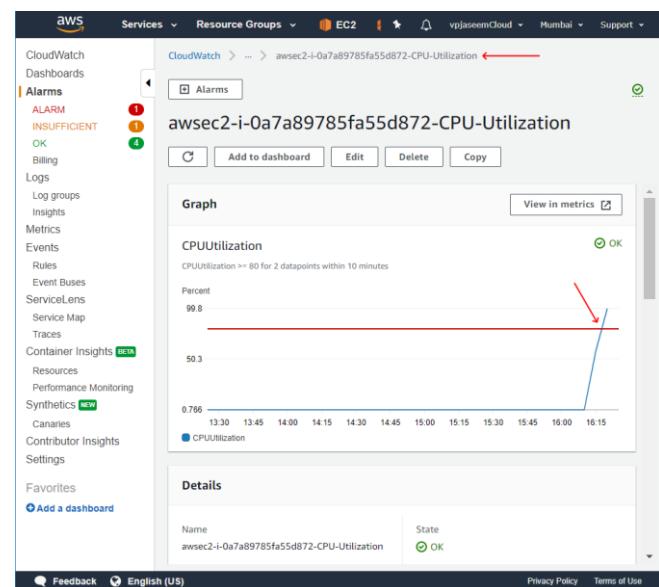
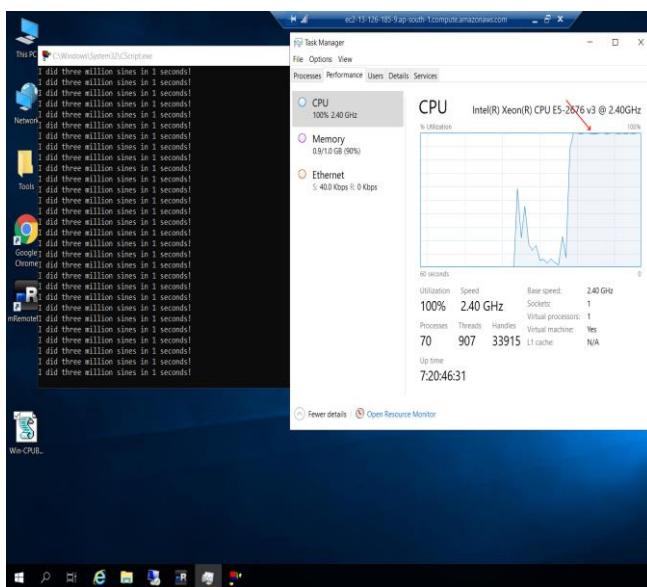
Amazon CloudWatch



- CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers.
- By default, basic metrics are monitored every 5 minutes
- Detailed monitoring is chargeable, and the sampling period will be 1 minute.

[LAB] Configure EC2 Alarm for CPU Utilization

Download [CPU Busy Script](#), Right click and Run, do not double click.





Instances | EC2 Management Con... CloudWatch Management Con... + ap-south-1.console.aws.amazon.com/cloudwatch/home?region=...

CloudWatch Dashboards Alarms INSUFFICIENT OK Billing Logs Log groups Insights Metrics Events Rules Event Buses ServiceLens Service Map Traces Container Insights Resources Performance Monitoring Synthetics Canaries Contributor Insights Settings Favorites Add a dashboard

Feedback English (US) © 2008 - 2020 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

CloudWatch > Alarms > awsec2-i-0a7a89785fa55d872-CPU-Utilization

awsec2-i-0a7a89785fa55d872-CPU-Utilization

Graph View in metrics

CPUUtilization
CPUUtilization >= 80 for 2 datapoints within 10 minutes

Percent: 99.8 (In alarm)
50.3
0.766

Time: 13:30, 13:45, 14:00, 14:15, 14:30, 14:45, 15:00, 15:15, 15:30, 15:45, 16:00, 16:15, 16:30

Details

Name: awsec2-i-0a7a89785fa55d872-CPU-Utilization	State: In alarm	Namespace: AWS/EC2	Data points to alarm: 2 out of 2
Type: Metric alarm	Threshold: CPUUtilization >= 80 for 2 datapoints within 10 minutes	Metric name: CPUUtilization	Treatment: Missing data treatment
		Instanced	Treat missing data as missing

Restarting

Recycle Bin

10:01 PM 14-Jul-20

CloudWatch Dashboards Alarms INSUFFICIENT OK Billing Logs Log groups Insights Metrics Events Rules Event Buses ServiceLens Service Map Traces Container Insights Resources Performance Monitoring Synthetics Canaries Contributor Insights Settings Favorites Add a dashboard

Feedback English (US) © 2008 - 2020 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

CloudWatch > Alarms

Alarms (6)

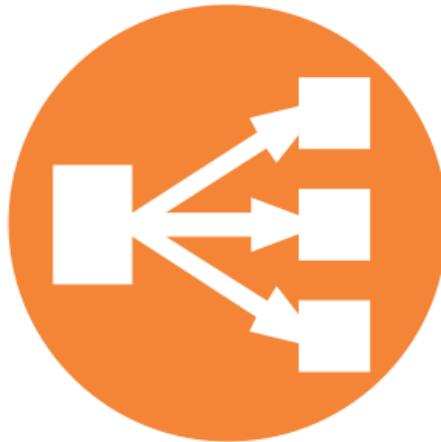
Hide Auto Scaling alarms Clear selection Create composite alarm Actions

Create alarm

Search: In alarm Any type

Name	State	Last state update	Conditions
awsec2-i-0a7a89785fa55d872-CPU-Utilization	In alarm	2020-07-14 22:01:31	CPUUtilization >= 80 for 2 datapoints within 10 minutes
TargetTracking window-autoscaling-group-Alarm	In alarm	2020-07-14 01:57:28	CPUUtilization <= 56 for 15 datapoints within 15 minutes
66e5e708-8997-e4b5-be7c-ebab9b98e567			

Elastic Load Balances (ELB)



- ELB can be Public or Private (depends on the subnets you choose)
- It is managed by AWS (we don't need to worry about scalability or availability)
- Used to distribute incoming traffic to multiple instances those are registered to it and does the health checks of instances.
- If an instance is unhealthy, ELB stops sending traffic to that.
- We can move our Web Servers to Private subnet, only keep ELB in public and send the traffic to private instance.
- Traffic is distributed between 2 AZs in round-robin fashion
- Traffic is distributed within AZs based on the instance having least number of connections.
- Recommended to access ELB via DNS name only not based on IP

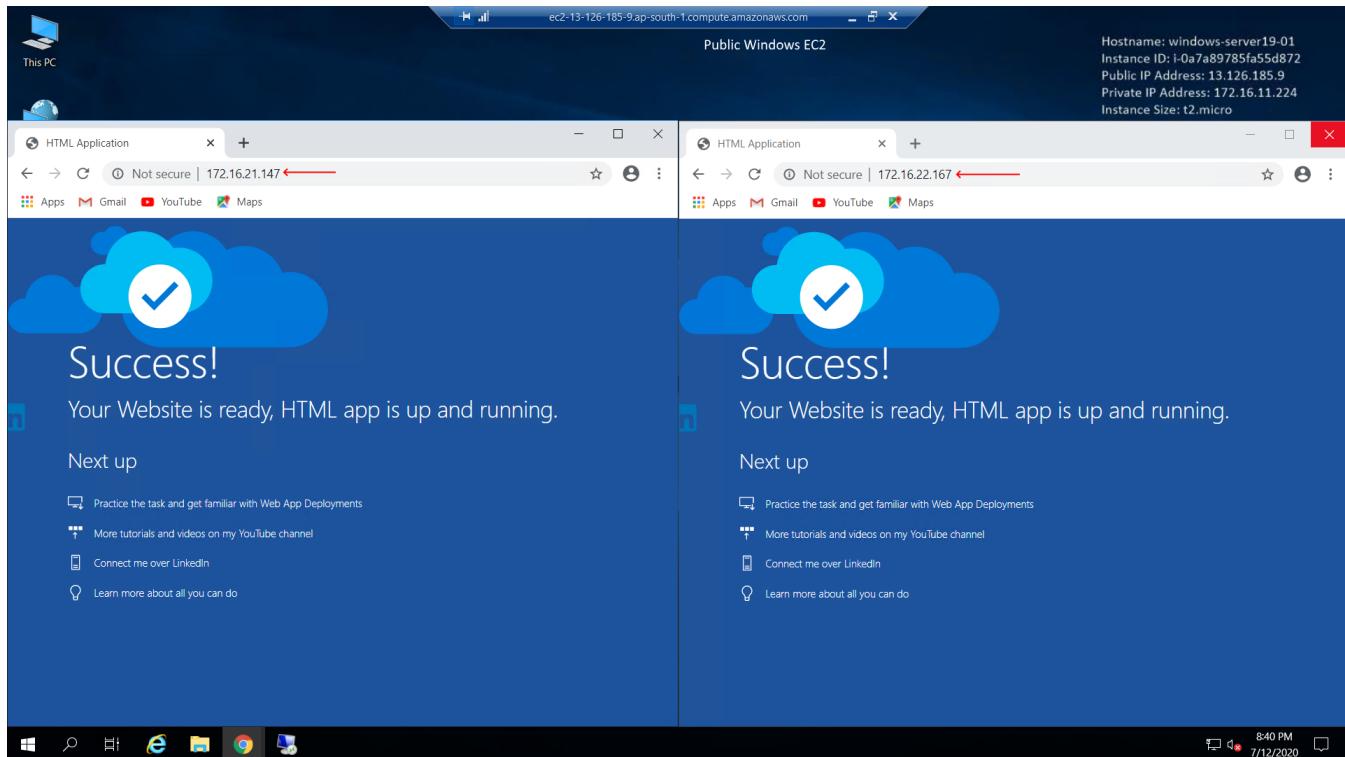
[LAB] Elastic Load Balances

- Deploy 2 web servers (either Windows or Linux) in private subnet. Use the launch templates with User Data to do so.
- In this example, I have deployed 2 Windows IIS Web Servers running same copy of website files.

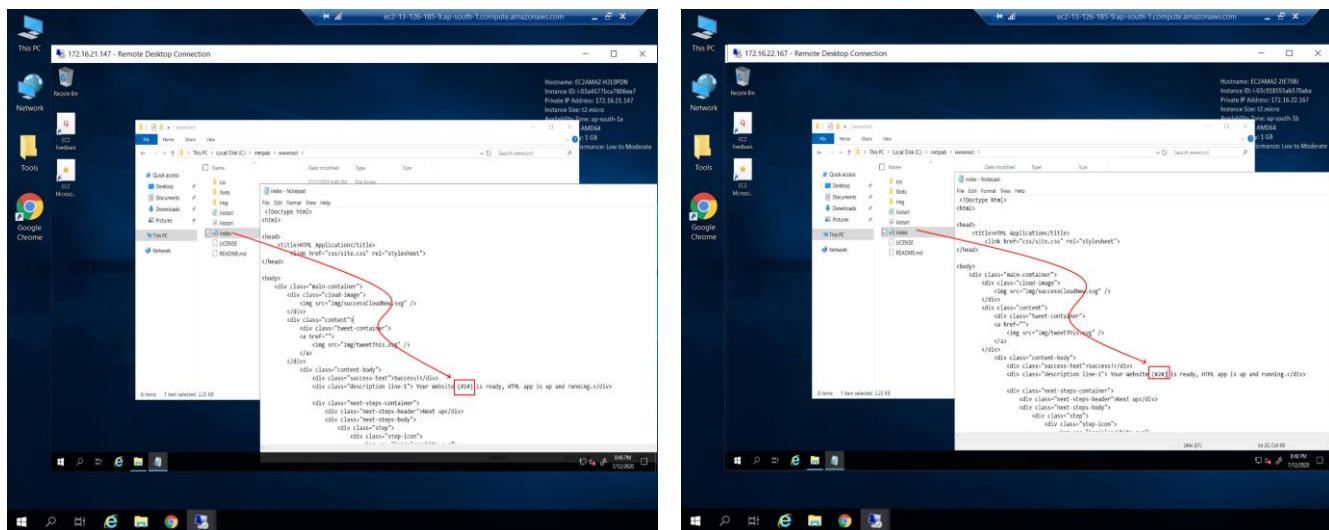
The screenshot shows the AWS EC2 Launch Templates page. On the left, there's a sidebar with various navigation options like EC2 Dashboard, Instances, Launch Templates, Images, etc. The main area shows a list of launch templates. Two templates are listed: 'lt-007e5174948bb3fa2' and 'lt-081bfacd8a2ff1805'. The second template is selected. A red box highlights the 'Actions' dropdown menu, and another red box highlights the 'Launch instance from template' option. A red arrow points from the 'Actions' menu to the 'Launch instance from template' option.

The screenshot shows the AWS EC2 Network settings page for two subnets: 'ap-south-1-private-subnet-1a' and 'ap-south-1-private-subnet-1b'. Both subnets are associated with the 'web-servers-security-group' sg-0348e96684b59cf2a. A red box highlights the security group selection for both subnets.

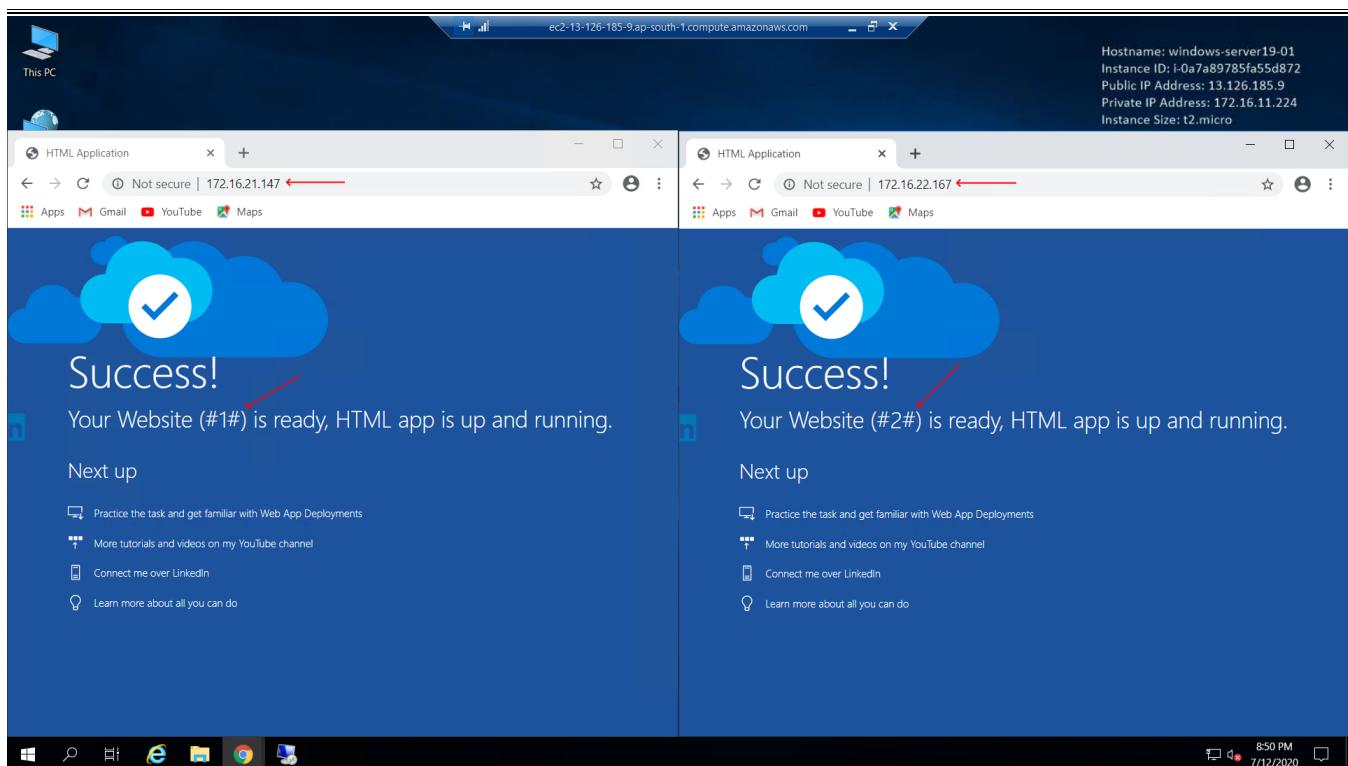
- These 2 websites are not accessible from internet since they are deployed in Private Subnet. Use Public Windows Server (Jump Host) to access these two servers with local IP.



- Let's modify these sites a bit so that we can identify which is which. Go to the C:\inetpub\wwwroot and add some characters in the index.html file.



- Now we have a small difference in the Websites, this is just for our understandings.



- Now we are going to deploy ELB so that it can distribute traffic between these two.

Step 1: Configure Load Balancer

Name: Scheme: internet-facing IP address type: IPv4

Step 3: Configure Security Groups

Assign a security group: Select an existing security group

Security Group ID	Name	Description	Actions
sg-00a3cf320d4054a9	aws-linu security-group	aws-linu security-group	Copy to new
sg-0df0fc4796752b96	aws-rds-mysql security-group	aws-rds-mysql security-group	Copy to new
sg-09558047168ed7b1a	default	default VPC security group	Copy to new
sg-0acbe99e5a53130c1	nat-instance security-group	nat-instance security-group	Copy to new
sg-055dc0fd47c71f88	private-windows security-group	private-windows security-group	Copy to new
sg-03bc17a19529cc88	public-windows security-group	public-windows security-group	Copy to new
sg-0348e99684b59c2a	web-servers security-group	web-servers security-group	Copy to new

DNS Management
ajclassroom.co.in

Type	Name	Value	TTL
CNAME	Host *	Points to *	TTL *
	www	web-app-load-balance	1 Hour

Records

Last updated 13-07-2020 03:06 AM

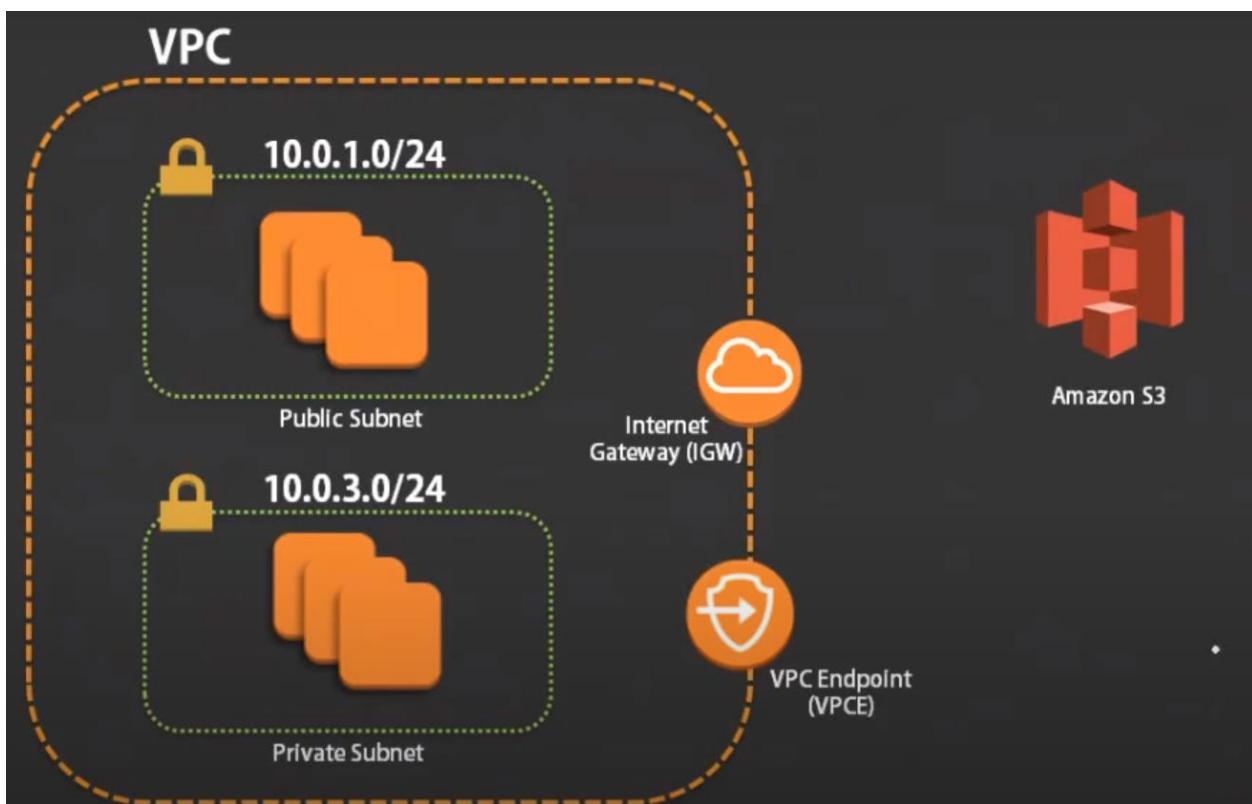
Type	Name	Value	TTL
NS	@	ns43.domaincontrol.com	1 Hour
NS	@	ns44.domaincontrol.com	1 Hour
SOA	@	Primary nameserver: ns43.domainc...	1 Hour

Next up

- Practice the task and get familiar with Web App Deployments
- More tutorials and videos on my YouTube channel
- Connect me over LinkedIn
- Learn more about all you can do

Success!
Your Website (#2#) is ready, HTML app is up and running.

VPC Endpoints



- AWS services like S3, DynamoDB, etc. are connected via Internet Gateway by default.
- Traffic has to go via IGW and come back although the services are still on AWS
- VPC Endpoint is like an IGW but it interconnects connects AWS Services via AWS internal network rather than traversing over internet.
- Originating and Destination resources should be in same region.

[LAB] Get Access to S3 from Private Subnet EC2 Instance

Step 1: Create a VPC Endpoint for S3

The screenshot shows the AWS VPC Dashboard. A red arrow points to the "Create Endpoint" button. The search bar at the top has "com.amazonaws.ap-south-1.s3" typed into it. The table below shows one existing endpoint entry.

Name	Endpoint ID	VPC ID	Service name	Gateway
vpc-endpoint	vpc-e0391ef9f56b71751	vpc-00c3b6efcf0ee9add	com.amazonaws.ap-south-1.s3	Gateway

Step 2: Configure Route Tables for the Endpoint

The screenshot shows the "Create Endpoint" configuration page. A red arrow points to the "Service category" dropdown, which is set to "AWS services". Another red arrow points to the "Service Name" dropdown, which is set to "com.amazonaws.ap-south-1.s3". Below this, a table lists route tables associated with the endpoint.

Route Table ID	Main	Associated With
rtb-0975ced770c1f30c9	Yes	2 subnets
rtb-0a418de72a2f60440	No	2 subnets

Step 3: Create a New VPC Route Table

The screenshot shows the "Create route table" configuration page. A red arrow points to the "Name" dropdown, which is set to "ap-south-1-private-route-table". Another red arrow points to the "Route Table ID" dropdown, which is set to "rtb-0a418de72a2f60440".

Step 4: Add a Route to the Route Table

The screenshot shows the "Route Table: rtb-0a418de72a2f60440" configuration page. A red box highlights a new route entry:

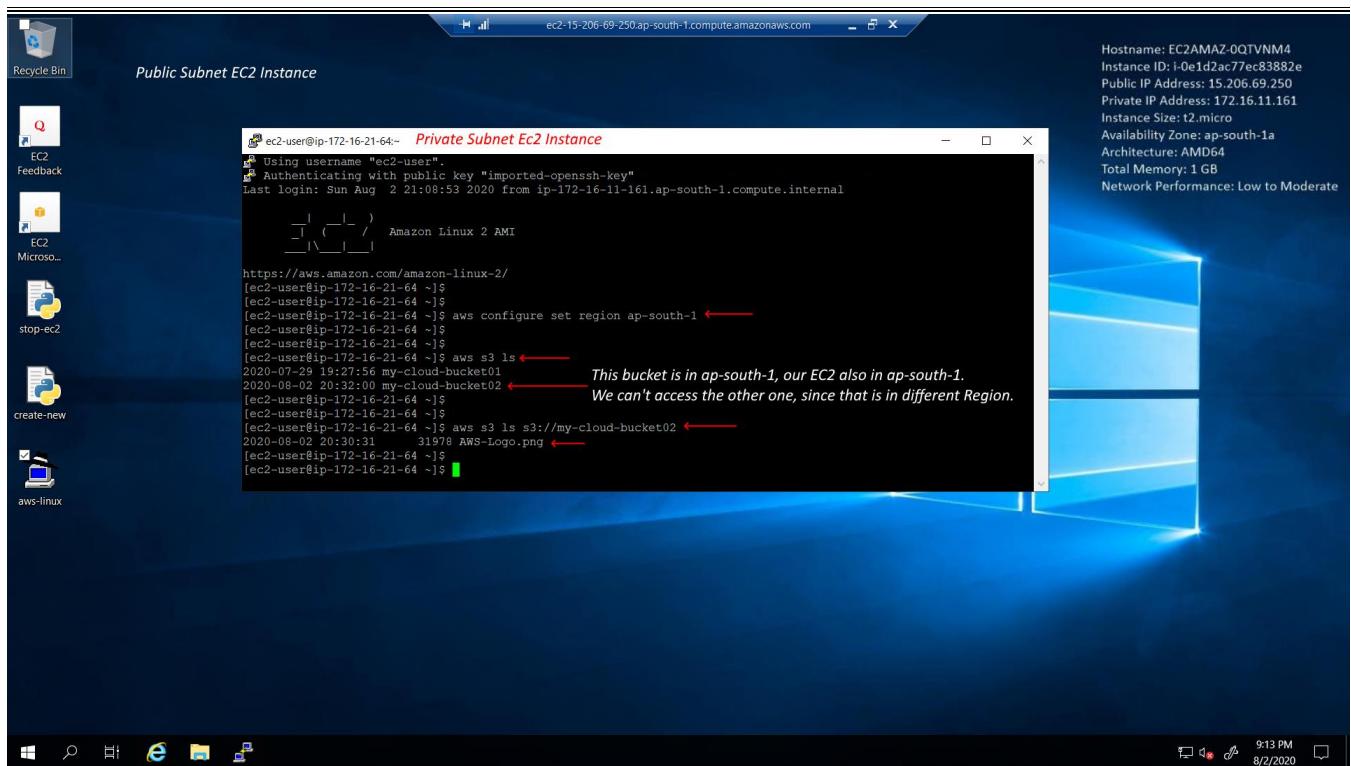
Destination	Target	Status
172.16.0.0/16	local	active
pl-78a54011 (com.amazonaws.ap-south-1.s3, 52.219.62.0/23, 3.5.212.0/23, 3.5.208.0/22, 52.219.64.0/22)	vpc-e0391ef9f56b71751	active
172.18.0.0/16	pxc-0fb69306fac467d6	active

Step 5: Launch an EC2 Instance in the Private Subnet

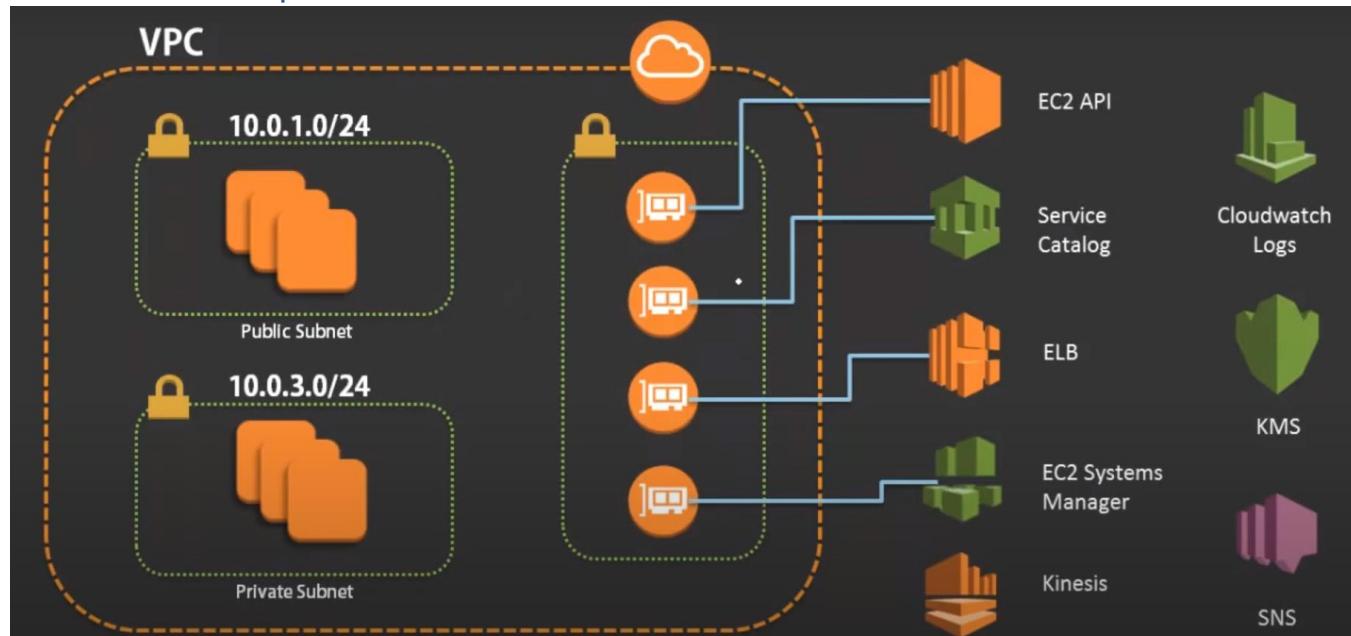
The screenshot shows the EC2 Dashboard. A red arrow points to the "Launch Instance" button. The instance list table shows several instances, with one instance named "private-linux" highlighted. A red arrow points to the "Subnet ID" column for this instance, which is listed as "09483554b239c25c (ap-south-1, private-subnet-1a)".

Step 6: Assign the IAM Role to the EC2 Instance

The screenshot shows the EC2 instance details page for "private-linux". A red box highlights the "IAM role" field, which is set to "s3-full-access-role-for-ec2".

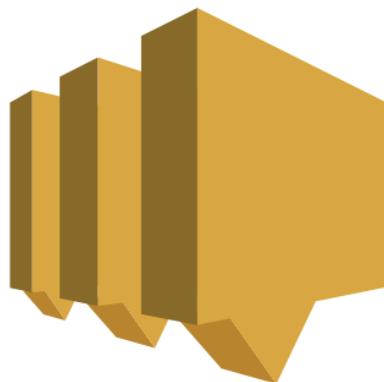


VPC Interface Endpoints



- Some other AWS services are accessed using VPC Interface Endpoint locally via AWS network rather than IGW.
- It's an Interface mapped to specific AWS service, once you deploy a load balancer, there will be an interface created automatically.

SNS - Simple Notification Service



- Used to get notification for AWS resources
- Amazon Simple Notification Service (SNS) is a fully managed messaging service for both system-to-system and app-to-person (A2P) communication
- Amazon SNS enables you to send messages or notifications directly to users with SMS text messages to over 200 countries, mobile push on Apple, Android, and other platforms or email (SMTP)
- Amazon SNS leverages the proven AWS cloud to dynamically scale with your application.

[LAB] SNS Configuration Notification email for ELB Number of Connections

The screenshot shows the AWS Services menu with several service categories expanded. Key visible services include Lambda, VPC, CloudFront, Route 53, API Gateway, Direct Connect, AWS App Mesh, AWS Cloud Map, Global Accelerator, Developer Tools (CodeStar, CodeCommit, CodeArtifact, CodeBuild, CodeDeploy, CodePipeline, Cloud9, X-Ray), Robotics (AWS RoboMaker), and Customer Enablement.

The screenshot shows the Amazon SNS Topics page. It displays a table with one entry: 'dynamodb' with ARN 'arn:aws:sns:ap-south-1:618927232701:dynamodb'. A red arrow points to the 'Topics' link in the left sidebar, and another red arrow points to the 'Create topic' button at the top right of the main content area.

The screenshot shows the 'Create Topic' page for 'sns-app-elb-monitoring-alarm'. It includes fields for 'Name' (sns-app-elb-monitoring-alarm) and 'Display name - optional' (sns-app-elb-monitoring-alarm). The page also contains sections for 'Encryption - optional', 'Access policy - optional', 'Delivery retry policy (HTTP/S) - optional', 'Delivery status logging - optional', and 'Tags - optional'. A red arrow points to the 'Create topic' button at the bottom right.

The screenshot shows the 'Details' page for the 'sns-app-elb-monitoring-alarm' topic. It lists the topic's name, display name, ARN, and topic owner. The 'Subscriptions' tab is selected, showing a table with no subscriptions found. A red arrow points to the 'Create subscription' button in the top right corner of the subscriptions table.

The screenshot shows the 'Create subscription' page for the 'sns-app-elb-monitoring-alarm' topic. It requires entering a 'Topic ARN' (arn:aws:sns:ap-south-1:618927232701:sns-app-elb-monitoring-alarm), 'Protocol' (Email), and 'Endpoint' (vpjazeem@gmail.com). A red arrow points to the 'Create subscription' button at the bottom right.

The screenshot shows the 'Details' page for the created subscription with ARN 'arn:aws:sns:ap-south-1:618927232701:sns-app-elb-monitoring-alarm:81668083-819f-4ab8-a798-3a948e424015'. The 'Status' is listed as 'Pending confirmation'. The 'Subscription filter policy' section indicates 'No filter policy configured for this subscription.' A red arrow points to the 'Topic' link in the details table.

AWS Notification - Subscription Confirmation

sns-app-elb-monitoring 11:30 PM (1 minute ago) to me ▾

You have chosen to subscribe to the topic: **arn:aws:sns:ap-south-1:618927232701:sns-app-elb-monitorin**

To confirm this subscription, click or visit the link below (If this was no action is necessary):

[Confirm subscription](#) ←

Please do not reply directly to this email. If you wish to remove yourself from future SNS subscription confirmation requests please send an email to [sns-opt-](#)

Amazon SNS

Subscription: 81668083-819f-4ab8-a798-3a948e424015

Details

ARN: arn:aws:sns:ap-south-1:618927232701:sns-app-elb-monitoring-alarm:81668083-819f-4ab8-a798-3a948e424015
Endpoint: vpijoseem@gmail.com
Topic: sns-app-elb-monitoring-alarm

Status: Confirmed
Protocol: EMAIL

Subscription filter policy | Redrive policy (dead-letter queue)

No filter policy configured for this subscription.
To apply a filter policy, edit this subscription.

[Edit](#)

CloudWatch Alarms

CloudWatch Alarms

Alarms (6)

[Create alarm](#)

Name	State	Last state update	Conditions
TargetTracking-windows-auto-scaling-group-AlarmHigh-10641761-9b8c-4e49-abec-2bd6fa42d797	OK	2020-07-14 22:10:20	CPUUtilization > minutes
awsapplicationelb-app-web-app-load-balancer-38aa1a134685ea5c5-High-Active-Connection-Count	Insufficient data	2020-07-14 22:09:01	ActiveConnection within 1 minute
awsce2-i-097a89785fa55d872-CPU-Utilization	OK	2020-07-14 22:06:31	CPUUtilization > 10 minutes
TargetTracking-windows-auto-scaling-group-AlarmLow-88641761-9b8c-4e49-be7c-ehb98fb98e57	In alarm	2020-07-14 01:57:28	CPUUtilization < 15 minutes
aws-dynamodb-table1-WriteCapacityUnitsLimit-BasicAlarm	OK	2020-07-07 11:29:06	ConsumedWrite datapoints within
aws-dynamodb-table1-ReadCapacityUnitsLimit-BasicAlarm	OK	2020-07-07 11:28:28	ConsumedRead datapoints within

CloudWatch Alarms

Create alarm

Specify metric and conditions

Metric

Graph

Select metric →

Select metric

Untitled graph

Your CloudWatch graph is empty.
Select some metrics to appear here.

All metrics | **Graphed metrics** | **Graph options** | **Source**

1,067 Metrics

- ApplicationELB** (47 Metrics)
- DynamoDB** (15 Metrics)
- EC2** (565 Metrics)
- NATGateway** (14 Metrics)
- States** (4 Metrics)
- Auto Scaling** (13 Metrics)
- EBS** (313 Metrics)
- Firehose** (2 Metrics)
- RDS** (88 Metrics)
- Usage** (6 Metrics)

Select a single metric to continue

Select metric

All | **ApplicationELB** | **Search for any metric, dimension or resource id** | **Graph search**

47 Metrics

- Per AppELB, per AZ, per TG Metrics** (14 Metrics)
- Per AppELB Metrics** (11 Metrics)
- TargetGroup** (1 Metric)

Select metric

Select metric

Untitled graph

1h 3h 12h 1d 3d 1w custom Line

Count

1 2 3

14:30 14:45 15:00 15:15 15:30 15:45 16:00 16:15 16:30 16:45 17:00 17:15 17:30

ActiveConnectionCount

All metrics Graphed metrics (1) Graph options Source

All > ApplicationELB > Per AppELB Metrics Search for any metric, dimension or resource id Graph search

LoadBalancer (11)

Metric Name

app/web-app-load-balancer/38aa1a134685eac5 RequestCount

app/web-app-load-balancer/38aa1a134685eac5 NewConnectionCount

app/web-app-load-balancer/38aa1a134685eac5 ActiveConnectionCount

app/web-app-load-balancer/38aa1a134685eac5 HTTPCode_Target_4XX_Count

app/web-app-load-balancer/38aa1a134685eac5 TargetResponseTime

app/web-app-load-balancer/38aa1a134685eac5 HTTPCode_Target_3XX_Count

app/web-app-load-balancer/38aa1a134685eac5 ProcessedBytes

app/web-app-load-balancer/38aa1a134685eac5 HTTPCode_ELB_4XX_Count

app/web-app-load-balancer/38aa1a134685eac5 ConsumedLCUs

Cancel Select metric

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Step 3 Add name and description

Step 4 Preview and create

Graph

Count

1 2 3

14:30 14:45 15:00 15:15 15:30 15:45 16:00 16:15 16:30 16:45 17:00 17:15 17:30

ActiveConnectionCount

Namespace AWS/ApplicationELB

Metric name ActiveConnectionCount

LoadBalancer app/web-app-load-balancer/38aa1a134685eac5

Statistic Q Maximum

Period 1 minute

Conditions

Threshold type Static Use a value as a threshold Anomaly detection Use a band as a threshold

Whenever ActiveConnectionCount is... Define the alarm conditions

Greater > threshold Greater/Equal >= threshold Lower/Equal <= threshold Lower < threshold

than... Define the threshold value 3 Must be a number

Additional configuration

Next

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Step 3 Specify metric and conditions

Step 4 Preview and create

Graph

Count

1 2 4 6 8

15:30 16:30 17:30

NewConnectionCount

Namespace AWS/ApplicationELB

Metric name NewConnectionCount

LoadBalancer app/web-app-load-balancer/38aa1a134685eac5

Statistic Q Sum

Period 1 minute

Conditions

Threshold type Static Use a value as a threshold Anomaly detection Use a band as a threshold

Whenever NewConnectionCount is... Define the alarm condition

Greater > threshold Greater/Equal >= threshold Lower/Equal <= threshold Lower < threshold

than... Define the threshold value 3 Must be a number

Additional configuration

Next

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

CloudWatch Alarms Create alarm

Step 1 Specify metric and conditions

Step 2 Configure actions

Step 3 Add name and description

Step 4 Preview and create

Add name and description

Name and description

Alarm name Define a unique name. alarm-app-elb-monitoring

Alarm description - optional Define a description for this alarm. alarm-app-elb-monitoring

Up to 1024 characters (24/1024)

Cancel Previous Next

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Step 2: Configure actions

Actions

Notification When in alarm, send a notification to "sns-app-elb-monitoring-alarm"

Step 3: Add name and description

Name and description

Name alarm-app-elb-monitoring

Description alarm-app-elb-monitoring

Create alarm

Feedback English (US) © 2006 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Search mail

ALARM: "alarm-app-elb-monitoring" in Asia Pacific (Mumbai)

Inbox

sns-app-elb-monitorin... 11:53 PM (3 minutes ago) to me

You are receiving this email because your Amazon CloudWatch Alarm "alarm-app-elb-monitoring" in the Asia Pacific (Mumbai) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [10.0 (14/07/20 18:19:00)] was greater than the threshold (3.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 14 July, 2020 18:23:48 UTC".

View this alarm in the AWS Management Console:
<https://ap-south-1.console.aws.amazon.com/cloudwatch/home?region=ap-south-1#s=Alarms&alarm=alarm-app-elb-monitoring>

Alarm Details:

- Name: alarm-app-elb-monitoring
- Description: alarm-app-elb-monitoring
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [10.0 (14/07/20 18:19:00)] was greater than the threshold (3.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Tuesday 14 July, 2020 18:23:48 UTC
- AWS Account: 618927232701
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:

SQS - Simple Queueing Service



- Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.

SNS	SQS
Publisher / Subscriber System. Topic is subscribed by other party	Queuing service for message processing. SNS can send message to SQS and SQS process later.
Message deliver to many subscribers	Messages lies in the queue, messages in the que are processed by single consumer. (Poll)
Use Case: Do other systems care about the event?	Use Case: Does your system care about an event?

Simple Email Service (SES)

- Amazon managed SMTP server, we must have a domain name to go with SES.
- Amazon Simple Email Service (SES) is a cost-effective, flexible, and scalable email service that enables developers to send mail from within any application.

[LAB] Simple Email Service

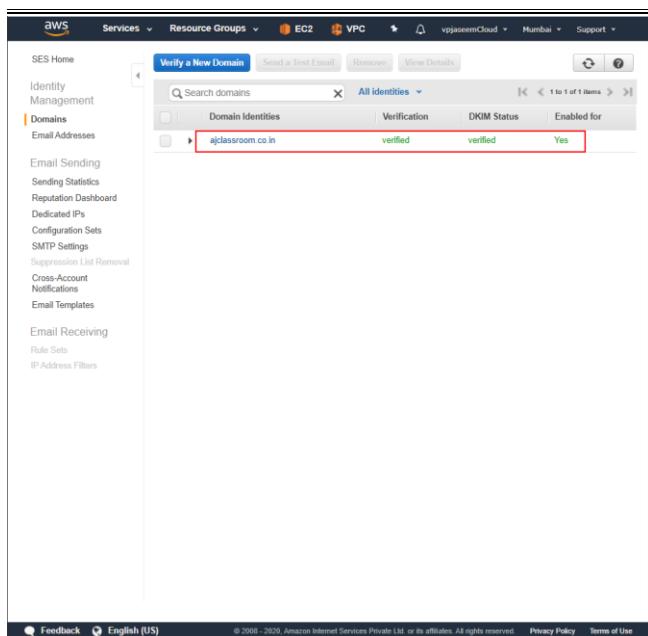
- I'm using Sandbox feature, can send emails to only to verified users. If you are in production, you can enable production grade.
- This is again used for API or Programs to send emails (e.g. Email marketing programs)

The screenshot shows the 'Verify a New Domain' dialog in the AWS SES console. Step 4, 'Verify This Domain', is highlighted with a red box and a red arrow pointing to the 'Verify This Domain' button.

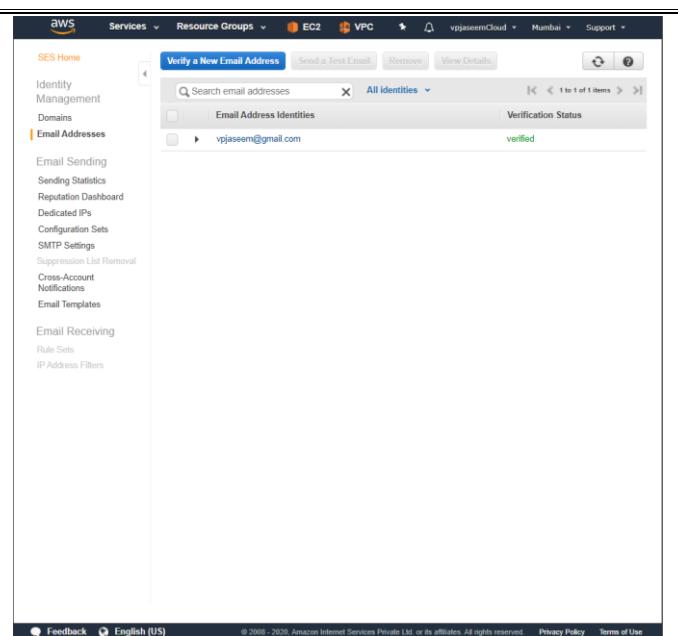
The screenshot shows the 'Verify a New Domain' dialog with the message: 'The domain ajclassroom.co.in has been added to the list of Verified Identities with a Status of "pending verification". Further action is needed to complete verification of this domain. See details below.' The 'Domain Verification Record' table shows entries for 'Name' (amazonses.ajclassroom.co.in), 'Type' (TXT), and 'Value' (OUpq0lMVEF77LX8/EKQ/eGe7l0VAdc5jk9H09+Ag=). The 'DKIM Record Set' table shows entries for 'Name' (wbc34eqjlp6wlmwwwthkrkimuyyude._domainkey.ajclassroom.co.in), 'Type' (CNAME), and 'Value' (wbc34eqjlp6wlmwwwthkrkimuyyude.dkim.amazones.com). The 'Email Receiving Record' table shows an entry for 'Name' (ajclassroom.co.in), 'Type' (MX), and 'Value' (10 inbound-smtp.us-east-1.amazonaws.com).

The screenshot shows the GoDaddy Domain Manager 'DNS Management' section. A newly added TXT record for 'Name' (amazonses.ajclassroom.co.in) and 'Value' (OUpq0lMVEF77LX8/EKQ/eGe7l0VAdc5jk9H09+Ag=) is highlighted with a red box and a red arrow pointing to it.

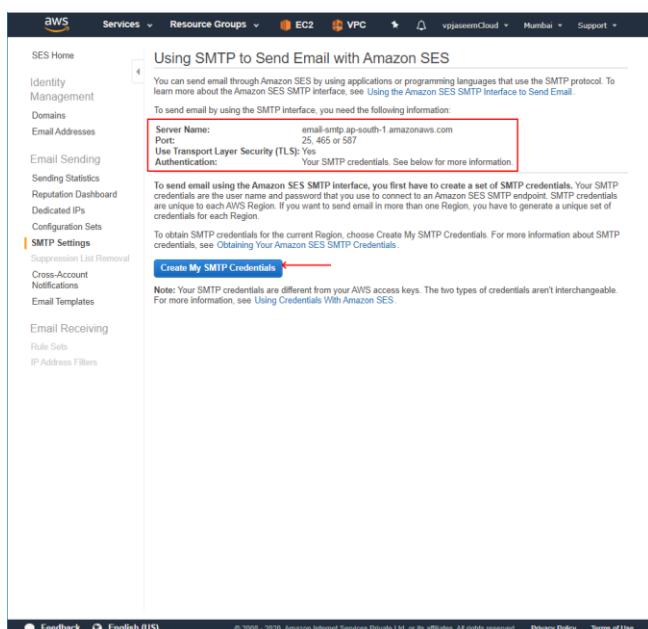
The screenshot shows the AWS SES 'Domains' list. The domain 'ajclassroom.co.in' is listed with a status of 'pending verification' and 'pending verification' for both 'Verification' and 'DKIM Status' columns.



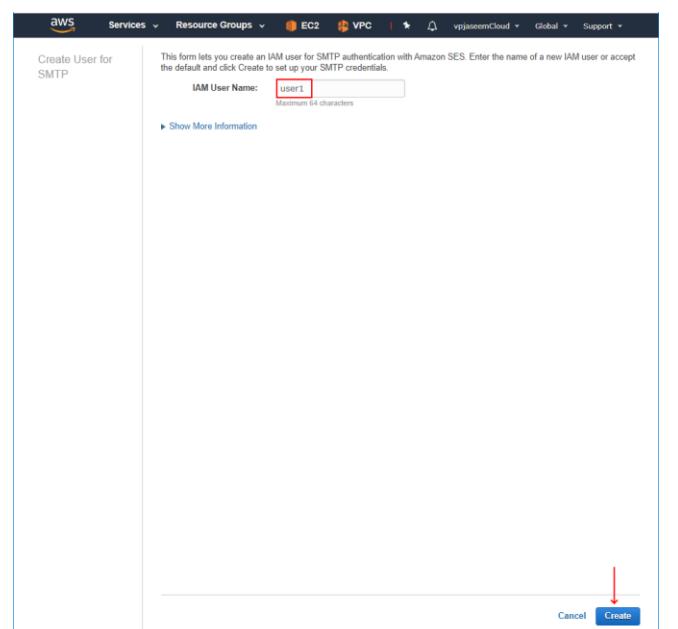
The screenshot shows the AWS SES Home page. Under 'Domain Identities', a domain 'ajclassroom.co.in' is listed with 'Verification' status as 'verified', 'DKIM Status' as 'verified', and 'Enabled for' as 'Yes'. A red box highlights this row.



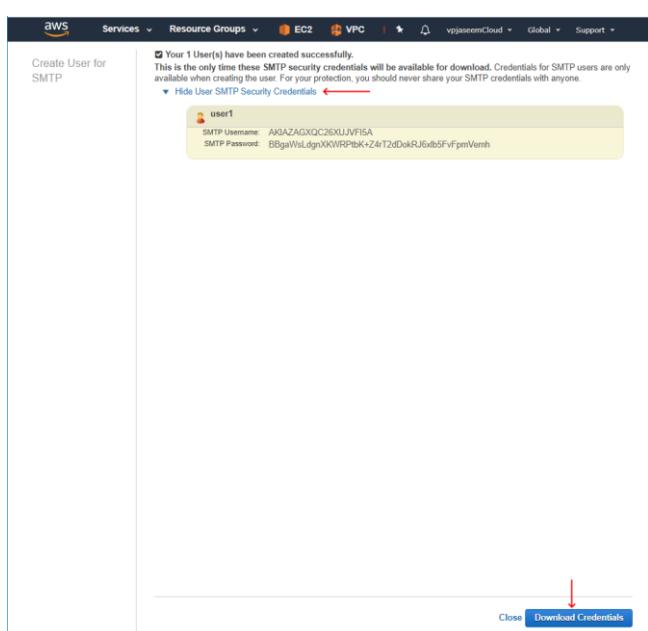
The screenshot shows the AWS SES Home page. Under 'Email Address Identities', an email address 'vpjaseem@gmail.com' is listed with a 'Verification Status' of 'verified'. A red box highlights this row.



The screenshot shows the 'Using SMTP to Send Email with Amazon SES' section. It displays configuration details: Server Name: 'email-smtp.ap-south-1.amazonaws.com', Port: '25, 465 or 587', Use Transport Layer Security (TLS): 'Yes', and Authentication: 'Your SMTP credentials. See below for more information.' A red box highlights the port number '25, 465 or 587'.



The screenshot shows the 'Create User for SMTP' page. It asks for an 'IAM User Name' ('user1') and has a note: 'This form lets you create an IAM user for SMTP authentication with Amazon SES. Enter the name of a new IAM user or accept the default and click Create to set up your SMTP credentials.' A red box highlights the 'IAM User Name' input field.

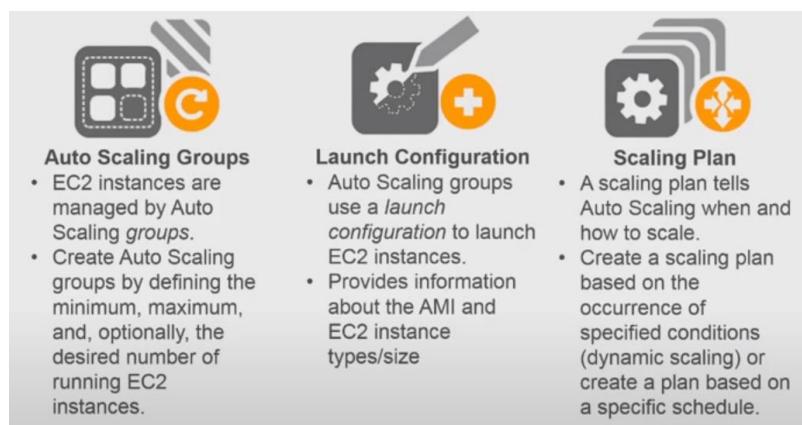


The screenshot shows the 'Create User for SMTP' page after successful creation. It displays a message: 'Your 1 user(s) have been created successfully. This is the only time these SMTP security credentials will be available for download. Credentials for SMTP users are only available when creating the user. For your protection, you should never share your SMTP credentials with anyone.' A red box highlights the message. Below it, there's a 'Hide User SMTP Security Credentials' link and a yellow box containing the generated credentials: 'User1', 'SMTP Username: AKIAZAGXQC6XUJF15A', and 'SMTP Password: B8gaWlLdgnXKRPlbK+Z4/T2dDokRJ6xb5FvPmVmeh'. At the bottom, there are 'Close' and 'Download Credentials' buttons, with a red arrow pointing to the 'Download Credentials' button.



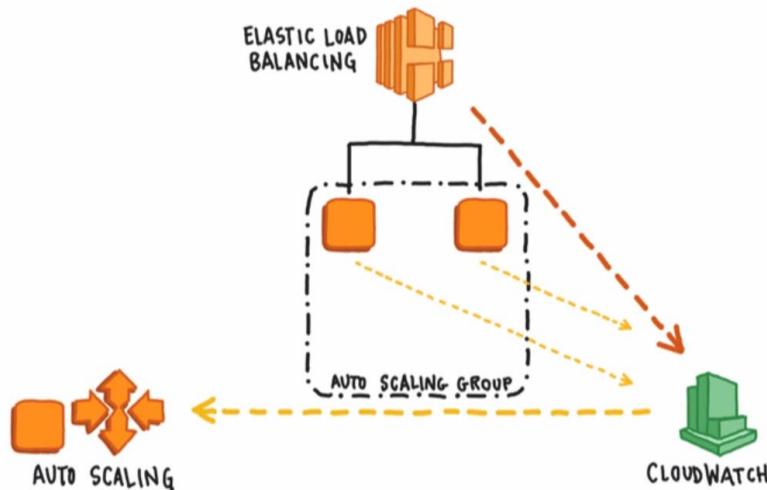
Auto Scaling

- Add more EC2 instances automatically when demand is more.
- Always keep minimum number of instances running.
- Works across AZs and multiple subnets.
- Elastic Load Balances (ELB) tied with all the instances
- Based on the Launch configuration, Autoscaling determine EC2 Instance Type, AMI, Security Group, etc.
- Installation and setup need to be fully automatic. Tuning the server for after the launch is called Bootstrapping - Done via User Data scripts.
- Use amazon Golden Image (Create an EC2 instance, install everything you need on that, convert that as a AMI). Auto Scaling will use this image to launch new EC2s
- If Using Golden Image is not practical, use Base AMI + User data to inject configurations (or Chef/Puppet/Ansible/AWS CodeDeploy)
- If the load is going down, Auto Scaling will terminate instance (wait for no active session)
- Termination Policy used to determine which instances to be terminated (Longest Running, Oldest launch configuration, Closest to full billing hour)
- Scaling Plan: When to do scale up and down
 - Default: Based on the launch configuration
 - Manual: Modify desired capacity via Console, API or CLI (Monitoring tool output fed to some sort of script to perform API call to increase capacity)
 - Scheduled: Scale in / out based on time
 - Dynamic Scaling: Based AWS CloudWatch metrics



- Commonly used for Web based workload with an ELB
- Elastic Load Balances (ELB): Distribute incoming web traffic to automatically, single point of entry to our application. It sends data about ELB and EC2 to CloudWatch.
- Autoscaling uses health check,
 - EC2 Instance Status: Instance is unhealthy if it is not running
 - ELB Health Checks: ELB sends some HTTP request to see if it is getting responses
 - Manual: Mark an instance unhealthy via API

- CloudWatch: Is a web service that enables to monitor and manage various metrics and configure alarms based on those data (e.g. CPU usage / Incoming traffic)



- Based on the load / demand, the entire system growing or shrinking
- Auto Scaler manages the Auto Scaling Group, Load balances (ELB)
- Scale out: Adding more instances
- Scale in: Removing instances
- Simple Scaling Policy: Use any CloudWatch Metric as load metric, specify single threshold beyond which you want to scale.
- Step Scaling Policy: Use any CloudWatch Metric as load metric, specify different threshold which you want to scale.
- Target Tracking Scaling: Specify CPU or Memory

[LAB] Auto Scaling Configuration

- Launch configuration: Collection of all the attributes those are required to launch an EC2

Launch Configuration Creation:

- Launch Configuration Name:** windows-web-servers-launch-configuration
- Amazon Machine Image (AMI):** Windows_Server-2019-English-Full-Base-2020.06.10 (ami-0c5cb28614564c01)
- Instance Type:** t2.micro (1 vCPU, 1 GiB, EBS)
- Additional Configuration (Optional):** IAM instance profile (Select IAM role), Monitoring (Enable CloudWatch Metrics and Logs), EBS-optimized instance.

Launch Configuration Advanced Details:

- User Data:** A PowerShell script to enable port 80 (http://inbound.ps1):


```
#!/bin/bash
# Allow port 80 inbound traffic
sudo ufw allow http
# Start the IIS service
net start w3svc
# Set the default document
powershell -Command "Set-WebConfiguration -Filter "system.webServer/handlers" -Name "staticFile" -Value @{path=""\\"; name=""\\"; file=""\\"; mimeType=""\\"; leaseType=""\\"; leaseInfinite=""\\"; leaseornings="1"; leaseAutoPrune="1"; leaseMaxAge="30000000"} -PSPath "IIS:\WebSites\Default Web Site"
```
- Storage (volumes):** An EBS volume (snap-06faf56a12afbe) of size 30 GiB.

Security Groups:

- Selected Security Group:** web-servers-security-group (sg-0348e96684b59cf2a)
- Other Security Groups:** public-windows-security-group (sg-03bc17a19529ccdb8), private-windows-security-group (sg-055d6bf0d47c71f88), default (sg-09558047168ed7b1a), nat-instance-security-group (sg-0acbe98e5a53130c1), nat-instance-security-group (sg-0d2a39d4b067cf384).

Key Pair (Login):

- Key Pair Options:** Choose an existing key pair (aws-windows) or Existing key pair (aws-windows).
- Agreement:** I acknowledge that I have access to the selected private key file (aws-windows.pem), and that without this file, I won't be able to log in to my instance.

Create Auto Scaling Group:

- Get Started:** Get started with EC2 Auto Scaling by creating an Auto Scaling group.
- Auto Scaling Group:** Create Auto Scaling group.

How it works: Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Pricing: Amazon EC2 Auto Scaling features have no additional fees beyond the costs for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Choose launch template or configuration

Launch configuration

Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

VPC

vpc-0c3b6efcfc0ed...
172.16.0.0/16

Create a VPC

Subnets

ap-south-1a | subnet-0f483554b259-25c
(ap-south-1-private-subnet-1a)
172.16.21.0/24

ap-south-1b | subnet-0a483554b259-25c
(ap-south-1-private-subnet-1b)
172.16.22.0/24

Create a subnet

Cancel Previous Skip to review Next

Configure

Step 2 Configure settings

Step 3 (optional) Configure advanced options

Step 4 (optional) Configure group size and scaling policies

Step 5 (optional) Add notifications

Step 6 (optional) Add tags

Step 7 Review

Load balancing - optional

Choose a load balancer to distribute incoming traffic for your application across instances. You can also set options that give you more control over checking the health of instances.

Enable load balancing

Application Load Balancer or Network Load Balancer

Classic Load Balancer

Choose a target group for your load balancer

Select target group

is-web-servers-target-group

Create a target group

Health checks - optional

Health check type

EC2 Auto Scaling automatically replaces instances that fail health checks. If you want to enable CloudWatch Metrics for your Auto Scaling group, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

EC2

ELB

Health check grace period

The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

600 seconds

Additional settings - optional

Monitoring

Enable group metrics collection within CloudWatch Metrics

Cancel Previous Skip to review Next

New EC2 experience Tell us what you think

EC2 Dashboard Events Tags Reports Lanes

Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Scheduled Instances Capacity Reservations

Images AMIs Bundle Tasks

Elastic Block Store Volumes Snapshots Lifecycle Manager

Network & Security Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

Load Balancing Load Balancers Target Groups

Auto Scaling Launch Configurations Auto Scaling Groups

Feedback English (US)

Configure group size and scaling policies

Step 1 Choose launch template or configuration

Step 2 Configure settings

Step 3 (optional) Configure advanced options

Step 4 (optional) Configure group size and scaling policies

Step 5 (optional) Add notifications

Step 6 (optional) Add tags

Step 7 Review

Group size - optional

Set the desired minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically tune the number of instances in the group.

Desired capacity

4

Minimum capacity

2

Maximum capacity

6

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand.

Target tracking scaling policy

Choose a desired outcome and leave to the scaling policy to automatically determine how to achieve that outcome.

None

Scaling policy name

Target Tracking Policy

Metric type

Average CPU utilization

Target value

80

Instances need

300 seconds warm up before including in metric

Disable scale-in to create only a scale-out policy

Instance scale-in protection - optional

If process from scale-in is enabled, newly launched instances will be protected from scale-in by default.

Enable instance scale-in protection

Cancel Previous Skip to review Next

- Make sure you remove other 2 EC2 instances from the load balancer that we configured in previous lab (ELB Lab)
- For that edit the target group and remove instances.
- Otherwise create a new load balancer after the Auto Scaling configuration and attach to this.

Step 1: Choose launch template or configuration

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

Tags (1)

Key	Value - optional
Name	win-web-server-auto-scale

Step 2: Configure advanced options

Step 3: Configure group size and scaling policies

Step 4: Configure group size and scaling policies

Group size

Desired capacity	2	Minimum capacity	2	Maximum capacity	5
------------------	---	------------------	---	------------------	---

Scaling policy

Target tracking scaling	Policy rule	Scaling policy name	Target Tracking Policy	Execute policy when
Take the action	Add or remove capacity units as required	Instances need	300 seconds to warm up before scaling in	As required to maintain Average CPU utilization at 80

Instance scale-in protection

Enable instance protection from scale in

Step 5: Add notifications

Notifications

No notifications

Step 6: Add tags

Tags (1)

Key	Value	Tag new instances
Name	win-web-server-auto-scale	Yes

Create Auto Scaling group

Instances | EC2 Manager

Not secure [\[redacted\]](http://web-app-load-balancer-1003193340.ap-south-1.elasticbeanstalk.com)

HTML Application

Success!

Your Website is ready, HTML app is up and running.

Next up

- Practice the task and get familiar with Web App Deployments
- More tutorials and videos on my YouTube channel
- Connect me over LinkedIn
- Learn more about all you can do

Create target group

search : am:aws elasticloadbalancing ap-south-1... Add filter

Name	Port	Protocol	Target type	Load Balancer	VPC ID	Monitoring
lls-web-servers-target-group	80	HTTP	instance	web-app-lo...	vpc-00c3b8efcf0ee9add	

Target group: lls-web-servers-target-group

Description Targets Health checks Monitoring Tags

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Registered targets

Instance ID	Name	Port	Availability Zone	Status	Description
i-05de3843923ae5c08e	win-web-server-auto-scale	80	ap-south-1a	healthy	This target is currently passing target group's health checks.
i-05824f2b0fc924c15	win-web-server-auto-scale	80	ap-south-1a	unhealthy	Health checks failed with these codes: [404]
i-0cfef5b625d6e59	win-web-server-auto-scale	80	ap-south-1b	healthy	This target is currently passing target group's health checks.
i-09952a71761fba44d	win-web-server-auto-scale	80	ap-south-1b	healthy	This target is currently passing target group's health checks.

Availability Zones

Availability Zone	Target count	Healthy?
ap-south-1a	2	Yes
ap-south-1b	2	Yes

[LAB] Test Auto Scaling Lab: Unhealthy Instance

- Auto Scaling replaces unhealthy instances as well as scale out (add more) and scale in (terminate unwanted)
- Let's test what if the health check fails? So far, we have 4 desired instances, if I shutdown an instance manually, the health check fails (get request to /index/html return 404).
- Auto Scaler will launch new instance and terminate the other one

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for New EC2 Experience, EC2 Dashboard, Events, Tags, Limits, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Load Balancers, Target Groups, and Auto Scaling.

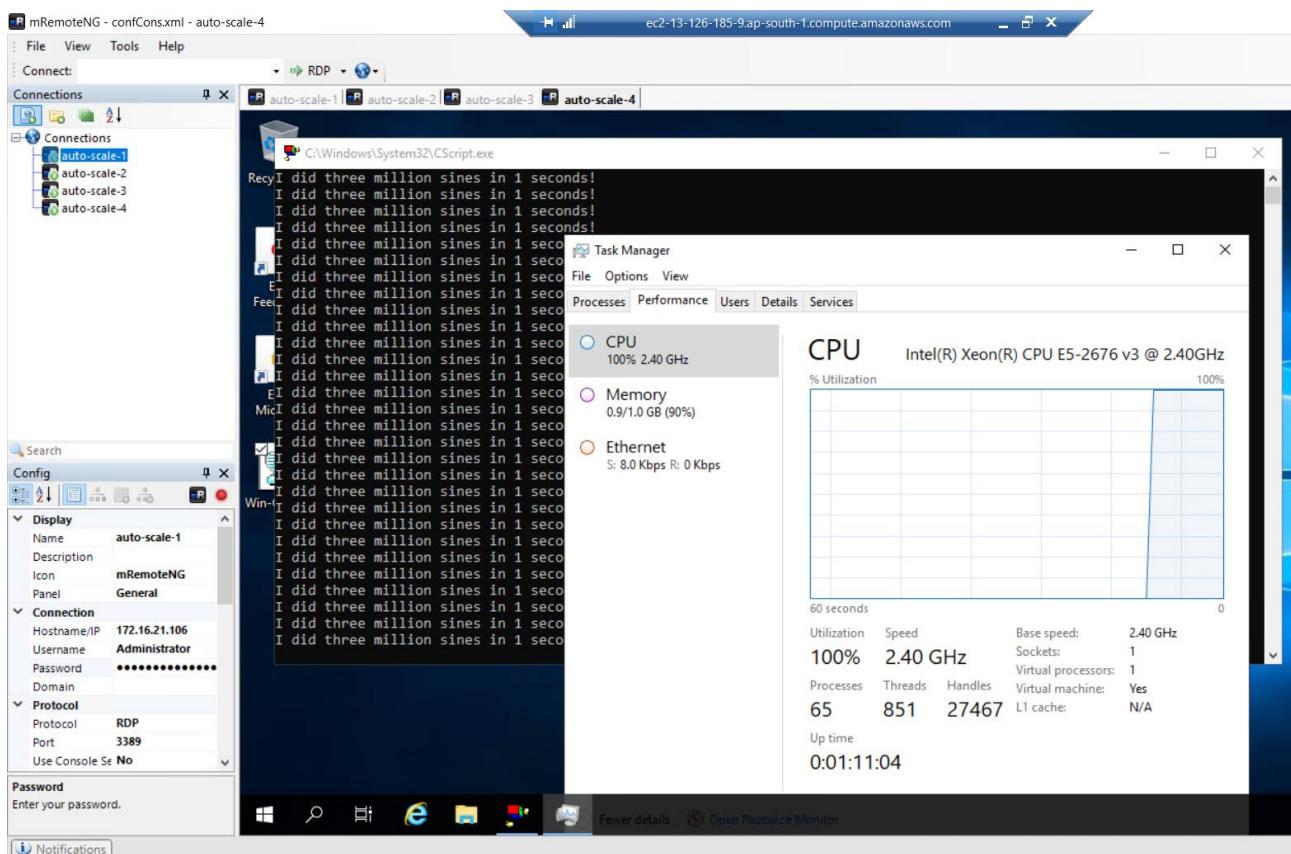
The main content area displays a table of 8 instances:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
win-web-server-auto-scale	i-0f881b2453a10642e	t2.micro	ap-south-1a	terminated	
win-web-server-auto-scale	i-0ea4dd460c59c1842	t2.micro	ap-south-1a	terminated	
win-web-server-auto-scale	i-0c0fefdf5b6256de59	t2.micro	ap-south-1b	running	2/2 checks
win-web-server-auto-scale	i-09952a71761fba44d	t2.micro	ap-south-1b	terminated	
win-web-server-auto-scale	i-07f00cd38b8972b83	t2.micro	ap-south-1b	running	Initializing
win-web-server-auto-scale	i-05de3843923e6c08e	t2.micro	ap-south-1a	running	2/2 checks
win-web-server-auto-scale	i-05824f2b0fc924c15	t2.micro	ap-south-1a	running	2/2 checks
win-web-server-auto-scale	i-034939a2acf57e15	t2.micro	ap-south-1b	stopped	

At the bottom of the table, it says "Select an instance above".

[LAB] Test Auto Scaling Lab: CPU Utilization Scale Out

- I'm going to run a script on 2 EC2s and that will over utilize the CPU. This activity kicks Scale Out.
- Download the script: [Win-CPUBUSY.VBS](#)
- I have used mRemoteNG software to connect all the instances and run the Script. You right click on the script and use 'Run with command prompt'



- At this stage the desired capacity changes automatically to 6

AWS Services Resource Groups EC2 VPC S3 Support vpjaseemCloud Mumbai

New EC2 Experience Tell us what you think

EC2 Dashboard New Events New Tags Limits Instances Instances Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts New Capacity Reservations Images AMIs Elastic Block Store Volumes Snapshots Lifecycle Manager Network & Security Security Groups New Elastic IPs New Placement Groups New Key Pairs New Network Interfaces

Launch Instance Connect Actions

Name : win-web-server-auto-scale Instance State : Running Add filter ? 1 to 6 of 6

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Check
<input type="checkbox"/>	win-web-server-auto-scale	i-0daadda18a637f08e	t2.micro	ap-south-1a	running	Initializin
<input type="checkbox"/>	win-web-server-auto-scale	i-0c0fefdf5b6256de59	t2.micro	ap-south-1b	running	✓ 2/2 checks
<input type="checkbox"/>	win-web-server-auto-scale	i-0810d21efb960bee6	t2.micro	ap-south-1a	running	✓ 2/2 checks
<input type="checkbox"/>	win-web-server-auto-scale	i-07f00cd38b8972b83	t2.micro	ap-south-1b	running	✓ 2/2 checks
<input type="checkbox"/>	win-web-server-auto-scale	i-05824f2b0fc924c15	t2.micro	ap-south-1a	running	✓ 2/2 checks
<input type="checkbox"/>	win-web-server-auto-scale	i-0033e067995cc32a6	t2.micro	ap-south-1b	running	✓ 2/2 checks

Select an instance above

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

[LAB] Test Auto Scaling Lab: CPU Utilization Scale In

- When you step the Script, the instance CPU utilization goes below 80% (threshold), that changes the desired instances value to go down.
- This will eventually terminate extra instances

Route 53

- Highly available and scalable DNS web service
- Route 53 registers domain name, acts as a registrar server
- Points Domain name to AWS Service (ELB, EC2)
- Health checks of resources
- It works on edge locations
- Alias: All AWS Services DNS names (ELD, EC2, S3, etc.) mapped to friendly Alias
- CNAME: One DNS name maps to another DNS name

Routing Policies

- **Simple Routing Policy:** One to One mapping, one domain name map to one resource
- **Failover Routing Policy:** Route the traffic to healthy instance, if the instance is unhealthy, route traffic to redundant instance.
- **Geolocation Routing Policy:** Routing based on geographic location of the user.
- **Geoproximity Routing Policy:** Routing based on geographic location of the resource, optimally shift traffic from one resource to another resource in different location.
- **Latency Based Routing:** When website is deployed in multiple regions, this routing is used.
- **Multivalue Answer Routing:** 8 servers get the request in random way.
- **Weighted Routing Policy:** Control the traffic routed to each resource, used for load balancing and testing (20% to some new instances and 80% to old instances)

[LAB] Route 53

Domain Registration

Left Screenshot: Registered domains

The 'Registered domains' page shows a search bar and a table with columns: Domain Name, Privacy Protection, Expiration Date, Auto Renew, and Transfer Lock. A red arrow points to the 'Register Domain' button.

Right Screenshot: Choose a domain name

The 'Choose a domain name' page shows a search bar with 'ajclassroom'. A red arrow points to the 'Check' button. Below it is a table for 'Availability for 'ajclassroom.com'' showing one result: ajclassroom.com (Status: Available, Price /1 Year: \$12.00). Another table for 'Related domain suggestions' is also shown.

Hosted Zone

Left Screenshot: Create Hosted Zone

The 'Create Hosted Zone' page has a 'Create Hosted Zone' button highlighted. It includes a description of Route 53 and a 'Create Hosted Zone' button.

Right Screenshot: Create Hosted Zone

The 'Create Hosted Zone' page shows a search bar and a table with columns: Domain Name, Type, Record Set Count, Comment, and Hosted Zone ID. A red arrow points to the 'Create Hosted Zone' button.

Bottom Left Screenshot: Create Hosted Zone

The 'Create Hosted Zone' form shows 'Domain Name: ajclassroom.co.in', 'Comment: ajclassroom.co.in', and 'Type: Public Hosted Zone'. A red arrow points to the 'Create' button.

Bottom Right Screenshot: Create Record Set

The 'Create Record Set' page shows a table for 'Weighted Only' record sets. One entry is highlighted with a red box: 'ajclassroom.co.in.' NS ns-753.awsdns-30.net, ns-157.awsdns-19.com, ns-1090.awsdns-08.org, ns-1995.awsdns-52.co.uk. Another entry is 'ajclassroom.co.in.' SOA ns-753.awsdns-30.net awsdns-hostmaster.amazon.

Type	Name	Value	TTL	Action
NS	@	ns43.domaincontrol.com	1 Hour	
NS	@	ns44.domaincontrol.com	1 Hour	
SOA	@	Primary nameserver: ns43.domainc...	1 Hour	
NS	@	ns-753.awsdns-30.net	1 Hour	
NS	@	ns-157.awsdns-19.com	1 Hour	
NS	@	ns-1090.awsdns-08.org	1 Hour	
NS	@	ns-1955.awsdns-52.co.uk	1 Hour	

Record Set Name:

Type: A – IPv4 address

Alias: Yes No

TTL (Seconds): 1m 5m 1h 1d

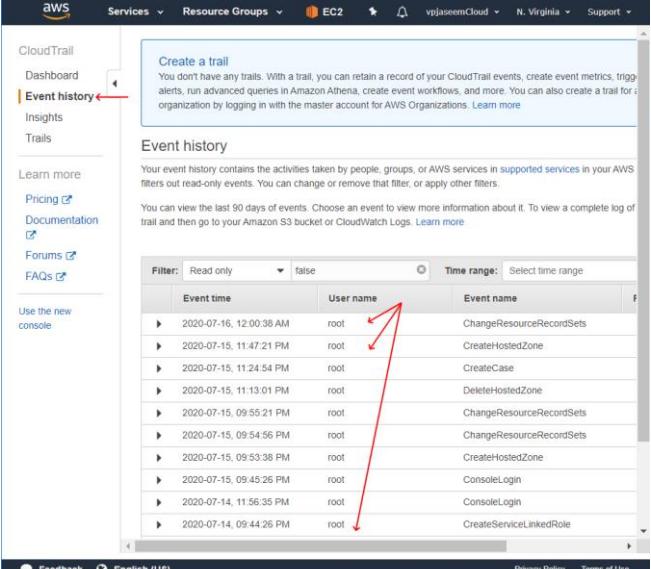
Value:

Routing Policy: Simple

AWS CloudTrail

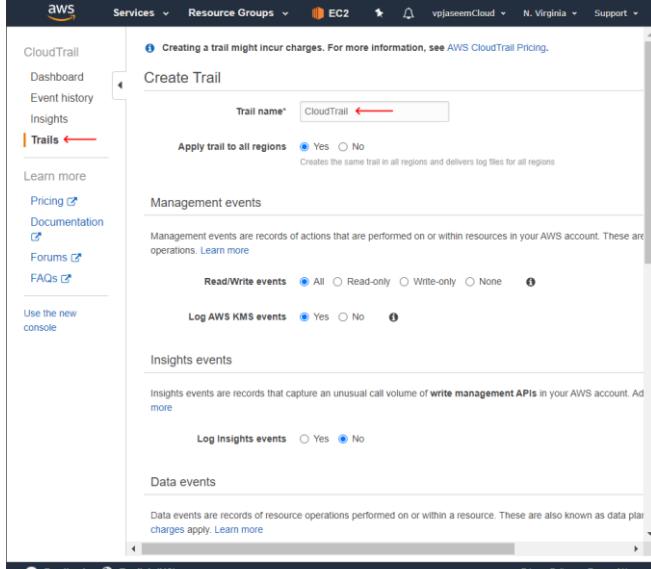
- With CloudTrail, you can view events for your AWS account. Create a trail to retain a record of these events. With a trail, you can also create event metrics, trigger alerts, and create event workflows.
- User Activity reordered in CloudTrail, Available in 90 days (event viewer logs, audit logs)
- If you want to retain the logs after 90 days, you can create a Trail to send the logs to S3

[LAB] Cloud Trail



The screenshot shows the AWS CloudTrail Event history page. It displays a table of events with columns for Event time, User name, and Event name. The table lists various AWS API calls made by the root user on different dates. A red arrow points to the 'Event name' column.

Event time	User name	Event name
2020-07-16, 12:00:38 AM	root	ChangeResourceRecordSets
2020-07-15, 11:47:21 PM	root	CreateHostedZone
2020-07-15, 11:24:54 PM	root	CreateCase
2020-07-15, 11:13:01 PM	root	DeleteHostedZone
2020-07-15, 09:55:21 PM	root	ChangeResourceRecordSets
2020-07-15, 09:54:56 PM	root	ChangeResourceRecordSets
2020-07-15, 09:53:38 PM	root	CreateHostedZone
2020-07-15, 09:45:26 PM	root	ConsoleLogin
2020-07-14, 11:56:35 PM	root	ConsoleLogin
2020-07-14, 09:44:26 PM	root	CreateServiceLinkedRole



The screenshot shows the AWS CloudTrail Create Trail page. It has fields for Trail name (CloudTrail), Apply trail to all regions (Yes selected), Read/Write events (All selected), Log AWS KMS events (Yes selected), and Log Insights events (No selected). A red arrow points to the 'Trail name' input field.

AWS Lambda



- Allows to execute our code without server (also called serverless computing). We don't need to manage any servers (EC2s) to run the code. AWS will provide managed environment to run the codes.
- Supports code C#, Java, Python, Node.js and Go languages
- 128MB Memory to 3008 MB memory, we can choose while creating Lambda
- Functions can run between 100ms to 5 minutes
- We should assign IAM role to Lambda so that it can access other AWS services
- We can run Lambda function manually or scheduled or based on some events
- Commonly used to run standalone scripts

[LAB] AWS Lambda Example

- Start and stop EC2 instances with Lambda, Python and CloudWatch to schedule
- Aim: Run EC2 instances only Mon-Fri 9Am to 6PM

Step 1: Create a Policy with below JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Start*",
        "ec2:Stop*"
      ],
      "Resource": "*"
    }
  ]
}
```

Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

[Visual editor](#) [JSON](#) [Import managed policy](#)

```

1 "{
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "logs:CreateLogGroup",
8         "logs:CreateLogStream",
9         "logs:PutLogEvents"
10      ],
11      "Resource": "arn:aws:logs:*:*:*"
12    },
13    {
14      "Effect": "Allow",
15      "Action": [
16        "ec2:Start*",
17        "ec2:Stop*"
18      ],
19      "Resource": "*"
20    }
21  ]
}

```

Character count: 235 of 6,144.

[Cancel](#) [Review policy](#)

Create policy

Review policy

Name* lambda-ec2-start-stop-policy [Use alphanumeric and '-' characters. Maximum 128 characters.](#)

Description lambda-ec2-start-stop-policy [Maximum 1000 characters. Use alphanumeric and '-' characters.](#)

Summary This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose [Show remaining](#). [Learn more](#)

Service	Access level	Resource
CloudWatch Logs	Limited: Write	arn:aws:logs:*
EC2	Limited: Write	All resources

* Required [Cancel](#) [Previous](#) [Create policy](#)

- This will allow EC2 Start and Stop and Upload logs to CloudWatch
- Next step, we create an IAM Role and attach this policy

Step 2: IAM Role for Lambda

- When attaching a permissions policy, search for and choose the IAM policy that you created.

Create role

Select type of trusted entity

Choose a use case

Common use cases

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

Lambda
Allows Lambda functions to call AWS services on your behalf.

Or select a service to view its use cases

API Gateway	CodeGuru	ElastiCache	Kinesis	RoboMaker
AWS Backup	CodeStar Notifications	Elastic Beanstalk	Lake Formation	S3
AWS Chatbot	Comprehend	Elastic Container Service	Lambda	SMS
AWS Support	Config	Elastic Transcoder	Lex	SNS
Amplify	Connect	Elastic Load Balancing	License Manager	SWF
AppStream 2.0	DMS	Forecast	Machine Learning	SageMaker
AppSync	Data Lifecycle Manager	GameLift	Macie	Security Hub
Application Auto Scaling	Data Pipeline	Global Accelerator	Managed Blockchain	Service Catalog
Application Discovery Service	DataSync	Glue	MediaConvert	Step Functions
Batch	DeepLens	Greengrass	Migration Hub	Storage Gateway
	Directory Service	GuardDuty	OpsWorks	Systems Manager

* Required Cancel Next: Permissions

Create role

Review

Provide the required information below and review this role before you create it.

Role name* lambda-ec2-start-stop-role

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

Role description Allows Lambda functions to call AWS services on your behalf

Maximum 1000 characters. Use alphanumeric and '+=.,@-_ characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies lambda-ec2-start-stop-policy

Permissions boundary Permissions boundary is not set

The new role will receive the following tag

Key	Value
Name	lambda-ec2-start-stop-role

* Required Cancel Previous Create role

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Step 3: Create Lambda Function

The screenshot shows the AWS Lambda Functions page. A red arrow points to the 'Create function' button at the top right of the main content area.

This screenshot shows the 'Create Lambda Function' wizard. It includes fields for 'Function name' (automate-ec2-start), 'Runtime' (Python 3.8), and 'Permissions' (using an existing role named 'lambda-ec2-start-stop-role'). A red box highlights the 'lambda-ec2-start-stop-role' selection in the dropdown.

The Lambda Designer interface shows a single function named 'automate-ec2-start'. It has no triggers or destinations defined. A red arrow points to the 'Skip this part' link.

The code editor displays the 'lambda_function.py' file with the following code:

```

import boto3
region = "us-east-1"
ec2 = boto3.client('ec2', region_name=region)
def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=[instances])
    print('started your instances: ' + str(instances))
  
```

The 'Basic settings' section is shown with the 'Edit' button highlighted. Other settings include Runtime (Python 3.8), Handler (lambda_function.lambda_handler), and Timeout (0 min 3 sec). A red arrow points to the 'Edit' button.

The 'AWS X-Ray' section shows the 'Active tracing' checkbox and a 'View traces in X-Ray' button.

The 'VPC' section indicates 'No VPC configuration'. The 'File system' section shows 'No file systems'.

This screenshot shows the 'Edit basic settings' page with various configuration options like Description, Runtime (Python 3.8), Handler (lambda_function.lambda_handler), Memory (128 MB), Timeout (10 seconds), and Execution role (using an existing role named 'lambda-ec2-start-stop-role'). A red box highlights the 'lambda-ec2-start-stop-role' selection in the dropdown.

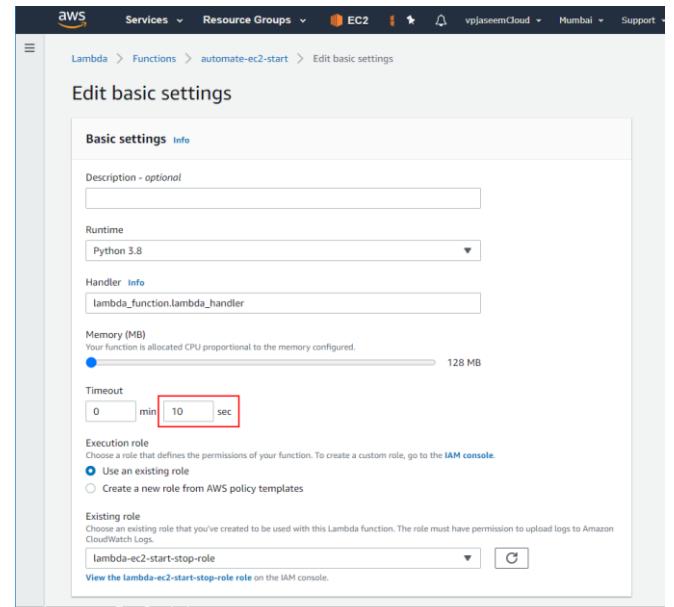
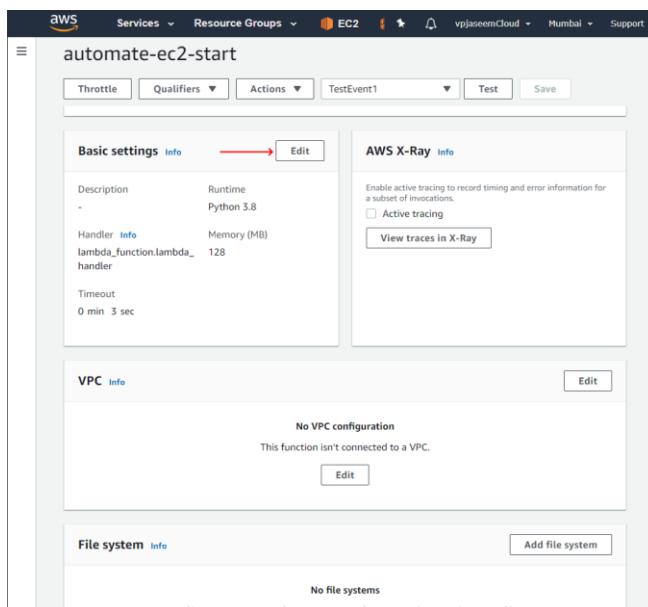
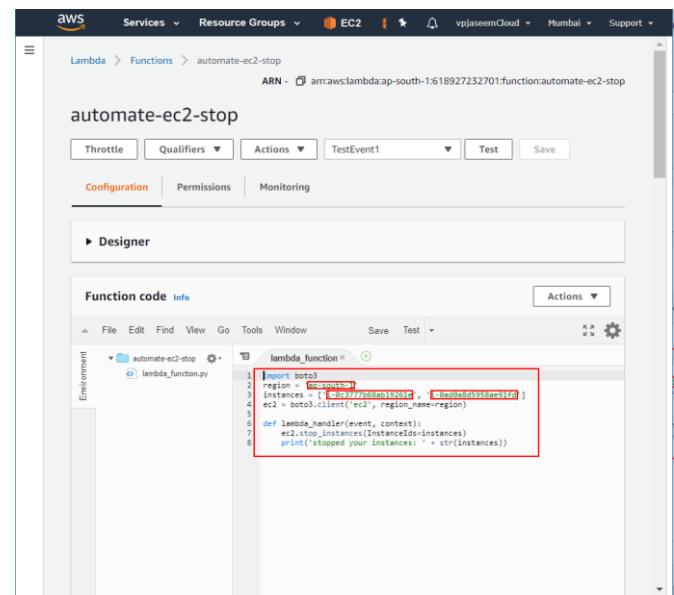
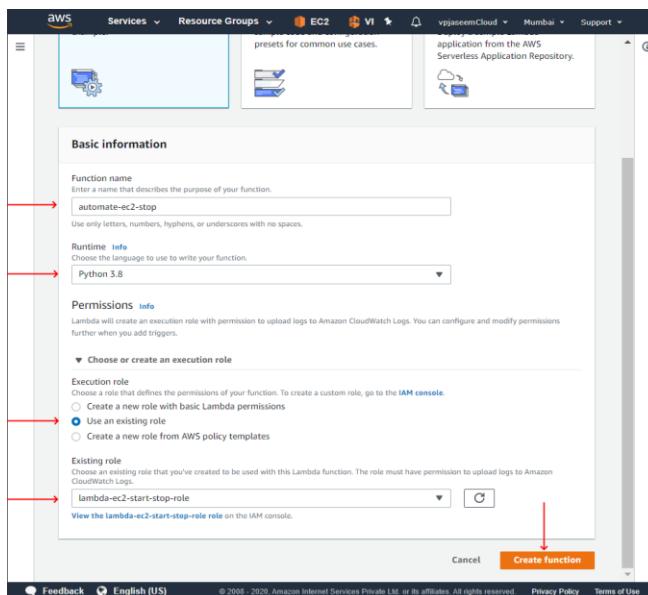
```
import boto3
region = 'us-west-1'
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

- Create another Lambda Function to stop EC2 instances

```
import boto3
region = 'us-west-1'
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```



Step 3: Test Lambda functions

The screenshot shows the AWS Lambda Functions page. It lists two functions: "automate-ec2-start" and "automate-ec2-stop". The "automate-ec2-stop" function is selected. A red arrow points from the "Test" button in the top navigation bar to the "Test" button in the "Configure test event" dialog.

The "Configure test event" dialog is open. It shows the "Event template" dropdown set to "hello-world" and the "Event name" input field containing "TestEvent1". A red arrow points from the "Create" button at the bottom right to the "Create" button in the Lambda function configuration page.

The Lambda function configuration page for "automate-ec2-stop" is shown. The "Actions" dropdown is set to "TestEvent1". A red arrow points from the "Test" button in the Actions dropdown to the "Test" button in the "Function code" tab of the configuration page.

The Lambda function configuration page shows a success message: "Your changes have been saved." The "Function code" tab is open, displaying the Python code for the lambda function. A red arrow points from the "Test" button in the Actions dropdown to the "Execution result" section.

The "Execution result" section shows the log output of the function execution. The log includes the start and end requests, the duration (445.35 ms), and the resources configured (128 MB). The log output also shows the function stopping the specified EC2 instances.

Step 4: Create CloudWatch Rule to Trigger Lambda Functions

The screenshot shows the AWS CloudWatch Events console with the following details:

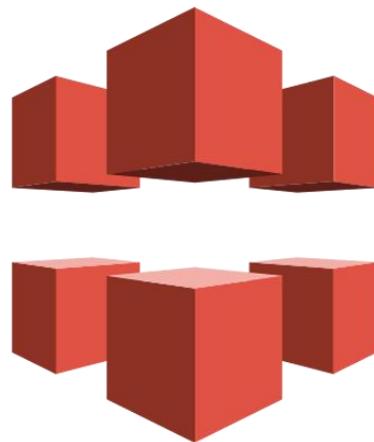
- Event Source:** Set to "Schedule".
- Cron expression:** Set to "55 19 * * ?".
- Targets:** A Lambda function named "automate-ec2-stop" is selected.
- Configure details:** A red arrow points to this button at the bottom right of the configuration pane.

AWS Docs: [Cron Schedule Strings](#)

- Create another Rule for starting the instances

Read More about the LAB: [AWS Docs](#)

AWS CloudFront



- While we access content from far region, it may take some time
- CloudFront will cache the data in nearest location and made available to users faster
- Speeds up the content distribution
- Works on AWS Edge Location
- It's a CDN (Content Delivery Network)
- Provide HTTPS by default

[LAB] AWS CloudFront Distribution for a load balanced Web Site

- Host a PHP website in 2 Linux EC2 instance, Connect to Load balancer to these instances via target group.
- Make sure your Web app is accessible via ELB DNS name.

The screenshot shows the AWS CloudFront 'Create Load Balancer' interface. In the 'Step 1: Select delivery method' section, the 'Web' delivery method is selected. In the 'Step 2: Create distribution' section, the 'Origin Settings' tab is active. The 'Origin Domain Name' dropdown is set to 'vpc1-ap-south-1-elb-289521710.ap-south-1.elb.amazonaws.com'. The 'Origin Path' field is empty. Under 'Minimum Origin SSL Protocol', 'TLSv1' is selected. Under 'Origin Protocol Policy', 'HTTP Only' is selected. Other settings include 'Origin Connection Attempts' (3), 'Origin Connection Timeout' (10), 'Origin Response Timeout' (30), 'Origin Keep-alive Timeout' (5), 'HTTP Port' (80), and 'HTTPS Port' (443). The 'Default Cache Behavior Settings' tab is visible at the bottom.

This screenshot shows the 'Step 1: Select delivery method' page. It displays a list of delivery methods: 'Web', 'RTMP', and 'CloudFront Live Streaming'. The 'Web' option is highlighted. Below the list, there is a note about RTMP being discontinued on December 31, 2020. A 'Get Started' button is present.

This screenshot shows the 'Step 2: Create distribution' page under the 'Default Cache Behavior Settings' tab. It includes sections for 'Viewer Protocol Policy' (set to 'Redirect HTTP to HTTPS'), 'Allowed HTTP Methods' (set to 'GET, HEAD'), and 'Field-level Encryption Config'. There are also sections for 'Cached HTTP Methods', 'Cache and origin request settings', 'Cache Based on Selected Request Headers', 'Object Caching', and various TTL settings.

This screenshot shows the 'Step 2: Create distribution' page under the 'Distribution Settings' tab. It includes sections for 'Forward Cookies', 'Query String Forwarding and Caching', 'Smooth Streaming', 'Restrict Viewer Access', 'Compress Objects Automatically', and 'Lambda Function Associations'. The 'Price Class' is set to 'Use All Edge Locations (Best Performance)'. The 'AWS WAF Web ACL' is set to 'None'. The 'Alternate Domain Names (CNAMEs)' field is empty. The 'SSL Certificate' is set to 'Default CloudFront Certificate (*.cloudfront.net)'. A note at the bottom states that this option requires CloudFront to serve the content over HTTPS.

This screenshot shows the 'Step 2: Create distribution' page with the 'Create Distribution' button highlighted. Other visible fields include 'Supported HTTP Versions' (set to 'HTTP/2, HTTP/1.1, HTTP/1.0'), 'Default Root Object' (set to 'index.php'), 'Logging' (set to 'Off'), 'Bucket for Logs', 'Log Prefix', 'Cookie Logging', 'Enable IPv6' (checked), 'Comment', and 'Distribution State' (set to 'Enabled').

The left pane shows the AWS CloudFront console with the 'Distributions' section selected. A red arrow points to the 'Distributions' link in the sidebar. The main area displays a table of distributions, with one row highlighted. The 'Domain Name' column for this row contains the value 'd2egm83spelhx.cloudfront.net', which is also highlighted with a red box. The right pane shows a browser window titled 'HTML Application'. The address bar shows the URL 'd2egm83spelhx.cloudfront.net'. The page content features a large blue cloud icon with a checkmark, the text 'AJ Labs Success!', and the message 'Your Website is ready, HTML/PHP app is up and running!'. Below this, there is a 'Next up' section with several links.

Note: You can map the CloudFront URL as a CNAME to existing custom domain

AWS Cloud Formation



- CloudFormation gives the ability to create the complete AWS environment using a template
- Describe the complete environment in JSON template
- Template: Collection of resources, attributes
- Stack: Once you feed this to AWS Cloud Formation, the resulting structure is called Stack. Once we delete the stack, the entire resources get deleted.
- For example, if you want to setup a RHCSA lab, at least you need 3 Linux machines, you planned to deploy that on AWS with proper VPC, Security Group, NACLs, etc. After all the configuration, you can create template out of this lab and share it with your friend so that they can deploy the same. It is a JSON export of existing configuration.
- We can create new template from Scratch using Designer or Export existing infrastructure as a Template.

AWS Reference Video: [CloudFormation](#)

[LAB] AWS CloudFormation - Exporting Existing Infra to Template (CloudFormer)

- Below is a small test infrastructure consisting of 3 EC2 Linux instances with PHP Running, 1 Load Balancer, NAT Instance, Management Windows.
- I have already deployed this in a new VPC



	VPC NAME	VPC CIDR		SUBNET NAME	SUBNET CIDR
FloudFormation	web-ap-vpc	10.57.0.0/16	Public	web-ap-vpc-public-subnet-ap-south-1a	10.57.1.0/24
				web-ap-vpc-public-subnet-ap-south-1b	10.57.2.0/24
			Private	web-ap-vpc-private-subnet-ap-south-1a	10.57.11.0/24

- Now I will create a template of the above setup using CloudFormer. Basically a JSON export of the entire setup.
- As part of the process, AWS will launch a CloudFormer EC2 instance with its own VPC and Subnets.
- By accessing that instance over internet, we can generate our template.

The screenshot shows the AWS CloudFormation console. On the left, there's a sidebar with 'CloudFormation registry' and 'Resource types'. The main area has a 'Create a CloudFormation stack' button highlighted with a red arrow. Below it, there's a 'Getting started' section with links like 'What is AWS CloudFormation?' and 'Documentation'. On the right, there's a 'Create stack' wizard with several steps. Step 1: 'Specify template' has 'Use a sample template' selected, with a red arrow pointing to it. Step 2: 'Specify stack details' has a 'Next' button at the bottom right. Step 3: 'Configure stack options' and Step 4: 'Review' are also visible.

CloudFormation > Stacks > Create stack

Specify stack details

Stack name: CloudFormer

Parameters

- Password: Password to log in to CloudFormer
- Username: Username to log in to CloudFormer
- VPCSelection: CreateNewVPC

Next

CloudFormation > Stacks > Create stack

Configure stack options

Name: CloudFormer

Permissions: IAM role: optional

Advanced options

- Stack policy**: Define the resources that you want to protect from unintentional updates during a stack update.
- Rollback configuration**: Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back.
- Notification options**
- Stack creation options**

Next

CloudFormation > Stacks > Create stack

Stack creation options

- Rollback on failure: Enabled
- Timeout: -
- Termination protection: Disabled

Capabilities

The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. Learn more

I acknowledge that AWS CloudFormation might create IAM resources.

Create stack

CloudFormation > Stacks > CloudFormer

CloudFormer (2020-08-10 00:08:32 UTC-0530) CREATE_COMPLETE

Timestamp	Logical ID	Status	Status reason
2020-08-10 00:08:32 UTC-0530	Route53Any	CREATE_IN_PROGRESS	-
2020-08-10 00:08:32 UTC-0530	VPCSubnet	CREATE_COMPLETE	-
2020-08-10 00:08:32 UTC-0530	CNRolePolicy	CREATE_COMPLETE	-
2020-08-10 00:08:32 UTC-0530	VPCAttachGateway	CREATE_COMPLETE	-
2020-08-10 00:08:32 UTC-0530	WebServerSecurityGroup	CREATE_COMPLETE	-
2020-08-10 00:08:32 UTC-0530	WebServerSecurityGroup	CREATE_IN_PROGRESS	-
2020-08-10 00:08:32 UTC-0530	VPCRouteTable	CREATE_COMPLETE	-
2020-08-10 00:08:32 UTC-0530	VPCSubnet	CREATE_IN_PROGRESS	-

New EC2 Experience

EC2 Dashboard

Launch Instance

Name	Instance ID	Instance Type
CloudFormer	i-0f3846fb9d2d26db1	t2.small
web-ap-1	i-0684972058c3d82	t2.micro
web-ap-2	i-0f79df90bc0aa05f29	t2.micro
web-ap-3	i-03df669cc429a25d	t2.micro
web-ap-vpc-management-windows	i-0fbba2f601639816	t2.micro
web-ap-vpc-nat-instance	i-03fd0dab42421fc	t2.micro

I (CloudFormer) Public DNS: ec2-13-
.amazonaws.com

Monitoring Tags

Public DNS (IPv4): ec2-13-233-225-157.ap-south-1.compute.amazonaws.com

IPv4 Public IP: 13.233.225.157

IPv6 IPs: -

Elastic IPs: -

AWS CloudFormer 0.41 (Beta)

Welcome to the AWS CloudFormation template creation utility. This utility helps you to create a CloudFormation template resources currently running in your account using a few simple steps. While the created template is complete and can be used in an AWS CloudFormation stack, it is a starting point for further customization. You should consider at least the following:

- Add Parameters to enable stacks to be customized at launch time.
- Add Mappings to allow the template to be customized to the specific environment.
- Replace static values with "Ref" and "Fn::GetAtt" functions to flow property data between resources where the property is dependent on the value of a property from a different resource.
- Remove any static IP addresses, availability zones and other environmental properties to create more generalized templates.
- Use CloudFormation metadata and on-host helper scripts to deploy files, packages and run commands on your instances.
- Customize any RDS Database, ElastiCache cluster or Redshift cluster passwords.
- Customize or add more stack outputs to list important information needed by the stack user.

Select the AWS Region: Asia Pacific (Mumbai)

When you press "Create Template" we will analyze all of the AWS resources in your account. This may take a little time.

Create Template

For more information on how to build a template see the [AWS CloudFormation User Guide](#). You can also check out our documentation demonstrating various template features.

By default, the account credentials will be used from the entries you typed in when AWS CloudFormer was created, overridden by clicking [here](#).

AWS CloudFormer

Template Information

Select the AWS region to introspect. The description is optional but will be displayed in the AWS Management Console to create a stack. You can optionally enter a filter for the resources. If you specify a filter, all resources with a value that contains the filter text will be selected automatically. Note that the filter is a case-insensitive match.

Template Description

Web Application CloudFormation Template

Resource Name Filter

Select resources matching filter: Select all resources in your account

AWS CloudFormer

Virtual Private Clouds

Select the Virtual Private Clouds (VPCs) to be included in the template. If you select a VPC, all VPC network and security resources will be selected for inclusion in the template by default, however, you can customize them in the next steps.

Amazon Virtual Private Clouds (VPCs)

- Select/Deselect all Amazon Virtual Private Clouds (VPCs)
- vpc-0a54928c8327672a8
- vpc-033c496e1786eff5
- vpc-04bdc71ef654d09fa
- vpc-8fd2cde7

AWS CloudFormer

Virtual Private Cloud Network Topologies

Select the Virtual Private Cloud (VPC) network configuration to be included in the template.

Amazon Virtual Private Cloud (VPC) Subnets

- Select/Deselect all Amazon Virtual Private Cloud (VPC) Subnets
- subnet-01c464a338f7fde8
- subnet-0866d60d4334b69db
- subnet-0ab5930416ad6673a
- subnet-05fb24895331dc89b
- subnet-9fd2d9f7
- subnet-2c18a457
- subnet-0f578b641a16621d4
- subnet-08c88e05fc0f5bb0
- subnet-0e2c3a137906707b9
- subnet-041e6b48

Amazon Virtual Private Cloud (VPC) Internet Gateways

- Select/Deselect all Amazon Virtual Private Cloud (VPC) Internet Gateways
- igw-023b73c22b2b4cd3b
- igw-061b410b81b0cf2e9
- igw-092ze0f2224637e48
- igw-b3e840db

Amazon Virtual Private Cloud (VPC) Customer Gateways

No resources found in this AWS account

AWS CloudFormer

Virtual Private Cloud Security Configuration

Select the Virtual Private Cloud (VPC) security configuration to be included in the template.

Amazon Virtual Private Cloud (VPC) Network ACLs

- Select/Deselect all Amazon Virtual Private Cloud (VPC) Network ACLs
- acl-033ddab1e853df233
- acl-065742a34c098439
- acl-592cea32
- acl-0cd80f0998e7652ee

Amazon Virtual Private Cloud (VPC) Route Tables

- Select/Deselect all Amazon Virtual Private Cloud (VPC) Route Tables
- rtb-00f3f110c9076081b
- rtb-0234e3f013c8cabf
- rtb-0ee57236e9b7cb8e0
- rtb-092dc0fd1ca5891db
- rtb-9e72aff5
- rtb-0026ce6fa03932d80
- rtb-06226285611b7489c

AWS CloudFormer

Network Resources

Select the network entry points to be included in the template. If you select an EIP or an ENI that has an instance or an Elastic Load Balancer that has instances or an Auto Scaling group associated with it, the instances or Auto Scaling group will be selected for inclusion in the template by default, however, you can customize them in the next step.

Elastic Load Balancers

No resources found in this AWS account

Amazon EC2 Elastic IP Addresses

No resources found in this AWS account

Amazon EC2 Network Interfaces

- Select/Deselect all Amazon EC2 Network Interfaces
- eni-055571d1cf7958114

Amazon CloudFront Distributions

- Select/Deselect all Amazon CloudFront Distributions
- d2egm83spelhfx.cloudfront.net

AWS CloudFormer

Managed Services

Select the managed services to include in the template. You can customize the content of the managed services in I

Auto Scaling Groups
No resources found in this AWS account

Elastic Beanstalk Applications
 Select/Deselect all Elastic Beanstalk Applications
 test

OpsWorks Stacks
No resources found in this AWS account

AWS CloudFormer

Managed Service Configuration

Select the configuration for managed services.

Auto Scaling Launch Configurations
 Select/Deslect all Auto Scaling Launch Configurations
 windows-web-servers-launch-configuration

Elastic Beanstalk Application Versions
 Select/Deselect all Elastic Beanstalk Application Versions
 test-source

Elastic Beanstalk Environments
No resources found in this AWS account

Elastic Beanstalk Configuration Templates
No resources found in this AWS account

OpsWorks Apps
No resources found in this AWS account

OpsWorks Layers
No resources found in this AWS account

OpsWorks Elastic Load Balancer Attachments

AWS CloudFormer

Compute Resources

Select the EC2 Instances and OpsWorks instances to be included in the template. If you select instances that are ass volumes, the volumes will be selected for inclusion in the template by default; however, you can customize them in

Amazon EC2 Instances
 i-03fd0dafb2421fbc
 i-06849720858cc3d82
 i-079df90bcaa05f29
 i-0fbba2f601639816
 i-03dff66cc428a25d

OpsWorks Instances
No resources found in this AWS account

AWS CloudFormer

Storage

Select the EBS Volumes, RDS Database Instances, ElastiCache clusters, Redshift clusters, DynamoDB tables, Simple buckets to be included in the template. Note that the master password of any RDS database instances will be han You should edit the final template with appropriate values.

Amazon Elastic Block Storage Volumes
No resources found in this AWS account

Amazon RDS Database Instances
No resources found in this AWS account

Amazon ElastiCache Cache Clusters
No resources found in this AWS account

Amazon Redshift Clusters
No resources found in this AWS account

Amazon DynamoDB Tables
 Select/Deslect all Amazon DynamoDB Tables
 aws-dynamodb-table1

Amazon S3 Buckets
 Select/Deselect all Amazon S3 Buckets
 elasticbeanstalk-ap-south-1-618927232701 (ap-south-1)

AWS CloudFormer

Storage Configuration

Select additional configuration for your Storage Services.

Amazon RDS DB Subnet Groups
 Select/Deselect all Amazon RDS DB Subnet Groups
 aws-rds-db-subnet-group
 default-vpc-8fd2cde7

Amazon RDS DB Parameter Groups
No resources found in this AWS account

Amazon ElastiCache Subnet Groups
No resources found in this AWS account

Amazon ElastiCache Parameter Groups
No resources found in this AWS account

Amazon RedShift Cluster Subnet Groups
No resources found in this AWS account

Amazon Redshift Cluster Parameter Groups
No resources found in this AWS account

AWS CloudFormer

Application Services

Select the SQS queues, SNS Topics and Kinesis Streams to be included in the template.

Amazon SQS Queues
No resources found in this AWS account

Amazon SNS Topics
 Select/Deslect all Amazon SNS Topics
 dynamodb
 sns-app-elb-monitoring-alarm

Amazon Kinesis Streams
No resources found in this AWS account

AWS CloudFormer

Intro DNS VPC Network Security Managed Services Managed Config Compute Storage Storage Config App Services Security Operational

Security Groups
Select the Security Groups and Policies to be included in the template. The default security groups in the account have been modified.

Amazon EC2 Security Groups

- Select/Deselect all Amazon EC2 Security Groups
- default
- web-ap-vpc-nat-security-group
- public-windows-security-group
- web-ap-vpc-windows-security-group
- web-ap-vpc-instance-security-group
- default
- web-ap-vpc-nat-instance-security-group
- default
- CloudFormer-WebServerSecurityGroup-1AVANL41MXNA3
- web-server-security-group
- default

Amazon RDS Security Groups

- Select/Deselect all Amazon RDS Security Groups
- default

Amazon ElastiCache Security Groups

Error enumerating resources: Use of cache security groups is not permitted in this API version for your account.

Amazon Redshift Cluster Security Groups

WS CloudFormer

Intro DNS VPC Network Security Managed Services Managed Config Compute Storage Storage Config App Services Security Operational

Operational Resources
Select the Auto Scaling Triggers to be included in the template.

to Scaling Policies
No resources found in this AWS account

to Scaling Scheduled Actions
No resources found in this AWS account

CloudWatch Alarms

- Select/Deselect all CloudWatch Alarms
- awsec2-l-0a7a89785fa55d872-CPU-Utilization

CloudTrail Trails
No resources found in this AWS account

AWS CloudFormer

Intro DNS VPC Network Security Managed Services Managed Config Compute Storage Storage Config App Services Security Operational

Summary
You have selected the following resources. We have automatically generated logical resource names for the template. You can assign your own names by editing them if you prefer. You can also select any output values that you want to return. Output values are displayed in the AWS management console when you select the stack in the AWS CloudFormation console. To change the logical name or to select output parameters, click on the modify link for each resource you want to change.

Amazon Virtual Private Clouds (VPCs)
vpc-033c496ef1786eff5 [Modify](#)

Amazon Virtual Private Cloud (VPC) Subnets

- subnet-0866d60d4334b69db [Modify](#)
- subnet-05fb24895331dc89b [Modify](#)
- subnet-0f578b641a16621d4 [Modify](#)

Amazon Virtual Private Cloud (VPC) Internet Gateways
igw-092e0f22224637e48 [Modify](#)

Amazon Virtual Private Cloud (VPC) DHCP Options
dopt-a21de7c9 [Modify](#)

Amazon Virtual Private Cloud (VPC) Network ACLs

AWS CloudFormation Template

You can save the AWS CloudFormation template in an existing S3 bucket in your account by selecting a bucket or button below. Alternatively, you can cut and paste the template content below and store it locally or in your source code. NOTE: If you move the template to an S3 bucket in a different AWS region from the one used to create the template, the new AWS region will likely fail since the template may have hardcoded values based on the original AWS region.

Template Name S3 Bucket

[Save Template](#) [Cancel](#)

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "vpc033c496ef1786eff5": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": "10.57.0.0/16",
        "InstanceTenancy": "default",
        "EnableDnsSupport": "true",
        "EnableDnsHostnames": "true",
        "Tags": [
          {
            "Key": "Name",
            "Value": "web-ap-vpc"
          }
        ]
      }
    },
    "subnet0866d60d4334b69db": {
      "Type": "AWS::EC2::Subnet",
      "Properties": {
        "CidrBlock": "10.57.11.0/24",
        "AvailabilityZone": "ap-south-1a",
        "VpcId": {
          "Ref": "vpc033c496ef1786eff5"
        },
        "Tags": [
          ...
        ]
      }
    }
  }
}
```

Congratulations!

You have created an AWS CloudFormation template and saved it to S3. You can now launch stacks using the template in the AWS Management Console. Note: If you launch the template, it will be launched in the same region as the S3 bucket in which you saved the template. This may not be the same region that you used to create the template.

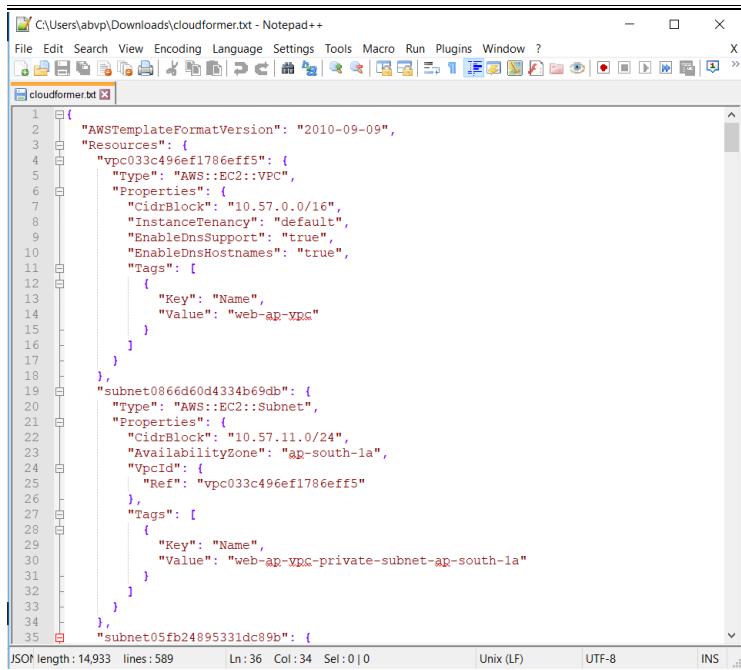
[Create Template](#) [Launch Stack](#)

Amazon S3 > my-cloud-bucket02

my-cloud-bucket02

Overview	Properties	Permissions	Management	Access points												
<input type="text" value="Type a prefix and press Enter to search. Press ESC to clear."/> Upload + Create folder Download Actions																
Viewing 1 to 2																
<table border="1"> <thead> <tr> <th>Name</th> <th>Last modified</th> <th>Size</th> <th>Storage class</th> </tr> </thead> <tbody> <tr> <td>AWS-Logo.png</td> <td>Aug 3, 2020 2:00:31 AM GMT+0530</td> <td>31.2 KB</td> <td>Standard</td> </tr> <tr> <td>cloudformer.template</td> <td>Aug 10, 2020 12:50:05 AM GMT+0530</td> <td>14.6 KB</td> <td>Standard</td> </tr> </tbody> </table>					Name	Last modified	Size	Storage class	AWS-Logo.png	Aug 3, 2020 2:00:31 AM GMT+0530	31.2 KB	Standard	cloudformer.template	Aug 10, 2020 12:50:05 AM GMT+0530	14.6 KB	Standard
Name	Last modified	Size	Storage class													
AWS-Logo.png	Aug 3, 2020 2:00:31 AM GMT+0530	31.2 KB	Standard													
cloudformer.template	Aug 10, 2020 12:50:05 AM GMT+0530	14.6 KB	Standard													
Viewing 1 to 2																

[Feedback](#) [English \(US\)](#)



```

1  {
2    "AWSTemplateFormatVersion": "2010-09-09",
3    "Resources": {
4      "vpc033c496ef1786eff5": {
5        "Type": "AWS::EC2::VPC",
6        "Properties": {
7          "CidrBlock": "10.57.0.0/16",
8          "InstanceTenancy": "default",
9          "EnableDnsSupport": "true",
10         "EnableDnsHostnames": "true",
11         "Tags": [
12           {
13             "Key": "Name",
14             "Value": "web-ap-vpc"
15           }
16         ]
17       },
18       "subnet0866d60d4334b69db": {
19         "Type": "AWS::EC2::Subnet",
20         "Properties": {
21           "CidrBlock": "10.57.11.0/24",
22           "AvailabilityZone": "ap-south-1a",
23           "VpcId": {
24             "Ref": "vpc033c496ef1786eff5"
25           },
26           "Tags": [
27             {
28               "Key": "Name",
29               "Value": "web-ap-vpc-private-subnet-ap-south-1a"
30             }
31           ]
32         }
33       },
34       "subnet05fb24095331dc89b": {
35     }
}

```

JSON length: 14,933 lines: 589 Ln: 36 Col: 34 Sel: 0 | 0 Unix (LF) UTF-8 INS .

- Delete the associated components in web-ap-vpc (VPC, Subnets, Security Groups, EC2s, ELB, Target Group, Internet Gateway).

[LAB] AWS Could Formation - Deploy from Template

Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready → Use a sample template Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL Upload a template file →

Upload a template file
Choose file cloudformer.txt →
JSON or YAML formatted file

S3 URL: <https://s3.ap-south-1.amazonaws.com/cf-templates-10y61651f1r-ap-south-1/202022U61-cloudformer.txt>

Cancel **Next**

Feedback English (US) Privacy Policy Terms of Use

Specify stack details

Stack name
Stack name
 Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters
There are no parameters defined in your template

Cancel Previous **Next**

Feedback English (US) Privacy Policy Terms of Use

Step 3 Configure stack options

Name Remove

Add tag

Permissions
Choose an IAM role to explicitly define how CloudFormation can create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses permissions based on your user credentials. [Learn more](#)

IAM role - optional
Choose an IAM role for CloudFormation to use for all operations performed on the stack.
 Sample-role-name

Advanced options
You can set additional options for your stack, like notification options and a stack policy. [Learn more](#)

Stack policy
Defines the resources that you want to protect from unintentional updates during a stack update.

Rollback configuration
Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#)

Notification options

Stack creation options

Cancel **Previous** **Next**

Feedback English (US) Privacy Policy Terms of Use

Stack policy

No stack policy
There is no stack policy defined

Rollback configuration

Monitoring time
CloudWatch alarm ARN

Notification options

No notification options
There are no notification options defined

Stack creation options

Rollback on failure
Enabled
Timeout
Termination protection
Disabled

Quick-create link

Cancel Previous **Create change set** **Create stack**

Feedback English (US) Privacy Policy Terms of Use

CloudFormation > Stacks

Stacks (1) Delete Update Stack actions

Filter by stack name Active View nested

Stack name	Status	Created time
web-app-stack	CREATE_COMPLETE	2020-08-10 01:00

Feedback English (US) Privacy Policy Terms of Use

New EC2 Experience Tell us what you think

EC2 Dashboard New

Events New

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts New

Capacity Reservations

Images

AMIs

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Launch Instance Connect Actions

Instance State: Running

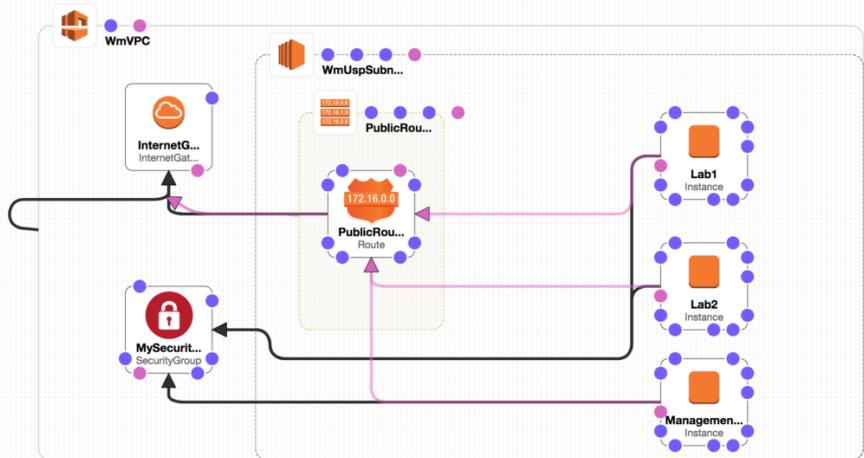
Name	Instance ID	Instance Type
web-ap-1	i-03130693b0cd87b66	t2.micro
web-ap-2	i-0ccfd90ef8be62fc0	t2.micro
web-ap-3	i-082d8205b3a6569c	t2.micro
web-ap-vpc-management-windows	i-0cd84965e6a0185a0	t2.micro
web-ap-vpc-nat-instance	i-0124d7d9904496a1	t2.micro

Select an instance above

Feedback English (US) Privacy Policy Terms of Use

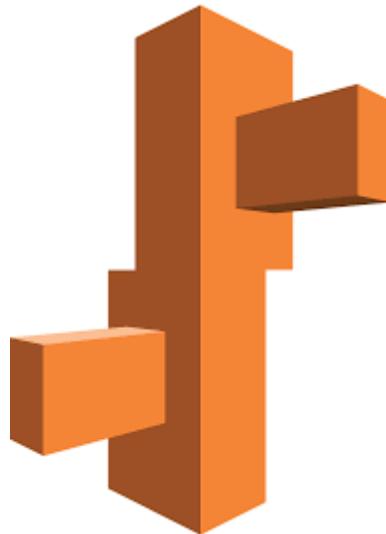
Note: I was able to get the EC2 instanced from the template, but for some reason the ELB not created.

AWS Could Formation - Designer



- AWS CloudFormation Designer is a graphic tool for creating, viewing, and modifying AWS CloudFormation templates.
- With Designer, you can diagram your template resources using a drag-and-drop interface, and then edit their details using the integrated JSON and YAML editor.
- Whether you are a new or an experienced AWS CloudFormation user, AWS CloudFormation Designer can help you quickly see the interrelationship between a template's resources and easily modify templates.

Elastic Beanstalk



- Fully managed PaaS Web application hosting platform in AWS
- Built-in Monitoring tools and Security
- Automatically scale-up
- You have access to the underlying AWS resources at any time
- Support versioning
- Environment (EC2s, Security Group, etc.)
- Environment Tier: Web Server Environment or Worker Environment
- Environment Health: Color coded indication of the health
- Beanstalk Web Server Environment contains,
 - Elastic Load Balancer
 - Autoscaling Group
 - EC2 Instances
 - Host Manager: Runs on every EC2s, provides performance statistics, events, application logs, etc. in CloudWatch
 - Security Group: Allows HTTP traffic
- Beanstalk Worker Environment contains,
 - Worker handles background tasks which are time intensive and processor intensive, send email notification, etc.
- Web server tier Create SQS message and place in SQS queue, in Worked environment has Daemon that pulls the SQL queue.

[LAB] Elastic Beanstalk

The screenshot shows the AWS Elastic Beanstalk landing page. At the top right, there is a prominent orange 'Create Application' button. The page features sections like 'How it works', 'Benefits and features', and 'Getting Started'. A red arrow points to the 'Create Application' button.

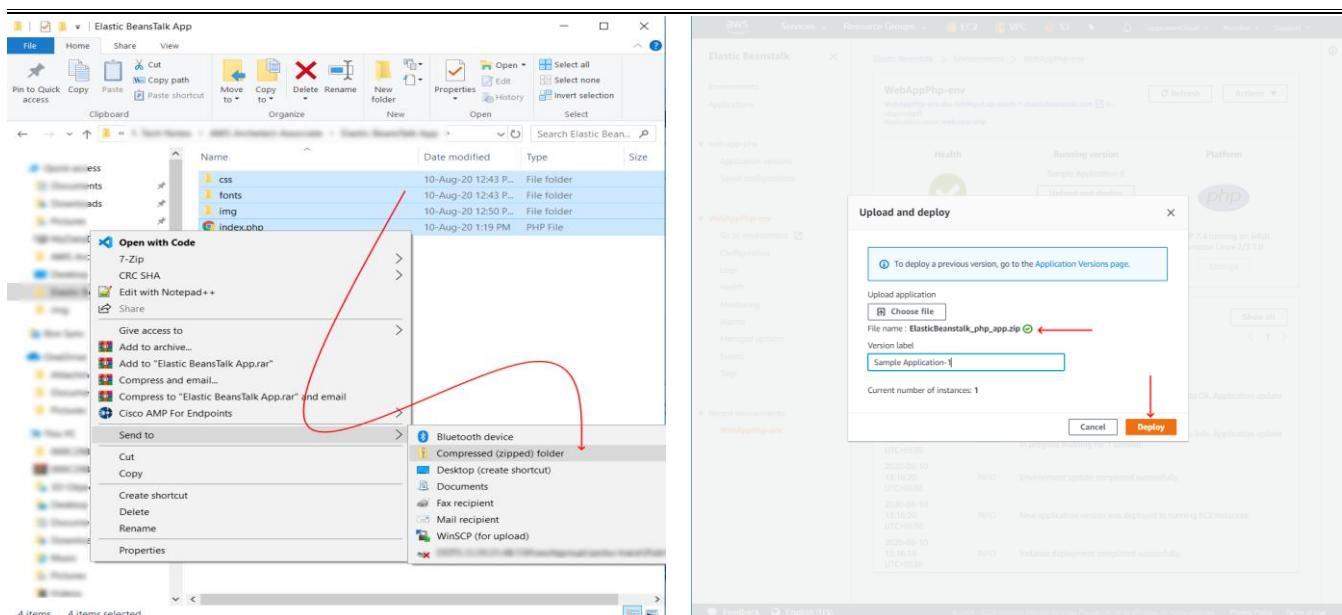
This screenshot shows the first step of the 'Create Application' wizard. It asks for the 'Application name' (set to 'web-app-php') and 'Application tags'. A red box highlights the 'Use: web-app-php' tag entry. The 'Create application' button at the bottom is also highlighted with a red arrow.

This screenshot shows the second step of the 'Create Application' wizard. It focuses on 'Platform' settings (PHP 7.4) and 'Application code' options (Sample application selected). A red box highlights the 'Platform' section, and another red arrow points to the 'Create application' button.

This screenshot shows the third step of the 'Create Application' wizard. It displays a progress bar and a log window showing the creation process: 'Creating WebAppPhp-env' and 'This will take a few minutes...'. A red arrow points to the 'Create application' button.

This screenshot shows the EC2 Instances page. It lists instances and highlights 'WebAppPhp-env'. Below, a detailed view of the instance shows its ID, state (running), type (t2.micro), and IP address (15.207.156.45). A red arrow points to the 'Instances' link in the sidebar.

This screenshot shows the Elastic Beanstalk Environment page for 'WebAppPhp-env'. It displays health status (OK), running version (Sample Application), and platform (PHP 7.4). A red arrow points to the 'Upload and deploy' button.



App Download Link: [Elastic Beanstalk Sample PHP App](#)

Other AWS Services (Theory Only)



- In this section, we will discuss about some AWS Feature Services, no labs are included since those require some prerequisites (mainly on coding section).
- These services / features are intended for developers.

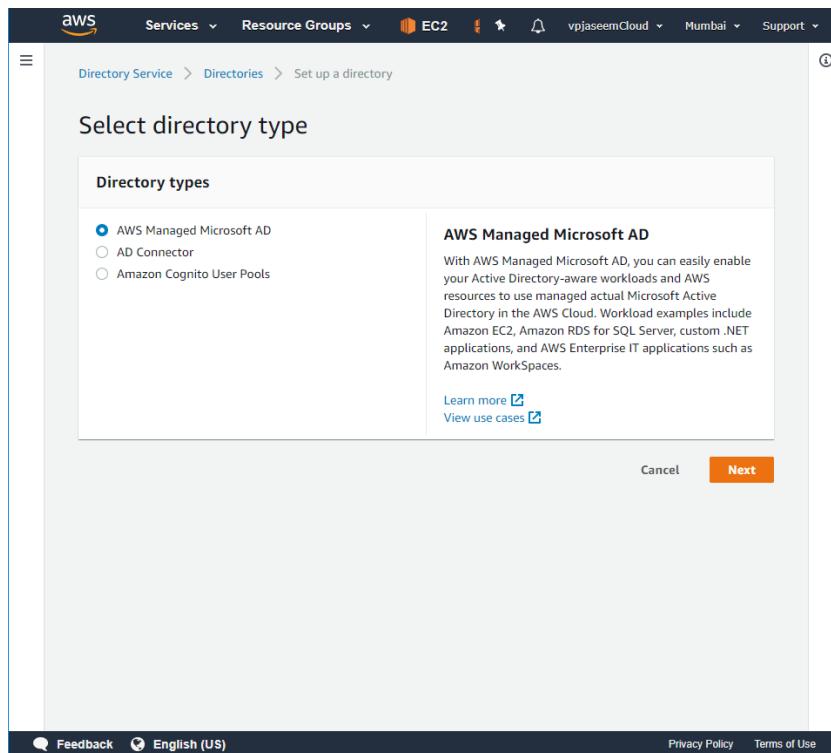
AWS Cognito

- For user management in web app (User signup, notification, etc.)
- Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily.
- Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.

AWS Reference: [Amazon Cognito](#)

AWS Directory Services

- Fully Managed Microsoft AD
- AD Connector to authenticate via on-prem AD
- Authentication from AD (AWS Managed or AD Connector to on-prem) and authorization via IAM
- Use Cognito to authenticate



AWS Reference: [AWS Directory Service](#)

Single Sign-On (SSO)

- Grant access to your AWS accounts and any business applications that support SSO using SAML 2.0.
- Users simply sign in once to a personalized user portal using their directory credentials and can then access all their assigned applications without additional sign-in prompts.
- Multiple AWS Accounts can be authenticated.

AWS Reference: [AWS Single Sign-On](#)

Storage Gateway

- The AWS Storage Gateway is a service connecting an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and AWS's storage infrastructure.
- File Gateway, Volume gateway, Tape Gateway are available.
- You can deploy an on-premise local VM that acts as a gateway to S3 storage.
- AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage.
- It is like a DropBox software installed on local computer.

AWS reference: [AWS Storage Gateway](#)

API Gateway

- Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.
- APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services.

Amazon Reference: [API Gateway](#)

YouTube: [AWS API gateway to Lambda Function](#)

AWS OpsWorks

- Fully managed configuration management service uses Chef and Puppet.
- Configuration management service helps to operate and change configurations instantly.
- For example, you want to change properties of 1000 servers running at a time, OpsWorks can be used.

Amazon Reference: [Aws OpsWorks](#)

YouTube: [AWS OpsWorks Demo](#)

RedShift

- Used for OLAP (Online Analytical Processing) systems or Data warehousing.
- Columnar databases, we usually select columns in data warehousing scenarios.

Amazon Reference: [What is Amazon Redshift?](#)

Kinesis

- Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information.
- Used for AI applications.
- CCTV Camera devices can stream videos to Kinesis directly and can be used for Machine Learning.

AWS Reference: [Kinesis](#)

YouTube: [AWS Kinesis Tutorial](#)

Elastic File System (EFS)

- Fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources.
- Replicated across AZs in a region
- Once EFS Volume can be mounted to N number of EC2 instances like a shared drive
- Could be mounted to on premise server as well (Over VPN or Direct Connect)
- No sizing to be done
- Using Amazon EFS with Microsoft Windows-based Amazon EC2 instances is not supported.

AWS Reference: [What is EFS?](#)

FXs

- Amazon FSx for Windows File Server provides fully managed Microsoft Windows file servers, backed by a fully native Windows file system.
- This is the NFS for Windows.

AWS Reference: [Amazon FSx](#)

Terra Forms

- We can template your infrastructure, compatible with AWS, Google Cloud and Microsoft Azure.
- This is like Cloud Formation template.
- Use Infrastructure as Code to provision and manage any cloud, infrastructure, or service.

About the Author:



Abdul Jaseem VP, a Network Consulting Engineer who has worked with complex IT infrastructure in multiple enterprise environments. He is currently employed with Cisco Systems India Pvt. LTD. TAC Team. He has seven plus years of industry experience in Cisco Collaboration, Networking, Cloud Computing, Virtualization and Security.

Presently lives in Bangalore and basically from a small village called Karuvarakundu in the state of Kerala, India. He loves conducting industry standard IT trainings for people across the globe.

You can connect him via Email: vpjaseem@gmail.com; [LinkedIn](#); [YouTube](#)