STUDY – AWS Developer Associate Certification Exam Tips
Register for the exam(USD 150, conducted online): www.webassessor.com
To reschedule exam date, do it in first 72hrs, else 50% exam fee will be charged and it can be done by emailing awscertification@aws.com

S3-Simple Storage Service
Dynamo DB
SQS-Simple Queue Service
SNS-Simple Notification Service
SWF-Simple Workflow Service
Deployment via AWS Beanstalk, cloud formation
--AWS Technology
--AWS API
--Security Best Practices
--Automation and Deployment tools
--Consistency Model
--AWS SDK
--Stateless and loosely couple apps
--REST APIs
--RDS-Relational Database Services
--messaging and queuing service

integrate application and data using Kinesis,sqs,sns,swf
IAM - Identity and Access Management, web IAM,authentication, authorisation,sso,google authenticator,setup usres, roles and groups
EC-Elastic Cache, cloud front
EC2-Elastic Compute Cloud, Elasticity and Scalability

Deployment and Security
-Cloud Security Best Practices
-Architecture
--shared security responsibility Model
--platform complainces
--security attributes(workload balancers)
--IAM
--VPC (Virtual Private Cloud)
--CIA, AAA< Ingress vs Egress

AWS Global Infra
Regions - Geographical Area
Availability Zones - like data centers in a region
Edge Location - are CDN (Content Delivery Network) end points for clound front

Networking and Content Delivery
-VPC-Virtal Private Cloud
-Route53-DNS Service
-Cloud Front-cache and edge location
-Direct Connect-dedicate lines into AWS for faster data transfer, generally used by corporates
-EC2-Elastic Compute Cloud, Virtual Machines
-EC2 Container Service-Docker Container, high performance apps
-Elastic Beanstalk-code build and deployment over AWS platform

LAMBDA Service-its a serverless service, via aws api, programmer can access these services and make great apps like AI app Alexa, Lex, Rekognition
-SSH,RDP,OS
-LightSail-service out of box for cloud, like building a CMS, wordpress, joomla sites, deployment and customize it

Storage
-S3-its object based storage in cloud, has virtual disk, e.g. dropbox uses S3 platform and make AWS S3 calls via its interface
-Glacier-files archival place, its very very low cost
-EFS-Elastic File Service-its a file based storage, can install DBs and apps into it, like a file share
-gateways-connect mechanism, VM images

Databases
-RDS-diff RDBMs like mysql, postgres, sql server, oracle

-Dynamo DB-no sql, very very scalable
-Redshift-AWS Dataware housing solution, stores hostorical database as well
-Elastic Cache- in memory database solution,data cache in cloud,for highly fast system like share market real time trending and analysis, fx real time processing

Migration
-snowball-copy TBs of data from one system to another, edge compute capacity
-DMS-DB Migration Service,Reinvent2015
-SMS-Server Migration Service-VMWare to AWS, 50 connect one once

Analytics
-Athena-sql queries into S3 and convert flat files as if they are RDBMS tables and queries like sql tables
-EMR-Elastic Map Reduce-process and analyse compute larest size of flat files
-Big Data-hadoop implementation,spark,scala, presto,flunk
-cloud search-elastic search,build search engines, e.g. angola

Kinesis-streams massive real time data like twittre, facebook feeds, social media analysis
-Data Pipelines-data movement from one service to another e.g. S3 to dynamodb
-quick signal-BI Tool

Inspector-inspect VMs
Certificate Manager-manages and install SSL Certificates
active Directory-AD Services
WAF-web application firewall, protects from app protection,cross site scripting,sql injections

Artifacts-iso Certificates, doc manager, archives, guidelines and compliance docs

Management tools
-cloud watch-ram, hdd, infra, services, cpu utilisation monito
-cloud formation-way to turning infra into software and api calls, docs desc aws env, manage and monitoring, failover view and management, load balancer view
-cloud trail-audit aws resources, aws env changes,
-OPS works-cheff for env build and Automation
-config-env monitory, alerting, notification, env changes

Service Catalog
-Trusted Advisor-scanning env and gives security tool tips

Application Services
-Step Function-visualize micro services
-SWF-way of coordinating manual and automated tasks, create workflows via wizard
-API Gateways-access backend services like lmabda, rest call for app development
-App Streaming
-Elastic Transcoder-video format conversion as per OS detected by the service

Developer tools
-code commit - git on aws
-code build, code deploy, code pipeline(ver of data)

Mobile Services
-mobile build for mobile apps, CDN, push notification,mobile services,Analytics, UI flow management
-cognito-sign in,sso, stores sign in related meta data
-device farm-farm of diff devices around the world so that the app developer can test regrously on these device platforms
-mobile analytics & pinpoint-analyse traffic, like google analytics

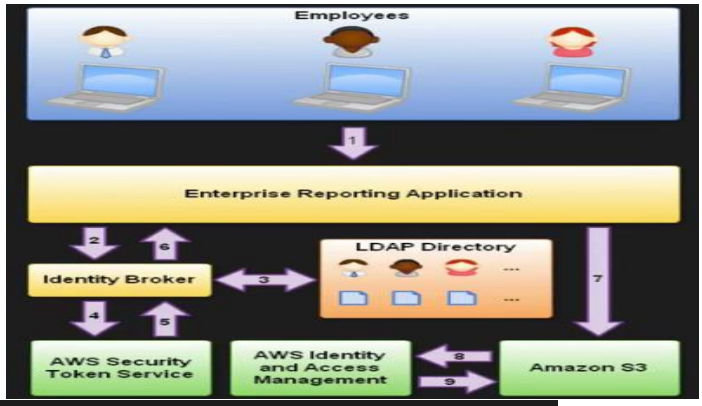
Business Productivity-workdocs, workmail
IoT-reinvent 2015-gather more info around this
AI-gather more info
-alexa-voice service e.g. aws echo system via lambda service
-lex-API to alexa lambda service
-polly-text to speech engine by aws
-machine learning-process volume of data for pattern analysis, behavioural analysis, face detection etc e.g. rekognition api by aws
-rekognition-picture analysis by aws, it can identify face, mountain, cycle and other various objects in a picture

messaging
-SNS,SQS,emails, other notification services

IAM
-setting users, groups and roles which can be assigned to users login and specifics they would have access to on AWS machines
-maf(multi factor authentication): google authenticator
-->access keyid and access keys can only be used when you programatically access the aws cloud
-->user and pwd cannot be used programatically to login to aws (only be used in aws-console)

STS-Security Tokens Service-grants users limited and temporary access to aws resources
-Federation (Active Directory)-security assertion markup language (SAML), doesnt need to be in IAM, SSO login possible without being in IAM. federation means combining or joining list of users from one domain to another domain
this is done by identity broker
identity store-services like AD, facebook, google etc
identities-user of a service like identity
-Federation with Mobile Apps-use fb,tw,google id to login
-Federation account access-access to resources in another

You are hosting a company website on some EC2 web servers in your VPC. Users of the website must log in to the site which then authenticates against the companies active directory servers which are based on site at the companies head quarters. Your VPC is connected to your company HQ via a secure IPSEC VPN. Once logged in the user can only have access to their own S3 bucket. How do you set this up?

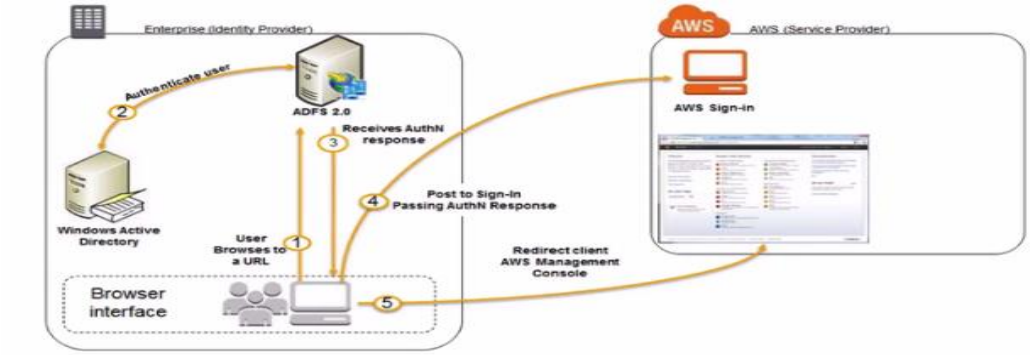


1. Employee enters their username and password
2. The application calls an Identity Broker. The broker captures the username and password
3. The Identity Broker uses the organization's LDAP directory to validate the employee's identity
4. The Identity Broker calls the new GetFederationToken function using IAM credentials. The call must include an IAM policy and a duration (1 to 36 hours), along with a policy that specifies the permissions to be granted to the temporary security credentials
5. The Security Token Service confirms that the policy of the IAM user making the call to GetFederationToken gives permission to create new tokens and then returns four values to the application: An access key, a secret access key, a token, and a duration (the token's lifetime)
6. The Identity Broker returns the temporary security credentials to the reporting application.
7. The data storage application uses the temporary security credentials (including the token) to make requests to Amazon S3.
8. Amazon S3 uses IAM to verify that the credentials allow the requested operation on the given S3 bucket and key
9. IAM provides S3 with the go-ahead to perform the requested operation.

1. Develop an Identity Broker to communicate with LDAP and AWS STS
2. Identity Broker always authenticates with LDAP first, THEN with AWS STS
3. Application then gets temporary access to AWS resources

1. Develop an Identity Broker to communicate with LDAP and AWS STS.
2. Identity Broker always authenticates with LDAP first, gets an IAM Role associate with a user.
3. Application then authenticates with STS and assumes that IAM Role
4. Application uses that IAM role to interact with S3

Active directory federation
Authentication via AD is possible in SAML format(security assertion markup language)
AD authentication happens first followed by security token by STS



1. The flow is initiated when a user (let's call him Bob) browses to the ADFS sample site (https://*Fully.Qualified.Domain.Name.Here*/adfs/ls/IdpInitiatedSignOn.aspx) inside his domain. When you install ADFS, you get a new virtual directory named adfs for your default website, which includes this page
2. The sign-on page authenticates Bob against AD. Depending on the browser Bob is using, he might be prompted for his AD username and password.
3. Bob's browser receives a SAML assertion in the form of an authentication response from ADFS.
4. Bob's browser posts the SAML assertion to the AWS sign-in endpoint for SAML (https://signin.aws.amazon.com/saml). Behind the scenes, sign-in uses the AssumeRoleWithSAML API to request temporary security credentials and then constructs a sign-in URL for the AWS Management Console.
5. Bob's browser receives the sign-in URL and is redirected to the console.

From Bob's perspective, the process happens transparently. He starts at an internal web site and ends up at the AWS Management Console, without ever having to supply any AWS credentials.

Web Identity Federation with mobile apps

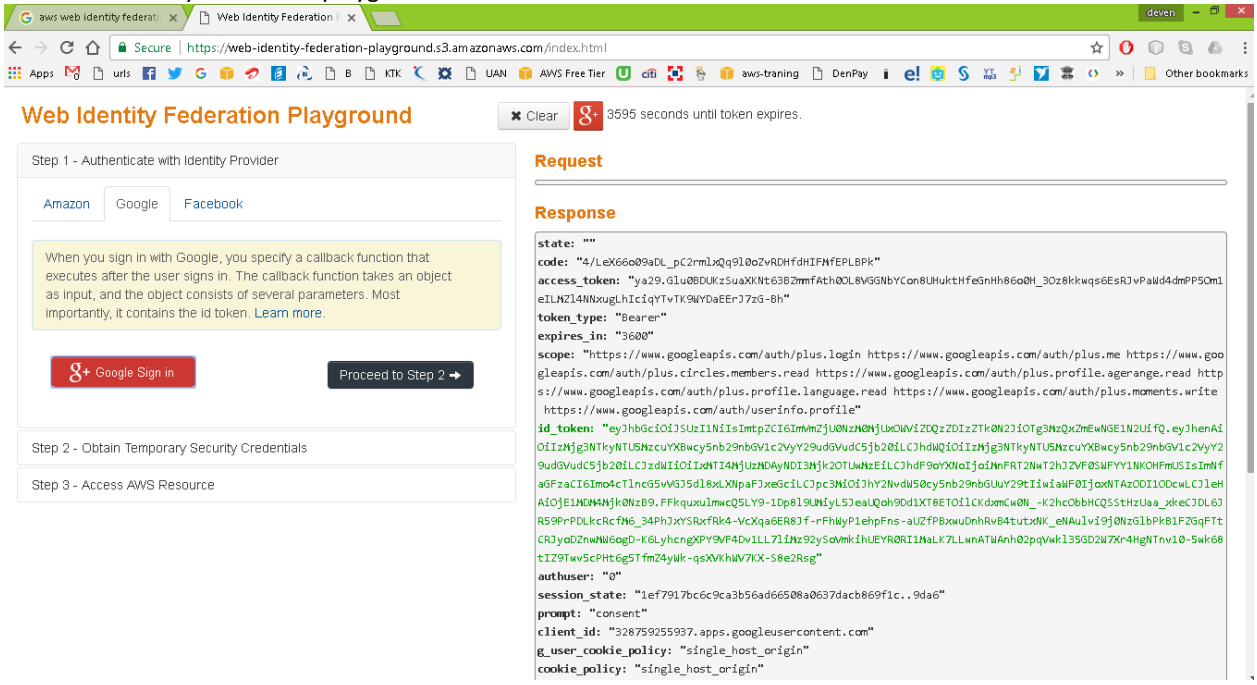
ARN-Amazon Resource Name

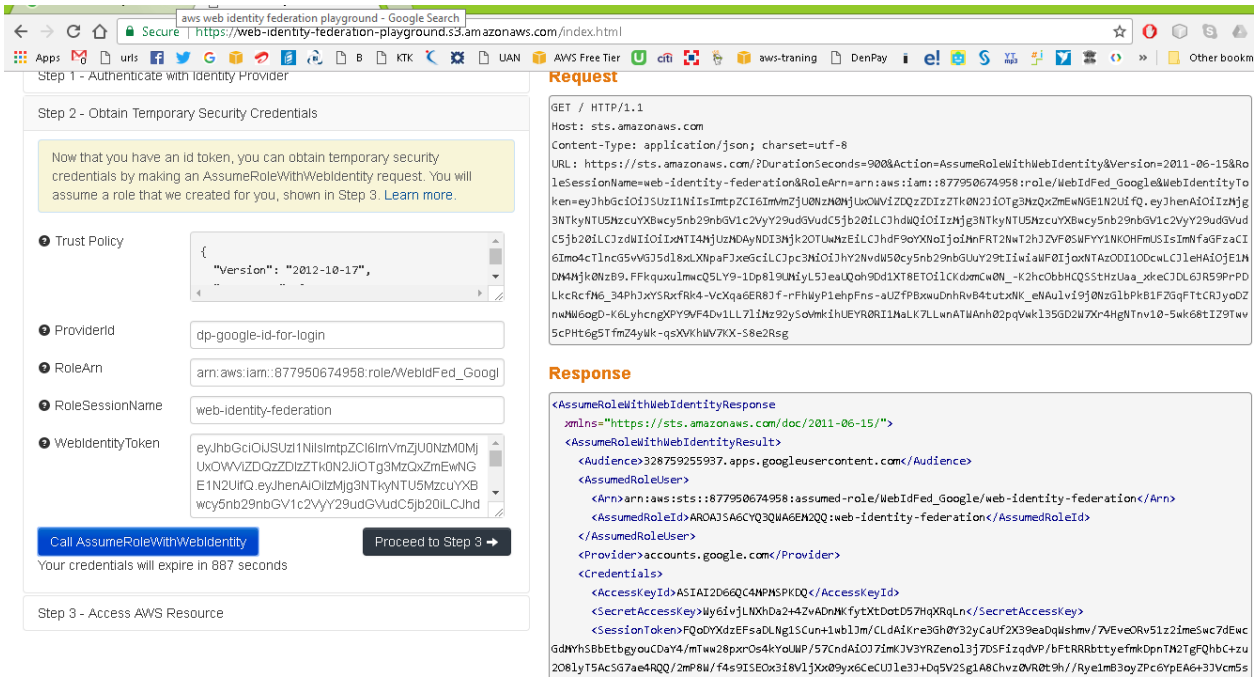
You can authenticate your application using applications like google, facebook, twitter etc to AWS account

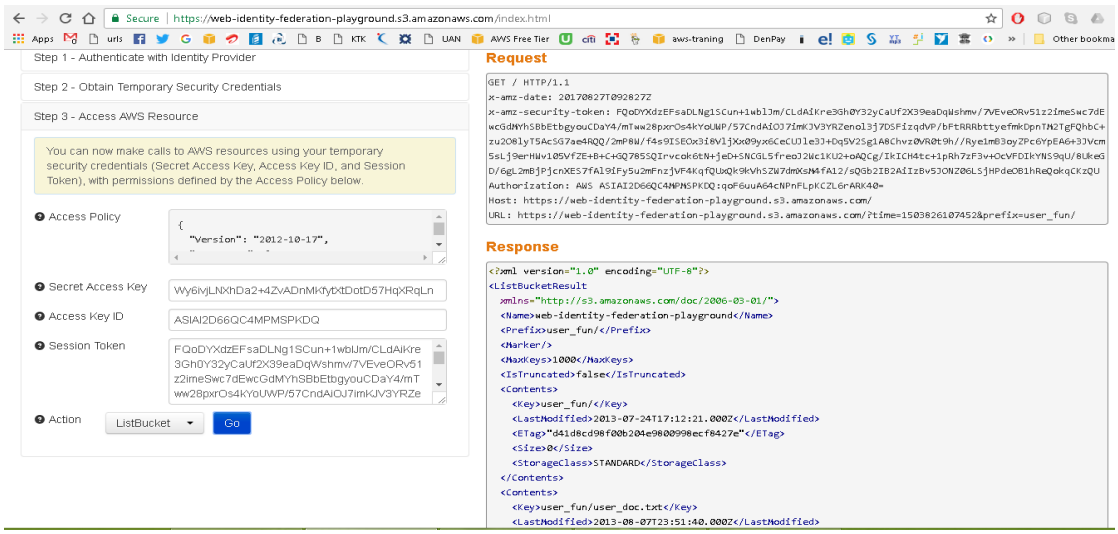
To set it up you need to code

Not for exam but you can follow: http://aws.amazon.com/articles/4617974389850313

See "web identity federation playground"







AWS sign in end point: https://signin.aws.amazon.com/saml

EC2-Elastic Compute Cloud

-Its resizeable compute capacity in cloud. It helps scaling up or down really fast. Virtual machine in the cloud

-Pay only to the capacity that you need as part of server needs

--OnDemand Pricing: hourly usage, no commitment whatsoever, you can destroy your server instance after usage

--Reserved: instance is preserved for some period say 1yr, 3yr etc

--spot pricing: bid on the price you want for instance capacity, save greatly, if apps have flexible start and end times, e.g.pharma, market research companies, genome companies generally goes for it.it also useful when you need really high computing urgently e.g. analysing doller being valued/devalued based on certain economic crisis in say EUR

-Dedicated Host pricing: physical EC2 instances dedicated for you, when you would have regulatroy data or demand and you want to move to cloud. E.g. big corporates generally go for it

-failover is managed by cloud itself

If the Spot instance is terminated by Amazon EC2, you will not be charged for a partial hour of usage. However, if you terminate the instance yourself, you will be charged for any hour in which the instance ran

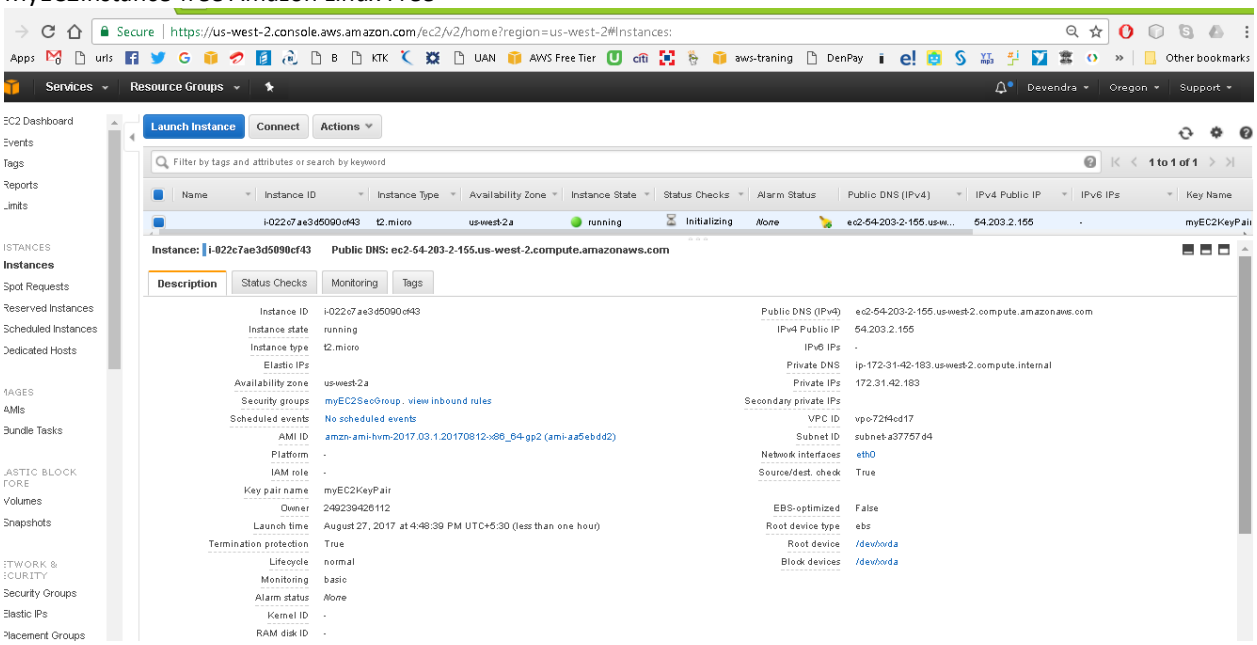
DIRTMCGFPX –ec2 instance types (Dr Mc GIFT PX). Exam will have scenario and we need to choose EC2 instances best suits in it

EC2 Instance Types

Family	Speciality	Use case
D2	Dense Storage	Fileservers/Data Warehousing/Hadoop
R4	Memory Optimized	Memory Intensive Apps/DBs
M4	General Purpose	Application Servers
C4	Compute Optimized	CPU Intensive Apps/DBs
G2	Graphics Intensive	Video Encoding/ 3D Application Streaming
I2	High Speed Storage	NoSQL DBs, Data Warehousing etc
F1	Field Programmable Gate Array	Hardware acceleration for your code.
T2	Lowest Cost, General Purpose	Web Servers/Small DBs
P2	Graphics/General Purpose GPU	Machine Learning, Bit Coin Mining etc
X1	Memory Optimized	SAP HANA/Apache Spark etc

EBS-Block based storage for DBs, OS, Applications etc, they are placed in a specific availabilty zones
GP2-General Purpose SSD – less than 10000 IOPS
Provisioned SSD-more than 10000 IOPS e.g. large RDBMSes, large nosql dbs
ST1-magnetic storage, very very high throughput optimised, e.g. used in big data, log processing, data warehpousing
SC1-cold hard disk storage, low cost, infrequently accessed work loads, e.g. a file server
HDD Magnetic Storage-very very cheep storage, very very less freq used
You cannot use 1 EBS volume to multiple EC2 instance instaead use EFS

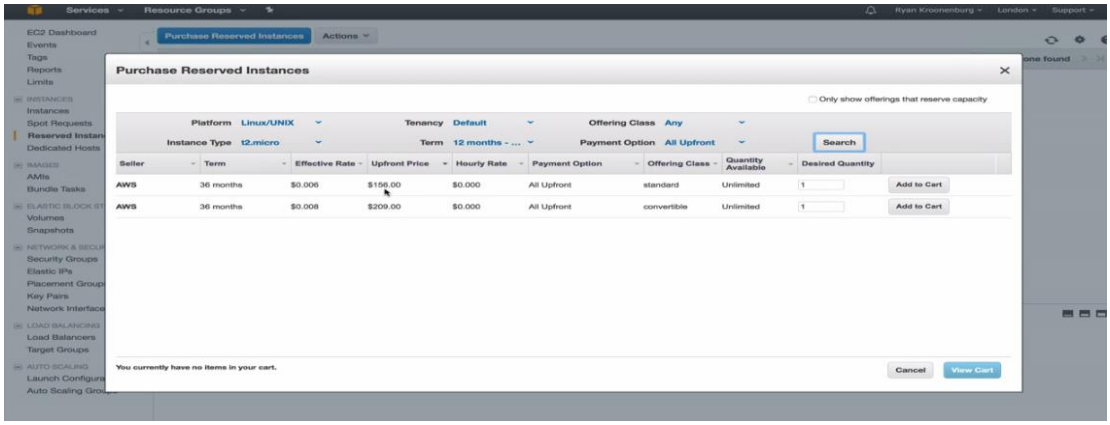
MyEC2Instance-free Amazon Linux Free



Login to EC2instance via SSH util or putty

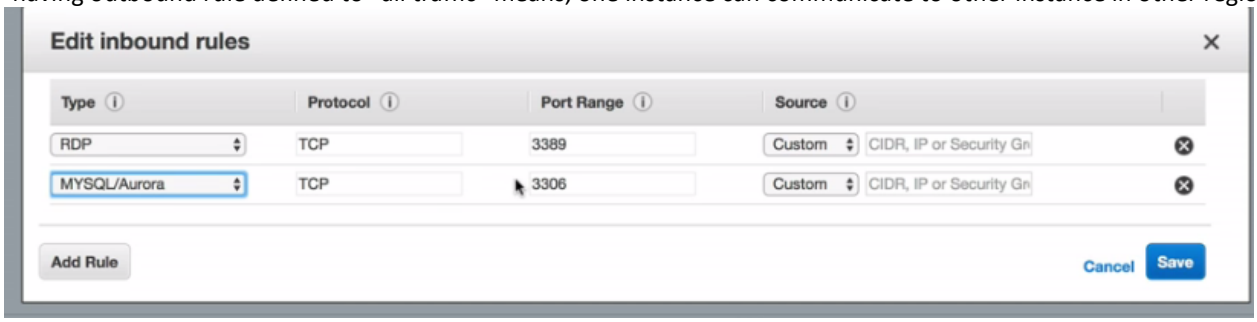
```
Ryans-iMac:SSH ryankroonenburg$ ssh ec2-user@52.56.97.82 -i MyEC2KeyPair.pem
The authenticity of host '52.56.97.82 (52.56.97.82)' can't be established.
ECDSA key fingerprint is SHA256:eGUphgMhgwBjMnSVaC0zOaoHFIA+15rPyoaQWq5Zh5s.
Are you sure you want to continue connecting (yes/no)?
```

Purchase reserved instance



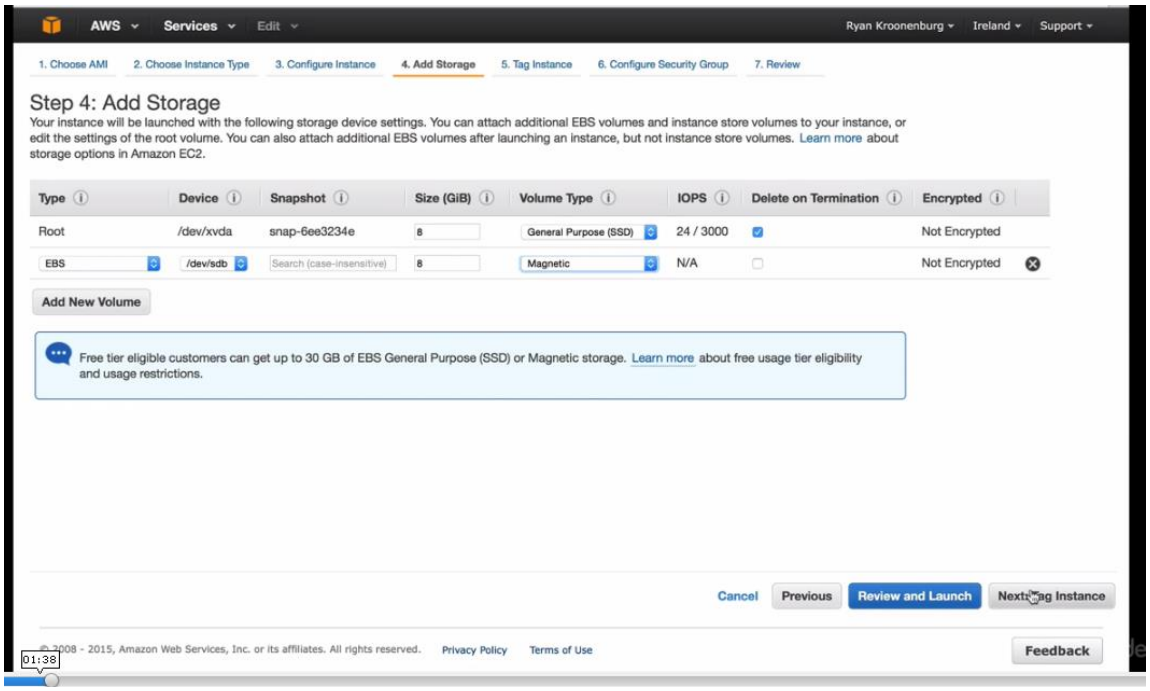
- Termination Protection is turned off by default, you must turn it on.
- On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated.
- EBS Root Volumes of your DEFAULT AMI's cannot be encrypted. You can also use a third party tool (such as bit locker etc) to encrypt the root volume, or this can be done when creating AMI's (lab to follow) in the AWS console or using the API.
- Additional volumes can be encrypted.

-Security Group is a virtual firewall, an instance is associate with one or more security groups
-Any changes made to security group (http, https, ssh, tcp) is made effective immedidately
-Inbound rules if defined means they will definately be outbound automatically by AWS irrespective of it being defined or not. These are statefull rules
-everything in AWS instance is blocked by default, by defining inbound rules, we allow specific type of web requests to AWS server and port like http call
-having outbound rule defined to "all traffic" means, one instance can communicate to other instance in other region as well



- All Inbound Traffic is Blocked By Default
- All Outbound Traffic is Allowed
- Changes to Security Groups take effect immediately
- You can have any number of EC2 instances within a security group.
- You can have multiple security groups attached to EC2 Instances
- Security Groups are **STATEFUL.**
  - If you create an inbound rule allowing traffic in, that traffic is automatically allowed back out again.
- You cannot block specific IP addresses using Security Groups, instead use Network Access Control Lists.
- You can specify allow rules, but not deny rules.

Change the EBS volume or attach a new volume type e.g. install a mysql DB or other application group by attachig a megnatic EBS volume to it



Changing EBS Volume to magnetic and mountid device to a directory

```
https://aws.amazon.com/amazon-linux-ami/2017.03-release-notes/
1 package(s) needed for security, out of 1 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-47-67 ~]$ sudo su
[root@ip-172-31-47-67 ec2-user]# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdb    202:16   0   8G  0 disk
[root@ip-172-31-47-67 ec2-user]# mkfs -t ext4 /dev/sdb
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 2097152 4k blocks and 524288 inodes
Filesystem UUID: 6c314f76-03cc-4089-8cd6-fcbfca31f950
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@ip-172-31-47-67 ec2-user]# mkdir dpfs
[root@ip-172-31-47-67 ec2-user]# ls
dpfs
[root@ip-172-31-47-67 ec2-user]#
[root@ip-172-31-47-67 ec2-user]#
[root@ip-172-31-47-67 ec2-user]# mount /dev/sdb /dpfs
mount: mount point /dpfs does not exist
[root@ip-172-31-47-67 ec2-user]# mount /dev/sdb dpfs
[root@ip-172-31-47-67 ec2-user]#
[root@ip-172-31-47-67 ec2-user]#
[root@ip-172-31-47-67 ec2-user]#
[root@ip-172-31-47-67 ec2-user]# lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdb    202:16   0   8G  0 disk /home/ec2-user/dpfs
[root@ip-172-31-47-67 ec2-user]#
```

Data persists in the magnetic drive even if we umount it from a dir

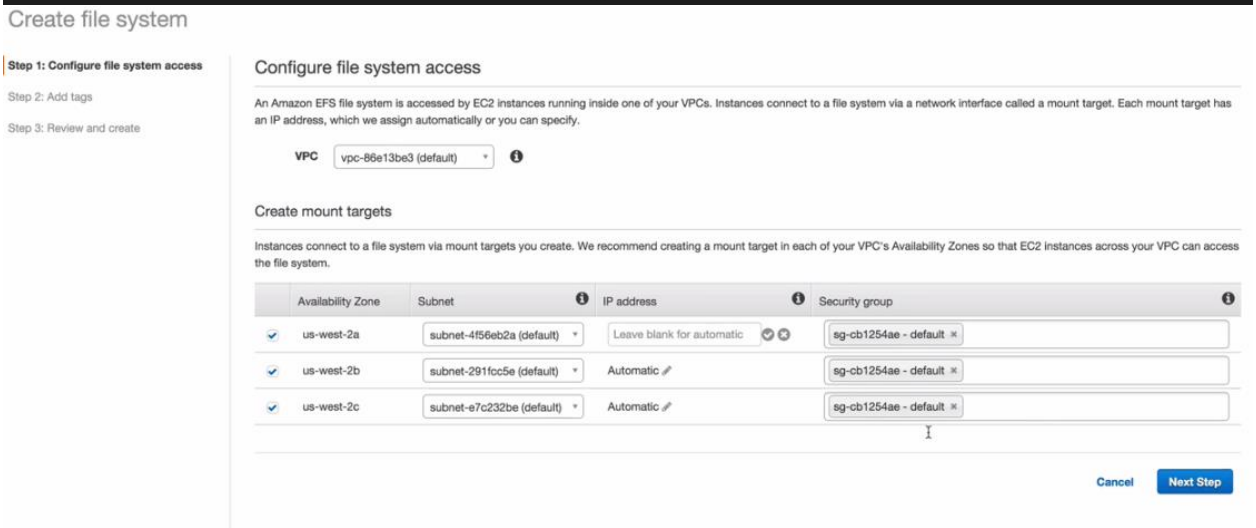
-Upgrading the volumes-stop instance(only for magnetic standard)-rest can be upgraded on the fly but best practice to stop the instance and then change, unmount from existing instance, take a snapshot,go to snapshot section, upgrade ec2 EBS instance, come back to volume, attach it to the ec2 running instance

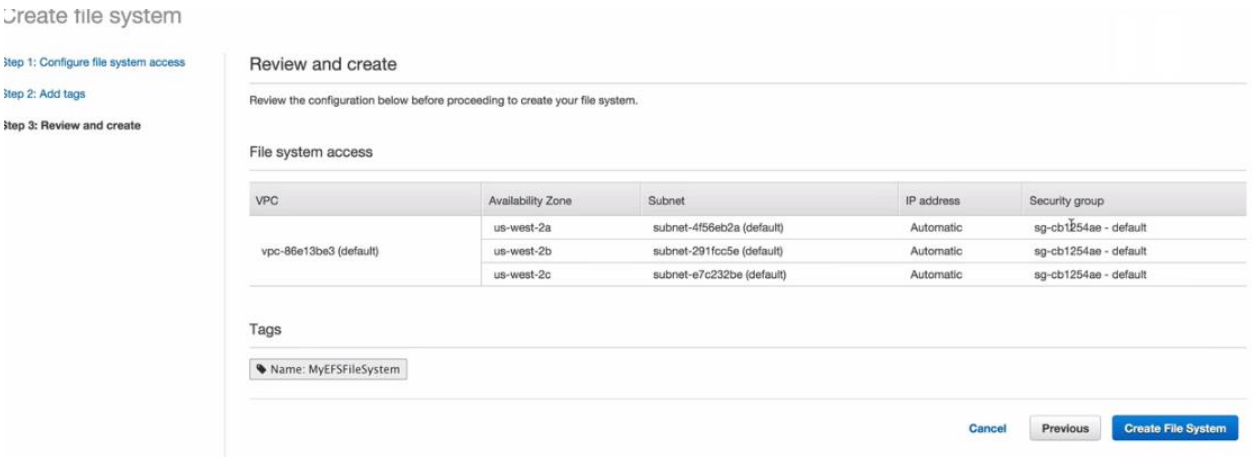
- EBS Volumes can be changed on the fly (except for magnetic standard).
- Best practice to stop the EC2 instance and then change the volume
- You can change volume types by taking a snapshot and then using the snapshot to create a new volume
- If you change a volume on the fly you must wait for 6 hours before making another change
- You can scale EBS Volumes up only
- Volumes must be in the same AZ as the EC2 instances

EFS

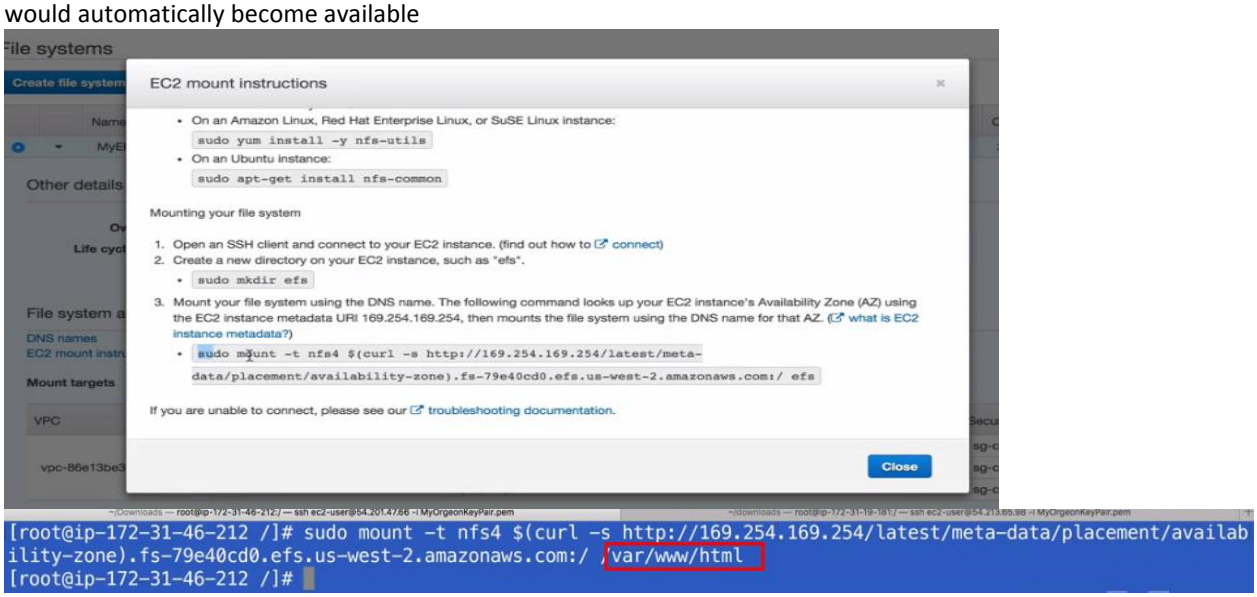
Amazon Elastic File System (Amazon EFS) is a file storage service for Amazon Elastic Compute Cloud (Amazon EC2) instances. Amazon EFS is easy to use and provides a simple interface that allows you to create and configure file systems quickly and easily. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

- Supports the Network File System version 4 (NFSv4) protocol
- You only pay for the storage you use (no pre-provisioning required)
- Can scale up to the petabytes
- Can support thousands of concurrent NFS connections
- Data is stored across multiple AZ's within a region
- Read After Write Consistency



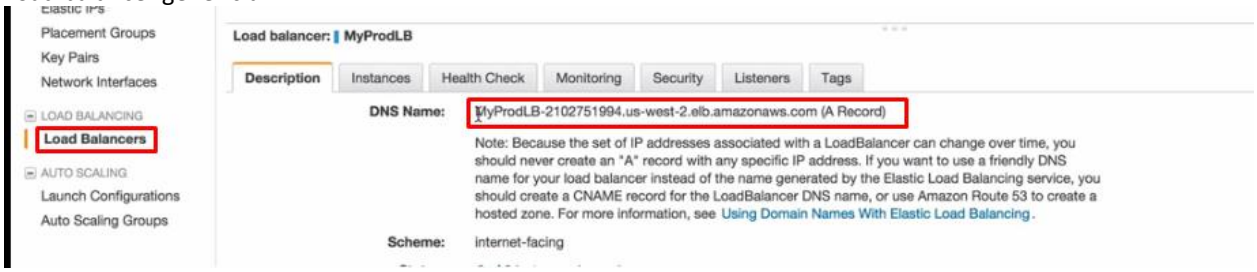


The security groups within EFS, EC2 instances, load balancers must be same for EFS to work accurately
What this means is that if we setup EFS and mount on our different instance within same security group and load balancers, our application (website or service) would be load balanced by AWS fairly automatically. If one instance goes down, other instances would automatically become available

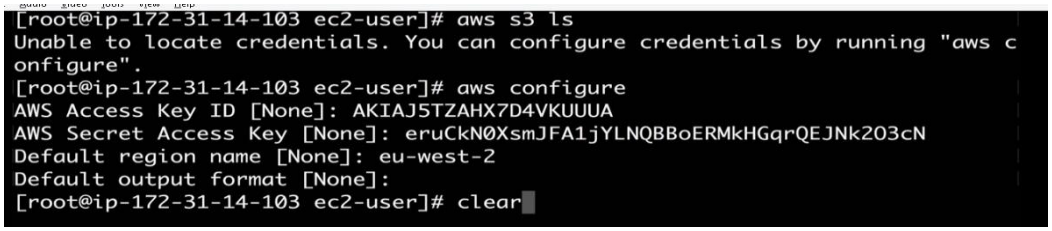


After mouting these on the instances, you have EFS setup which would make sure that the instances always have latest copy of changes made at any instance and would load balanced in case of one instance goes down

Load balancer generic url:



AWS CLI – Command Line Interface



Aws <service> help
aws s3 ls
available commands: cp, ls, mb, mv, presign, rb, rm

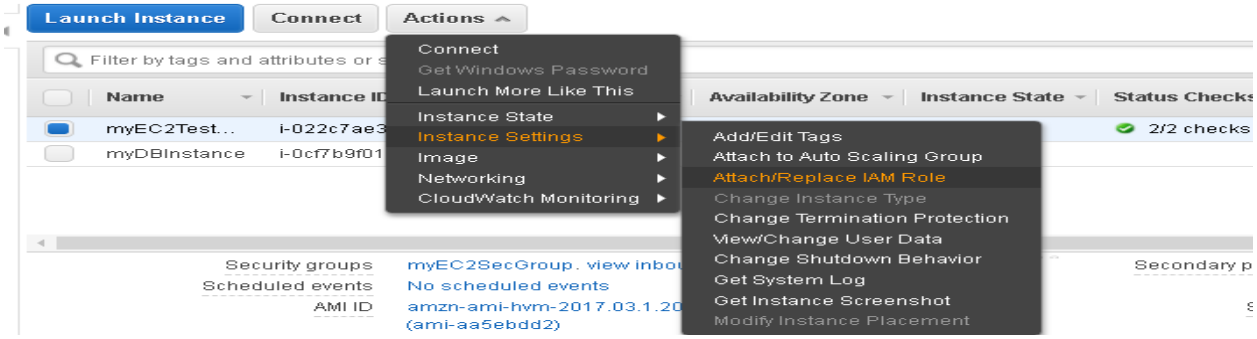
```
[root@ip-172-31-14-103 ec2-user]# cd ~
[root@ip-172-31-14-103 ~]# ls
[root@ip-172-31-14-103 ~]# cd .aws
[root@ip-172-31-14-103 .aws]# ls
config  credentials
[root@ip-172-31-14-103 .aws]# nano credentials
```

Credentials file would have secret_access_key and secret_access_id
With these credentials, anyone can login to aws console via command line, what people normally make mistakes, use these in their code and upload code into github, scripts or hackers poll these repositories and can use these against further hacking or mining bit coins and owners get charged by AWS. To make it secure, AWS roles play a great role.

Configure IAM to prevent hacking into AWS via access key & ID

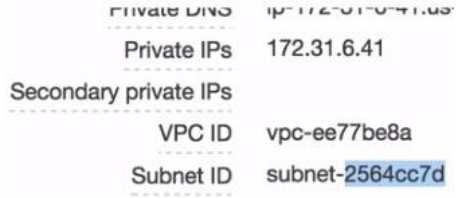
Click on Roles->create a new role-> choose AWS roles types

ROLES ARE GLOBAL, a specific region or zone cannot have specific roles
You can assign role while creating an instance or can modify as well



Configuring and assigning roles to instances wont store access keys but still you can aws cli commands from your program.

CLI Commands(docs.aws.amazon.com/cli/latest/references/ec2/index.html)
3 important commands for exam
->describe-instances: aws ec2 describe-instances
->describe-images: aws ec2 describe-images –images-ids ami-5731123e
->start-instances:
->Terminate instances: `aws ec2 terminate-instances --instance-ids i-1234567890abcdef0`
->run-instances:




```
aws ec2 describe-images --owners amazon --filters "Name=platform,Values=windows"
 "Name=root-device-type,Values=ebs"
```

To launch an instance in EC2-VPC

This example launches a single instance of type t2.micro into the specified subnet.

The key pair named MyKeyPair and the security group sg-903004f8 must exist.

Command:

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

S3 Buckets: objects based storage. DropBox is based on S3 only

Copy from one bucket in one region to another

```
[root@ip-172-31-10-48 ec2-user]# aws s3 cp --recursive s3://acloudguru-useast1 /home/ec2-user
download: s3://acloudguru-useast1/ACG-Austin.jpg to ./ACG-Austin.jpg
[root@ip-172-31-10-48 ec2-user]# ls
ACG-Austin.jpg
[root@ip-172-31-10-48 ec2-user]# aws s3 cp --recursive s3://acloudguru-euwest2 /home/ec2-user
fatal error: An error occurred (InvalidRequest) when calling the ListObjects operation: You are attempting to operate on a bucket in a region that requires Signature Version 4.  You can fix this issue by explicitly providing the correct region location using the --region argument, the AWS_DEFAULT_REGION environment variable, or the region variable in the AWS CLI configuration file.  You can get the bucket's location by running "aws s3api get-bucket-location --bucket BUCKET".
[root@ip-172-31-10-48 ec2-user]# aws s3 cp --recursive s3://acloudguru-euwest2 /home/ec2-user --region eu-west-2
download: s3://acloudguru-euwest2/pool.jpg to ./pool.jpg
```

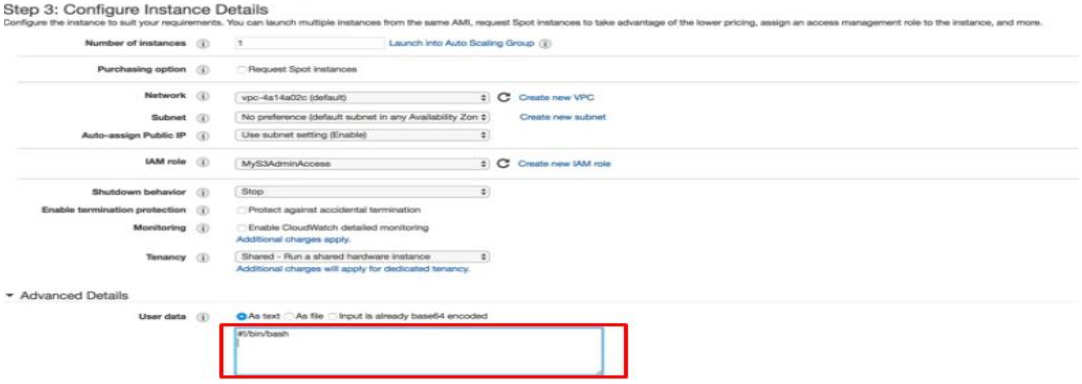
Habbit of puttin region is best practice but the command may or not work based on what region your commands gets fired in at runtime

Bash Scripting

aws-study | s3bootstrap.sh

```
#!/bin/bash
yum update -y
yum install httpd24 php56 git -y
service httpd start
chkconfig httpd on
cd /var/www/html
echo "<?php phpinfo();?>" > test.php
git clone https://github.com/acloudguru/s3
```

While creating an instance, you can assign bash commands and initiate basically various other aspects of env properties



When instance will be initiated, the bash would run, status can be check whether any pending update or not

```
[root@ip-172-31-51-50 ec2-user]# yum update -y
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main/latest                                         | 2.1 kB     00:00
amzn-updates/latest                                      | 2.3 kB     00:00
No packages marked for update
[root@ip-172-31-51-50 ec2-user]#
```

Installing php composer

Run script as mentioned above

```
[root@ip-172-31-46-11 html]# ls
s3  test.php
[root@ip-172-31-46-11 html]# curl -sS https://getcomposer.org/installer | php
```

```
[root@ip-172-31-46-11 html]# php composer.phar require aws/aws-sdk-php
Running composer as root/super user is highly discouraged as packages, plugins and scripts cannot always b
e trusted
[root@ip-172-31-46-11 html]# ls
composer.json  composer.lock  composer.phar  s3  test.php  vendor
[root@ip-172-31-46-11 html]# cd vendo
bash: cd: vendo: No such file or directory
[root@ip-172-31-46-11 html]# cd vendor
[root@ip-172-31-46-11 vendor]# ls
autoload.php  aws  bin  composer  guzzlehttp  mtdowling  psr
[root@ip-172-31-46-11 vendor]# nano autoload.php
```

Autoload.php downloads the latest sdk when run on, this can be scheduled on a particular frequency

Use php to log into S3 instance

```
[root@ip-172-31-46-11 /]# cd /var/www/html/s3
[root@ip-172-31-46-11 s3]# ls
cleanup.php  connecttoaws.php  createbucket.php  createfile.php  readfile.php  README.md
[root@ip-172-31-46-11 s3]# nano createbucket.php
```

Connection.php

```php
<?php
// Include the SDK using the Composer autoloader
require '/var/www/html/vendor/autoload.php';
$client = new Aws\S3\S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);
```

Calling module

```php
<?php
//copyright 2015 - A Cloud Guru.

//connection string
include 'connecttoaws.php';

// Create a unique bucket name
$bucket = uniqid("acloudguru", true);

// Create our bucket using our unique bucket name
$result = $client->createBucket(array(
    'Bucket' => $bucket
));

//HTML to Create our webpage
echo "<h1 align=\"center\">Hello Cloud Guru!</h1>";
echo "<div align = \"center\"><img src=\"https://s3-eu-west-1.amazonaws.com/acloudguru/logo.png\"></img><$
echo "<h2 align=\"center\">You have successfully created a bucket called {$bucket}</h2>";
echo "<div align=\"center\"><a href=\"createfile.php?bucket=$bucket\">Click Here to Continue</a></div>";
?>
```

createFile.php

```php
//get the bucket name
$bucket = $_GET["bucket"];

//create the file name
$key = 'cloudguru.txt';

//put the file and data in our bucket
$result = $client->putObject(array(
    'Bucket' => $bucket,
    'Key'    => $key,
    'Body'   => "Hello Cloud Gurus!"
));

//HTML to create our webpage
echo "<h2 align=\"center\">File - $key has been successfully uploaded to $bucket</h2>";
echo "<div align = \"center\"><img src=\"https://s3-eu-west-1.amazonaws.com/acloudguru/logo.png\"></img><$
echo "<div align = \"center\"><a href=\"readfile.php?bucket=$bucket&key=$key\">Click Here To Read Your Fi$
?>
```

Readfile.php

```php
<?php
//connection string
include 'connecttoaws.php';

//code to get our bucket and key names
$bucket = $_GET["bucket"];
$key = $_GET["key"];

//code to read the file on S3
$result = $client->getObject(array(
    'Bucket' => $bucket,
    'Key'    => $key
));
$data = $result['Body'];

//HTML to create our webpage
echo "<h2 align=\"center\">The Bucket is $bucket</h2>";
echo "<h2 align=\"center\">The Object's name is $key</h2>";
echo "<h2 align=\"center\">The Data in the object is $data</h2>";
echo "<div align = \"center\"><img src=\"https://s3-eu-west-1.amazonaws.com/acloudguru/logo.png\"></img><$
echo "<div align = \"center\"><a href=\"cleanup.php?bucket=$bucket&key=$key\">Click Here To Remove Files $
?>
```

Cleanup.php

```php
<?php
//Connection String
include'connecttoaws.php';

//Code to get our bucketname and file name
$bucket = $_GET["bucket"];
$key = $_GET["key"];

//buckets cannot be deleted unless they are empty
//Code to delete our object
$result = $client->deleteObject(array(
    'Bucket' => $bucket,
    'Key'    => $key
));

//Code to tell user the file has been deleted.
echo "<h2 align=\"center\">Object $key successfully deleted.</h2>";

//Code to delete our bucket
$result = $client->deleteBucket(array(
    'Bucket' => $bucket
));
```

EC2 Metadata

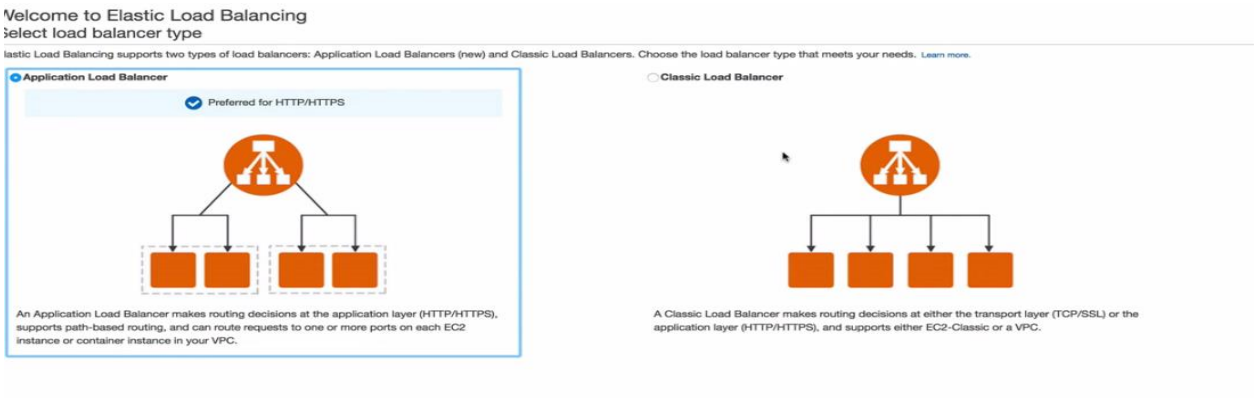
```
[root@ip-172-31-25-36 ec2-user]# curl http://169.254.169.254/latest/meta-data/
services/[root@ip-172-31-25-36 ec2-user]# curl http://169.254.169.254/latest/meta-data/pub
lic-ipv4
54.153.76.126[root@ip-172-31-25-36 ec2-user]#
```

Use metadata in a programming language

```php
<?php
        // create curl resource
        $ch = curl_init();
        $publicip = "http://169.254.169.254/latest/meta-data/public-ipv4";

        // set url
        curl_setopt($ch, CURLOPT_URL, "$publicip");

        //return the transfer as a string
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

        // $output contains the output string
        $output = curl_exec($ch);

        // close curl resource to free up system resources
        curl_close($ch);

        // close curl resource to free up system resources
        curl_close($ch);
```

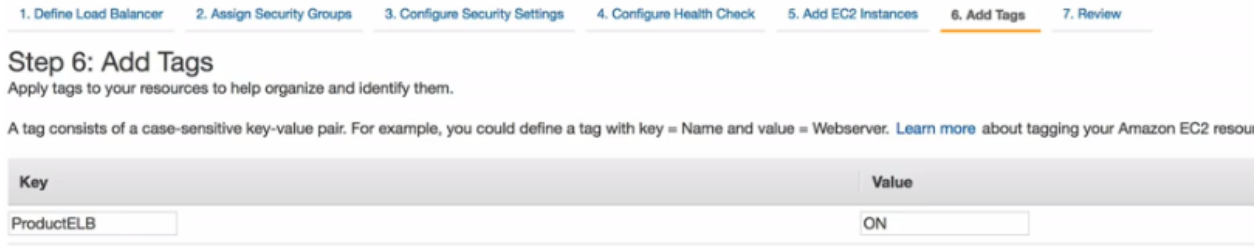
Elastic Load Balancers

Application load balancers are introduced somewhere in 2016



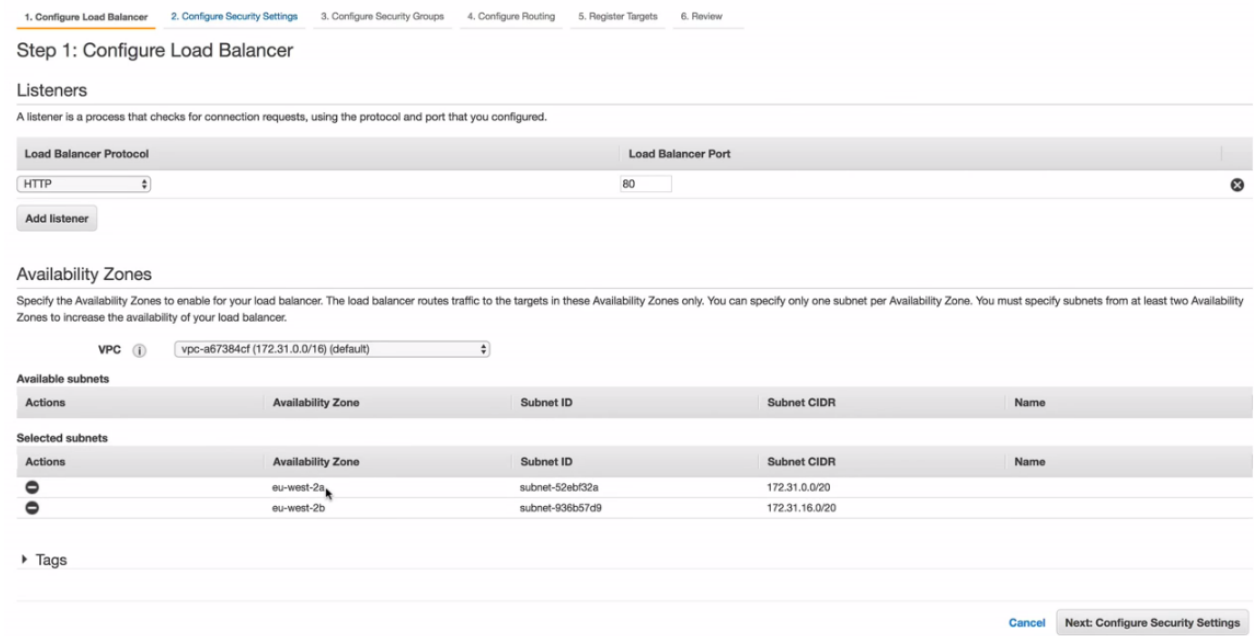
Main configuration which work as Classic Load Balancers

Tagging an ELB: important as it will be flagged in resource group all the time, so when you are done with free tier, you must delete the ELB to get saved from unnecessary charges

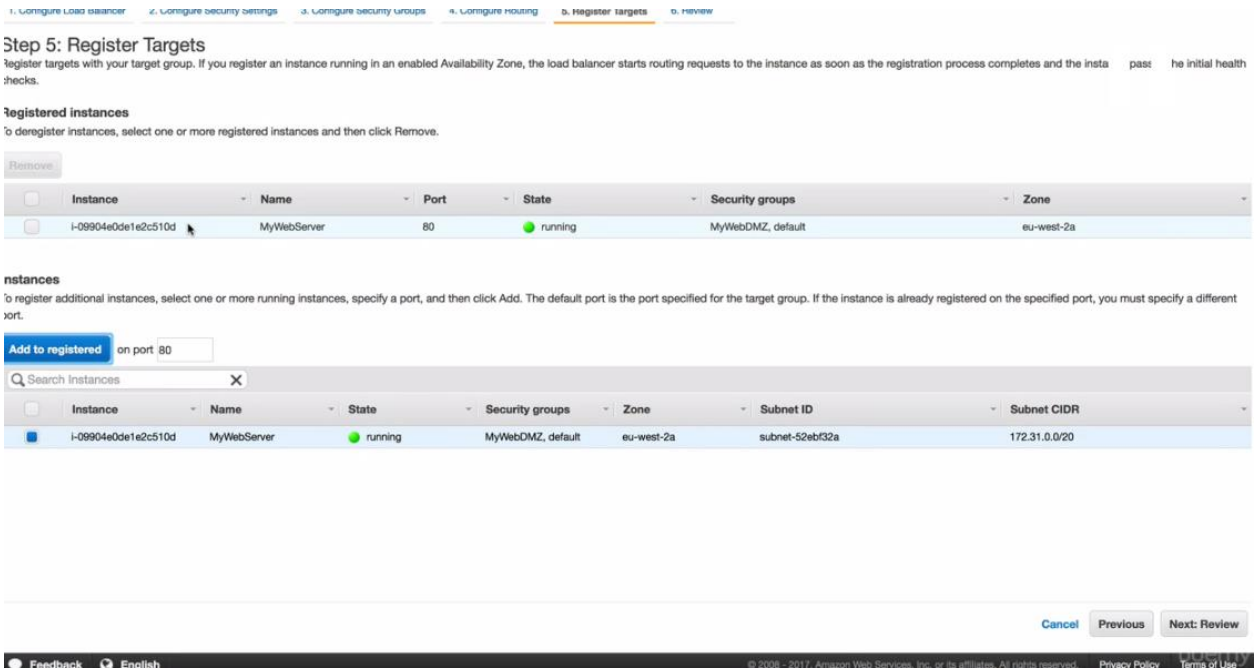


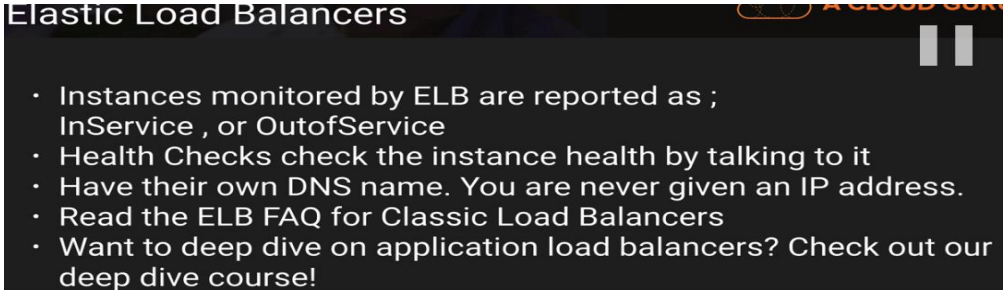
For load balancer, we dont get public ip address, what we get is the DNS name which is resolved automatically at run time, if 2 or more instances are tagged to load balancers, and if a specific healthcheck script is failing on one instance, load balancer will pick service from another instance automatically, users wont actually know if service has a downtime. Like facebook, google never goes down, its because of elastic load balancers

Application load balancers



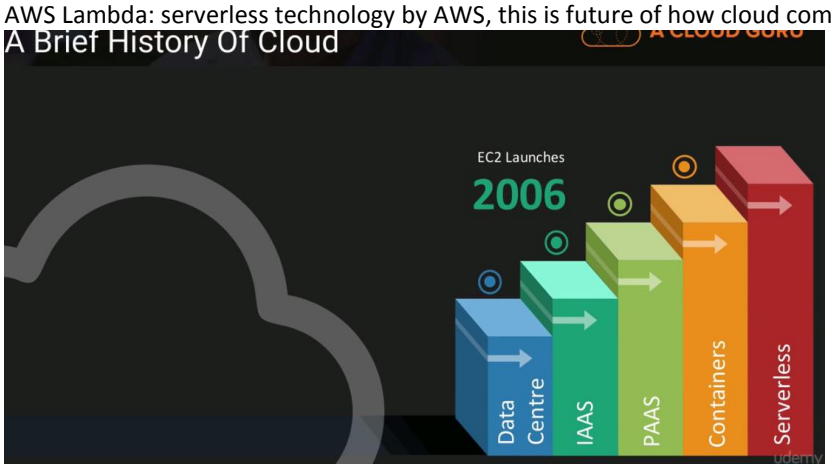
Instances has to be registered with the load balancers



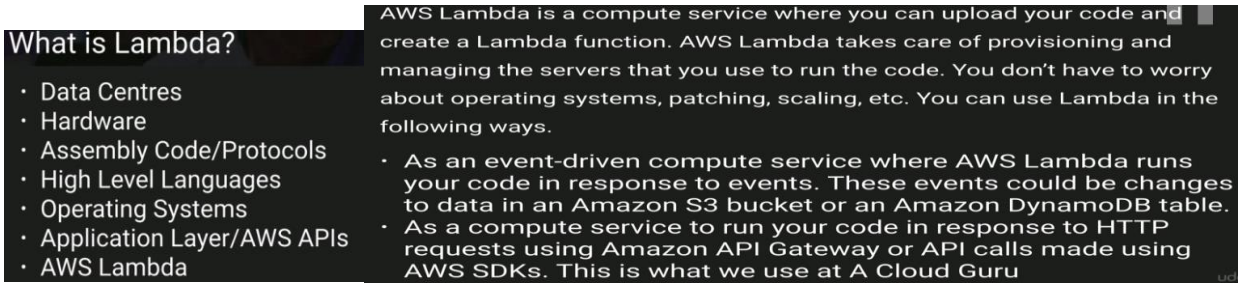


Exam tips on SDKs (must visit below before exam to get to know latest available SDKs)
https://aws.amazon.com/tools
android, iOS, JavaScript, Java, .Net, Php, Python, Ruby, Go, Node.JS, C++
SDKs have default region as US-EAST-1
Node.js dont have a default region

AWS Lambda: serverless technology by AWS, this is future of how cloud computing is done

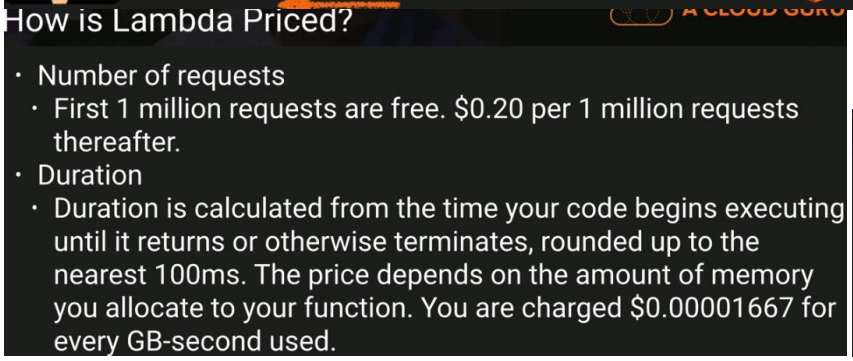


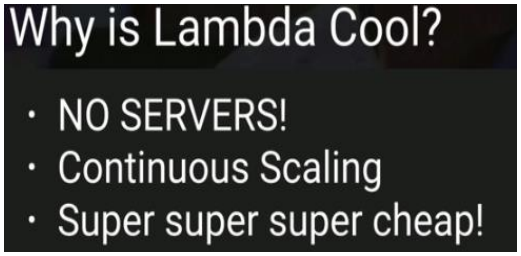
IaaS: Infrastructure as a Service. E.g. you can interact with Inra via API calls
PaaS: Platform as a Service, e.g. Elastic BeanStalk, upload code, AWS would analyse it and alloacate services and resource to it
Containers: e.g. Docker, virtual environments
Problem with above is that the companies has to still hire people to manage these on aws
Lambda: in reinvent 2015, you dont have to worry about managing any of above, this is sort of serverless intelligence, you still be able to work seemlessly. You have to just work on your code and application. Upload on lambda and it will take care of it



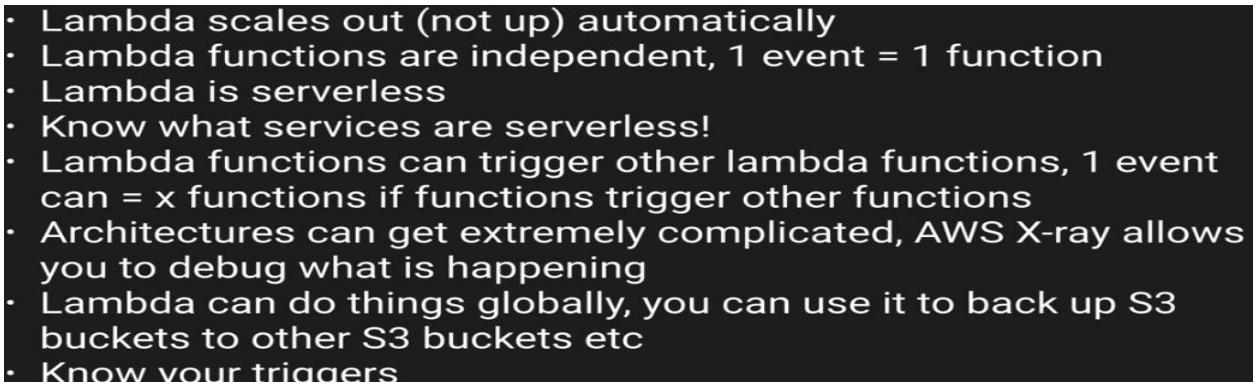
Lambda functions are deployed at run time like JIT activation and killed automatically when not in use to give betters responses to millions of calls to a single service. This means it scales out automatically

Python, java, c#, node.js are supported in lambda

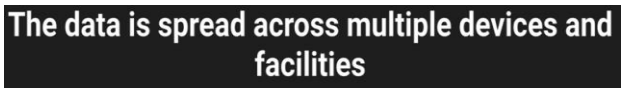


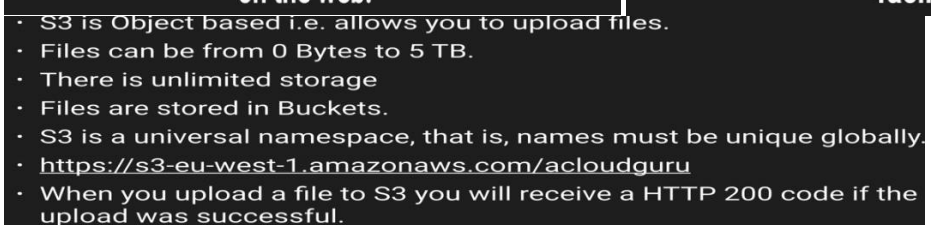


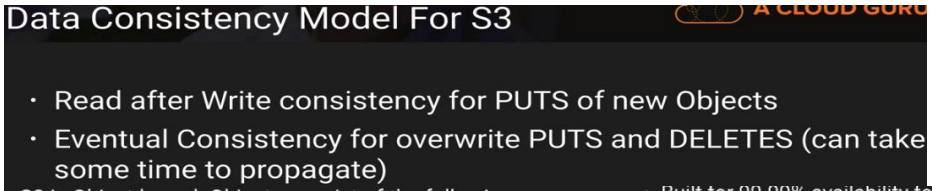
AWS cannot allow any lambda calls to be made if function or service or piece of code takes more than 5 mins. If it does, you have to break it



S3: simple storage service



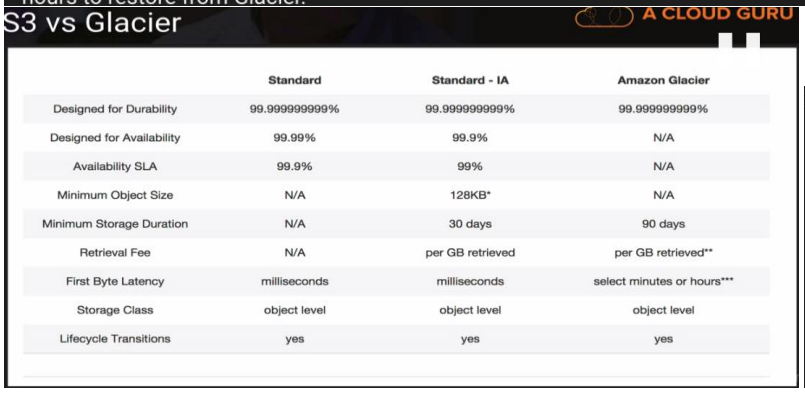




- S3 is Object based. Objects consist of the following;
  - Key (This is simply the name of the object)
  - Value (This is simply the data and is made up of a sequence of bytes).
  - Version ID (Important for versioning)
  - Metadata (Data about the data you are storing)
  - Subresources
    - Access Control Lists
    - Torrent
- S3 - 99.99% availability, 99.999999999% durability, stored redundantly across multiple devices in multiple facilities and is designed to sustain the loss of 2 facilities concurrently.
- S3 - IA (Infrequently Accessed) For data that is accessed less frequently, but requires rapid access when needed. Lower fee than S3, but you are charged a retrieval fee.
- Reduced Redundancy Storage - Designed to provide 99.99% durability and 99.99% availability of objects over a given year.
- Glacier - Very cheap, but used for archival only. It takes 3 - 5 hours to restore from Glacier.

- Built for 99.99% availability for the S3 platform.
- Amazon Guarantee 99.9% availability
- Amazon guarantees 99.999999999% durability for S3 information. (Remember 11 x 9's).
- Tiered Storage Available
- Lifecycle Management
- Versioning
- Encryption
- Secure your data using Access Control Lists and Bucket Policies

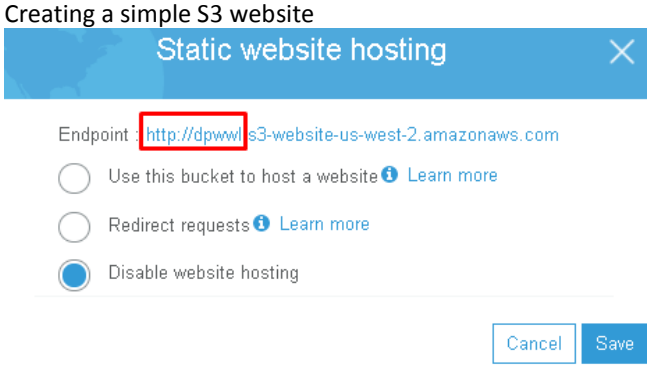
Glacier is an extremely low-cost storage service for data archival. Amazon Glacier stores data for as little as $0.01 per gigabyte per month, and is optimized for data that is infrequently accessed and for which retrieval times of 3 to 5 hours are suitable.



	Standard	Standard - IA	Amazon Glacier
Designed for Durability	99.9999999999%	99.9999999999%	99.9999999999%
Designed for Availability	99.99%	99.9%	N/A
Availability SLA	99.9%	99%	N/A
Minimum Object Size	N/A	128KB*	N/A
Minimum Storage Duration	N/A	30 days	90 days
Retrieval Fee	N/A	per GB retrieved	per GB retrieved**
First Byte Latency	milliseconds	milliseconds	select minutes or hours***
Storage Class	object level	object level	object level
Lifecycle Transitions	yes	yes	yes

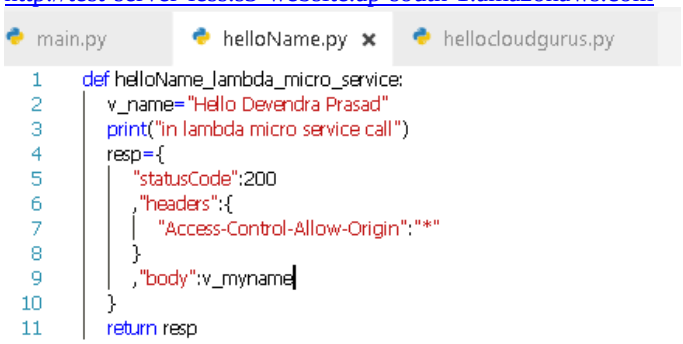
- Charged for;
  - Storage
  - Requests
  - Storage Management Pricing
  - Data Transfer Pricing
  - Transfer Acceleration

Creating a simple S3 website



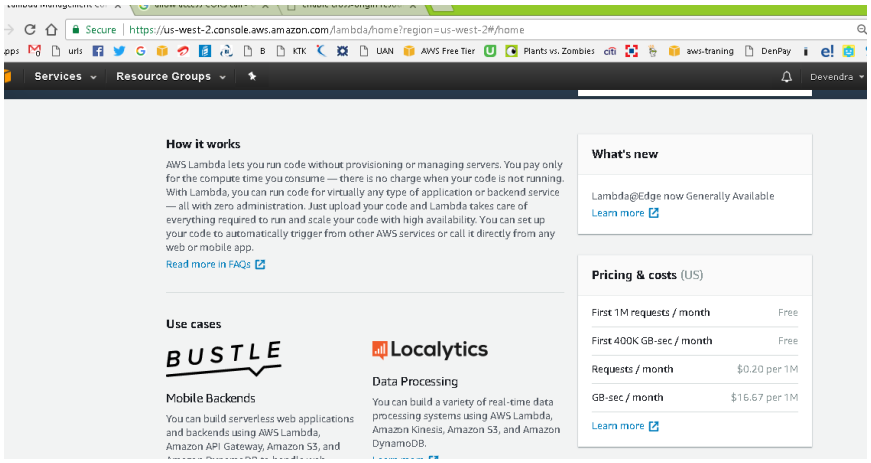
CORS: Cross Origin Resource Sharing
Way of allowing resource sharing from one bucket to another

Micro Services architecture using Lambda
http://test-server-less.s3-website.ap-south-1.amazonaws.com

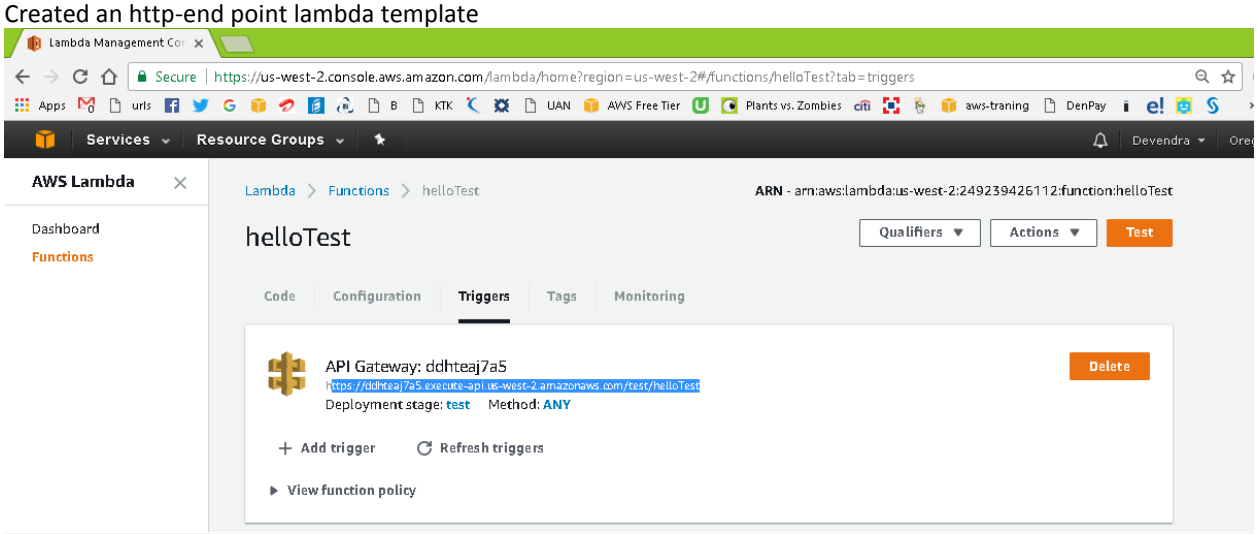


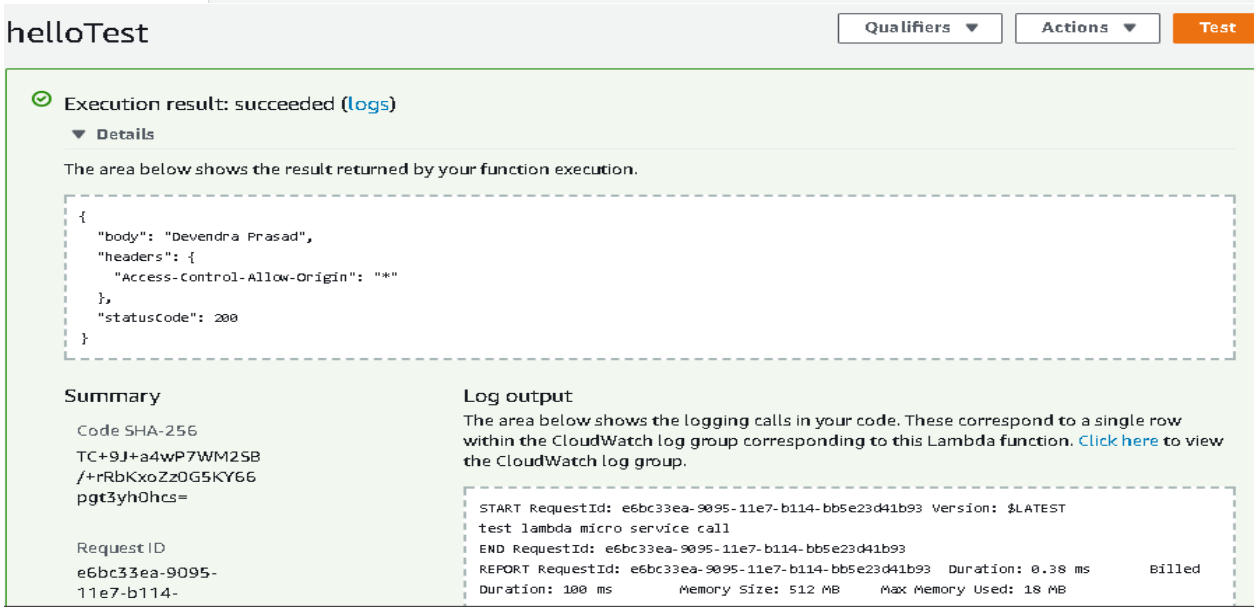

```
1  def helloName_lambda_micro_service:
2      v_name="Hello Devendra Prasad"
3      print("in lambda micro service call")
4      resp={
5          "statusCode":200
6          ,"headers":{
7              "Access-Control-Allow-Origin":"*"
8          }
9          ,"body":v_myname
10     }
11     return resp
```

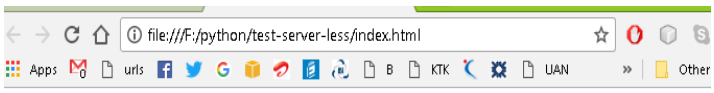
Lambda pricing



Created an http-end point lambda template







AWS Lambda-Microservcie Architecture Test

AWS Lambda-Microservcie Architecture Test

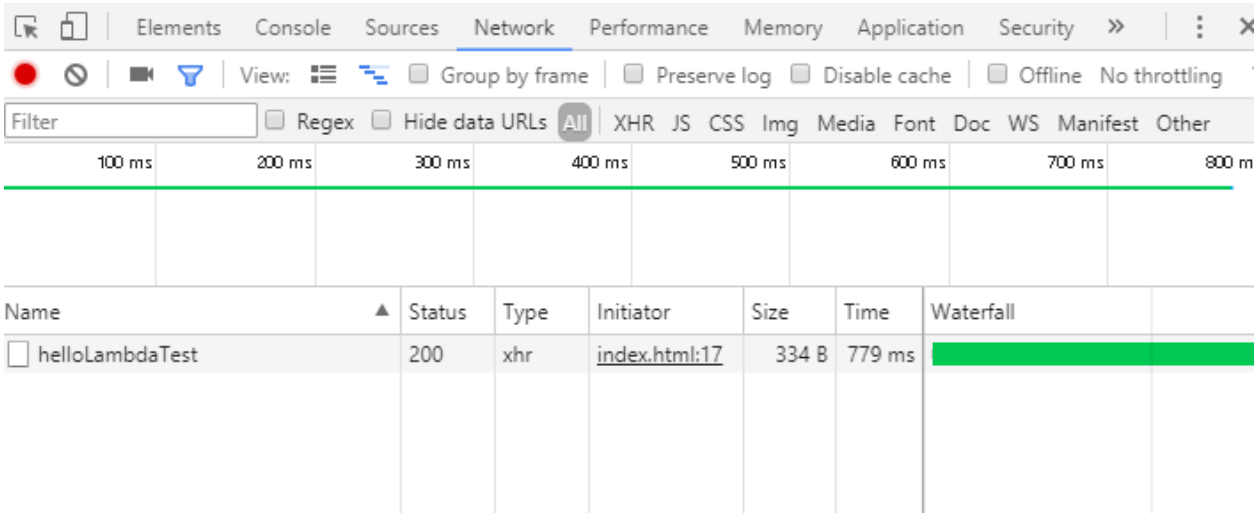
{{Print here}}

Click to Test Hello Lambda Call

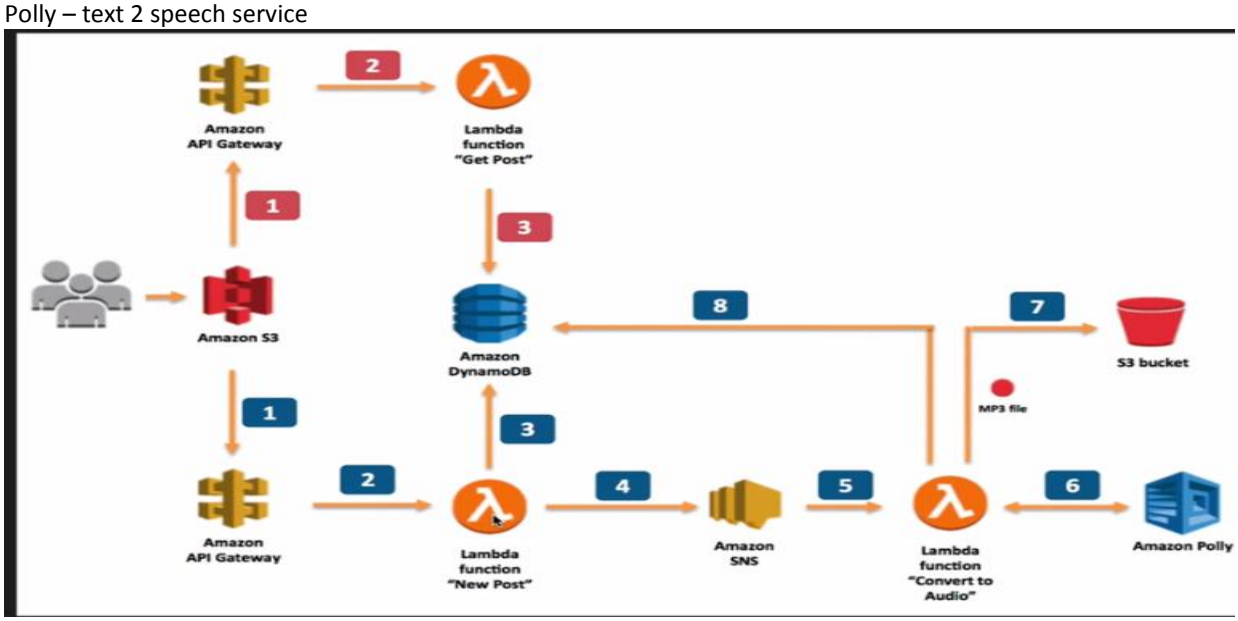
Url Shortner Lambda Test

Devendra Prasad

Click to Test Hello Lambda Call
Url Shortner Lambda Test



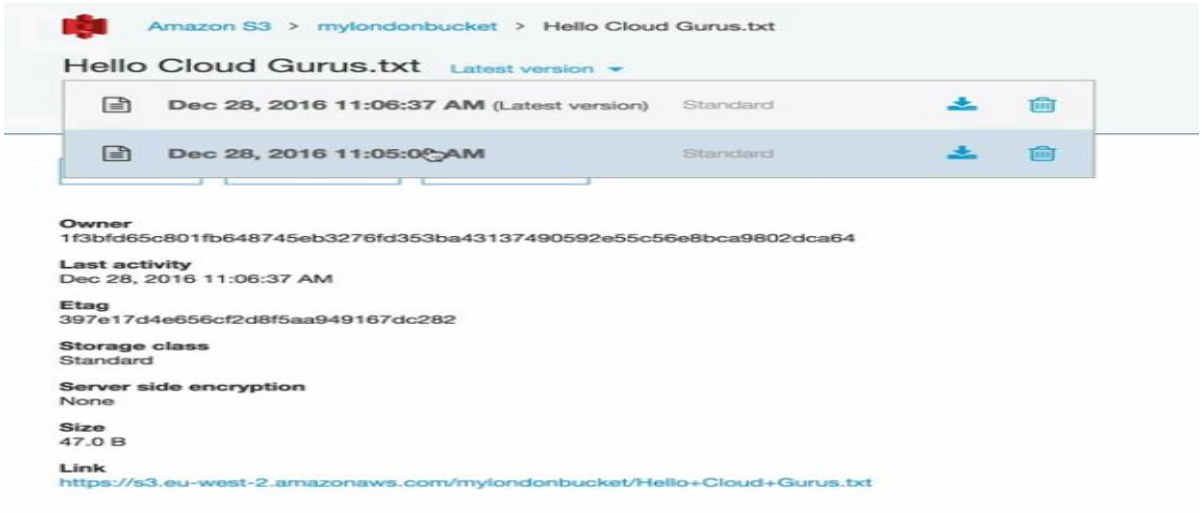
Polly – text 2 speech service



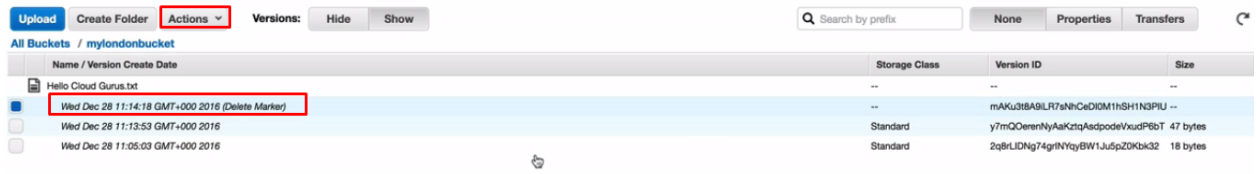
Steps to create polly enabled website
create an s3 bucket as a website
create s3 bucket to store polly enabled audio files
create a dynamo db which would store the posts as texts
create a custom IAM role and attach custom policy document to it which should enable, polly,s3,dynamo db,sns and logs or any other custom services used in the services
create custom lambda function to save data into dynamo db, read from it, and make a trigger that would convert any incoming text to audio and save in s3
cerate an API gateway (GET, PUT, CORS, Content-Type='application.json', queryString parameters etc), deploy API and get a API key
got to S3 buckey and change bucket policy to make all objects that are saved via API gateway as public by default
create your web page to call this api gateway from js and all configurations behind the scene will trigger via this api gateway

alexa speech notes are only 90sec long, more than this doesnt work

S3 versioning
Once enabled, you cannot remove versioning from the S3 bucket, it can only be disabled

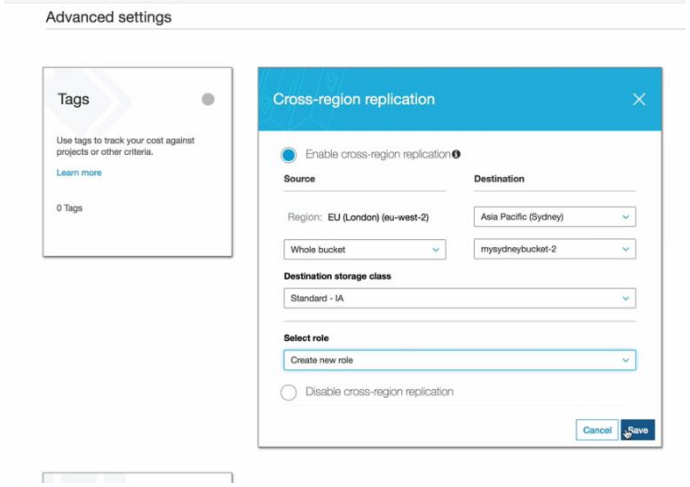


As an architect, you must choose the files policy or life cycle management(delete or acrhite off to glacier) and dont want versioning on large files which are continuously updated as this would increase the cloud size. File sizes must be small.
If we delete the version info(from drop down), it cannot be restored. But if we delete the file / object, it can be restored from versioning
Via old console, restore a deleted object easily but in new console, we must use a glacier command to restore versioned file



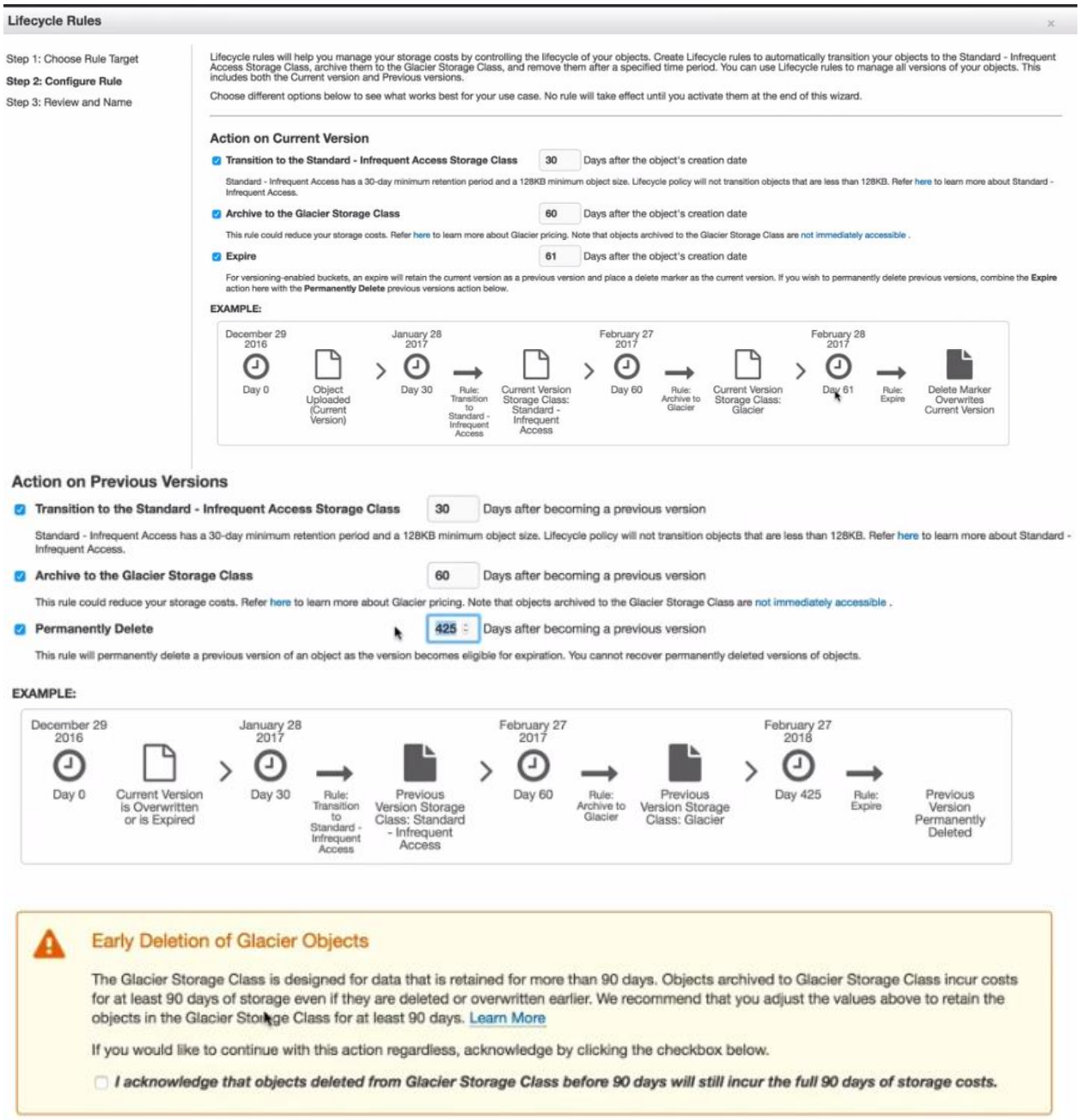
- Stores all versions of an object (including all writes and even if you delete an object)
- Great backup tool.
- Once enabled, Versioning cannot be disabled, only suspended.
- Integrates with Lifecycle rules
- Versioning's MFA Delete capability, which uses multi-factor authentication, can be used to provide an additional layer of security.

Cross Region S3 Bucket Replication
Choose bucket you want to replicate to, goto properties, choose



Versioning is must in target bucket
Role help diff buckets in same or different region talk to each other
Existing objects in source bucket are not automatically copied over to destination. New objects will be automatically copied or updates to older object(s) will be replicated including all versions and permissions.

S3 Object life cycle management



- Can be used in conjunction with versioning.
- Can be applied to current versions and previous versions
- Following actions can now be done;
  - Transition to the Standard - Infrequent Access Storage Class (128Kb and 30 days after the creation date).
  - Archive to the Glacier Storage Class (30 days after IA, if relevant)
  - Permanently Delete

Cloud Front – Chargeable service

A content delivery network (CDN) is a system of distributed servers (network) that deliver webpages and other web content to a user based on the geographic locations of the user, the origin of the webpage and a content delivery server.

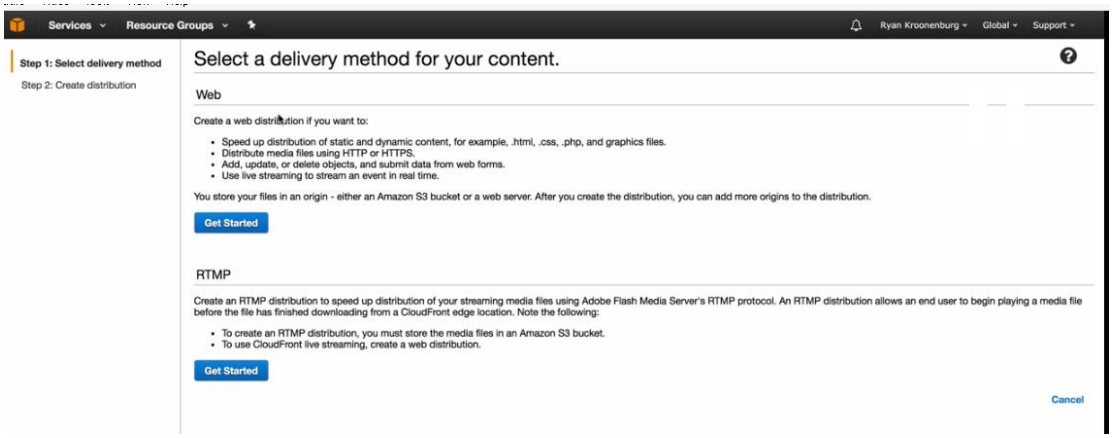
50edge locations currently

- Edge Location - This is the location where content will be cached. This is separate to an AWS Region/AZ
- Origin - This is the origin of all the files that the CDN will distribute. This can be either an S3 Bucket, an EC2 Instance, an Elastic Load Balancer or Route53.
- Distribution - This is the name given the CDN which consists of a collection of Edge Locations.
  - Web Distribution - Typically used for Websites.
  - RTMP - Used for Media Streaming.

Amazon CloudFront can be used to deliver your entire website, including dynamic, static, streaming, and interactive content using a global network of edge locations. Requests for your content are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

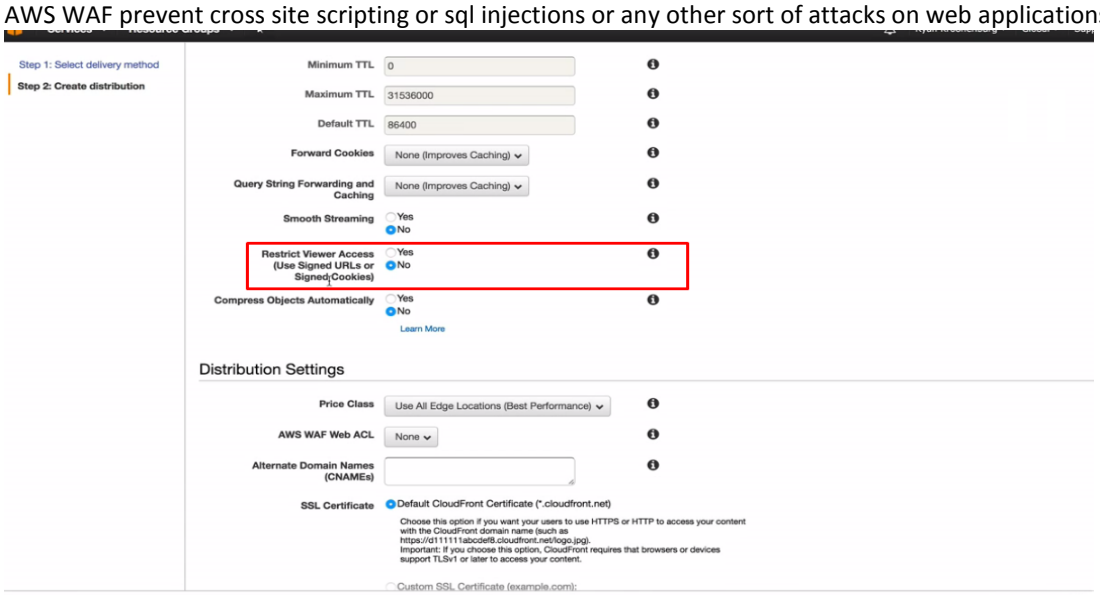
Amazon CloudFront is optimized to work with other Amazon Web Services, like Amazon Simple Storage Service (Amazon S3), Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Load Balancing, and Amazon Route 53. Amazon CloudFront also works seamlessly with any non-AWS origin server, which stores the original, definitive versions of your files.

AWS Console->Services->cloud front

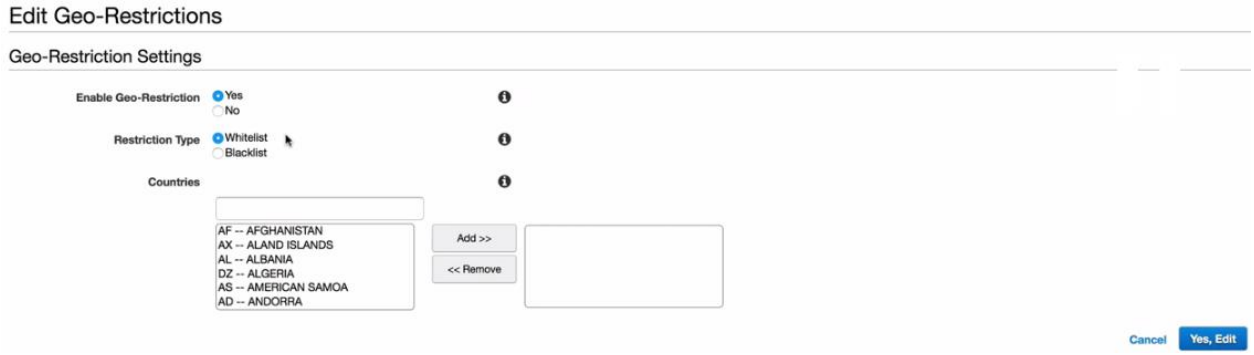


Secure cloud front urls or accesses

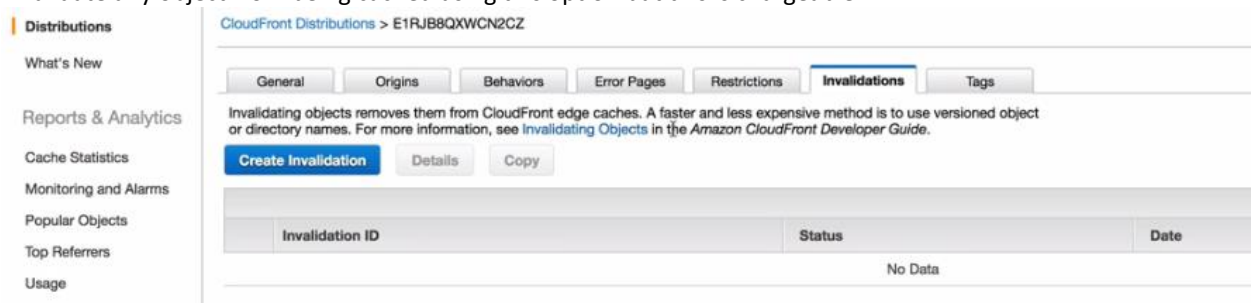
AWS WAF prevent cross site scripting or sql injections or any other sort of attacks on web applications hosted on aws or cloud front



Restriction in cloud front, can be edited once distribution is created



Invalidation- for critical objects to be prevented to be cached over to edge location, we can override the TTL settings and quickly invalidate any object from being cached using this option but this is chargeable



Securing buckets

- By default, all newly created buckets are PRIVATE
- You can setup access control to your buckets using;
  - Bucket Policies
  - Access Control Lists
- S3 buckets can be configured to create access logs which log all requests made to the S3 bucket. This can be done to another bucket.

SSE-Server side encryption – KMS-Key Managemenrt Server,C-Customer Managed, S3-managed bucket policy

Client Side Encryption

- In Transit;
  - SSL/TLS
- At Rest
  - Server Side Encryption
    - S3 Managed Keys - **SSE-S3**
    - AWS Key Management Service, Managed Keys - **SSE-KMS**
    - Server Side Encryption With Customer Provided Keys - **SSE-C**

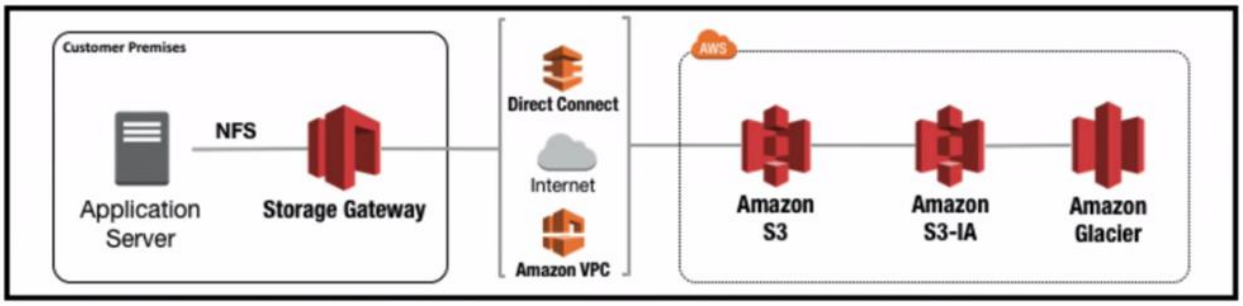
S3-Storage Gateways

AWS Storage Gateway is a service that connects 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. The service enables you to securely store data to the AWS cloud for scalable and cost-effective storage.

AWS Storage Gateway's software appliance is available for download as a virtual machine (VM) image that you install on a host in your datacenter. Storage Gateway supports either VMware ESXi or Microsoft Hyper-V. Once you've installed your gateway and associated it with your AWS account through the activation process, you can use the AWS Management Console to create the storage gateway option that is right for you.

- File Gateway (NFS)
- Volumes Gateway (iSCSI)
  - Stored Volumes
  - Cached Volumes
- Tape Gateway (VTL)

File Gateway – flat files, doc, ppt, text, audio, video images etc

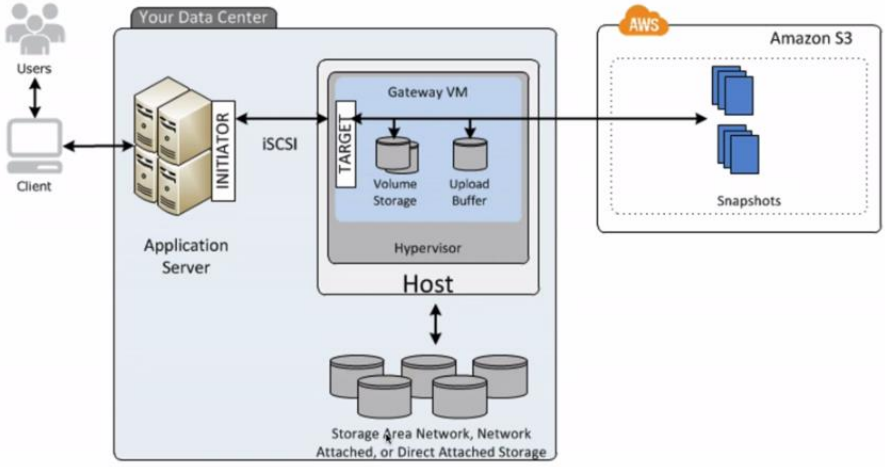


Volume Gateway –disc volume copy e.g. OS, VMs, Databases etc

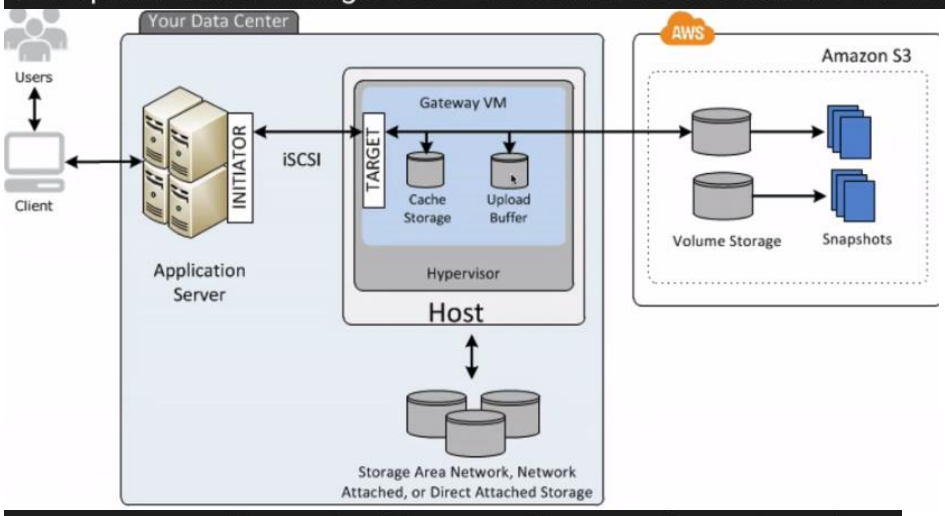
The volume interface presents your applications with disk volumes using the iSCSI block protocol.

Data written to these volumes can be asynchronously backed up as point-in-time snapshots of your volumes, and stored in the cloud as Amazon EBS snapshots.

Snapshots are incremental backups that capture only changed blocks. All snapshot storage is also compressed to minimize your storage charges.
Stored volumes let you store your primary data locally, while asynchronously backing up that data to AWS. Stored volumes provide your on-premises applications with low-latency access to their entire datasets, while providing durable, off-site backups. You can create storage volumes and mount them as iSCSI devices from your on-premises application servers. Data written to your stored volumes is stored on your on-premises storage hardware. This data is asynchronously backed up to Amazon Simple Storage Service (Amazon S3) in the form of Amazon Elastic Block Store (Amazon EBS) snapshots. 1 GB - 16 TB in size for Stored Volumes.



Cached volumes let you use Amazon Simple Storage Service (Amazon S3) as your primary data storage while retaining frequently accessed data locally in your storage gateway. Cached volumes minimize the need to scale your on-premises storage infrastructure, while still providing your applications with low-latency access to their frequently accessed data. You can create storage volumes up to 32 TiB in size and attach to them as iSCSI devices from your on-premises application servers. Your gateway stores data that you write to these volumes in Amazon S3 and retains recently read data in your on-premises storage gateway's cache and upload buffer storage. 1 GB - 32 TB in size for Cached Volumes.



Tape Gateway offers a durable, cost-effective solution to archive your data in the AWS Cloud. The VTL interface it provides lets you leverage your existing tape-based backup application infrastructure to store data on virtual tape cartridges that you create on your tape gateway. Each tape gateway is preconfigured with a media changer and tape drives, which are available to your existing client backup applications as iSCSI devices. You add tape cartridges as you need to archive your data. Supported by NetBackup, Backup Exec, Veam etc

- File Gateway - For flat files, stored directly on S3.
- Volume Gateway
  - Stored Volumes - Entire Dataset is stored on site and is asynchronously backed up to S3.
  - Cached Volumes - Entire Dataset is stored on S3 and the most frequently accessed data is cached on site.
- Gateway Virtual Tape Library (VTL)
  - Used for backup and uses popular backup applications like NetBackup, Backup Exec, Veam etc

SNOWBALL

AWS Import/Export Disk accelerates moving large amounts of data into and out of the AWS cloud using portable storage devices for transport. AWS Import/Export Disk transfers your data directly onto and off of storage devices using Amazon's high-speed internal network and bypassing the Internet.

- Snowball
- Snowball Edge
- Snowmobile

Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of AWS. Using Snowball addresses common challenges with large-scale data transfers including high network costs, long transfer times, and security concerns. Transferring data with Snowball is simple, fast, secure, and can be as little as one-fifth the cost of high-speed Internet.

80TB snowball in all regions. Snowball uses multiple layers of security designed to protect your data including tamper-resistant enclosures, 256-bit encryption, and an industry-standard Trusted Platform Module (TPM) designed to ensure both security and full chain-of-custody of your data. Once the data transfer job has been processed and verified, AWS performs a software erasure of the Snowball appliance.

AWS Snowball Edge is a 100TB data transfer device with on-board storage and compute capabilities. You can use Snowball Edge to move large amounts of data into and out of AWS, as a temporary storage tier for large local datasets, or to support local workloads in remote or offline locations.

Snowball Edge connects to your existing applications and infrastructure using standard storage interfaces, streamlining the data transfer process and minimizing setup and integration. Snowball Edge can cluster together to form a local storage tier and process your data on-premises, helping ensure your applications continue to run even when they are not able to access the cloud.

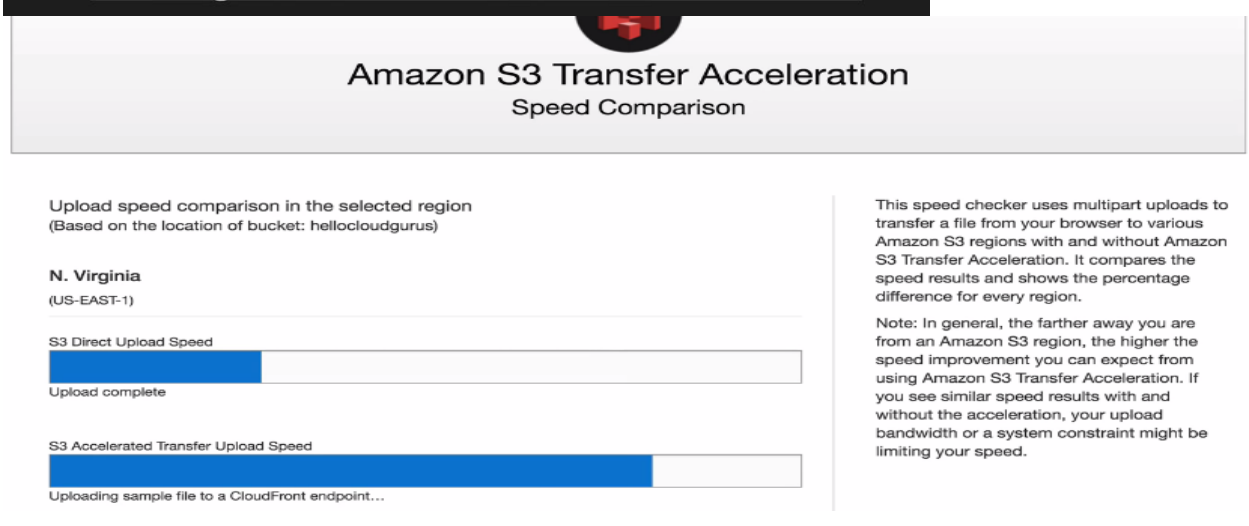
SnowMobile: petabyte to Exabytes of storage and transfers to and from of AWS cloud
Helps complete data center migrations

AWS Snowmobile is an Exabyte-scale data transfer service used to move extremely large amounts of data to AWS. You can transfer up to 100PB per Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck. Snowmobile makes it easy to move massive volumes of data to the cloud, including video libraries, image repositories, or even a complete data center migration. Transferring data with Snowmobile is secure, fast and cost effective.

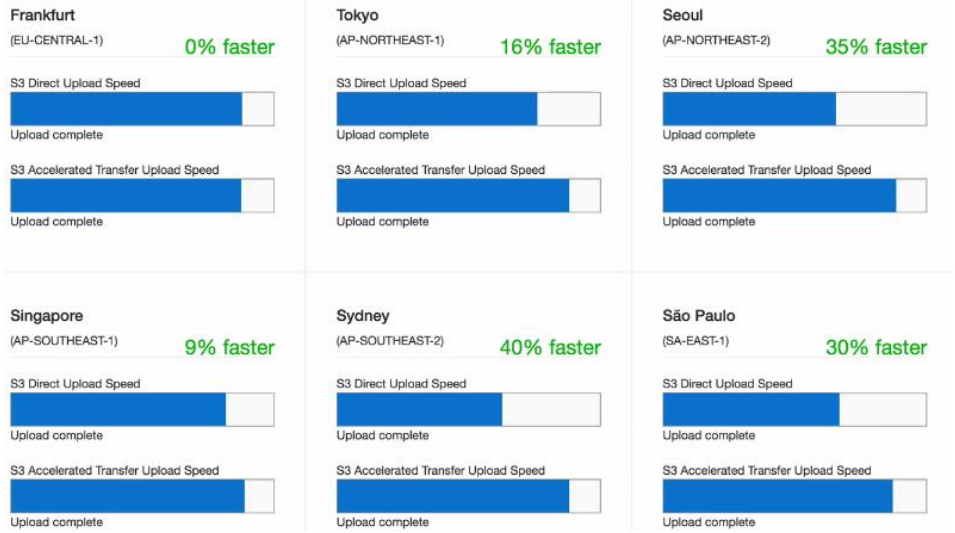
What is S3 Transfer Acceleration?

S3 Transfer Acceleration utilises the CloudFront Edge Network to accelerate your uploads to S3. Instead of uploading directly to your S3 bucket, you can use a distinct URL to upload directly to an edge location which will then transfer that file to S3. You will get a distinct URL to upload to;

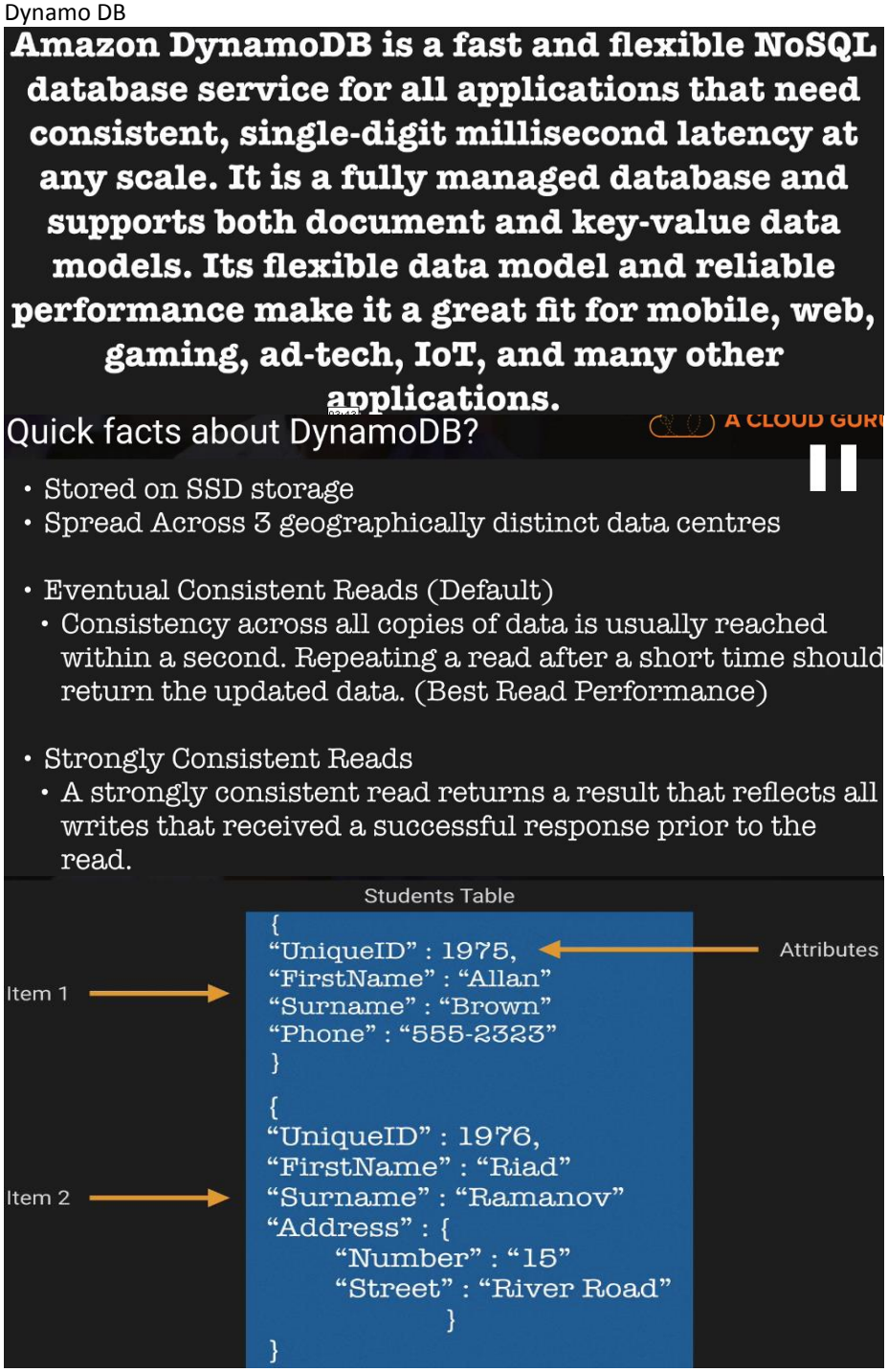
acloudguru.s3-accelerate.amazonaws.com



Speed of S3 transfer would actually depend on the region you are in and amount of data you are going to transfer



Dynamo DB



Pricing:

- Provisioned Throughput Capacity
  - Write Throughput $0.0065 per hour for every 10 units
  - Read Throughput $0.0065 per hour for every 50 units

- First 25 GB stored per month is free
- Storage costs of $0.25 GB per month there after.

Let's assume that your application needs to perform 1 million writes and 1 million reads per day, while storing 28 GB of data.

First, you need to calculate how many writes and reads per second you need. 1 million evenly spread writes per day is equivalent to 1,000,000 (writes) / 24 (hours) / 60 (minutes) / 60 (seconds) = 11.6 writes per second.

A DynamoDB Write Capacity Unit can handle 1 write per second, so you need 12 Write Capacity Units. For write throughput, you are charged on $0.0065 for every 10 units.

So ($0.0065/10) * 12 * 24 = $0.1872 per day.

Similarly, to handle 1 million strongly consistent reads per day, you need 12 Read Capacity Units. For read throughput you are charged $0.0065 for every 50 units.

So ($0.0065/50) * 12 * 24 = $0.0374 per day.

Storage costs is $0.25 per GB per month. Lets assume our database is 28 GB. We get the first 25 GB for free so we only pay for 3 GB of storage which is $0.75 per month.

Total Cost = $0.1872 per day + $0.0374 per day
Plus Storage of 0.75 per month

(30 x ($0.1872 + $0.0374)) $0.75 = $7.488

With Free Tier You Get
25 Read Capacity Units
25 Write Capacity Units

Dynamo DB Index

Two Types Of Primary Keys Available;

Single Attribute (think unique ID)
- Partition Key (Hash Key) composed of one attribute.

Composite (think unique ID and a date range)
- Partition Key & Sort Key (Hash & Range) composed of two attributes.

Partition Key
- DynamoDB uses the partition key's value as input to an internal hash function. The output from the hash function determines the partition (this is simply the physical location in which the data is stored).
- No two items in a table can have the same partition key value!

Partition Key and Sort key.

- DynamoDB uses the partition key's value as input to an internal hash function. The output from the hash function determines the partition (this is simply the physical location in which the data is stored).
- Two items can have the same partition key, but they **must have a different sort key**.
- All items with the same partition key are stored together, in sorted order by sort key value.

Local Secondary Index

- Has the SAME Partition key, different sort key.
- Can ONLY be created when creating a table. They cannot be removed or modified later.

Global Secondary Index

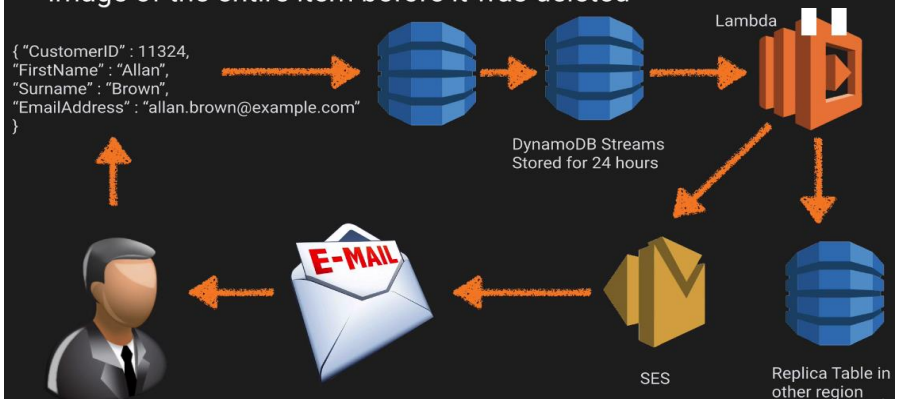
- Has DIFFERENT Partition key and different sort key.
- Can be created at table creation or added LATER.

DynamoDB Streams

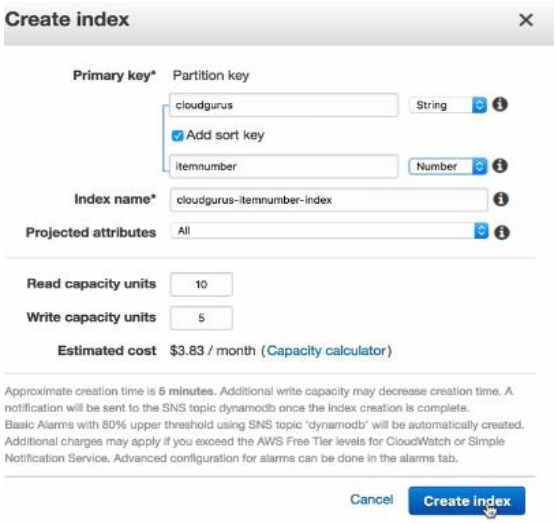
A CLOUD GURU

Used to capture any kind of modification of the DynamoDB tables.

- If a new item is added to the table, the stream captures an image of the entire item, including all of its attributes.
- If an item is updated, the stream captures the "before" and "after" image of any attributes that were modified in the item.
- If an item is deleted from the table, the stream captures an image of the entire item before it was deleted

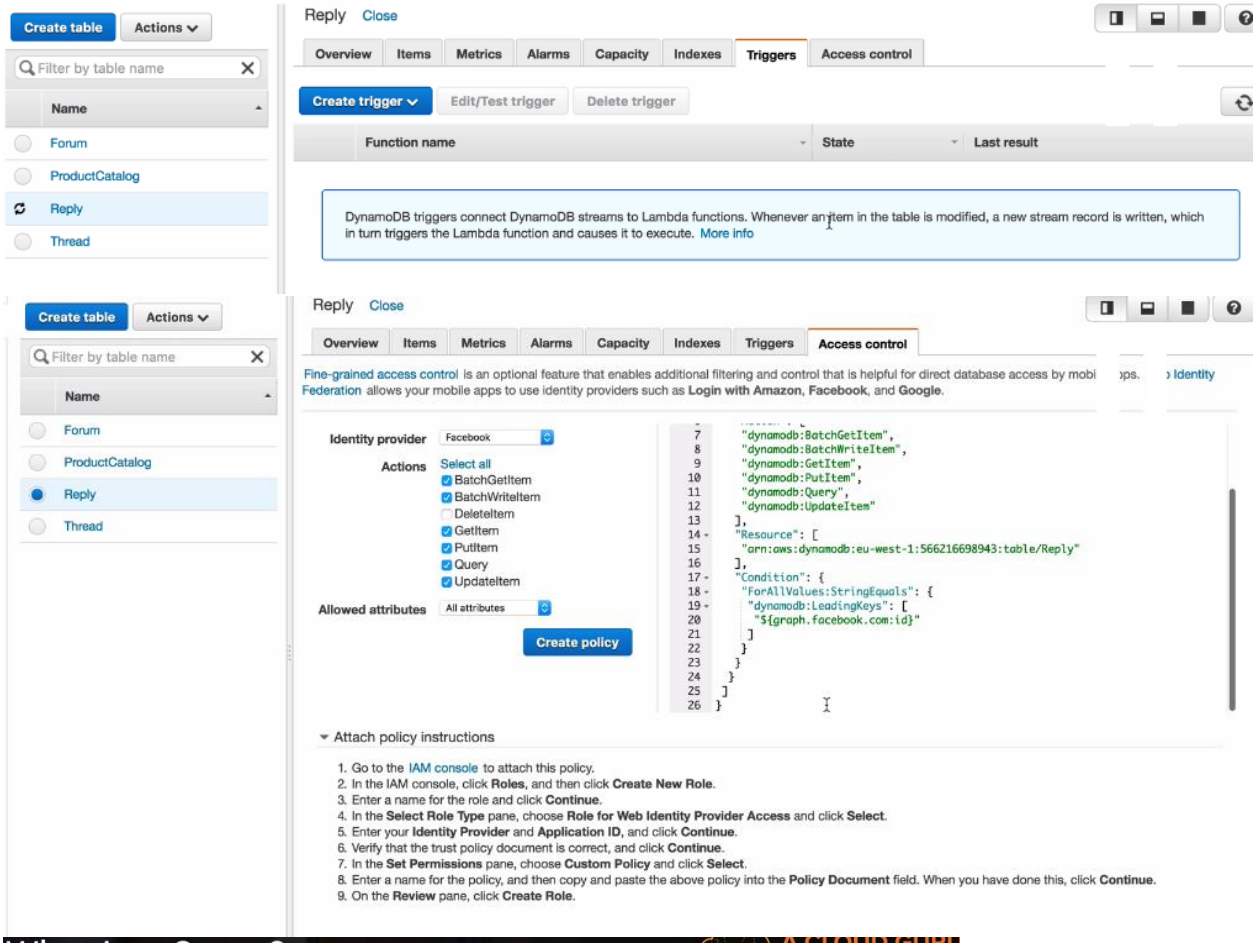


We can have max upto 5 local or global secondary indexes



We can have either existing attribute or create a new one as GSI at run time as well

Name	Status	Type	Partition key	Sort key	Attributes	Read capacity	Write capacity	Size
cloudgurus-itemnumber-	Active	GSI	cloudgurus (String)	itemnumber (Numbe	ALL	10	5	0
PostedBy-index	N/A	LSI	Id (String)	PostedBy (String)	Id, ReplyDate	-	-	0



What is a Query?

A CLOUD GURU

- A Query operation finds items in a table using only primary key attribute values. You must provide a partition attribute name and a distinct value to search for.
- You can optionally provide a sort key attribute name and value, and use a comparison operator to refine the search results.
- By default, a Query returns all of the data attributes for items with the specified primary key(s); however, you can use the **ProjectionExpression** parameter so that the Query only returns some of the attributes, rather than all of them.
- Query results are always sorted by the sort key. If the data type of the sort key is a number, the results are returned in numeric order; otherwise, the results are returned in order of ASCII character code values. By default, the sort order is ascending. To reverse the order, set the **ScanIndexForward** parameter to false.
- By Default is eventually consistent but can be changed to be strongly consistent.

What is a Scan?

A CLOUD GURU

- A Scan operation examines every item in the table. By default, a Scan returns all of the data attributes for every item; however, you can use the **ProjectionExpression** parameter so that the Scan only returns some of the attributes, rather than all of them.

What should I use? Query vs Scan?

Generally, a Query operation is more efficient than a Scan operation.

A Scan operation always scans the entire table, then filters out values to provide the desired result, essentially adding the extra step of removing data from the result set. Avoid using a Scan operation on a large table with a filter that removes many results, if possible. Also, as a table grows, the Scan operation slows. The Scan operation examines every item for the requested values, and can use up the provisioned throughput for a large table in a single operation.

For quicker response times, design your tables in a way that can use the Query, Get, or BatchGetItem APIs, instead. Alternatively, design your application to use Scan operations in a way that minimizes the impact on your table's request rate.

Provisioned Throughput

- Unit of Read provisioned throughput
  - All reads are rounded up to increments of 4 KB
  - Eventually Consistent Reads (default) consist of 2 reads per second.
  - Strongly Consistent Reads consist of 1 read per second.

- Unit of Write provisioned throughput
  - All writes are 1 KB
  - All writes consist of 1 write per second

The Magic Formula

Question - You have an application that requires to read 10 items of 1 KB per second using eventual consistency. What should you set the read throughput to?

(Size of Read rounded to nearest 4 KB chunk / 4 KB) x no of items = read throughput

Divide by 2 if eventually consistent.

Question - You have an application that requires to read 10 items of 1 KB per second using eventual consistency. What should you set the read throughput to?

- First we calculate how many read units per item we need

- 1KB rounded to the nearest 4 KB increment = 4.
- 4 KB / 4KB = 1read unit per item.

- 1 x 10 read items = 10
- Using eventual consistency we get 10 / 2 = 5
- 5 units of read throughput

You have an application that requires to read 10 items of 6 KB per second using eventual consistency. What should you set the read throughput to?

- First we calculate how many read units per item we need
- 6 KB rounded up to nearest increment of 4 KB is 8 KB.
- 8 KB / 4KB = 2 read units per item.

- 2 x 10 read items = 20
- Using eventual consistency we get 20 / 2 = 10

- 10 units of read throughput

You have an application that requires to read 5 items of 10 KB per second using eventual consistency. What should you set the read throughput to?

- First we calculate how many read units per item we need
- 10 KB rounded up to nearest increment of 4 KB is 12 KB.
- 12 KB / 4 KB = 3 read units per item.

- 3 x 5 read items = 15
- Using eventual consistency we get 15 / 2 = 7.5

- 8 units of read throughput

You have an application that requires to read 5 items of 10 KB per second using strong consistency. What should you set the read throughput to?

- First we calculate how many read units per item we need
- 10 KB rounded up to nearest increment of 4 KB is 12 KB.
- 12 KB / 4 KB = 3 read units per item.

- 3 x 5 read items = 15
- **Using strong consistency we DON'T divide by 2.**

- 15 units of read throughput

You have an application that requires to write 5 items, with each item being 10 KB in size per second. What should you set the write throughput to?

- Each write unit consist of 1 KB of data. You need to write 5 items per second with each item using 10 KB of data.

- 5 x 10 KB = 50 write units.

- Write throughput of 50 Units

You have an application that requires to write 12 items of 100 KB per item each second. What should you set the write throughput to?

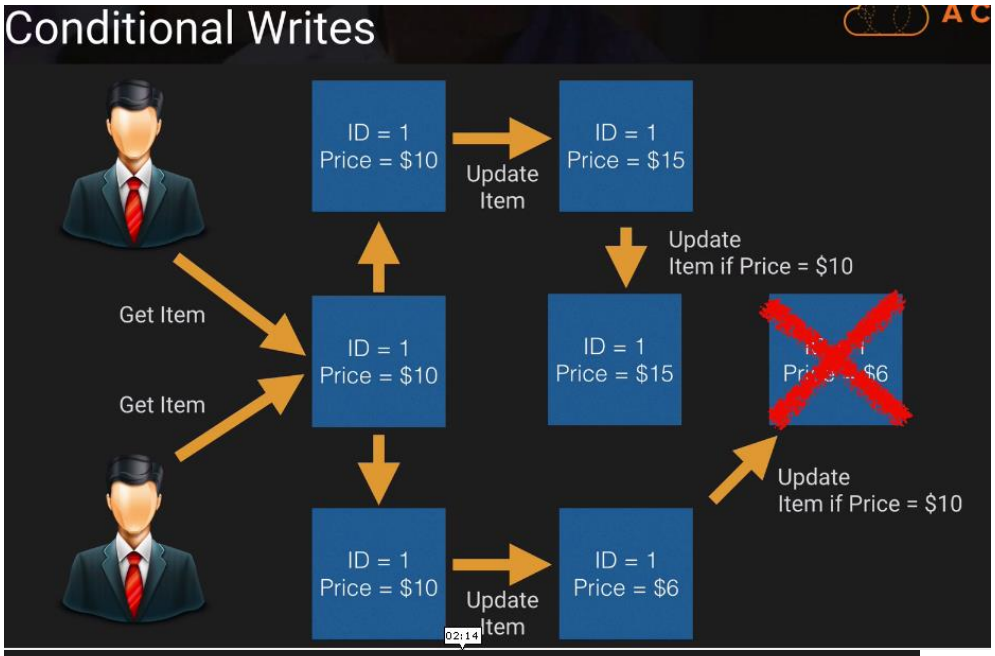
- Each write unit consist of 1 KB of data. You need to write 12 items per second with each item having 100 KB of data.

- 12 x 100 KB = 1200 write units.

- Write throughput of 1200 Units

400 HTTP Status Code - ProvisionedThroughputExceededException

You exceeded your maximum allowed provisioned throughput for a table or for one or more global secondary indexes.



If item = $10 then update to $12

Note that conditional writes are idempotent. This means that you can send the same conditional write request multiple times, but it will have no further effect on the item after the first time DynamoDB performs the specified update. For example, suppose you issue a request to update the price of a book item by 10%, with the expectation that the price is currently $20. However, before you get a response, a network error occurs and you don't know whether your request was successful or not. Because a conditional update is an idempotent operation, you can send the same request again. and DynamoDB will update the price only if the current price is still $20.

DynamoDB supports atomic counters, where you use the **UpdateItem** operation to increment or decrement the value of an existing attribute without interfering with other write requests. (All write requests are applied in the order in which they were received.) For example, a web application might want to maintain a counter per visitor to their site. In this case, the application would need to increment this counter regardless of its current value.

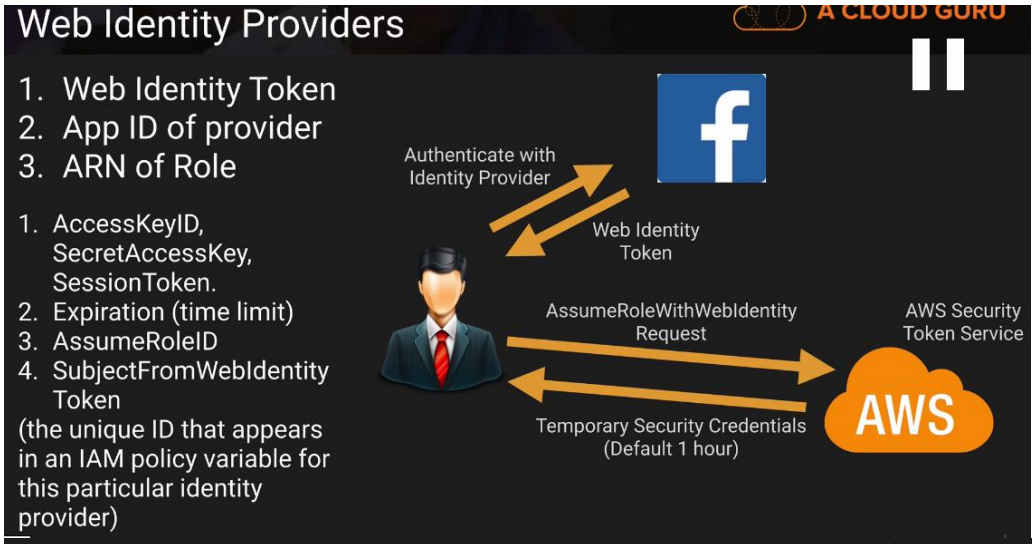
Atomic counter updates are not idempotent. This means that the counter will increment each time you call UpdateItem. If you suspect that a previous request was unsuccessful, your application could retry the UpdateItem operation; however, this would risk updating the counter twice. This might be acceptable for a web site counter, because you can tolerate with slightly over- or under-counting the visitors. However, in a banking application, it would be safer to use a conditional update rather than an atomic counter.

If your application needs to read multiple items, you can use the **BatchGetItem** API. A single **BatchGetItem** request can retrieve up to 1 MB of data, which can contain as many as 100 items. In addition, a single BatchGetItem request can retrieve items from multiple tables.

Web Identity Providers

You can authenticate users using Web Identity providers (such as Facebook, Google, Amazon or any other Open-ID Connect-compatible Identity provider). This is done using AssumeRoleWithWebIdentity API

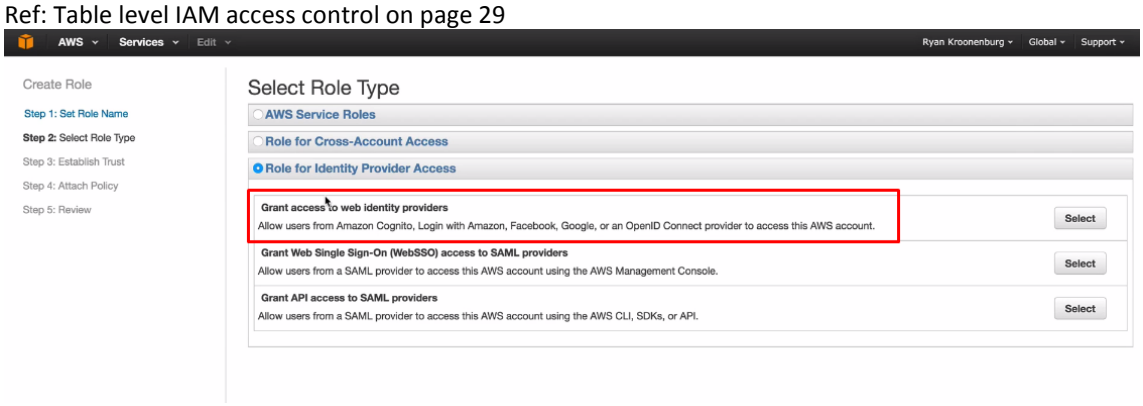
You will need to create a role first.



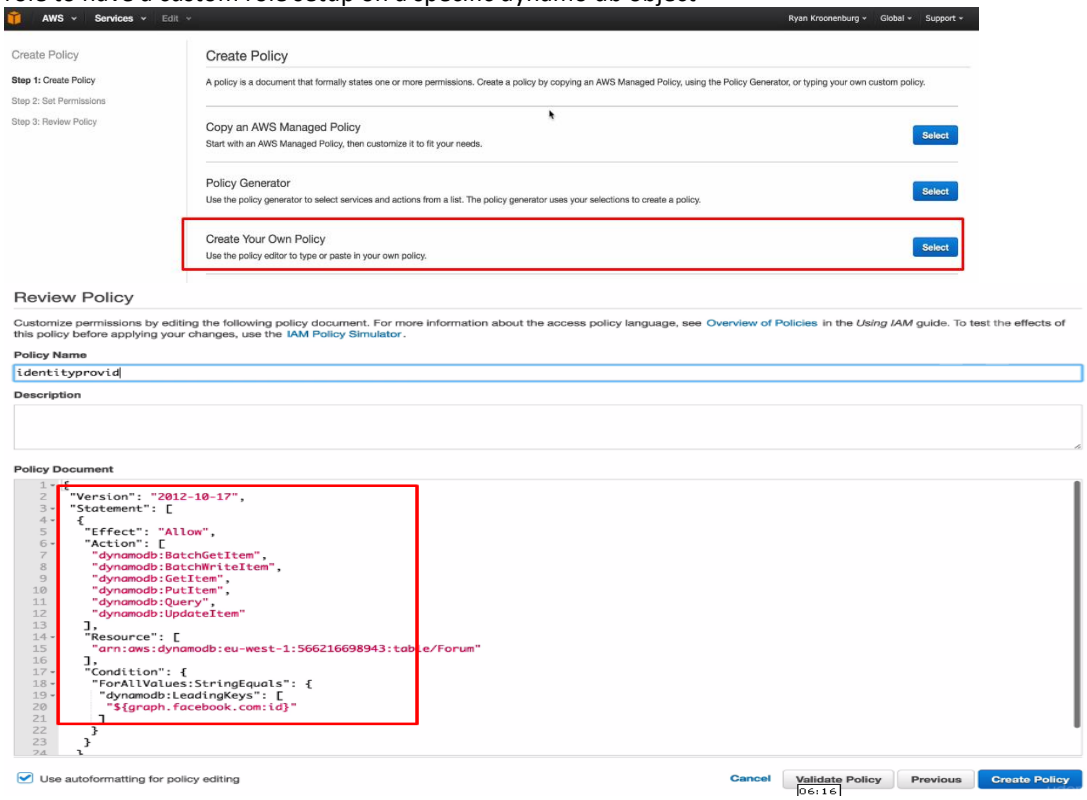
1. User Authenticates with ID provider (such as Facebook)
2. They are passed a Token by their ID provider.
3. Your code calls **AssumeRoleWithWebIdentity** API and provides the providers token and specifies the ARN for the IAM Role
4. App can now access Dynamodb from between 15 minutes to 1 hour (default is 1 hour).

ARN-Amazon Resource Name

Ref: Table level IAM access control on page 29



Once a role is created with unique App ID, create a new policy and paste document generated in step on page 29 over here in IAM role to have a custom role setup on a specific dynamo db object



Go back to the custom role created and attach this custom policy to that role that we created to have custom facebook or google web identity role

SQS-Simple Queue Service

Amazon SQS is a web service that gives you access to a message queue that can be used to store messages while waiting for a computer to process them.

Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. A queue is a temporary repository for messages that are awaiting processing.

Using Amazon SQS, you can decouple the components of an application so they run independently, with Amazon SQS easing message management between components. Any component of a distributed application can store messages in a fail-safe queue.

Messages can contain up to 256 KB of text in any format. Any component can later retrieve the messages programmatically using the Amazon SQS API.

The queue acts as a buffer between the component producing and saving data, and the component receiving the data for processing.

This means the queue resolves issues that arise if the producer is producing work faster than the consumer can process it, or if the producer or consumer are only intermittently connected to the network.

Amazon SQS is engineered to always be available and deliver messages. One of the resulting tradeoffs is that SQS does not guarantee first in, first out delivery of messages. For many distributed applications, each message can stand on its own, and as long as all messages are delivered, the order is not important.

If your system requires that order be preserved, you can place sequencing information in each message, so that you can reorder the messages when the queue returns them.

1. Asynchronously pulls the task messages from the queue
2. Retrieves the named file
3. Processes the conversion
4. Writes the image back to Amazon S3
5. Writes a “task complete” message to another queue
6. Deletes the original task message
7. Checks for more messages in the worker queue

- Does not offer FIFO
- 12 hours visibility time out
- Amazon SQS is engineered to provide “at least once” delivery of all messages in its queues. Although most of the time each message will be delivered to your application exactly once, you should design your system so that processing a message more than once does not create any errors or inconsistencies.
- 256kb message size now available
- Billed at 64kb “Chunks”
- A 256kb message will be 4 x 64kb “chunks”

SQS Pricing

- First 1 million Amazon SQS Requests per month are free
- $0.50 per 1 million Amazon SQS Requests per month thereafter ($0.00000050 per SQS Request)
- A single request can have from 1 to 10 messages, up to a maximum total payload of 256KB.
- Each 64KB ‘chunk’ of payload is billed as 1 request. For example, a single API call with a 256KB payload will be billed as four requests.

SQS - Default Visibility Time Out

Default Visibility Time Out is 30 Seconds

Maximum Time Out is 12 Hours

When you receive a message from a queue and begin processing it, you may find the visibility timeout for the queue is insufficient to fully process and delete that message. To give yourself more time to process the message, you can extend its visibility timeout by using the **ChangeMessageVisibility** action to specify a new timeout value. Amazon SQS restarts the timeout period using the new value.

SQS long polling is a way to retrieve messages from your SQS queues. While the traditional SQS short polling returns immediately, even if the queue being polled is empty, SQS long polling doesn’t return a response until a message arrives in the queue, or the long poll times out. SQS long polling makes it easy and inexpensive to retrieve messages from your SQS queue as soon as they are available.

Maximum Long Poll Time Out = 20 seconds

SNS – Simple Notification Services – JSON based Service

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

It provides developers with a highly scalable, flexible, and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or other applications.

Amazon SNS follows the "publish-subscribe" (pub-sub) messaging paradigm, with notifications being delivered to clients using a "push" mechanism that eliminates the need to periodically check or "poll" for new information and updates.

With simple APIs requiring minimal up-front development effort, no maintenance or management overhead and pay-as-you-go pricing, Amazon SNS gives developers an easy mechanism to incorporate a powerful notification system with their applications.

Push notifications to Apple, Google, Fire OS, and Windows devices, as well as Android devices in China with Baidu Cloud Push.

Besides pushing cloud notifications directly to mobile devices, Amazon SNS can also deliver notifications by SMS text message or email, to Amazon Simple Queue Service (SQS) queues, or to any HTTP endpoint.

To prevent messages from being lost, all messages published to Amazon SNS are stored redundantly across multiple availability zones.

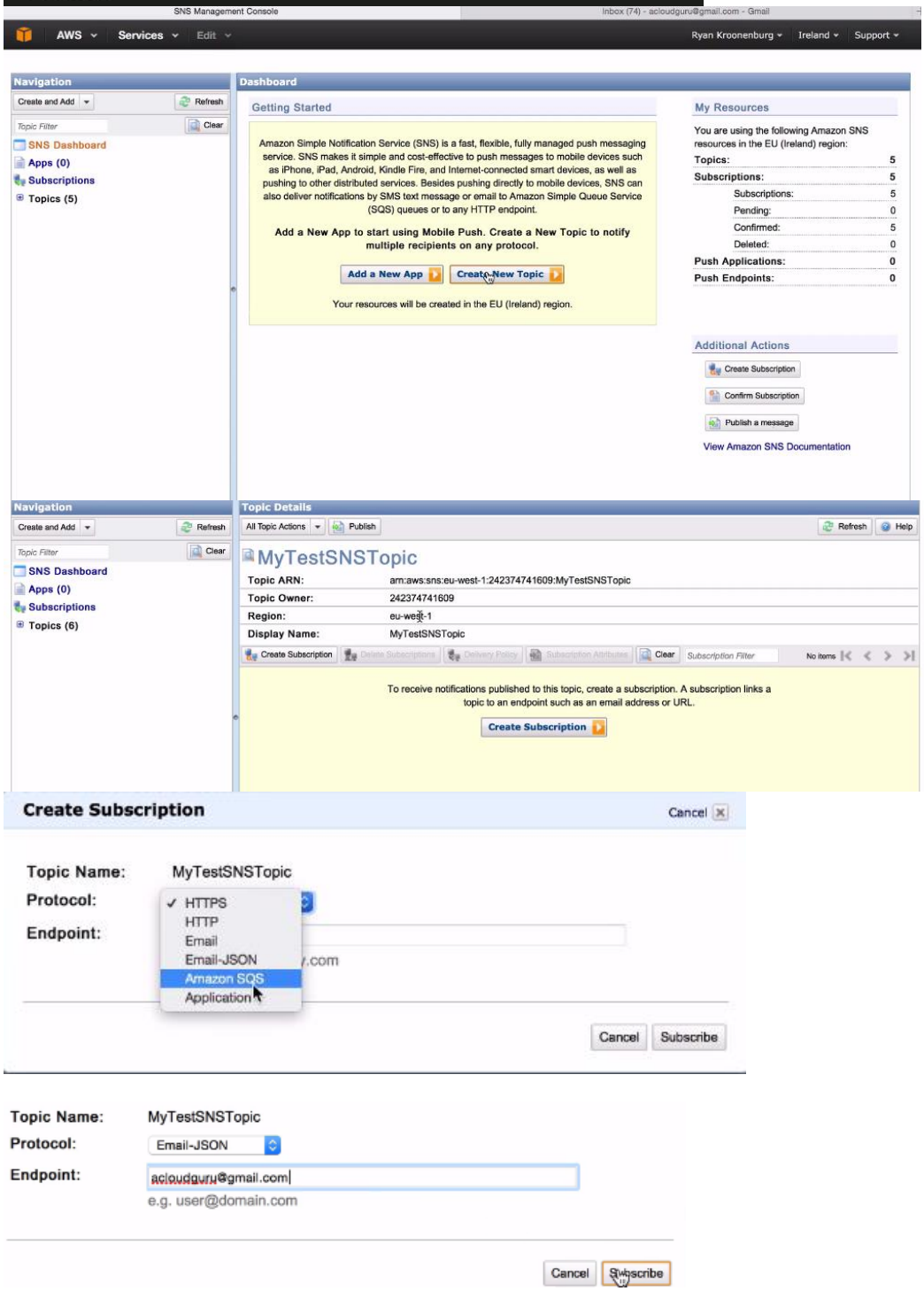
SNS allows you to group multiple recipients using topics. A topic is an "access point" for allowing recipients to dynamically subscribe for identical copies of the same notification.

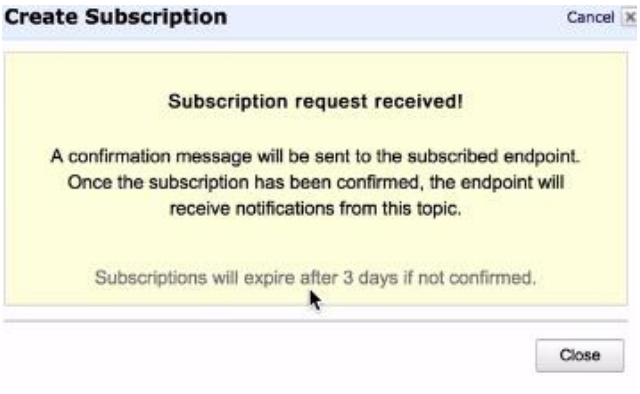
One topic can support deliveries to multiple endpoint types -- for example, you can group together iOS, Android and SMS recipients. When you publish once to a topic, SNS delivers appropriately formatted copies of your message to each subscriber.

- Instantaneous, push-based delivery (no polling)
- Simple APIs and easy integration with applications
- Flexible message delivery over multiple transport protocols
- Inexpensive, pay-as-you-go model with no up-front costs
- Web-based AWS Management Console offers the simplicity of a point-and-click interface

- Both Messaging Services in AWS
- SNS - Push
- SQS - Polls (Pulls)

- Users pay $0.50 per 1 million Amazon SNS Requests
- $0.06 per 100,000 Notification deliveries over HTTP
- $0.75 per 100 Notification deliveries over SMS
- $2.00 per 100,000 Notification deliveries over Email




Create Subscription
Cancel
Subscription request received!
A confirmation message will be sent to the subscribed endpoint. Once the subscription has been confirmed, the endpoint will receive notifications from this topic.
Subscriptions will expire after 3 days if not confirmed.
Close

AWS Notification - Subscription Confirmation Inbox x

MyTestSNSTopic no-reply@sns.amazonaws.com via amazonses.com 2:42 PM (0 minutes ago)
to me

{
"Type" : "SubscriptionConfirmation",
"MessageId" : "59954705-229f-47e4-a734-b8fc8e0a5b21",
"Token" : "2336412f37fb687f5d51e6e241d7700ae500a62eb0d11a07328df3fc47a6f56f43a53f94495c2cd5ca857a39bd
1f4db6dffb9217e9e4803d99f3b107de95356bd7620c1f7a1c72d6ce08860d3fc85ed5bb70205071825fe0e571
05028d43b943a6495a4c5bfbded6fd3ff1365bd13656390b07e2cb6bc19d37d907a517fe35a2",
"TopicArn" : "arn:aws:sns:eu-west-1:242374741609:MyTestSNSTopic",
"Message" : "You have chosen to subscribe to the topic arn:aws:sns:eu-west-1:242374741609:MyTestSNSTopic.\nTo confirm the subscription, visit the SubscribeURL included in this message.",
"SubscribeURL" : "https://sns.eu-west-1.amazonaws.com/?Action=ConfirmSubscription&TopicArn=arn:aws:sns:eu-west-1:242374741609:MyTestSNSTopic&Token=2336412f37fb687f5d51e6e241d7700ae500a62eb0d11a07328df3fc47a6f56f43a53f94495c2cd5ca857a39bd1f4db6dffb9217e9e4803d99f3b107de95356bd7620c1f7a1c72d6ce08860d3fc85ed5bb70205071825fe0e57105028d43b943a6495a4c5bfbded6fd3ff1365bd13656390b07e2cb6bc19d37d907a517fe35a2",
"Timestamp" : "2015-02-04T14:42:31.522Z",
"SignatureVersion" : "1",
"Signature" : "YZaVV5wQ93g1UYA8xGn+tGwqLRc72ry489b9rdn8sjOd6B3AKSj/FMg7vL7biNylYj3HcI7Sx51pC8qQMRP30ZgUE93jlmxxVdQD161FxI4jCpT/62SRyqvfGzl3u68J4oEAJPQMo/O+7uCLBhfgLGxY2nKolaAX4TUx9BNY8W7PX9jRF9EJgZD7158l084H5mDzTAnq5RvytzhWpcVSLof9784MkRMoN1ay5hsRQf9j/Jst9VaodEwfDjCfHSyNcC+IZbEKcs+BgYOKg4CLs0+1uYRiZ+XM4U1VFeKmXnqHFjaad3OVvHI7VaXWJclEcEE6aS/6z2+Uf+xcLcYHfA==",
"SigningCertURL" : "https://sns.eu-west-1.amazonaws.com/SimpleNotificationService-d6d679a1d18e95c2f9ffcf11f4f9e198.pem"
}

Publish Cancel

Topic Name: MyTestSNSTopic

Subject: This is a second test

Up to 100 printable ASCII characters (optional).

Message: Hello World!!!!!

Up to 256KB of Unicode text.

Time to Live (TTL):

TTL is the number of seconds since the message was published. When you use TTL, messages that remain undelivered for the specified time will expire.

- (•) Use same message body for all protocols
- ( ) Use different message body for different protocols

Cancel | Publish Message

Publish Cancel

Topic Name: MyTestSNSTopic

Subject: This is a second test

Up to 100 printable ASCII characters (optional).

Message:

```
{
  "default": "ENTER YOUR MESSAGE",
  "email": "ENTER YOUR MESSAGE",
  "sqs": "ENTER YOUR MESSAGE",
  "http": "ENTER YOUR MESSAGE",
  "https": "ENTER YOUR MESSAGE",
  "sms": "ENTER YOUR MESSAGE",
  "APNS": "{\"aps\":{\"alert\": \"ENTER YOUR MESSAGE\",\"sound\":\"default\"} }",
  "GCM": "{ \"data\": { \"message\": \"ENTER YOUR MESSAGE\" } }",
  "ADM": "{ \"data\": { \"message\": \"ENTER YOUR MESSAGE\" } }",
  "BAIDU": "{\"title\":\"ENTER YOUR TITLE\",\"description\":\"ENTER
```

Up to 256KB of Unicode text.

Time to Live (TTL):

TTL is the number of seconds since the message was published. When you use TTL, messages that remain undelivered for the specified time will expire.

- ( ) Use same message body for all protocols
- (•) Use different message body for different protocols

Cancel | Publish Message

This is a second test Inbox x

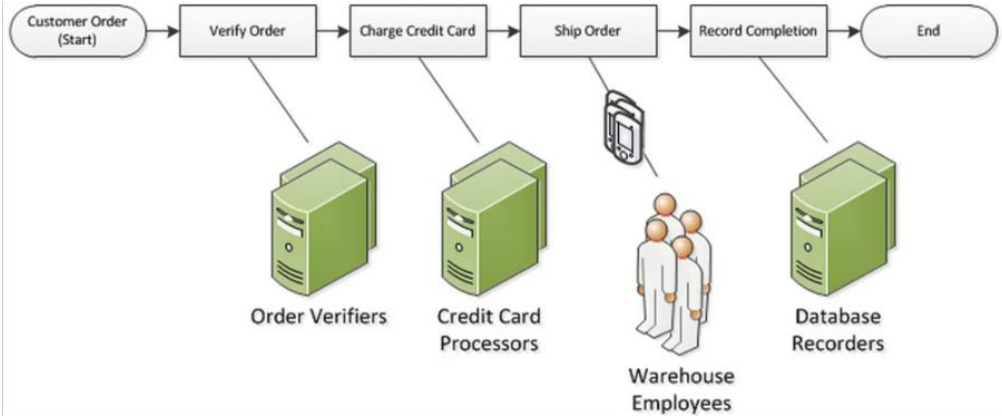
MyTestSNSTopic no-reply@sns.amazonaws.com via amazonses.com 2:45 PM (0 minutes ago)

to me


```
{
  "Type" : "Notification",
  "MessageId" : "55ac3dfd-bd67-5a7d-a606-70b18fd499f7",
  "TopicArn" : "arn:aws:sns:eu-west-1:242374741609:MyTestSNSTopic",
  "Subject" : "This is a second test",
  "Message" : "hello world 2",
  "Timestamp" : "2015-02-04T14:45:07.664Z",
  "SignatureVersion" : "1",
  "Signature" : "Yz3p+WtfFTm8GUMNcYILWfa9wCGC0MN+eKAQ4Vvi4tYDimsn8Dhde7EZ+
3efiIRNNPZnqEj4WkR89GmKUGkwQkE48/j2HnFaWSwTNim6+XtbWwrDq4d4LMB/YBVLkeK2ZETHOIKMLtrOseIhF8Mrdr
ispl6owMdUW4zoHBGGZlMoyiQNTHBMalypdzSl+HOHYk6XFz/jcK95GNCVECdPlezhxVX7WeAe8tsOt
JYZeOERY2fKVAIJUeGWJ5rm4MmPhrSgJXxVDWm6uHWOZLXWSHI+QJ0U7mQW3k1YRAnfk5ZLiRqVzJKeSZ
iyy6VtOvRPX4i2h+cb9lqy/ek8dR1uzw==",
  "SigningCertURL" : "https://sns.eu-west-1.amazonaws.com/SimpleNotificationService-d6d679a1d18e95c2f9ffcf11f4f9e1
98.pem",
  "UnsubscribeURL" : "https://sns.eu-west-1.amazonaws.com/?Action=Unsubscribe&SubscriptionArn=arn:aws:sns:eu-west-1:
242374741609:MyTestSNSTopic:0eb6b983-87b8-449f-b666-3a2a2209c04a",
  "MessageAttributes" : {
    "AWS.SNS.MOBILE.MPNS.Type" : {"Type":"String","Value":"token"},
    "AWS.SNS.MOBILE.WNS.Type" : {"Type":"String","Value":"wns/badge"},
    "AWS.SNS.MOBILE.MPNS.NotificationClass" : {"Type":"String","Value":"realtime"}
  }
}
```


Amazon Simple Workflow Service (Amazon SWF) is a web service that makes it easy to coordinate work across distributed application components. Amazon SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.

Tasks represent invocations of various processing steps in an application which can be performed by executable code, web service calls, human actions, and scripts.



Workers are programs that interact with Amazon SWF to get tasks, process received tasks, and return the results.

The decider is a program that controls the coordination of tasks, i.e. their ordering, concurrency, and scheduling according to the application logic.

The workers and the decider can run on cloud infrastructure, such as Amazon EC2, or on machines behind firewalls. Amazon SWF brokers the interactions between workers and the decider. It allows the decider to get consistent views into the progress of tasks and to initiate new tasks in an ongoing manner.

At the same time, Amazon SWF stores tasks, assigns them to workers when they are ready, and monitors their progress. It ensures that a task is assigned only once and is never duplicated. Since Amazon SWF maintains the application's state durably, workers and deciders don't have to keep track of execution state. They can run independently, and scale quickly.

Your workflow and activity types and the workflow execution itself are all scoped to a domain. Domains isolate a set of types, executions, and task lists from others within the same account.

You can register a domain by using the AWS Management Console or by using the RegisterDomain action in the Amazon SWF API.

SWF Domains

The parameters are specified in JavaScript Object Notation (JSON) format.

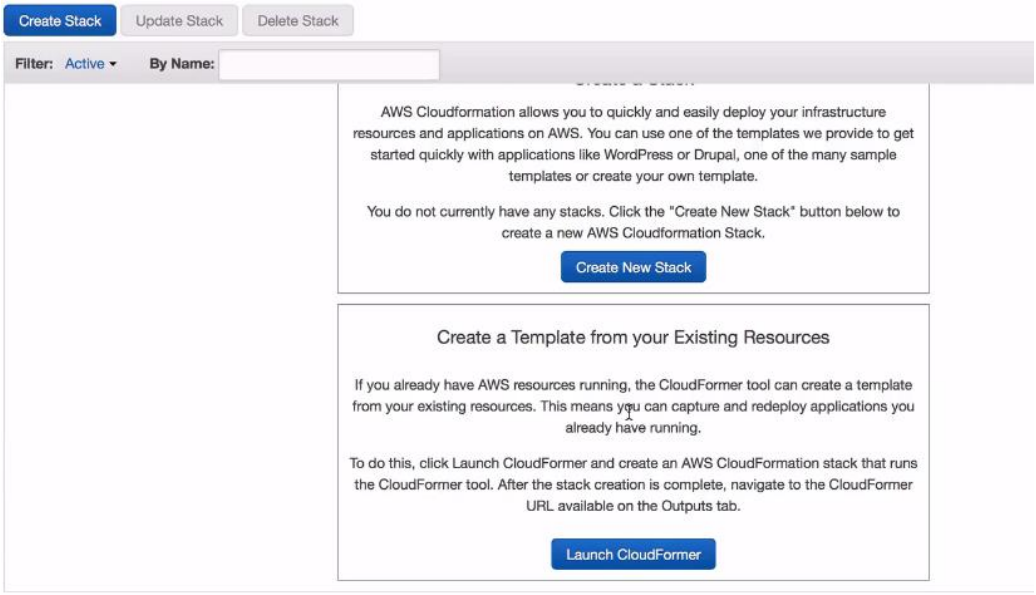
```
https://swf.us-east-1.amazonaws.com
RegisterDomain
{
  "name" : "867530901",
  "description" : "music",
  "workflowExecutionRetentionPeriodInDays" : "60"
}
```

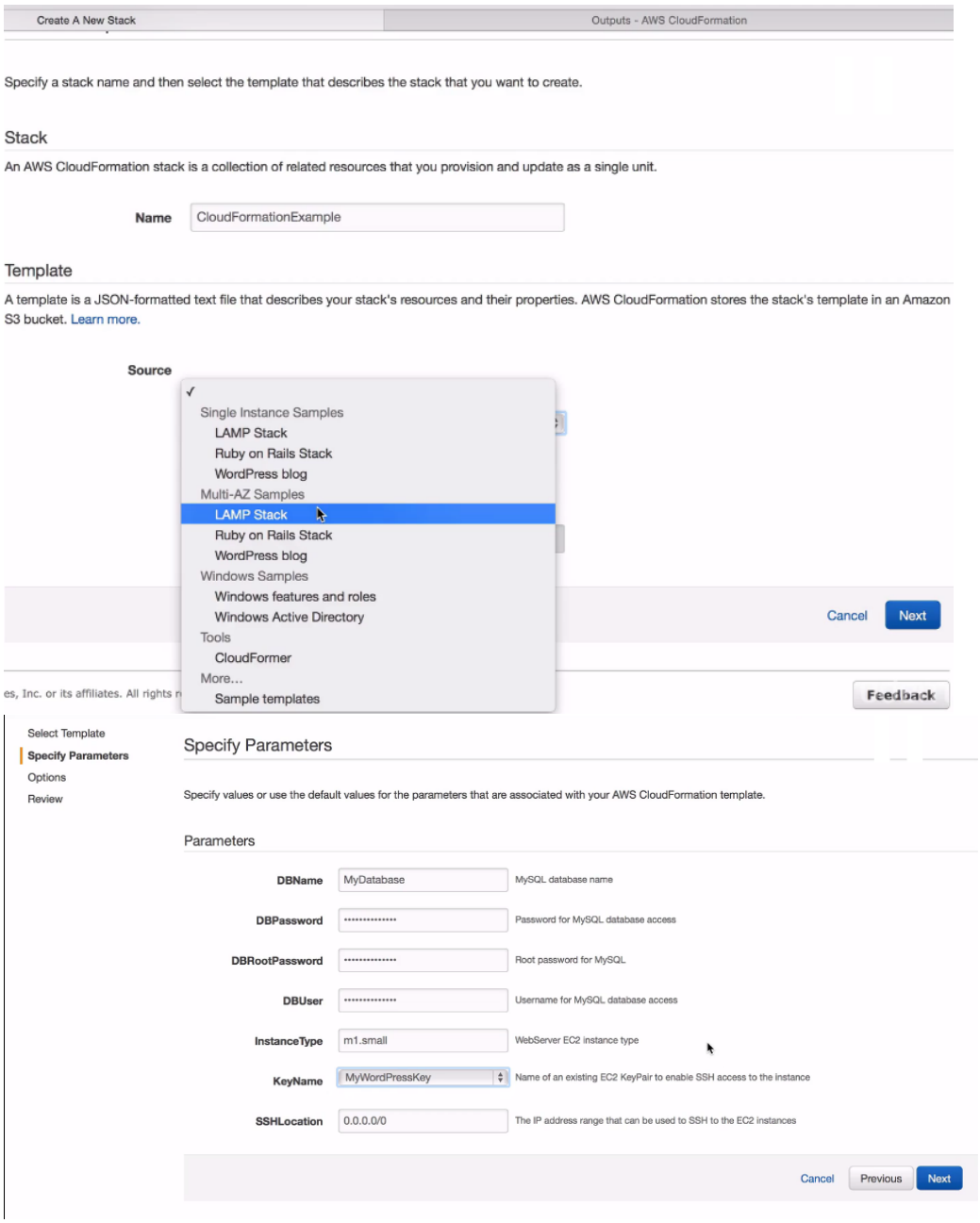
Maximum Workflow can be 1 year and the value is always measured in seconds.

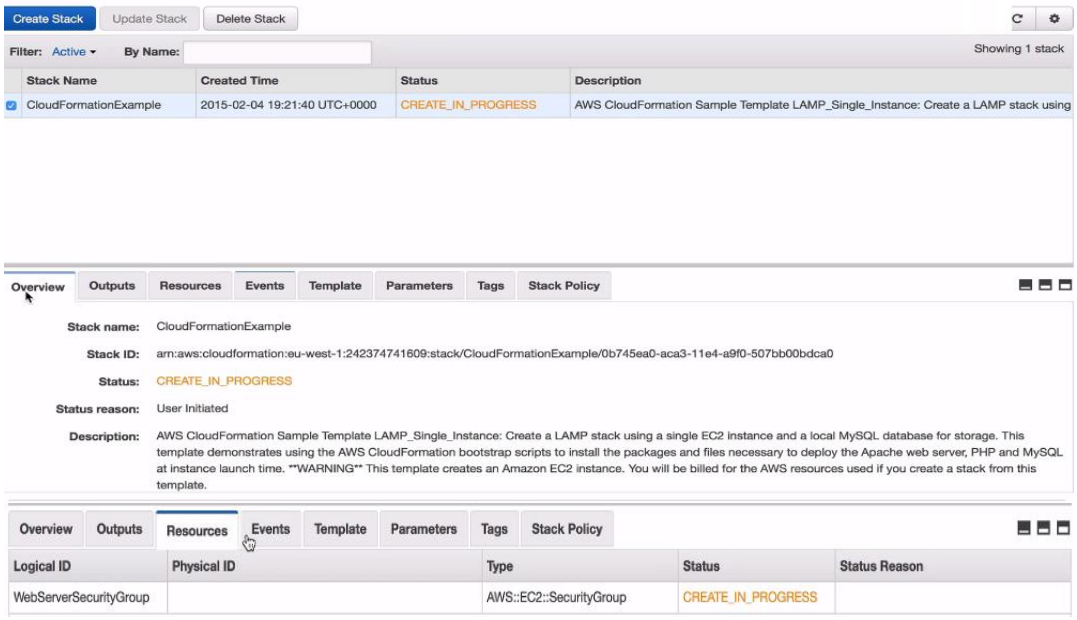
SWF vs SQS

- Amazon SWF presents a task-oriented API, whereas Amazon SQS offers a message-oriented API.
- Amazon SWF ensures that a task is assigned only once and is never duplicated. With Amazon SQS, you need to handle duplicated messages and may also need to ensure that a message is processed only once.
- Amazon SWF keeps track of all the tasks and events in an application. With Amazon SQS, you need to implement your own application-level tracking, especially if your application uses multiple queues.

Cloud Formation – itself is but we pay for the resources its gonna install

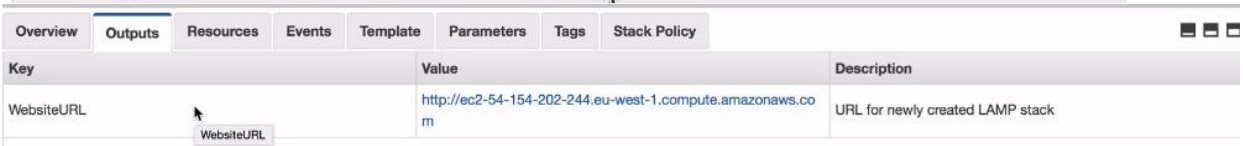



Create A New Stack
Outputs - AWS CloudFormation
Specify a stack name and then select the template that describes the stack that you want to create.
Stack
An AWS CloudFormation stack is a collection of related resources that you provision and update as a single unit.
Name
CloudFormationExample
Template
A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. Learn more.
Source
Single Instance Samples
LAMP Stack
Ruby on Rails Stack
WordPress blog
Multi-AZ Samples
LAMP Stack
Ruby on Rails Stack
WordPress blog
Windows Samples
Windows features and roles
Windows Active Directory
Tools
CloudFormer
More...
Sample templates
Cancel
Next
Feedback
Select Template
Specify Parameters
Options
Review
Specify Parameters
Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.
Parameters
DBName
MyDatabase
MySQL database name
DBPassword
Password for MySQL database access
DBRootPassword
Root password for MySQL
DBUser
Username for MySQL database access
InstanceType
m1.small
WebServer EC2 instance type
KeyName
MyWordPressKey
Name of an existing EC2 KeyPair to enable SSH access to the instance
SSHLocation
0.0.0.0/0
The IP address range that can be used to SSH to the EC2 instances
Cancel
Previous
Next


Create Stack
Update Stack
Delete Stack
Filter: Active
By Name:
Showing 1 stack
Stack Name
Created Time
Status
Description
CloudFormationExample
2015-02-04 19:21:40 UTC+0000
CREATE_IN_PROGRESS
AWS CloudFormation Sample Template LAMP_Single_Instance: Create a LAMP stack using

Overview
Outputs
Resources
Events
Template
Parameters
Tags
Stack Policy
Stack name: CloudFormationExample
Stack ID: arn:aws:cloudformation:eu-west-1:242374741609:stack/CloudFormationExample/0b745ea0-aca3-11e4-a9f0-507bb00bdca0
Status: CREATE_IN_PROGRESS
Status reason: User Initiated
Description: AWS CloudFormation Sample Template LAMP_Single_Instance: Create a LAMP stack using a single EC2 instance and a local MySQL database for storage. This template demonstrates using the AWS CloudFormation bootstrap scripts to install the packages and files necessary to deploy the Apache web server, PHP and MySQL at instance launch time. **WARNING** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

Overview
Outputs
Resources
Events
Template
Parameters
Tags
Stack Policy
Logical ID
Physical ID
Type
Status
Status Reason
WebServerSecurityGroup
AWS::EC2::SecurityGroup
CREATE_IN_PROGRESS

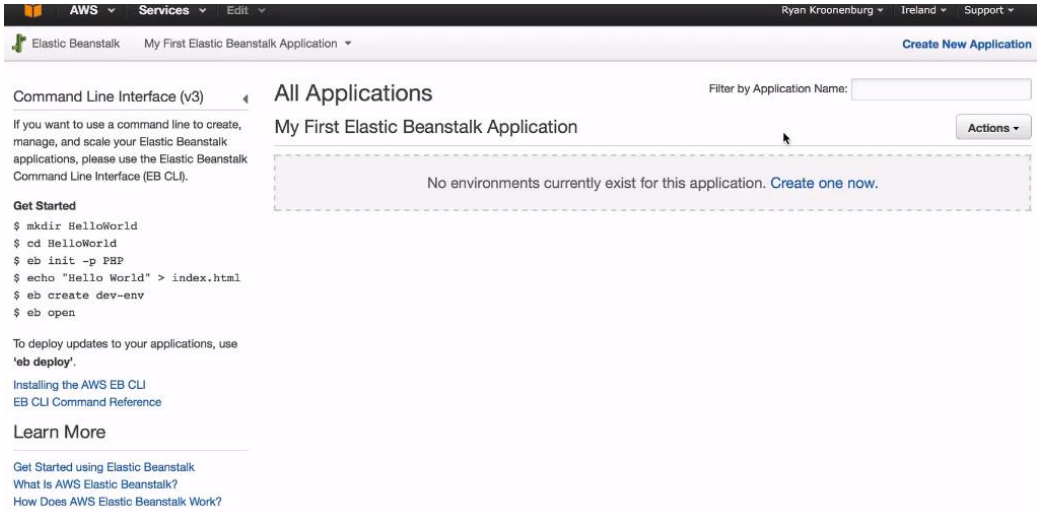


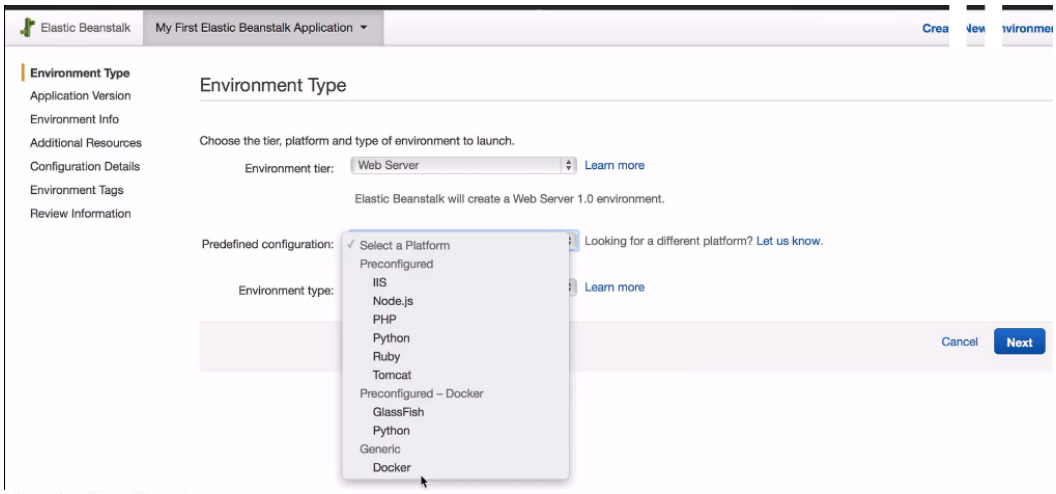

```
"Outputs" : {
  "Logical ID" : {
    "Description" : "Information about the value",
    "Value" : "Value to return"
  }
}
```

```
"Outputs" : {
  "BackupLoadBalancerDNSName" : {
    "Description": "The DNSName of the backup load balancer",
    "Value" : { "Fn::GetAtt" : [ "BackupLoadBalancer", "DNSName" ]},
    "Condition" : "CreateProdResources"
  },
  "InstanceID" : {
    "Description": "The Instance ID",
    "Value" : { "Ref" : "EC2Instance" }
  }
}
```

Fn:GetAtt: its the value that we use to return a result of key value pair. In above example, getAtt will return me DNSName of the BackUpLoadBalancer.
Rollback on cloud formation is enabled by default

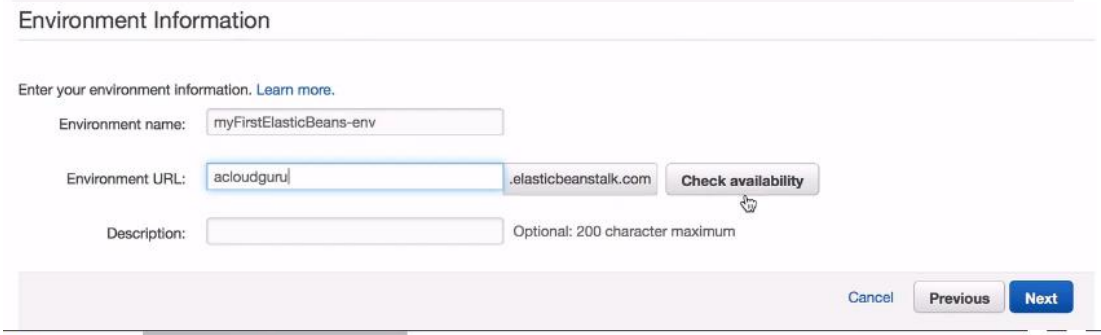
ElasticBeanstalk: service like cloud formation which is free by AWS but we pay for the resources we are gonna use via elastic beanstalk

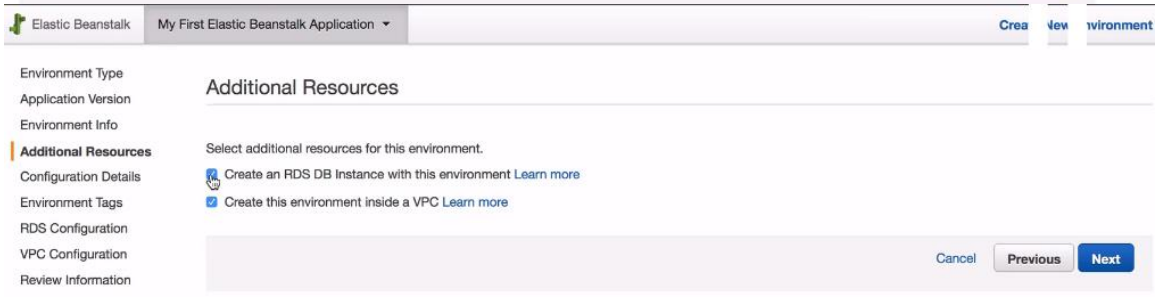


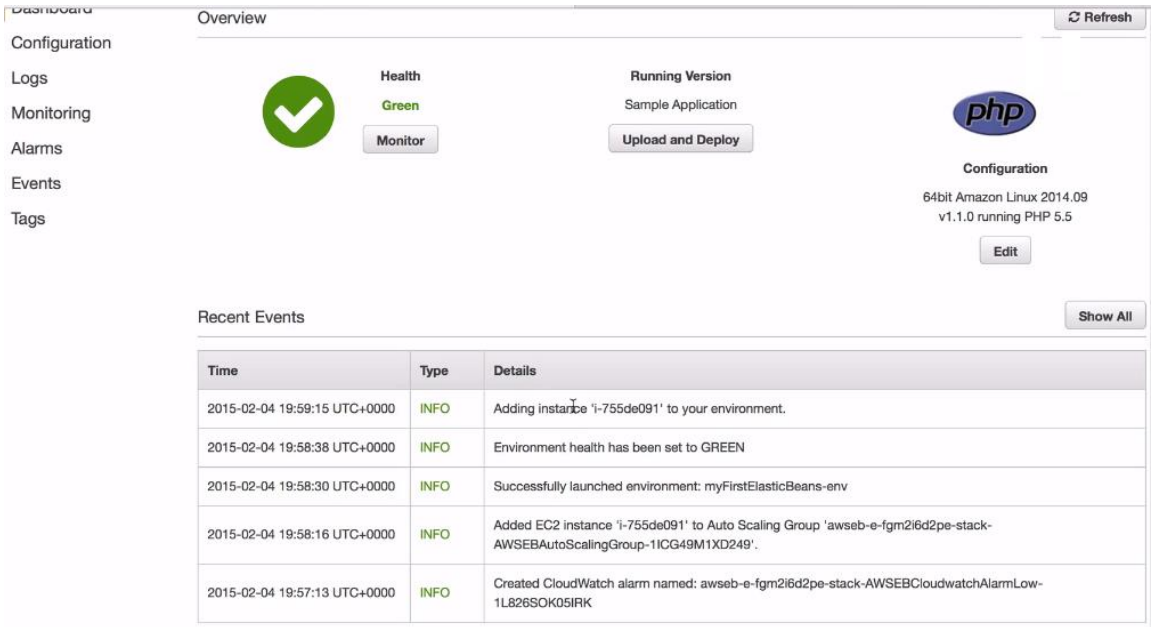

Elastic Beanstalk
My First Elastic Beanstalk Application
Environment Type
Application Version
Environment Info
Additional Resources
Configuration Details
Environment Tags
Review Information
Environment Type
Choose the tier, platform and type of environment to launch.
Environment tier:
Web Server
Learn more
Elastic Beanstalk will create a Web Server 1.0 environment.
Predefined configuration:
Select a Platform
Preconfigured
IIS
Node.js
PHP
Python
Ruby
Tomcat
Preconfigured – Docker
GlassFish
Python
Generic
Docker
Looking for a different platform? Let us know.
Environment type:
Learn more
Cancel
Next

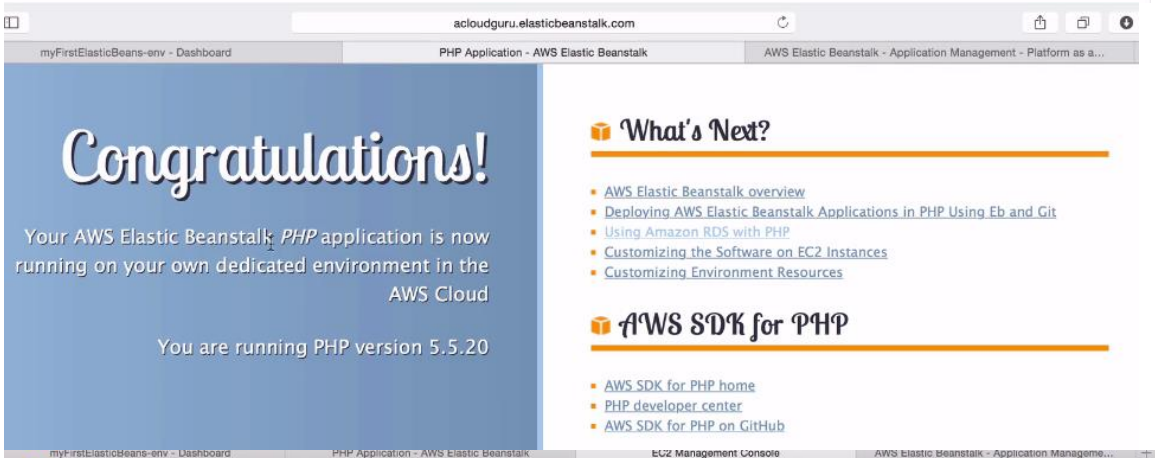

Application Version
Select a source for your application version.
Source:
Existing application version
Sample Application
Upload your own (Learn more)
Choose File
no file selected
S3 URL
(e.g. https://s3.amazonaws.com/s3Bucket/s3Key)

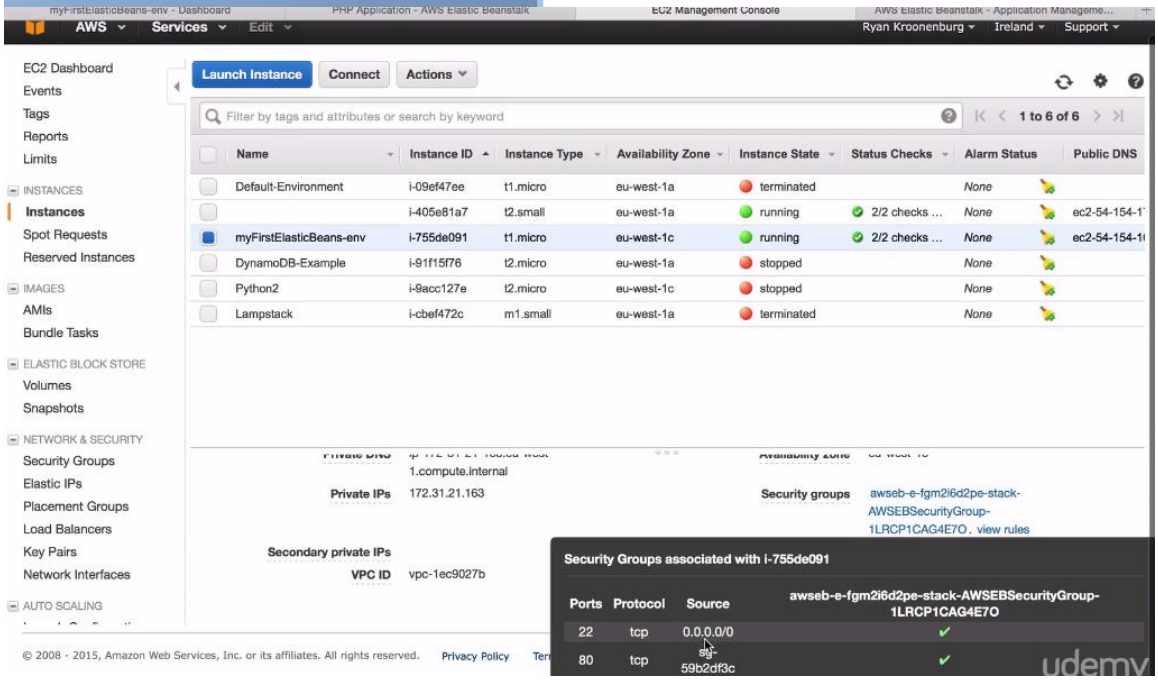

Deployment Limits
Elastic Beanstalk will update your application in batches so as to avoid downtime when deploying. Learn more
Batch size:
Percentage
30
%
of the fleet at a time
Fixed
1
instances at a time
Cancel
Previous
Next


Environment Information
Enter your environment information. Learn more.
Environment name:
myFirstElasticBeans-env
Environment URL:
acloudguru
.elasticbeanstalk.com
Check availability
Description:
Optional: 200 character maximum
Cancel
Previous
Next


Elastic Beanstalk
My First Elastic Beanstalk Application
Environment Type
Application Version
Environment Info
Additional Resources
Configuration Details
Environment Tags
RDS Configuration
VPC Configuration
Review Information
Additional Resources
Select additional resources for this environment.
Create an RDS DB Instance with this environment Learn more
Create this environment inside a VPC Learn more
Cancel
Previous
Next


Dashboard
Configuration
Logs
Monitoring
Alarms
Events
Tags
Overview
Refresh
Health
Green
Monitor
Running Version
Sample Application
Upload and Deploy
Configuration
64bit Amazon Linux 2014.09
v1.1.0 running PHP 5.5
Edit
Recent Events
Show All
Time
Type
Details
2015-02-04 19:59:15 UTC+0000
INFO
Adding instance 'i-755de091' to your environment.
2015-02-04 19:58:38 UTC+0000
INFO
Environment health has been set to GREEN
2015-02-04 19:58:30 UTC+0000
INFO
Successfully launched environment: myFirstElasticBeans-env
2015-02-04 19:58:16 UTC+0000
INFO
Added EC2 instance 'i-755de091' to Auto Scaling Group 'awseb-e-fgm2i6d2pe-stack-AWSEBAutoScalingGroup-1ICG49M1XD249'.
2015-02-04 19:57:13 UTC+0000
INFO
Created CloudWatch alarm named: awseb-e-fgm2i6d2pe-stack-AWSEBCloudwatchAlarmLow-1L826SOK05IRK


acloudguru.elasticbeanstalk.com
myFirstElasticBeans-env - Dashboard
PHP Application - AWS Elastic Beanstalk
AWS Elastic Beanstalk - Application Management - Platform as a...
Congratulations!
Your AWS Elastic Beanstalk PHP application is now running on your own dedicated environment in the AWS Cloud
You are running PHP version 5.5.20
What's Next?
AWS Elastic Beanstalk overview
Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git
Using Amazon RDS with PHP
Customizing the Software on EC2 Instances
Customizing Environment Resources
AWS SDK for PHP
AWS SDK for PHP home
PHP developer center
AWS SDK for PHP on GitHub


AWS
Services
Edit
Ryan Kroonenburg
Ireland
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Spot Requests
Reserved Instances
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Load Balancers
Key Pairs
Network Interfaces
AUTO SCALING
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 6 of 6
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS
Default-Environment
i-09ef47ee
t1.micro
eu-west-1a
terminated
None
i-405e81a7
t2.small
eu-west-1a
running
2/2 checks ...
None
myFirstElasticBeans-env
i-755de091
t1.micro
eu-west-1c
running
2/2 checks ...
None
DynamoDB-Example
i-91f15f76
t2.micro
eu-west-1a
stopped
None
Python2
i-9acc127e
t2.micro
eu-west-1c
stopped
None
Lampstack
i-cbef472c
m1.small
eu-west-1a
terminated
None
1.compute.internal
Private IPs
172.31.21.163
Security groups
awseb-e-fgm2i6d2pe-stack-AWSEBSecurityGroup-1LRCP1CAG4E7O. view rules
Secondary private IPs
VPC ID
vpc-1ec9027b
Security Groups associated with i-755de091
Ports
Protocol
Source
awseb-e-fgm2i6d2pe-stack-AWSEBSecurityGroup-1LRCP1CAG4E7O
22
tcp
0.0.0.0/0
80
tcp
sg-59b2df3c
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
udemy

Configurations

Environment Properties

The following properties are passed into the application as environment variables. Learn more.

Property Name	Property Value
AWS_ACCESS_KEY_ID Specifying this and AWS_SECRET_KEY provides your credentials to your application in the environment properties.	
AWS_SECRET_KEY Specifying this and AWS_ACCESS_KEY_ID provides your credentials to your application in the environment properties.	
PARAM1 A predefined environment property that will be available to your running application.	
PARAM2 A predefined environment property that will be available to your running application.	

My First Elastic Beanstalk Application ▸ myFirstElasticBeans-env (acloudguru.elasticbeanstalk.com)

Actions

- Save Configuration
- Load Configuration
- Swap Environment URLs
- Restart App Server(s)
- Rebuild Environment
- Terminate Environment

- Dashboard
- Configuration
- Logs
- Monitoring
- Alarms
- Events
- Tags

Tags

Key	Value
Name	myFirstElasticBeans-env
elasticbeanstalk:environment-name	myFirstElasticBeans-env
Beanstalkname	MyElasticBeanstalk
elasticbeanstalk:environment-id	e-fgm2i6d2pe

Shared Responsibilty Model – security considerations

Shared Responsibility Model for Infrastructure Services

Infrastructure services, such as Amazon EC2, Amazon EBS, and Amazon VPC, run on top of the AWS global infrastructure. They vary in terms of availability and durability objectives but always operate within the specific region where they have been launched. You can build systems that meet availability objectives exceeding those of individual services from AWS by employing resilient components in multiple Availability Zones.

Figure 1 depicts the building blocks for the shared responsibility model for infrastructure services.

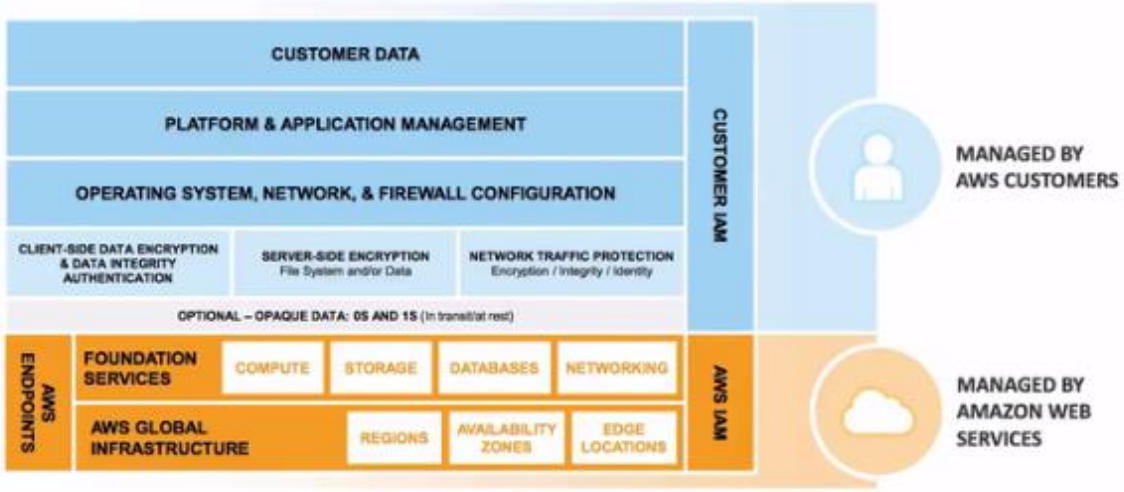


Figure 1: Shared Responsibility Model for Infrastructure Services

Building on the AWS secure global infrastructure, you install and configure your operating systems and platforms in the AWS cloud just as you would do on premises in your own data centers. Then you install your applications on your platform. Ultimately, your data resides in and is managed by your own applications. Unless you have more stringent business or compliance requirements, you don't need to introduce additional layers of protection beyond those provided by the AWS secure global infrastructure.

For certain compliance requirements, you might require an additional layer of protection between the services from AWS and your operating systems and platforms, where your applications and data reside. You can impose additional controls, such as protection of data at rest, and protection of data in transit, or introduce a layer of opacity between services from AWS and your platform. The opacity layer can include data encryption, data integrity authentication, software- and data-signing, secure time-stamping, and more.

Shared Responsibility Model for Container Services

The AWS shared responsibility model also applies to container services, such as Amazon RDS and Amazon EMR. For these services, AWS manages the underlying infrastructure and foundation services, the operating system and the application platform. For example, Amazon RDS for Oracle is a managed database service in which AWS manages all the layers of the container, up to and including the Oracle database platform. For services such as Amazon RDS, the AWS platform provides data backup and recovery tools; but it is your responsibility to configure and use tools in relation to your business continuity and disaster recovery (BC/DR) policy.

For AWS Container services, you are responsible for the data and for firewall rules for access to the container service. For example, Amazon RDS provides RDS security groups, and Amazon EMR allows you to manage firewall rules through Amazon EC2 security groups for Amazon EMR instances.

Figure 2 depicts the shared responsibility model for container services.

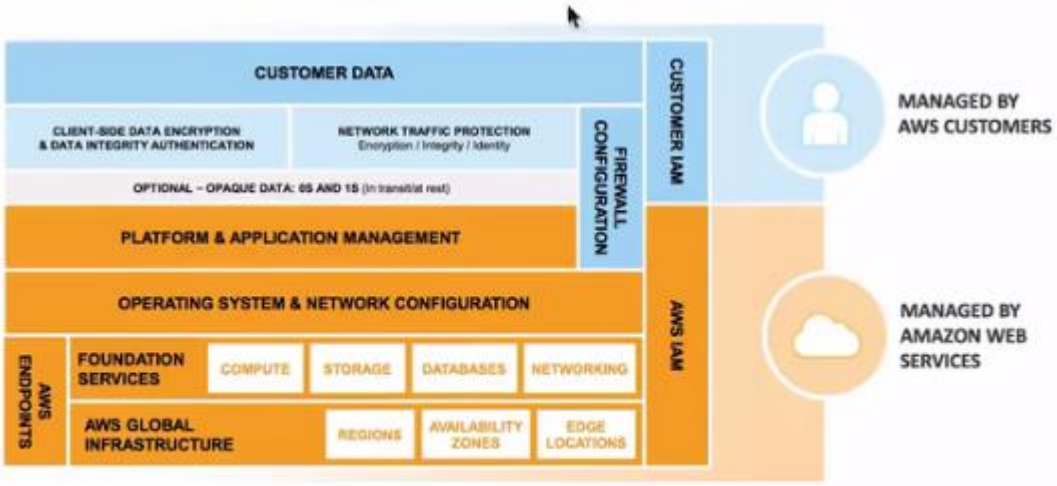


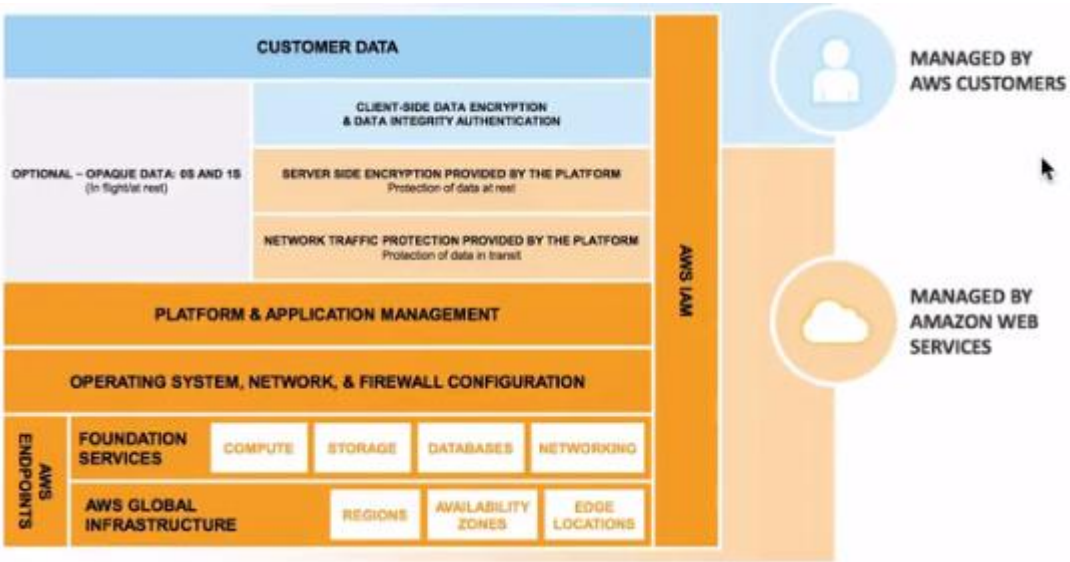
Figure 2: Shared Responsibility Model for Container Services

Shared Responsibility Model for Abstracted Services

For abstracted services, such as Amazon S3 and Amazon DynamoDB, AWS operates the infrastructure layer, the operating system, and platforms and you access the endpoints to store and retrieve data. Amazon S3 and DynamoDB are tightly integrated with IAM. You are responsible for managing your data (including classifying your assets), and for using IAM tools to apply ACL-type permissions to individual resources at the platform level, or permissions based on user identity or user responsibility at the IAM user/group level. For some services, such as Amazon S3, you can also use platform-provided encryption of data at rest, or platform-provided HTTPS encapsulation for your payloads for protecting your data in transit to and from the service.

Figure 3 outlines the shared responsibility model for AWS abstracted services:

This is inclusive of lambda services



If you've used the internet, you've used DNS. DNS is used to convert human friendly domain names (such as http://acloud.guru) into an Internet Protocol (IP) address (such as http://82.124.53.1).

IP addresses are used by computers to identify each other on the network. IP addresses commonly come in 2 different forms, IPv4 and IPv6.

IPv4 vs IPv6

The IPv4 space is a 32 bit field and has over 4 billion different addresses (4,294,967,296 to be precise).

IPv6 was created to solve this depletion issue and has an address space of 128bits which in theory is 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses

or 340 undecillion addresses.

VPCs were not IPv6 until Dec2016 but they are now IPv6. EC2 services are on IPv6
The problem is ISPs are still on IPv4.

Top Level Domains

If we look at common domain names such as google.com, bbc.co.uk. acloud.guru etc you will notice a string of characters separated by dots (periods). The last word in a domain name represents the "top level domain". The second word in a domain name is known as a second level domain name (this is optional though and depends on the domain name).

.com
.edu
.gov
.co.uk
.gov.uk
.com.au

These top level domain names are controlled by the Internet Assigned Numbers Authority (IANA) in a root zone database which is essentially a database of all available top level domains. You can view this database by visiting -

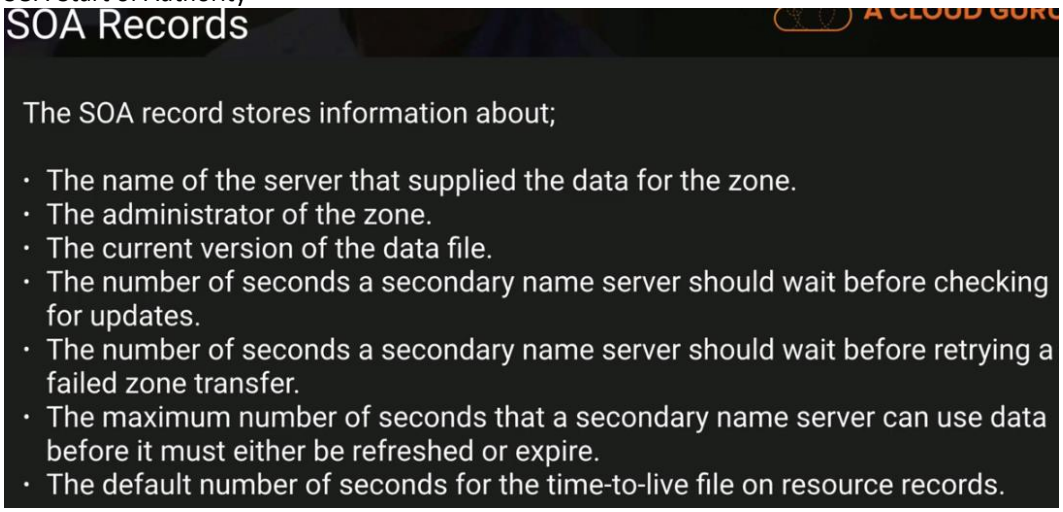
http://www.iana.org/domains/root/db

Domain Registrars

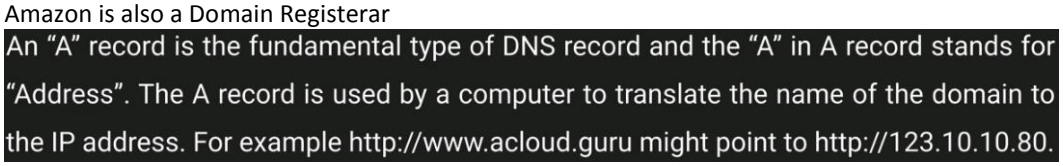
Because all of the names in a given domain name have to be unique there needs to be a way to organize this all so that domain names aren't duplicated. This is where domain registrars come in. A registrar is an authority that can assign domain names directly under one or more top-level domains. These domains are registered with InterNIC, a service of ICANN, which enforces uniqueness of domain names across the Internet. Each domain name becomes registered in a central database known as the WhoIS database.

Popular domain registrars include GoDaddy.com, 123-reg.co.uk etc.

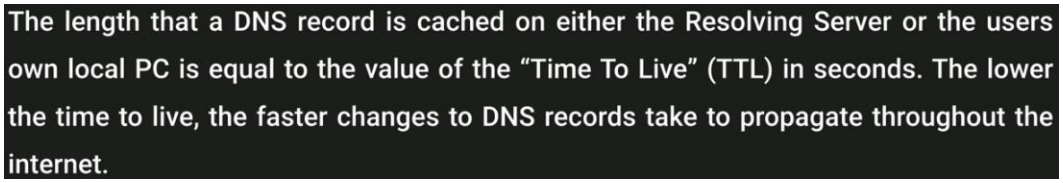
SOA-Start of Authority



Amazon is also a Domain Registerar

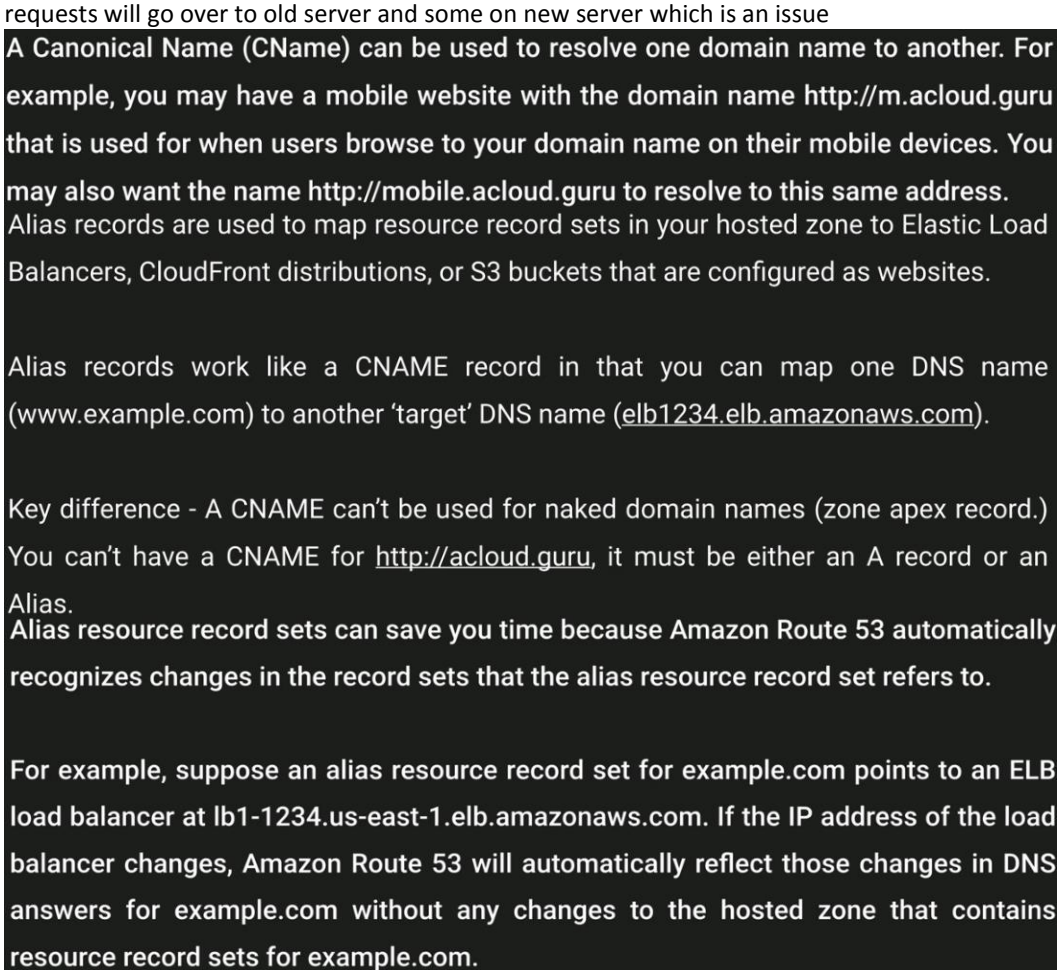


Elastic Load Balancer will never have IPv4 or IPv6 address, it will have a DNS name. That means we cant use 'A' record to resolve to Elastic Load Balancer or an alias record.



When we switch DNS names from one registrar to another, it takes couple of hours which TTL and generally its 2days

Before actually doing DNS migration, we must drop TTL to 300ms or 5mins and then wait for 2 days. If we dont do that, some DNS requests will go over to old server and some on new server which is an issue



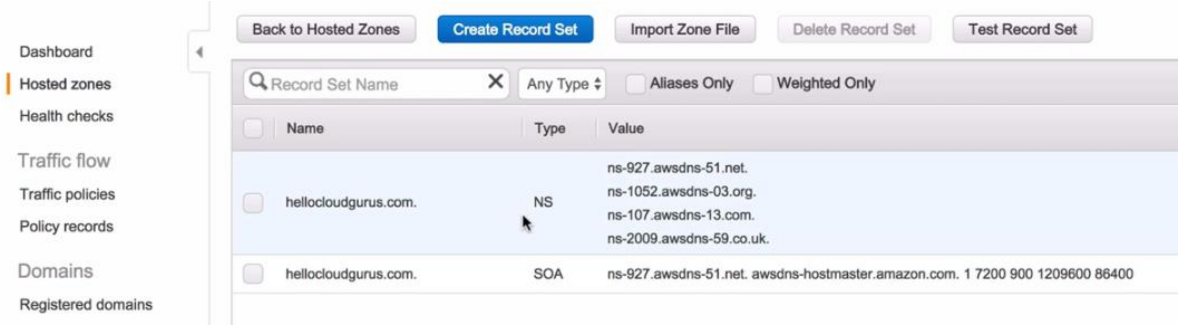
Choosing alias record over cname is that alias name hit on AWS is free but Cname service is charged. Alias record does allow to map naked record names back to elastic load balancers e.g. acloud.guru without www

Route53 – its a global service

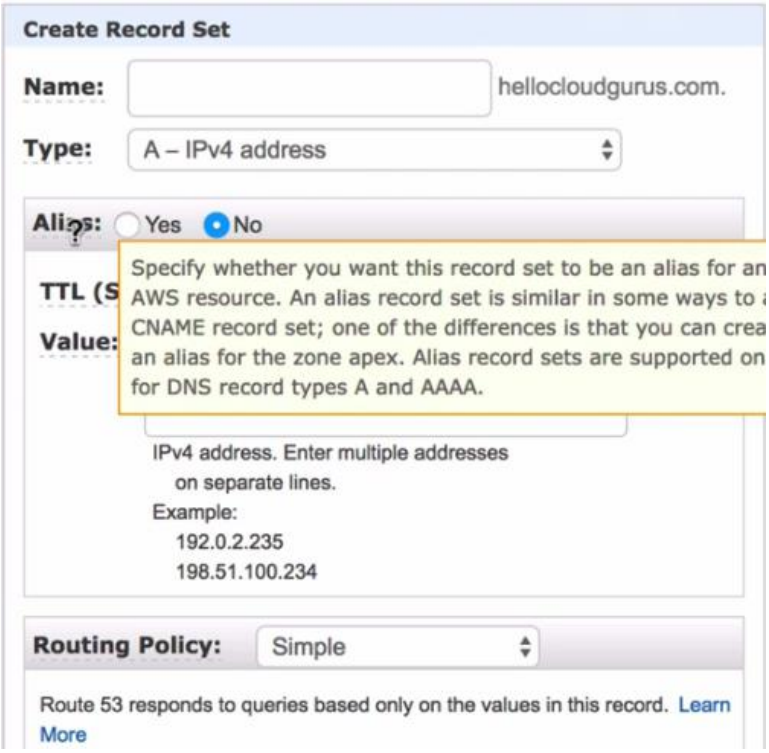
Creating EC2 instances either in same region or in any part of the world, and then making classic Elastic Load balancers to deivide the load and then finally assigning it to the route53 service will make the system as fault tolerant.
Simple routing policy
5 types of routing policies are as follows
-simple
-weighted
-latency
-failover
-geolocation
Simple – there is no intelligence built into it. Its a like round robin routing policy

This is the default routing policy when you create a new record set. This is most commonly used when you have a single resource that performs a given function for your domain, for example, one web server that serves content for the http://acloud.guru website.

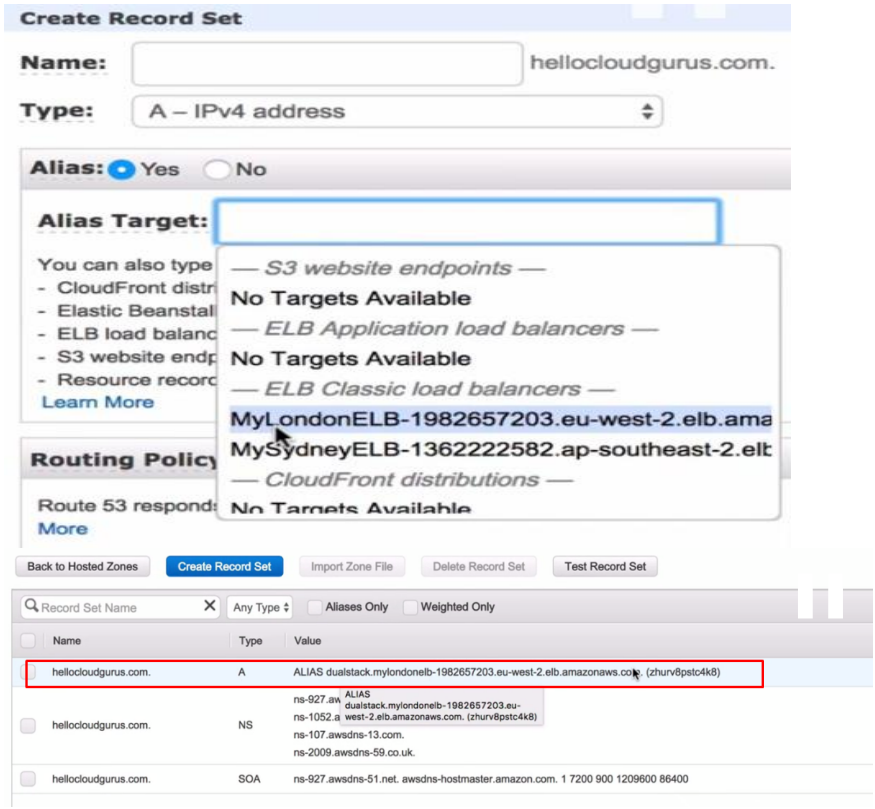
By default below 2 record sets are created



These has to be attached to the EC2 instances
Create a record set – below is an example of a naken domain name

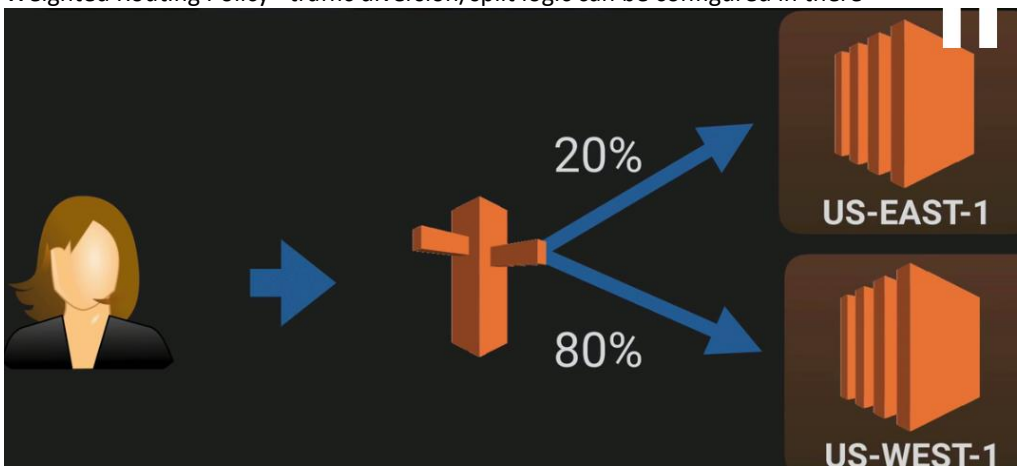


Different ELBs are listed

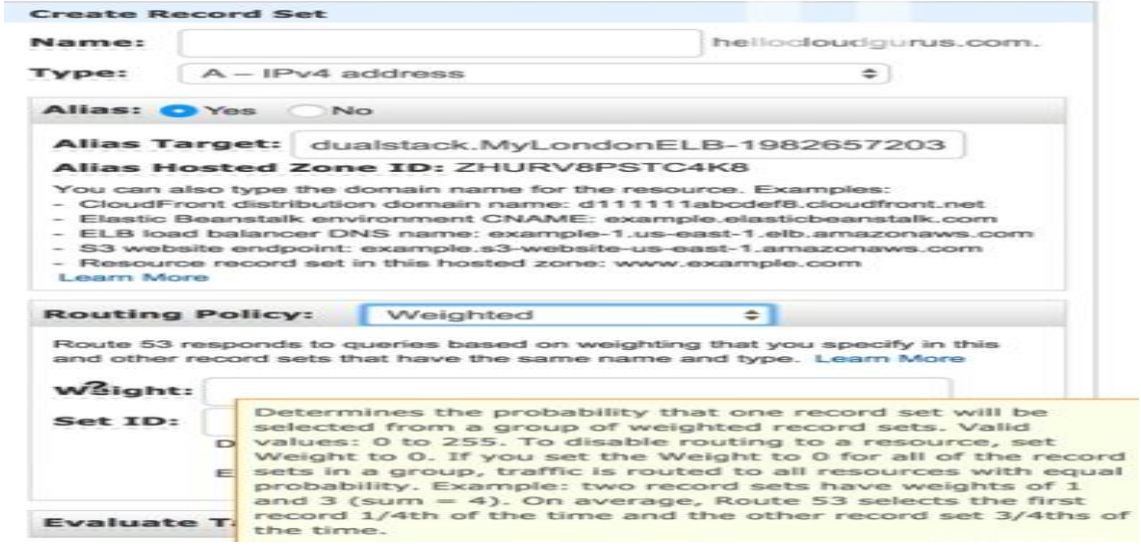


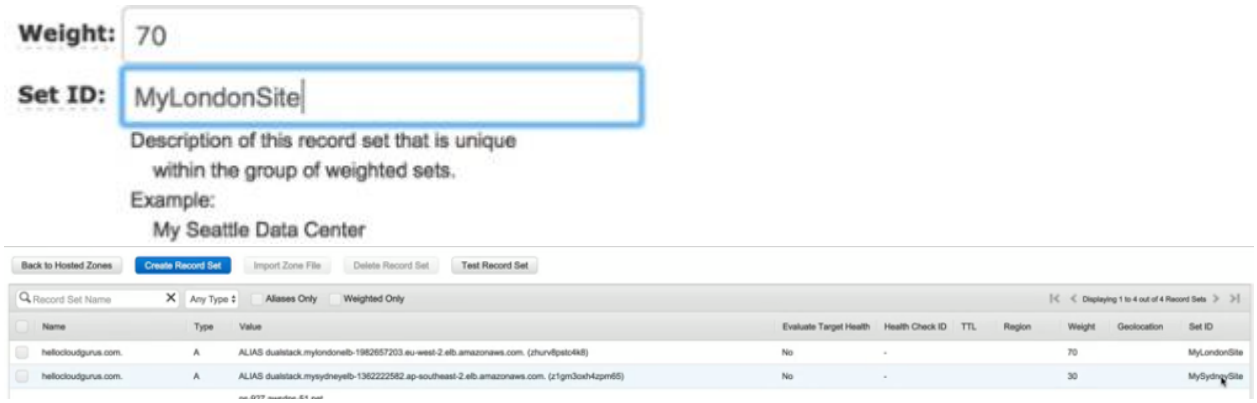
ELBs dont have IPv4 addresses and they can only be accessed by going into their DNS names

Weighted Routing Policy –traffic diversion/split logic can be configured in there

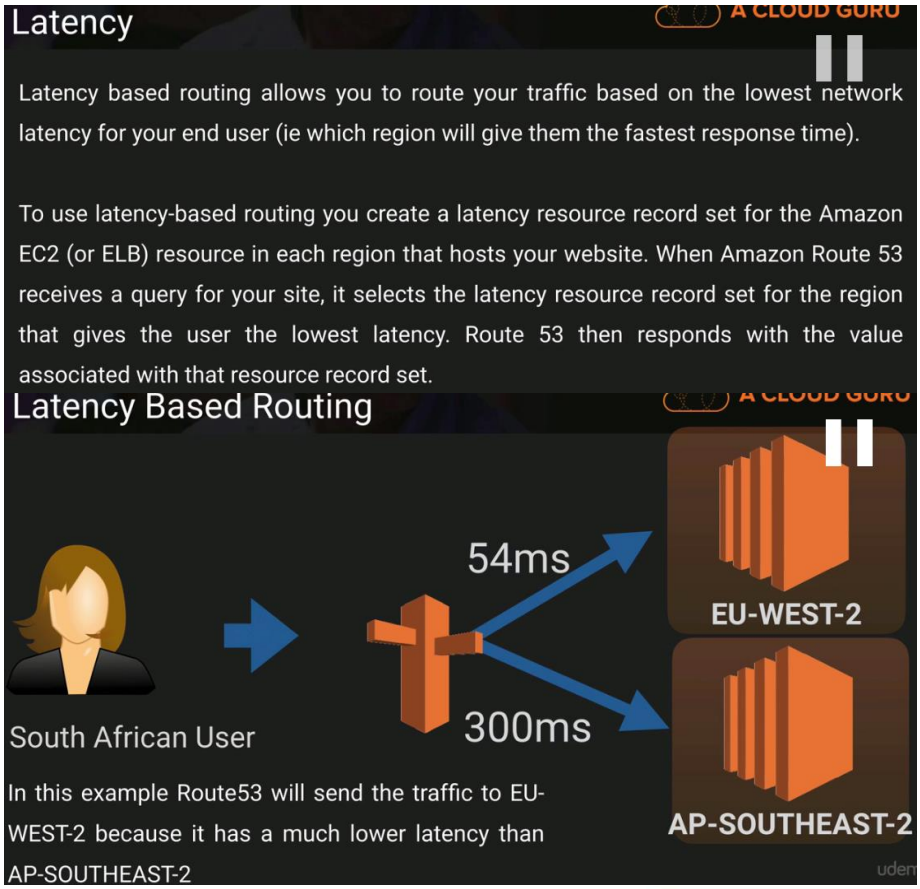


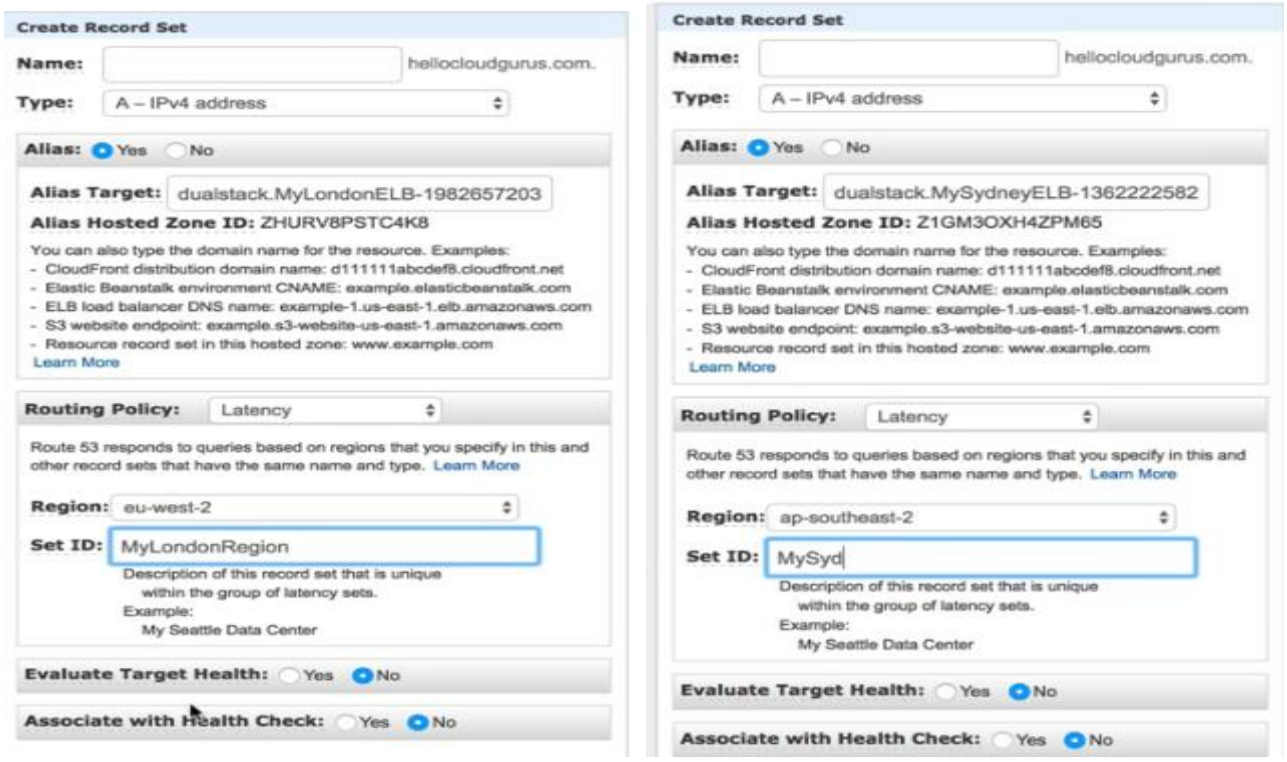
To achiev this 2 load balancers or depending upon scenarios, n load balancers, are needed to split the traffic and while creating routing policies, ELBs can be configured





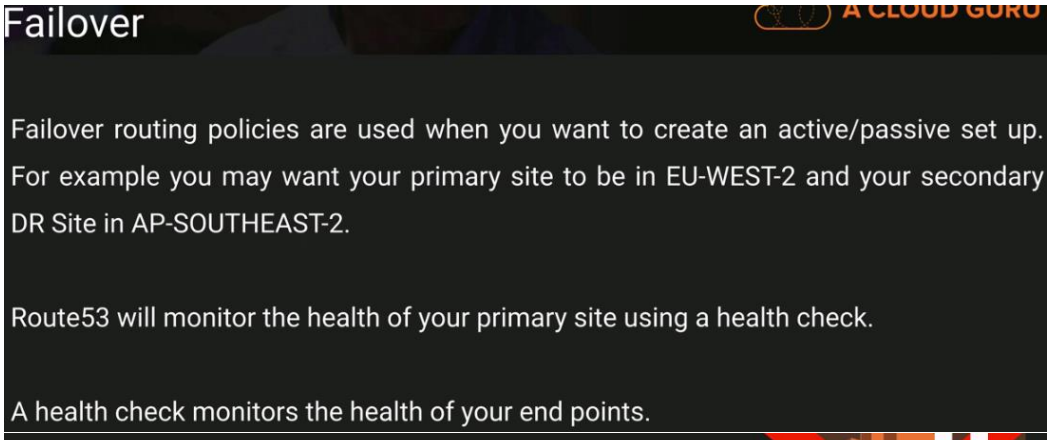
Amazon takes global view of routing policies and depending upon traffic load, it decides loads at run time

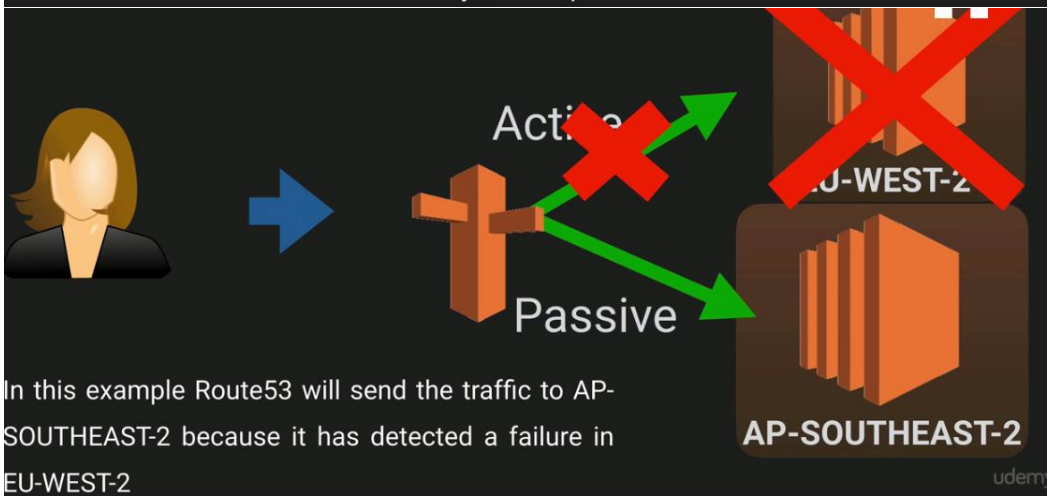




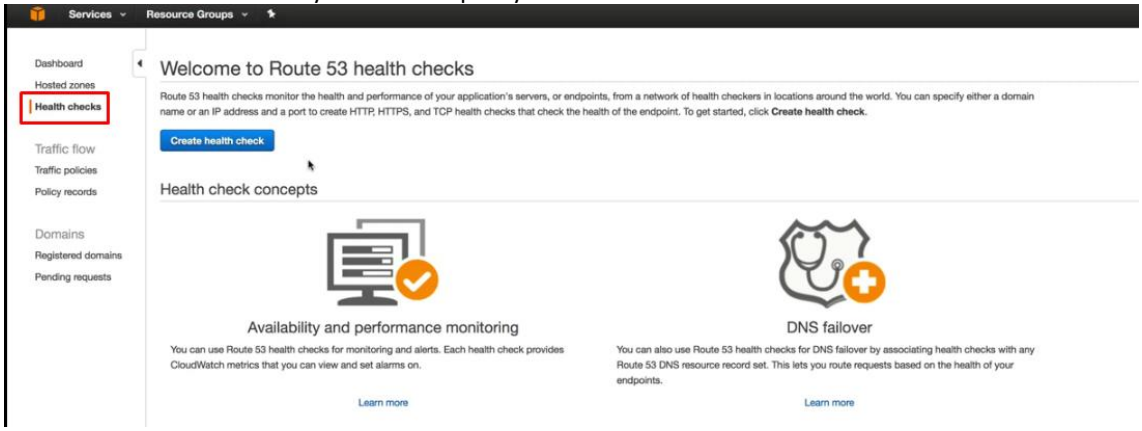
vyprVPN to test as if we are in different locations

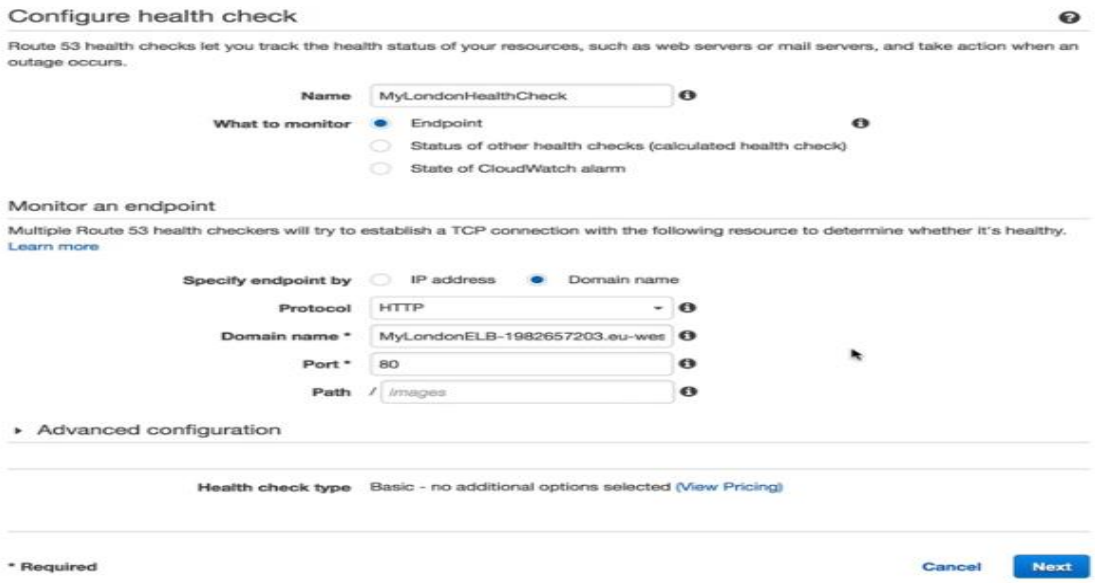
test to mimic as if the end point is in different location so that Route53 can be tested



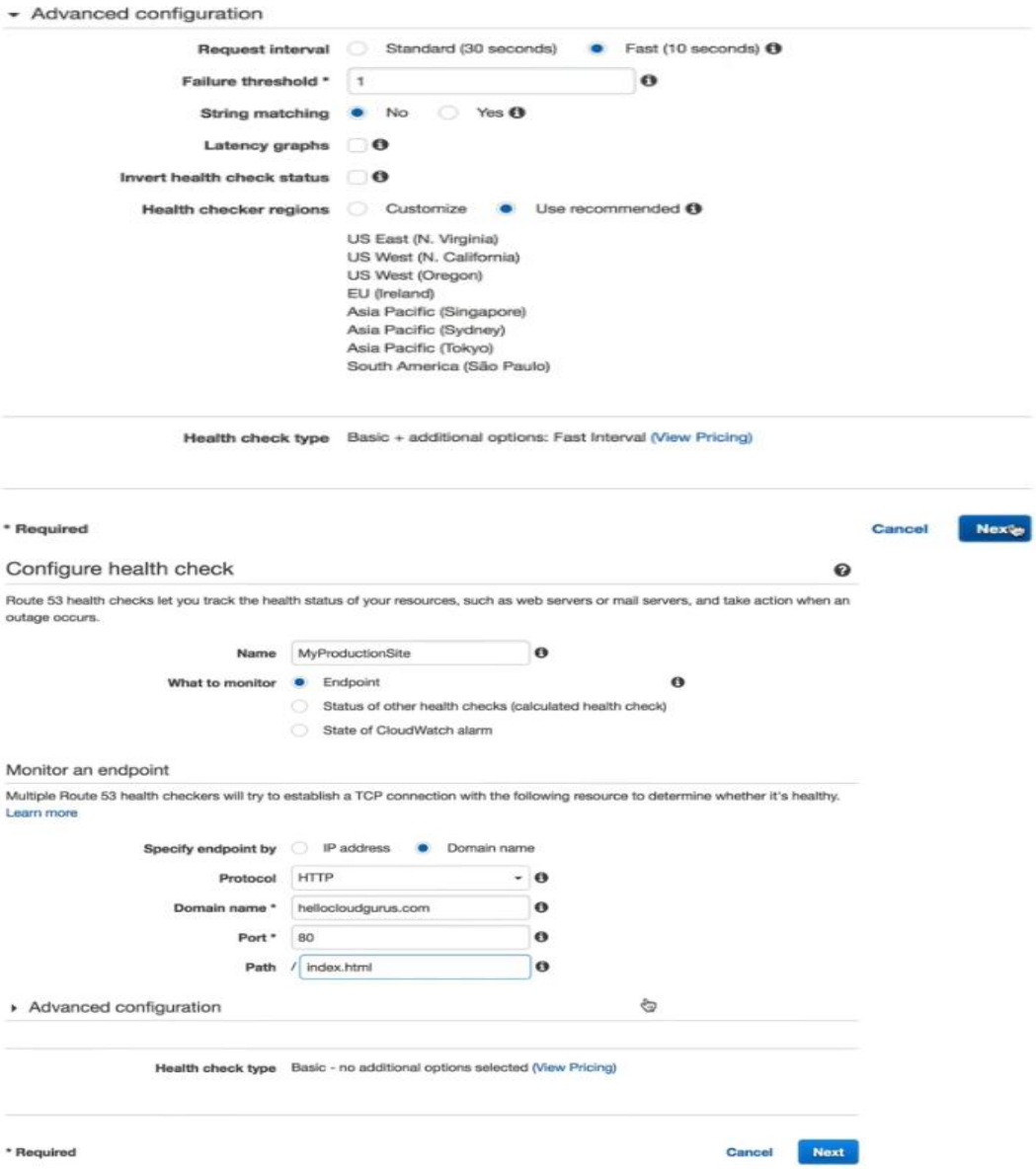


Before creating this, we need to create health check as follow (by using DNS name of elastic load balancers)
As based on healthcheck only the failover policy will work





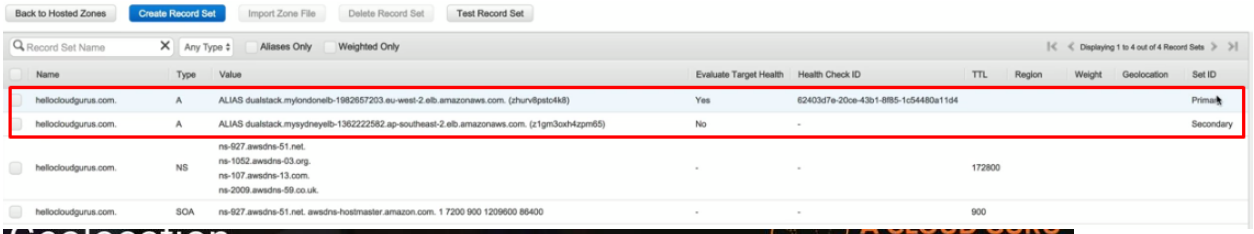
Forcefully making a health check fail, set advance configurations

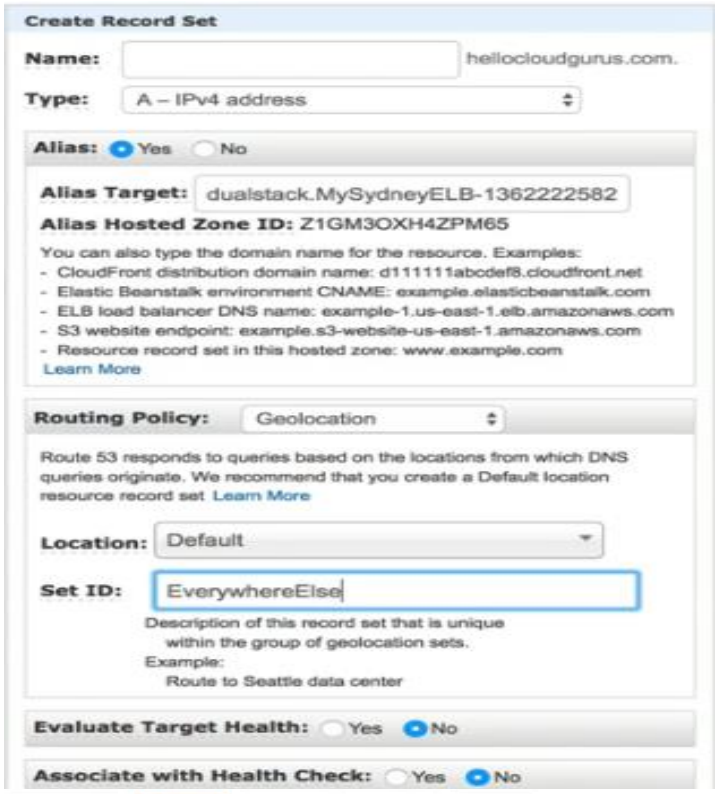


You can create an alarm on the health check failure. SNS topic,sending mail can be easily configured

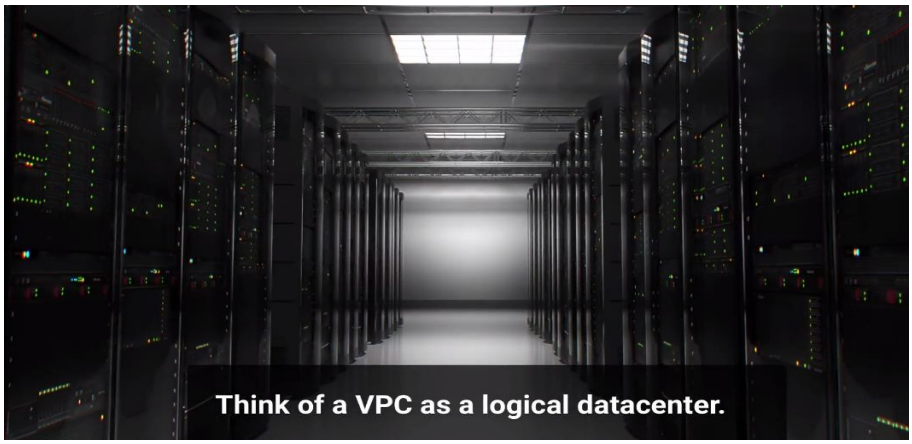
Creating an A record and pointing it to the primary health check which points back to domain name (hellocloudguru.com) is actually an error (circular loop kind of scenario) and hence this A record cant be set.PFB

hence it has to be the DR ELB DNS name must be resolved as an end point





VPC – Virtual Private Cloud

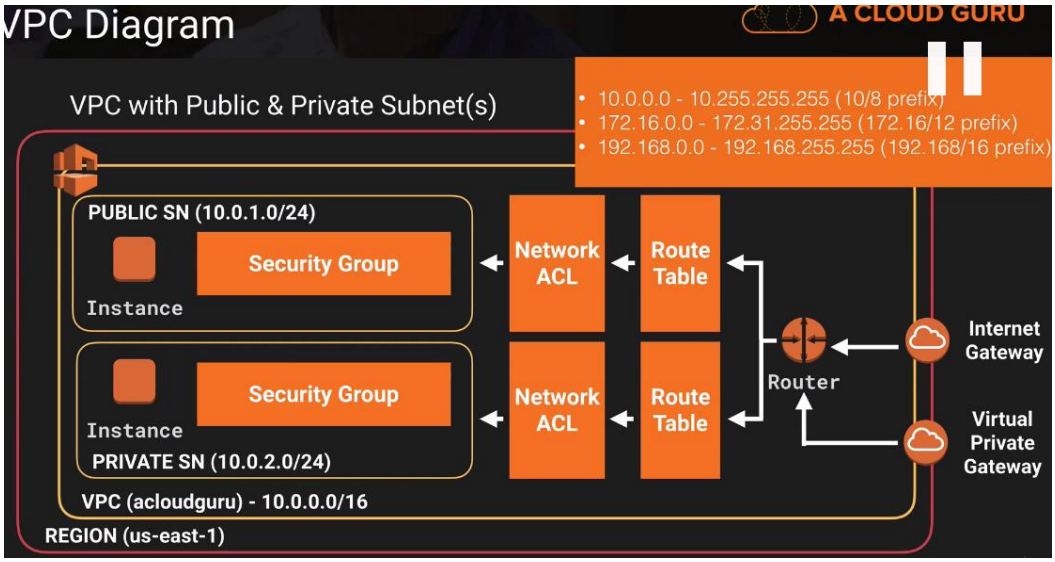


AWS regions – Deployed VPCs (dont span region but can do span availability zones)

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the Amazon Web Services (AWS) Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

You can easily customize the network configuration for your Amazon Virtual Private Cloud. For example, you can create a public-facing subnet for your webservers that has access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access. You can leverage multiple layers of security, including security groups and network access control lists, to help control access to Amazon EC2 instances in each subnet.

Additionally, you can create a Hardware Virtual Private Network (VPN) connection between your corporate datacenter and your VPC and leverage the AWS cloud as an extension of your corporate datacenter.



1 subnet or network address range equals 1 availability zone

Security groups can span multiple subnets and multiple availability zones, same with network ACLs, same with route tables

Security groups are statefull and network access control list are stateless

What can you do with a VPC?

- Launch instances into a subnet of your choosing
- Assign custom IP address ranges in each subnet
- Configure route tables between subnets
- Create internet gateway and attach it to our VPC
- Much better security control over your AWS resources
- Instance security groups
- Subnet network access control lists (ACLS)

Default VPC vs Custom VPC

A CLOUD GUR

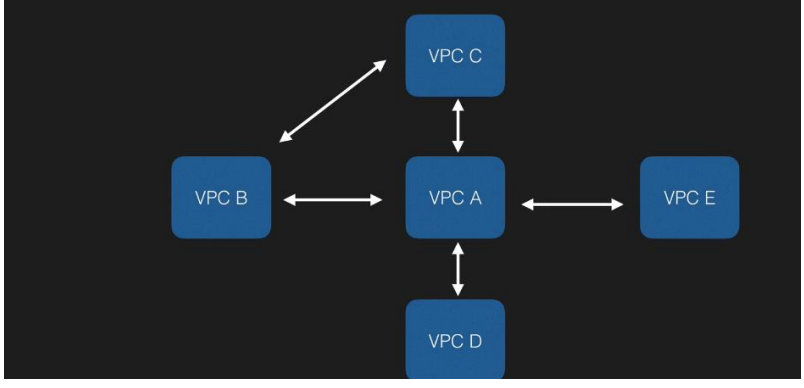
- Default VPC is user friendly, allowing you to immediately deploy instances
- All Subnets in default VPC have a route out to the internet
- Each EC2 instance has both a public and private IP address
- If you delete the default VPC the only way to get it back is to contact AWS.

VPC Peering

A CLOUD GUR

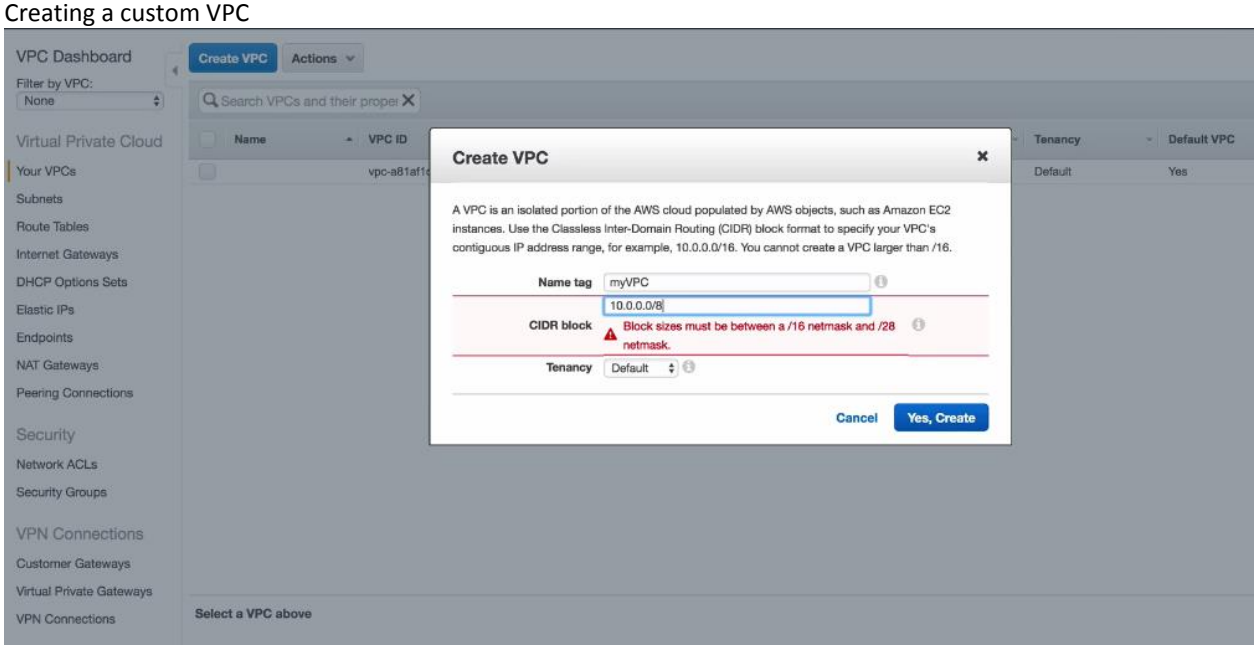
- Allows you to connect one VPC with another via a direct network route using private IP addresses.
- Instances behave as if they were on the same private network
- You can peer VPC's with other AWS accounts as well as with other VPCs in the same account.
- Peering is in a star configuration, ie 1 central VPC peers with 4 others. NO TRANSITIVE PEERING!!!

VPC Transitive Peering



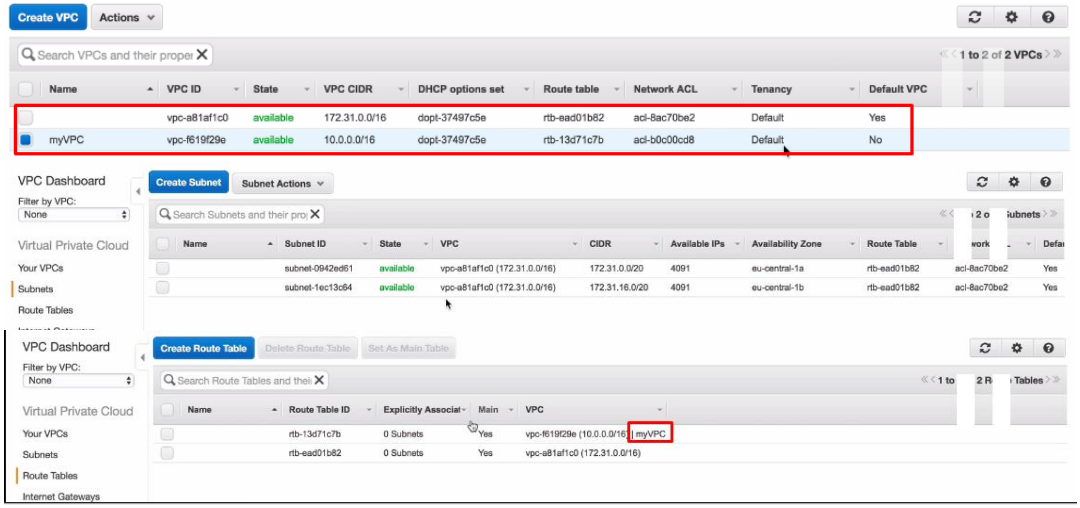
Transitive peering means setting up individual peers to talk to each other, they cannot go via the other VPC peer. Hence star peering become important, VPCs can talk to each other via the other

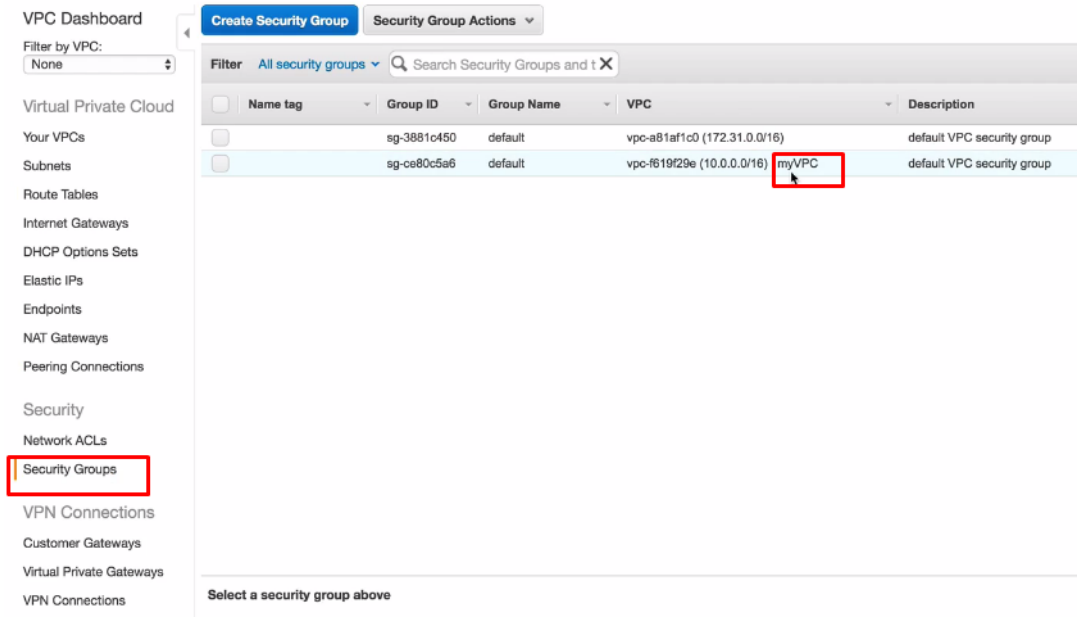
Creating a custom VPC

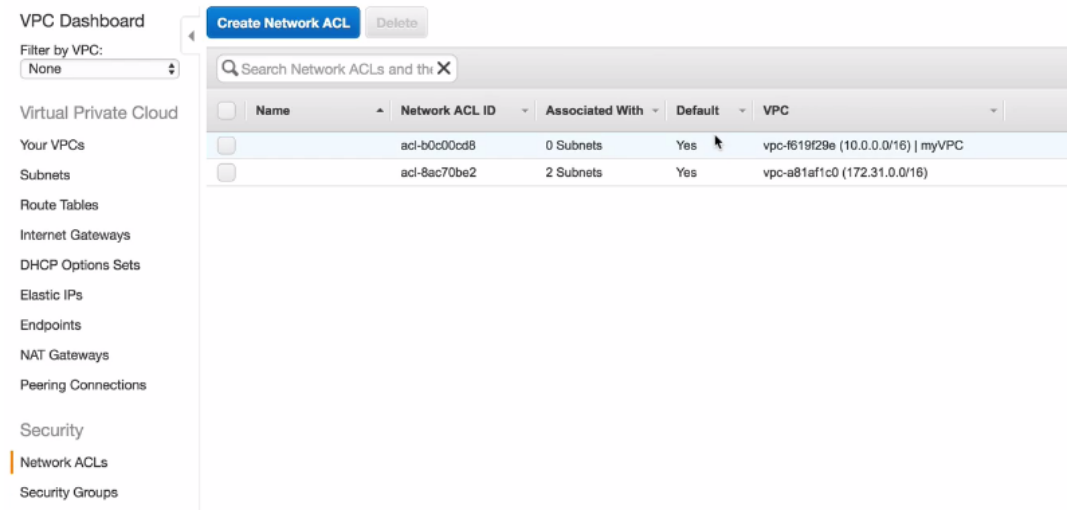


The tenancy can be dedicate which means it will be private cloud and is quite expensive. The tenance is default and is public

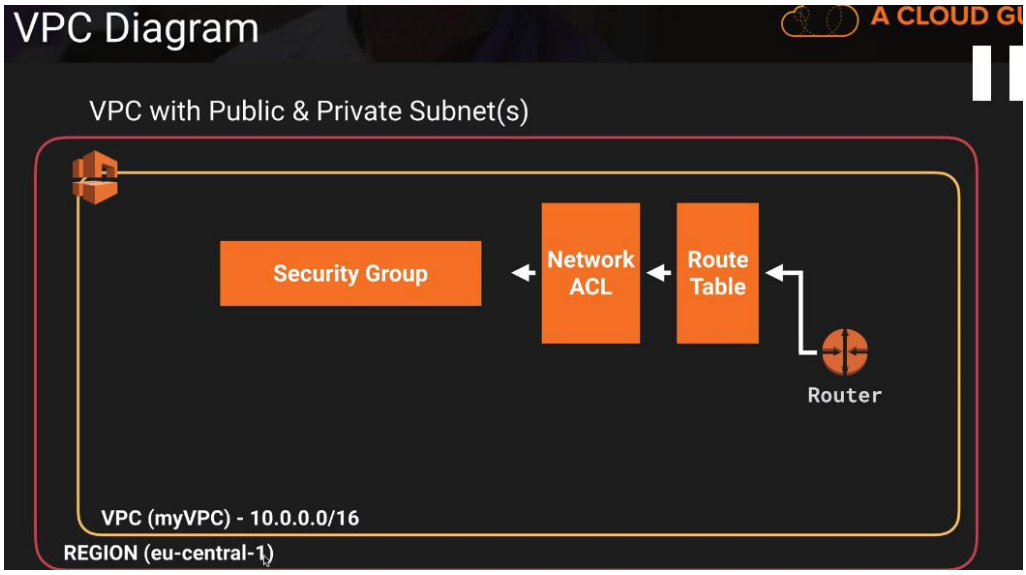
VPC create option creates some default entries as shown in below screens





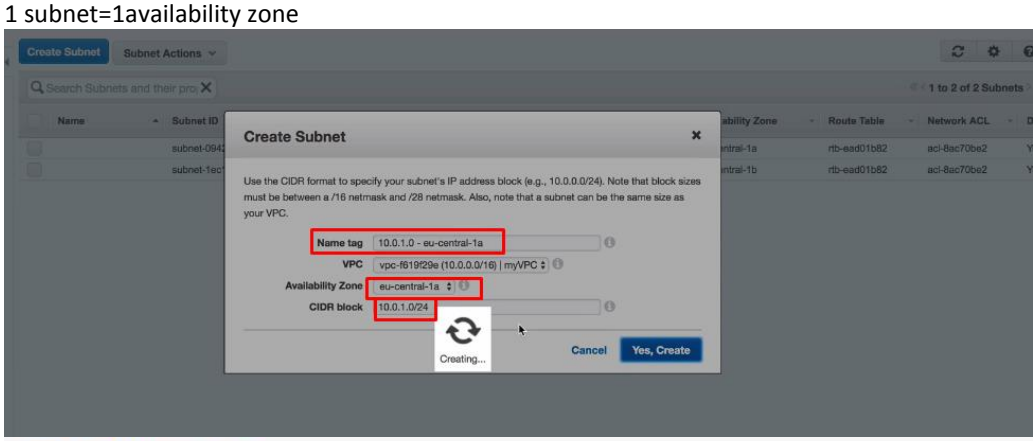


Default VPC create wizard doesn't createWithout subnet masks, we cant deploy anything into VPC without a subnet



Creating a subnet

1 subnet=1availability zone



VPC and Subnet Sizing

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

When you create a VPC, we recommend that you specify a CIDR block from the private (non-publicly routable) IP address ranges as specified in RFC 1918:

- `10.0.0.0 - 10.255.255.255` (10/8 prefix)
- `172.16.0.0 - 172.31.255.255` (172.16/12 prefix)
- `192.168.0.0 - 192.168.255.255` (192.168/16 prefix)

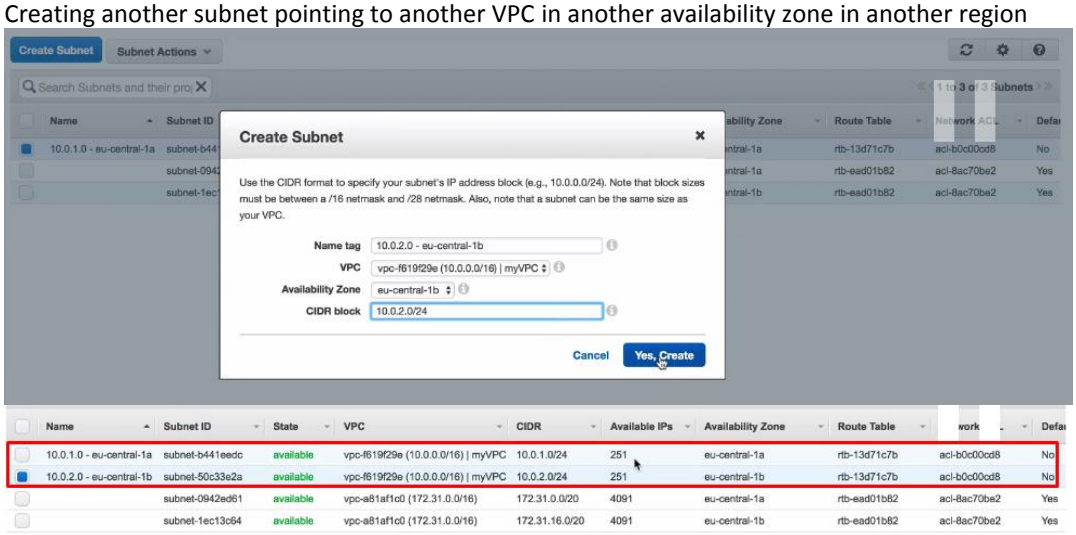
The CIDR block of a subnet can be the same as the CIDR block for the VPC (for a single subnet in the VPC), or a subset (for multiple subnets). The allowed block size is between a `/28` netmask and `/16` netmask. If you create more than one subnet in a VPC, the CIDR blocks of the subnets cannot overlap.

For example, if you create a VPC with CIDR block `10.0.0.0/24`, it supports 256 IP addresses. You can break this CIDR block into two subnets, each supporting 128 IP addresses. One subnet uses CIDR block `10.0.0.0/25` (for addresses `10.0.0.0 - 10.0.0.127`) and the other uses CIDR block `10.0.0.128/25` (for addresses `10.0.0.128 - 10.0.0.255`).

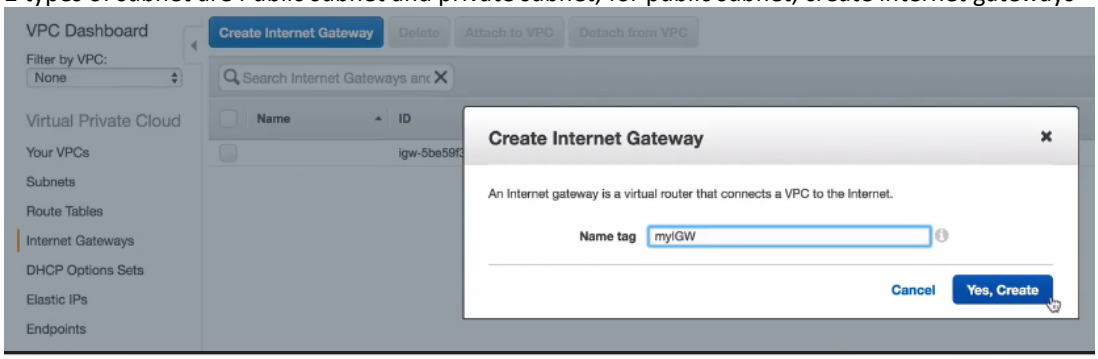
The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance. For example, in a subnet with CIDR block `10.0.0.0/24`, the following five IP addresses are reserved:

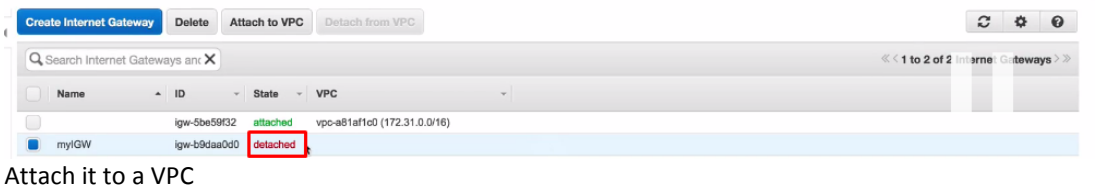
- `10.0.0.0`: Network address.
- `10.0.0.1`: Reserved by AWS for the VPC router.
- `10.0.0.2`: Reserved by AWS. The IP address of the DNS server is always the base of the VPC network range plus two; however, we also reserve the base of each subnet range plus two. For more information, see Amazon DNS Server.
- `10.0.0.3`: Reserved by AWS for future use.
- `10.0.0.255`: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

Creating another subnet pointing to another VPC in another availability zone in another region



2 types of subnet are Public subnet and private subnet, for public subnet, create internet gateways

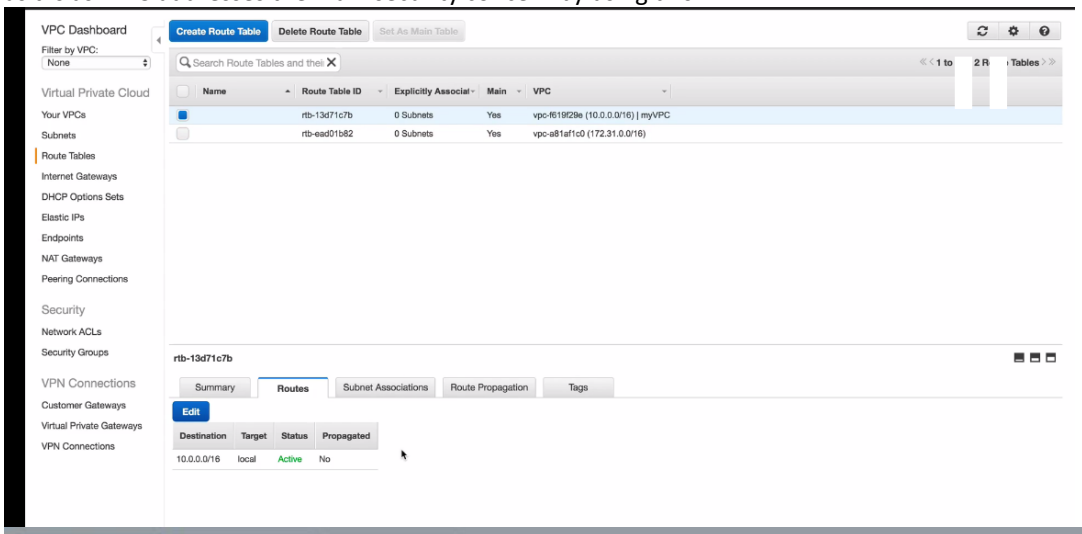


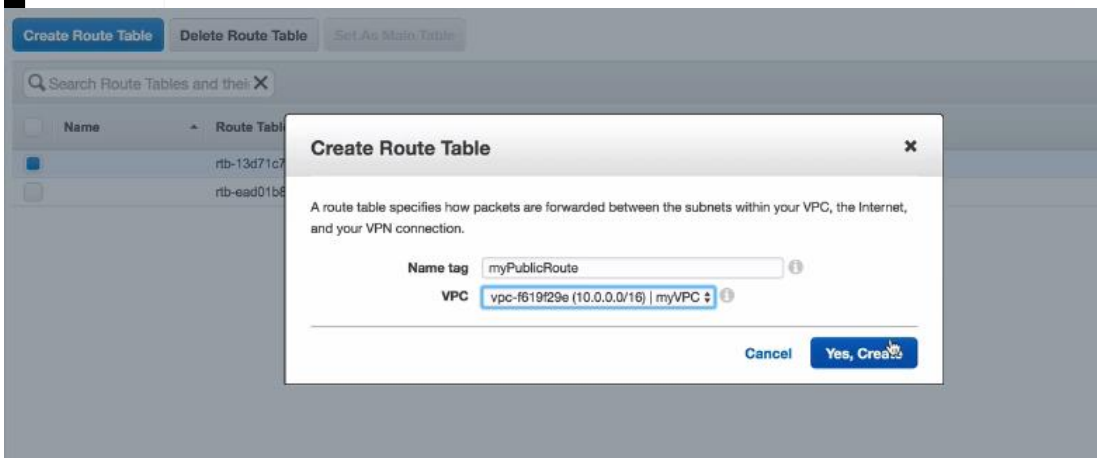


Attach it to a VPC

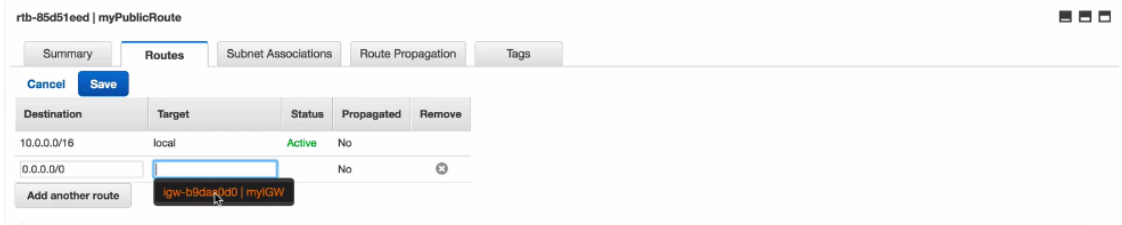
We cannot attach multiple internet gateways to a VPC, its 1-1 relationship. Internet gateways are engineered to be highly resilient anyways.

Creating routes so that VPCs can talk to each other (multiple VPCs into same subnet)
The default one created automatically. The best practice is to leave this as is, this is the main route table and is private and let it be as t is as AWS addresses the main security concern by using this



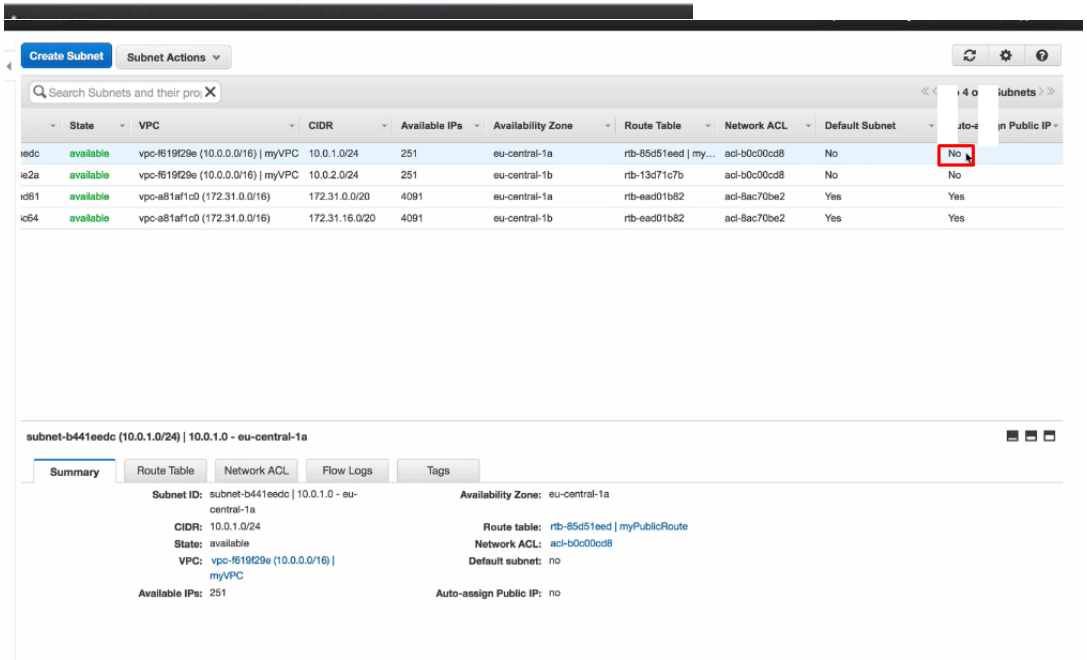


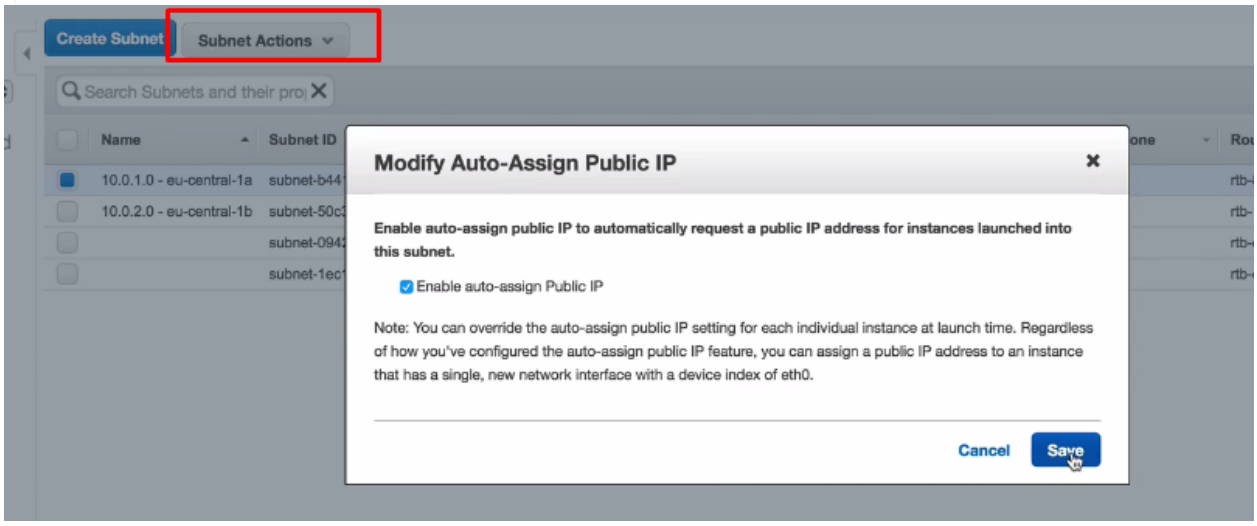
Attach a subnet to this route
We have created a new route table and is not our main route table and we have given it route to the internet via internet gateway configuration



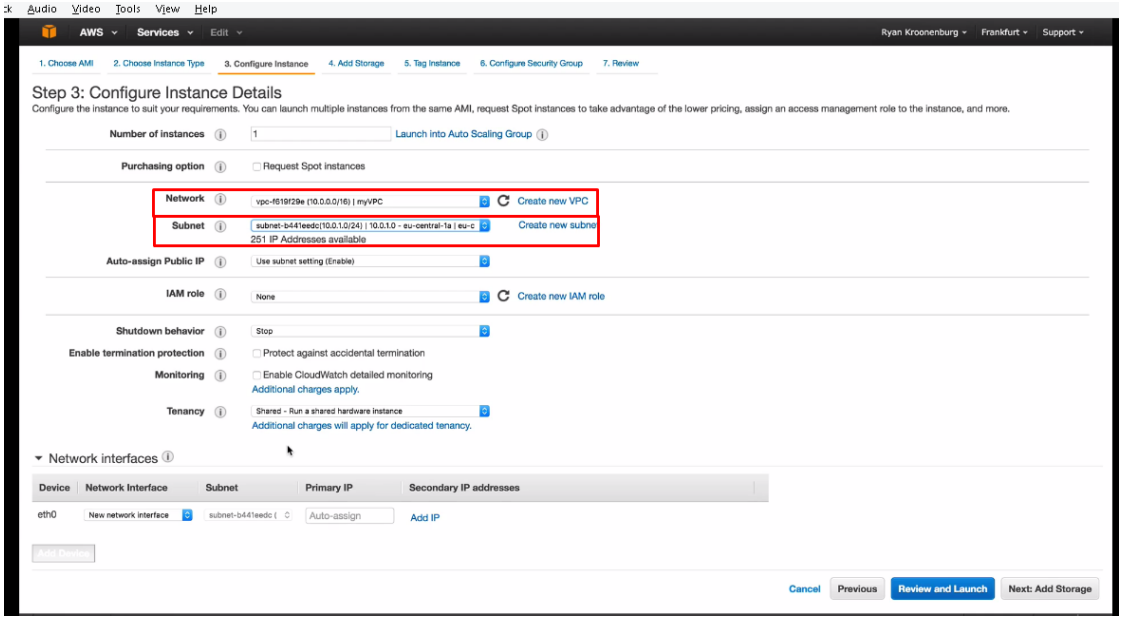

rtb-85d51eed | myPublicRoute
Summary
Routes
Subnet Associations
Route Propagation
Tags
Cancel
Save
Associate
Subnet
CIDR
Current Route Table
subnet-b441eedc (10.0.1.0/24) | 10.0.1.0 - eu-central-1a
10.0.1.0/24
Main
subnet-50c33e2a (10.0.2.0/24) | 10.0.2.0 - eu-central-1b
10.0.2.0/24
Main


Edit
Subnet
CIDR
subnet-b441eedc (10.0.1.0/24) | 10.0.1.0 - eu-central-1a
10.0.1.0/24
The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:
Subnet
CIDR
subnet-50c33e2a (10.0.2.0/24) | 10.0.2.0 - eu-central-1b
10.0.2.0/24


Create Subnet
Subnet Actions
Search Subnets and their pro
State
VPC
CIDR
Available IPs
Availability Zone
Route Table
Network ACL
Default Subnet
Public IP
available
vpc-f619f29e (10.0.0.0/16) | myVPC
10.0.1.0/24
251
eu-central-1a
rtb-85d51eed | my...
acl-b0c00cd8
No
available
vpc-f619f29e (10.0.0.0/16) | myVPC
10.0.2.0/24
251
eu-central-1b
rtb-13d71c7b
acl-b0c00cd8
No
available
vpc-a81af1c0 (172.31.0.0/16)
172.31.0.0/20
4091
eu-central-1a
rtb-ead01b82
acl-8ac70be2
Yes
available
vpc-a81af1c0 (172.31.0.0/16)
172.31.16.0/20
4091
eu-central-1b
rtb-ead01b82
acl-8ac70be2
Yes
subnet-b441eedc (10.0.1.0/24) | 10.0.1.0 - eu-central-1a
Summary
Route Table
Network ACL
Flow Logs
Tags
Subnet ID: subnet-b441eedc | 10.0.1.0 - eu-central-1a
CIDR: 10.0.1.0/24
State: available
VPC: vpc-f619f29e (10.0.0.0/16) | myVPC
Available IPs: 251
Availability Zone: eu-central-1a
Route table: rtb-85d51eed | myPublicRoute
Network ACL: acl-b0c00cd8
Default subnet: no
Auto-assign Public IP: no

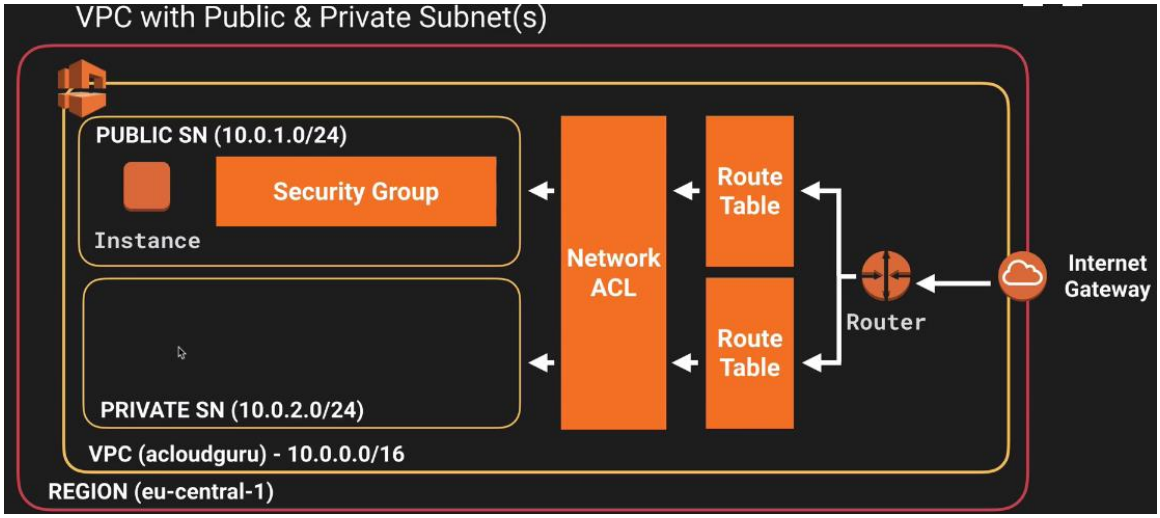

Create Subnet
Subnet Actions
Search Subnets and their pro
Name
Subnet ID
10.0.1.0 - eu-central-1a
10.0.2.0 - eu-central-1b
Modify Auto-Assign Public IP
Enable auto-assign public IP to automatically request a public IP address for instances launched into this subnet.
Enable auto-assign Public IP
Note: You can override the auto-assign public IP setting for each individual instance at launch time. Regardless of how you've configured the auto-assign public IP feature, you can assign a public IP address to an instance that has a single, new network interface with a device index of eth0.
Cancel
Save

Any EC2 instance further being created into this subnet or VPC will be automatically asssigned an IP address

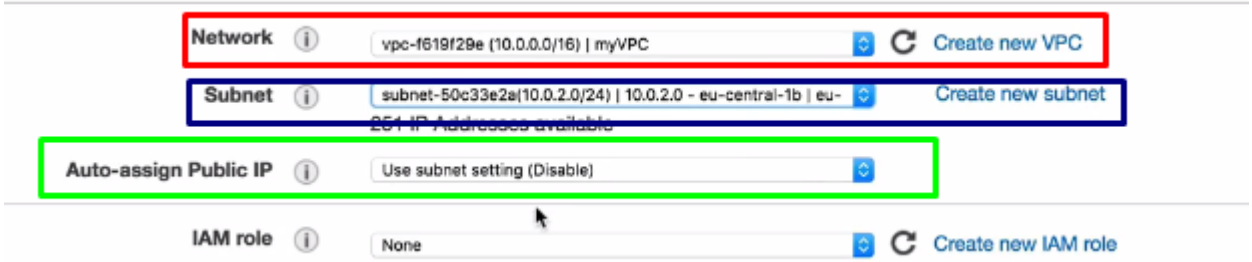


Auto assigned IP setting can be marked enabled here as well while creating an EC2 instance

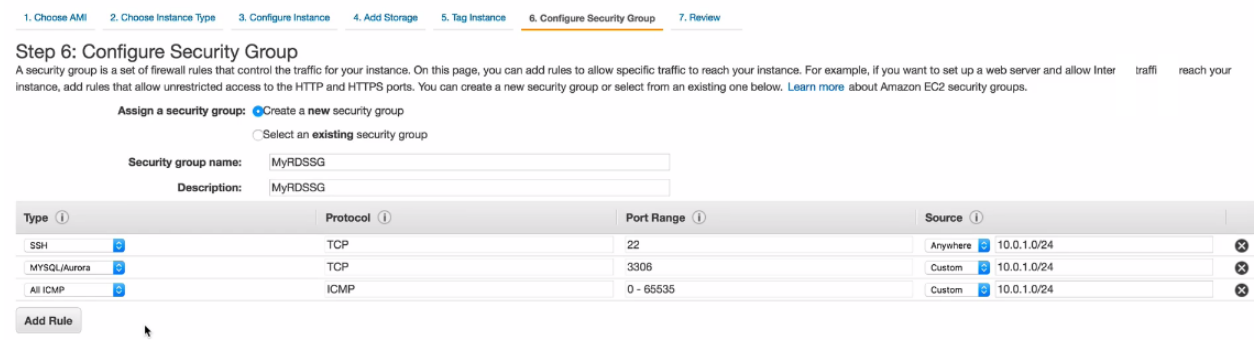
What we did so far is as follows



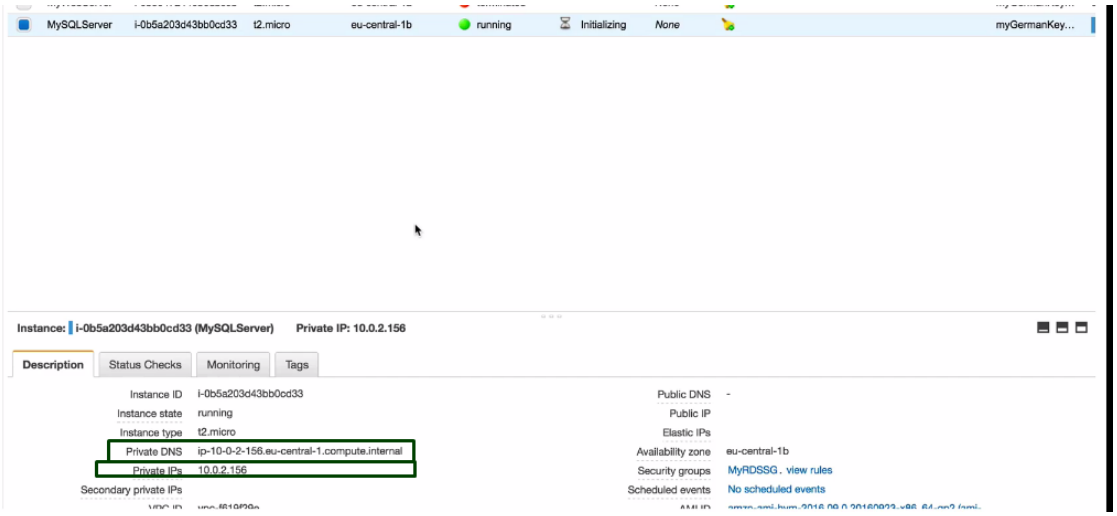
Creating EC2 instance in private cloud



Creating a private VPC and having potential services like DB services lying there and assigning a CIDR network range. e.g. any IP within that range will be able to access to the DB service we have in private VPC EC2 instance



ICMP rule is for public ping. Security groups are statefull i.e. creating these inbound rules, outbound rules are automatically created



Enter into public EC2 instance

```
Ryans-iMac:SSH ryankroonenburg$ ls
myGermanKeyPair.pem
Ryans-iMac:SSH ryankroonenburg$ chmod 0600 myGermanKeyPair.pem
Ryans-iMac:SSH ryankroonenburg$ nano myGermanKeyPair.pem
Ryans-iMac:SSH ryankroonenburg$ ssh ec2-user@54.93.120.28 -i myGermanKeyPair.pem
The authenticity of host '54.93.120.28 (54.93.120.28)' can't be established.
ECDSA key fingerprint is SHA256:0Xuoya8kDJuBatmzprc7WOR1yVVmV5DNxQFk35vi6xU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '54.93.120.28' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
[ec2-user@ip-10-0-1-13 ~]$ sudo su
[root@ip-10-0-1-13 ec2-user]# cl
```

If you have made ping from public subnet to private subnet EC2 instance, since the connection is established due to ICMP security setting, ping will contunue to run as connection established. To kill it, ctrl+c

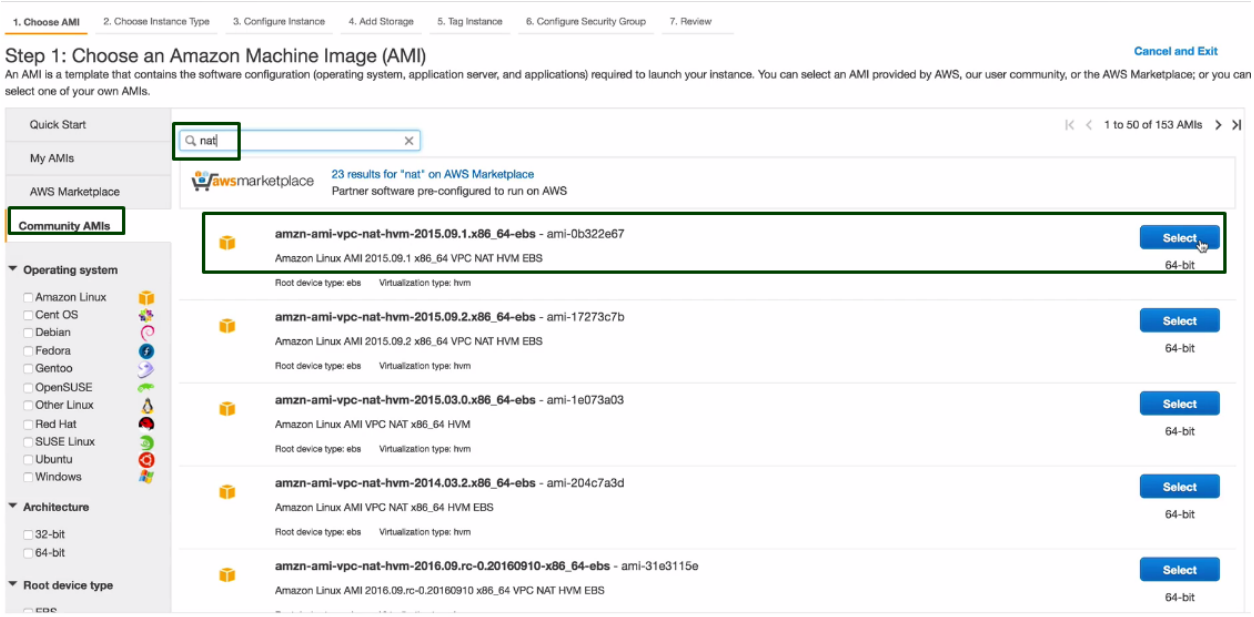
Entering into EC2 instance which is in private subnet using public key pair

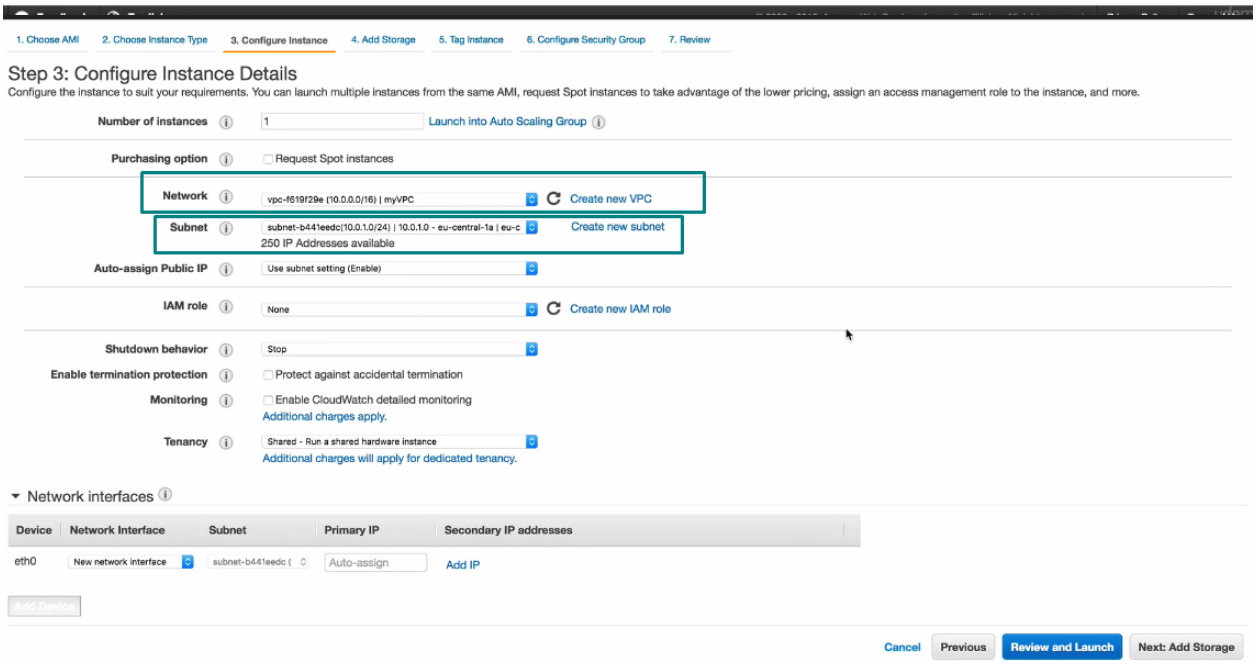
```
[root@ip-10-0-1-13 ec2-user]# nano myGermanKeyPair.pem
[root@ip-10-0-1-13 ec2-user]# chmod 0600 myGermanKeyPair.pem
[root@ip-10-0-1-13 ec2-user]# ssh ec2-user@10.0.2.156 -i myGermanKeyPair.pem
The authenticity of host '10.0.2.156 (10.0.2.156)' can't be established.
ECDSA key fingerprint is f2:a3:f1:62:49:2a:ae:8d:c0:bc:40:a7:b0:a6:f1:97.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.156' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
[ec2-user@ip-10-0-2-156 ~]$
```

This private instance by default can only talk to ports that were assigned by creating security groups which is 80,22,and ICMP and cant do anything else as they are private. They can only talk to other EC2 instance within those IP and port ranges. To resolved this issue, we have to create NAT (Network Address Translations) gateways

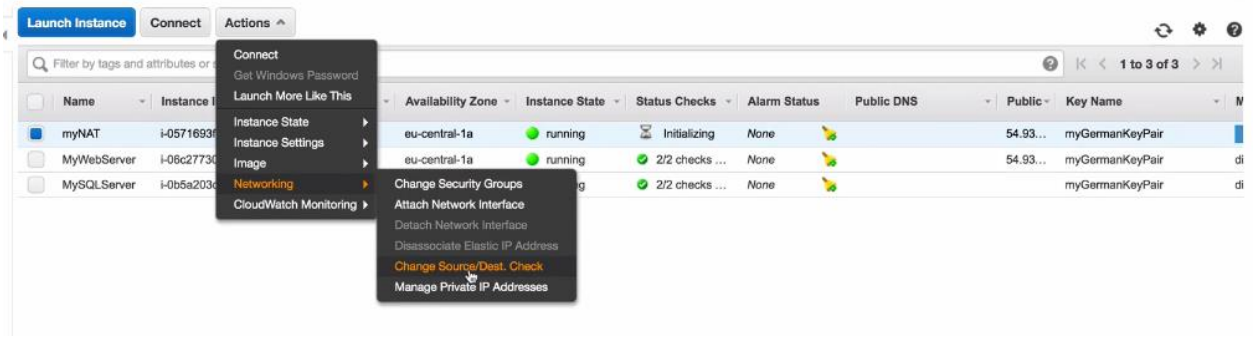
Creating a NAT and assigning that into EC2 instance in private subnet and do stuff need to do in that instance like install DB patches, server patches and SQL instances

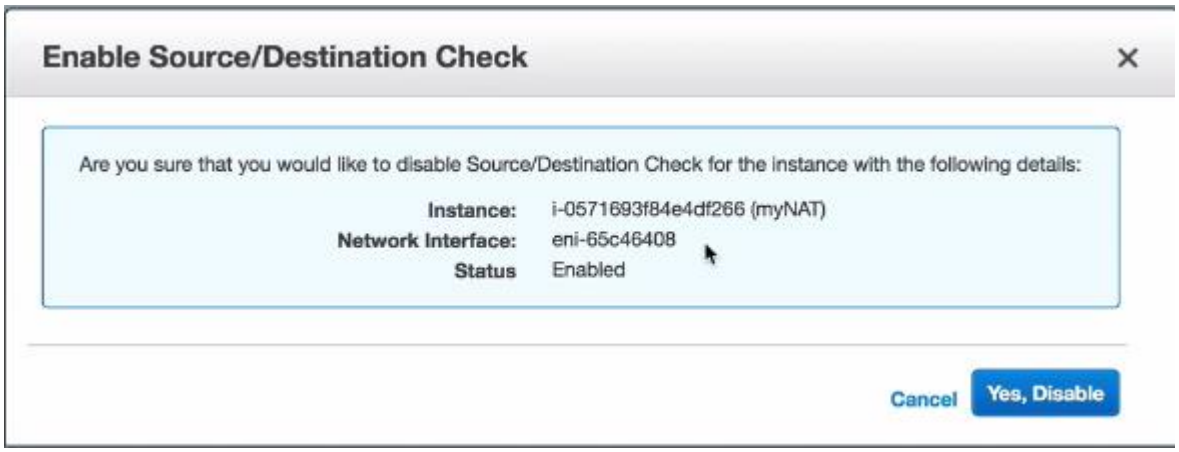
NAT instance is nothing but an EC2 instance that acts as the gateway to the internet and we do all our activities via this NAT instance by installing and configuring NAT gateways over NAT instance



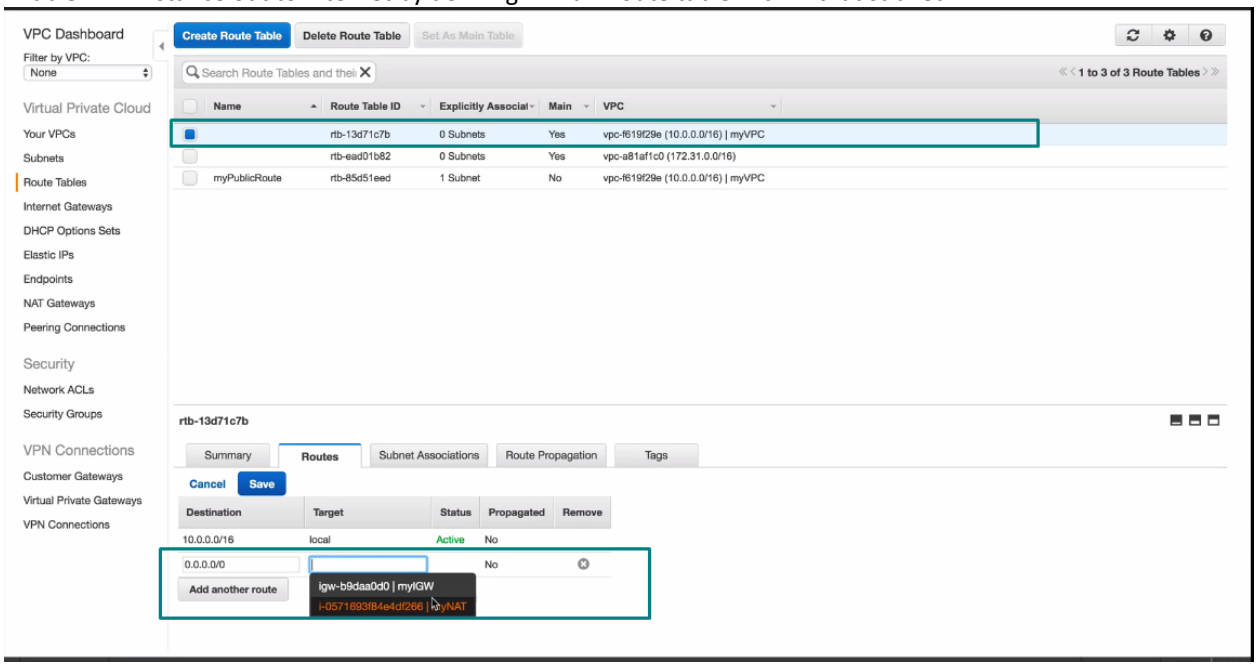


For NAT gateways (scaled by AWS) and no security group is needed, AWS handles BUT for NAT instances they have to be behind security groups





Disable it so that NAT works as desired. I want to make traffic out to the internet and response coming out of that traffic into NAT.
AWS always recommends to have http and https defined for NAT instances or any security groups that are behind NAT instances
Enable NAT instance out to internet by defining in main route table with in that subnet

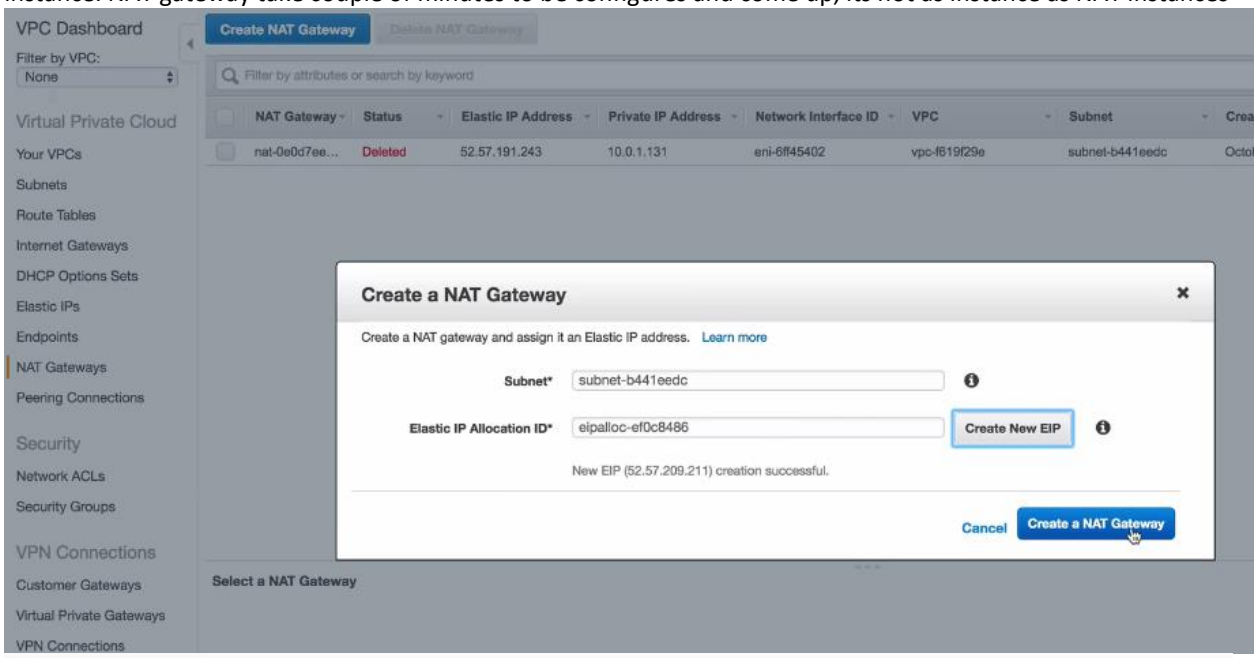


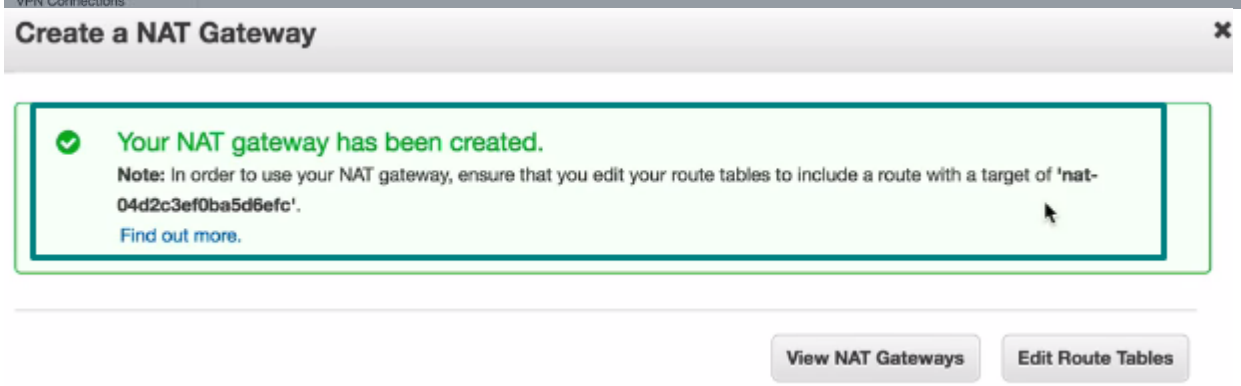
In theory what we have done is that any instances now on into private subnet, should have access to internet via this NAT instance as we have associate main route with NAT instance
Now below command would work

```
[root@ip-10-0-2-156 ec2-user]# yum install mysql -y
Total                                              18 MB/s | 8.4 MB  00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mysql-config-5.5.52-1.13.amzn1.x86_64                       1/4
  Installing : mysql55-libs-5.5.52-1.13.amzn1.x86_64                       2/4
  Installing : mysql55-5.5.52-1.13.amzn1.x86_64                            3/4
  Installing : mysql-5.5-1.6.amzn1.noarch                                  4/4
  Verifying  : mysql-5.5-1.6.amzn1.noarch                                  1/4
  Verifying  : mysql55-libs-5.5.52-1.13.amzn1.x86_64                       2/4
  Verifying  : mysql-config-5.5.52-1.13.amzn1.x86_64                       3/4
  Verifying  : mysql55-5.5.52-1.13.amzn1.x86_64                            4/4

Installed:
  mysql.noarch 0:5.5-1.6.amzn1

Dependency Installed:
  mysql-config.x86_64 0:5.5.52-1.13.amzn1          mysql55.x86_64 0:5.5.52-1.13.amzn1
  mysql55-libs.x86_64 0:5.5.52-1.13.amzn1
```

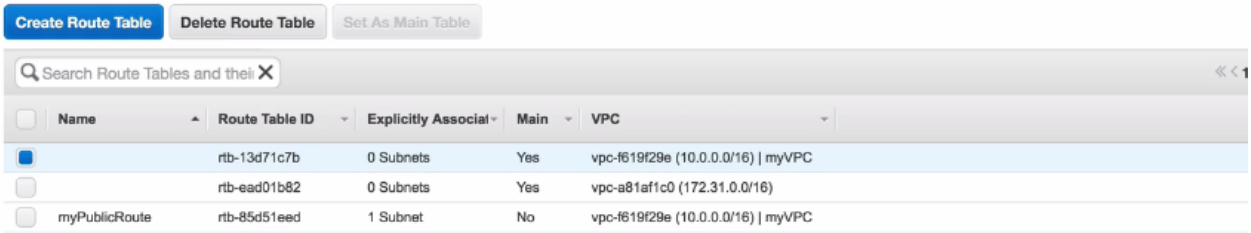
Try the above VIA NAT gateways (always in public subnets)

AWS handles and scales it, you dont have to create security groups or any consideration which you would do in case of NAT instance. NAT gateway take couple of minutes to be configures and come up, its not as instance as NAT instances

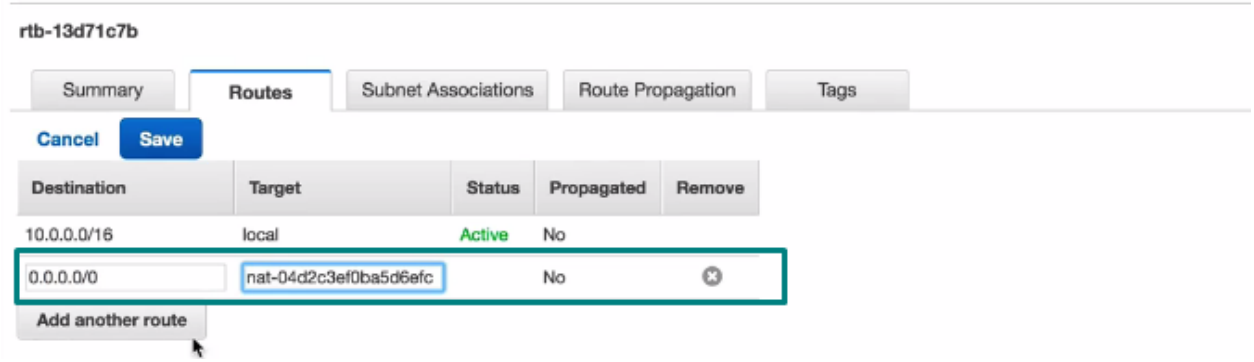




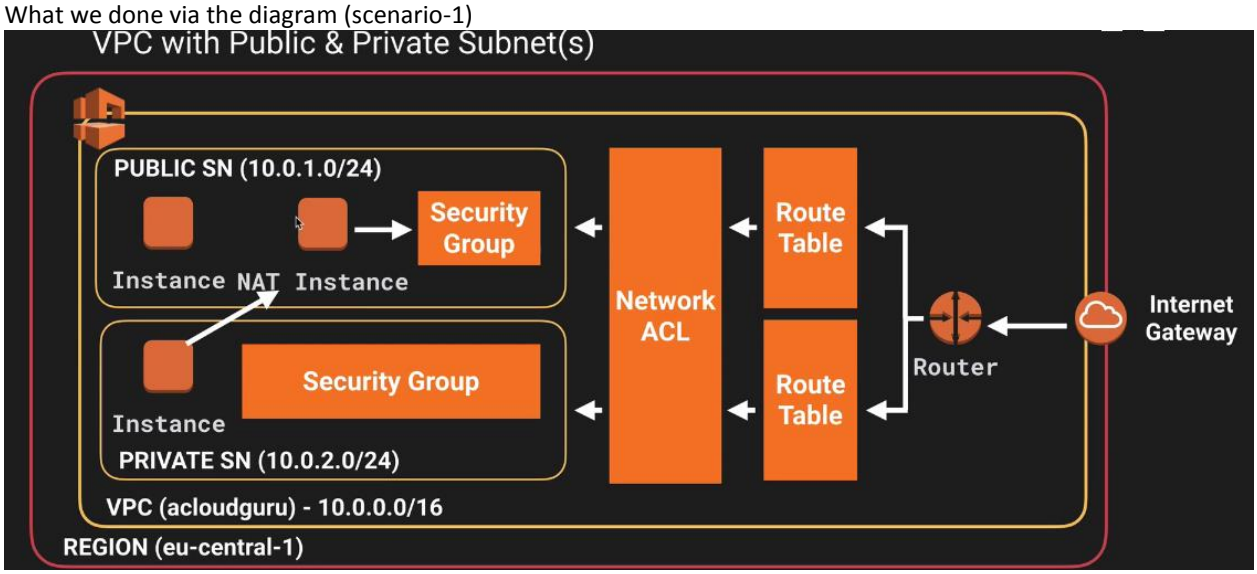
Edit main route table



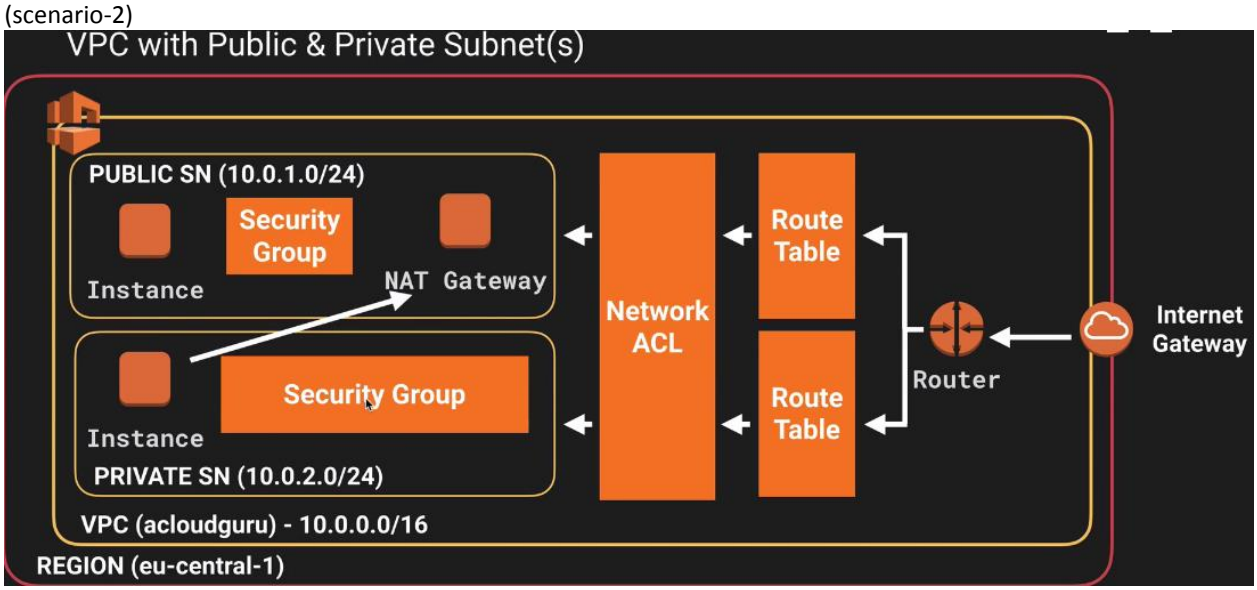
Edit and attach NAT gateway to the main route table in the subnet



What we done via the diagram (scenario-1)



(scenario-2)



Comparison of NAT Instances and NAT Gateways

The following is a high-level summary of the differences between NAT instances and NAT gateways.

Attribute	NAT gateway	NAT instance
Availability	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.	Use a script to manage failover between instances.
Bandwidth	Supports bursts of up to 10Gbps.	Depends on the bandwidth of the instance type.
Maintenance	Managed by AWS.You do not need to perform any maintenance.	Managed by you, for example, by installing software updates or operating system patches on the instance.
Performance	Software is optimized for handling NAT traffic.	A generic Amazon Linux AMI that's configured to perform NAT.
Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.
Type and size	Uniform offering; you don't need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload.
Public IP addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an Elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new Elastic IP address with the instance.
Private IP addresses	Automatically selected from the subnet's IP address range when you create the gateway.	Assign a specific private IP address from the subnet's IP address range when you launch the instance.
Security groups	Cannot be associated with a NAT gateway. You can associate security groups with your resources behind the NAT gateway to control inbound and outbound traffic.	Associate with your NAT instance and the resources behind your NAT instance to control inbound and outbound traffic.
Network ACLs	Use a network ACL to control the traffic to and from the subnet in which your NAT gateway resides.	Use a network ACL to control the traffic to and from the subnet in which your NAT instance resides.
Flow logs	Use flow logs to capture the traffic.	Use flow logs to capture the traffic.
Port forwarding	Not supported.	Manually customize the configuration to support port forwarding.
Bastion servers	Not supported.	Use as a bastion server.
Traffic metrics	Not supported.	View CloudWatch metrics.

Exam Tips - NAT instances

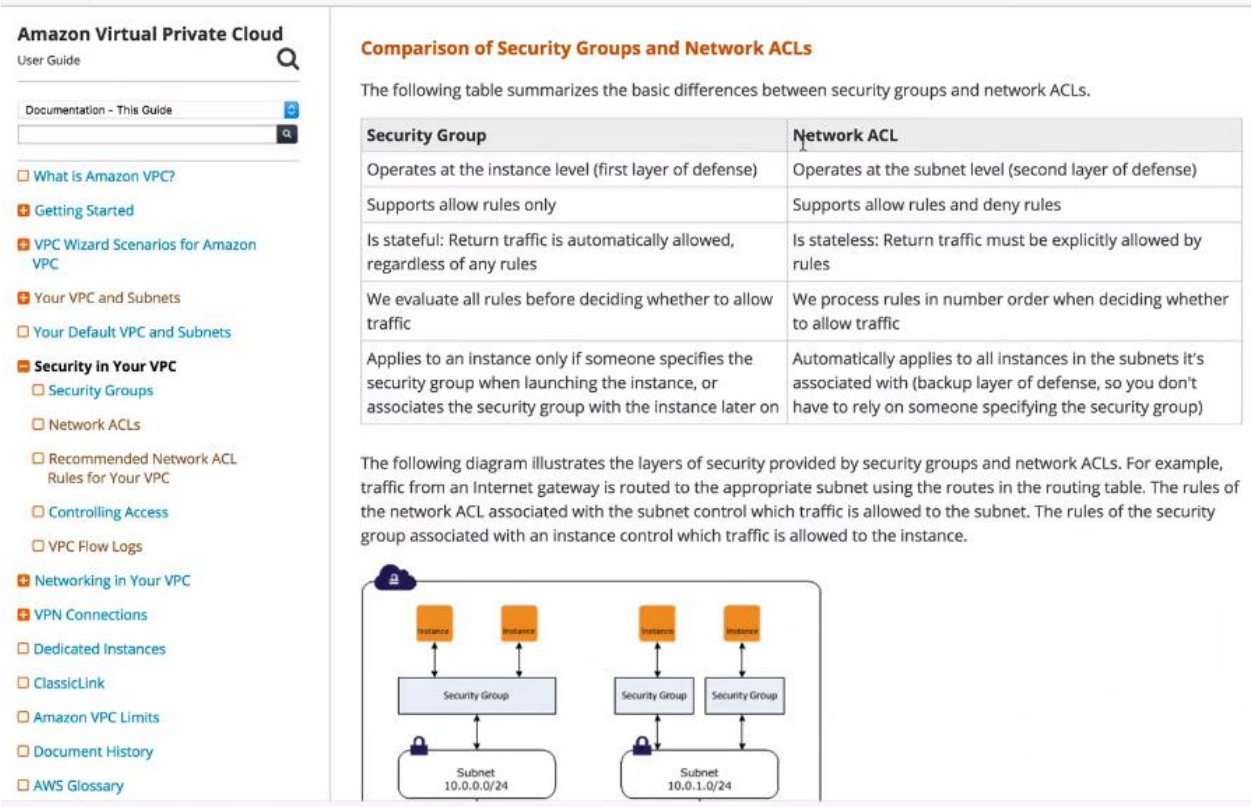
A CLOUD GURU

- When creating a NAT instance, Disable Source/Destination Check on the Instance
- NAT instance must be in a public subnet
- There must be a route out of the private subnet to the NAT instance, in order for this to work
- The amount of traffic that NAT instances supports, depends on the instance size. If you are bottlenecking, increase the instance size
- You can create high availability using Autoscaling Groups, multiple subnets in different AZ's and a script to automate failover
- Behind a Security Group.

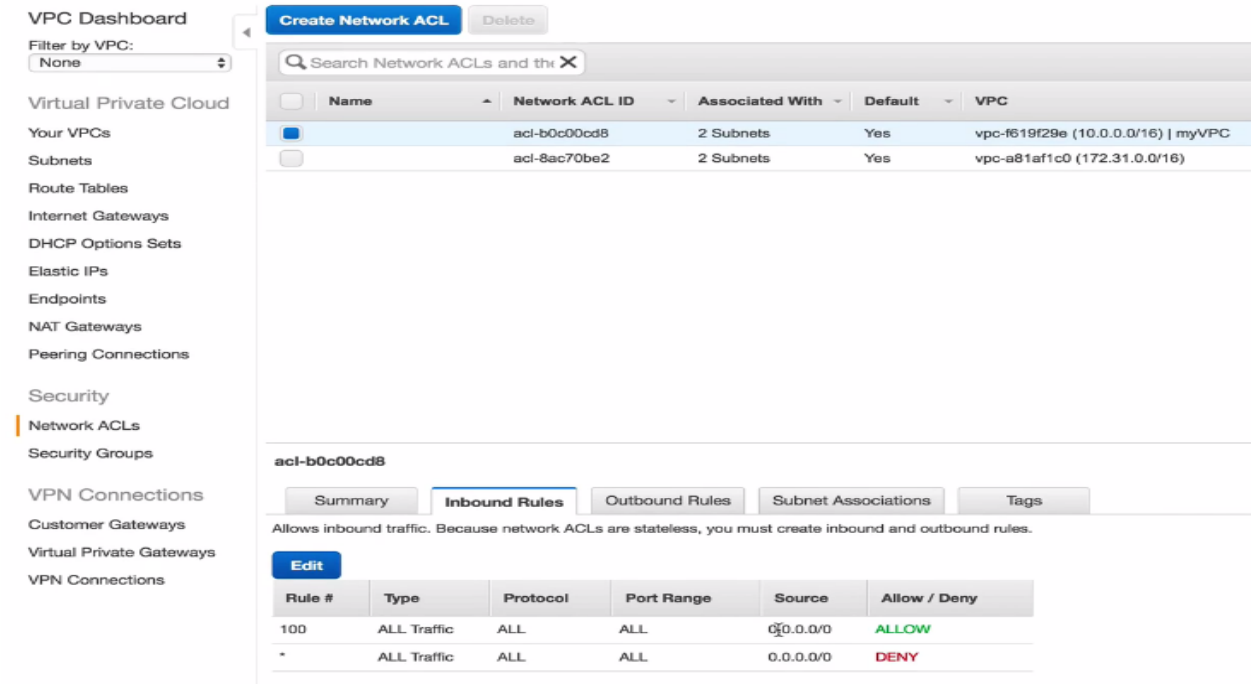
Exam Tips - NAT Gateways

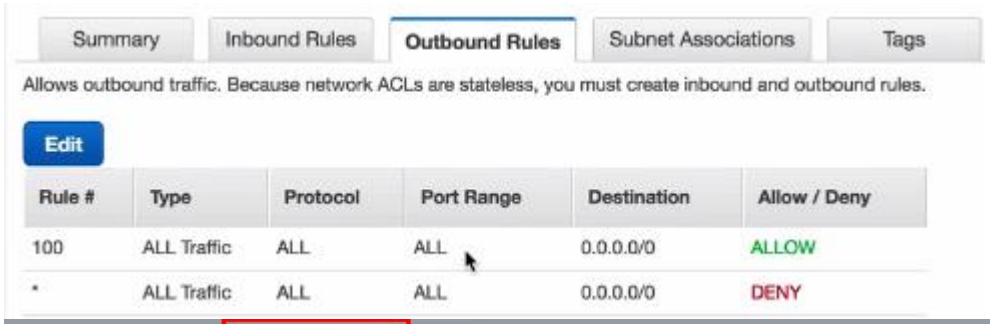
- Very new, may not be in the exams yet.
- Preferred by the enterprise
- Scale automatically up to 10Gbps
- No need to patch
- Not associated with security groups
- Automatically assigned a public ip address
- Remember to update your route tables.
- No need to disable Source/Destination Checks

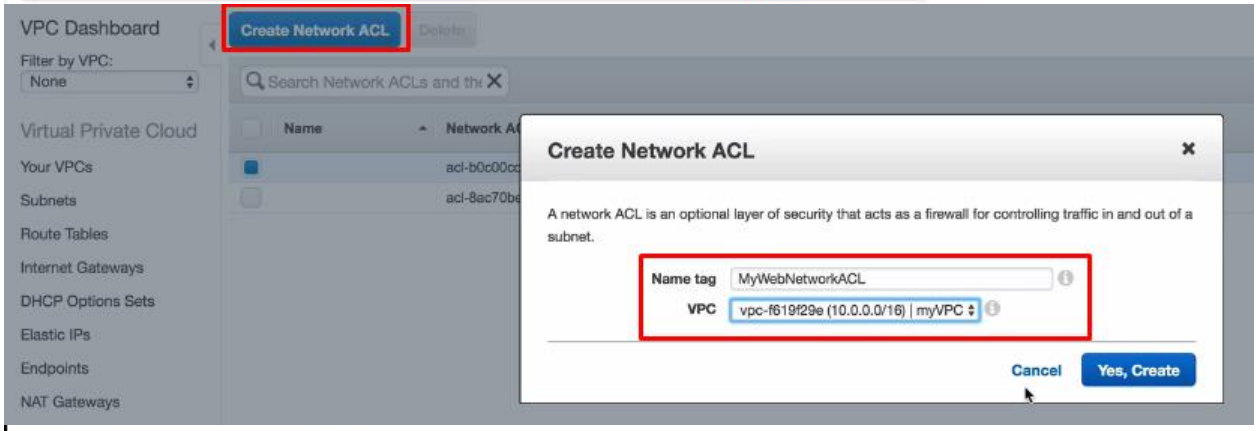
Network Access Control List vs Security Groups

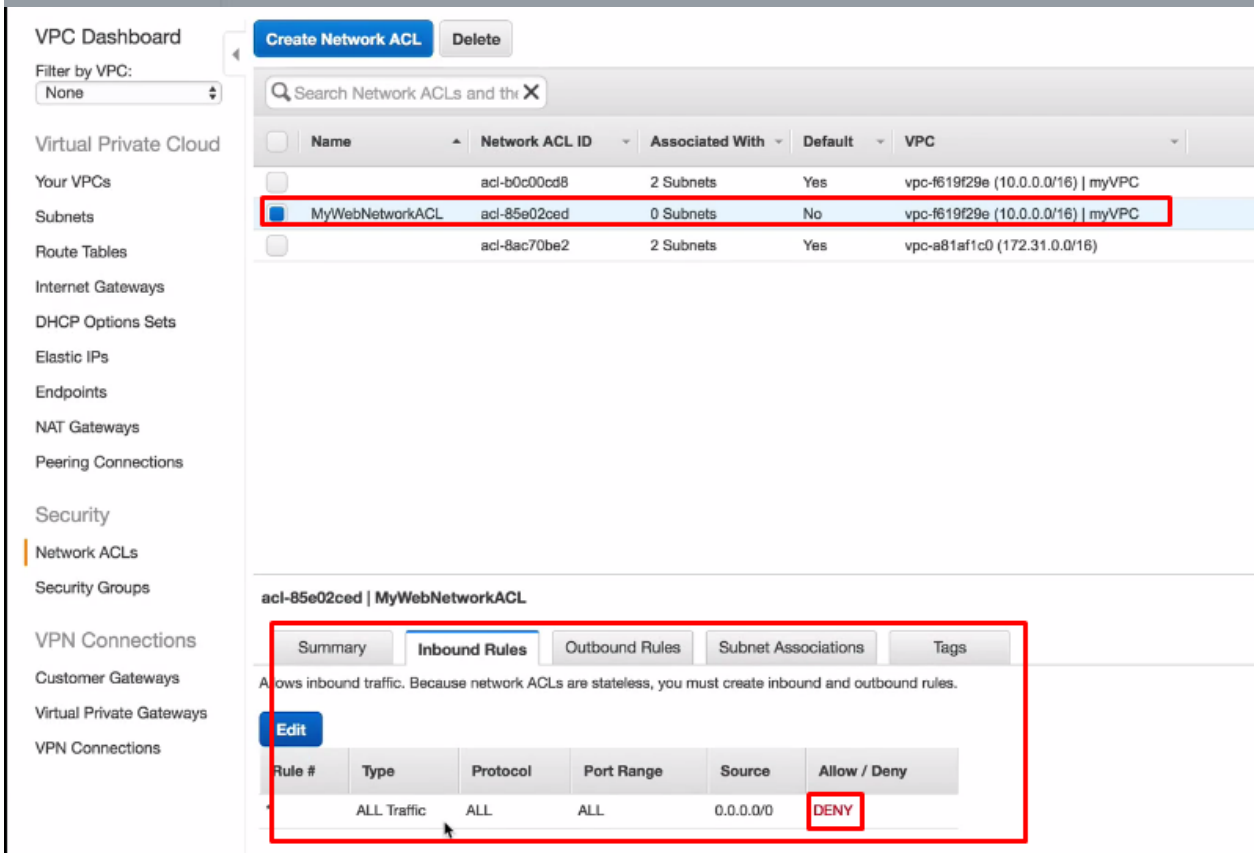


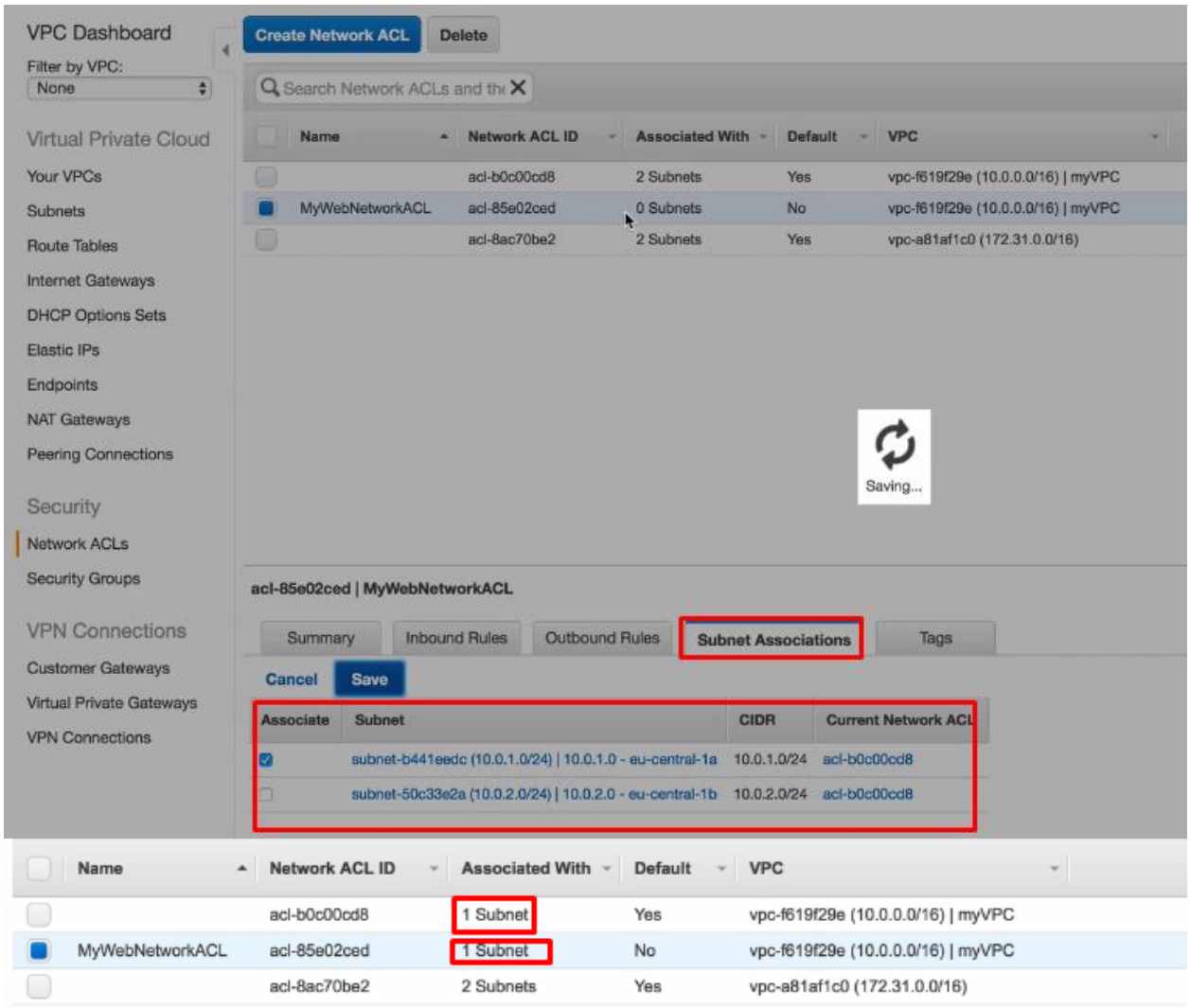
When we create custom VPC, **default network ACL** is created by default which has all inbound and outbound rules defined automatically. On the other hand, if we create acustom network ACL, all inbound and outbound rules are denied by default. One subnet is only be associated with one network ACL




Summary
Inbound Rules
Outbound Rules
Subnet Associations
Tags
Allows outbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.
Edit
Rule # | Type | Protocol | Port Range | Destination | Allow / Deny
100 | ALL Traffic | ALL | ALL | 0.0.0.0/0 | ALLOW
* | ALL Traffic | ALL | ALL | 0.0.0.0/0 | DENY

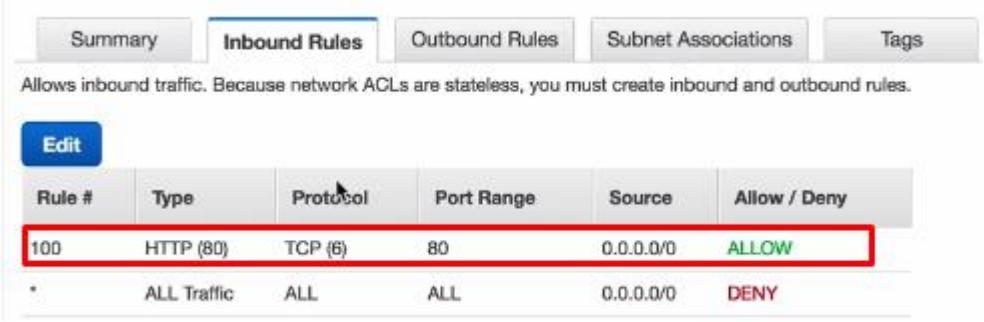

VPC Dashboard
Create Network ACL
Delete
Filter by VPC:
None
Search Network ACLs and th
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
NAT Gateways
Name
Create Network ACL
A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.
Name tag
MyWebNetworkACL
VPC
vpc-f619f29e (10.0.0.0/16) | myVPC
Cancel
Yes, Create


VPC Dashboard
Create Network ACL
Delete
Filter by VPC:
None
Search Network ACLs and th
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
NAT Gateways
Peering Connections
Security
Network ACLs
Security Groups
VPN Connections
Customer Gateways
Virtual Private Gateways
VPN Connections
Name | Network ACL ID | Associated With | Default | VPC
acl-b0c00cd8 | 2 Subnets | Yes | vpc-f619f29e (10.0.0.0/16) | myVPC
MyWebNetworkACL | acl-85e02ced | 0 Subnets | No | vpc-f619f29e (10.0.0.0/16) | myVPC
acl-8ac70be2 | 2 Subnets | Yes | vpc-a81af1c0 (172.31.0.0/16)
acl-85e02ced | MyWebNetworkACL
Summary
Inbound Rules
Outbound Rules
Subnet Associations
Tags
Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.
Edit
Rule # | Type | Protocol | Port Range | Source | Allow / Deny
* | ALL Traffic | ALL | ALL | 0.0.0.0/0 | DENY

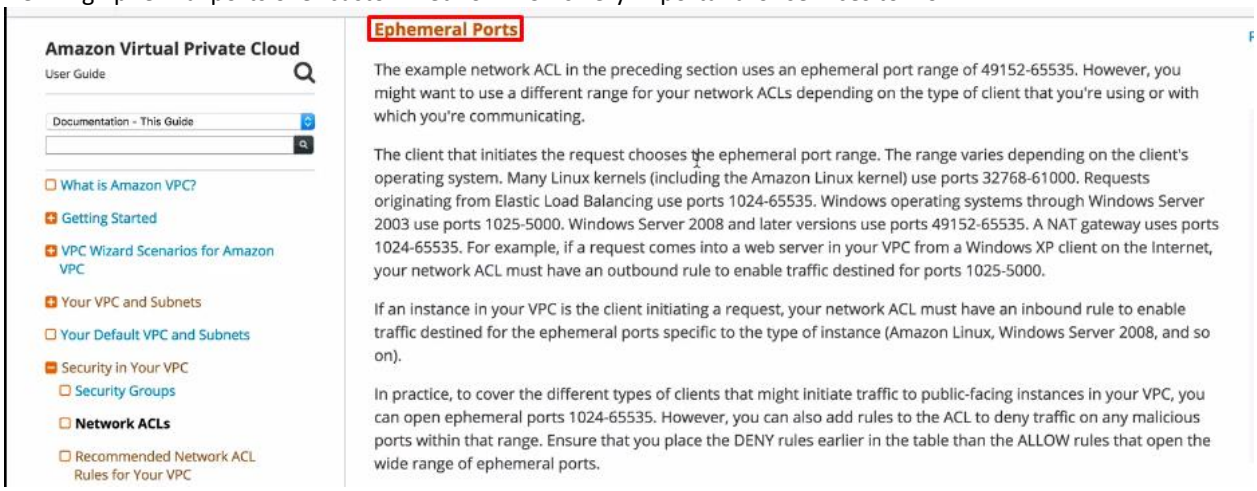


Name	Network ACL ID	Associated With	Default	VPC
	acl-b0c00cd8	1 Subnet	Yes	vpc-f619f29e (10.0.0.0/16) | myVPC
MyWebNetworkACL	acl-85e02ced	1 Subnet	No	vpc-f619f29e (10.0.0.0/16) | myVPC
	acl-8ac70be2	2 Subnets	Yes	vpc-a81af1c0 (172.31.0.0/16)

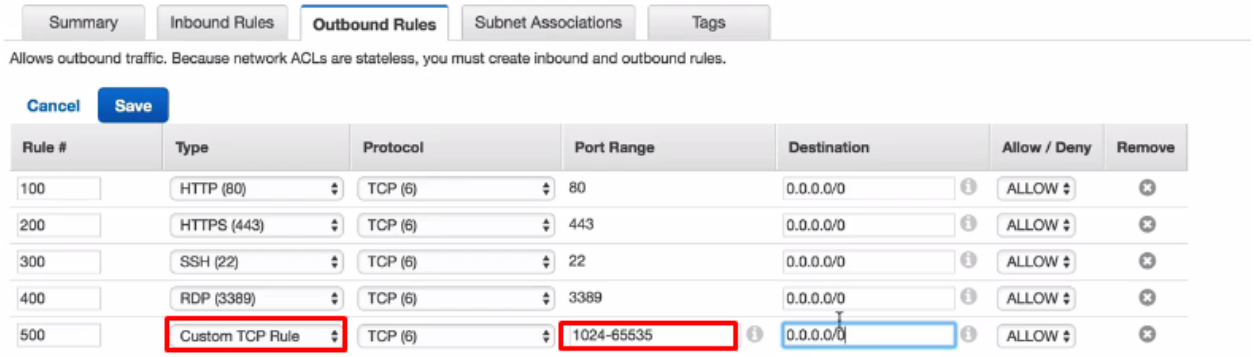
Define inbound and outbound rules for internet facing services to working via this custom network ACL over public subnet



Defining Ephermal ports over custom network ACL is very important for services to work.



Same has to be done for inbound and outbound rules



Below will be ignoed as rule 100 will be run first and would allow everything i.e. blocking specific IP address have to be in right sequence

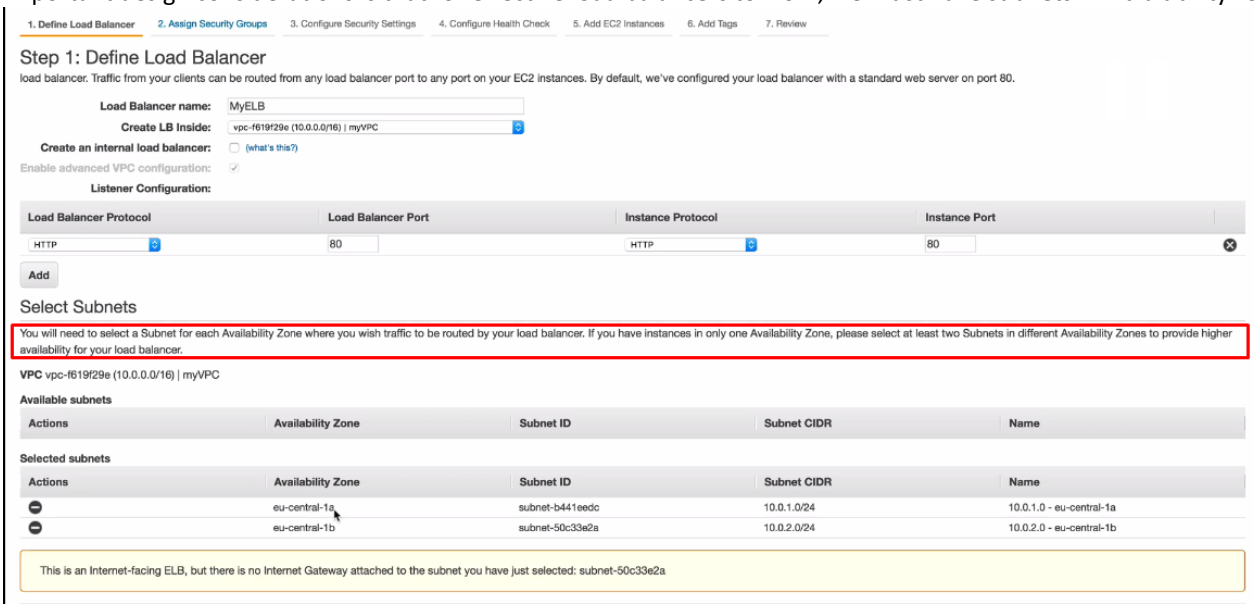
Below rule will basically be run and a specific Ipaddress will be blocked

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
99	HTTP (80)	TCP (6)	80	31.51.9.6/32	DENY

On contrary, in security groups, we have allow ports and cant deny a specific port. All are denied by default and can only allow specific or all IP addresses in security groups

- Your VPC automatically comes a default network ACL and by default it allows all outbound and inbound traffic.
- You can create a custom network ACL. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.
- Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
- You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.
- A network ACL contains a numbered list of rules that is evaluated in order, starting with the lowest numbered rule.
- A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
- Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

Important design considerations is that for effective load balancers to work, we must have subnets in 2 aialability zones,see below



BASTIONS – are EC2 instances dedicated to administer public and private subnets and EC2 instances in VPCs. It sits in public subnet having access to all instances in subnets (private or public) and admins can monitor instances.some time these are also called jump boxes

VPC flow log: logging VPC actvities inside out, flow log can be created using custom IAM, control flow log groups, log stream and this then can be viewed in Cloud Watch Console

- Think of a VPC as a logical datacenter in AWS
- Consists of IGW's (Or Virtual Private Gateways), Route Tables, Network Access Control Lists, Subnets, Security Groups
- 1 Subnet = 1 Availability Zone
- Security Groups are Stateful, Network Access Control Lists are Stateless
- Can Peer VPCs both in the same account and with other AWS accounts.
- NO TRANSITIVE PEERING

- When creating a NAT instance, Disable Source/Destination Check on the Instance
- NAT instance must be in a public subnet
- Must have an elastic IP address to work
- There must be a route out of the private subnet to the NAT instance, in order for this to work
- The amount of traffic that NAT instances supports, depends on the instance size. If you are bottlenecking, increase the instance size
- You can create high availability using Autoscaling Groups, multiple subnets in different AZ's and a script to automate failover
- Your VPC automatically comes a default network ACL and by default it allows all outbound and inbound traffic.
- You can create a custom network ACL. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.
- Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
- You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed.
- A network ACL contains a numbered list of rules that is evaluated in order, starting with the lowest numbered rule.
- A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
- Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

- A NAT is used to provide internet traffic to EC2 instances in private subnets
- A Bastion is used to securely administer EC2 instances (using SSH or RDP) in private subnets. In Australia we call them jump boxes.
- If you want resiliency, always have 2 public subnets and 2 private subnets. Make sure each subnet is in different availability zones.
- With ELB's make sure they are in 2 public subnets in 2 different availability zones.
- With Bastion hosts, put them behind an autoscaling group with a minimum size of 2. Use Route53 (either round robin or using a health check) to automatically fail over.
- NAT instances are tricky to make resilient. You need 1 in each public subnet, each with their own public IP address, and you need to write a script to fail between the two. Instead where possible, use NAT gateways.