

EC2 (Linux) from WebUI (M3/S4)

31 March 2019 20:22

- 1) Create a Security Group (Add inbound rule, open port 22 for 'My IP')
- 2) Create a KeyPair and download the .pem file.
- 3) Convert the .pem file .ppk file using PuttyGen. (<https://goo.gl/Tc8w3Q>)
- 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

```
sudo su
apt-get update
apt-get install apache2
service apache2 start
cd /var/www/html
echo "Hello, world!" > index.html
```

- 8) Access the webpage using the ip address of EC2 in the browser.

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 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/>

The URL should end with a /

Userdata for EC2

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 (Windows) from WebUI (M3/S4)

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 "Download 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 Password" and note down the password.
- 9) Click on the .rdp file on the Desktop and log into Windows with the above password.

S3 Creating a WebSite (M3/S14)

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 `Public Access settings`. Click on edit.

Unselect `Block new public bucket policies (Recommended)` and `Block public and cross-account access if bucket has public policies (Recommended)`. 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-test-786/*"]
  }]
}
```

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.

RDS from WebUI (M3/S16)

31 March 2019 20:23

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 `myrecipiesdb`.
- 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/installers/HeidiSQL_9.5.0.5196_Setup.exe

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 myrecipiesdb on left pane. Go to the Query Tab and run the below queries.

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

```
INSERT INTO recipies  
  (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"

Glacier (M3/S23)

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.

<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>

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 that SNS Topic is optional. If not specified the status of the job should be checked manually.

```
{
  "Type": "archive-retrieval",
  "ArchiveId": "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`

Vault
/ |
a1 a2 a3

VPC Wizard (M3/S28)

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 be 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.

Elastic Beanstalk (M3/S32)

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>

- 1) Download the java-tomcat-v3.zip from <https://goo.gl/8QUers> to the Desktop.
- 2) Click on Actions -> Create Environment. Select Web sever 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 downloaed 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"

10 April 2019 20:06

31 March 2019 20:06

- Multiple profiles
- Documentation

1a) AWS CLI for Windows

1b) AWS CLI for Ubuntu

After the installation add aws binary to the PATH.

```
sudo apt-get install python3 python3-pip
```

```
pip3 install awsccli --upgrade
```

2) Generate the credentials from the below link

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-0a313d6098716f372 --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/userguide/install-linux.html#install-linux-awscli>

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

- 1) SG
- 2) KP
- 3) EC
- 4) login
- 5) terminate

```
subnet-e502d38c
sg-089d0577fcb3e52d1
```

ingress → EC2

← outgress

— {
— {
— {

AWS SDK Examples for EC2 (M4/S17)

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 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

Others

<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.

Code for creating an ondemand EC2 instance

package com.amazonaws.samples;

```
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
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
            AWSCredentials credentials = new ProfileCredentialsProvider("default").getCredentials();
```

```
            // Create the EC2 client to call different EC2 APIs
            AmazonEC2 amazonEC2Client = AmazonEC2ClientBuilder.standard()
                .withCredentials(new AWSStaticCredentialsProvider(credentials)).withRegion("us-east-1").build();
```

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

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

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

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

```
    }
```

```
}
```

1 - Java
1 - Eclipse (IDE)
✓ - AWS Tool Kit for Eclipse

VS
add on for AWS
C#
Linux
Mac
Win
VS Code (free)
Microsoft

10 April 2019 20:06

SNS (M5/S4)

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.

Alarm (M5/S14)

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 Alerts (M5/S19)

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.

CloudWatch Logs (M5/S22)

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.

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

- Name as CloudWatchAgentServerRole
- 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 -E ./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 -> `Imageld`, `InstanceId`,

`InstanceType`, `cpu` -> Select a metric

The graph should be populated.

Volume Check (M5/S52)

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

10 April 2019 20:06

Reserved Instances (M6/S4)

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 (M6/S6)

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-d0cb0b1d5ae5>

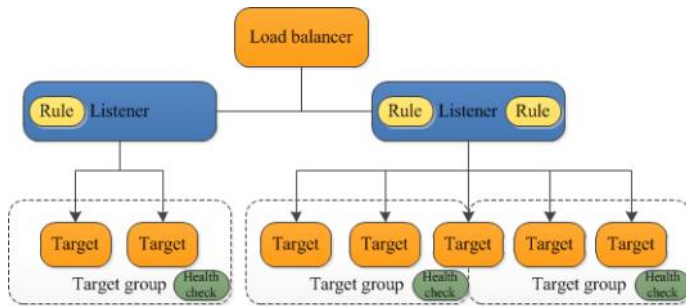
What is ALB - <http://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>

<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>

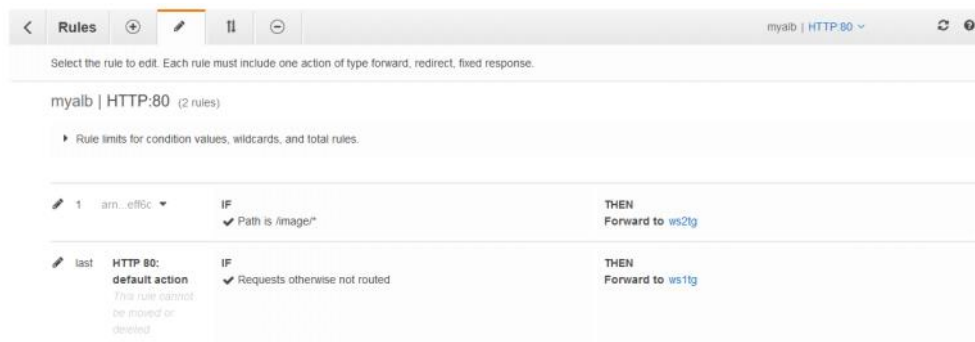
Monitoring the ELB - <http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-monitor-logs.html>

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



- demo of path based routing
- demo of health check
- with application load balancer
- diff between different lb
- Warming up (AWS vs GCP)

- 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 img folder and put an image in it.
- Create two target groups one for each Linux instance with the webserver attached
- Create a Application ELB with the first target group
- Modify the listeners to map /img/* path pattern to the second target group
- Now all the urls with img should be redirected to the second target group



Non Fixed AutoScaling (M6/S25)

15 August 2017 19:39

- AutoScalingGroups
- Launch Configuration vs Launch Templates
- Scaling Policy Types
- Cooldowns

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

<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

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

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

Fixed AutoScaling (M6/S25)

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.

RDS Mutlti AZ Failover (M6/S40)

08 April 2019 08:06

- 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>>