**RESEARCH DOCUMENT – Research on Amazon Web Service**

|  |  |
| --- | --- |
| **Document Purpose** | This document contains the research information on Amazon Web Service |
| **Document id** | GS-NA-RES-001-0.1 |
| **Document status** |  |
| **Client** | Internal |

**Change Control**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Rev** | **Written / Revised by** | **Written / Revised date** | **Comment** | **Reviewed by** | **Appr. by** | **Appr. date** |
| 0.1 | Hariharan T | 14-12-2022 | Research document first revision. |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Additional Revision Comments, if any:**

Amazon Web Service

[1 Core Services of AWS 3](#_Toc121943842)

[1.1 Elastic Compute Cloud (EC2) 3](#_Toc121943843)

[1.2 Simple Storage Service (S3) 5](#_Toc121943844)

[1.3 Relational Database Service (RDS) 6](#_Toc121943845)

[1.4 Route53 7](#_Toc121943846)

# Core Services of AWS

* Elastic Compute Cloud (EC2)
* Simple Storage Service (S3)
* Relational Database Service (RDS)
* Route 53

## Elastic Compute Cloud (EC2)

EC2 instance is essentially virtual server which is operating system-agnostic.

**Description**: Solution to computing needs.

* Run applications
* Virtual desktop
* Installing and use 3rd party software
* Computing

**Features:**

* Elastic computing service can expand or retract as needed
* Rules can be set, based on that scaling up and down happens automatically

**Process to avail EC2 service,**

1. **Amazon Machine Image (AMI)**

**Description:** Image is the combination of an Operating System and then some application pre-installed. Applications like (Java or Python).

Images are updated frequently by amazon, we will not need to worry about that.

Amazon only updates the images and not the software in that image.

1. Choose Instance Type

**Description**: This is basically a spec for our instance, specs like number of CPUs, Memory, Internal storage.

Amazon categorized these as a family we have to choose the type based on our needs.

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose | vCPU | Memory | Network |
| General Purpose t3.large | 2 | 8 | Up to 5Gbps |
| Memory Optimized r5.large | 2 | 16 | Up to 10Gbps |
| Compute Optimized c5.large | 2 | 4 | Up to 10Gbps |

1. Configure Instance

**Description**: Configure your instances like number of instances and various other configuration based on our needs.

1. Add Storage

**Description:** Add storage for our instance.

Storage is termed as Elastic Block Storage (EBS).

This is a storage service that makes easy to calculate charges for storage. This volume is independent to our EC2 instance and can be retained and deleted when the EC2 instance is terminated.

1. Tag Instance

**Description**: This is basically to tag our instance for our own purpose.

1. Configure Security Group

**Description:** Setup security-based configuration for our instance.

Example security group scenarios,

* IP-based communication rules for a single or group of EC2 instances.
* Control who can SSH into EC2 instance.
* Allow access between EC2 instances.
* Allow access database.
* Accept HTTP requests.

1. Review and Add Instance to Key Pair

**Description**: Review the configuration and add the instance to key value pair for grouping purpose.

**Pricing:** (based on On-demand instances)

|  |
| --- |
| We can able to on and off instance based on our use case.  Pricing based on, On Demand Instance. There is other price model are Reserved instances and spot instances, which are cheaper than this.  These are just sample (given in article) pricing details, I’ll update this in future documents with current pricing. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AMI | Rate / hr ($) | Daily rate ($) | AMI | Rate / hr ($) |
|  |  |  |  |  |
| Amazon Linux 2 | 0.126 | 3.024 | Amazon Linux 2 | 0.126 |
|  |  | 0 |  |  |
| Amazon Linux 2 | 0.013 | 0.312 | Amazon Linux 2 | 0.013 |

**EC2 Instance Pricing Types**

* On-demand instances (Cheap)
* Reserved instance (Cheaper)
* Spot instance (Cheapest)

**Current EC2 free offering**

Each month for one year!

* 750 hours of t2.micro running Linux
* 750 hours of t2.micro running Windows Server
* 750 hours of Elastic Load Balancing + 15 GB processing
* 30GB Elastic Block Storage.

**Additional Pricing**

* Elastic Block Storage - $0.10 USD per GB/month
* Auto Scaling Groups – Free
* Load Balancer - $0.0225 USD/hour

## Simple Storage Service (S3)

**Description**: Used to store files.

Can store any type/kind of file max file size is 5GB (according to amazon doc). But there are other methods to upload up-load a large object in separate operations up to 5TB.

**Bucket:**

Bucket is your root resource to which you can add, delete or modify objects.

We can add all sort of configuration option like permission, hosting and logging.

**Can be configured to,**

* Trigger events when objects are added/modified/deleted.
* Preserve older versions of objects
* Replicate objects across regions.

Buckets are accessed via URL, which we can access the objects with in them.

We can use it as a static website, S3 makes it simple to use for hosting a static website.

While configure S3 there is a option to enable static website hosting for the bucket.

Bucket itself can able to response to requests and be enable to interact with Route53 URLs.

**This is a quick and easy way to get a static website up with minimal cost.**

**To solve latency issue,**

If hosting site like assets for a website like images, to solve latency issue we can automatically replicate files to other regions.

But Amazon also provide better solution by using its other service called Cloud Front, By Cloud Front we can cache our content (sometimes called edging) at locations around the world.

**Pricing,**

Prices are different based on region

These are just sample (given in article) pricing details, I’ll update this in future documents with current pricing.

|  |  |  |
| --- | --- | --- |
| Amount of data stored ($) | Number of Request ($) differ for type request | Amount of data transferred ($) |
|  |  |  |
| 0.023 / GB | 0.005 / 1000 put request | 0.09 / GB (1st GB month is free and then this price is appicable) |

## Relational Database Service (RDS)

**Description:** Collection of AWS services to manage relational databases.

**Managed Databases Aspects,**

* Scheduled automated backups
* Simple software updates
* Managed Infrastructure

**Currently Amazon offers RDS Database options,**

* MySQL
* PostgreSQL
* Microsoft SQL Server
* MariaDB
* ORACLE DATABASE
* Amazon Aurora

Each carries a different price and different configuration options.

Amazon takes care of security aspects.

With RDS, access controlled via trusty security group and with security group we can allow our application to access database and bock any other connection. We can also configure ways for BI tools to connect so that the user can query the database.

AWS also provides NoSQL and document-based database.

* DynamoDB (NoSQL Database)
* DocumentDB (MangoDB-compatableNoSQL Database)

**Pricing,**

RDS pricing depends on types of database, region, EC2 instance type.

These are just sample (given in article) pricing details, I’ll update this in future documents with current pricing.

|  |  |  |
| --- | --- | --- |
| Types of database | Region | Ec2 instance type |
|  |  |  |
| MySQL Database | Oregon Region | db.m4.xlarge Instance |

MySQL is an open source, there is no license needed to run.

You can expect other database like MS SQL SERVER, Oracle cost a lot more per hour.

## Route53

**Description:** Solution for DNS needs.

We can use domain names that we already own or we can register new one through AWS.

Domain Name System (DNS): System translates human -readable URLs to IP address.

EC2 instances can be configure with the public IP addresses, but certain amazon resources like S3 buckets or Load Balancers are little more complicated and don’t have static visible IP addresses.

So Route53 allows us to set up URL resolution to AWS resources directly, bypassing any for to see an IP. Of course, behind the scenes, IP resolution is happening, but Route53 conveniently abstract this way.

Route53 is Core letting users interact with services in AWS.

Setup Route53,

* Setting hosted zone
  + Hosted zone is nothing but a root domain name like example.com or google.com
* Using hosted zone, we can use Route53 to setup subdomain like www.example .com and configure them to route AWS resources.

There are few more features that Route53 provides. One of the most useful is health check service.

* This allows us to set up regular checks for the given URL path.
* Health check will send us the alert based on different rules, like if the URL instance gets a response of 503 or 404.
* This is an essential tool if we have customer facing web application. And at $0.50 per month for each alarm.

**Pricing,**

|  |  |  |
| --- | --- | --- |
| Per Hosted Zone | Queries | DNS Entries |
|  |  |  |
| 0.50 USD /month | 0.40 USD/ 1M | Free |