**Refactoring The Topology**

In previous project we lift our onpremise topology and shift it to AWS. We were responsible to managing the infrastructure. In this project, we will re-architect the topology with AWS services to improve agility and continuity.

|  |  |
| --- | --- |
| **SaaS/PaaS Services** | **IaaS / On-prem Services** |
| Beanstalk | Tomcat on VM |
| ELB in Beanstalk | NginX Load Balancer |
| Autoscaling | None |
| EFS/S3 | NFS |
| RDS | MySQL on VM |
| Elastic Cache | Memcached on VM |
| Active MQ | RabbitMQ on VM |
| Route53 | GoDaddy, Local DNS |
| CloudFront | None |

## **Flow Of Execution**

* Login to AWS Account
* Create Key pair for Beanstalk Instance Login
* Create Security Group For Elasticcace, RDS & ActiveMQ
* Create RDS, Amazon Elastic Cache, Amazon Active MQ
* Create Elastic Beanstalk Environment
* Update SG of Backend to Allow Traffic From Bean SG
* Update SG of Backend to Allow Internal Traffic
* Launch Ec2-Instance for DB Initializing
* Login to the instance and Inititialize RDS DB
* Change healthcheck on beanstalk to /login
* Add 443 https Listenet to ELB
* Build Artifact with Backend Information
* Deploy Artifact to Beanstalk
* Create CDN with SSL Cert
* Update Entry in GoDaddy DNS Zones

## **AWS Services For This Project**

Before starting with the topology let’s summarize this services as a reminder.

### **1. Elastic Beanstalk**

Elastic Beanstalk is a service for deploying and scaling web applications and services. Upload your code and Elastic Beanstalk automatically manages deployment, from capacity provisioning, load balancing and autoscaling to application health monitoring. This service; It is AWS's PaaS service.

### **2. Autoscaling**

AWS Auto Scaling automatically creates all your scaling policies and sets goals for you based on your preferences. AWS Auto Scaling monitors your application and adds or reduces capacity to your resource groups in real-time as demand changes.

### **3. Elastic File System – EFS**

Amazon Elastic File System (EFS) is designed to deliver serverless, fully elastic file storage that helps you share file data without the need to provision or manage storage capacity and performance. Amazon EFS file systems can automatically scale from gigabytes to petabytes without needing to provision storage.

* Amazon EFS is a file storage service for use with Amazon compute (EC2, containers, serverless) and on-premises servers. EFS; It offers a file system interface, file system access semantics (such as high consistency and file locking), and storage that can be accessed by thousands of EC2 instances simultaneously.

### **4. Simple Storage Service – S3**

Amazon S3 is an object storage service built to retrieve and store data of any size, from anywhere (IoT, Web, Mobile, etc.). Provides 99.999999999% durability. It is a simple storage service that offers industry-leading durability, availability, performance, security, and virtually unlimited scalability at very low costs. S3 offers comprehensive security and compliance features that meet even the most stringent regulatory requirements. Here, while you take care of your data, AWS is responsible for managing the server.

**5. Relational Database Servise – RDS: Amazon Aurora**

Amazon Relational Database Service (Amazon RDS) is a managed service that makes it easy to set up, run, and scale a relational database in the cloud. It provides cost-effective and resizable capacity, as well as the freedom to focus on your applications and business by managing time-consuming database administration tasks. Amazon RDS allows you to access the features of a known PostgreSQL, MySQL, MariaDB, Oracle, SQL Server database. Additionally, Amazon Aurora, also part of RDS, has its own versions of Aurora MySQL and Aurora PostgreSQL.

### **6. Amazon MQ**

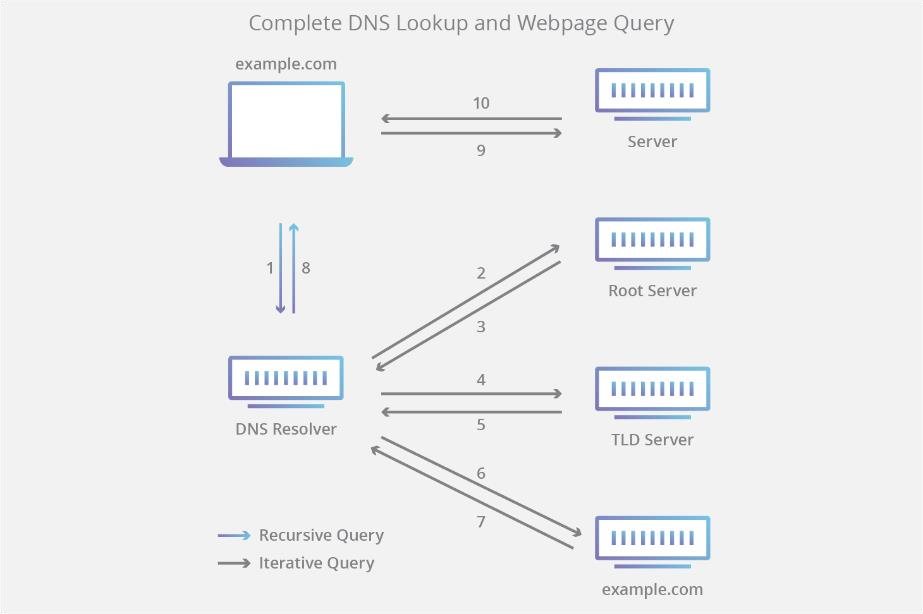
It is a fully manageable open source message broker service offered by AWS. Amazon MQ is a managed message broker service that makes it easy to set up and run message brokers on AWS for Apache ActiveMQ and RabbitMQ. Amazon MQ reduces your operational burden by managing the provisioning, installation, and maintenance of message brokers for you. Because Amazon MQ accesses your existing applications through industry-standard APIs and protocols, you can easily migrate to AWS without having to rewrite code.

### **7. Amazon Route53**

Amazon Route 53 is a highly available and scalable cloud domain name system (DNS) web service. It is designed to provide a highly reliable and cost-effective way for developers and businesses to direct end users to internet applications by translating names like www.example.com into numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is also fully compatible with IPv6.

Amazon Route 53 traffic flow makes it easy for you to manage traffic through a variety of routing types, including latency-based routing, geo-DNS, geo-proximity, and weighted round-trip, all of which can be combined with DNS failover to create a variety of low-latency, fault-tolerant architectures .

**How DNS Works?**

[](https://private-user-images.githubusercontent.com/54971670/260381718-b0691d41-8153-45bf-a1d7-29d531d439c9.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..eADk6F3oI0q5NP9PtniLMYPAMNsxGPxdFzmpqn2fxHg)

1. A user types 'example.com' into a web browser and the query goes to the internet. It is retrieved by a DNS recursive resolver.
2. The resolver then queries a DNS root name server.
3. The root server then responds to the resolver with the address of a parent 4th-level domain (TLD) DNS server (such as .com or .net) that stores information for domains. When searching for example.com, our request is directed to the “.com” TLD.
4. The resolver then makes a request to the “.com” TLD.
5. The TLD server then responds with example.com, which is the IP address of the domain's name server.
6. Finally, the recursive resolver sends a query to the domain's name server. The IP address for example.com is then returned from the name server to the resolver.
7. The DNS resolver then responds to the web browser with the IP address of the initially requested domain.
8. When step 8 of the DNS lookup returns the IP address for example.com, the browser can request the web page:
9. The browser makes an HTTP request to the IP address.
10. The server at this IP returns the web page to render in the browser.

### **8. Cloudfront**

Amazon Cloudfront is a global content delivery network (CDN) service that securely delivers data, video, applications and APIs to viewers with low latency and high transfer speeds. In addition to physical locations directly connected to the AWS global infrastructure, Cloudfront provides resources for your applications; DDoS mitigation is integrated with services such as Amazon S3, ELB or AWS Shield for EC2.

**AWS Topology**

A screenshot of a computer

Description automatically generated