

DOCUMENTATION OF MY PROJECT

EXPLANATION AND REASON FOR EC2 CONNECTING WITH JAVA

- EC2(Elastic Compute Cloud) is a service provided by Amazon Web Services that allows us to rent virtual computers in the cloud.
- It's like having our own virtual server in the cloud where we can choose the size and configuration of the virtual machine.
- It's flexible, scalable, cost-effective, and perfect for running applications, websites, and databases without investing in physical hardware.
- EC2 is used with java to run the application on the internet because it provides scalable, flexible computing resources in the cloud.
- It provides benefits such as easy management, flexibility in database integration, high availability, rapid deployment and scaling and cost optimization, making it a powerful combination for working websites efficiently in the cloud.

EXPLANATION AND REASON FOR RDS CONNECTING WITH EC2

- RDS (Relational Database Service) is a managed database service provided by Amazon Web Services that makes it easy to set up, operate, and scale a relational database in the cloud.
- With RDS, we choose from popular database engines such MySQL, PostgreSQL, Oracle, and Microsoft SQL Server. I used MySQL workbench in this project.
- It also provides automated backups, automated software patching, automated scaling, and the ability to replicate databases for high availability and performance.

- It simplifies the process of running a relational database in the cloud allowing you to focus on your application data rather than managing the underlying database infrastructure.
- By connecting RDS in our EC2 application, we can store and retrieve data from RDS in our EC2 and allow our application to access and manipulate data easily.
- By connecting RDS in our EC2 application, we can ensure that the data is securely stored and accessed, allowing us to scale and manage relational database for your applications in an efficient way.

EXPLANATION AND REASON FOR S3 CONNECTING WITH EC2

- S3(Simple Storage Service) is a cloud-based object storage provided by Amazon Web Services which allows us to store and retrieve files like documents, images, videos, and other types of data in the cloud.
- We use buckets as containers to store objects where we can create, configure, and delete buckets to organize objects.
- The object in it is individual data that we store in S3. We can manage objects in S3 using AWS console, SDKs, or APIs.
- S3 is a scalable, durable, and cost-effective object storage that allows us to manage files in the cloud with flexible access controls and data management features.
- Generally, S3 and EC2 are used together in AWS for data storage, data backup, data transfer, scalability, and flexibility. They provide high performance applications in the cloud.

REASON FOR SELECTING S3, EC2, RDS FOR MY PROJECT

- These services are popular and largely used because of their ease of use, and performance when compared to other services.

- These also come with comprehensive documentation, community support, and a wide range of features that make them suitable for a variety of uses for different applications.
- It provides a reliable and secure infrastructure which makes these services appealing to all businesses use.
- Apart from these services above, some services have some disadvantages like cost, complexity and learning curve, familiarity.
- Users may select the services based on their requirements.
- In this project we used the images to store in browser, so we used S3 services. If there is a usage of storing file, we might have used EFS (Elastic File System) which is a file storage.

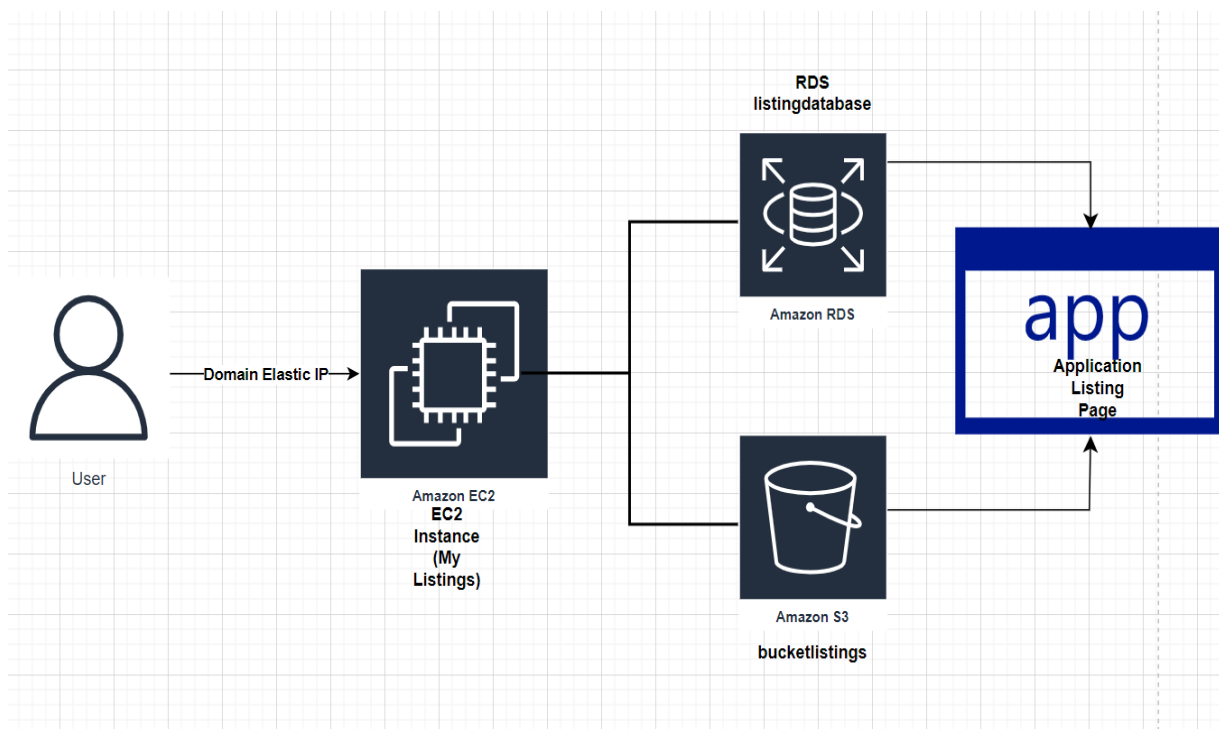
SCALING YOUR PROJECT: MANUAL vs AUTO

- In **Manual Scaling**, we can manually adjust the capacity of our AWS resources to meet our needs.
- To manually scale your application, we use AWS CLI or AWS SDKs to perform actions like launching or terminating, changing instance types or modify RDS database instances.
- In **Auto Scaling**, the service automatically adjusts the capacity of our resources based on predefined rules.
- Using Auto Scaling in combination with other AWS services like EC2, RDS, and Elastic Load Balancing, you can create highly scalable applications that automatically adapt to change.
- Auto scaling is preferred over manual scaling in AWS because it automatically adjusts resources based on demand, increases availability, optimizes costs, provides flexibility, and simplifies management for many applications in AWS.
- I preferred using Auto scaling in my project.

STEPS FOLLOWED TO RUN THE PROJECT (spring-listings-mvc)

1. I uploaded the project **spring-listings-mvc** on my local computer.
2. I created an EC2 instance **My Listings** in AWS console.
3. I configured the EC2 instances by following the commands in digital ocean to get Tomcat Apache page in the browser by visiting <http://<IP>:8080>.
4. Using Tomcat Apache on EC2 provides a scalable, reliable, and cost-effective solution for running web applications in the cloud, with additional benefit of easy deployment, and vibrant community for support.
5. I created access key details under Security Credentials in IAM of AWS Management Console.
6. I created a bucket on S3 **bucketlistings** and connected it to the EC2 instance.
7. I created a new RDS instances **listingdatabase** and connected it to the EC2 instance.
8. After creating the above services, I changed the respective bucket name in **ListingsController.java** and the URL in **Database.java**.
9. Open the MySQL workbench created with RDS instance and upload the **CREATE_DATABASE.sql** in it and run it.
10. I uploaded the correct jdbc driver using FileZilla.
11. I deployed the WAR file by visiting <http://<IP>/manager>.
12. I opened IAM on AWS and created a role which helps in connecting S3 with EC2 instance.
13. Finally, I tested by visiting the link and viewed the listing page as a result.

DIAGRAM OF HOW THE SERVICES ARE CONNECTED



PROS AND CONS OF THREE CLOUD PLATFORM

CLOUD PLATFORM	ADVANTAGES	DISADVANTAGES
AWS (Amazon Web Services)	<ul style="list-style-type: none">Global infrastructure with data centers in multiple regions.High reliability and scalability.Extensive documentation and community support.	<ul style="list-style-type: none">Can be complex and overwhelming for beginners.Pricing structure can be complicated.Some services may have a steeper learning curve.

AZURE	<ul style="list-style-type: none"> • Wide range of services for both Windows and Linux environments. • Strong focus on enterprise customers. • Robust security and compliance features. 	<ul style="list-style-type: none"> • Not as extensive as AWS in terms of service offerings and global infrastructure. • Documentation and community support may not be as AWS. • May require additional setup and configuration for certain services.
GCP (Google Cloud Platform)	<ul style="list-style-type: none"> • Strong focus on data analytics and machine learning. • Competitive pricing in certain areas. • Integrated with Google's data and analytics services. 	<ul style="list-style-type: none"> • Smaller ecosystem compared to AWS and Azure. • May not have the same level of enterprise readiness as AWS and Azure. • Limited presence in certain regions compared to AWS and Azure.

COMPARISONS OF THREE CLOUD PLATFORM

FEATURES	AWS	AZURE	GOOGLE CLOUD
Age	11 years old.	5 years old.	6 years old.
Pricing	Per second pricing with a 60-second minimum.	Per-minute basis.	Per-minute basis.
Compute	EC2(Elastic Compute Cloud) provides all the computing administration.	With Microsoft Azure, you can create virtual machines and scale sets for virtual machines.	As part of GCP (Google Cloud Platform), GCE (Google Compute Engine) does a similar function.

Block Storage	Amazon EBS (Elastic Block System) with up to 64 TB per volume.	Azure Managed Disks with up to 32 TB per disk.	Google Cloud Persistent Disk with up to 64 TB per volume.
Market Share	Market leader with the largest market share.	Second largest market share and strong enterprise focus.	Growing market share with focus on data analytics and machine learning.
Career Prospects (Updated 2022)	AWS Developers make \$117,553 per year.	Azure Developers make \$124,029 per year.	Google Cloud Developer make \$115,083 per year.
Regions and Availability	AWS divides into 22 geographical regions, 14 data centers and there are over 114 edge locations.	Azure operates in 54 regions, each with at least 3 availability zones and 116 edge locations.	Google Cloud comprises 34 cloud regions, 103 zones, and 200 plus edge locations.
Notable Users	Expedia, Netflix, Coursera, Coinbase, Formula 1, Intuit, Airbnb, Lyft, Food and Drug Administration (FDA), Coca Cola.	Bosch, Audi, ASOS, HSBC, Starbucks, Walgreens, 3M, FedEx, Walmart, HP, Mitsubishi Electric, Renault.	Toyota, Equifax, Nintendo, Spotify, The Home Depot, Target, Twitter, PayPal, UPS.
Networking	Amazon Virtual Private Cloud (VPC).	Azure Virtual Network (VNET).	Cloud Virtual Network.
Security	AWS Security Hub.	Azure Security Center.	Cloud Security Command Center.
File Storage	EFS (Elastic File System).	Azure Files.	ZFS (Zettabyte File System) and Avere.
Load Balancing	Elastic Load Balancing.	Load Balancing for Azure.	Cloud Load Balancing.
Automation	AWS Opsworks.	Azure Automation.	Compute Engine Management.
Caching	Elastic Cache.	Redis Cache.	Cloud CDN (Content Delivery Network).

DNS (Domain Name System) services	Amazon Route 53.	Azure Traffic Manager.	Cloud DNS.
Backup Services	Glacier.	Archive Storage, Backup, Site Recovery.	Nearline (frequently accessed data), Coldline (infrequently accessed data).
Compliance	AWS CloudHSM (Hardware Security Models).	Azure Trust Center.	Compute Engine Management.
Storage Services	S3, EBS, EFS, Storage Gateway, Snowball, Snowball Edge, Snowmobile.	Blob Storage, File Storage, Disk Storage, Data Lake Store.	Cloud Storage, Persistent Disk, Transfer Appliance, Transfer Service.