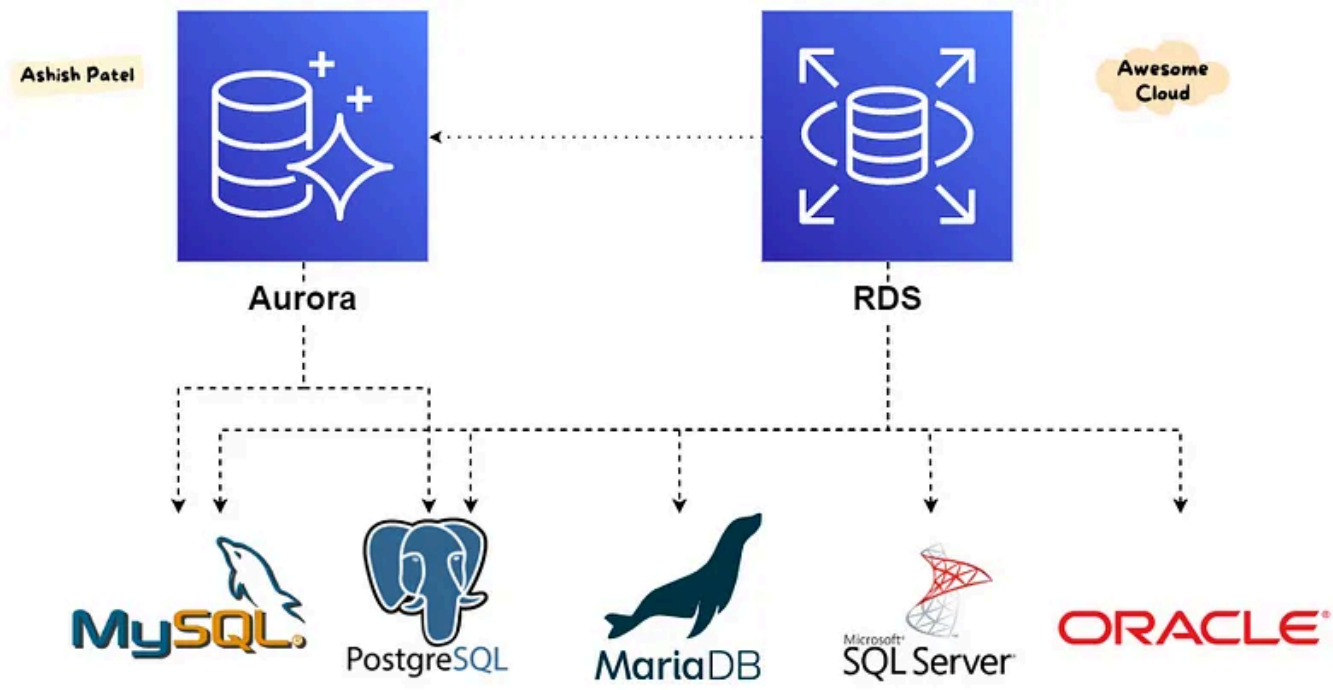
AWS — Difference between Amazon Aurora and Amazon RDS

Ashish Patel · Follow
Published in Awesome Cloud
7 min read · Feb 7, 2022

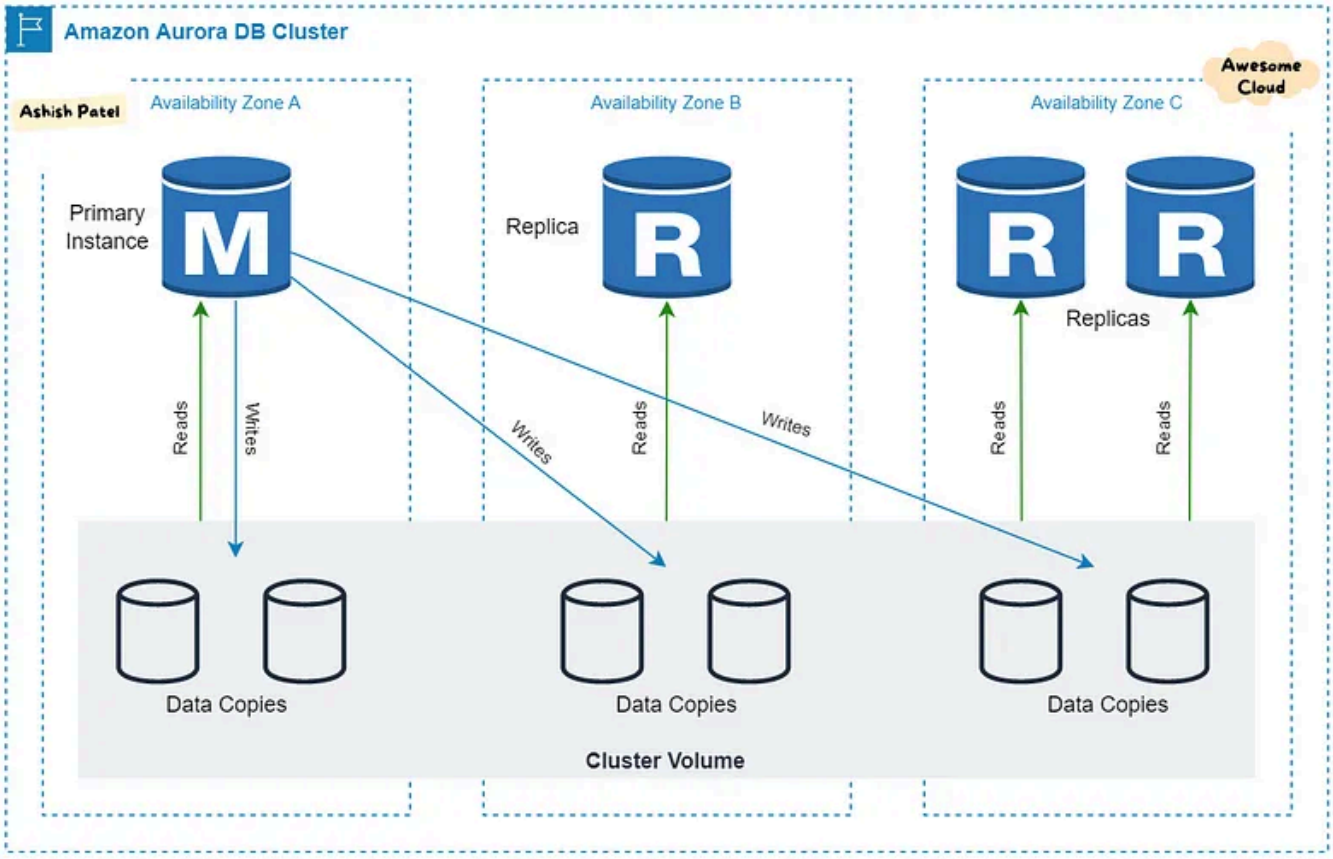
Comparison: Amazon Aurora vs Amazon RDS.



Awesome Cloud — Amazon Aurora vs Amazon RDS

Amazon Aurora

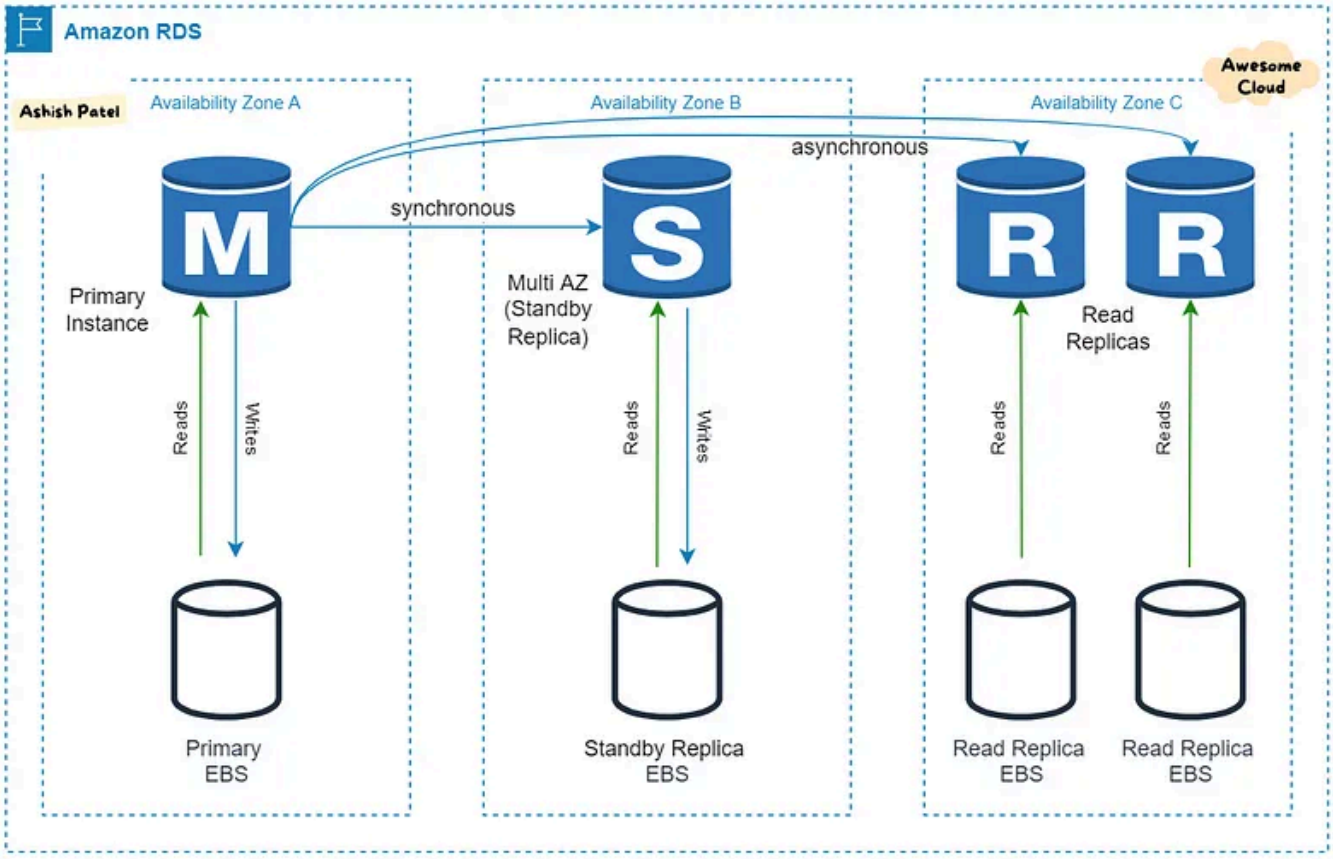
Amazon Aurora is a fully managed MySQL- and PostgreSQL-compatible relational database built for the cloud that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases.



Amazon Aurora

Amazon RDS

Amazon RDS (Relational Database Service) is a managed SQL database service a relational database in cloud which makes it easy to provision, setup, patching, and backups. It supports Aurora, MySQL, PostgreSQL, MariaDB, Microsoft SQL Server, and Oracle database engines.



Amazon RDS

Key Differences: Amazon Aurora vs Amazon RDS

Architecture Design

RDS architecture is similar to installing database engine on Amazon EC2 manually but leaving the provisioning and maintenance to AWS. RDS provides many features like automatic failover, backups, etc. RDS use Amazon EBS volumes for database and log storage. To achieve reliability, you need to enable the Multi-AZ feature on your RDS instance and replicate it synchronously to a standby replica in another Availability Zone.

Aurora database storage is reliable and fault-tolerant by design. Aurora's database storage is separate from the instances. In Aurora, data has 6 copies (as 10GB chunks distributed) to three Availability Zones. Hence, even if you have only one Aurora instance, your data will still have 6 copies.

Performance

RDS uses SSDs storage for better I/O throughput performance. You can choose between two SSD-backed storage options, that is, one that is optimized for high-

performance OLTP applications and another one for cost-effective general-purpose use.

Aurora gives two times throughput performance provided by PostgreSQL or five times the throughput provided by standard MySQL running on similar hardware.

Aurora's performance is higher and more consistent. Aurora writes logs directly to the storage without keeping log buffers. The replication to the replicas is asynchronous and for only cached data. Because the replicas also share the same storage cluster, the replica lag is small and consistent over time. Due to its unique storage design, Aurora's performance stays consistent when the load increases.

Database Engine Support

RDS is compatible with MySQL, PostgreSQL, MariaDB, Microsoft SQL Server, and Oracle.

Aurora is compatible with PostgreSQL and MySQL.

It means you can run your existing database tools and applications on both without any modifications.

Availability and Durability

Aurora offers higher availability and better durability than RDS, due to its unique storage model, and ability to perform continuous backups and restore with a very low RPO (recovery point objective).

In Aurora, data is durable by design. You always have multiple copies of your data. Every Aurora cluster has six storage nodes, spread across three AZs, even if you have just one compute node. In RDS, you have to max out your read replicas for this level of durability.

Resiliency

Aurora is more resilient than RDS because of its architecture design. It has fast recovery from failures. If a compute node crashes, Aurora can recover quickly. It can start new read replicas with minimal lag, and if the writer fails, another replica can be promoted to take over without waiting for the other nodes to reach consensus. All the shared state is in the data nodes, so failed nodes can be replaced almost immediately.

Storage

RDS storage auto scaling automatically scales storage capacity up to 64 TiB (except SQL Server's 16 TiB) in response to growing database workloads, with zero downtime. With RDS Storage Auto Scaling, you simply set your desired maximum storage limit, and Auto Scaling takes care of the rest.

Aurora automatically increases storage from a minimum of 10 GB to a maximum of 128 TiB. This is done in increments of 10 GB without any impact on the database performance. You are not required to provide the storage in advance.

Scalability

Vertical Scaling: Both, Aurora and RDS, allow you to scale the memory and compute resources up and down, to a maximum of 244 GiB of RAM and 32 vCPUs. Scaling operations can be done within a few clicks.

Dynamic Scaling: *Aurora Auto Scaling* dynamically adjusts the number of Aurora Replicas provisioned for an Aurora DB cluster using single-master replication. It enables your Aurora DB cluster to handle sudden increases in connectivity or workload. When the connectivity or workload decreases, It removes unnecessary Aurora Replicas, so that you don't pay for unused provisioned DB instances.
RDS does not support such Auto Scaling.

Replication

RDS allows you to provision up to 5 replicas, and the process of replication is slower compared to Aurora.

Aurora allows you to provision up to 15 replicas, and the replication is done in milliseconds.

Aurora scales faster because it can add new read replicas quickly. Because replicas all use the shared storage volume, a new replica can serve queries almost immediately. It doesn't have to wait to replicate data from the other nodes. Aurora does some asynchronous cache replication between nodes, but nothing synchronous. This reduces the inter-node I/O, which means Aurora can have more replicas.

Failover

In RDS, Failover to read replica is done manually, which could lead to data loss. You can use Multi-AZ (Standby instance) feature for automatic failover, and to prevent downtime and data loss.

In Aurora, Failover to read replica is done automatically to prevent data loss. Failover time is faster on Aurora.

Cluster Endpoints

In RDS, there is a cluster endpoint which you use for your write queries. It is the DNS endpoint pointing to your current master db instance. During a failover, RDS routes this endpoint to the new master by a simple DNS change. However, for read replicas, you have to balance the load in your application using the instance endpoints. RDS does not provide a load balancer for read replicas.

In Aurora, you still use cluster endpoint for your write queries. It also provides a reader endpoint acting as a load balancer for your read replicas. So you can use this endpoint for your read queries. In the case of a failover, one of the read replicas become master and is removed from this reader set.

Backup

RDS creates and saves automated backups of your DB instance during the backup window of your DB instance. RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases according to the backup retention period that you specify. If necessary, you can recover your database to any point in time during the backup retention period. While the snapshot is being taken, storage I/O may be interrupted while data is copied, affecting database performance.

Aurora backs up your cluster volume automatically and retains restore data for the length of the backup retention period. Aurora backups are continuous and incremental so you can quickly restore to any point within the backup retention period. No performance impact or interruption of database service occurs as backup data is being written.

For both, Backups are stored in Amazon S3.

Pricing

Aurora is more expensive than RDS for the same workloads. Aurora pricing is mainly based on instance size and storage is billed according to actual usage.

Amazon Aurora Features

Aurora Serverless

Aurora Serverless lets you run Aurora without having to guess how many compute nodes you need. It automatically starts and stops nodes to match the needs of your application. It scales up to meet a spike in demand and scales down when things are quiet. The data remains in the shared storage volume, independent of any scaling.

Aurora Multi-Master

In a multi-master cluster, all DB instances have read/write capability. The notions of a single read/write primary instance and multiple read-only Aurora Replicas don't apply. There isn't any failover when a writer DB instance becomes unavailable, because another writer DB instance is immediately available to take over the work of the failed instance.

Aurora Global Database

Aurora Global Database is designed for globally distributed applications, allowing a single Amazon Aurora database to span multiple AWS regions. It replicates your data with no impact on database performance, enables fast local reads with low latency in each region, and provides disaster recovery from region-wide outages.

Other features

- Aurora Parallel Query.
- Aurora DB Cloning.
- Aurora DB Backtrack.
- Aurora DB Activity Monitoring.

Summary

Aurora's unique architecture gives you more durability, scalability, resiliency, and performance when compared to RDS. Although there is a small increase in cost, it is recommend using Aurora for enterprise-level applications. If you are looking for a native high availability solution and/or read-intensive workload, then Aurora is a perfect match.

View more from *Awesome Cloud*

- Difference between SQS and SNS
- Difference between Application load balancer and Network load balancer
- Difference between Security Groups and NACL
- Difference between Internet Gateway and NAT Gateway
- Difference between Secrets Manager and Parameter Store
- Difference between EKS and ECS

Happy Clouding!!!

AWS | Amazon Rds | Amazon Aurora | Aurora | Rds

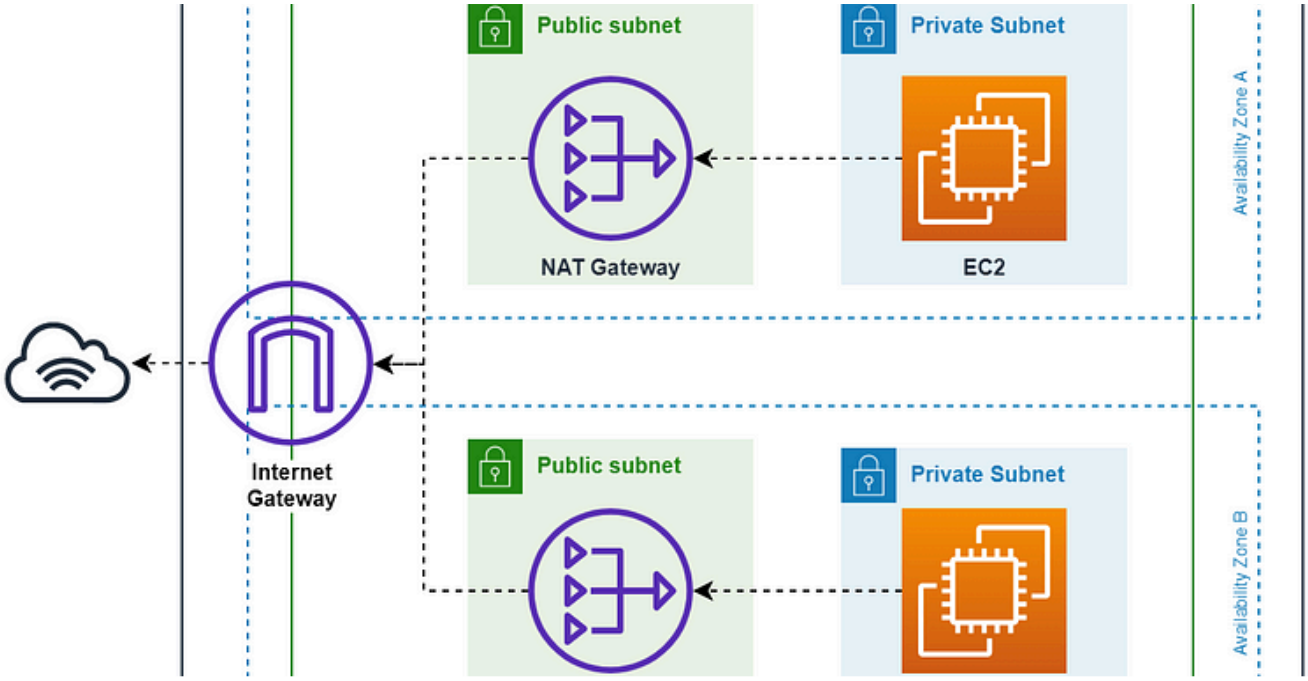
Follow

Written by Ashish Patel

18.9K Followers · Editor for Awesome Cloud

Cloud Architect • 4x AWS Certified • 6x Azure Certified • 1x Kubernetes Certified • MCP • .NET • Terraform • DevOps • Blogger [https://bit.ly/iamashishpatel]

More from Ashish Patel and Awesome Cloud



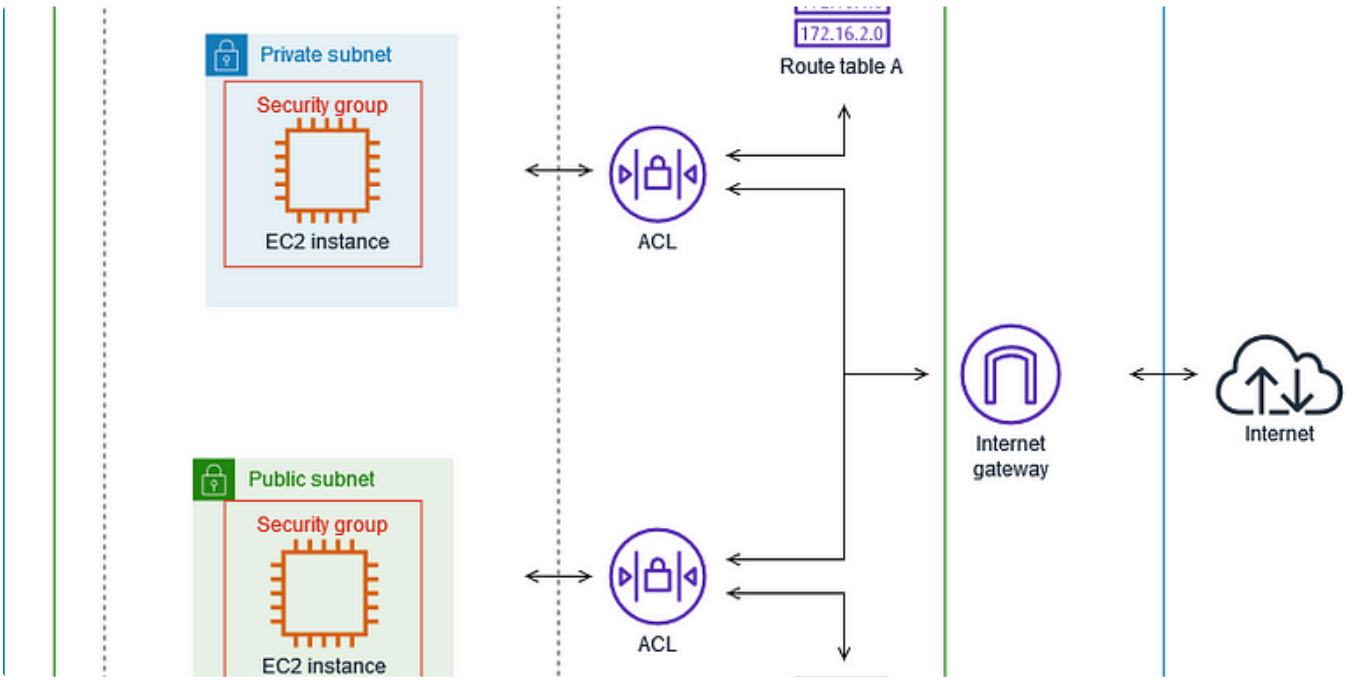
Ashish Patel in Awesome Cloud

AWS—Difference between Internet gateway and NAT gateway

Internet gateway vs NAT gateway in AWS

2 min read · May 25, 2019

1.2K 12



Ashish Patel in Awesome Cloud

AWS—Difference between Security Groups and Network Access Control List (NACL)

Security Group vs NACL in AWS

2 min read · Apr 14, 2019

1.6K 21

Awesome Cloud Ashish Patel

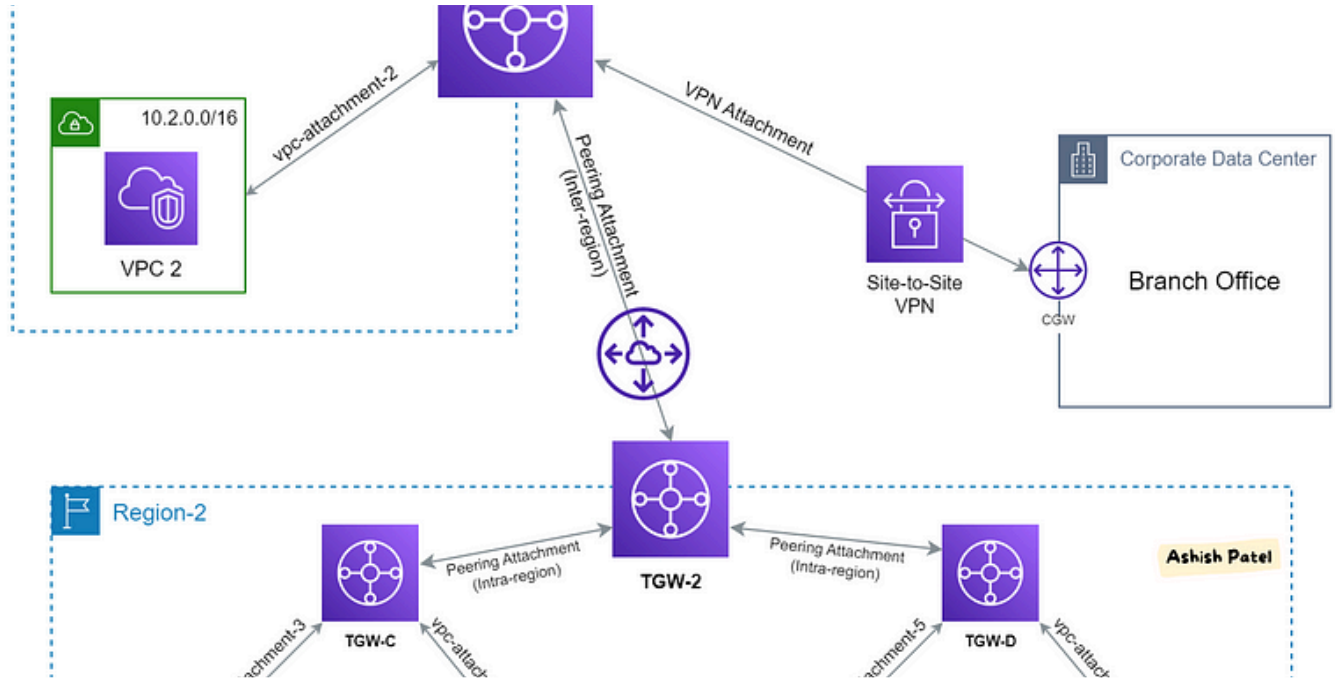
Ashish Patel in Awesome Cloud

AWS—Difference between Amazon EventBridge and Amazon SNS

Comparisons: AWS EventBridge vs AWS SNS.

5 min read · Mar 29, 2022

499 3



Ashish Patel in Awesome Cloud

AWS—Difference between VPC Peering and Transit Gateway

Comparisons: AWS VPC Peering vs AWS Transit Gateway in AWS

4 min read · Jan 8, 2023

270 3

See all from Ashish Patel

See all from Awesome Cloud

Recommended from Medium

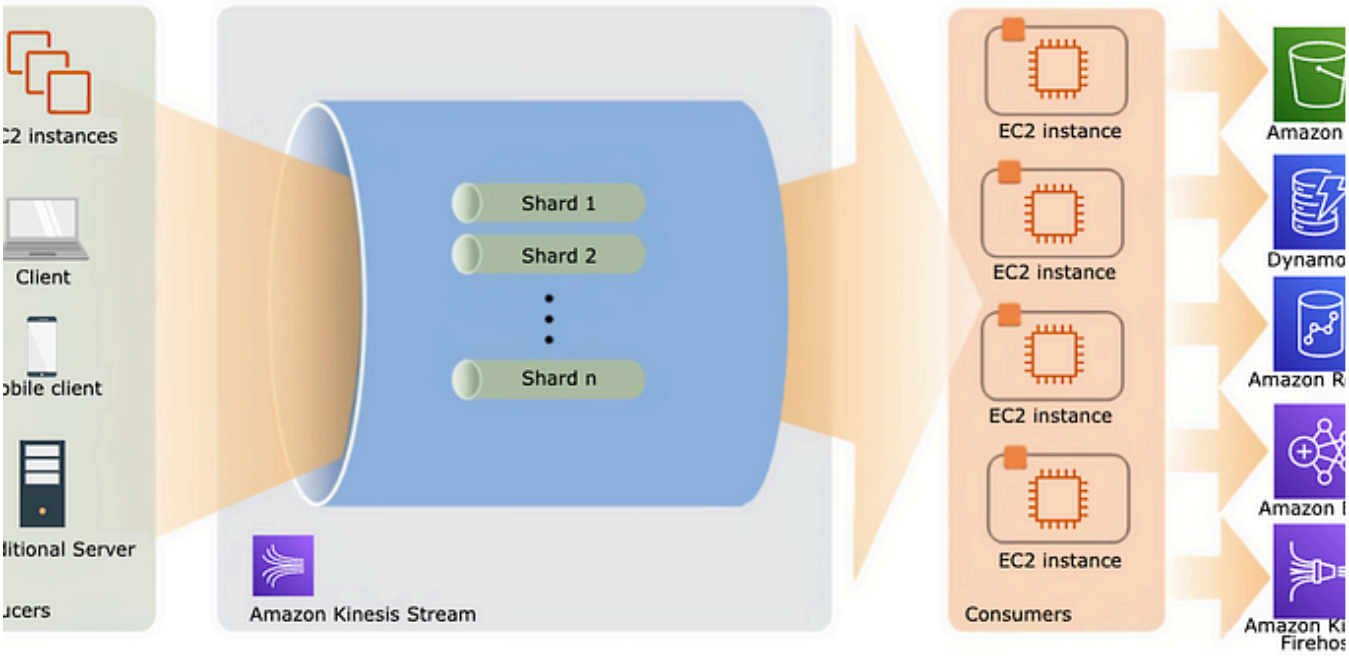
Pankaj Makhijani in AWS Tip

Implementing Secure Access to RDS with Private Subnet, RDS Proxy, and SSH Tunnel

Boost Amazon RDS security with private subnets, RDS Proxy, and SSH tunnels. Learn key strategies to safeguard your cloud database...

10 min read · Feb 12, 2024

68



DSL

Streamlining Real-Time Data Processing with AWS Kinesis, Lambda, and Terraform

So I have been preparing for the AWS solution architect exam and I came across this question.

10 min read · Feb 5, 2024

99

Lists

Natural Language Processing

1489 stories · 1001 saves

Faisal Kuzhan

DAY_25/90 => INTERVIEW QUESTIONS ON AWS

Below are some basic AWS interview questions along with the answers. ✍️

17 min read · Feb 12, 2024

91 3

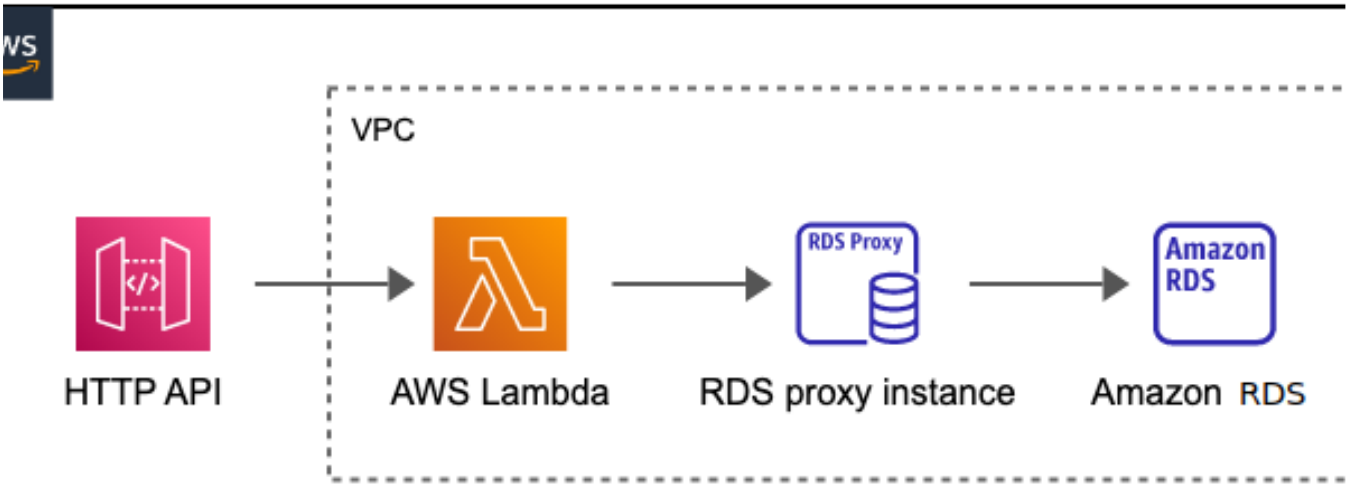
Neal Davis

Top 20 AWS Solutions Architect Interview Questions

Being well-prepared is crucial for any tech interview, especially when aiming for a specialized position in cloud computing. For those...

9 min read · Dec 11, 2023

312 6



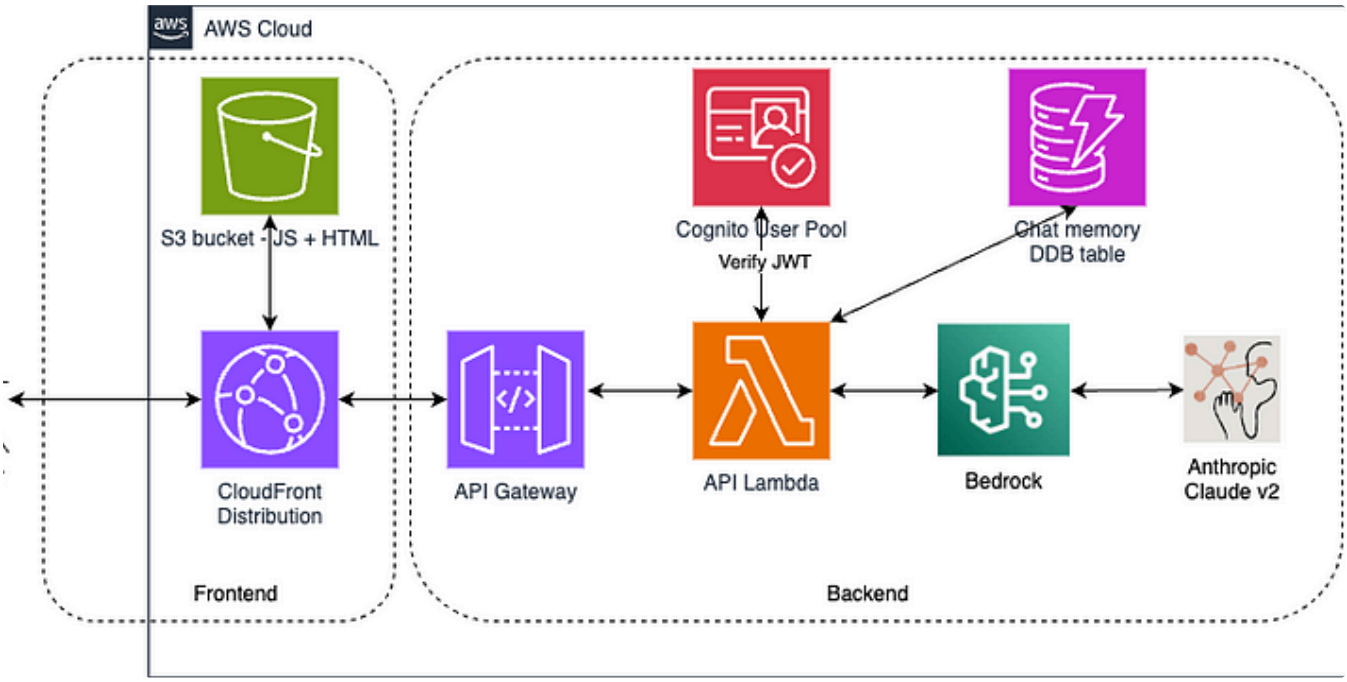
Amir Ayat

Securing AWS Lambda to AWS RDS Connection: A Guide Using VPC and Proxy

AWS Lambda, a serverless compute service, and AWS RDS, a managed relational database service, necessitate a robust security strategy to...

4 min read · Mar 1, 2024

8



Greg Biegel

Building a serverless chatbot with Amazon Bedrock

This post shows how easy it is to build a completely serverless chatbot web application with AWS services. Crucially, no data in your...

4 min read · Jan 11, 2024

190 2

See more recommendations