How (and when) to leverage the AWS backbone to deliver global services

27 August 2025 - 2 min. read

Amazon CloudFront | Amazon Route 53 | AWS backbone | AWS Global Accelerator | AWS Global Network | AWS WAF | Multi-region deployment

Introduction

The AWS backbone, or AWS Global Network, is Amazon's privately owned, high-capacity fiber-optic infrastructure that interlinks AWS Regions, Availability Zones, and Edge Locations worldwide. The AWS backbone controls routing and network performance, ensuring consistently low latency and high throughput between AWS services.

Let's see how leveraging this high-speed and low-latency network can be better than replicating your whole infrastructure to get closer to the end user.

AWS Edge Services

AWS edge services are cloud capabilities designed to bring computing, storage, and networking closer to the end user, rather than relying on centralized infrastructure in AWS Regions. Edge services aim to reduce latency by physically shortening the distance data must travel.

Some examples of edge services are:

- **Amazon CloudFront**: A global content delivery network (CDN) for distributing static and dynamic content with low latency by caching it at edge locations.

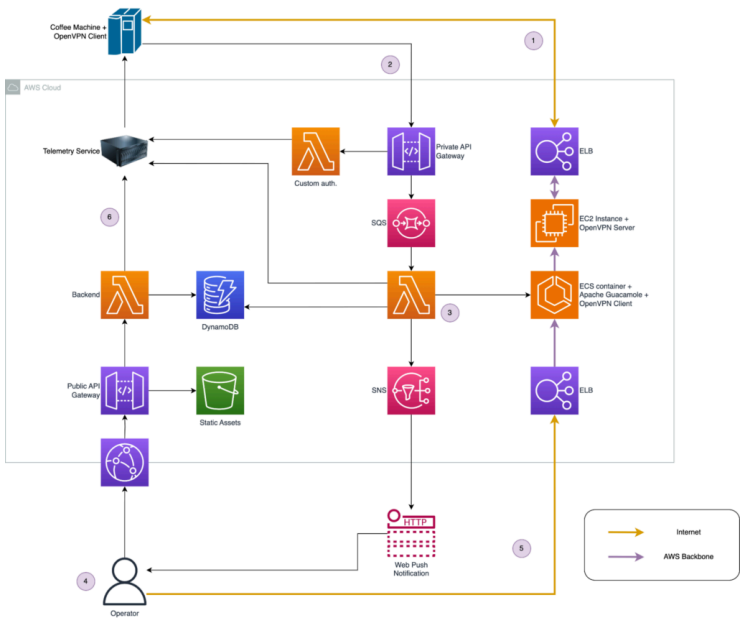
- **CloudFront Functions**: A serverless feature of CloudFront, CloudFront Functions are lightweight JavaScript functions that help with latency-sensitive CDN customizations.
- **AWS WAF:** A web application firewall that protects applications from common web exploits and bot attacks. When integrated with a CloudFront distribution, WAF rules are applied at all edge locations.
- **AWS Global Accelerator**: A networking service that optimizes traffic routing over the AWS backbone for improved performance.
- **Amazon Route 53:** A highly available and scalable cloud Domain Name System (DNS) that uses the network of edge locations to achieve low latency.
- **AWS Storage Gateway:** A hybrid cloud storage service that caches data locally for low-latency access

One of these services will be the key to our solution.

Use Case: Global Remote Desktop Service

The goal was to create a web application that would allow coffee vending machine operators to access the machines and provide remote support when problems arise. Usually, this was done through TeamViewer, but new machine models stopped supporting it, so a custom solution based on the Remote Desktop Protocol (RDP) was required.

We decided to leverage Apache Guacamole, a clientless remote desktop gateway that supports RDP.



When an operator installs a new machine, they register it through the web app, which registers it into the OpenVPN server and sends the VPN profile to the telemetry service.

The telemetry service is a web app that leverages IoT Core and runs in the same AWS account as the remote desktop solution. It manages and provides information and functionalities about coffee machines.

If a user encounters a problem with a coffee machine, they can press a button to send a support request, and the following flow starts:

1. The machine connects to the VPN server and starts the RDP flow
2. Thanks to the VPN channel, the machine can contact a private API Gateway to provide all information required for the RDP connection: its IP address, a user, and a password.
3. A Lambda function performs the following tasks:
   - It starts an ECS task from an AMI where Apache Guacamole and the OpenVPN client are installed. Guacamole is publicly exposed and protected by a password
   - It retrieves from the telemetry service which operator should be notified
   - It sends a notification to the operator through the web app, leveraging SNS and Web Push Notifications.
4. The operator asks the web app to start the RDP connection to the coffee machine.
5. The operator connects to the machine through RDP and performs support operations.
6. The operator closes the VPN and RDP connections through the web app when the support tasks are terminated.

The Single-Region Limitation

The whole application was deployed in the Ireland Region (eu-west-1), in the same region as the telemetry application.

Unfortunately, the RDP connection encountered latency problems. The machines are scattered worldwide; some places have slow or unstable internet connections. This

situation made the RDS connection barely usable.

A solution was needed to reduce the latency between the machine, the operator, and the Guacamole instance.

Multi-Region Deployment vs. Leveraging the Backbone

Multi-Region Deployment

Multi-region architecture involves deploying workloads in multiple AWS Regions, depending on user distribution. This approach ensures that most users connect to a Region geographically close to them, minimizing latency.

This kind of setup has some trade-offs:

- **Data synchronization**: Replication across regions introduces challenges in consistency. You must choose between strong consistency (with added latency) and eventual consistency (with the potential for brief data mismatches).
- **Operational overhead**: Every new region deployed is essentially an environment to maintain and monitor.
- **Cost implications**: Operating multiple full-scale deployments can increase infrastructure and data transfer costs, particularly for cross-region replication.

Let’s check these points for our solution.

Data synchronization

Data synchronization is not a problem. Most of the services utilized are serverless and stateless. The only data synchronization required is the data saved on DynamoDB, which can be achieved easily with DynamoDB Global Tables.

Amazon DynamoDB Global Tables is a fully managed, serverless, multi-region, and multi-active database. Global Tables replicate automatically across chosen Regions to achieve fast, local read and write performance.

Operational overhead

As mentioned in the previous point, this is a minor problem due to using many serverless services that reduce operational complexity, so nearly no maintenance is required.

Cost implications

Cost implications are the real problem for our use case.

It could sound strange, since I already said that most of the infrastructure relies on serverless services, so you should pay per request, and the number of resources deployed should not matter.

DynamoDB Global Tables require you to pay N times (where N is the number of Regions in which you replicated your infrastructure) the cost for data at rest, the number of inserts and updates that you should pay for a single table, since every data and write operation is replicated in the chosen Regions. Also, you will pay the inter-region traffic required for the synchronization. In our use case, the amount of data is not so large as to make this an unbearable cost.

The real problem was the replication of the EC2 instance with the OpenVPN server installed.

The EC2 instance required a different OpenVPN server license for every AWS region, which was much higher than the desired cost.

But we knew a way to design a more affordable solution.

Leveraging the Backbone

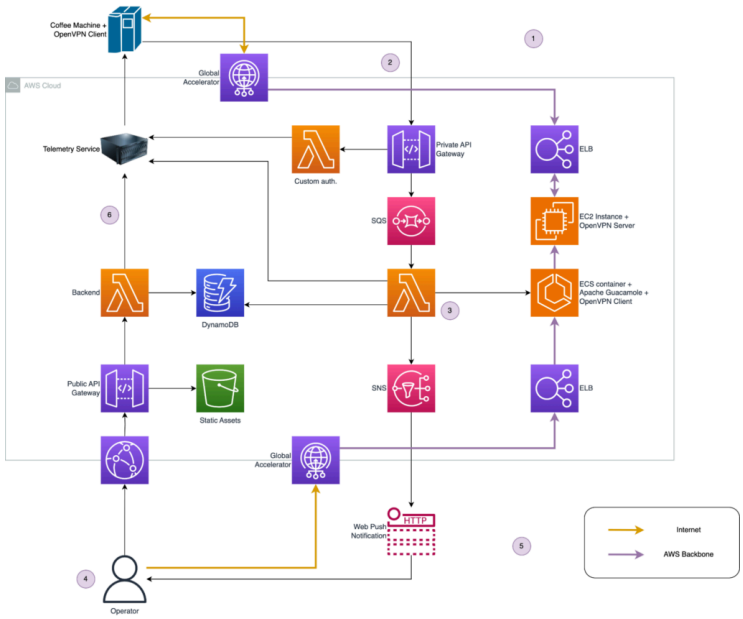
This solution aims to make the most of the AWS backbone by leveraging AWS Edge Services.

In our use case, the game changer is the introduction of Global Accelerator. Thanks to Global Accelerator, we can deploy the infrastructure in only one Region and reduce the latency enough to make the RDP connection smooth as if every user is in the same Region of the infrastructure.

This is possible because the user and machine connections travel across the Internet only to reach the nearest AWS Edge Location instead of traveling across the Internet from their location to the Ireland Region.

Once the Edge Location is reached, the AWS Backbone handles the traffic, providing high speed and low latency to the RDP connection.

Thanks to Global Accelerator, we achieved a solution with performance nearly similar to that of a multi-region solution, but costs slightly higher than that of a single-region solution.



Conclusion

When designing a global solution, the default recommendation is often to build a multi-region architecture for the following reasons:

- AWS resources are closer to end users, providing the minimum latency achievable
- It helps provide high availability and disaster recovery in case of a regional outage

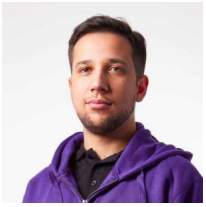
However, we saw that sometimes multi-region solutions are not the best choice, especially when costs must be low and licenses or costly services must be replicated for every region.

Sometimes, using backbone-powered services like Global Accelerator is a good strategy to obtain performance very close to a full multi-region deployment while maintaining low complexity and costs.

About Proud2beCloud

Proud2beCloud is a blog by beSharp, an Italian APN Premier Consulting Partner expert in designing, implementing, and managing complex Cloud infrastructures and advanced services on AWS. Before being writers, we are Cloud Experts working daily with AWS services since 2007. We are hungry readers, innovative builders, and gem-seekers. On Proud2beCloud, we regularly share our best AWS pro tips, configuration

insights, in-depth news, tips&tricks, how-tos, and many other resources. Take part in the discussion!



Daniele Papa

DevOps Engineer and backend developer @ beSharp. I like playing video games and board games in my free time. In the last few years, I approached the Cloud environment, and now I switch between IAM roles and role-play games.
