AWS Architecture Design Exercise

Prepared by Sean Meadows

Overview

Tech Company Inc. would like architectural guidance on moving their application, PeopleApp, into AWS. The application currently resides on-premise in Tech Company's corporate datacenter. Moving it into AWS will ensure it is scalable, elastic, on redundant architecture, and can grow organically with the company. This document will recommend AWS resources that are in line with key requirements and address concerns that Tech Company Inc. has with moving into AWS.

Recommended Architecture

Based on the key requirements and concerns for the PeopleApp application presented by the Tech Company Inc. team, we recommend the following AWS architecture for the web application. This document will outline a solution for each requirement/concern and detail the services that will meet the challenges.

Scalability and Effective Distribution of Load

To meet the requirements for scaling and load distribution, the presentation and application tiers can be served via EC2 instances running in AWS. The frontend auto scaling group would be served by the main application load balancer, while the backend auto scaling group would be served by an internal load balancer. The EC2 instances will be configured with Auto Scaling to scale-out across multiple Availability Zones as demand on the application grows (assuming that the application frontend and backend configurations are able to be served via Windows or Linux EC2 instances). The data tier can be served by RDS instances coupled with read replicas in the main AZ for scalability (assuming a read-heavy database workload, also assuming if using Microsoft SQL server, the databases are not hindered by the read replica limitations), while maintaining a multi-AZ copy of the databases for high availability (assuming that the application uses a traditional relational database that is able to be moved in RDS).

Lack of Disaster Recovery and Self-Healing Architecture

The nature of the new application architecture in AWS will allow for failed instances, or the highly unlikely event of an entire failed AZ. This is because the configuration of all three application tiers will see the resources in a multi-AZ configuration. The presentation and application tier resources will ideally be stateless, and in the event of a failure of a single (or multiple) instances, traffic will automatically cease to be routed to the unresponsive instances and they will be replaced, thus healing themselves. This is achieved by a combination of health checks, auto scaling, and load balancing within AWS. For the data tier, RDS would also be configured in a multi-AZ deployment, so the use of automatic failover would allow for failover to an up-to-date standby in the event of a failure. Additionally, Lambda can be leveraged to ensure that the application could recover from the highly unlikely event of an entire failed Region. This can be accomplished by Lambda functions that create EC2 and RDS snapshots at regular intervals. One function creates the snapshots, and another function copies these snapshots to a DR region. Further Lambda functions can be used to clean-up old or outdated snapshots in both the production and DR regions to reduce storage costs. Coupling this with a pilot-light or warm-standby configuration for application components in the DR region would allow for rapid recovery.

Shared Content Storage, Performance Demands, and Content Latency

To meet the needs for content storage for the application in AWS, the use of Amazons Elastic File System is recommended. EFS provides storage via a fully managed file system that can be mounted to the multiple EC2 instances in the new environment. EFS can scale to thousands of concurrent connections while maintaining low latency. EFS stores files, directories, and links redundantly across multiple AZs, and scales size automatically, so there is no need to set capacity. AWS Backup can be used with EFS to ensure that content stored within EFS is protected as well.

Data Security

Data, both at rest and in-transit, needs to be as secure as possible in the new AWS environment. This is achieved multiple ways on the different tiers. The presentation tier EC2 instances will reside in a private subnet in the VPC and can only be reached through the application load balancer. The backend and data tier instances will likewise be in private subnets and not internet facing. All the EC2 instances will be configured to have their root volumes (and any additional volumes as needed) encrypted. EFS will also be encrypted when creating the file system, and encryption in transit will be enabled when mounting it. RDS instances will also have encryption enabled to protect data. As encryption will be enabled for the EC2 and RDS resources, all snapshots created from these resources will also be encrypted.

Managing Cost for Ingestion of Customer Updates

As AWS Direct Connect will be recommended for connection to the existing datacenter (as outlined below), Direct Connect can be leveraged to help in managing costs for bulk updates of customer information. Direct Connect provides predictable network performance and has no cost for data transfer IN (cost is $0.00 per GB in all locations). This would allow for more predictable modelling of costs for bulk RDS updates over Direct Connect (assuming that “increased customer updates on AWS” only refers to bulk updates of RDS data via batch executes in RDS).

AWS Integration with Existing Datacenter and Management Systems

Connecting the existing datacenter to AWS can be achieved in two ways, either via Direct Connect or via an AWS-managed VPN. For this solution and to meet the data security needs, Direct Connect is recommended (assuming that the client datacenter is in an area where they can run fiber to an AWS Direct Connect location). Direct Connect bypasses the public internet and provides a secure, dedicated, ultra-low latency connection into AWS. Direct Connect provides more predictable network performance and cost savings. Existing management systems can be used in conjunction with new AWS resources as well. Network monitoring tools can be used to monitor traffic related to the new cloud application, and server and infrastructure tools can be used in conjunction with available built-in AWS tools to monitor EC2 instances. Additional tools can be installed and configured on the new EC2 instances (assuming these tools work with Windows/Linux) for server monitoring or remote access. Some existing security tools can be used with AWS resources, but other advanced security testing (certain types of penetration testing, certain simulated events, and certain stress testing) may require authorization from AWS prior to testing.

Architecture Diagram