Payment Gateway Migration

Modernizing one of the world's leading payment processors with AWS

Candid provided a team to facilitate the migration of our client's core payment card processing platform to AWS, enhancing operations, providing MRAA high availability, and preparing them for an upcoming strategic merger

THE PROBLEM

A payment technology and processing provider in the consumer finance, automotive, healthcare, and accounts receivable management industries based in the Southwest U.S. and headquartered in Atlanta, reached out to Candid with an interesting dilemma.

This corporation was already benefiting from public cloud, but they were leveraging a single database that provided neither the technical capabilities nor user experience desired. Due to the rapid growth they were experiencing, and an upcoming merger with another payment processor, a modern Payment Gateway solution with increased scalability was required for continued success excelling.

The client was utilizing a legacy solution, that despite working properly for the current need, did not offer scalability and was not optimized for the future. The solution did not offer any flexibility when it came to Disaster Recovery or High Availability. And if the system went down, they would lose revenue and even clients of their own.

THE CANDID SOLUTION

The Candid team proposed migrating the entire Payment Gateway solution to AWS, to leverage Multi-Region Active-Active (MRAA) and expand business globally, to embrace faster DevOps execution, to replicate key tables in the SQL server database, to attain a greater level of security and to reduce risk in the delivery cycle. With a request for zero downtime during the migration, we knew this would be an interesting project that could take some time, but were able to complete it in only five months; with MRAA remaining on hold pending a beta program launch.

Candid reviewed and finalized a plan for the future state Payment Gateway architecture leveraging AWS. Beginning with building and testing the architecture in AWS, Candid split the sandboz out from production, and began the migration, establishing Active/Active across two regions with AWS Auto Scaling. We also created a lower environment testing database from an existing production database and validated that the solution would not only meet but exceed acceptance criteria.

Due to the business urgency with the upcoming merger, it was vital for Candid to ensure a seamless deployment. The team utilized canary deployment technique to safely make changes. A canary deployment creates a secondary version of application components with the new changes that a small subset of the workloads is directed to, enabling a controlled incremental release of new functionality. If the changes work as expected, the remainder is rolled onto the new cloud platform. If however, the change fails or creates unanticipated issues, the release can quickly be rolled back in an automated process without impacting a significant portion of the migration process.

Utilizing Matter, Candid's cloud automation platform partner to generate terraform enabled us to accelerate timelines and provide success more quickly than would traditionally be possible.

EC2

The solution Candid built implements AWS Auto Scaling in the payment gateway. Traffic comes in from merchants through Amazon CloudFront, and routes to one of two types of Windows servers based on the hostname. These two types are in an auto-scaling group to elastically respond to customer traffic.

MICROSOFT

A third Windows instance type (that doesn't auto-scale) runs periodic jobs and reports against an Amazon Relational Database Service (RDS) instance running SQL Server to settle transactions. We use SQL Server RDS, internal to the Windows servers they use Microsoft IIS.

AWS

The solution, as mentioned, uses RDS. It additionally leverages AWS Auto Scaling, AWS Elastic Load Balancing, AWS Lambda, Amazon Route 53, AWS System Manager Agent (SSM), Amazon Elasticsearch Service, Amazon ElastiCache, Amazon CloudWatch, Amazon CloudFront, Amazon DynamoDB, AWS WAF, and AWS Shield Advanced.

NETWORKING & SECURITY

We have Amazon Virtual Private Cloud (VPC) in U.S.-West-2 peered with a VPC in U.S.-East-1. Each VPC is the same, with public and application subnets. The application subnets in each VPC can communicate.

We solely used security groups to restrict network access to the Amazon EC2 M5 m5.2xlarge instances. This approach was chosen for the general-purpose nature of the client workload that was roughly analogous to their Azure-based instance capabilities. Passwords, shared secrets and VPN keys are kept in SSM parameter store as secrets. These are set on the instances at boot time.

ENCRYPTION

For encryption, log files are captures from the Amzon EC2 instances and shipped to Amazon Elasticsearch. Sentry.io is used to identify exceptions. AWS Lambda is monitored through Amazon CloudWatch logs and metrics.

All network traffic is Transport Layer Security (TLS) or on AWS Site-to-Site VPN (VPN) to backend providers. The EC2 instances are stateless, and do not store data on them, and the RDS uses encrypted storage.

Remote server access is implemented, managed, and restricted through a Windows Bastion host that connects through Remote Desktop Protocol. It is accessible only after connecting to a VPN per environment; SSM can also be used. Authentication to the Windows Bastion via RDP is handled through AWS Managed Microsoft AD.

MANAGEMENT

Our client manages their own Amazon EC2 instances, and are license-included. Pricing models are all on-demand at this time.

PERFORMANCE CONSIDERATIONS, MONITORING

We started with the Azure analog, and utilized Sentry.io and Stackify to monitor both the application and the instuances. We've tuned the Auto-Scaling parameters to keep the fleet CPU at around 40%. Alarms roll up to pager duty to notify an on-call technician to look at both infrastructure and application errors.

ACTIVE DIRECTORY

AWS Managed Microsoft AD is used for authentication to the Windows Bastion. A primary active directory exists in an account outside of our control, and there are shared instances set up in each VPC.

DISASTER RECOVERY

The system is designed to connect intelligent systems through MRAA and is currently running in one region. In the event of a failure, we point the DNS Records at the standby region and manually activate the settlement EC2 there. After MRAA goes live, health checks will cause the system to automatically do a full fail over to the other region.

OUTCOMES

Our client gained dramatic scalability and security from this project. Today we are further eliminating legacy technical debt to aid their transformation into the world leader in payment processing.

- *Zero downtime migration between public clouds*
- *Multi-region, Active-active*
- *Faster DevOps execution*
- *Increased security and reduced risk in the delivery cycle.*
- *Lower Operational Overhead*
- *More actionable metrics for the overall business*
- *Greater degree of standardization*

CANDID

Contact Candid to find out how we can put advice into action for you.

404.815.4599 | INFO@CANDID.CLOUD

Revised: October 9, 2019 10:29 AM)