Business Critical Application Development with Serverless Technology

CANDID

PROTECTING CUSTOMER DATA

In early 2018, a national bank proactively sought to increase the security of private data for more than 1 million customer records. The internal team designed a portal to communicate with customers, and to direct them to a credit protection app. But, there were reliability concerns with the portal, and the bank needed to quickly adjust. To remediate, they brought in Candid who helped re-architect, rebuild, and release a new solution within 20 hours, to meet a critical deadline for hundreds of thousands of users. In this case study, we'll examine the four key steps Candid took to quickly and effectively overhaul the application.

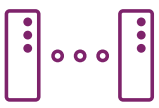
STEP 1: SERVERLESS

The first app, written for Amazon EC2, wasn't able to adapt to changing requirements quickly enough. To provide the scalability required for high-transaction volumes, Candid proposed a serverless option leveraging AWS Lambda.

Serverless would provide the least amount of development resources, at less cost while delivering a better level of security. With the change to serverless, Candid started over on the app development with all new infrastructure-as-code, as well as adapting the existing app code to work with AWS Lambda.

MULTI-REGION SOLUTION

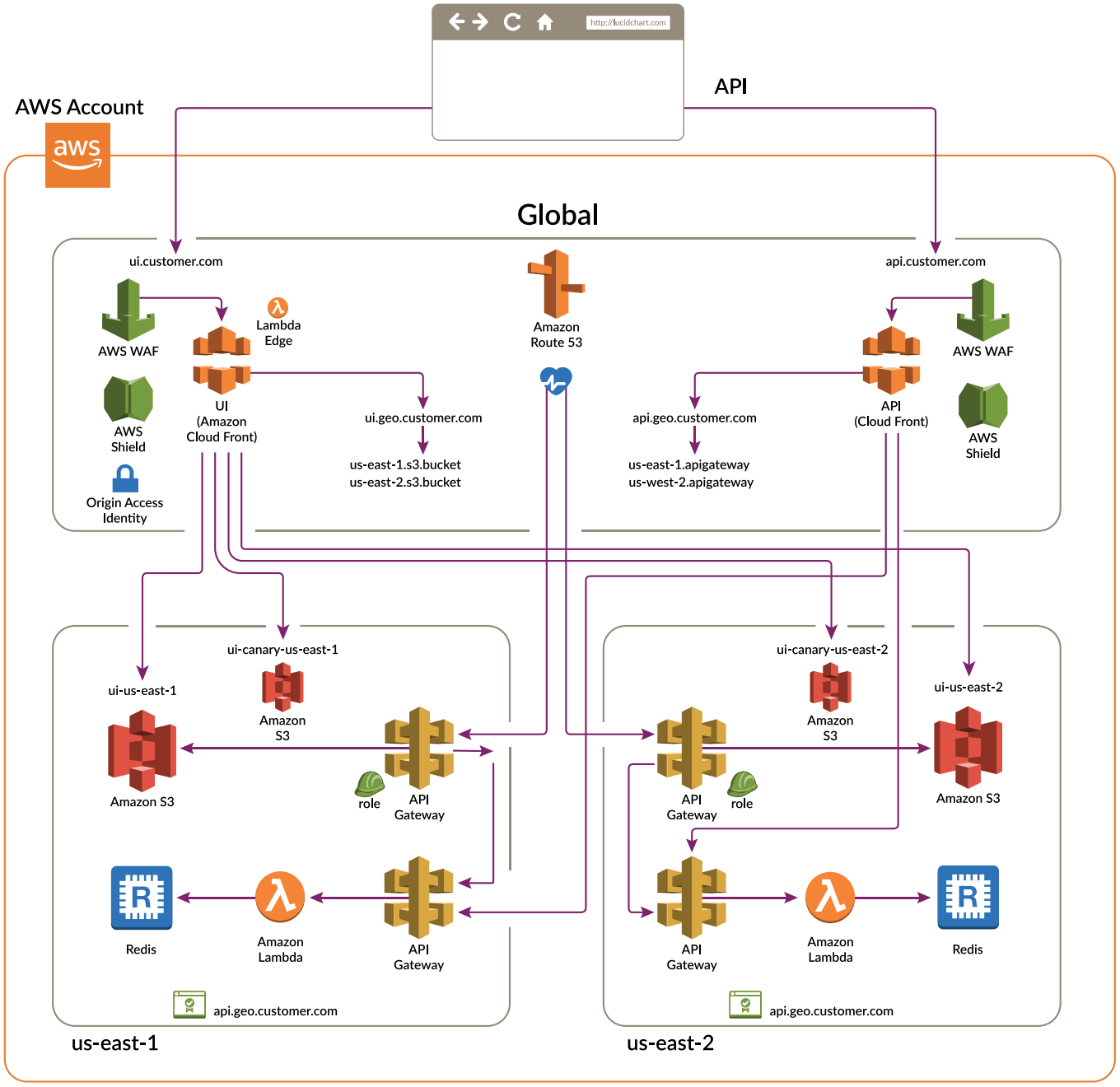
CANARY IMPLEMENTATION



CONTINUOUS IMPLEMENTATION

STEP 2: CONTINUOUS IMPLEMENTATION / CONTINUOUS DELIVERY VIA GOCD

For any large application overhaul like this, Candid uses a process called Continuous Implementation / Continuous Delivery (CI/CD), which allows for rapid, structured updates. Rather than wait for major milestones to be updated within the changing application, CI/CD frequently compiles and pushes updates to constantly improve. Using continuous deployments, gradual changes to the application have little-to-no impact on users and allows for more robust QA and error identification. Manually running a CI/CD process is inefficient, so Candid uses a tool called GoCD to automatically push updates when changes are implemented. By implementing CI/CD, Candid was able to create 3-4 releases a week, allowing rapid bug fixes and improving the application.



STEP 3: CANARY DEPLOYMENT

Due to the business urgency of the application, and the large user base, it was vital for Candid to ensure a seamless deployment. The team utilized a technique known as canary deployments to safely make changes. A canary deployment creates a secondary version of application components with the new changes that a small subset of the user base is directed to, enabling a controlled incremental release of new functionality. If the changes work as expected, the remainder of the user base is rolled onto the new application. 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 user base.

Candid utilized canary deployments on both the user-facing application and the back-end services to achieve a zero-impact deployment.

STEP 4: MULTI-REGION ACTIVE/ACTIVE SOLUTION

With customer communications about the new credit protection capabilities, the organization anticipated high volume on the application as users checked their credit and identity. One of the key goals for Candid, then, was to minimize downtime and maximize reliability. To achieve this, the team pursued a multi-region active/active solution in Ohio and Virginia. By distributing load across multiple AWS systems and geographical regions, they reduced the risk of a comprehensive failure. If the UI or application services in Ohio do not operate properly for any reason, the traffic is quickly routed to the Virginia region. Further, the team implemented latency-based routing – connecting the user to the region that could most quickly resolve their request, increasing efficiency and user satisfaction.

To ensure application health, Candid implemented regular health checks on each region's UI services. Route 53 had DNS entries configured for the UI and service components for each region, and each DNS entry is attached to a health check for that region. All of this is done in a secure manner, over the internet, without exposing any customer data through the health checks. If either piece of the application fails three consecutive checks, each 10 seconds apart, Route 53 will fail over the DNS to the healthy region and re-route traffic to it. This ensured that even if specific AWS regions went down, the organization's application could remain up.

LOOKING FORWARD

The revamped application launched successfully, and the organization avoided further issues. Impressed by the team, the company requested Candid assistance for three cloud migrations later in 2018.

CANDID

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

Candid
817 West Peachtree Street
Suite M-100
Atlanta, Georgia 30308

404.815.4599

info@candid.cloud