



CASE STUDY

CI /CD ROLLOUT TO STREAMLINE FREQUENT RELEASES



ABOUT ACCESS.IO

AXCESS.IO is a niche provider of Managed Cloud services to the businesses worldwide and has served an ever-growing number of clients since its inception. In a relatively short period of time, AXCESS.IO has quickly become a niche consulting firm specializing in Cloud Advisory, Cloud Managed Services, and DevOps Automation.

www.access.io
+91- 80 889 20295 / +1 866 CLOUD
sales@access.io



CLIENT (RD OVERSEAS) BACKGROUND

RD Overseas is a well-recognized professional manufacturer of high-quality car accessories.

Established in 1999, RD has been making driving more secure, convenient and entertaining for millions of customers for more than two decades. All of the company's systems and operations are certified under ISO 9001 certification since 2006. As part of their commitment to delivering the highest-quality service to their extensive customer base, RD has their own customer care department available 24 hours a day, seven days a week.

THE PROBLEM/ REQUIREMENT

RD is building a vehicle tracking system for Indian market. They need a solution that can scale to 100s of thousand devices in very short amount of time. This can potentially generate 100s of millions of IoT events every month.

They also intend to provide solution to different markets and segments, which requires very high velocity in software delivery.

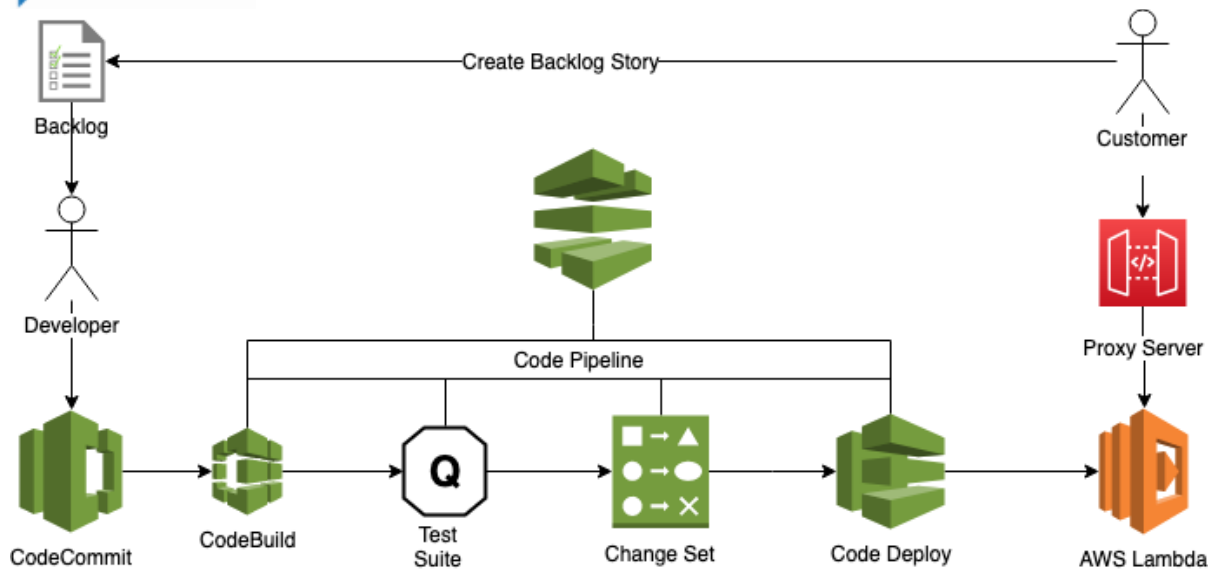
OUR SOLUTION

We built an AWS Serverless solution handle the volume using services like AWS lambda, API Gateway, AWS IoT gateway. Most of the codes are run through AWS lambda in this solution.

We configured AWS DevOps solution using AWS CodePipeline, CodeCommit, CodeBuild, and CodeDeploy to deliver the software releases in a very automated fashion. We configured AWS pipeline to trigger activities based on the workflow configured.

Here is the workflow:

1. Customer creates the backlog story in JIRA.
2. The developers commit software changes in the AWS CodeCommit.
3. Any commit generates AWS CloudWatch logs, which in turn generates AWS notifications. There are different topics configured for different repositories.
4. Respective code pipeline is activated, which in turn triggers the build followed by automated tests. The CodeBuild creates the build artifact and pushes to AWS S3.
5. On successful execution of the automated test, the code deployed in AWS Lambda.
6. Customer is notified on successful deployment through an email configured through SNS Notifications.



Note: Build stages used input as well as output artifacts to resolve and store the Amazon S3 artifact bucket. To make this process even more efficient, we used pipeline, which zips and shares the files to respective locations.

The role of AWS services in the overall solution:

- **AWS CodeCommit:** We used AWS CodeCommit, a managed Source Code Control System, as code provider in AWS Pipelined. We configured CloudWatch logs for any check-in, which in turn raised SNS notifications. Code pipeline was triggered on the notifications to trigger the build through AWS CodePipeline workflow.
- **AWS CodeBuild:** We configured AWS CodeBuild as build provider in the AWS code pipeline. This builds the software and creates the artifacts for deployment into different environments. Build Artifacts are pushed into AWS S3, which is picked up AWS CodeDeploy during deployments.
- **AWS CloudFormation:** AWS CloudFormation was used to maintain stack as code. Any change would trigger the change in the stack. This not only programmatically provisions AWS resources, but this also maintains the stack details in simple readable text files.

Note: No third-party application or solution was used in the process.

THE FINAL OUTCOME

RD Overseas Pvt. Ltd. can now execute multiple deployments each day with minimal interruptions to their operations.