Serverless Architecture

Case Study

gslab

Reducing costs by switching to AWS Fargate based serverless deployment

An early-stage cloud management product improves security and availability while becoming auto-scalable

Executive Summary

Our client wanted to implement a completely scalable, containerized and cost optimized architecture for their cloud management product. GS Lab created an auto-scalable, immutable and highly available architecture using AWS Fargate which resulted in cost reductions of 85% per month. Our solution also made the product more secure while giving our customer the ability to scale up or scale down resources based on their requirements.

Overview

Our customer had developed an early stage cloud management product. Just like any other early stage product, there were lots of unknowns about immediate adoption as well as the speed at which adoption will shoot up. The product was built on AWS with a typical EC2 setup. Certain assumptions were made during capacity planning. However a lot of uncertainties are involved when it comes to early stage products and their adoption.

Challenge

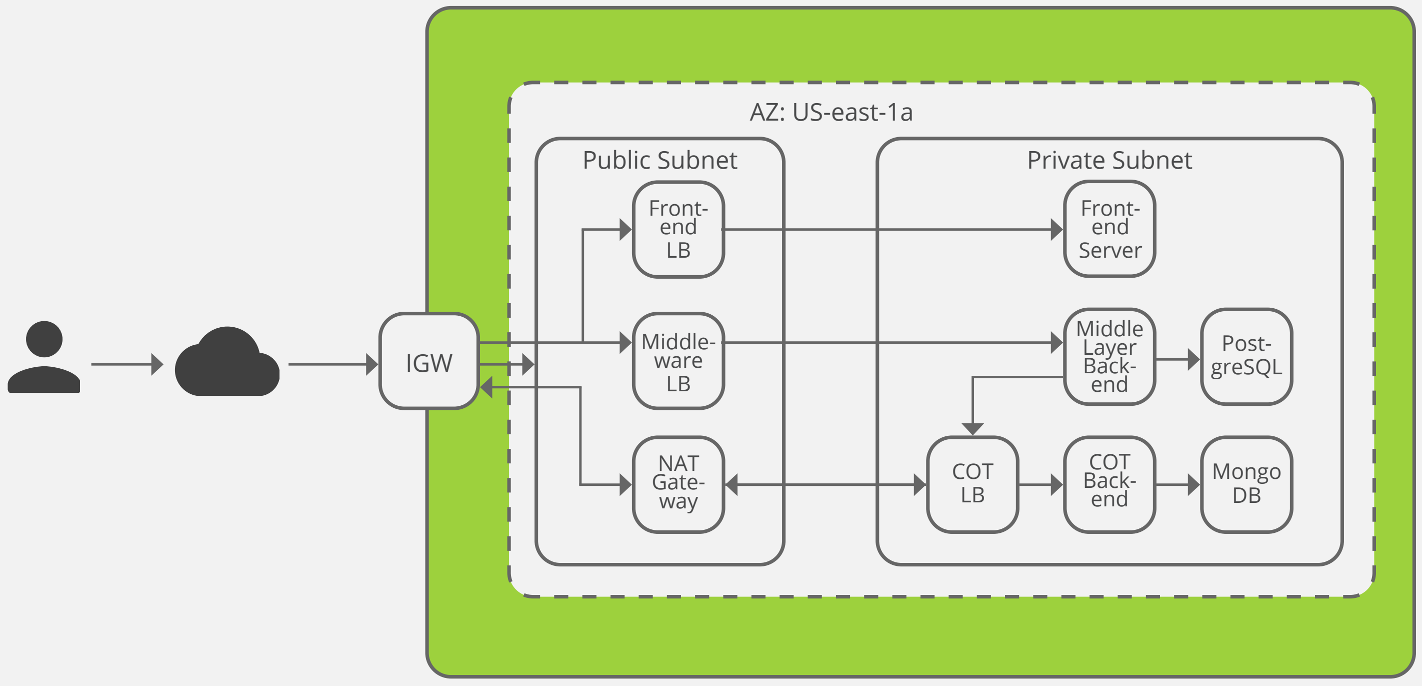
The customer's product had a long sales pipeline. There was a possibility of huge surge in traffic but at the same time there was a greater risk of clear wastage of AWS resources if many deals in the pipeline failed to materialize. The product team wanted availability, scalability and security without inefficient resource utilization.

Serverless architecture was one of the best solutions in this scenario. But AWS Lambda works on a function-as-a-service model. This would have meant restructuring a lot of components.

Technical problem definition:
We observed the following issues with the AWS deployment:

Front end server and middleware server were highly underutilized (CPU utilization less than 7%). They were large M5s costing approx $500 each month.

1. Though load balancers were used for the servers, the auto-scaling was not configured. Hence there always was a single instance running behind the LBs.
2. The product was deployed in a single AZ.



Original deployment

Possible solutions
Front end and back end servers had very low CPU utilization and were good candidates for cost optimization.

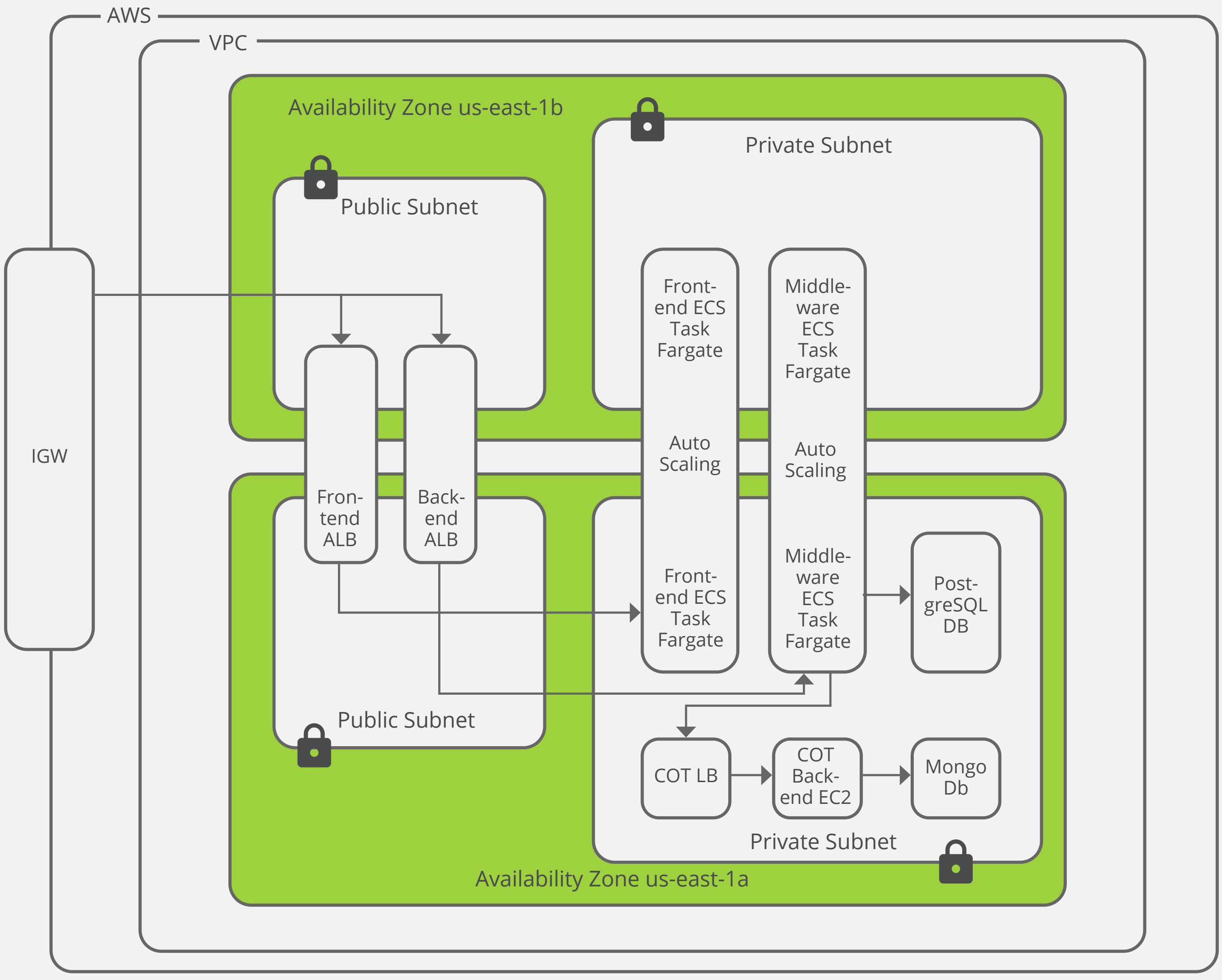
Solution

There were three possible solutions:
1. Downgrading instances.
2. Use of reserved instances.
3. Containerization of the deployment using Fargate where CPU and memory configurations can be changed at any time.

Estimating the usage was extremely difficult at the point of development. The features in the application would increase in near future which could boost demand. Hence the third option seemed to be a more apt solution for the clients needs.

GS Lab made the following changes to the architecture to take care of the issues discussed:

1. The front end and middleware was replaced using Fargate deployment where auto-scaling spanned across two AZs.
2. The ECS containers were deployed in the application specific VPC in the private subnet.
3. Classic load balancers were replaced with application load balancers in the public subnet.
4. ECR (Elastic Container Registry) was used to store the container image. Fargate pulls the image from ECR at the time of deployment.
5. Container metrics and logs were monitored through CloudWatch.


AWS
VPC
Availability Zone us-east-1b
Private Subnet
Public Subnet
IGW
Front-end ECS Task Fargate
Middle-ware ECS Task Fargate
Auto Scaling
Auto Scaling
Fron-tend ALB
Back-end ALB
Front-end ECS Task Fargate
Middle-ware ECS Task Fargate
Post-greSQL DB
Public Subnet
COT LB
COT Back-end EC2
Mongo Db
Private Subnet
Availability Zone us-east-1a

Our solution

Server pricing comparison

A] Old deployment cost

	Price Per Hour	*Hours*	*Days*	*No Of Instances*	*Total Price*
M5 Large EC2 instance	0.086	24	30	10	619.2
				Total Cost Per Month	**$619.2**

Numbers are indicative in nature

B] New serverless deployment using Fargate

	Allocation	*Price Per Hour*	*Hours*	*Days*	*No Of Instances*	*Total Price*
CPU	0.25 CPU	0.04048	24	30	10	72.864
Memory	1 GB	0.004445	24	30	10	32.004
				Total Cost Per Month		**$104.868**

Numbers are indicative in nature

What is AWS Fargate

AWS Fargate is a serverless compute engine for containers that works with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS). Fargate removes the need to provision & manage servers while allowing you to specify & pay for resources based on usage per application. It also improves security through application isolation by design. Fargate runs each task or pod in its own kernel providing the tasks and the pods their own isolated compute environment. This enables your application to have workload isolation and improved security by design. It works on pay per use pricing. Fargate for containers is what Lambda is to functions. Lambda is FaaS (function as a service), whereas Fargate is CaaS (container as a service).

Impact

Reduced Costs

Improved Scalability

Improved security

1. The consolidated cost for these two servers reduced by a whopping 85% per month.
2. We containerized the deployment with auto scaling and high availability using two AZs. If the workload increases or a task goes down for some reason, Fargate launches a new task in a different AZ.
3. We created an immutable deployment that is more secure as the underlying infrastructure is not directly accessible.
4. The memory and CPU allocations can be changed anytime depending on the requirements.

Great Software Laboratory (GS Lab) has been the technology partner of choice to 100+ organizations across North America, Europe and Asia-Pacific for over 16 years. Leveraging our expertise in 130+ tools & technologies, we have created 300+ ‘first-of-its-kind’ solutions to real-world problems. Our ‘Beyond code’ philosophy ensures that we not only push boundaries of existing technologies but also try out newer problem solving approaches to keep our customers one step ahead of their competitors. Our global team of 1200+ employees is adept at creating ‘real value’ at each stage of the customer growth journey, right from proof-of-concepts to completely scaled up products. For more information about our solutions & offerings, please visit www.gslab.com

Copyright©2020 Great Software Laboratory. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, transmitted in any form or by any means,electronic, mechanical, photocopying, recording, or otherwise, without the express written permission from Great Software Laboratory. The information contained herein is subject to change without notice. All other trademarks mentioned herein are the property of their respective owners.

gslab™

www.gslab.com