

About Poshmark

Poshmark is an online marketplace app that makes it very easy to sell used clothing and accessories online. This makes them seem very similar to eBay for the national market and Craigslist for local selling, but they bring a unique experience that emphasizes social media and being mobile-friendly. With over two million sellers and millions of shoppers, Poshmark brings together a vibrant community every day to express themselves and share their love of fashion.

Problem Statement

Poshmark had all their Technology/Ecommerce Infrastructure – a total of over a thousand Servers – hosting a variety of Applications/Clusters – in the USus-Wwest-1 Region of AWS. Having their product launched very recently, the Poshmark Business Team of Poshmark needed their Applications and Infrastructure to be agile enough to allow for experimentation and deployment of multiple iterations of their product. However, Poshmark's Team had, in the recent past, experienced multiple failures of its Applications running on AWS. Poshmark's User base was growing aggressively, and Business wanted to ensure that not only its tech stack and the underlying Infra was flexible enough to sustain constant changes. In addition, they needed to ensure their, but its website also does not experience any disruptions. With these objectives ideas in mind, Mactores and the Poshmark's team worked together to define the most set out to discuss on the viable solutions for the long term

Proposed Solution

The Mactores' Team and Poshmark's teams had extensive discussions to understand Poshmark's current Infrastructure and Applications, and the relevant pain points. This helped both teams agree on the improvements required to meet the technical objectives.

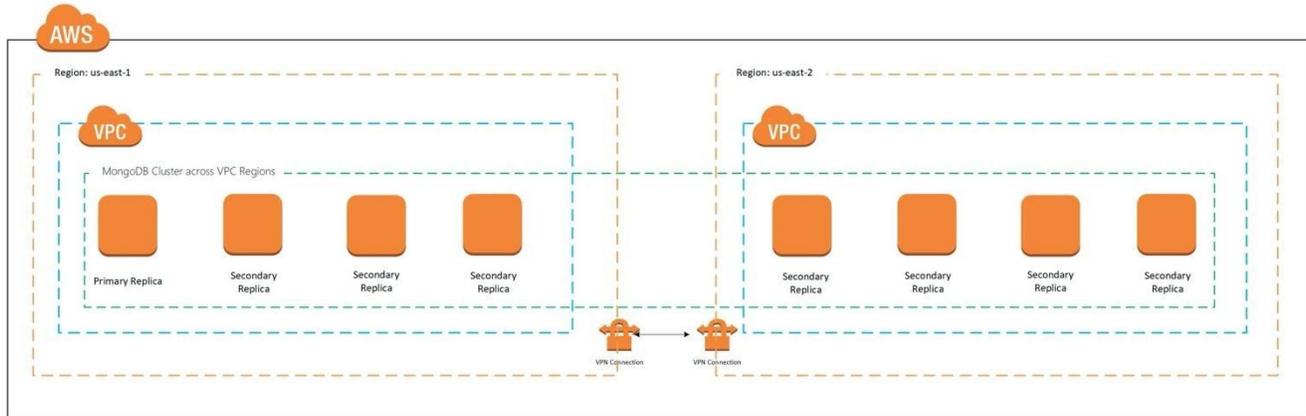
Existing Postmark's Infrastructure, in the AWS us-Eeast-1 Region, hosted a variety of Applications, including Apache Webservers, ES Clusters, MongoDB Clusters, Active Directory Servers and DNS Servers. As per the AWS Well Architected Framework, an assessment of the current Applications and Infra setup was conducted against a set of best practices for architecting systems in the Cloud. Poshmark wanted to make full use of the features of the AWS Cloud for its Applications, by hosting its Application not only on more Servers, but also across more DCs, thus achieving higher resiliency. T

hey then realized us-west-1 Region of AWS, consisting of two Availability Zones, could be a limitation for a highly resilient Infrastructure. Further, on analyzing Poshmark's website statistics, it was found that most of Poshmark's User Base was located closer to the us-west-2 Region, as opposed to the us-east-1 Region. It was, thus, mutually decided that moving the Infrastructure to the new Region was the best possible option available. Apart from moving to the new Region, it was also decided that upgrading the current MongoDB cluster to accommodate a higher number of replicas would also be part of the Resolution.

With the idea of moving the Infra to the new Region, Mactores re-designed the Network and Landing Zone for the new setup in the us-west-2 Region. A comprehensive plan was laid out on how each of the Applications, Database Clusters and other components of the Infrastructure would be migrated to the other Region. To facilitate data movement from the current Region to the new Region, network connectivity between the two Regions was established with CISCO Routers/AWS VPC Peering across different VPCs in the different Regions.

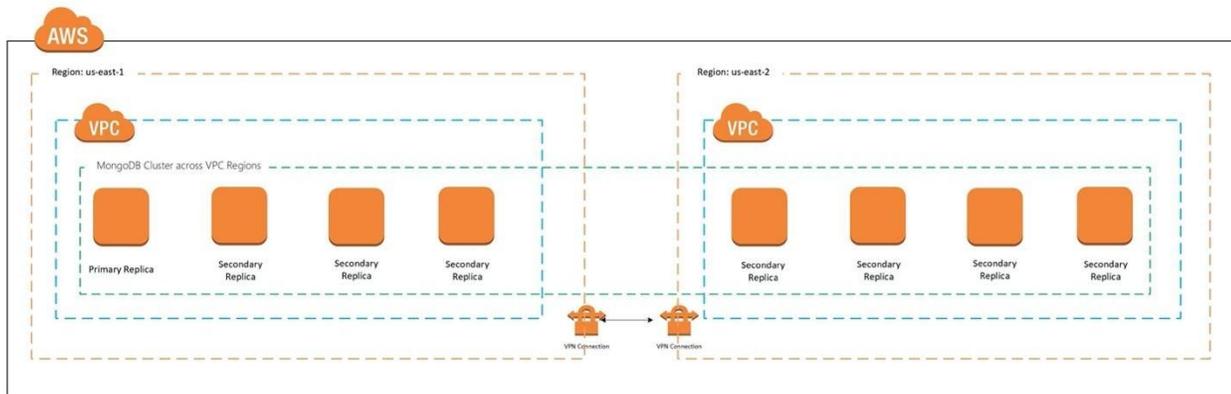
MongoDB Replicas

Poshmark – MongoDB Cluster across VPC Regions



EBS to new Region

Poshmark – MongoDB Cluster across VPC Regions



Deploying multiple Network Components without an automated Solution would prove to be very time consuming and error prone. For deployment of the Landing Zone and AWS Network Components in the new Region, Terraform/CloudFormation Templates were used. In the pre-cutover phase, using CFTs provided the flexibility to deploy different components only when required. Terraform Scripts/CFTs were used for provisioning of all the Servers. For migration of Application Servers, AMI Backups were created and shared across Regions. MongoDB Clusters were extended to the new Region by adding Secondaries from the new Region to the same MongoDB Clusters. Priorities of every node was set in such a way to make sure that the new Primary Replica would come up in the new Region. With minimal/zero downtime cutover, R53 provided the best solution by using weighted DNS routing.

Third party applications or solutions used

Not Applicable

Start Date – October 1, 2018

End Date – December 20, 2018

Outcomes & results

This project helped Poshmark migrate the platform seamlessly from us-west-1 to us-west2 AWS region without downtime. On the new Region, Poshmark was able to make full use of the AWS Cloud by deploying their Applications in multiple AZs, thus being confident of a Highly Available Ecommerce Website. By using Automation Tools like Terraform Scripts, theand as Landing Zone and leveragingthe Well Architected Framework, Poshmark was able to build their Infrastructure. In addition, tThey could also serve their customers better with their website observing much lower response times.

Lessons Learned

It's important to continue to invest into a DevOps pipeline to automate code deployments, ensuring infrastructure version upgrades are made in the timely manner. If an organization maintains healthy DevOps culture, security, scalability availability and uptime become extremely important.

About Mactores Cognition Inc

Mactores quickly solves core business problems and drives disruptive change by applying the latest automation technologies in Data Analytics, AI/ML and DevOps. We design, deploy, integrate and manage rapid migration and transformation solutions to accelerate enterprise data platform migrations using automation developed over dozens of successful use cases.