Something about us:

Our website is an Ecommerce website which specifically serves the various products of the SABELLA brand. SABELLA, Malaysia's No 1 ironless clothing brand started with a humble beginning. First planted its seed under a small room in a apartment at Ukay Perdana where the Founder SITI FATIMAH ABD SAMAT or fondly known as Siti Sabella previously worked as a full time housewife. Our signature ironless kurung, Lorez Kurung with the plain material effortlessly sets the tone and trend in the market which has been recognized as our main root. SABELLA creative, fresh innovative ideas let it blooms more beautifully after we come out with Zalia Kurung followed by Queeny Kurung which is become new benchmark to our product range.

Through ups and downs, sweats and tears with hard work, perseverance and sheer determination, SABELLA has marked its winning point with a loyal and huge fan base known as SABELLA FAMILY. This clearly defines SABELLA as a brand that brings love and unity.

The challenges faced on-premises:

The website was initially hosted on a shared hosting platform with limited number of requests, and everything was working fine. But then, with continuous publicity and growth in the no. of customers, the daily requests being made to the website started increasing exponentially which was solved every time by upgrading the account resources on the shared hosting platform.

But we soon realized that manually scaling the infrastructure will eventually lead us to a dead end and this situation could severely affect our sales. Furthermore, as we were going to have a huge publicity event in the upcoming months, we were expecting a massive increase in the website requests and when we tried to simulate the traffic on our hosting platform, we started facing performance and availability issues. Even after switching to a dedicated hosting server, a single server was still not able to handle the traffic and provide the performance expected.

A lot of solutions and workarounds were applied at the on-premises end to achieve the expected scalability and availability, but we still couldn't achieve the expected results and with addition of each new component to the existing infrastructure, the solution was getting more and more complicated and difficult to manage with a lot of manual interventions required even for the smallest amount of change.

Reason for choosing AWS:

As the on-premises infrastructure was falling short to the requirements, we needed a solution where our website can maintain high availability regardless of the size of the website requests without sacrificing performance. The solution also needed to ensure that incoming traffic is properly distributed among the servers deployed in the infrastructure and the infrastructure scales automatically with the ever-growing needs with minimal to no manual intervention.

This was when we decided to go with AWS where we were able to take advantage of the wide range of services available on their cloud platform to upgrade the availability, scalability and performance of our website. With AWS, we didn't have to worry about the resource depletion and setting up the complex infrastructure for something like load balancing and instance scaling. It was all possible with a few clicks and some custom changes and we were able achieve the expected results for our website within a short amount of time which tool months on the on-premises environment with no avail.

With our web tier on AWS, we were able to concentrate on the website development and enhancement without worrying about the infrastructure layer. Once the infrastructure setup was complete, we almost never had to perform any manual changes to scale the infrastructure or repair it where the infrastructure automatically scaled to needs of our requirements while maintaining the availability and performance of our website.

Due to the global reach of AWS, we were also able to find an AWS datacenter for our cloud resources in our geographical proximity which also helped us to reduce the latency between our clients which were mostly from the same region as our website. This was also a plus factor which added up to enhancement of the performance of our website.
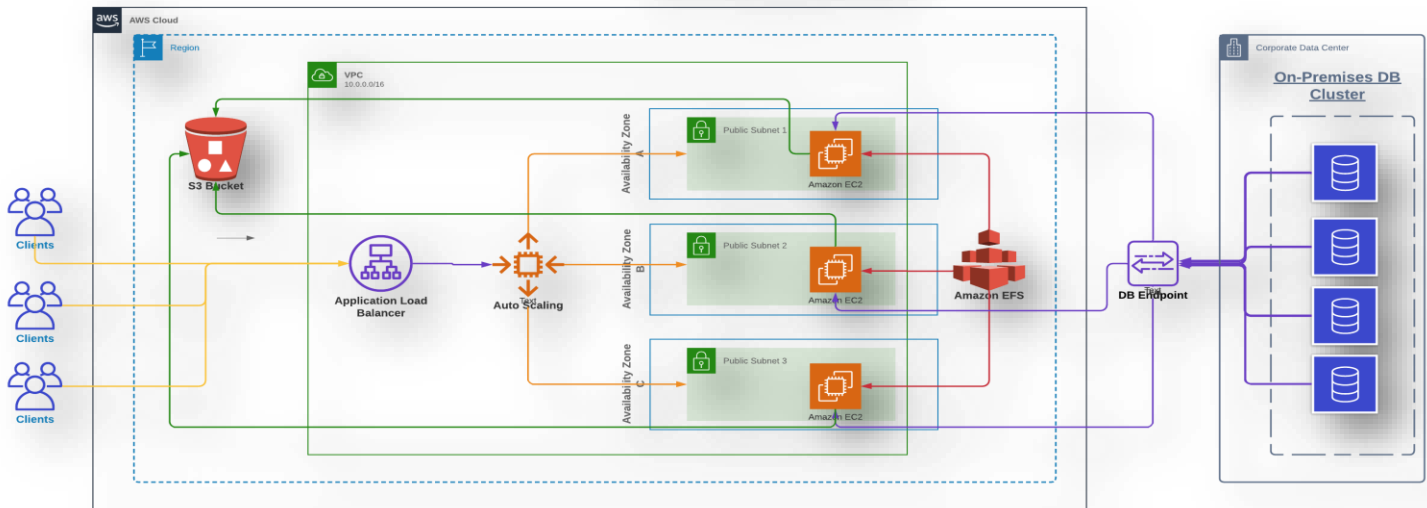
Our infrastructure on AWS:

Sabella Cloud AWS Architecture

Taking advantage of the wide range of services made available by AWS, we were able to achieve our ideal infrastructure on AWS. We first migrated our on-premises web server data to AWS EC2 instance and with the help of highly elastic and scalable nature of AWS, we were able to test and quickly switch between the instance types to find a perfect fit for our web tier requirements, both in aspects of costs as well as performance.

Once our website was ready on the EC2 instance, an application load balancer was deployed in front of an autoscaling group where various types of scaling policies were used with the help of highly customized CloudWatch alarms based on custom CloudWatch metrics collected from our service statistics to scale out and scale in our instances accordingly based on the fluctuations in the client traffic. The newly scaled instances by the autoscaling group would automatically get registered to the ALB which will allow the ALB to automatically distribute the incoming client traffic among them. This combination of ALB and ASG allowed us to achieve high availability and scalability at time of high traffic automatically without any kind of manual interventions.

AWS Elastic Filesystem was used to share the website data among the EC2 instances spawned by the autoscaling group while the static files of the website were served from the S3 bucket configured in the same region as of the web tier providing a highly durable storage for our critical files and low latency content delivery to our clients in the same region. Though there were much better solutions like CloudFront for content delivery with caching support, but, due to some cost constraints, we had to stick with S3 which itself wasn't a bad decision.

The flow of the requests was encrypted in-transit with TLS encryption right from client's browser to the EC2 instance running the web server. An on-premises db cluster was used for the database tier of the website to comply with some regulatory requirements. We also have regular backups configured for our static files on S3 with data replication to another S3 bucket and the EFS backup with the help of AWS Backup service.

This way, we were able to achieve our goals with AWS and from the day of setup till now, there haven't been many times when we had to look back at our infrastructure for modification or manual scaling and even if we must do so, it has mostly been to apply security patches to our instance or the website code.

Services Used:

S3, Elastic FileSystem (EFS), EC2, Application Load Balancer (ALB), AutoScaling Group, IAM, AWS Backup, AWS VPC

Benefits of using AWS:

- Least management overhead with little to no manual interventions required to manage the services provided.
- Scaling in seconds. As compared to our infrastructure on-premises, we had to go through a lot of hassle to upgrade the resources of our servers. On AWS, it can be done with just a few clicks.
- We were able to achieve high availability and scalability with AWS which wasn't possible on-premises, as our infrastructure now scales up and down automatically according to our requirements.
- With our infrastructure on AWS, we are now able to concentrate on the sales and development aspects which had become a second priority when we had our infrastructure hosted on-premises.
- Backing up our data has also become easy with AWS provided solutions.

About Geeks Solutions

Geeks Solutions is a managed IT services provider with a diverse, multidisciplinary, expert technical team ready to be deployed for technical support services to manage various IT managed services like server management, cloud management & server monitoring. Over 15 years of Professional IT service experience, our nationwide clients are consistently happy with our 24x7 support to their businesses.

GEEKS SOLUTIONS
Managed IT Services You Can Trust