applied cloud computing

Upgrading to AWS with ACC for Better Scalability

TABLE OF CONTENTS

GIVING OUR BEST BY PROVIDING THE BEST SERVICES USING AWS.

Amazon Web Services, Inc. (AWS) is an Amazon subsidiary that offers metered pay-as-you-go cloud computing platforms and APIs to consumers, businesses, and governments. Cloud computing is the on-demand, pay-as-you-go distribution of IT services over the Internet instead of purchasing, operating, and maintaining physical data centres and servers, one can rent computing power, storage, and databases from a cloud provider like Amazon Web Services on an as-needed basis (AWS). These web services for cloud computing include a variety of basic abstract technical infrastructure and distributed computing building blocks and utilities. Virtual computers on AWS have hardware central processing units (CPUs) and graphics processing units (GPUs) for processing, local/RAM memory, hard-disk/SSD storage, a choice of operating systems, networking, and pre-loaded software applications like web servers, databases, and customer relationship management (CRM).

WHY SHOULD ONE USE CONTAINERIZATION

Containerization is one of the most effective virtualization methods available to programmers, containers increase efficiency in two ways: they maximize resource utilization and reduce overhead. Containers enable a host to use nearly all available resources when correctly designed. It is a way of modernizing monolithic application to light weight.
Container images are measured in megabytes rather than gigabytes, making them more lightweight than virtual machines. Containers are easier to deploy, run, and administer than traditional servers. In milliseconds, containers spin up because of their low order of magnitude.
It is a DevOps methodology specially designed to encourage the developers to integrate their code into a shared repository early and often, and then to deploy the code quickly and efficiently.
Containers are mostly employed to keep the applications separate from the environment in which they'll run.
Containerized apps can thus be deployed more quickly and efficiently in nearly any environment. Containers are a convenient way to package and run your apps. One can manage the containers that run the applications in a production environment and ensure that there is no downtime. For instance, if one container fails, another must be started.

Containers are comparable to virtual machines (VMs), except they allow programs to share the same operating system (OS). As a result, containers are considered light. A container is like a virtual machine; it has its own filesystem, CPU, memory, process space and other resources. They are portable across clouds and OS distributions because they are isolated from the underlying infrastructure.
Containers are popular because they offer additional benefits, such as:
Container image creation is more simple and efficient than using VM images for developing and deploying agile applications.
Continuous development, integration, and deployment: ensures that container images are built and deployed reliably and often, with rapid and efficient rollbacks

HOW KUBERNETES HELPS FOR EFFICIENCY

Kubernetes is a platform for resiliency in distributed systems. It handles your application's scaling and failover, as well as providing deployment strategies. Kubernetes, for instance, can effortlessly manage your system's canary deployment. One may use Kubernetes to perform load-balancing and service discovery. Kubernetes can expose a container by utilizing its own IP address or its DNS name. If there is a lot of traffic going to a container, Kubernetes can load balance and spread it so that the deployment stays stable.

Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

It helps organize your storage, Kubernetes lets you mount any storage system you choose, including local storage, public cloud providers, and more.

KEY ADVANTAGES OF USING KUBERNETES

DEVELOPMENT AND RELEASE TIMEFRAME REDUCTION

Kubernetes enhances the development, release, and deployment processes much smoother: for example, it enables container integration and makes managing access to storage resources from many providers much easier.
Furthermore, with microservices-based architectures, the application is divided into functional units that communicate with one another via APIs, allowing the development team to be divided into smaller groups, each specialized in a certain feature. This structure enables IT teams to work with more focus and efficiency, resulting in faster release times.

OPTIMIZING IT COSTS

Kubernetes can help enterprises save time and money by assuring scalability across many environments through dynamic and intelligent container administration. Thanks to native autoscaling like HPA(horizontal pod auto scaler), VPA(vertical pod auto scaler) logics and integrations with major cloud vendors capable of dynamically delivering resources, resource allocation is automatically adjusted to the actual application needs, while low-level manual activities on the infrastructure are considerably minimized.

INCREASED SOFTWARE SCALABILITY AND AVAILABILITY

Kubernetes enables dynamic peak management by scaling applications and underlying infrastructure resources up or down based on the organization's changing needs. For example, as the date of an event approaches, an e-ticketing system will see a surge in ticket demand.

Kubernetes will be able to dynamically request new HW resources to be allocated to the infrastructure providing the service, thanks to its native Autoscaling APIs such as HPA and VPA. This will assure the service's performance. Once the emergency has passed, Kubernetes will scale down the resources that are no longer required, preventing waste.

THE NEED FOR FLEXIBILITY IN MULTI-CLOUD ENVIRONMENTS

Containerization and Kubernetes are flexible for example, enabling the solution to deliver on the promises of the new hybrid and multi-cloud settings, ensuring that applications may run in any public or private environment without losing functionality or performance. As a result, the risk of lock-in is lessened

KUBERNETES' ADVANTAGES FOR BUSINESSES INCLUDE

- Control and automate upgrades and deployments.
- Save money by maximizing infrastructure resources through better hardware utilization
- Container orchestration across many hosts
- Many problems caused by the growth of containers can be solved by grouping them into "pods".
- Real-time scaling of resources and applications
- Application testing and autocorrection

THE CLIENT SIDE POC IMPLEMENTATION BY OUR TEAM

The client being an established player itself provides a full variety of innovative products to the Banking, Insurance, and Financial Services industries, as well as the mid-market ERP area. They believe in cultivating a culture of excellence that will benefit the clients and when their clients evolve, they evolve as well.
The company provides core software to Financial Services providers and ERP solutions to mid-market enterprises across the Asia Pacific, Middle East, Africa, and India.

The company had to increase their scalability, they had their own POC(Proof of Concept) on which with guidance and research they wanted to migrate to containerization, basically docker containers are small containers that include everything needed to run an application, so you don't have to rely on what the host already has installed. Earlier the company had a monolithic ERP application that they wanted to modernize and shift to AWS after our team suggested them.

The company basically wanted their application needs to be containerized they had their code jar file containerized which was then deployed in EKS(Elastic Kubernetes Cluster) as Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem.

Kubernetes services, support and tools are widely available, and then load testing was performed as load testing is a non-functional software testing procedure that evaluates a software application's performance under a certain load. It determines how the software application works when numerous people access it at the same time. It is used to identify performance bottlenecks and assure the stability and smooth operation of software applications prior to deployment and then the customer application was hosted on a Kubernetes cluster on-premises.

The customer had recurring maintenance issues with the application they wanted memory optimization and modifications in the code and changes in the config files to increase the memory of the application and wanted to move it to AWS Cloud's Elastic Kubernetes Cluster (EKS) for improved support and manageability and the team designed and suggested an architecture for the same and delivered it as per clients requirement. The key roles and responsibilities of developers were underlying application code, refactoring and converting application architecture and for DevOps developers were creating AWS containerization resources, containerize Application using EKS

The project included documentation defining for creating and granting access to IAM (Identity Access Management) to users, creating RDS(Relational Database Service) according to company's specifications and sharing credentials with the company's team. The company's team will combine those credentials with WAR files and share them with the DevOps team. Then EKS cluster and worker nodes are started, for docker images, then creating the docker file. Then to join the EKS cluster, we created a Bastion host and installed the appropriate software, created a docker image of the company's team's WAR file.
Then created ECR repositories and uploaded docker images to them,created the deployment, service, and ingress load balancer files, then deployed EKS.

The company's team then performed UAT(User Acceptance Testing). There was an issue related to application load balancer which the company was not satisfied with so then we suggested they migrate it to network load balancer to increase their scalability.The project was successful on various criterias as kubernetes cluster was successfully deployed in the application (EKS Cluster). Configured and saved application logs, as well as AWS API logs (VPC Flow Logs and CloudTrail logs) to the centralizedS3 bucket,routing production traffic from on-premises to AWS environment resulted in a successful Active-Active failover test. The project was delivered as per the delivery plan of three weeks.

WORKFLOW OF THE PROJECT

Access to the application code was provided to ACC developers by clients team
The team at ACC analyzed the code and defined required changes to convert it from monolithic to microservices.
The project setup included creating a project using Springboot application and converting the application architecture from monolithic to microservices.
The developer team required the client teams help to prioritize the modules.
Then Created IAM (Identity Access Management) users and provided access.
Creation of RDS as per clients requirements and then the credentials were shared with the developer team.

The Developer team integrated those credentials with the WAR files and shared the same with our DevOps team.

Then they had to launch EKS clusters and worker nodes.

Then created a docker file for docker images and created a Bastion host to connect to the EKS cluster and install required software then we created a docker image of the WAR file shared by the clients team. Created ECR repositories and pushed docker image into it, made deployment, service, ingress load balancer file and performed EKS deployment.

Which was then followed by UAT testing by developer and tester.The Documentation of all the deliverables would be handed over to the clients team once deployment process completes.

ASSUMPTIONS

The assumptions included the following steps between the clients and our Team

We would receive dedicated / partial assistance from clients for any dependencies identified throughout the project execution. The scope of work had been determined; any additional scope will be contracted separately on an effort basis.

All materials created as part of this project will be sent to the clients team any travel expenses will be billed to the client as part of this contract, the clients team will participate in the containerization process.

ABOUT ACC

ACC is an AWS Advance Partner with AWS Mobility Competency. Awarded The Best BFSI industry Consulting Partner for the year 2019, ACC has had several successful cloud migration and application development projects to its credit.

Our business offerings include Digitalisation, Cloud Services, Product Engineering, Big Data & Analytics and Cloud Security. ACC has developed several products to its credit. These include Ottohm – Enterprise Video and OTT Platform, Atlas API – API Management and Development Platform, Atlas CLM – Cloud Life Cycle Management, Atlas HCM – HR Digital Onboarding and Employee Management, Atlas ITSM – Vendor Onboarding and Service Management and Smart Contracts – Contract Automation and Managem*ent.*

acc

applied cloud computing

www.appliedcloudcomputing.com

Shubho Pramanik

(+91) 90297 20294

shubho@acc.ltd

Rogin Rappai

(+91) 8828478321

rogin.rappai@acc.ltd

also reach us at