CSDO 1020 – Next Generation DevOps

Big Bang Case Study

This is a case study for a fictitious business providing the group project team a context for the DevOps CI/CD Pipeline modernization group project.

Company Overview

Big Bang is a global e-commerce web-based application that helps its customers to purchase a vast variety of goods. The application has grown from a few servers and applications in the founder's garage in the Metro Vancouver area to several hundred servers and applications in a co-located data center.

Big Bang's business has grown exponentially due to COVID19 and unexpected popularity. They had problems automating and scaling their global footprint and rolling out new continuous deployment capabilities to update their applications quickly. Because of this growth and the company's desire to innovate faster, Big Bang is committing to adopt Agile, DevOps, and SRE principles and practices in phases:

- In phase one, they are looking to design and implement a traditional CI/CD Pipeline using OSS tools for their workloads running in a co-located data center.
- In phase two, and as their DevOps maturity level increases, they are looking to experiment with Next Generation DevOps using OSS tools where their multi-tiered distributed applications will span hybrid/multi-cloud environments.

Executive Statement

Our competitive advantage has always been our focus on the customer, with our ability to provide excellent customer service and minimize downtimes. The goals are to build skills and invest in training our team while addressing immediate market needs through incremental innovations.
Our 3-year strategic plan is to create an ecosystem of new products by increasing autonomous development and operation capabilities, creating a path to modernize the legacy systems, and enabling data-driven decisions powered by Machine Learning and Artificial Intelligence.
Our top priorities are to provide best-in-class services to our customers, support analytics capabilities and real-time predictions, and cost management. In addition, we wish to improve our development and operations to rapidly iterate on our deployments of new functionality and bug fixes.

Existing Technical Environment

- Big Bang's systems are currently hosted in a co-location facility.
- There are separate environments for development and testing.
- All systems are running on different Linux distributions and a mixture of physical and virtual servers.
- Infrastructure provisioning and patch management are done manually with the support of shell scripts.
- Batch Jobs are based on Linux shell script and crontab and performed on VMs created for each job.
- Apache Subversion is used as a version control system.
- Hudson legacy tool is used for continuous integration.
- Hundreds of non-containerized web applications and microservices-based APIs run on a mixture of application servers.

- Data stored in a mixture of SQL and NoSQL databases.
- Monitoring is currently done via various proprietary software where actions are reactive and not proactive. Alerts are sent via email and often ignored.

General Requirements

- The DevOps CI/CD Pipeline should be agnostic, which can be applied to any environment, whether on-premises and/or the cloud. Consider cloud-native applications when designing your pipeline.
- Select appropriate architecture, design patterns, and best practices for the specific use case when designing your solution for each service and state the rationale behind your decision.
- Big Bang recently started expanding their on-premises development and operations environments to the Public Cloud.

Business Requirements

- Adopt Agile, DevOps, and SRE principles and practices to increase the company's maturity level.
- Decrease infrastructure administration, operational costs, and adapt to seasonality.
- Achieve a 15% - 25% cost reduction over the current development and operation models.
- Provide a minimum of 99.99% availability for all customer-facing systems.
- Provide financial management for your containers in a hybrid/multi-cloud deployment.
- Minimize operational complexity and increase automation.
- Increase speed and reliability of development workflow and rolling new product features.
- Provide centralized visibility and proactive action on systems performance and usage.
- Provide analytics capabilities to provide insights into cross-selling trends.
- Make predictions and generate reports on trends based on customer's transactional and operational data.

Technical Requirements

- Dynamically scale and provision new environments based on customer's activities.
- Maintain and manage multiple container-based environments.
- Improve and standardize OSS tools necessary for automation.
- Use Git-based repository for Source Control Management instead of Subversion.
- Use Git as a single source of truth for your entire system (infrastructure and applications).
- Allow developers to deploy container-based workloads in highly scalable environments.
- Allow developers to be productive and run experiments without compromising security and operations.
- Create a flexible and scalable platform for developers to create API services.
- Use cloud-native design principles for all new applications.
- Microservices-based APIs run on Kubernetes cluster in a hybrid/multi-cloud deployment.
- Batch Jobs must be triggered automatically in a cost-effective way utilizing one or more of the Public Cloud services without the need of dedicated or on-demand servers.
- Reduce latency for Big Bang customers.
- Use keys and secrets management optimized for security.
- Provide consistent logging, log retention, monitoring, and alerting capabilities.
- Create real-time data analytics .
- Allow the use of MLOps and AIOps to facilitate operations predictions capabilities in a hybrid/multi-cloud deployment.