Case Study | Learning product mapped to Dev Ops best Practices

Overview

Spotmentor (a former startup, presently E&Y Spotmentor) is an end to end skilling, careers and learning platform solving a critical problem in the HR tech space.

The platform uses NLP technologies to identify job skill demands and suggest appropriate personalized learning content based on employee skill gaps. Spotmentor was acquired by Ernst & Young LLP in February 2020.

It serves some of the largest clients in IT, Finance, Telecom space and is disrupting the way people consume personalized learning across the globe. Skill training and learning are focus of every organization and Spotmentor targets it very well by utilizing the multitude of data available.

Tool Set

Jenkins DSL
Kops
AWS
EFK Stack
(Elasticsearch,
Fluentd and Kibana)
Git and Github
SonarQube
Grafana,
Prometheus and
Alerts Manager

About Calance

Calance is an IT Services firm operating in the United States, Canada and India. We provide Consulting, Application Development & Systems Integration, Managed Services, IT Staffing and pre-built Products & Solutions. With a long history of success in IT Services, we help clients tackle many of today's technology challenges. For more information, please contact us at info@calance.com Visit our website at www.calanceus.com

Challenges

Spotmentor an end to end HR technology LXP product faced multiple challenges during the lifecycle of product development. Few challenges identified during the stint were:

- As a startup with financial limitations it is challenging to hire a separate operations and security team. The focus has been to hire people who can develop features quickly and quickly arrive at a product market fit.
- It is a Python web application developed using Flask framework code and with more modules being incorporated it started showing increasing cyclical dependencies.
- A single server hosting the entire application was becoming challenging to monitor.
- Dying services and application outages were a frequent challenge. The firm started losing multiple clients during the product demonstration.

Manually deploying applications with a very frequent release cycle started becoming a real challenge.

Our Solution

A model shift to DevOps was our only rescue. The tasks to start out in our DevOps journey were being explored by the founders. Focus was given to areas that consumed significant development time. Application deployment was a clear winner. It took 2 hours to deploy and check all the configurations when a release was planned. We urgently needed a CI/CD pipeline established to automate our builds and deployment. Jenkins was considered and a pipeline was developed within 10 days. The second challenge targeted was the increasing cyclical dependency. A lot of solutions were discussed a proposed but eventually a decision was made to break the code into microservices. However. the CI/CD and recently build did not align with microservices. Also, the entire infrastructure could not support microservices.

After multiple rounds of conversation with the founding team it was decided that some budget would be allocated to solving this. We started working on building the microservices infrastructure while the development team continued breaking the application to smaller chunks. Docker and Kubernetes were our go to technology. A Kubernetes cluster was setup on AWS using Kops (Infrastructure as Code). A private subnet was configured and our objective was to create a highly secure system. The CI/CD was modified to deploy microservices applications. This project was finished within 1.5 months. The infrastructure was established to support new microservices very quickly. Latest GitOps techniques were used for application delivery and state management. During the process developers upskilled themselves to perform operations as well.

Results

The results were extraordinary. The average outage of the servers which were in the range of 5-10 per week were minimized to 0.1 per week (averaged over a period of 1 month). The deployment time which initially took 2 hours was reduced to 10 minutes at a click of a button. All the members of the team were comfortable with server operations as well eventually resulting in harmonious development and operations aka DevOps.