Intuit Modern SaaS Platform

Apr, 2019

Laks
Agenda

● What is the Modern SaaS platform?

● Why we built it?

● How we built it?
## Goals

<table>
<thead>
<tr>
<th></th>
<th>Measured value</th>
<th>Expected Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recoverability</td>
<td>X</td>
<td>10X faster</td>
</tr>
<tr>
<td>Release cycle time (days)</td>
<td>Y</td>
<td>1.4Y faster</td>
</tr>
<tr>
<td>(PR to deploy)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release frequency (days)</td>
<td>Z</td>
<td>3Z faster</td>
</tr>
</tbody>
</table>
Monolith vs Microservice
Service Onboarding without M-SaaS

- Create Service on Service Portal
- Get an Cloud Account - Work with Team A
- Create all the basic resources in AWS - Work with Team B
- Create Pipeline - Contact Team C
- Don’t forget you have to create the Git Repo - Team D
- Whitelist Nexus - Same Team D
- Set up CD …btw I used Spinnaker it is great!!! - This person helped me
- If you have Web server just talk to Team E
- Monitoring Alerting Logs….. - Contact yourself
- : 
- : 
- : 
- Take care of AWS SG, EC2, Auto scaling, R53, Ingress, Egress

Takes few weeks to set up and run the service

Steep learning curve

Multiple AWS Accounts

Take care of security patches periodically
Monolith vs Microservice
I want to develop new microservice

Service Onboarding - M-SaaS world

VELOCITY OPERABILITY SECURITY
| **VELOCITY** | • Ship with increased velocity  
• Significantly less coordination with other teams to ship features  
• Faster decision making |
| **OPERABILITY** | • Spend zero time managing infrastructure  
• Focus on E2E service ownership  
• Enables continuous operations by self remediation |
| **SECURITY** | • Hygiene and compliance baked into the platform  
• Zoning and containment  
• RBAC, Encryption, Logging |
| **HOW** | • Immutable systems  
• Declarative configuration  
• Decoupling the components  
• Abstraction of infrastructure |
Application Hosting Evolution
Monolith vs Microservice -> Containers
Containers for Production

● Docker originally
  ○ Standardise the container format
  ○ Focussed on simplifying the packaging and running the application on a single node.

● Enterprise quality application needs to
  ○ Scaling up or down the containers on demand
  ○ Keeping storage consistent with multiple instances of an application
  ○ Distributing load between the nodes
  ○ Launching new containers on different machines when something fails
Kubernetes aims to support an extremely diverse variety of workloads, including stateless, stateful, and data-processing workloads. If an application can run in a container, it should run great on Kubernetes.

Kubernetes is comprised of a set of independent, composable control processes that continuously drive the current state towards the provided desired state.

What is NOT

Kubernetes is not a traditional, all-inclusive PaaS (Platform as a Service) system.

Does not deploy source code and does not build your application.

Does not dictate logging, monitoring, or alerting solutions.

Does not provide application-level services, such as middleware (e.g., message buses), data-processing frameworks (for example, Spark), databases (e.g., mysql), caches, nor cluster storage systems (e.g., Ceph) as built-in services.
Jenkins Pipeline on IBP2
Gitops Flow

1. Commit Code
2. webhooks triggers Jenkins Blue Ocean
3. maven builds springboot
4. maven builds docker container
5. maven uploads docker image to Artifactory
6. Jenkins updates image in k8s
7. argo cd sync
8. argo creates manifests from k8s and deploys to k8s
9. Deployments contain:
   1. Deployment object
   2. Ingress/ALB
   3. Service
   4. Replica Set
Argoproj is a set of tools for scheduling and orchestrating work on Kubernetes.

- **Argo Workflows** - Container-native Workflow Engine
- **Argo CD** - Declarative GitOps Continuous Delivery
- **Argo Events** - Event-based Dependency Manager
Argo Workflows - Workflow Engine for Kubernetes

What is Argo Workflows?

• K8s controller and workflow CRD
• Every workflow step is a pod
• DAG and Step based DSL
Some Statistics

- Multiple business segments
- Thousands of developers onboarded
- Hundreds of Kubernetes Clusters
- Thousands of Nodes
- 22,000 Compute Cores
- 2,000 Namespaces
- 21,000 pods
Some Statistics (cont…)

- Hundreds of Services
- 700 deployments per day
# Some Statistics

<table>
<thead>
<tr>
<th></th>
<th>Expected Improvement</th>
<th>Actual Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recoverability</td>
<td>X</td>
<td>10X faster</td>
</tr>
<tr>
<td>Release cycle time (days)</td>
<td>Y</td>
<td>1.4Y faster</td>
</tr>
<tr>
<td>(PR to deploy)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release frequency (days)</td>
<td>Z</td>
<td>7Z faster</td>
</tr>
</tbody>
</table>
Intuit Open Source Softwares

Argoproj

Adobe  BLACKROCK  CoreFiling  Canva  Cratejoy

Interline  CYRUS BIOTECHNOLOGY  DATADOG

equinor  Google  Gladly  GitHub

Interline  intuit

KintoHub  Localytics  NVIDIA  Preferred Networks  SAP Hybris

Styra  QuantaBio

Karate

Web-Services Testing Made Simple.

https://github.com/intuit/karate
Thank You

laks_s99

in/laks1/

GREAT INTERNATIONAL DEVELOPER SUMMIT™
Save the Date for 2020!
April 20-24 | Bangalore, India

Join the mailing list for the latest news, offers, ticket sales and much more!

www.developersummit.com