**Deployment for Paylah app of DBS Bank**

**Group 1 Project by Bing, Cliff and JT**

1. **Deployment Services: Cloud Formation**

**CloudFormation is good for:**

CloudFormation is a powerful tool if you know how to use it and you need granular control over your AWS stack.

It’s not the easiest to use, but it opens up a lot of avenues for those who’ve mastered it.

It tends to suit:

* Teams with a lot of workloads who need a way of streamlining provisioning
* Teams with complex applications who need more control over their stacks
* Teams who need to manage their resources more closely for cost or security reasons
* Teams that have dedicated DevOps engineers ([or a good DevOps as a service partner](https://www.justaftermidnight247.com/devops/)) to be the managers and gatekeepers of CloudFormation

It can also be described as infrastructure automation or Infrastructure-as-Code (IaC) tool and a cloud automation solution because it can automate the setup and deployment of various Infrastructure-as-a-Service (IaaS) offerings on the AWS CloudFormation supports virtually every service that runs in AWS. (A full list of supported services is available here.)

You can use CloudFormation to automate the configuration of workloads that run on the most popular AWS services, like the EC2 compute service, the S3 storage service, and the IAM service for configuring access control.

## **How it works – CloudFormation**

The user writes a YAML or Json template specifying the AWS services used.

Once uploaded, CloudFormation can then create and deploy the specified resources.

AWS resources collected together to run one application are known as a stack, and it’s possible to reuse your template to deploy that stack in multiple environments.

### **Pros**

* Brings code review features to infra – because it’s IaC, you can now review your infra changes as you would code changes, allowing for greater oversight ([often used in DevOps](https://www.justaftermidnight247.com/insights/what-is-devops-as-a-service-the-really-big-and-fun-guide/))
* Granular control – the main benefit of CloudFormation is offering the experienced user granular control and oversight of their AWS stack
* Large community support – although it’s complex, CloudFormation is an established tool tool with a big community and resource base
* Security - AWS CloudFormation helps to ensure that all AWS resources are configured in a secure manner by using security policies and rules. This helps to protect the infrastructure from potential threats.

1. **Deployment Strategy: Canary Deployment.**

What makes developing a mobile banking app unique?

Banking as an industry sounds very traditional, but the opposite is true for digital banking. If you’re a banking institution today, you know that your customers want access to a sleek app that’s enjoyable, fast, and secure. And when you put those customer needs together with what banking apps are functionally designed to accomplish, you get three characteristics that make the process of developing these apps unique from other types of apps.

1. Usage rate is extremely high

When you give banking customers the option to check their account balance anytime they want, they will do that. Several times a day, even. Multiply that by multiple thousands of users and you have the pressure of daily active users that cause technical challenges that simply don’t exist on apps with less rigorous usage.

2. Banking customers have strong expectations

Due to the frequency of use and the availability of so many competing mobile banking options, customers demand for a high level of in-app user experience. At the very least, a banking app needs to be fast-loading and simple to use. Beyond that, product teams who support these apps must be wary of any moment there is any type of friction, because users will feel it immediately and will typically be very vocal about it.

## **How Canary Deployments Work**

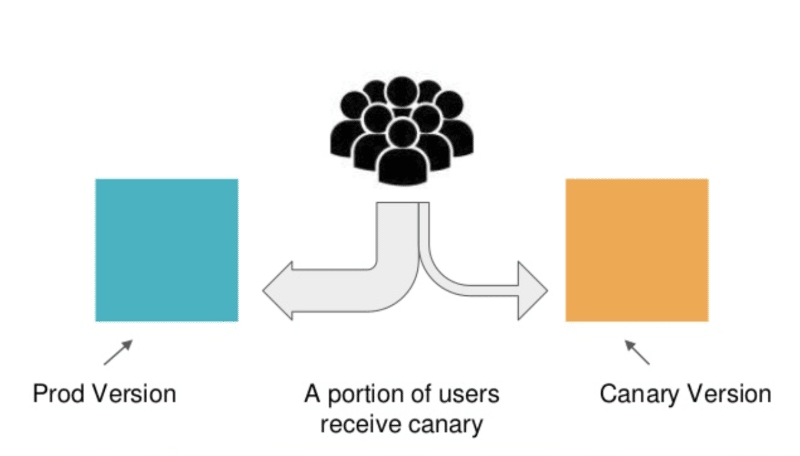
Here is a general process for canary deployments:

1. In the beginning, the current version receives 100% of user traffic
2. A new deployment, the “canary” is performed with brand new pods and only a small amount of traffic, e.g. 5% while maintaining 95% of users on the older version.
3. Different types of tests (e.g. smoke tests) can be performed on the new version with no impact on the bulk of the users.
4. A decision regarding the current canary/subset of traffic takes place in an automated manner.
   * If the new version works as expected, a larger portion of live traffic is sent to the new version and the process repeats again for different percentages of canary traffic (e.g. 5%, 25%, 50%, 75%, 100%).
   * If the new version has issues, the service is switched back to the original version. This has minimal impact on most users. The canary version is destroyed and everything is back to the original state.

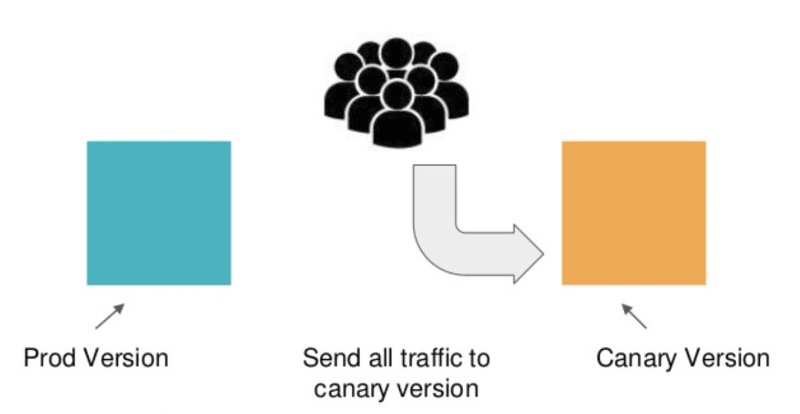
5. In the end, 100% of traffic goes to the new version and the old version can be discarded.

## **A Visual Example of Canary Deployments**

With canaries, the new version of the application is gradually deployed, initially receiving a small subset of live traffic. Only a small number of live users view the new version while the rest continue using the current version.



The small subset of live users acts as an early warning for potential problems in the new version. As confidence increases, the canary version is scaled up and more users are allowed to view the new version. Eventually, all live traffic goes to the new version, and thus the canary version becomes the new production version.



## **Canary Deployment Benefits**

Here are three key benefits of canary deployments:

* Capacity testing – when deploying a new microservice to replace a legacy system, it is useful to be able to test in a production environment how much capacity you’ll need. By launching a canary version and testing it on a small fraction of your users, you can predict how much capacity you’ll need to scale the system to full size.
* Early feedback – many issues that affect end-users only occur in a production environment. Canary deployments can expose features to users in a realistic environment, to observe errors or bugs and obtain user feedback. This allows quick feedback from users, allowing developers to add new features and deliver what the end-user needs. This helps improve the software and the user experience.
* Easy rollback – in a canary deployment, if any severe issues are detected, rollback is instantaneous. It is just a matter of switching traffic back to the primary version or adjusting a feature flag.