



SUMMIT  
ONLINE

B U I O 8

# Kubernetes GitOps on AWS

Jason Umiker

Solutions Architect  
Amazon Web Services

# Has any of this happened to you?

# The deployment just failed

But it worked on my machine / in the X environment!



# All of the sudden there's a problem...

Did anyone just change anything? Who? When? What?

Development



Operations, Network,  
DBAs, etc.



# All of the sudden there's a problem...

We've lost the server(s)/environment!?! We have documentation on how to rebuild it right?





Things just seem generally



# What is GitOps?

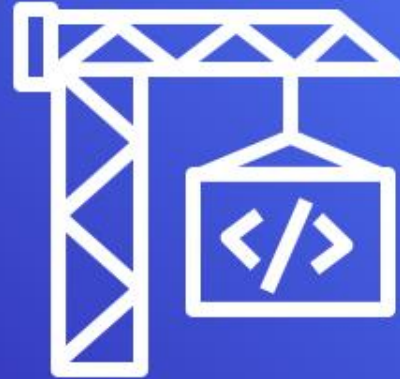


# Modern software development

Traditionally there have been three key activities/systems in managing change



Changes to  
the code



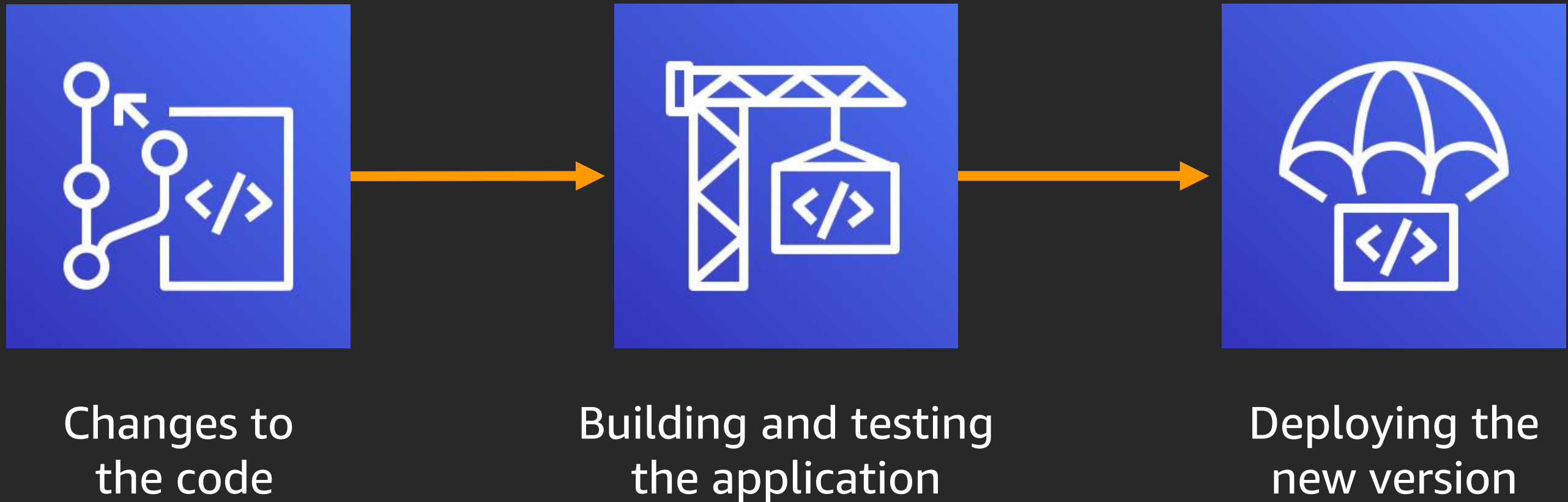
Building and testing  
the application



Deploying the  
new version

# CI/CD

Automating that on success is called Continuous Integration / Continuous Deployment (CI/CD)



# Application code vs. infrastructure

We used to have bespoke and long-lived hardware that we deployed the code to



Application code

vs.



Infrastructure

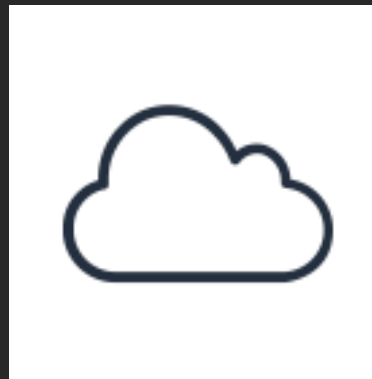
# Application code vs. infrastructure

But with the cloud we don't need to worry about hardware and can ask for things as 'code'



Application code

vs.



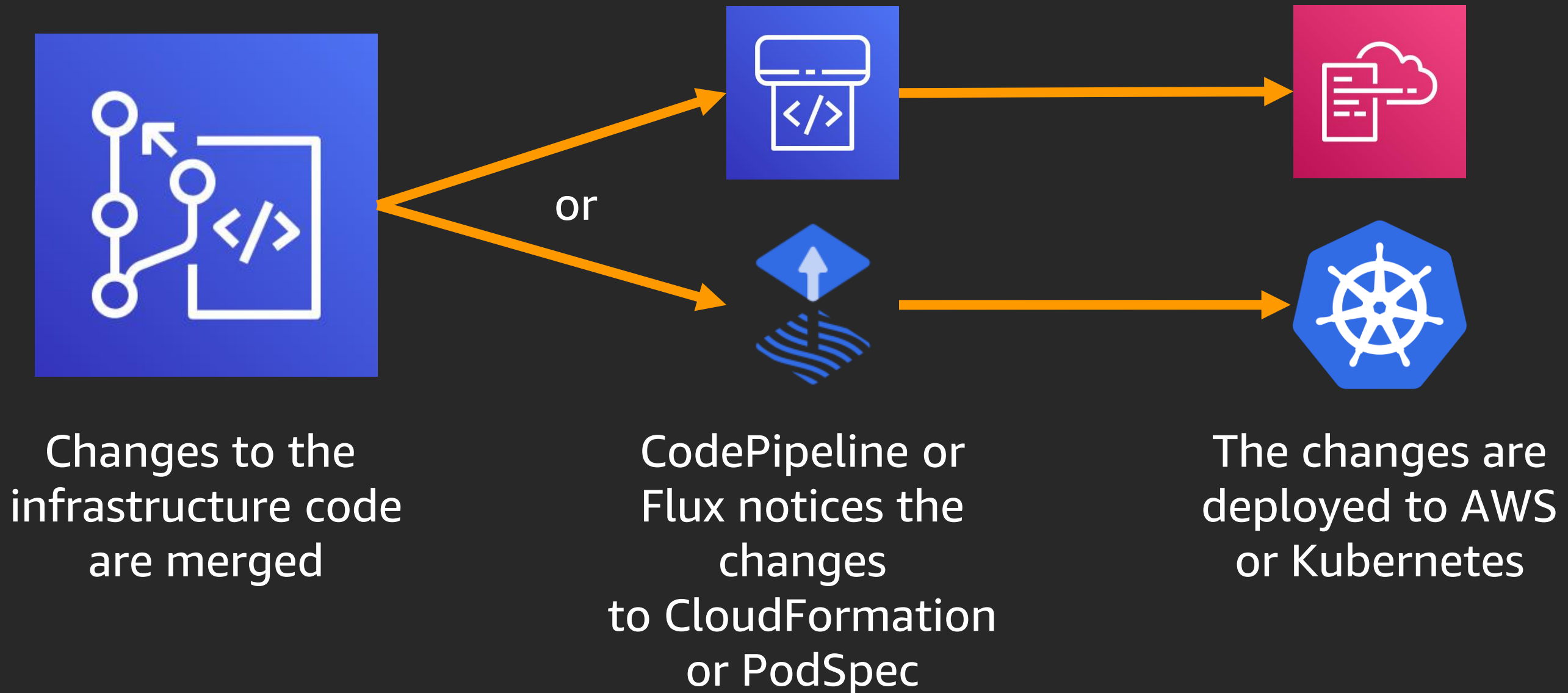
JSON/  
YAML/  
CDK Code

Infrastructure



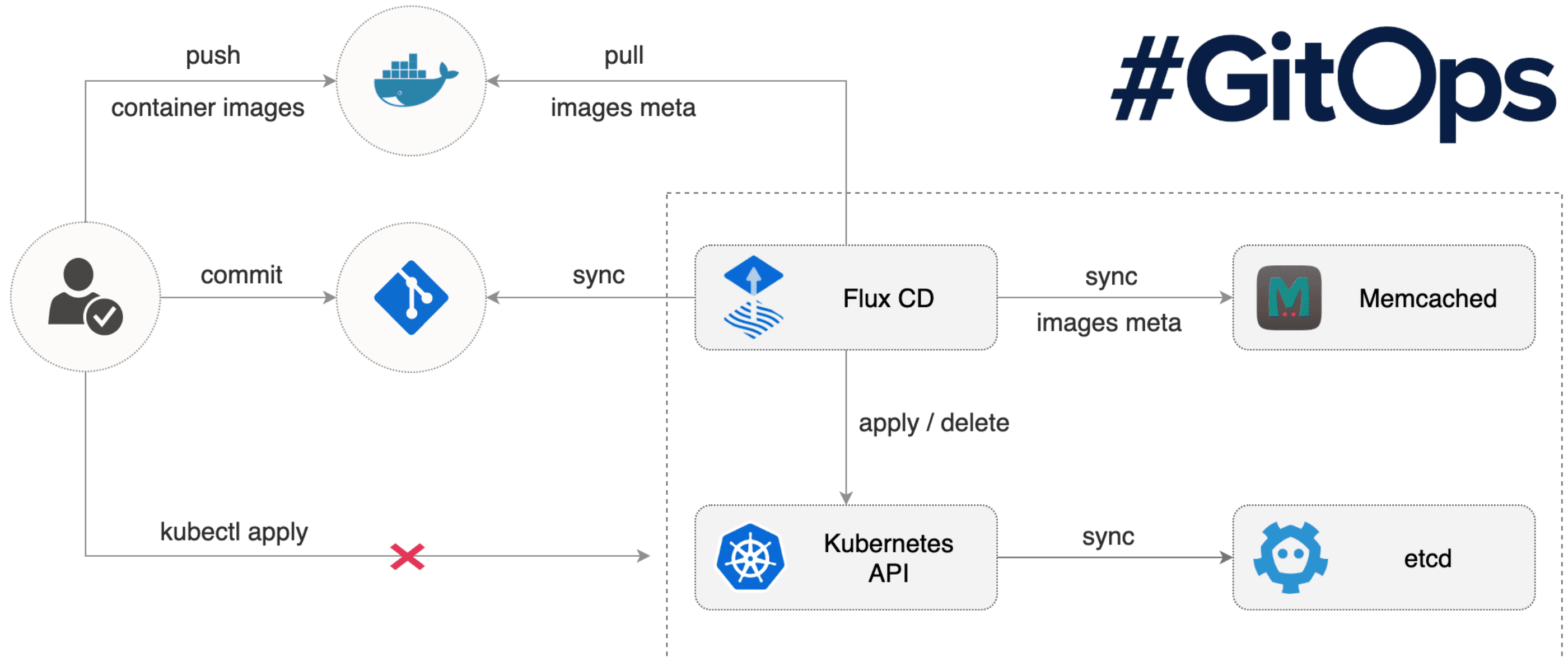
# GitOps

Can we apply git-driven CI/CD processes to infrastructure code?



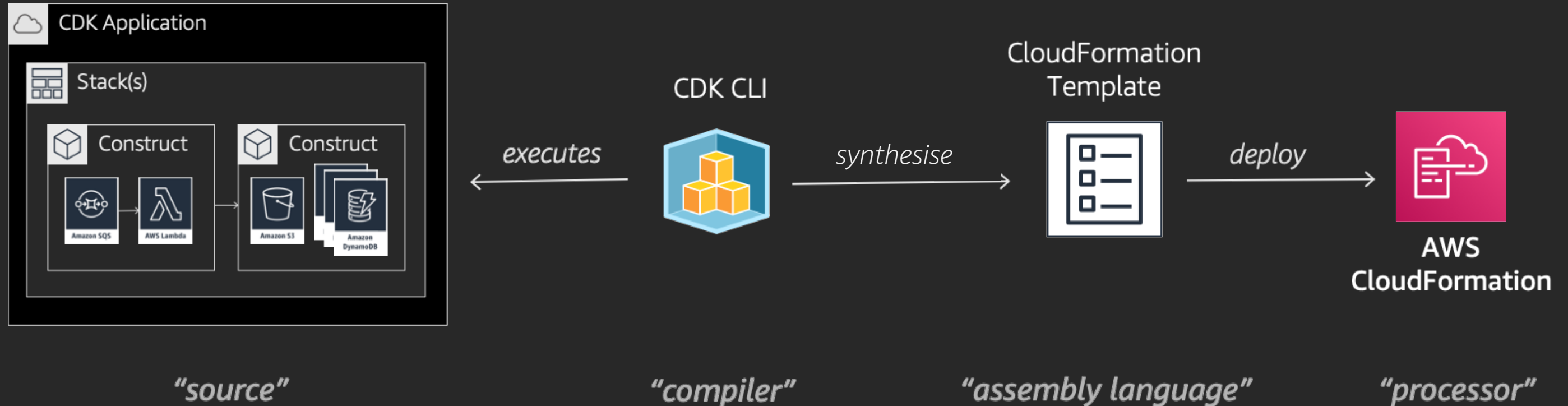
# What is Flux?

# #GitOps



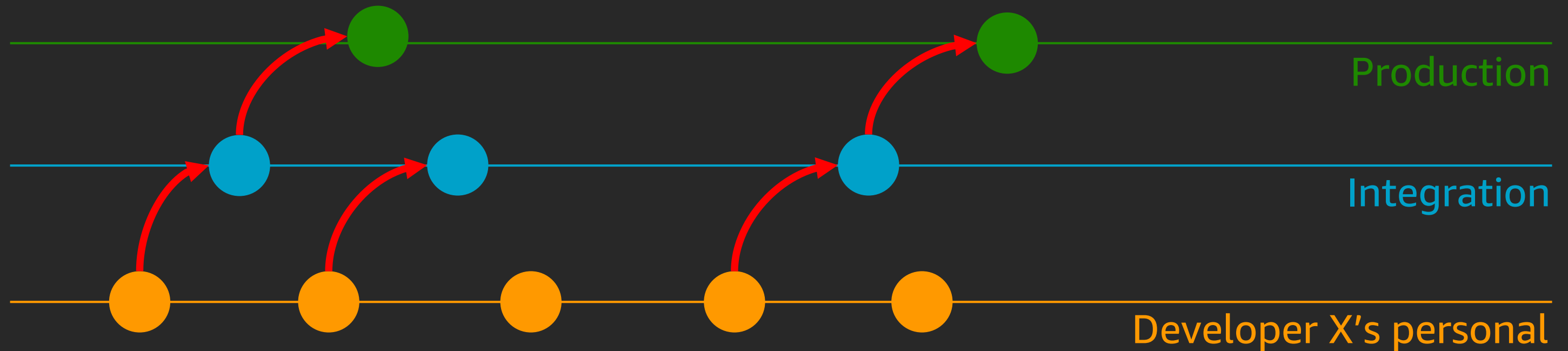
# What is the AWS Cloud Development Kit (AWS CDK)

The big picture - from AWS CDK app to CloudFormation to provisioned infrastructure



# What is GitOps?

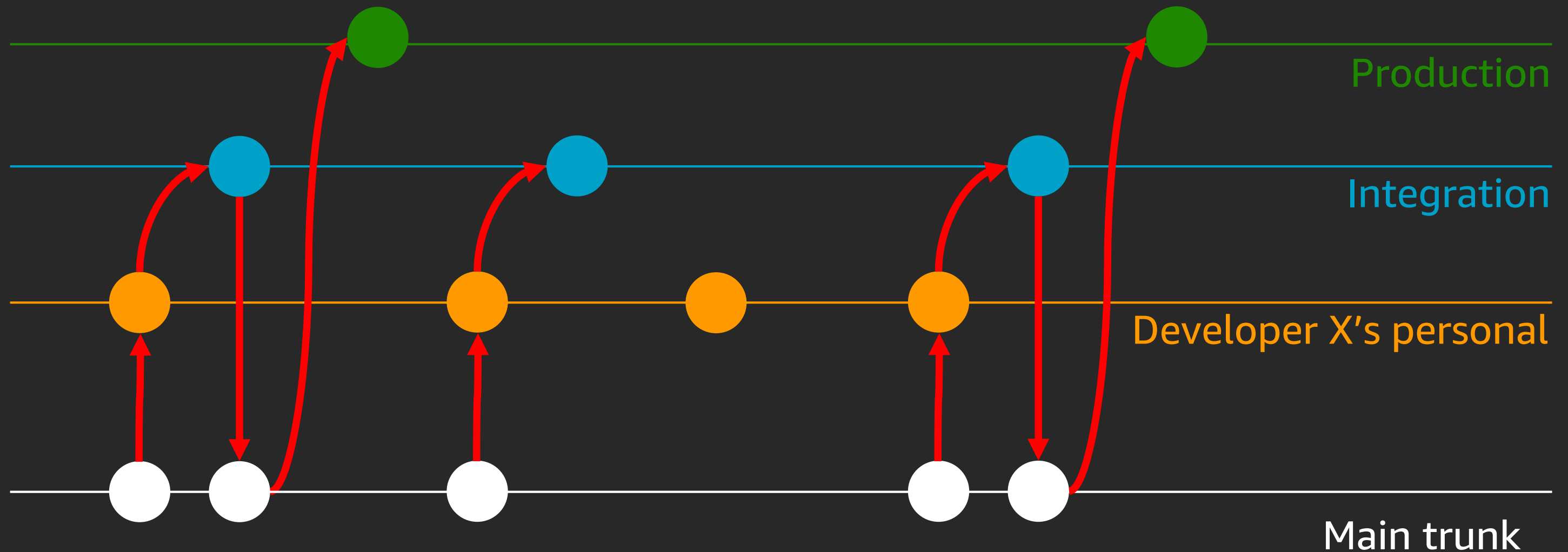
Control via PRs & environment branches (merge to them and it'll deploy there on successful tests)





# What is GitOps?

Control via PRs & environment branches (merge to them and it'll deploy there on successful tests)



# Why GitOps?

# Git is the tool that developers are already using

Instead of dev and ops using separate tools/workflows GitOps helps them play nicely together



Management

Ops

Infra code

Devs

App code

git

# Git is a great source of truth

It keeps track of the full history of changes serving as a great auditing and reporting tool



Who

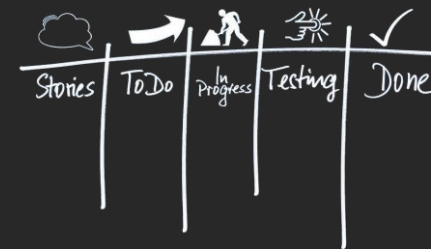
```
/**
 * Simple HelloButton() method.
 * @version 1.0
 * @author John Doe <doe.j@example.com>
 */
HelloButton()
{
    JButton hello = new JButton( "Hello, wor
    hello.addActionListener( new HelloBtnList

    // use the JFrame type until support for t
    // new component is finished
    JFrame frame = new JFrame( "Hello Button"
    Container pane = frame.getContentPane();
    pane.add( hello );
    frame.pack();
    frame.show();           // display the fra
}
```

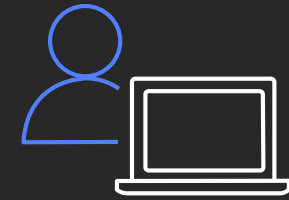
What



When



Why



Who



# Git is a great source of truth

It keeps track of the full history of changes serving as a great auditing and reporting tool



Who

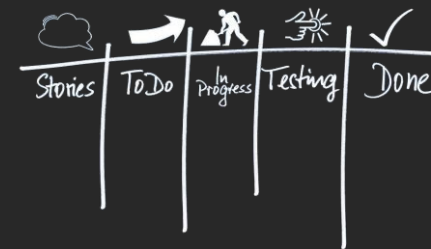
```
/**
 * Simple HelloButton() method.
 * @version 1.0
 * @author John Doe <doe.j@example.com>
 */
HelloButton()
{
    JButton hello = new JButton( "Hello, wor
    hello.addActionListener( new HelloBtnList

    // use the JFrame type until support for t
    // new component is finished
    JFrame frame = new JFrame( "Hello Button"
    Container pane = frame.getContentPane();
    pane.add( hello );
    frame.pack();
    frame.show();           // display the fra
}
```

What



When



Why



Who

**Enforced peer review (via pull request) of every change!**

# Git is a great place to control change

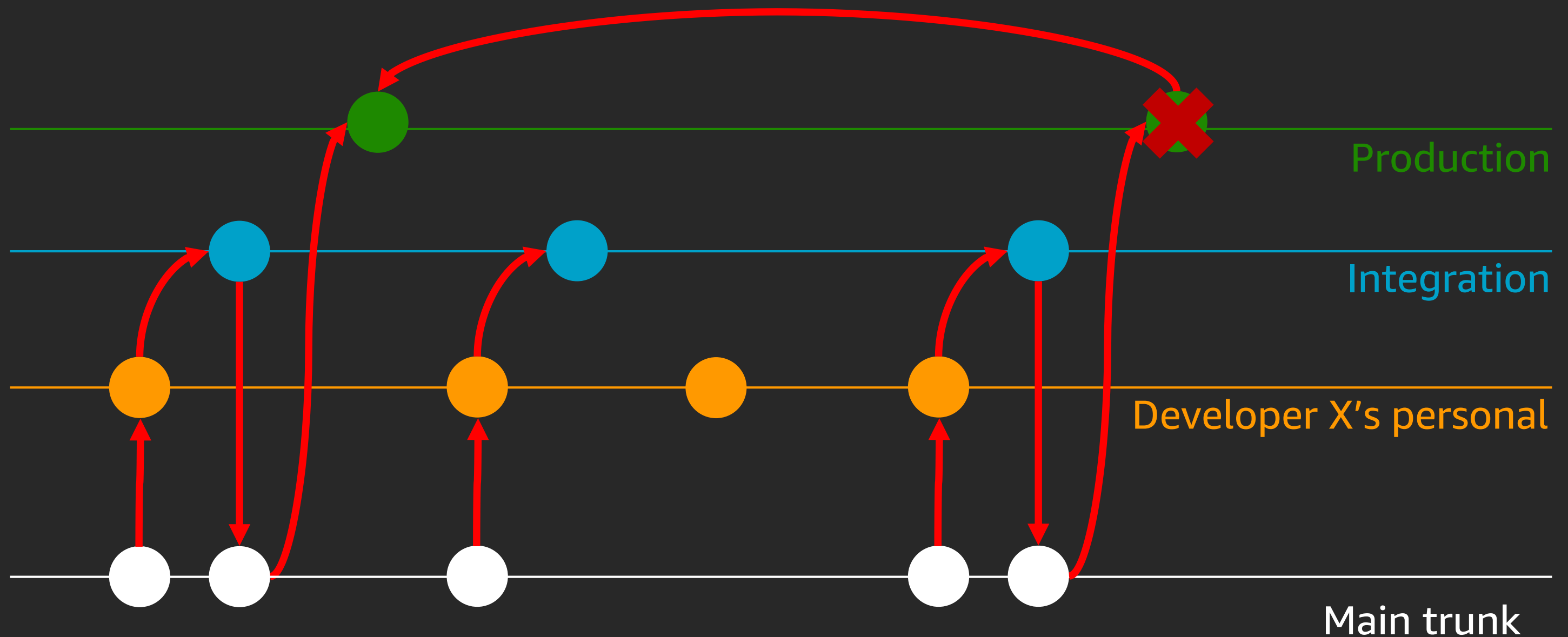
It is easier to control change if you force it all through one entry point to one pipeline



Requiring approved **pull requests** leading to mandatory **security tests** via a **DevSecOps pipeline** as the only way in!

# Git can help with rollbacks

Because you have the full change history you can just re-deploy older commits to the trunk\*



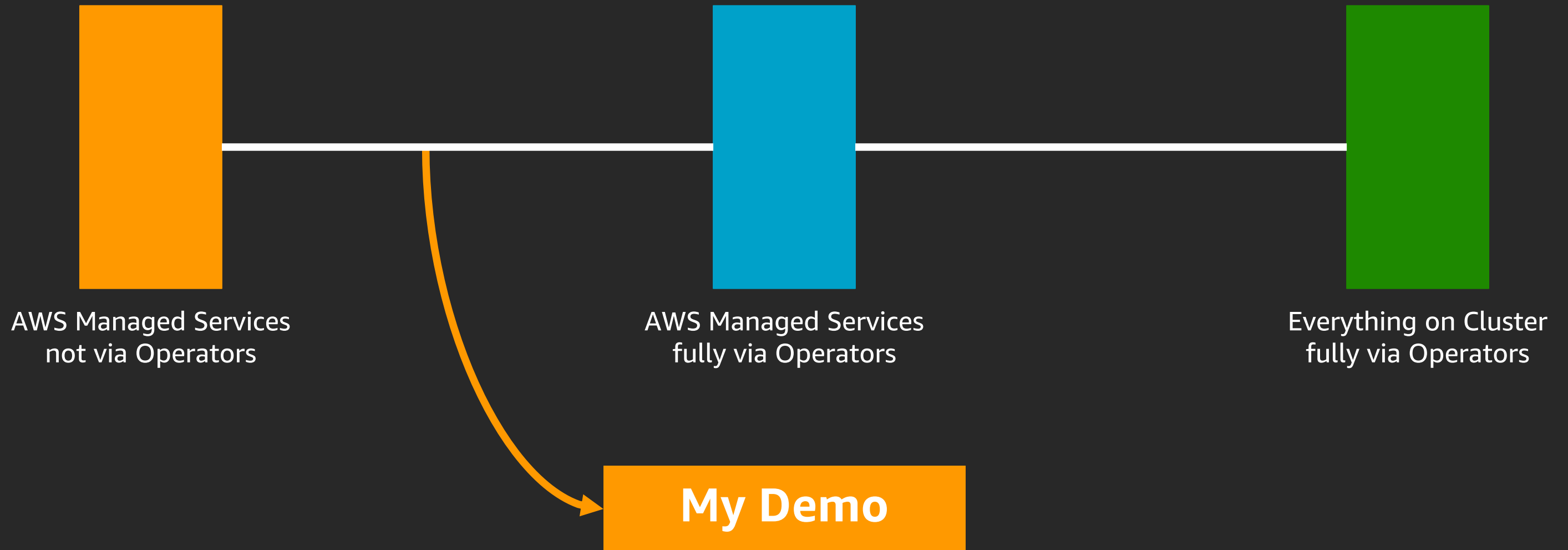
\*some architecture/thought needs to go into things to allow for this re: DB schema change etc.

# Cloud vs. Kubernetes management



# CDK/CloudFormation vs. Kubernetes Operators

It's a spectrum!

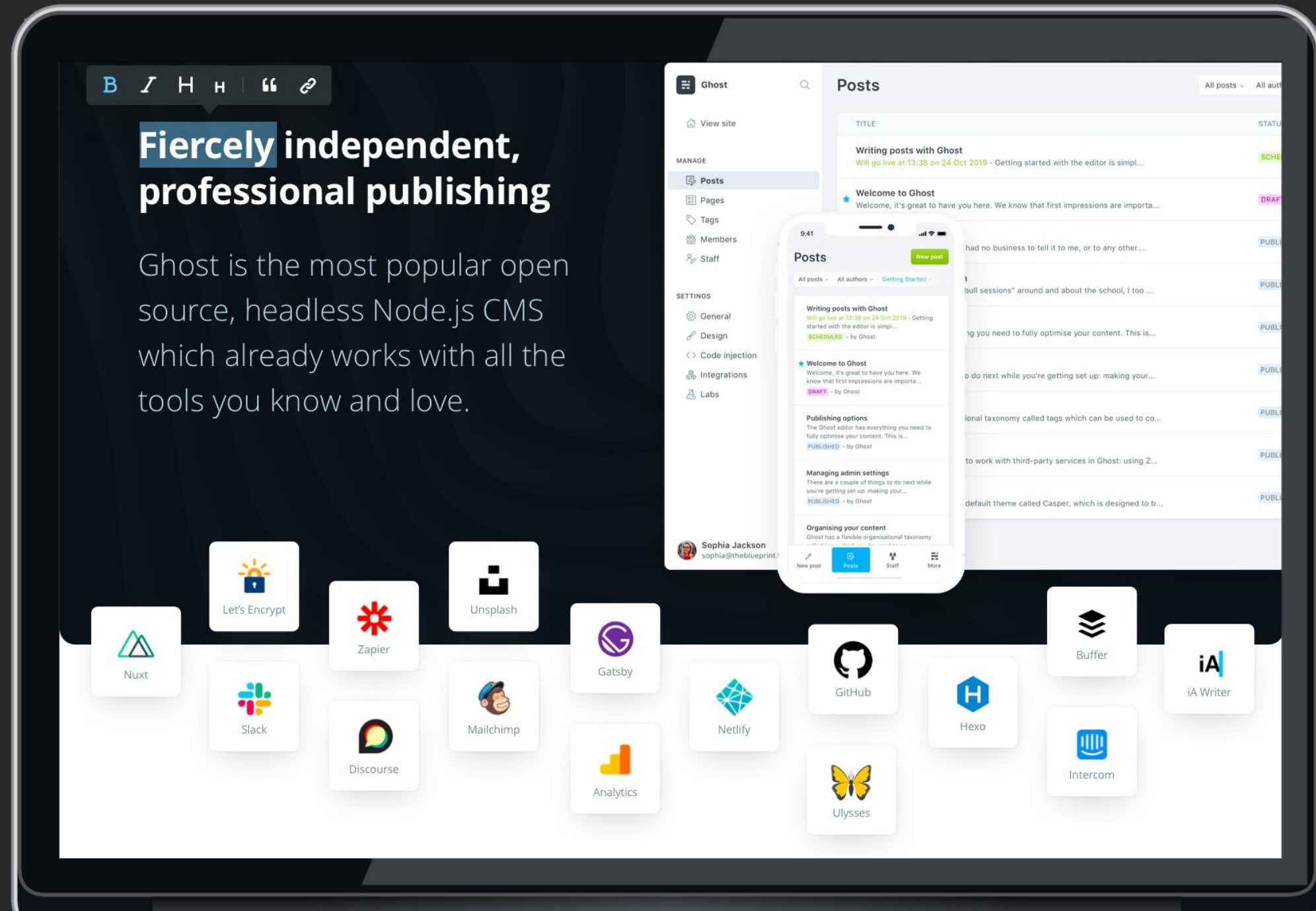


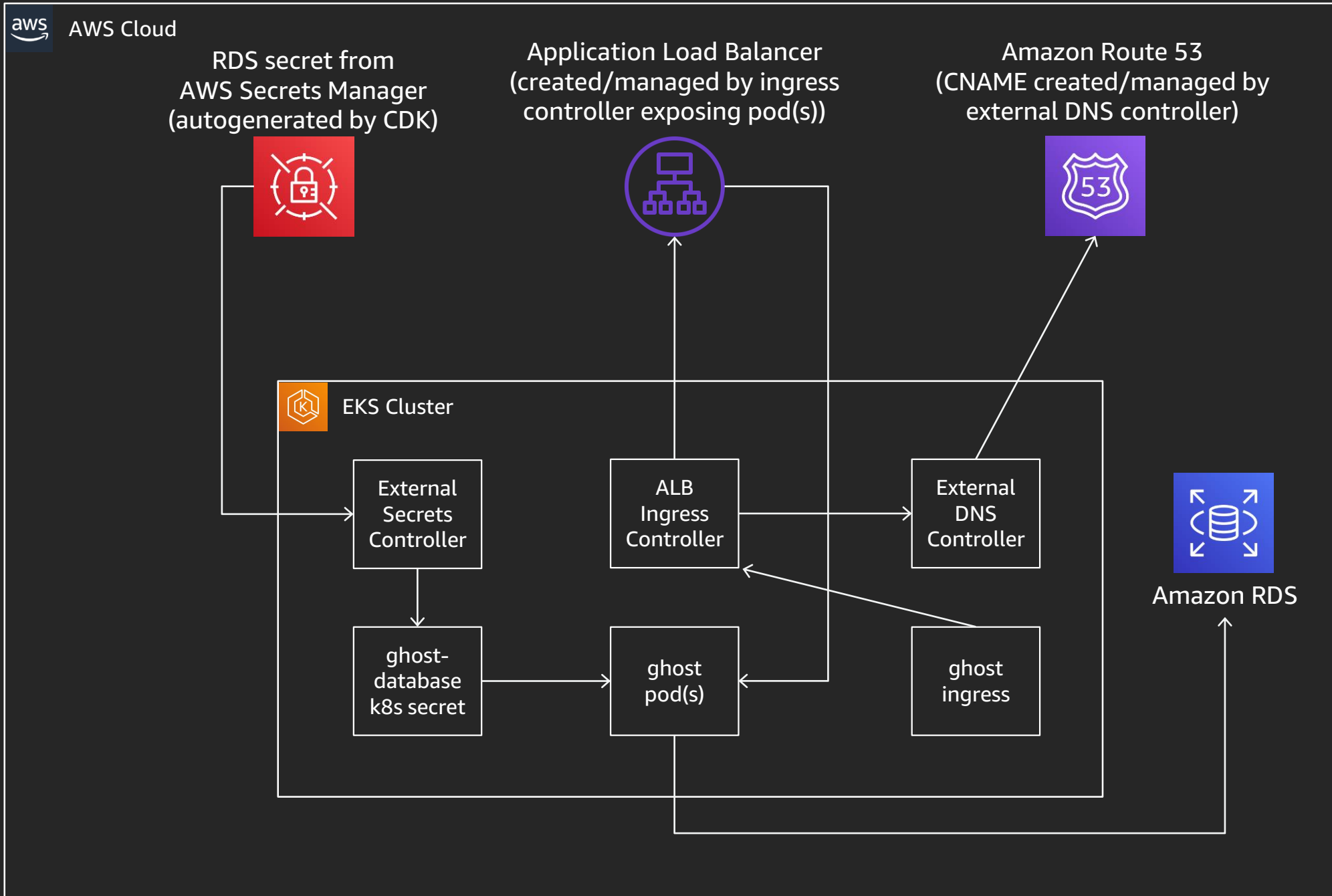
# Demo

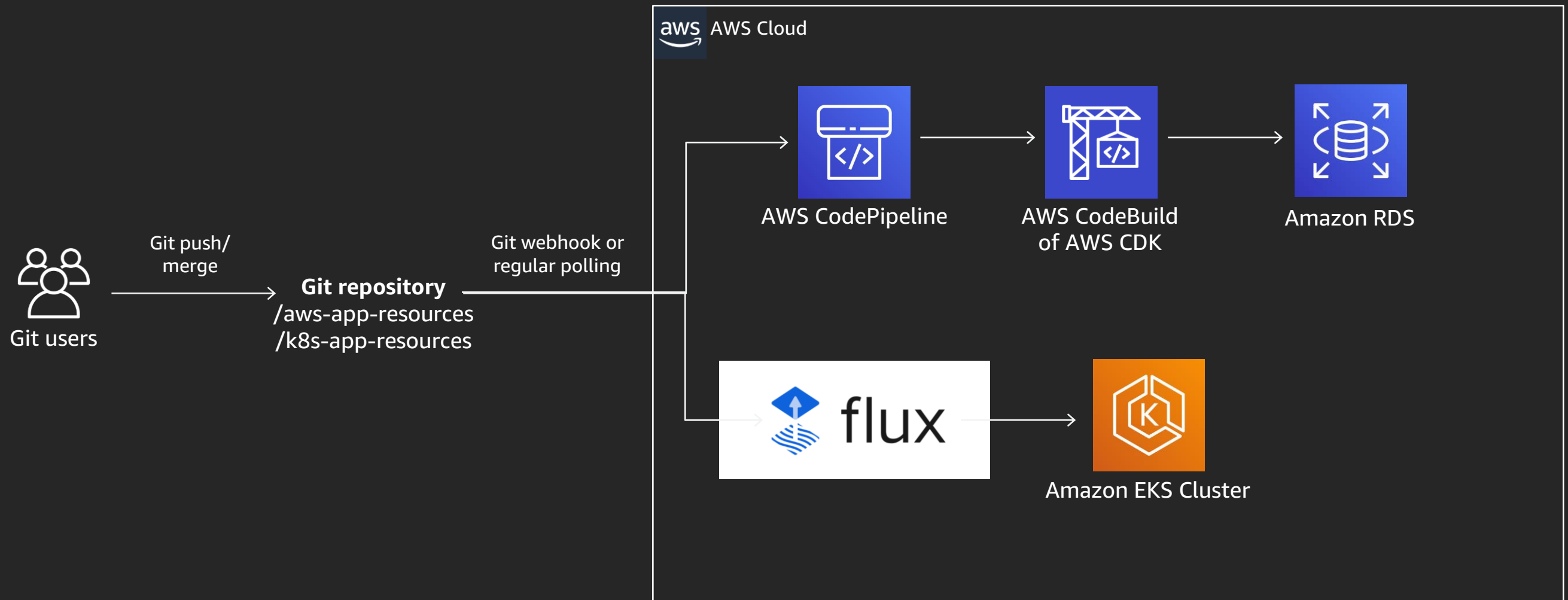
# Our Demo app is Ghost

Ghost is a good example as it is both opensource and distributed as a 12-factor Docker image.

It is stateless (storing all its state in a MySQL database) and gets its environment settings from environment variables at runtime.







<https://github.com/jasonumiker/k8s-plus-aws-gitops>



# In closing

Go back to the office and try GitOps so that:

- Your app works in every environment
- Your developers and operations play nice together with the same tool
- You know the who, what, when, why (and who approved) of every change
- You are in control of your environments and changes to them
- You can roll back to a known, and self-documenting, earlier state

**GO BUILD!**

# Thank you!