Kubernetes SriLanka

CI/CD Strategies with Kubernetes

Binura Gunasekara
Kubernetes Sri Lanka / May 2020

About me

Binura Gunasekara

@bgxcode

Senior Software Architect
Platformer Cloud (Sydney)

platformer™

Understanding CI/CD

Basics first!

Continuous Integration and Continuous Delivery / Deployment

What is CI/CD?

- Not an isolated technique (or set of technologies)
- More often than not, the effectiveness reflects the structure of your Organization (ie. Conway's Law)
    - "DevOps"
- Requires a whole other substrata of tech to keep up with the "aggressiveness" of CI/CD.
    - "DevOps tools" (prefer to call it **SRE** - Site Reliability Engineering)
        - Monitoring
        - Alerting
        - Centralized Log Aggregation and Ingestion

Le' Principles

Deploy to Production often. (Promote application images through 'Environments')

Fix issues as quickly as you find them (This is why you need monitoring and alerting).

100% test coverage ***!=*** *100% bug-free software*

Automate repetitive tasks (Building/Tests/Deployments/Scaling/etc.)

A Generic CI/CD Pipeline

Where does Kubernetes fit into this story?

Kubernetes can automate away some of your Operational nightmares.

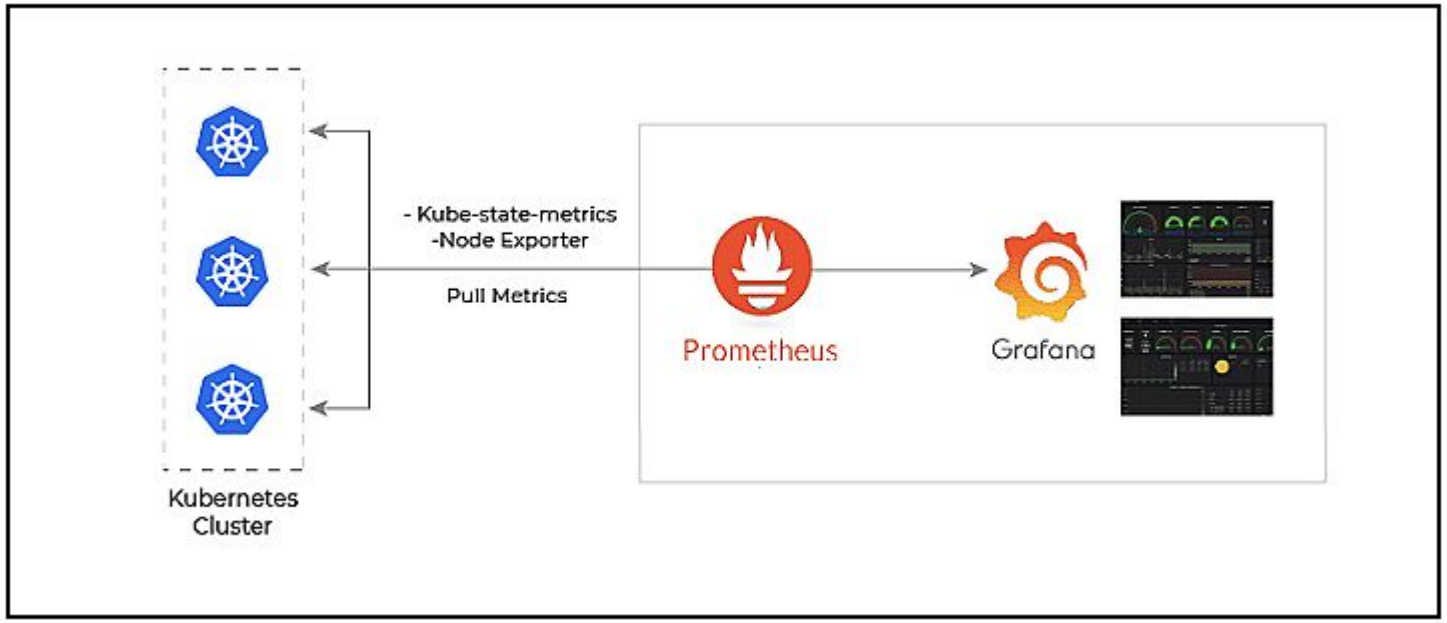
- Zero-downtime deployments (there are other advanced strategies like Blue/Green Deployments, Canary, etc.)
- Deployment Rollbacks
- Auto-scaling (based on Resource Usage or Ingress Traffic)
- Service Discovery
- Self-healing applications (health checks etc.)

What Kubernetes doesn't do

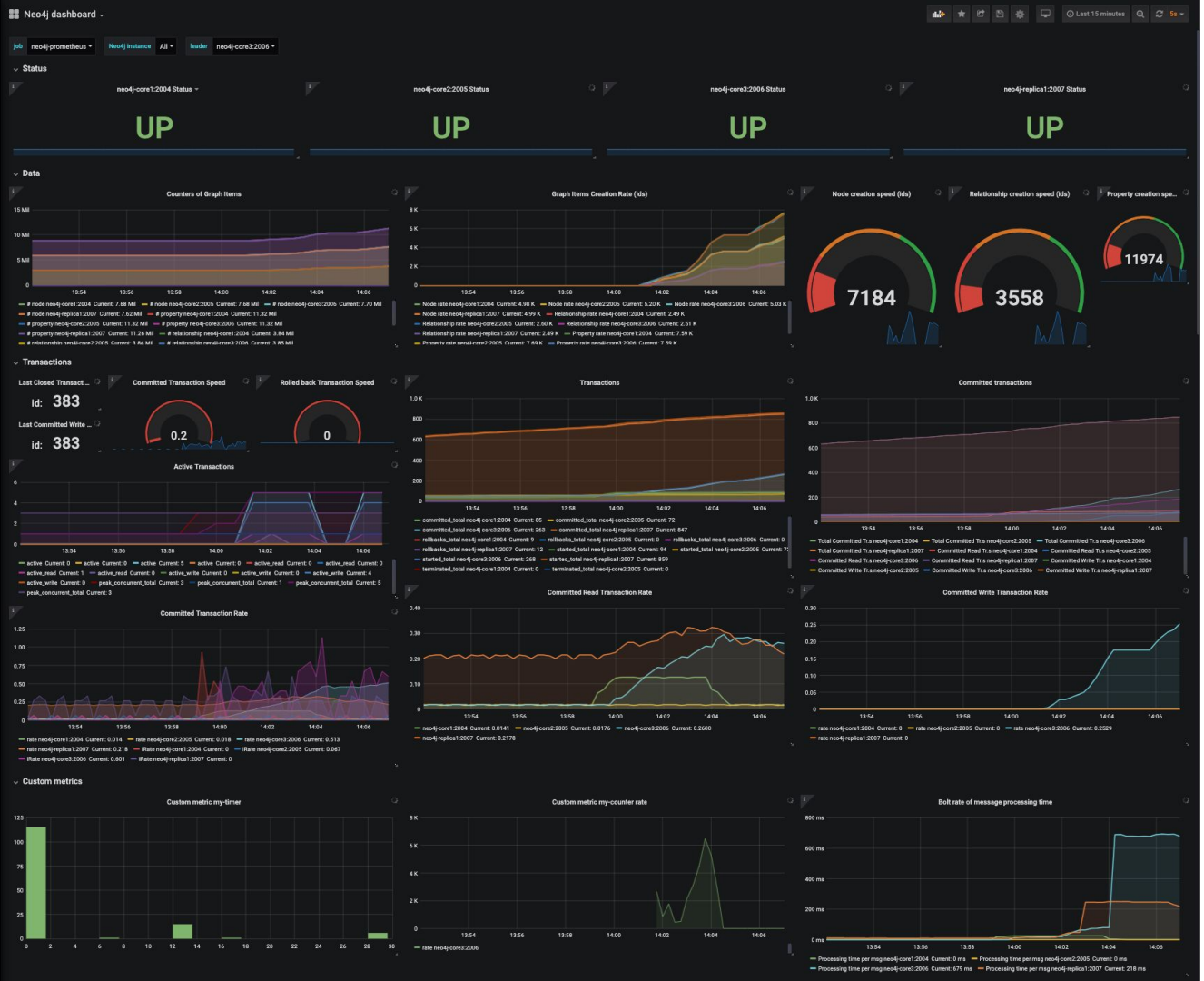
- K8s supports high velocity deployments - but doesn't provide a built-in CI/CD framework (outside the scope of what Kubernetes is supposed to be)
  - No monitoring/alerts
  - No archival logs (it does provide temporary log aggregation)
  - No deployment strategies (other than rollovers)

- The Cloud Native Ecosystem is where you look.

Monitoring & Alerts



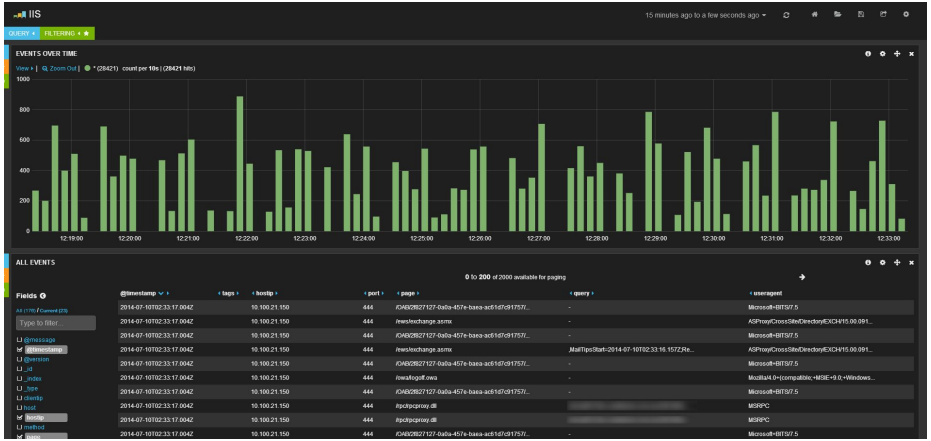
Alternatives - **Elastic Stack**



A Grafana Dashboard

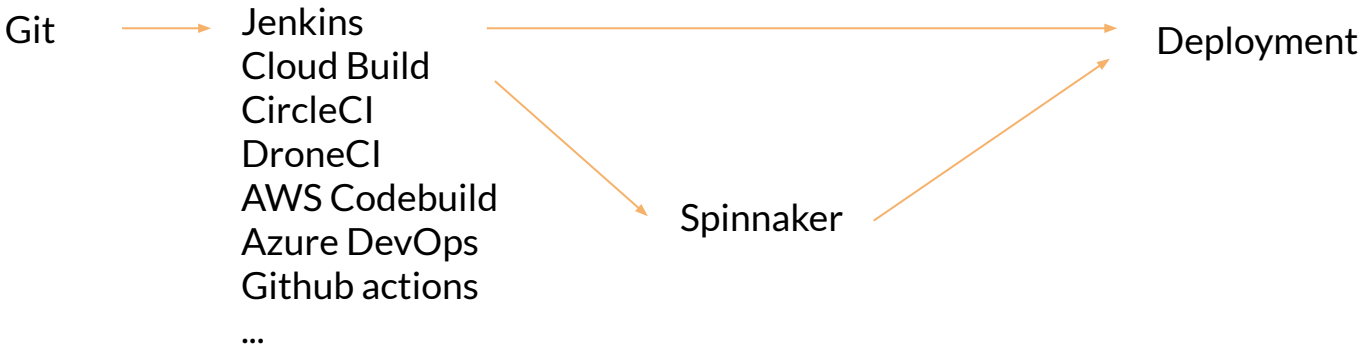
Logging

Splunk



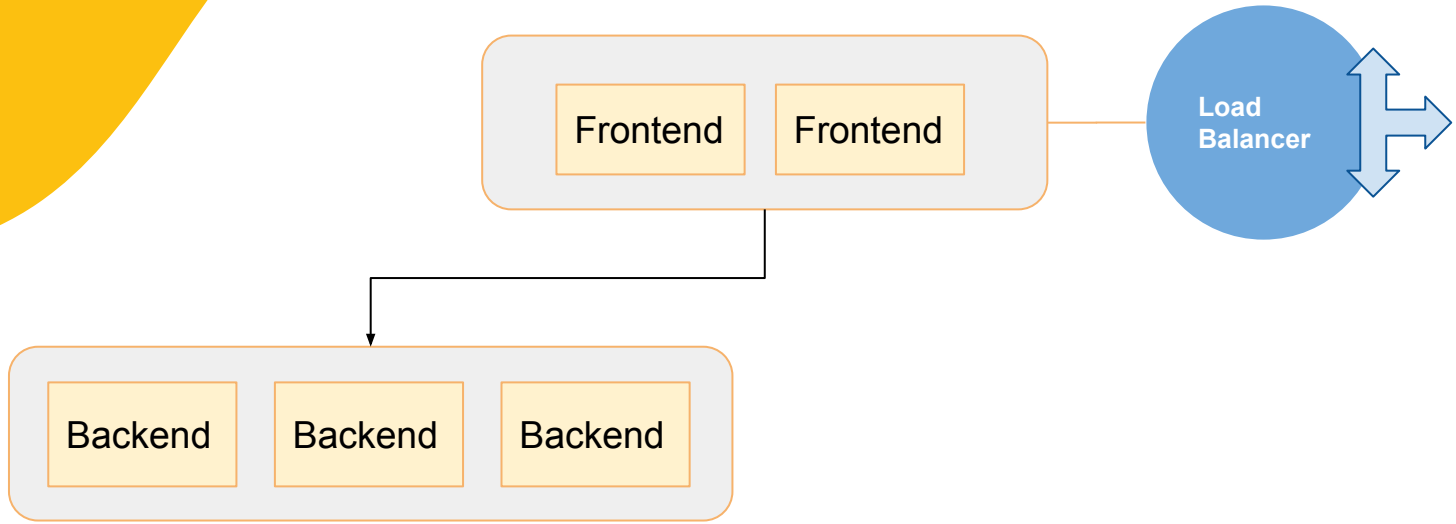
ELK
Stack

There’s an endless combinations of tools you can use to achieve the same purpose.


Git
Jenkins
Cloud Build
CircleCI
DroneCI
AWS Codebuild
Azure DevOps
Github actions
...
Spinnaker
Deployment

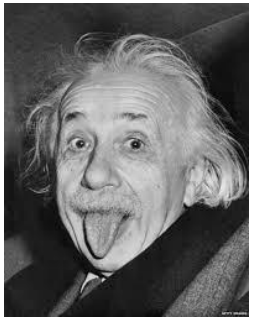
Time for a demo

JENKINS!


Frontend
Frontend
Load Balancer
Backend
Backend
Backend

“Too many tools will ruin the soup.”

- **Albert Einstein**



Key Takeaway

It’s important to remember to not get carried away and standardise toolchains across your Organization; **otherwise it will create information silos between teams and add to your technical debt.**

Kill all ‘em birds with one stone

Achieving CI/CD with minimum effort - Deploying on Platformer Cloud

Inb4 - shameless marketing pitch

Thank you!