TEKsystems Global Services

DevOps

Presented by: Siddharth Jinagouda, Anudeep Krishnadas

Date of Presentation: 25 - August - 2020

Agenda

- *Introduction – Origins of DevOps, Software creation and consumption*
- *Continuous Innovation, DevOps Model*
- *DevOps Stories, Case studies*
- *CICD - Continuous integration, Continuous Deployment*
- *DevOps Tools- GIT, Pipeline, Infrastructure as Code*
- *DevOps Architecture – Microservices, Docker, Kubernetes*
- *DevOps Monitoring*
- *Code*
- *Questions*

Where it all started..

Origins of DevOps

Auto Manufacturing

In the U.S., innovations in auto manufacturing optimized around speed of production at lower costs and assessed quality in the end.

With testing and integration saved for the end of auto production and software development, problems with quality went undetected and unused parts accumulated. In the 1950s, Japanese automakers started to assess quality earlier by applying innovative concepts during assembly.

- **1913:** Ford Motor Company invents the moving assembly line, increasing production by more than 87%.[1]
- **1926:** Buick opens the first unified assembly line, the most efficient car assembly system to date.[2]
- **1950:** Toyota starts using the andon cord concept, empowering line workers to pull the cord to address problems with quality mid-production.[3]

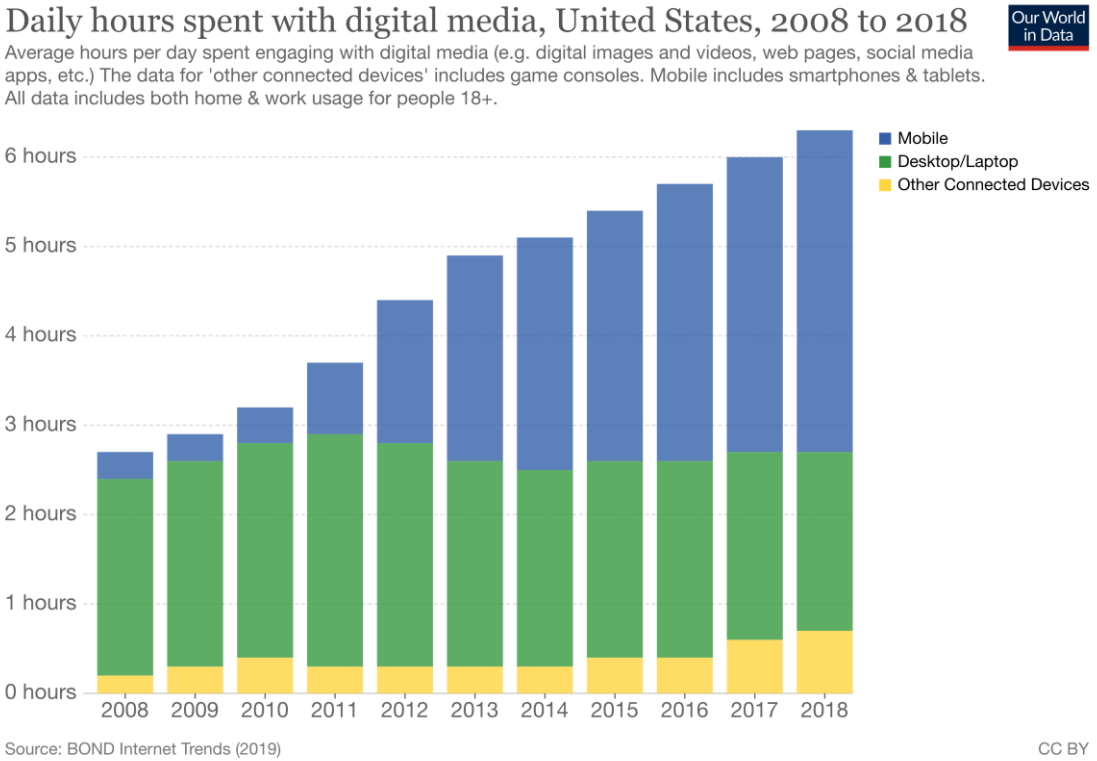
Cost

Speed

Quality

Alternative optimization: High quality, slower speed On average, an Aston Martin car takes 200 hours to manufacture,[4] compared to 31 hours for the average Toyota.[5]

Consumption



Consumption

65 Billion Messages sent everyday through Whatsapp – May 2018

95 Million posts on Instagram – June 2016.

1 Billion Videos watched daily - Youtube

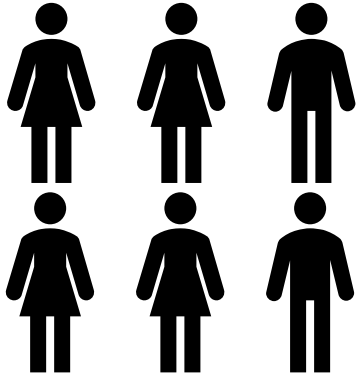
Time to Buy and Consume

Single touch

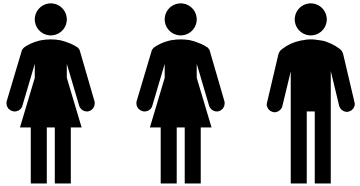
Creating and delivering software?

Traditional Teams - Silos

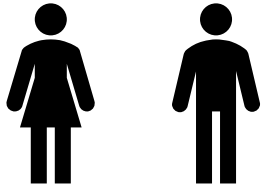
Development Team



Testing Team



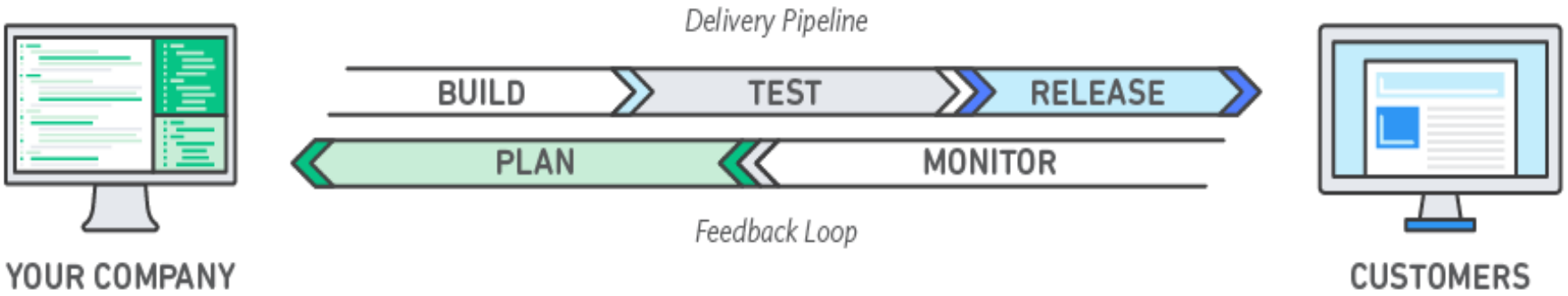
Operations Team



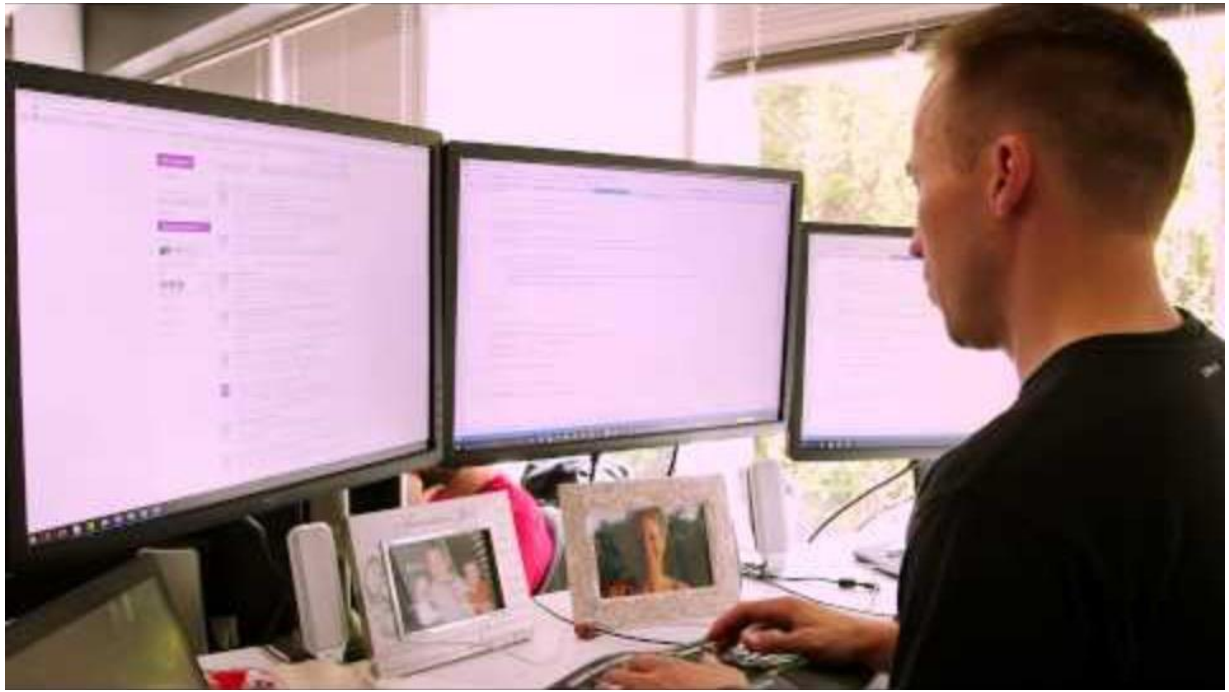
Continuous Innovation

DevOps Model

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability for Continuous Innovation.



DevOps Stories



DevOps Practices

- Continuous Integration
- Continuous Delivery
- Microservices
- Infrastructure as Code
- Monitoring and Logging

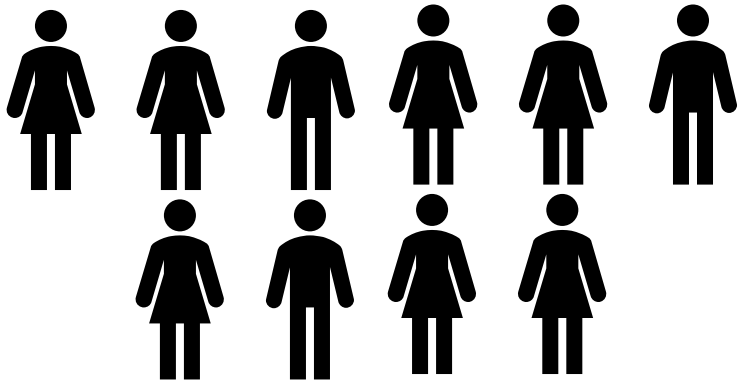
DevOps Practices - Continuous Integration

Continuous integration is a software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.

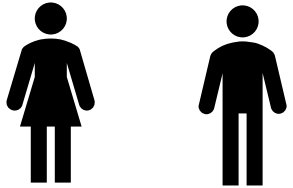
Automation frameworks for Tests – Selenium, Cucumber, Protractor

DevOps Practices - Continuous Integration Teams

Engineering Teams



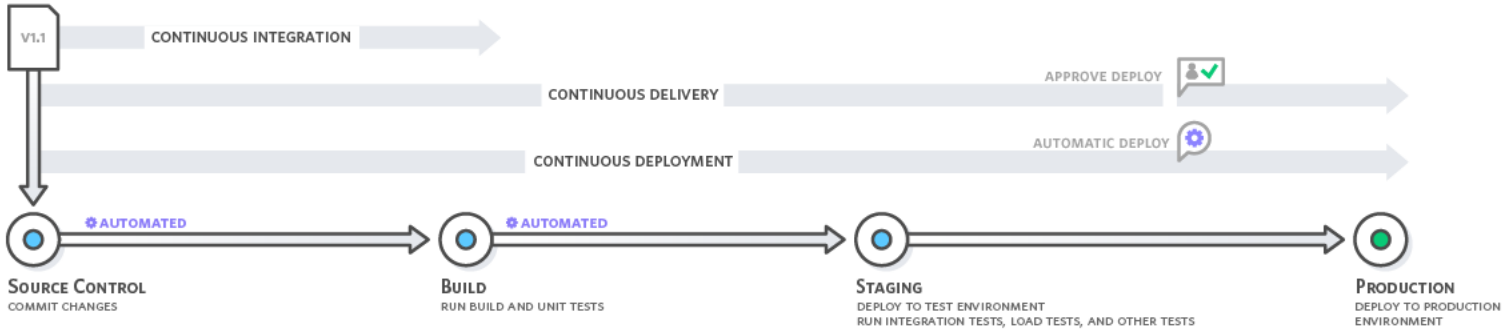
Operations Team



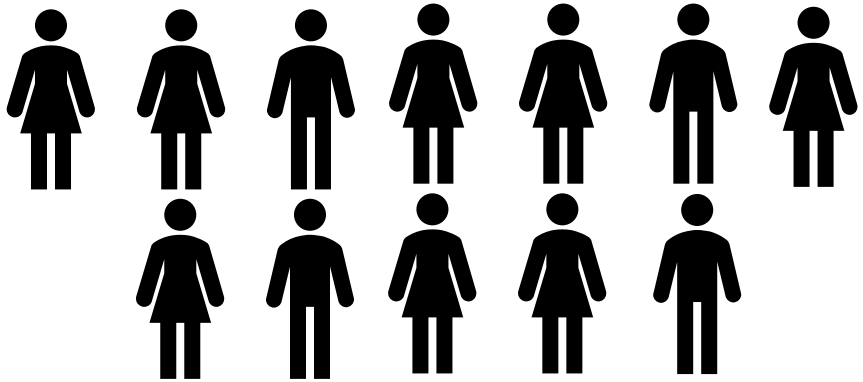
DevOps Practices - Continuous Delivery

Continuous delivery is a software development practice where code changes are automatically prepared for a release to production.

Continuous delivery expands upon Continuous Integration by deploying all code changes to a testing environment and/or a production environment after the build stage.



DEVOPS Teams



DevOps Tools - Version Control - GIT

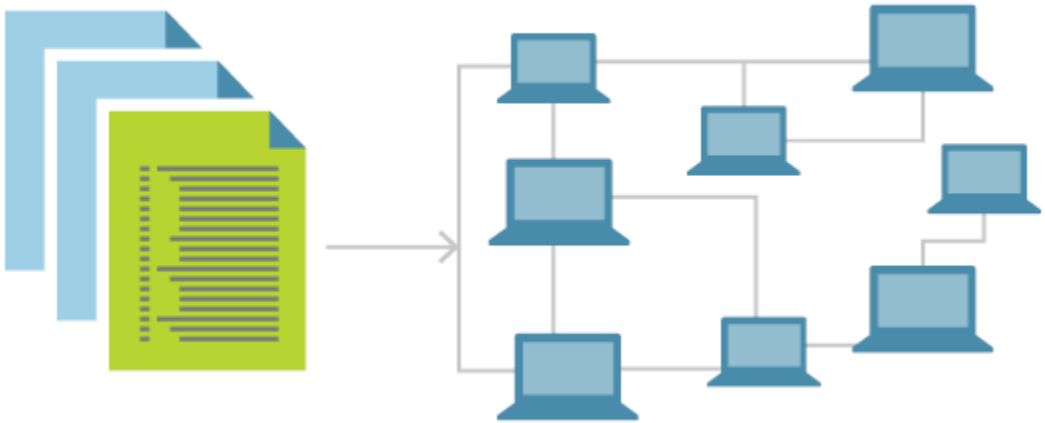
- Repository
- Make back up
- Keep History
- Collaborate

CI CD Pipeline

- Jenkins
- GITLab CI/CD

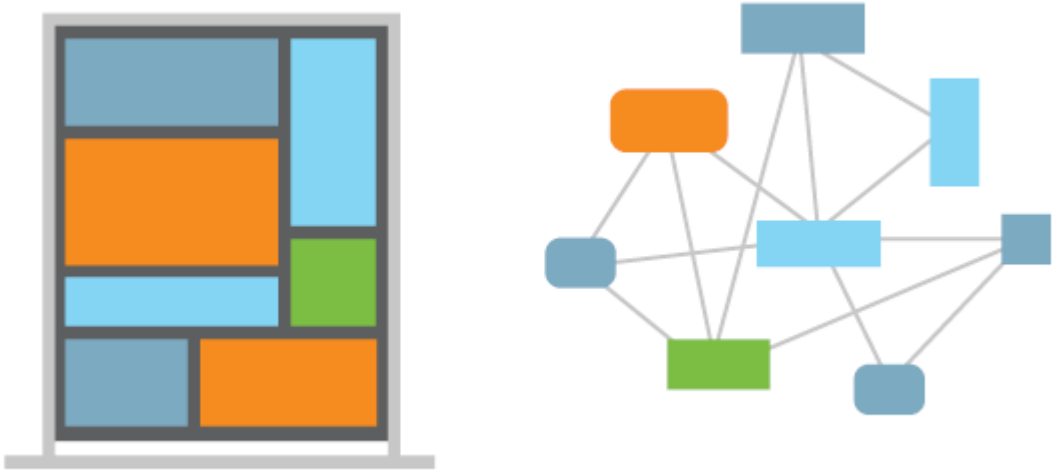
DevOps Practices - Infrastructure as Code

Manage infrastructure – networks, virtual machines, load balancers, connection topology through code.



DevOps Practices - Microservices

Building an application as a set of deployable small services.



Microservices - Docker

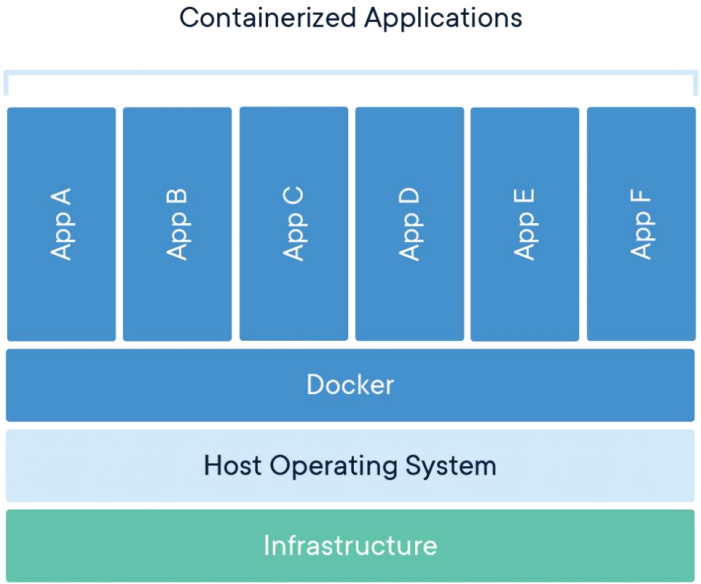
Docker provides the ability to package and run an application in a loosely isolated environment called a container.

Docker provides tooling and a platform to manage the lifecycle of your containers:

- Develop your application and its supporting components using containers.
- The container becomes the unit for distributing and testing your application.
- When you’re ready, deploy your application into your production environment, as a container or an orchestrated service. This works the same whether your production environment is a local data center, a cloud provider, or a hybrid of the two.

Docker Containers

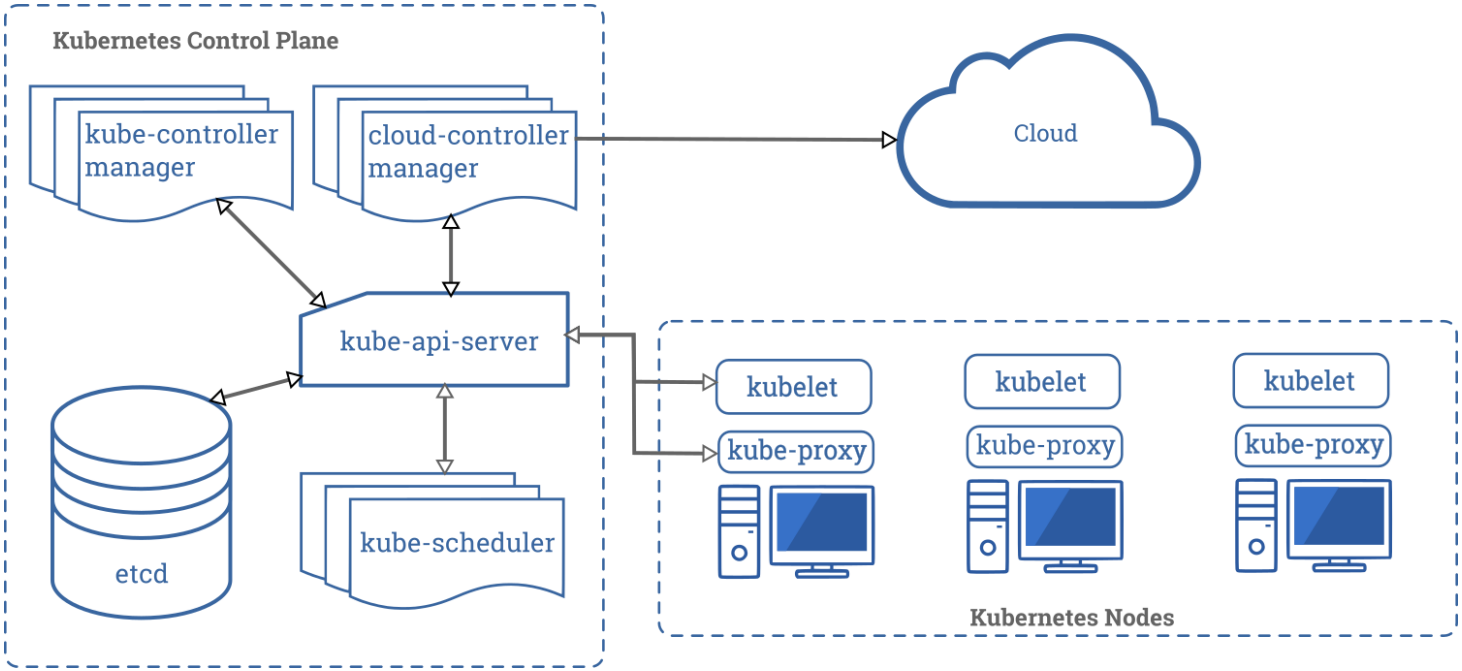
A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.



Kubernetes

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

Kubernetes Components



Monitoring

Grafana

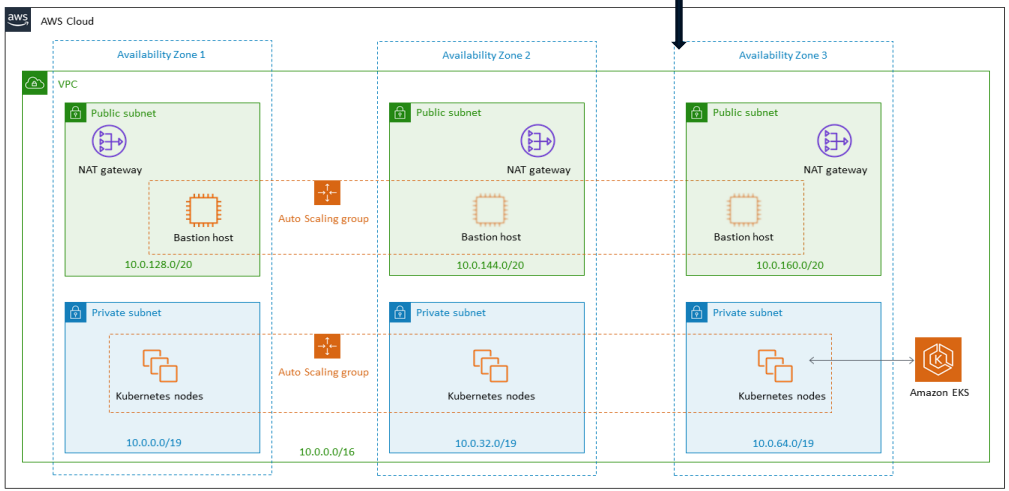
ELK stack

Splunk

DevOps

GitLab

Merge to Git Lab Branch → Build (docker) → Test → Deploy to environments



Reference

https://aws.amazon.com/devops/what-is-devops/

https://docs.microsoft.com/en-us/azure/devops/learn/what-is-devops

https://git-scm.com/

https://gitlab.com/explore

https://github.com/

https://www.jenkins.io/

https://docs.gitlab.com/ee/ci/

https://cucumber.io/

https://www.protractortest.org/#/

https://www.docker.com/

https://kubernetes.io/

https://grafana.com/

https://www.elastic.co/what-is/elk-stack

https://www.splunk.com/

TEKsystems
Own change

QUESTIONS

TEKsystems
Own change

THANK YOU