DevSecOps in the Real World: Best Practices for CI/CD and Microservices

Hendri Karisma

Hello!

my name is Hendri Karisma

- Technical Lead
- Working on platform system
- Before working for AI and DevEx

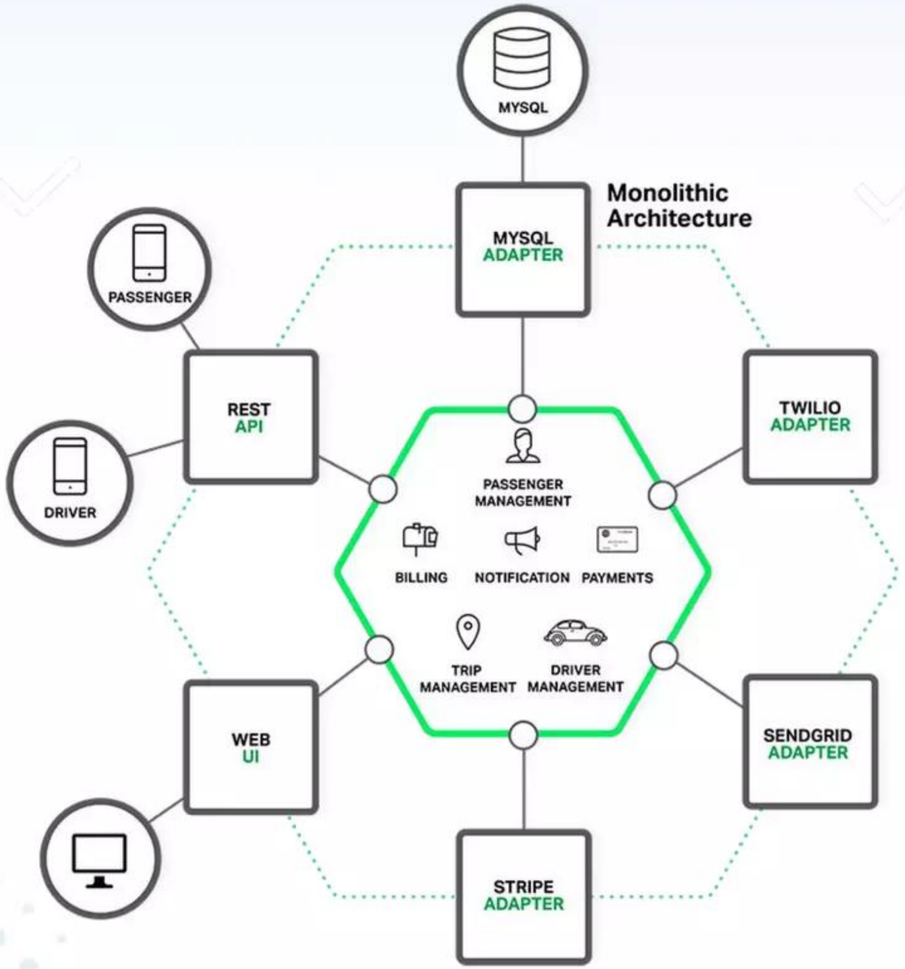
START TO LEAD

Micro-services

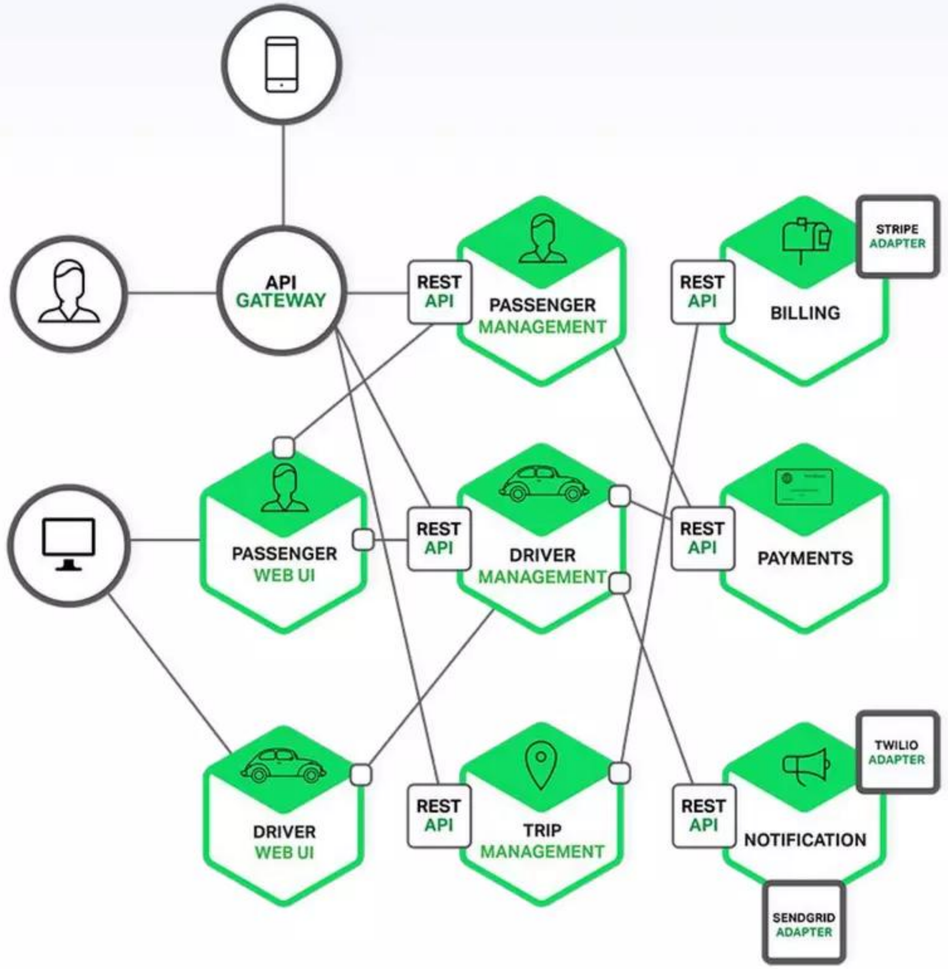
an architectural style that structures an application as a collection of services

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities
- Owned by a small team

Monolith vs Microservices?



Monolith Architecture



Microservices Architecture

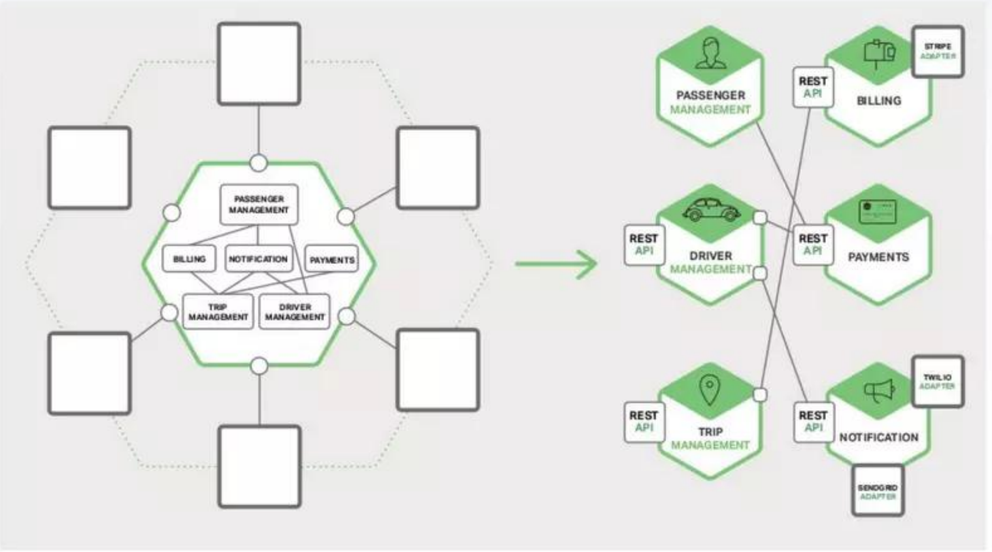
Why microservices?

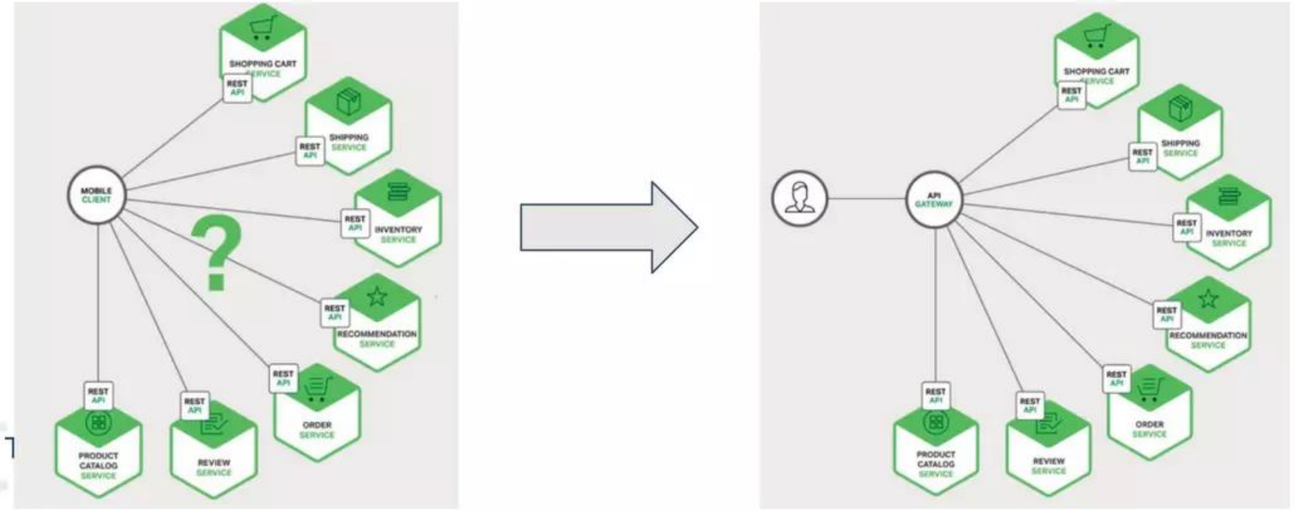
Wish our system could :

- Small so more modular, tackles the complexity issue. Lightweight
- Reusability
- Reliable
- Each service independent :
  - loosely coupled
  - Scalability

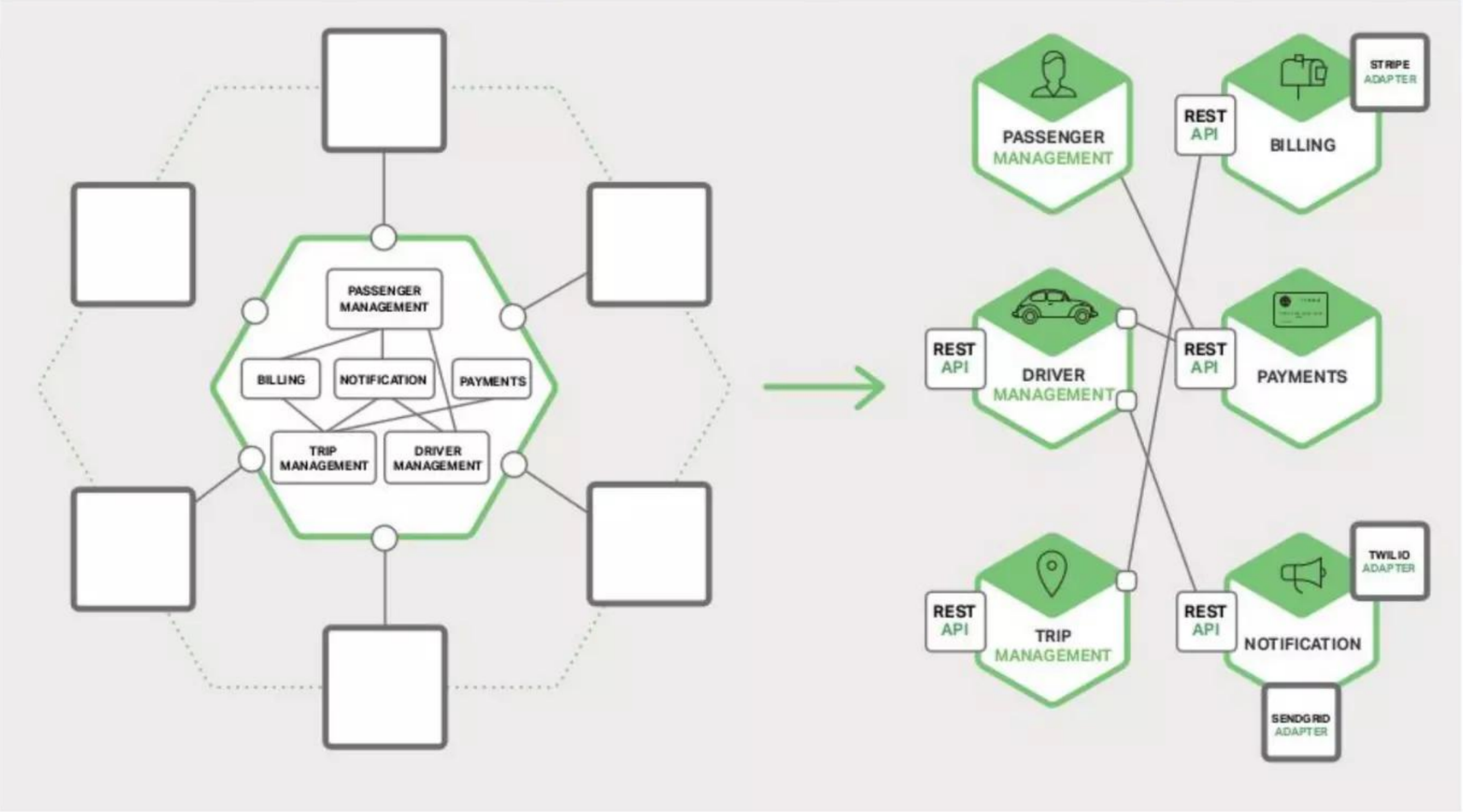
Challenge microservices?

- Communication : Latency and complexity
- Database: each services have their own database, transaction
- Testing: Integration testing
- Changes: could impact multiple services
- Deployment: machines x services, configuration, secrets management, monitoring

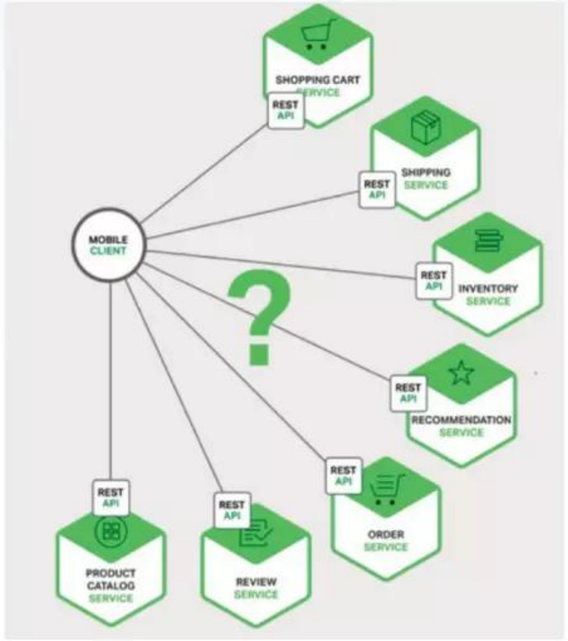


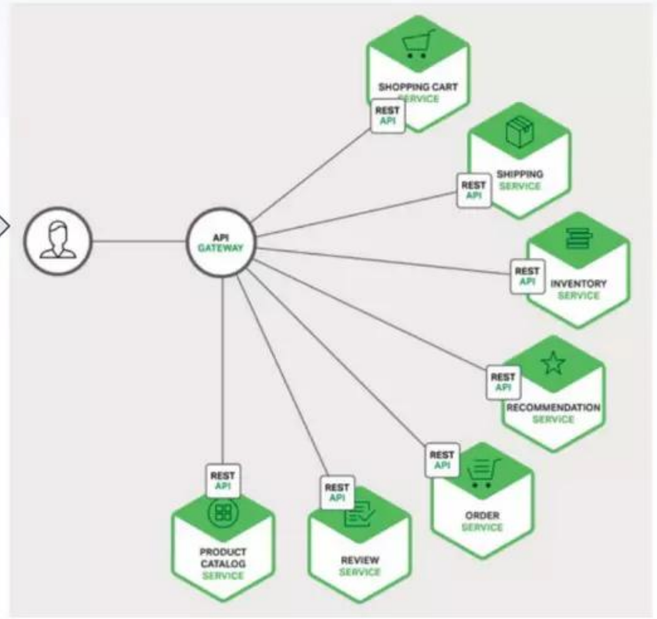


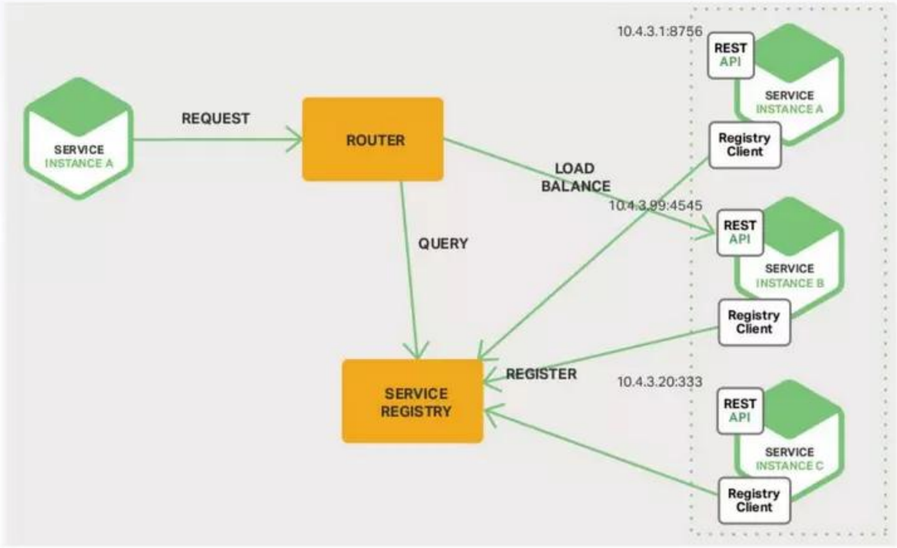
Communication

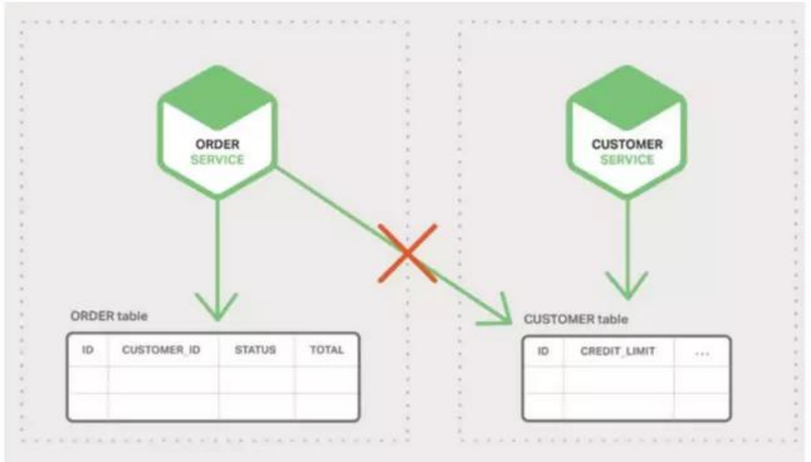


Communication









START TO LEAD

Orchestration

Entails actively controlling all elements and interactions like a conductor directs the musicians of an orchestra

One service controller handles all communications between microservices, and directs each service to perform the intended function.

Disadvantage :

- the controller needs to directly communicate with each service and wait for each service's response
- impacted by downstream network and service availability (latency)
- More tight coupling then we could say it's a distributed monolithic.

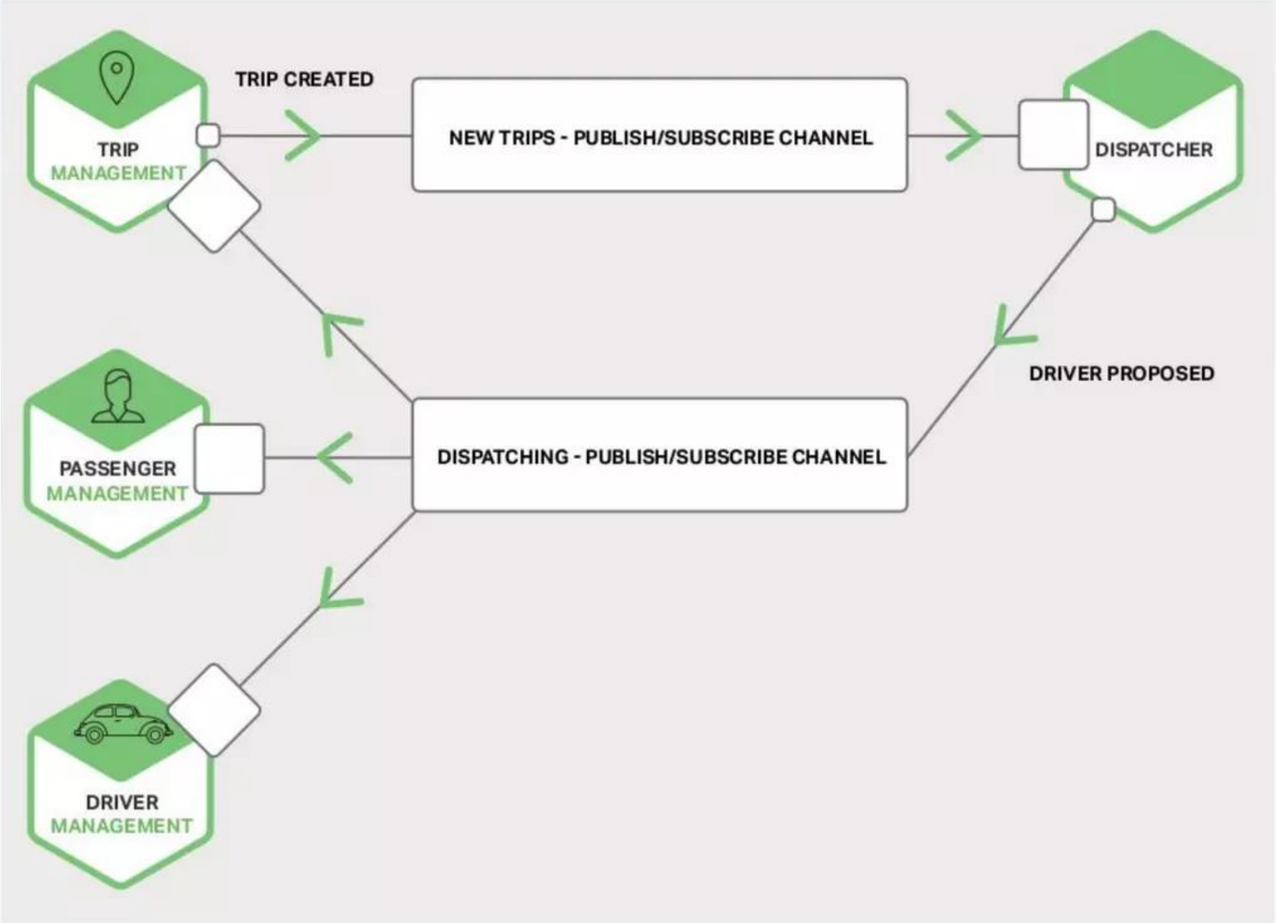
Choreography

Asynchronous process: Each service works independently and consumes the data that relates to it to perform its task.

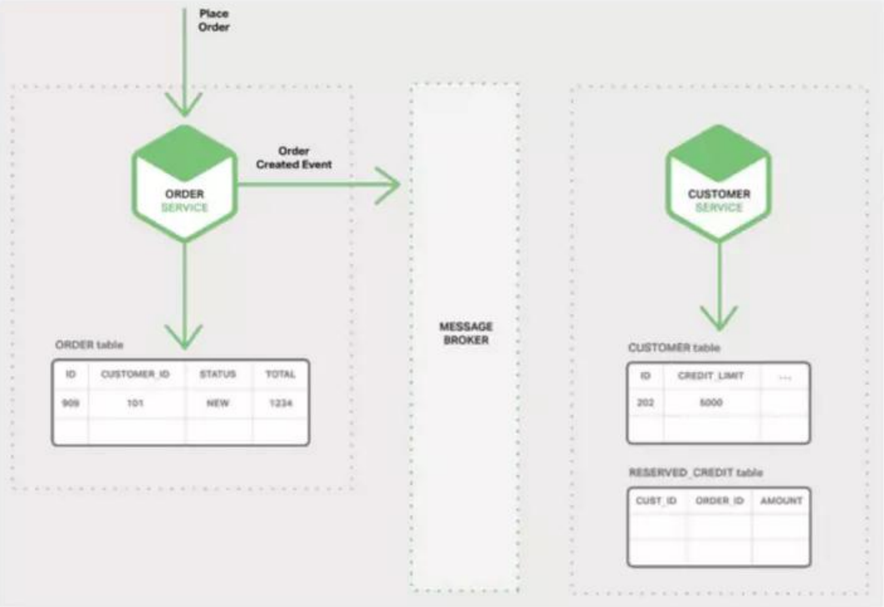
- Event Driven Architecture
- Decouples client from the service
- Message Buffering
- Flexible style

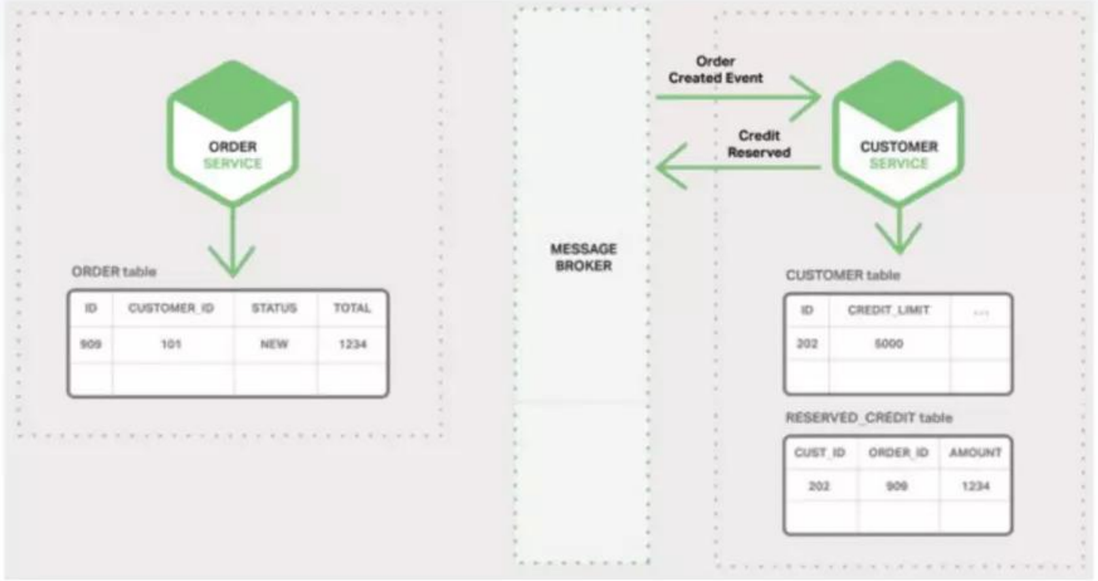
Tech: RabbitMQ, Apache Kafka, ActiveMQ, etc

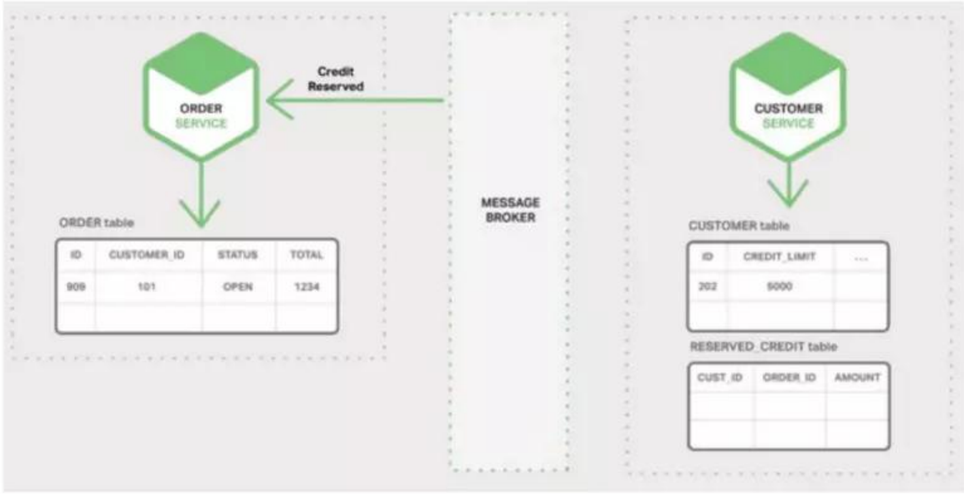
Event Driven #1



Event Driven #2 (communications)

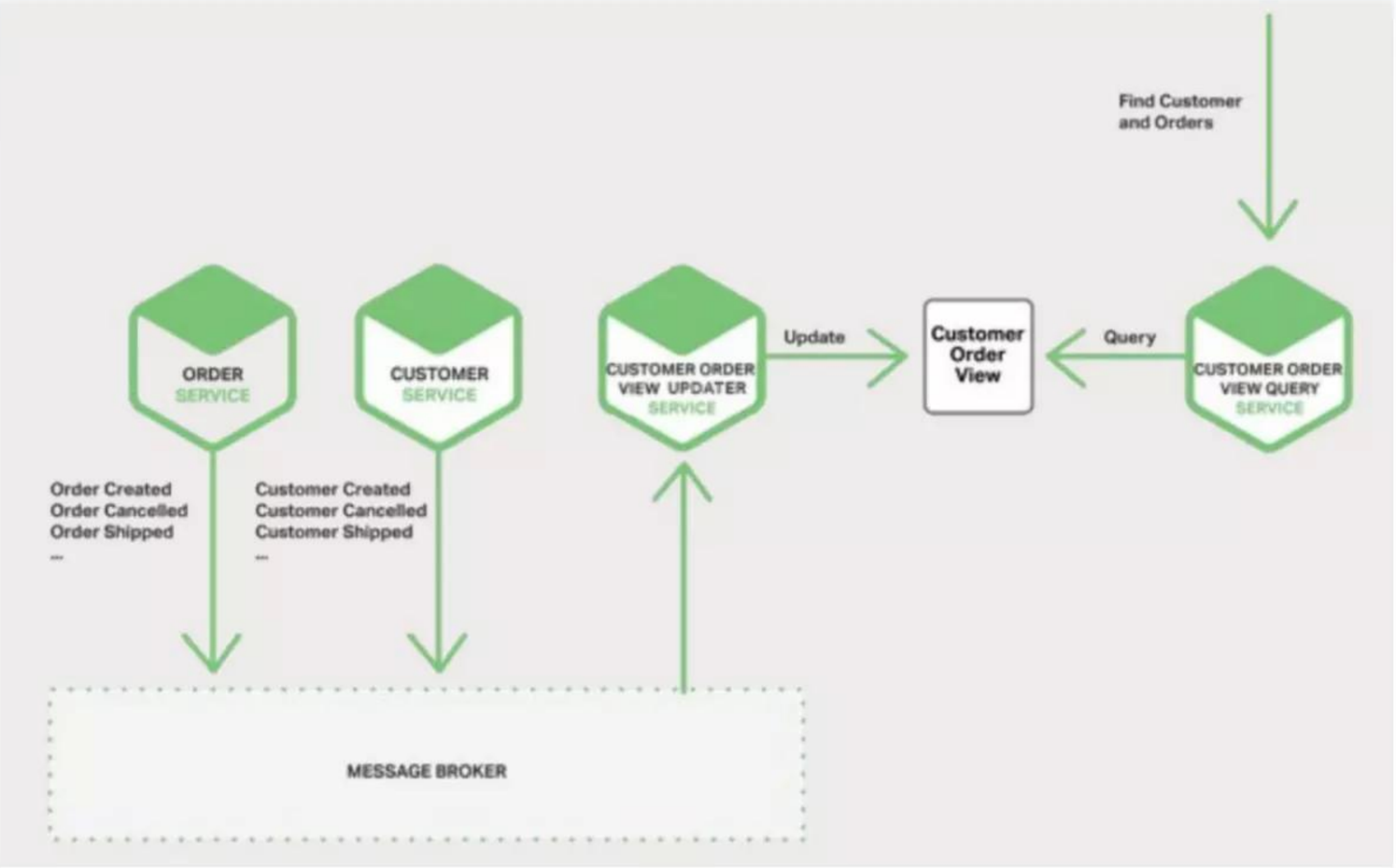




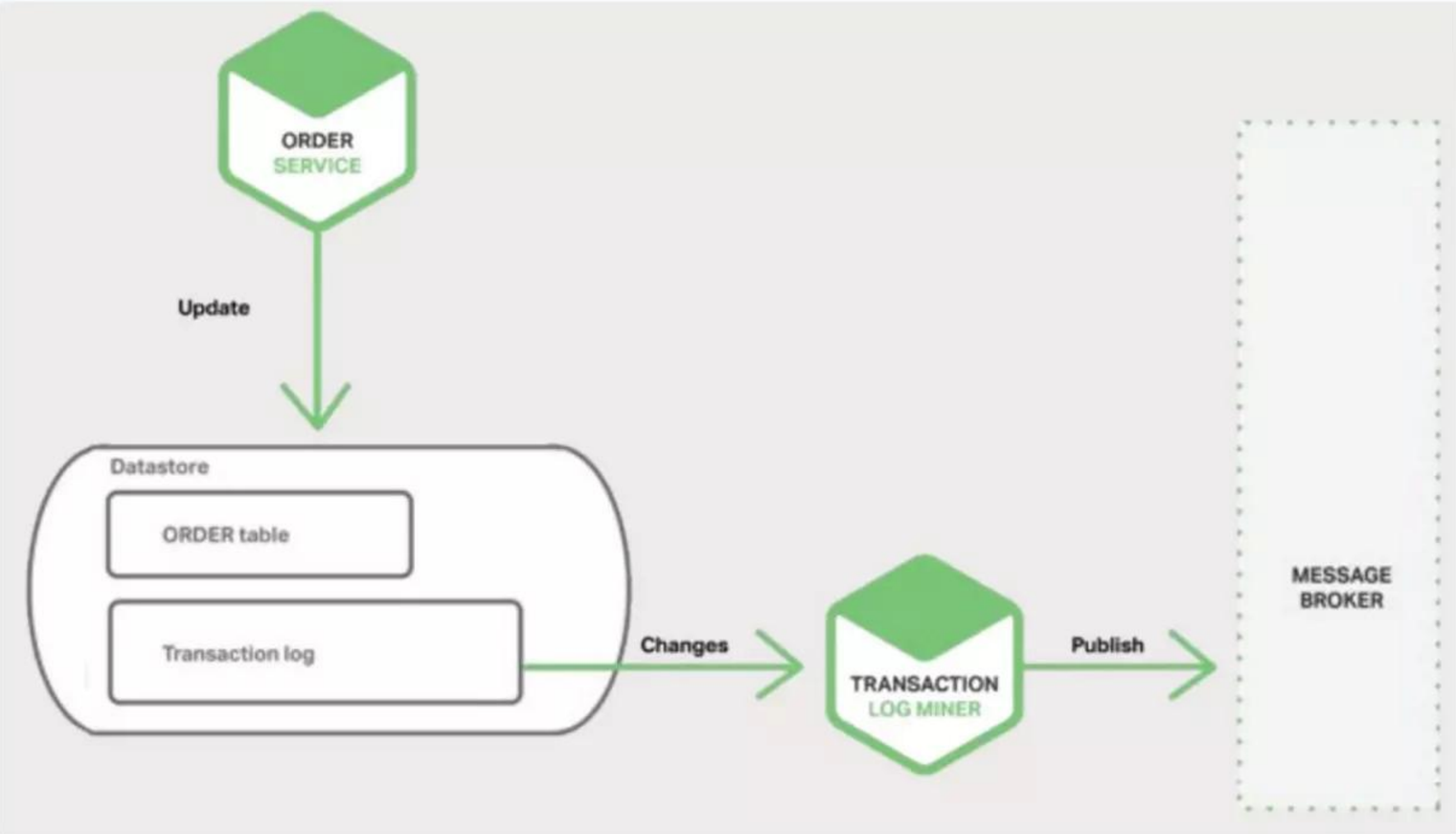


START TO LEAD

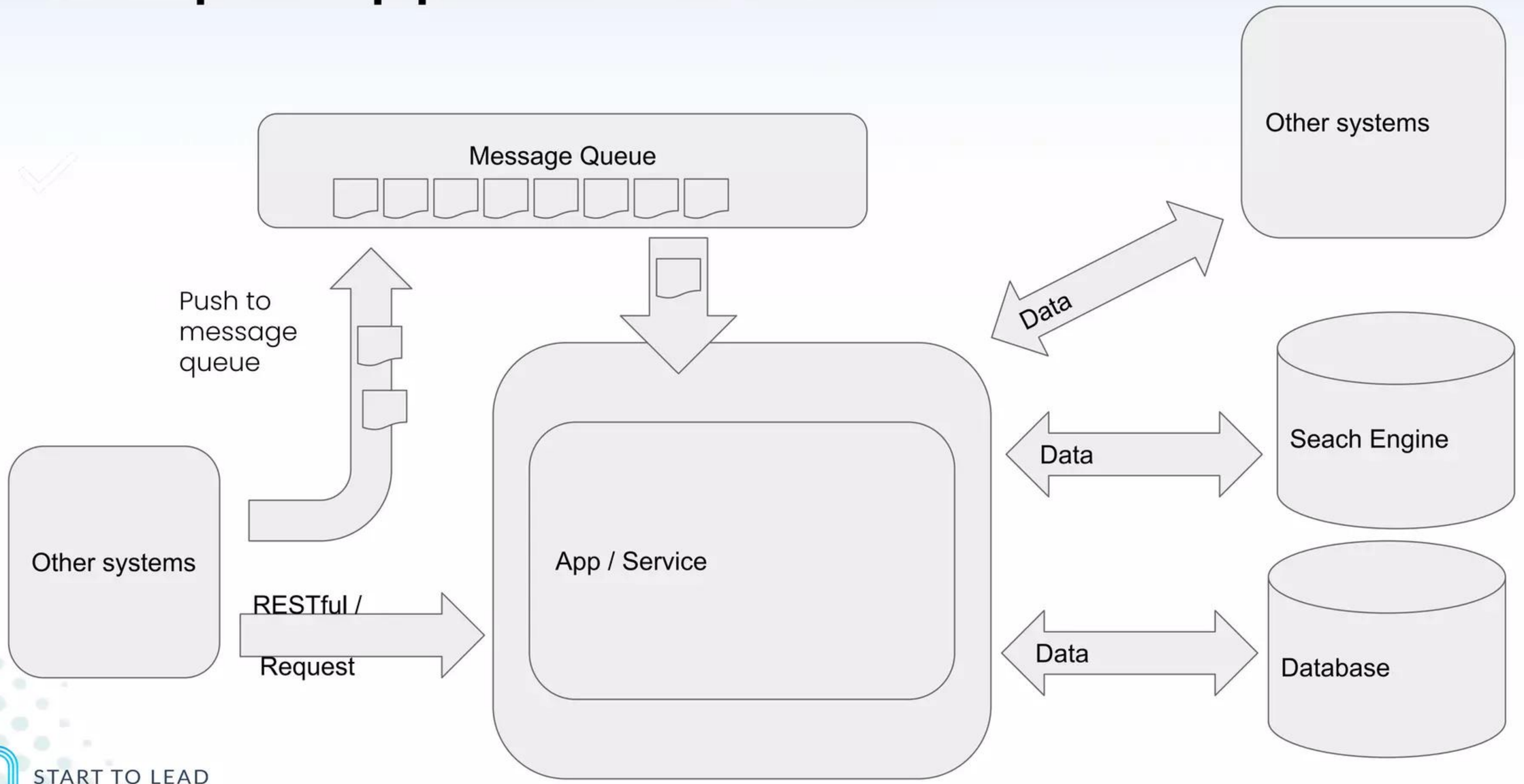
Event Driven (Data Aggregation)



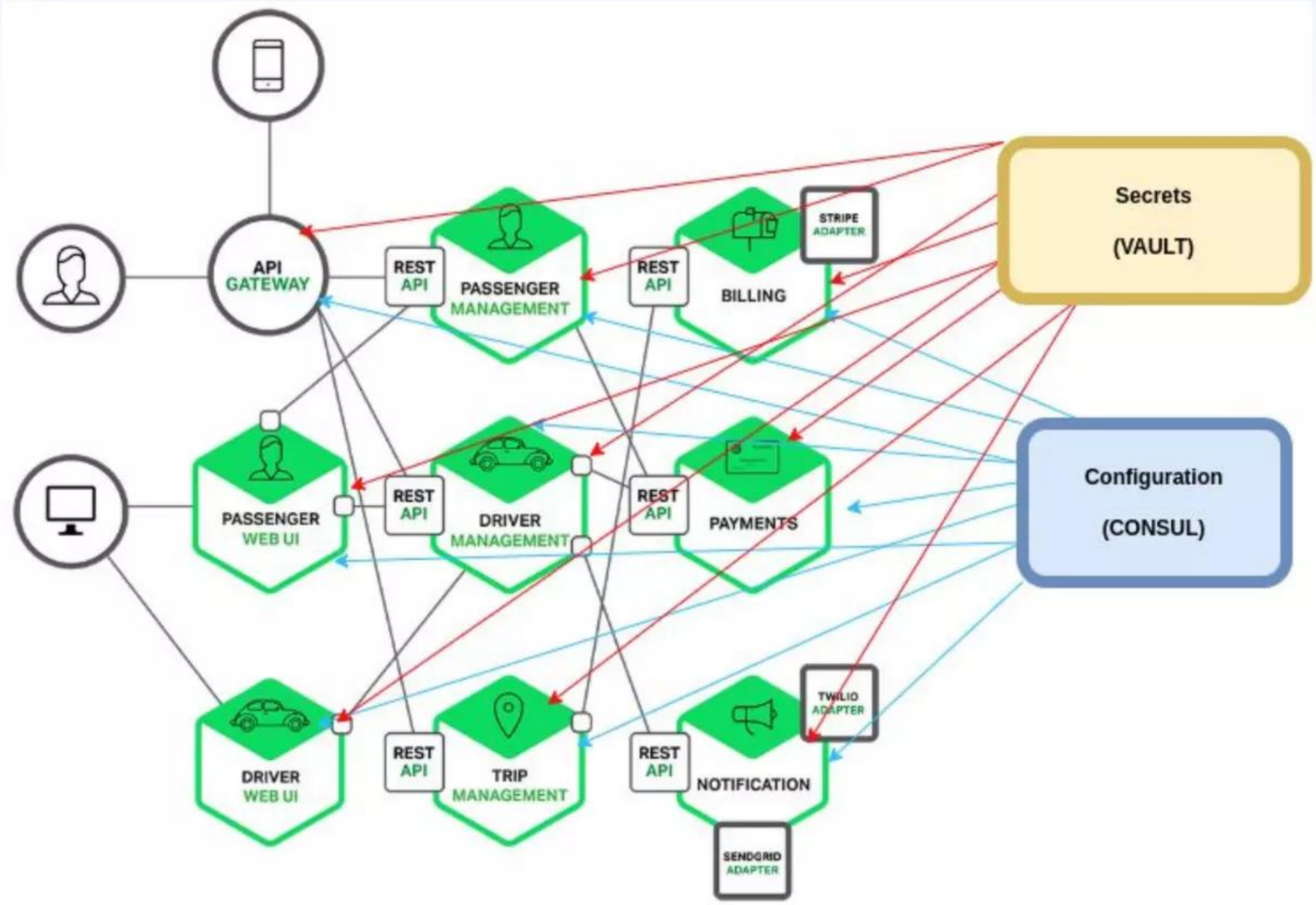
Event Driven (Data Logging)



Sample App Architecture

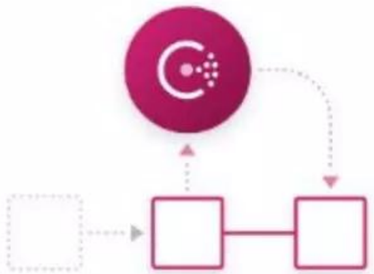


Configuration

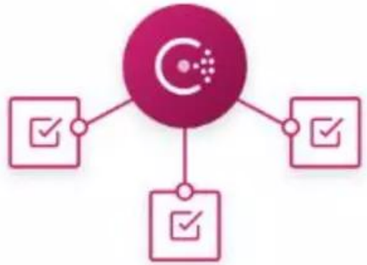


Configuration #2

HashiCorp Consul



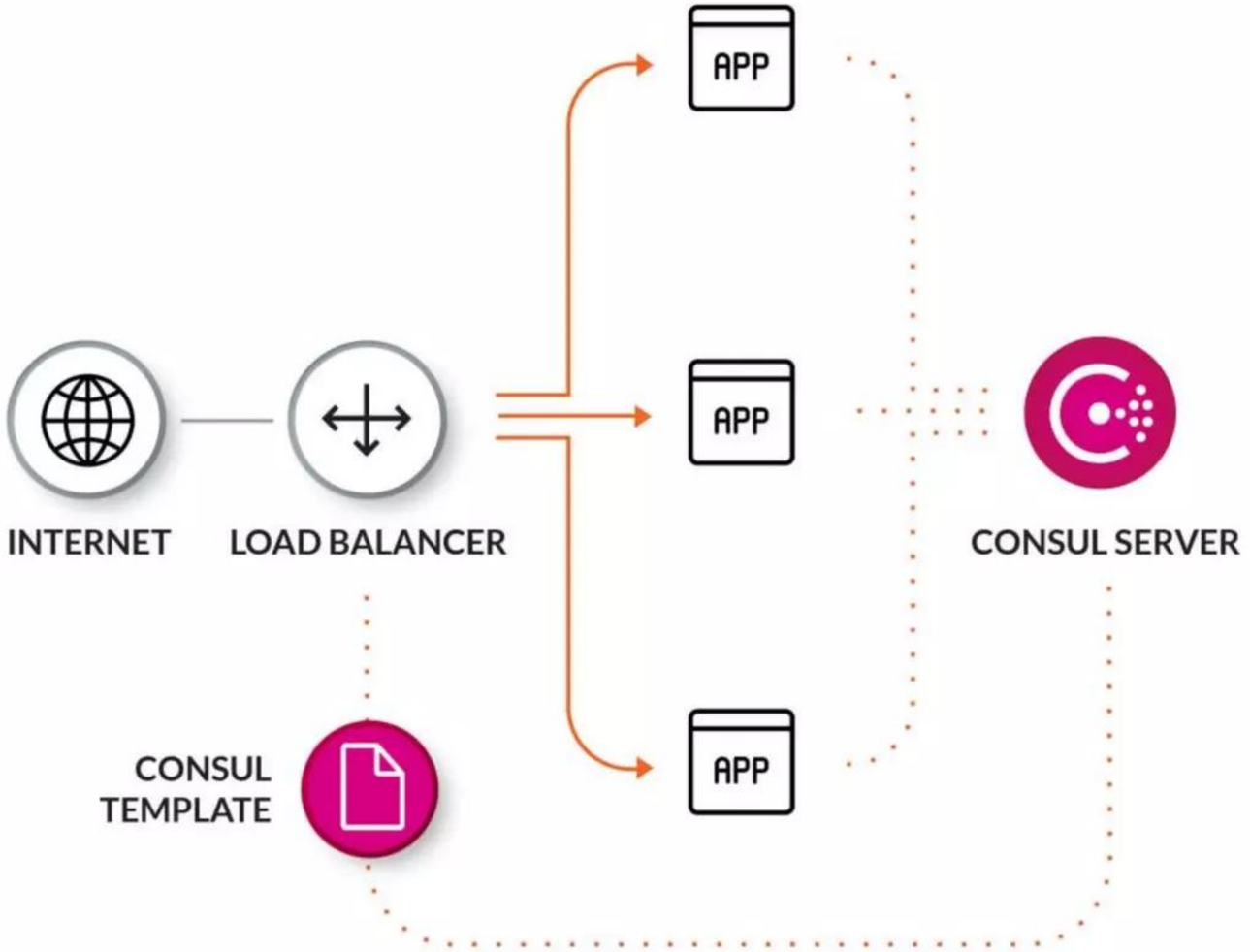
Service Discovery



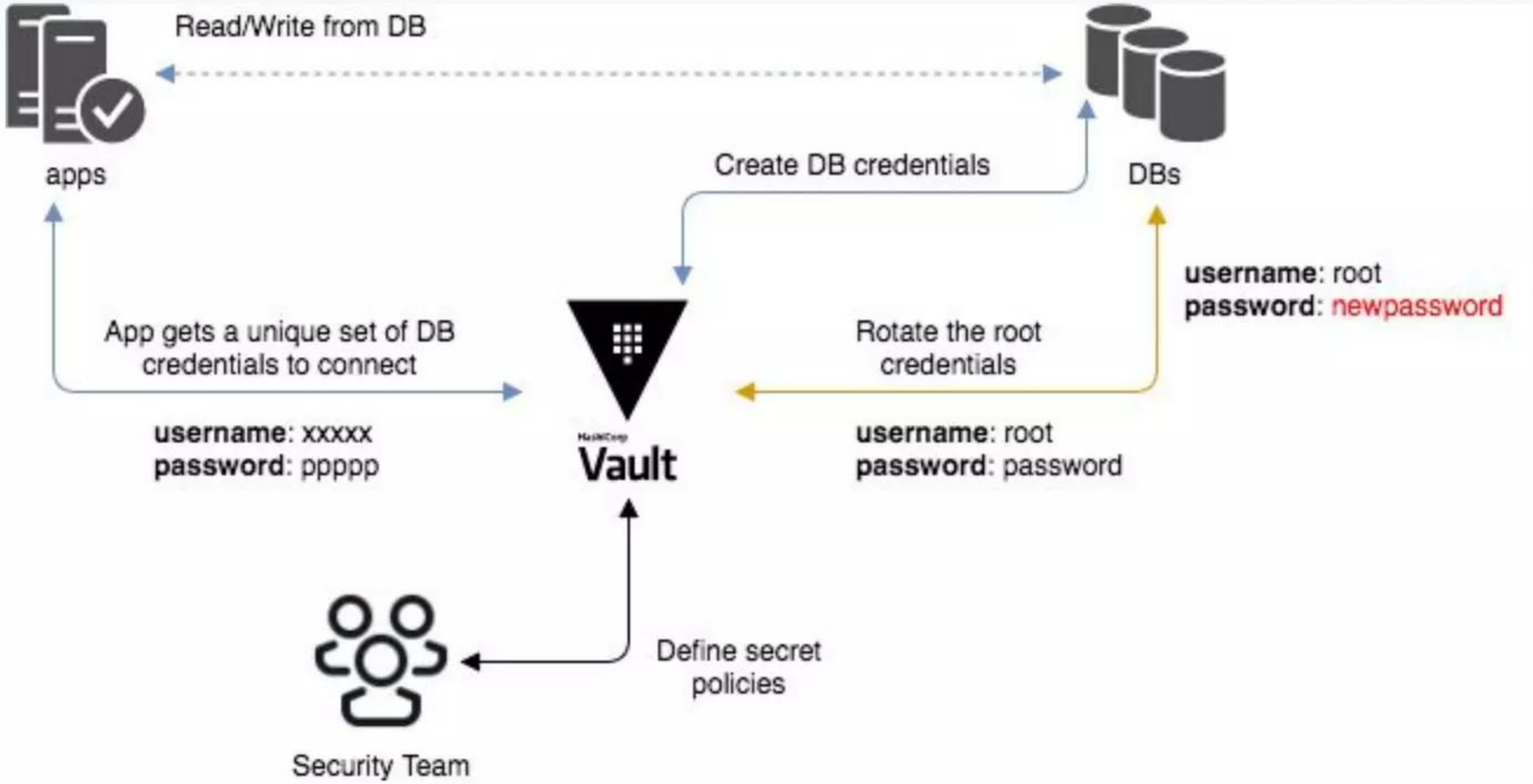
Service Configuration

Service Segmentation

Configuration #3



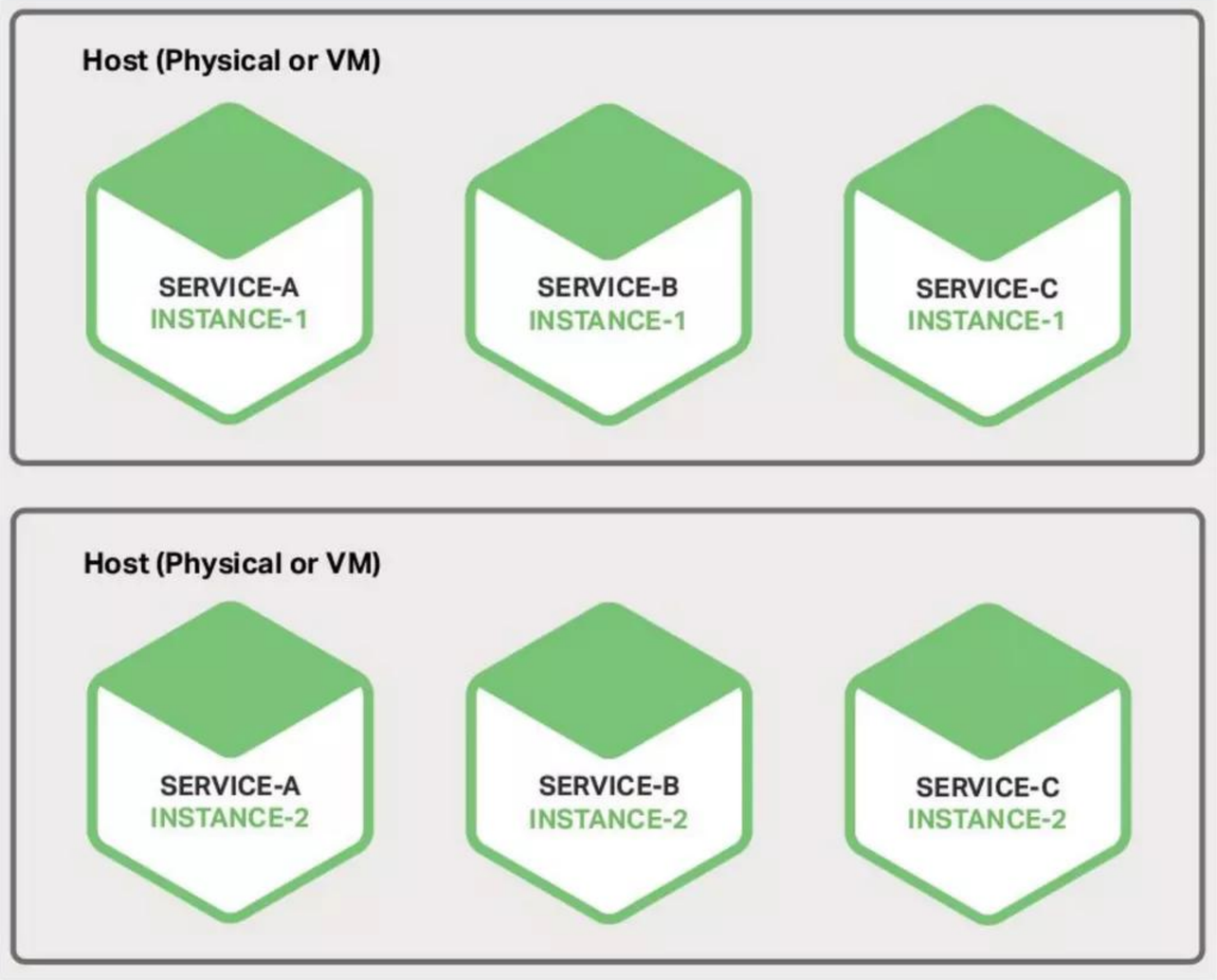
Secrets


Read/Write from DB
apps
DBs
Create DB credentials
username: root
password: newpassword
App gets a unique set of DB credentials to connect
Rotate the root credentials
username: xxxxx
password: ppppp
Vault
username: root
password: password
Define secret policies
Security Team

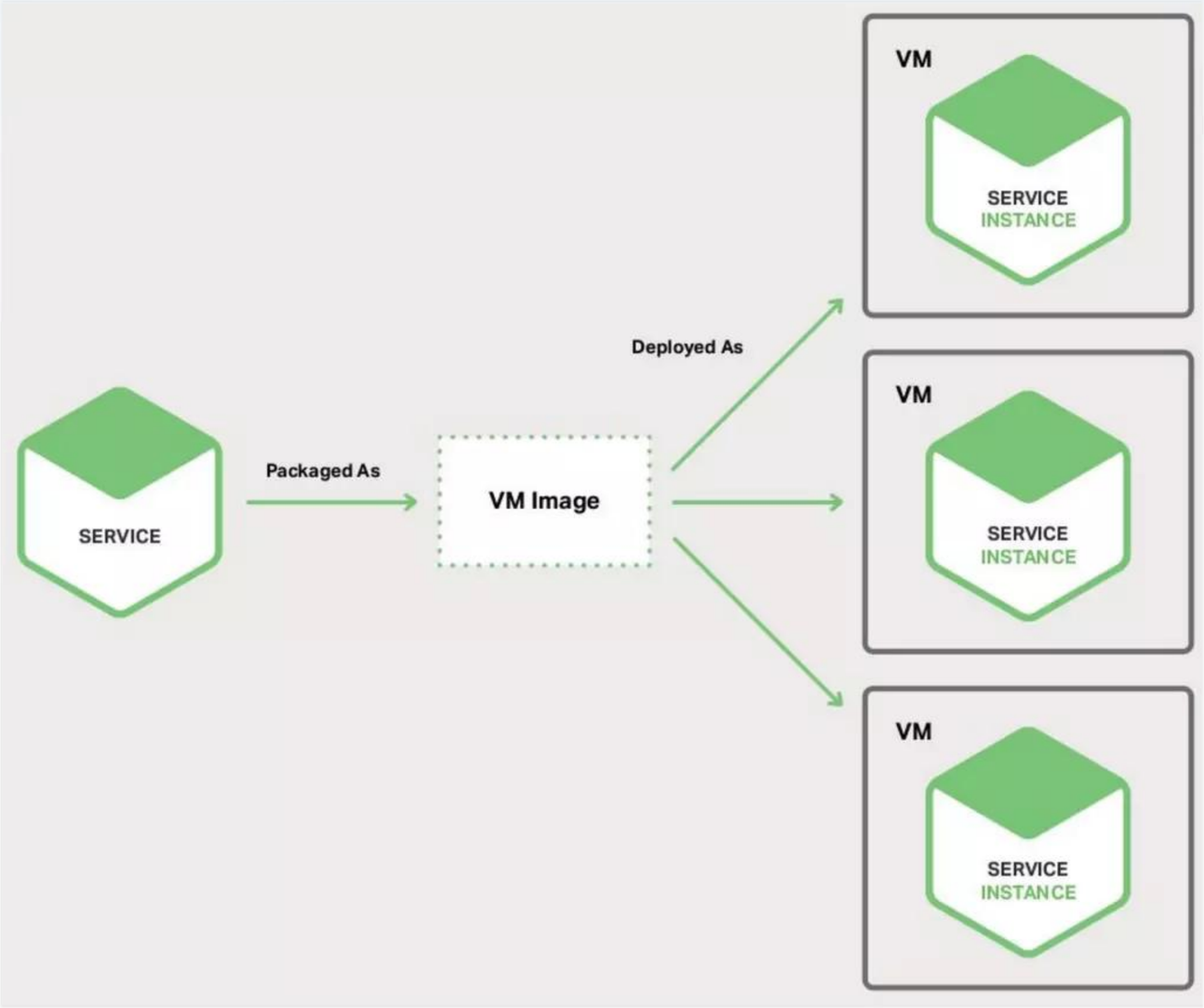
Deployment

- Classic : on top Bare metal server or VM
- Using Virtual machine for each instance (app node)
- Using Container and Container Orchrestation

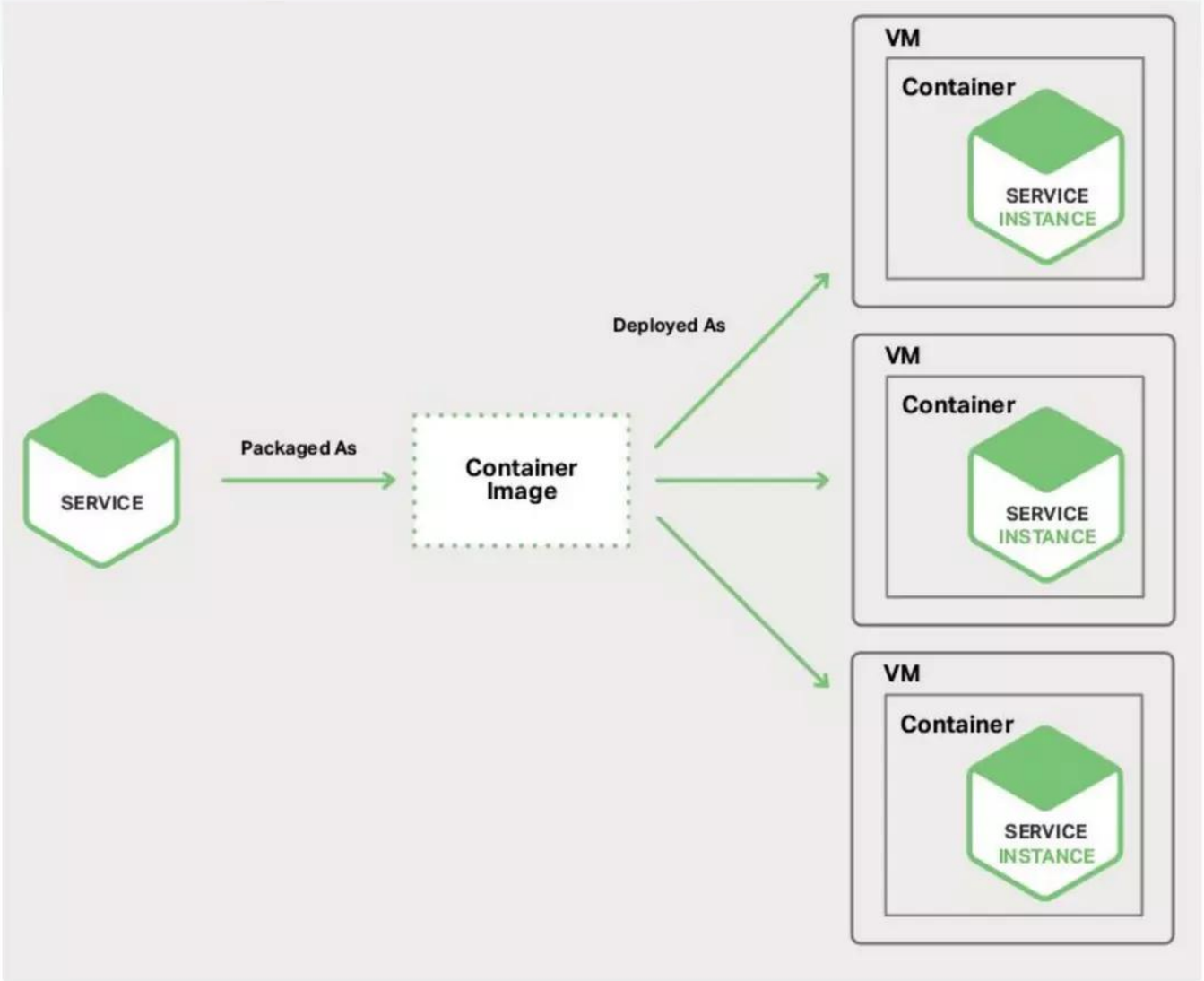
Deployment #1



Deployment #2

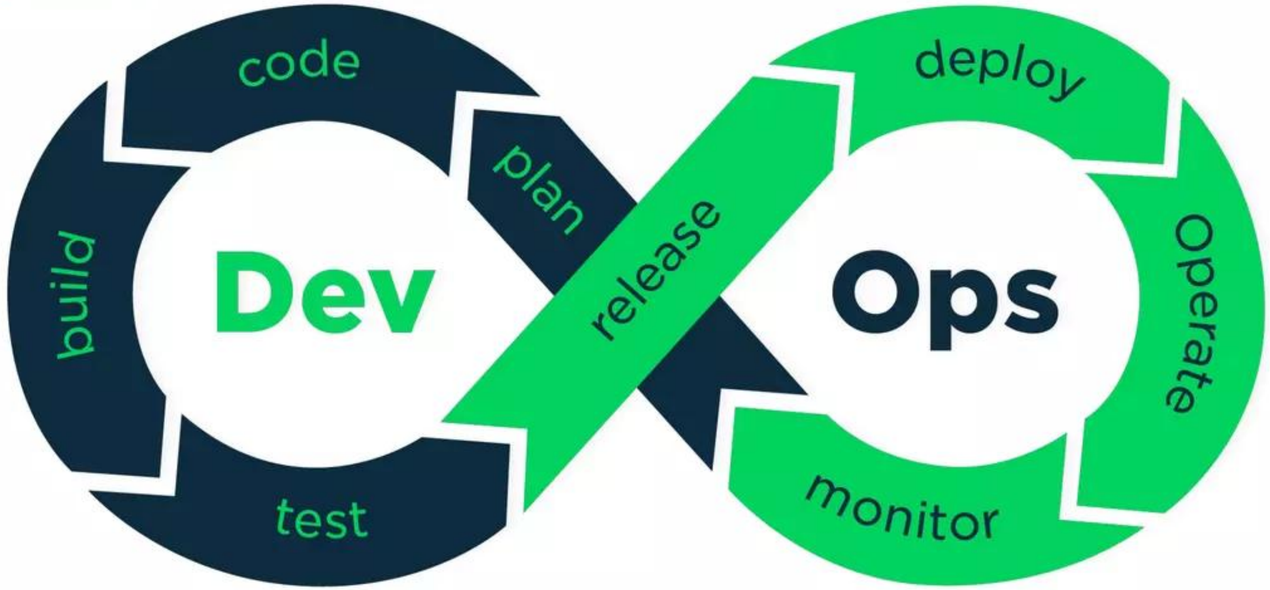


Deployment #3



DevOps

DevOps is the **combination of cultural philosophies, practices, and tools** that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.

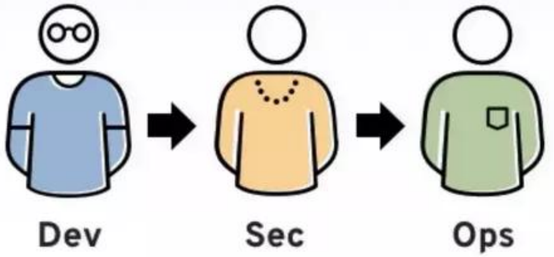


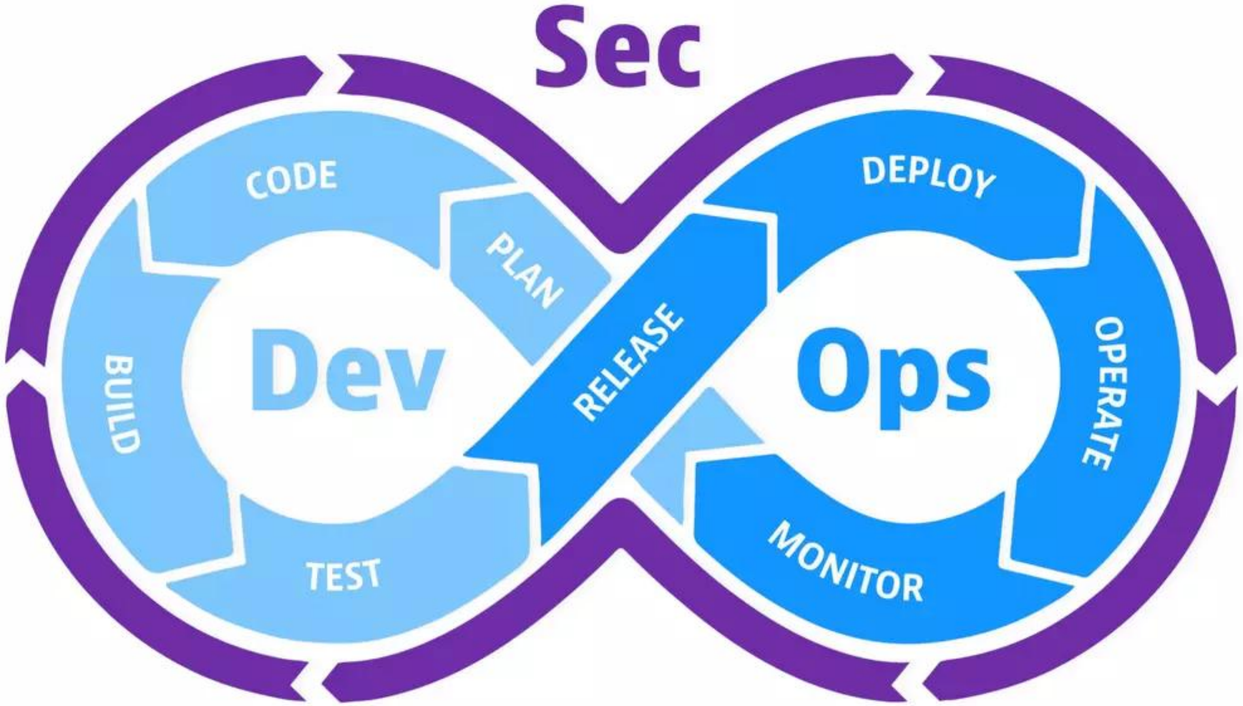
DevSecOps

Now, in the collaborative framework of DevOps, security is a shared responsibility integrated from end to end.

The term "DevSecOps" to emphasize the need to build a security foundation into DevOps initiatives.

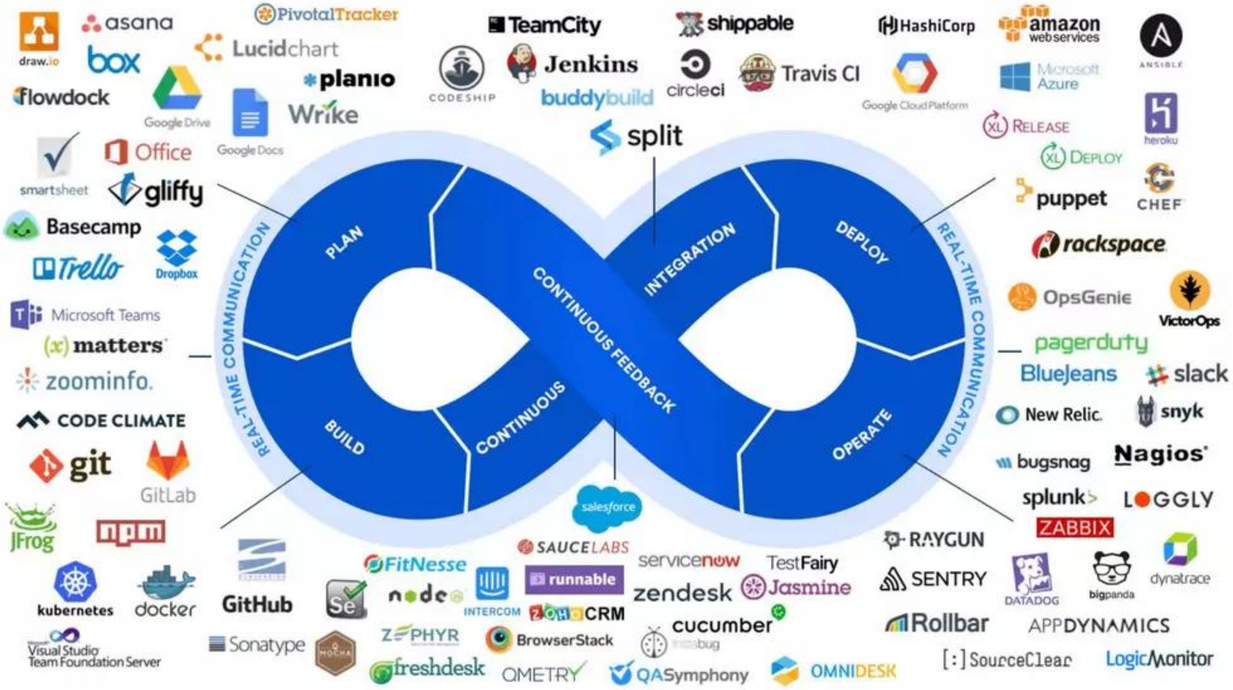
It also means automating some security gates to keep the DevOps workflow from slowing down.





Tech to Support DevSecOps

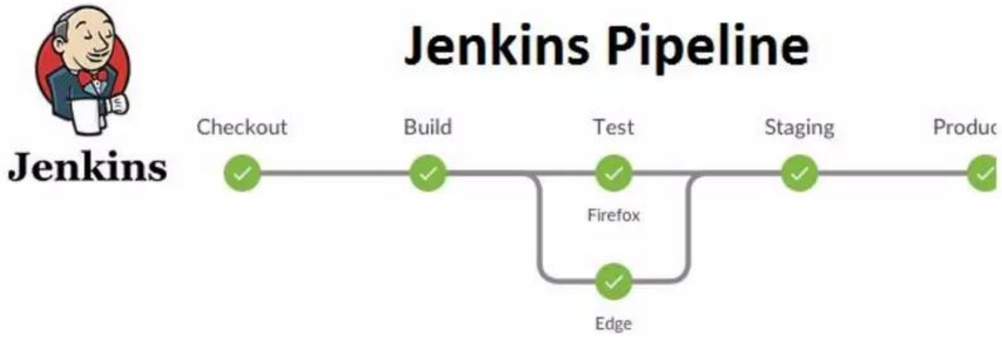
- **Automation Server**: Jenkins, Bamboo, Github Action, travis, circleci, ansible, etc
- **Infrastructure as code** / CLI: AWS SDK, GCP SDK, terraform, chef, etc
- **Registry/Repository** : docker hub, helm, GCR, etc
- **Security scanner** : Sonarqube, trivy, etc
- **Container Orchestration**: Kubernetes, Docker swarm
- **Configuration & Secret : vault & consul**
- Code Repository: bitbucket, github, gitlab, etc

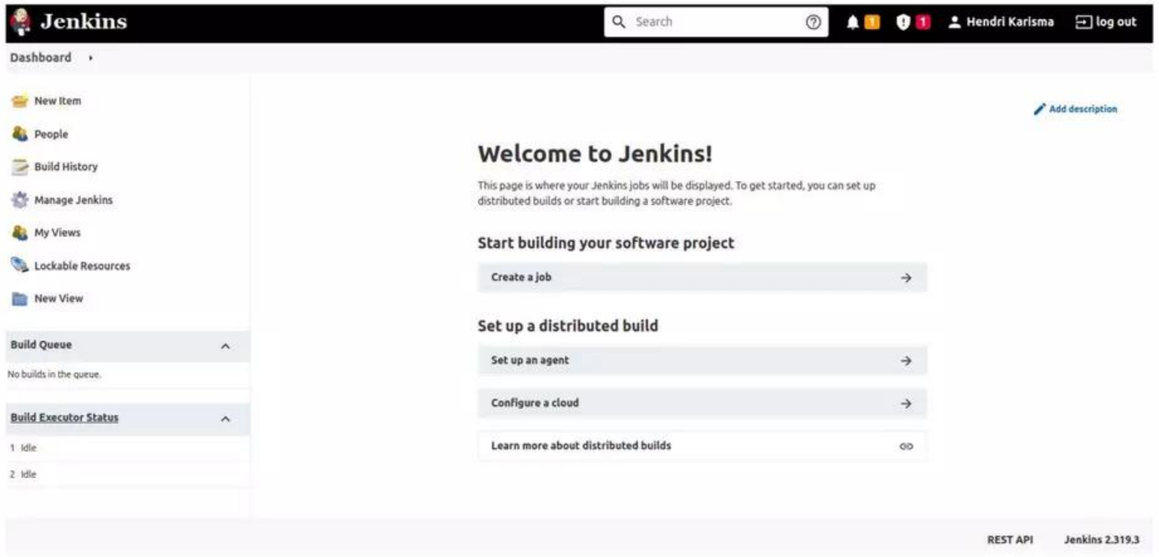


START TO LEAD

Jenkins

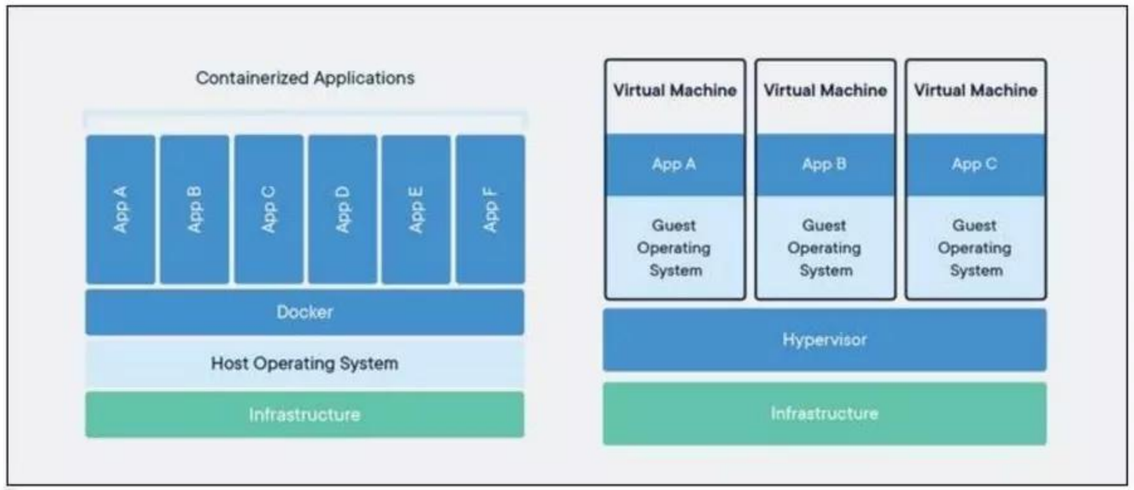
An open source extensible automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery.

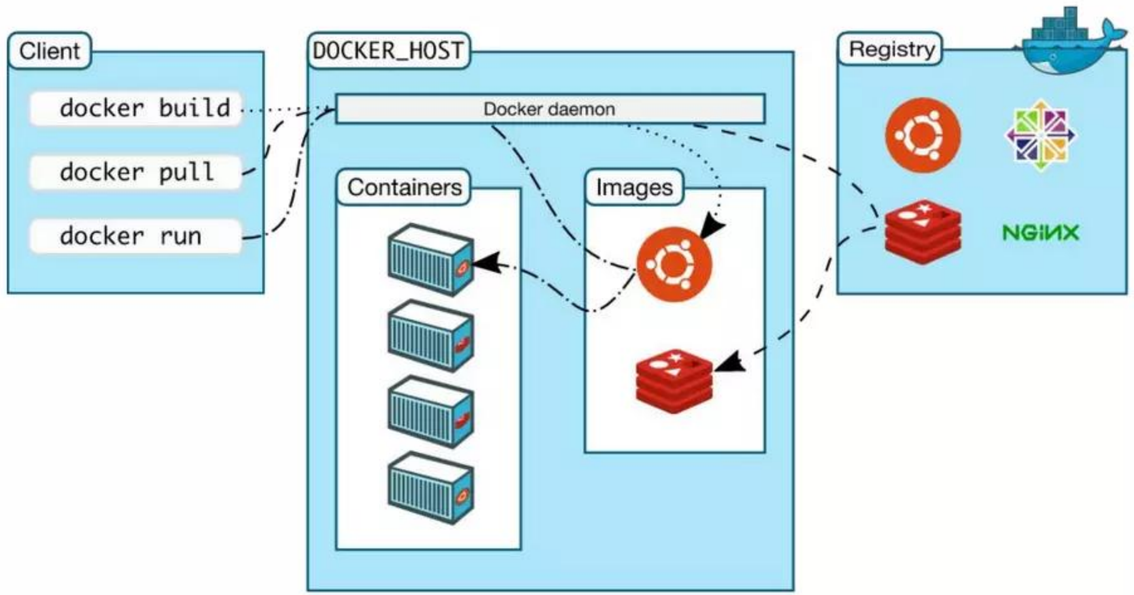




Docker

an open source containerization platform. It **enables developers to package applications into containers**—standardized executable components combining application source code with the operating system (OS) libraries and dependencies required to run that code in any environment.



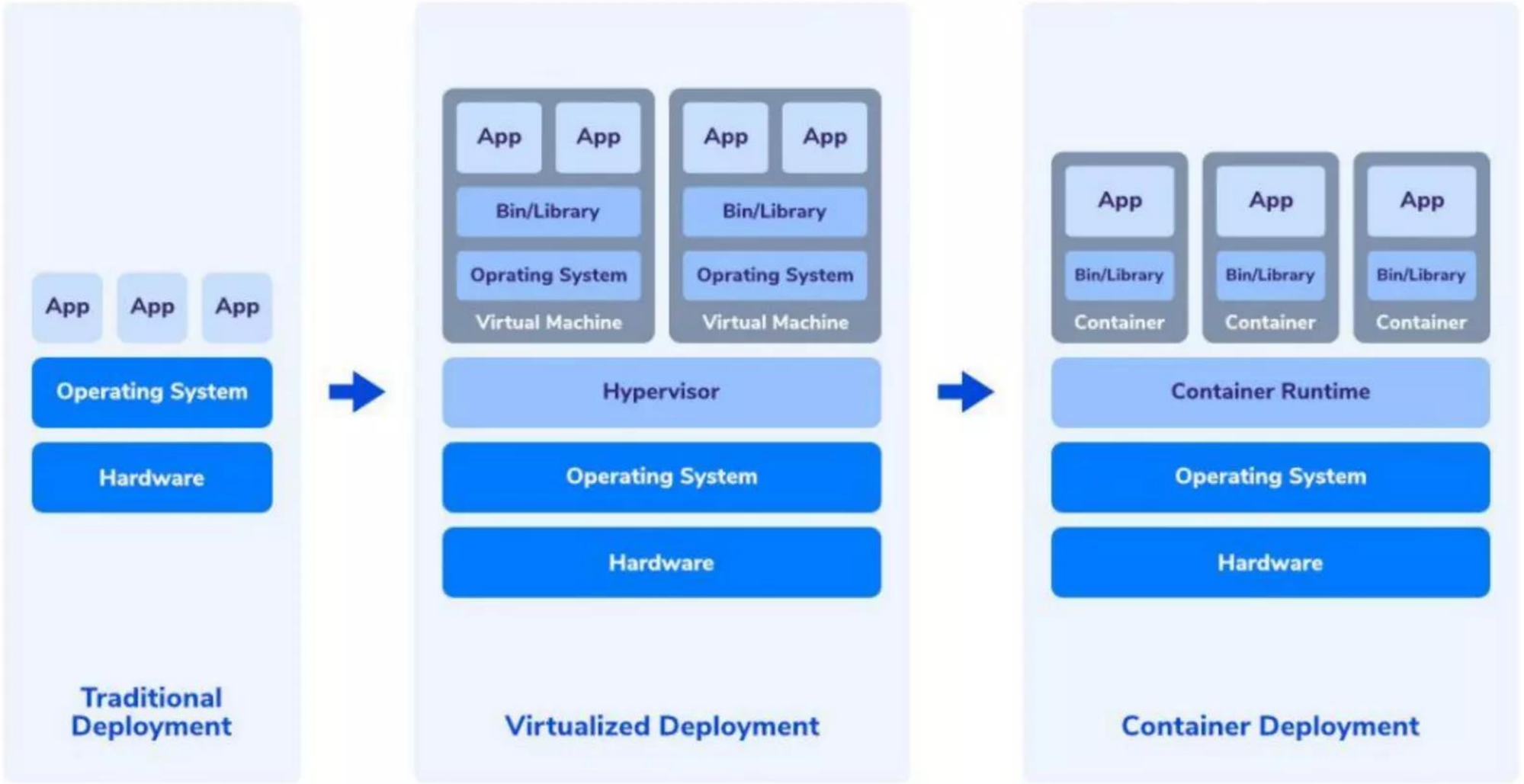


Kubernetes

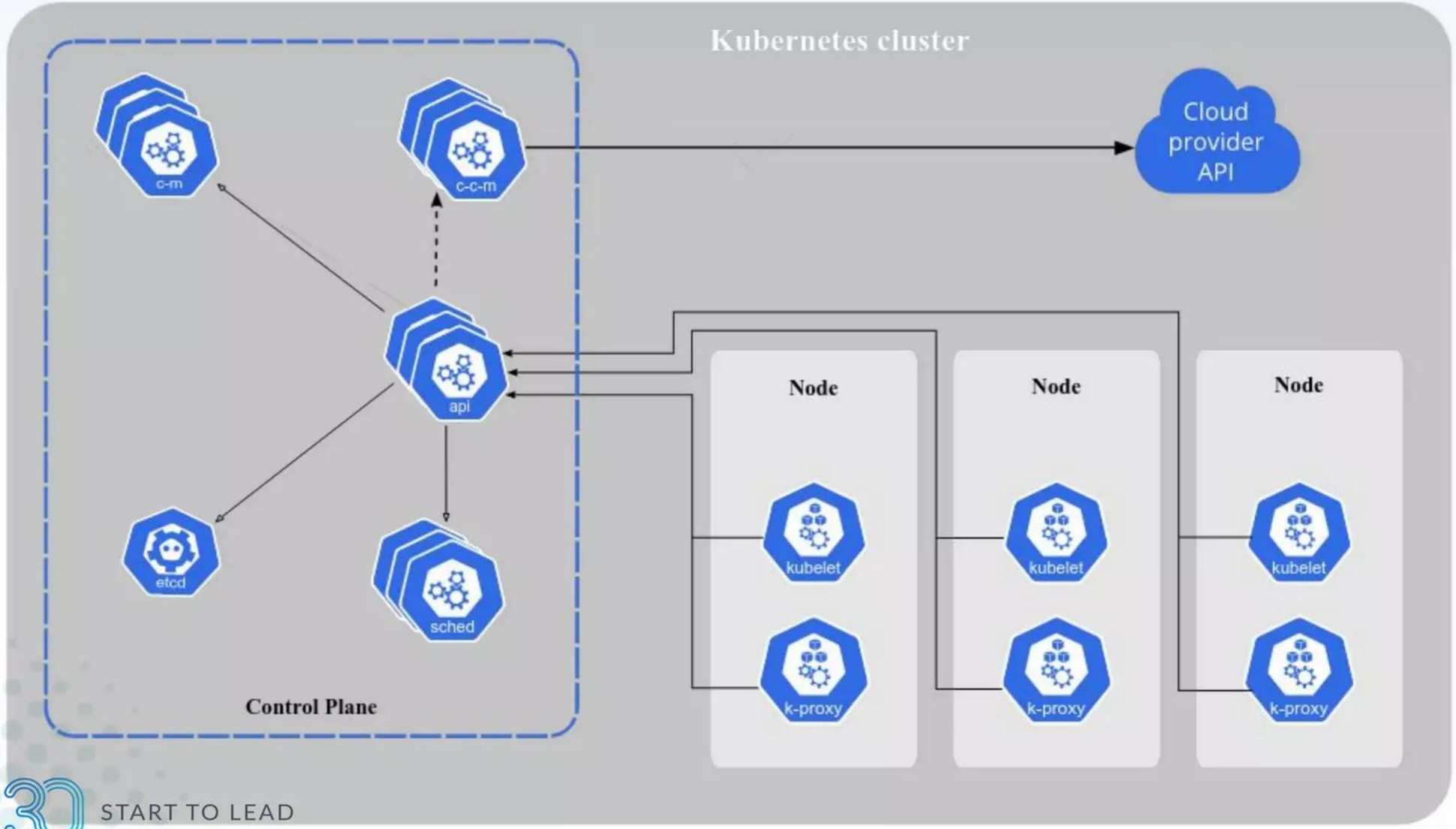
Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. Google originally designed Kubernetes, but the Cloud Native Computing Foundation now maintains the project.

also known as K8s. K8s could automate the deployment, scaling, and management of containerized applications.

Kubernetes


App
App
App
Operating System
Hardware
Traditional Deployment
App
App
Bin/Library
Oprating System
Virtual Machine
App
App
Bin/Library
Oprating System
Virtual Machine
Hypervisor
Operating System
Hardware
Virtualized Deployment
App
App
App
Bin/Library
Bin/Library
Bin/Library
Container
Container
Container
Container Runtime
Operating System
Hardware
Container Deployment

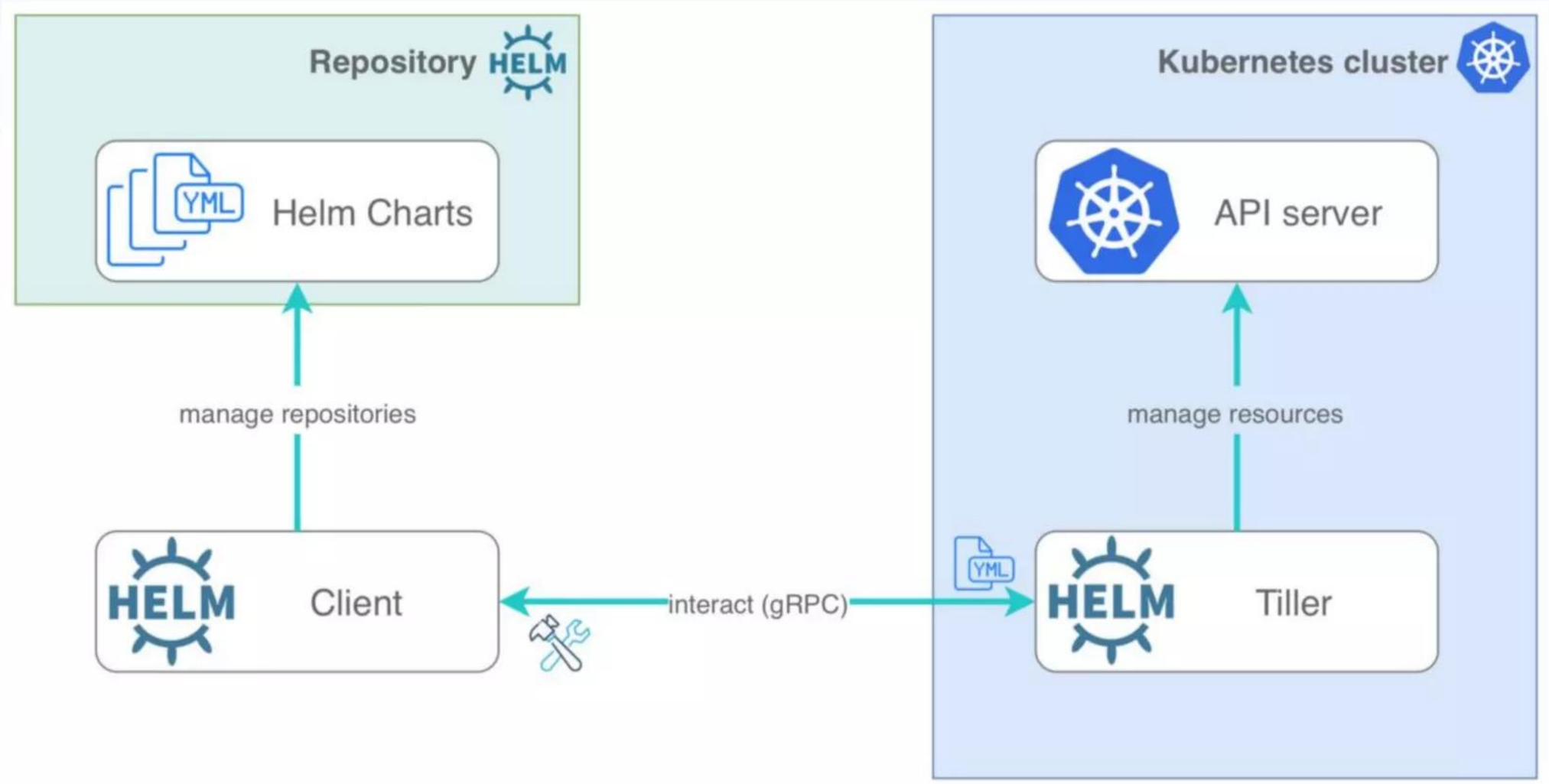
Kubernetes


Kubernetes cluster
c-m
c-c-m
Cloud provider API
api
etcd
sched
Control Plane
Node
kubelet
k-proxy
Node
kubelet
k-proxy
Node
kubelet
k-proxy
API server
Cloud controller manager (optional)
Controller manager
etcd (persistence store)
kubelet
kube-proxy
Scheduler
Control plane
Node

Helm and Helm Chart

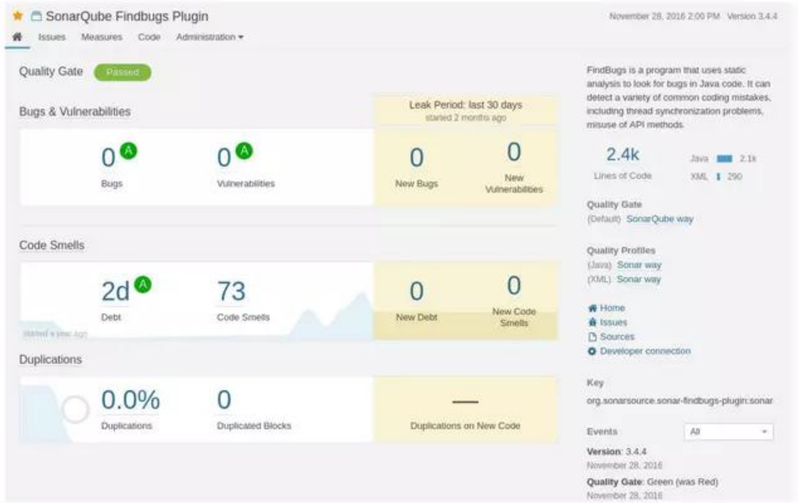
- helps you manage Kubernetes applications — Helm Charts help you define, install, and upgrade even the most complex Kubernetes application.
- Helm Charts are simply **Kubernetes YAML manifests combined into a single package that can be advertised to your Kubernetes clusters**. Once packaged, installing a Helm Chart into your cluster is as easy as running a single helm install, which really simplifies the deployment of containerized applications.

Helm


Repository
HELM
YML
Helm Charts
manage repositories
HELM
Client
interact (gRPC)
Kubernetes cluster
API server
manage resources
YML
HELM
Tiller

Sonarqube

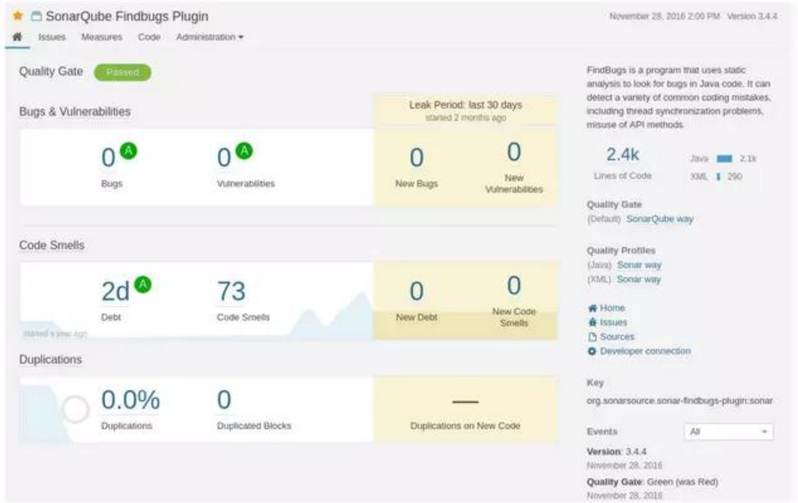
SonarQube is a **Code Quality Assurance tool that collects and analyzes source code**, and provides reports for the code quality of your project. It combines static and dynamic analysis tools and enables quality to be measured continually over time.



START TO LEAD

Sonarqube

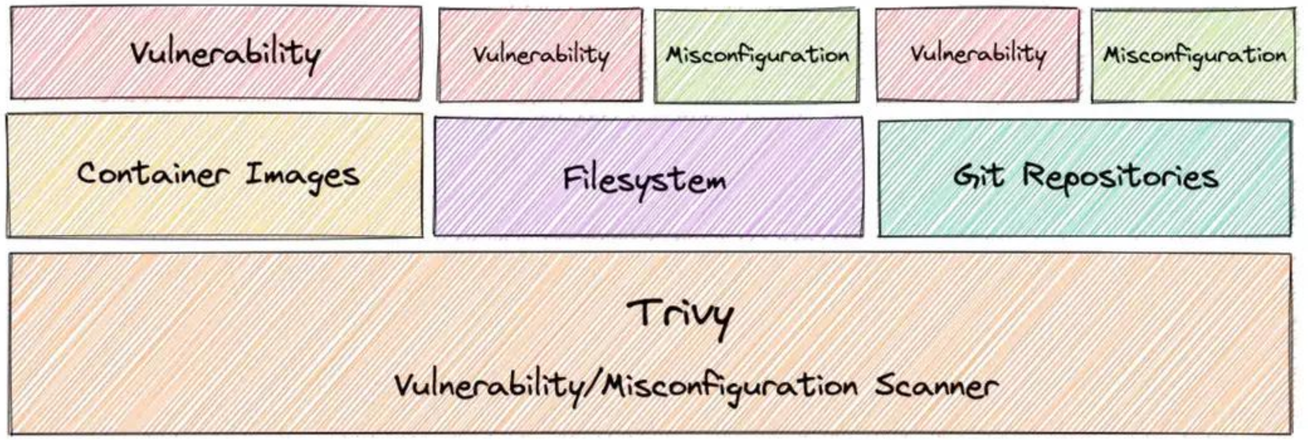
SonarQube is a **Code Quality Assurance tool that collects and analyzes source code**, and provides reports for the code quality of your project. It combines static and dynamic analysis tools and enables quality to be measured continually over time.



START TO LEAD

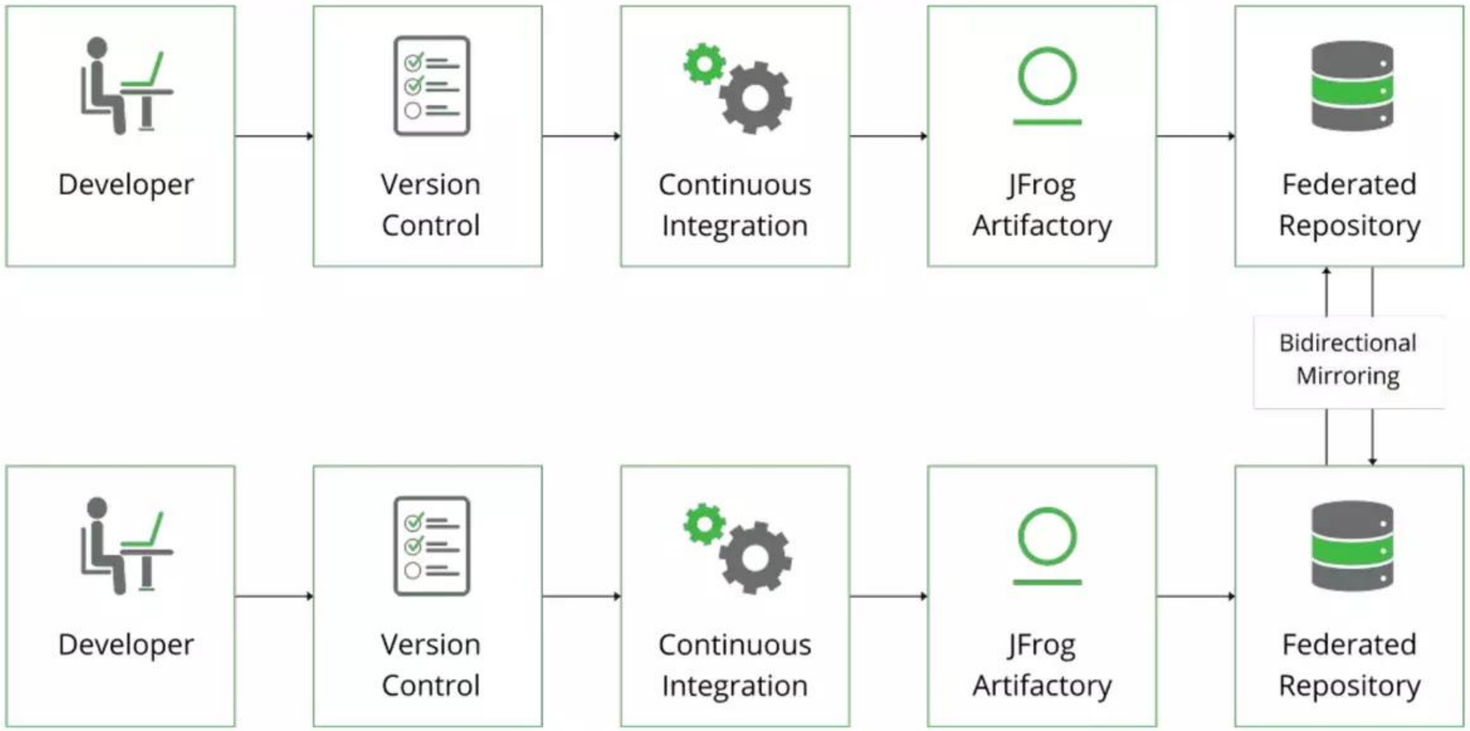
Trivy

- Trivy (tri pronounced like trigger, vy pronounced like envy) is a simple and comprehensive scanner for vulnerabilities in container images, file systems, and Git repositories, as well as for configuration issues.
- Trivy detects vulnerabilities of OS packages (Alpine, RHEL, CentOS, etc.) and language-specific packages (Bundler, Composer, npm, yarn, etc.).
- In addition, Trivy scans Infrastructure as Code (IaC) files such as Terraform, Dockerfile and Kubernetes, to detect potential configuration issues that expose your deployments to the risk of attack.
- https://aquasecurity.github.io/trivy/v0.21.3/

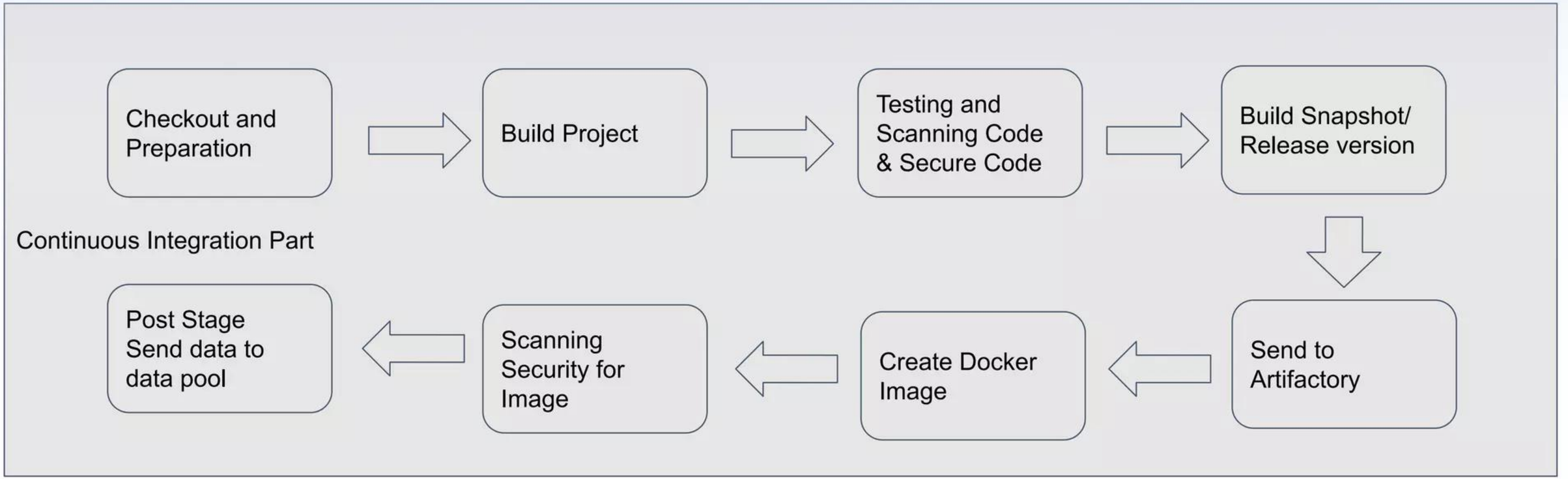


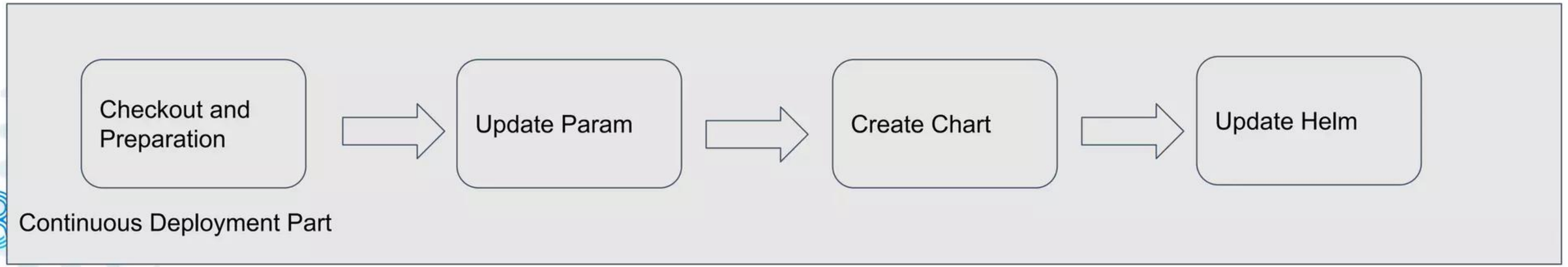
Artifact and Image Repository

- For App Artifact could use Jfrog Artifactory, Nexus, devpi for python, etc
- Docker Image (registry) : GCR, ECR, GCR



Pipeline CI/CD



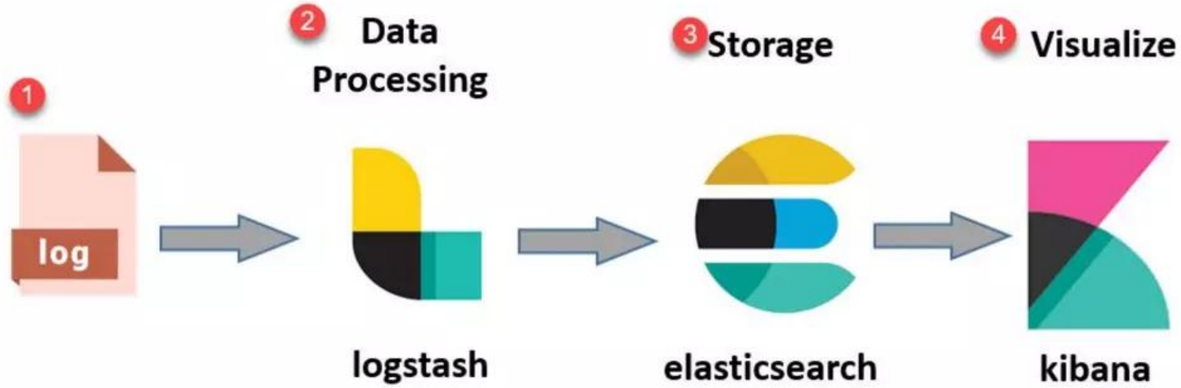


Create Repository for Images

- Base Image for Build
- Base Image for Development
- Default base image
- Jenkins pipeline
- Create standard for the project structure

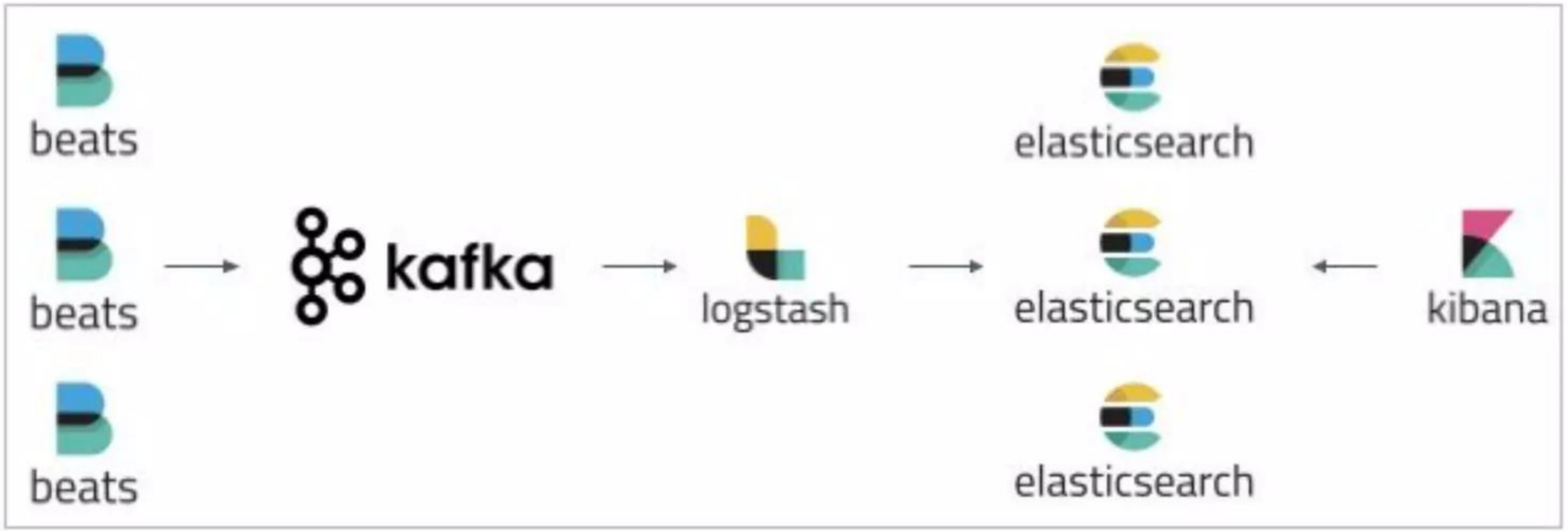
APM (Application Performance Management

- the monitoring and management of performance and availability of software applications. APM strives to detect and diagnose complex application performance problems to maintain an expected level of service.
- Tools: DataDog, New Relic, Splunk, ELK Stack (Elasticsearch Logstash Kibana), Grafana, Promotheus

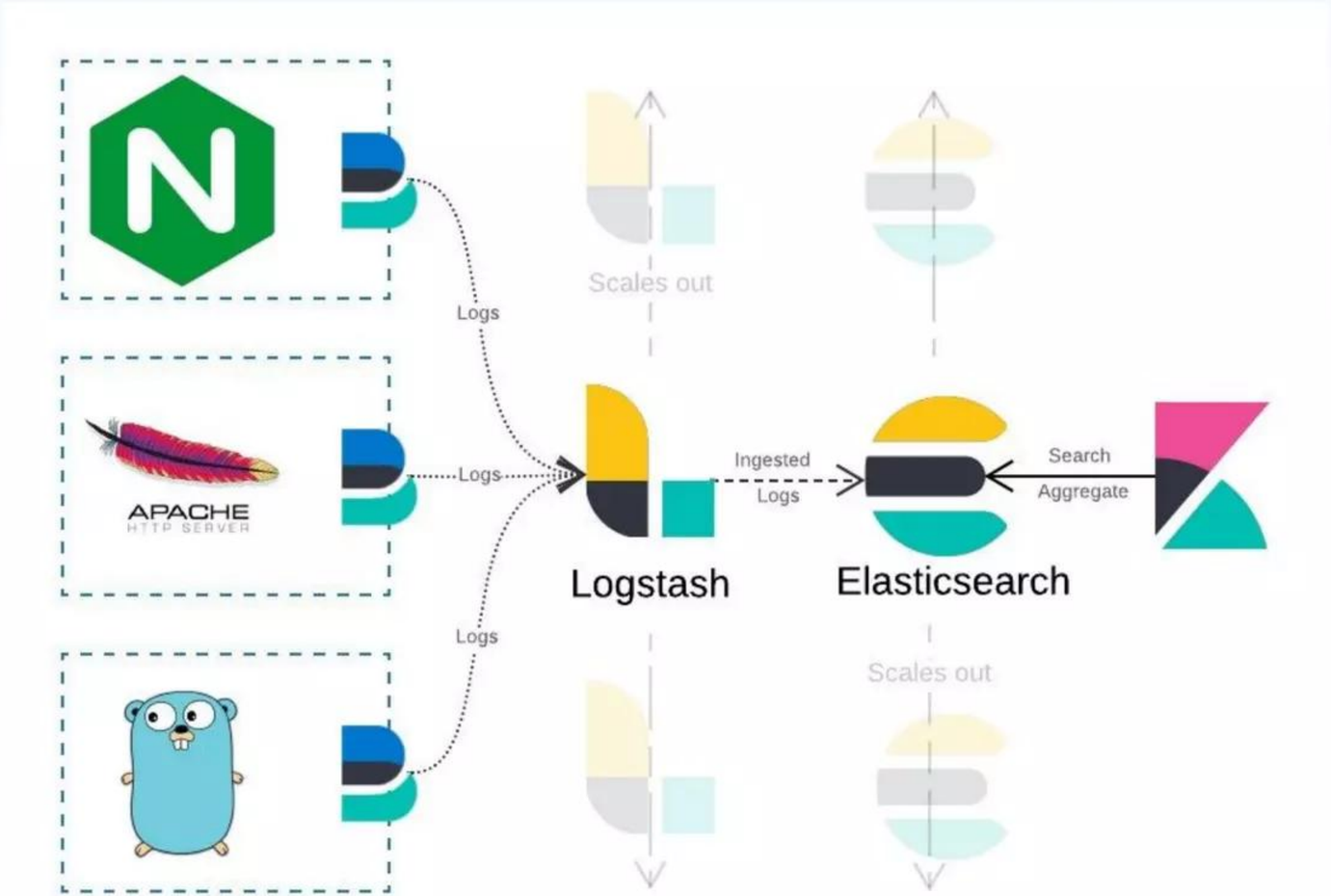


© guru99.com

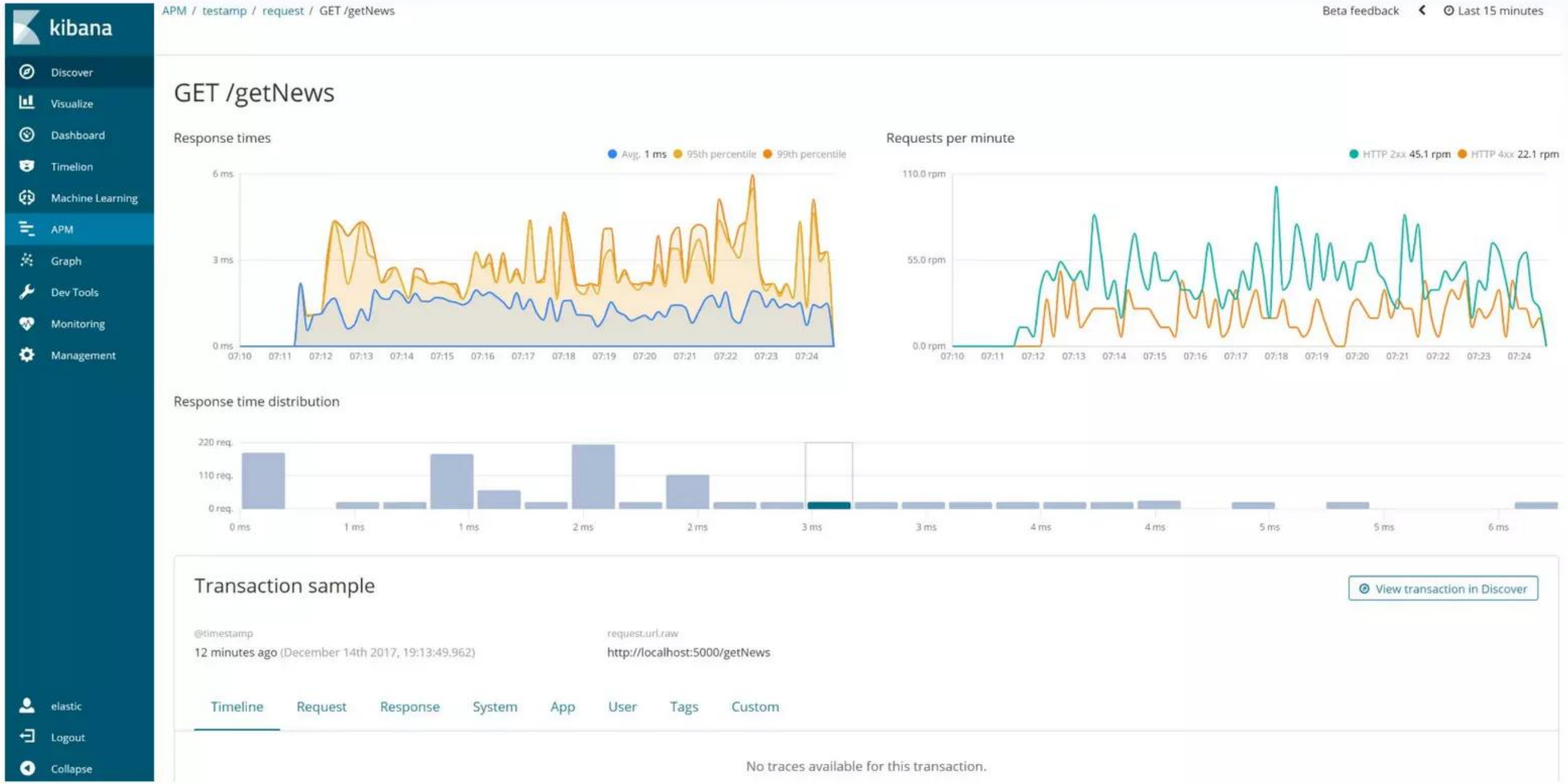
ELK Stack #1


beats
beats
beats
kafka
logstash
elasticsearch
elasticsearch
elasticsearch
kibana

ELK Stack #2

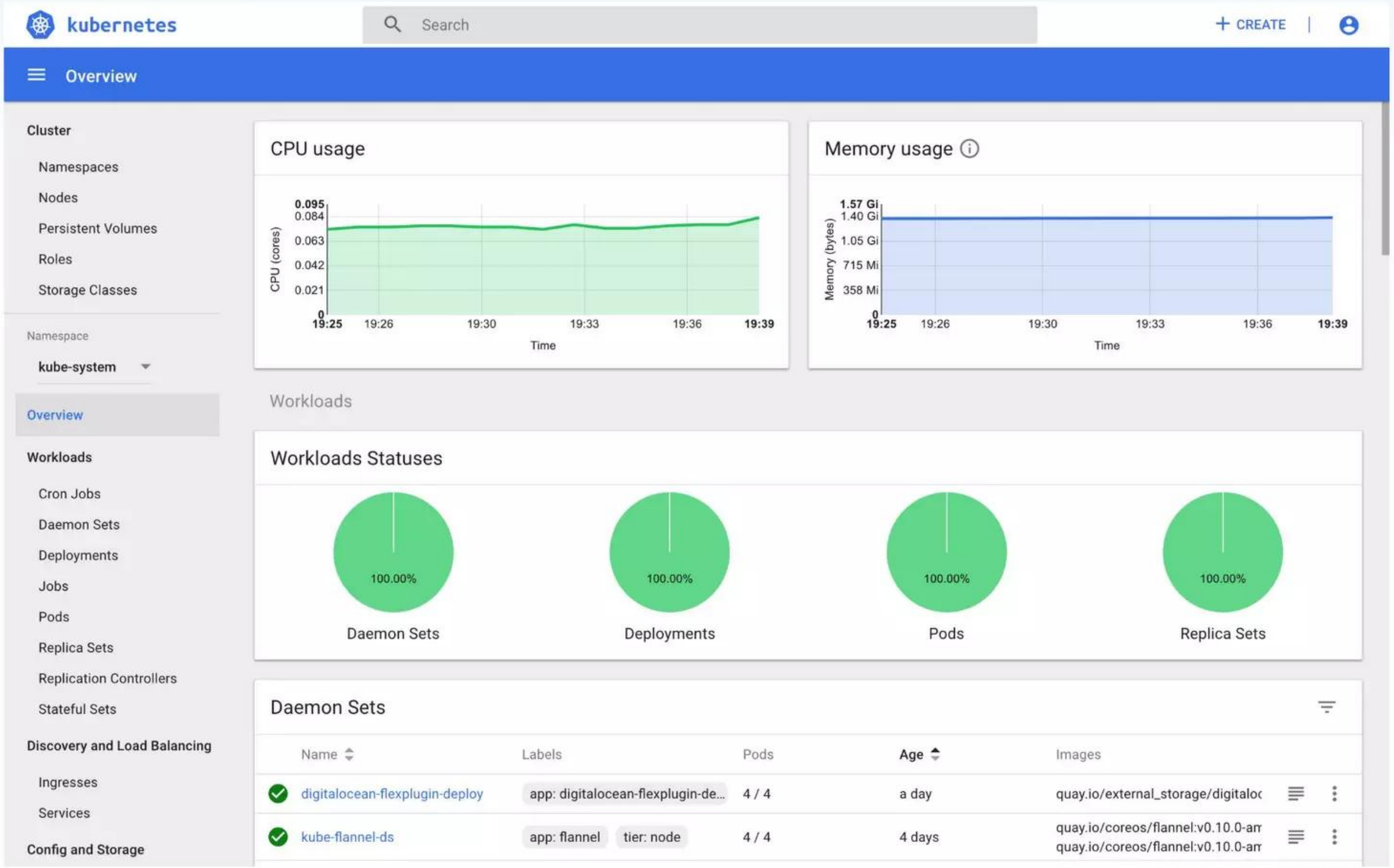

Logs
Logs
Logs
Scales out
Scales out
Ingested
Logs
Search
Aggregate
Logstash
Elasticsearch
APACHE
HTTP SERVER

ELK Stack #3

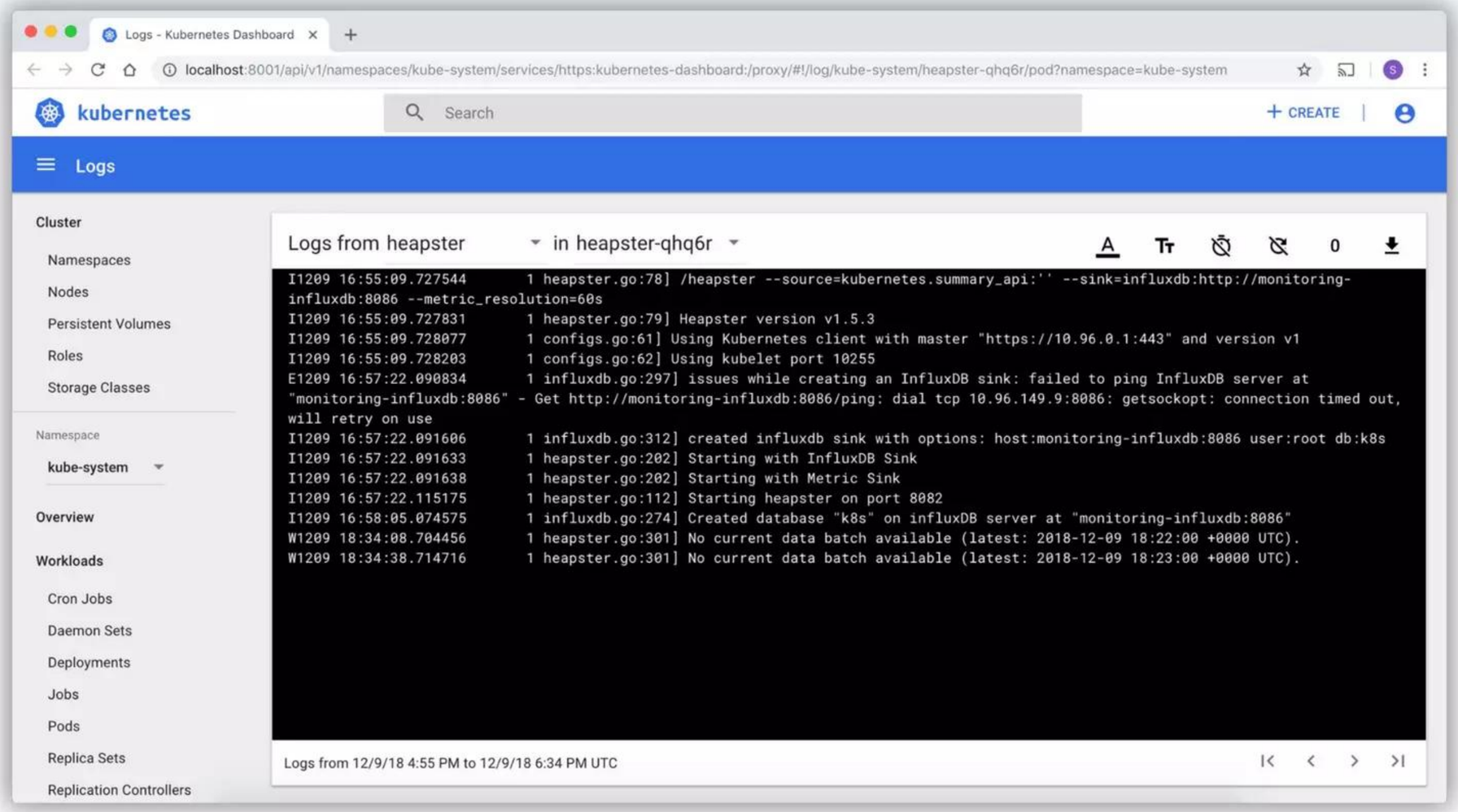


START TO LEAD

Kubernetes Dashboard



Kubernetes Dashboard



App Tech Stack

- Programming Language: java, kotlin, python, golang, php
- Database: mysql, postgre, sql server, mongodb, redis, etc
- Search engine: elasticsearch, solr
- Messaging app: RabbitMQ, Kafka

DevSecOps Tech Stack

Git Repo

JFrog ARTIFACTORY

Save app artifact

BigQuery

Optional, pick 1

CI Pipeline

Checkout → Build → Test & Analysis → Push To Artifactory → Build Image & Scan → Push to Image Registry

Live on to K8s

sonarqube

Scan Code coverage and secure code

trivy

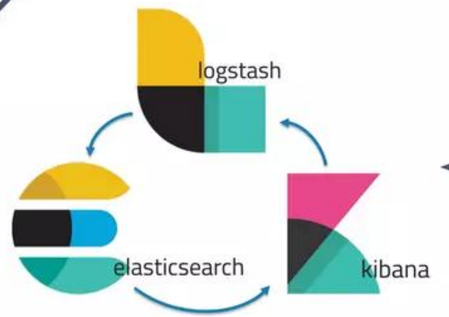
Scan Docker Image

Docker Private Registry

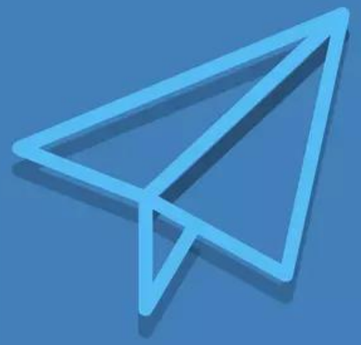
Save the image

CD Pipeline

Pull Image → Update Config → Helm upgrade



START TO LEAD



THANK YOU

NEURON
START TO LEAD

Leading Self
Leading Team
Leading Business