dynatrace

Observing a Kubernetes Cluster

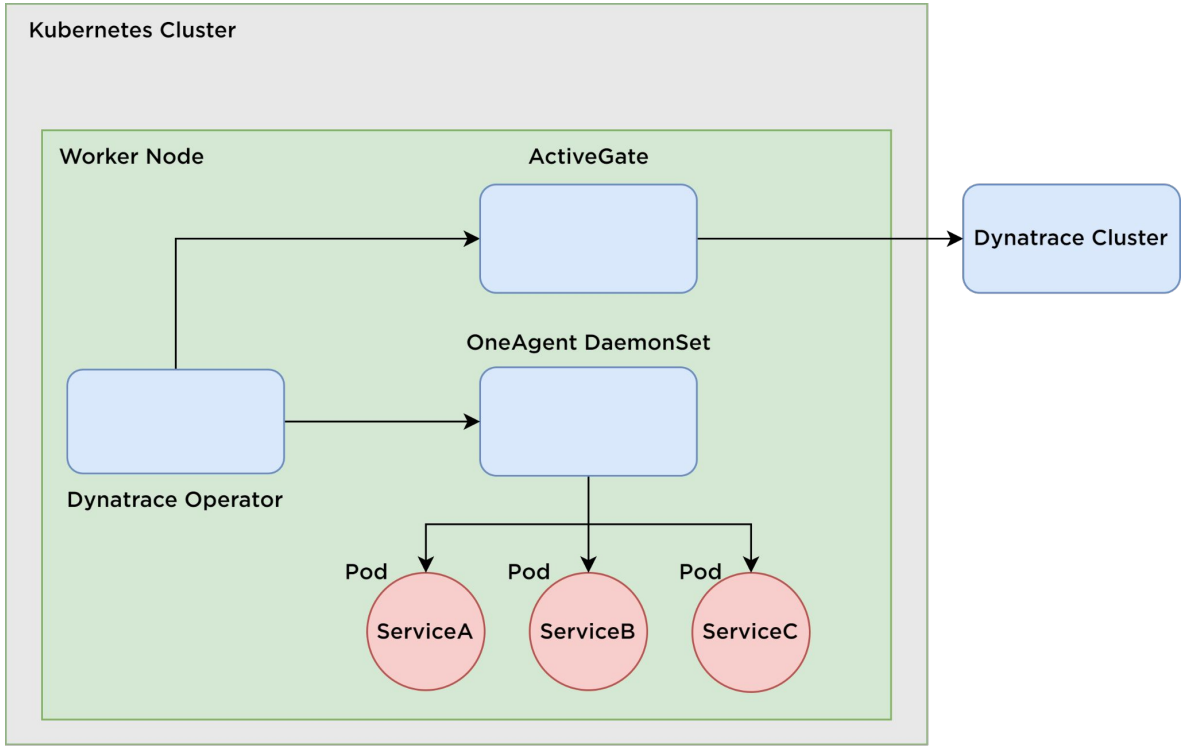
Kastenhofer, Sachsenhofer
Data Science and Engineering
WS22/23

Goal

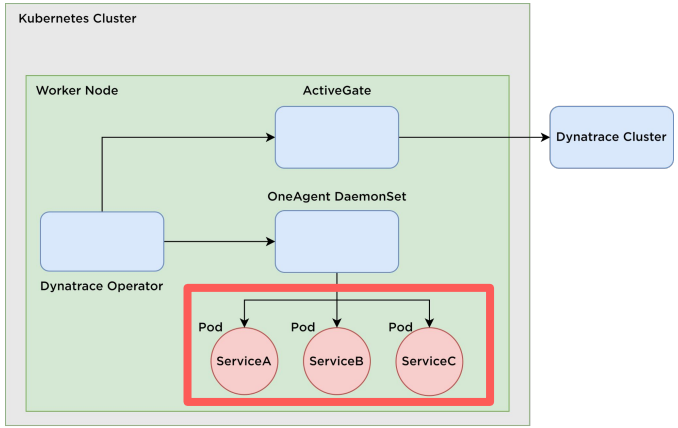
Observe K8s Cluster with Dynatrace

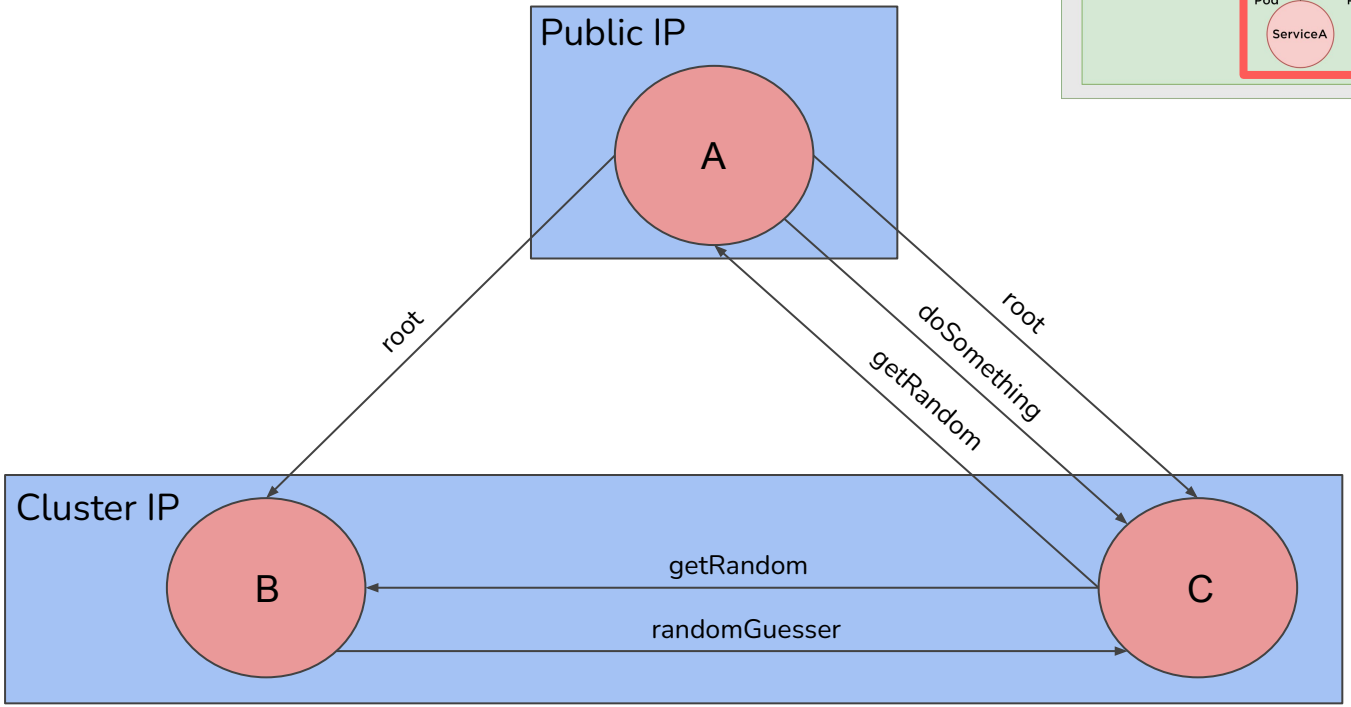
- Several services running in pods (at least 3)
- Get metrics about their health, status and how they are communicating
- The services are implemented in Java
- They throw randomly exceptions and produce delays
- The Dynatrace software should observe this behavior and visualize some metrics

Architecture

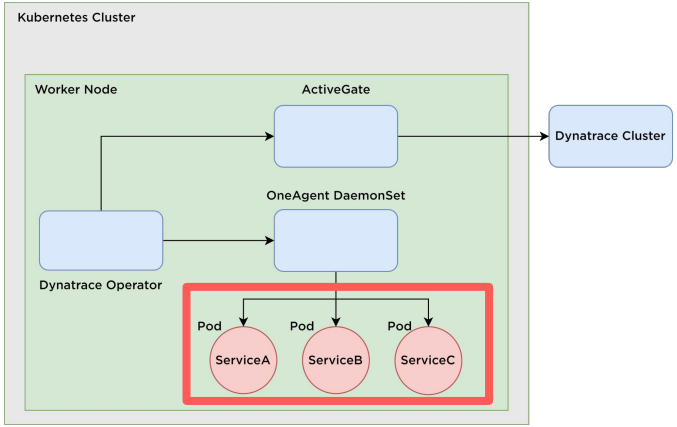


Services


Kubernetes Cluster
Worker Node
ActiveGate
Dynatrace Cluster
OneAgent DaemonSet
Dynatrace Operator
Pod
Pod
Pod
ServiceA
ServiceB
ServiceC

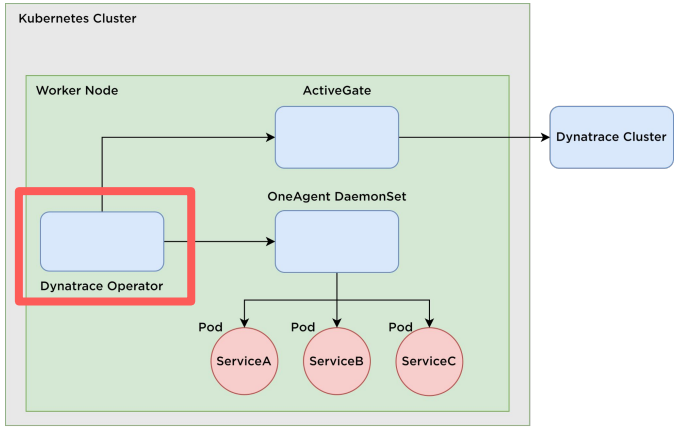

Public IP
A
root
root
doSomething
getRandom
Cluster IP
B
C
getRandom
randomGuesser

Services

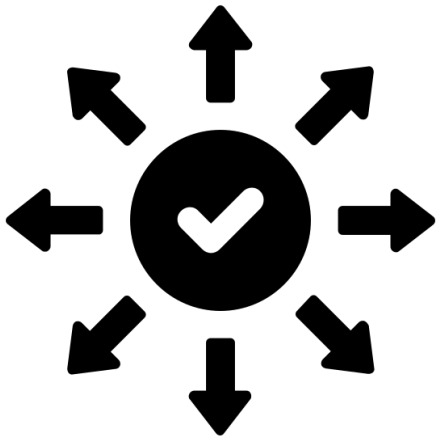



```
No  Completed at                        Action
0       0.46s                             Service C here!!
3       0.51s                             Service C here!!
4       0.54s                             Service C here!!
5       0.55s                             Service C here!!
7       0.57s                             Service C here!!
1       0.57s              Response A: 88 - Response B: 65!
8       0.58s                             Service C here!!
9       0.59s                    Random-Guesser result: 3!
2       0.61s               Response A: 18 - Response B: 0!
10      0.62s                             Service C here!!
6       0.62s              Response A: 44 - Response B: 95!
11      0.72s                    Random-Guesser result: 7!
13      0.72s {"timestamp":"2023-01-29T12:53:29.791+00:00","status":500,"error":"Internal Server Error","path":"/"}
14      0.74s              Response A: 37 - Response B: 30!
12      4.69s                             Service B Delay!
```

Dynatrace Operator

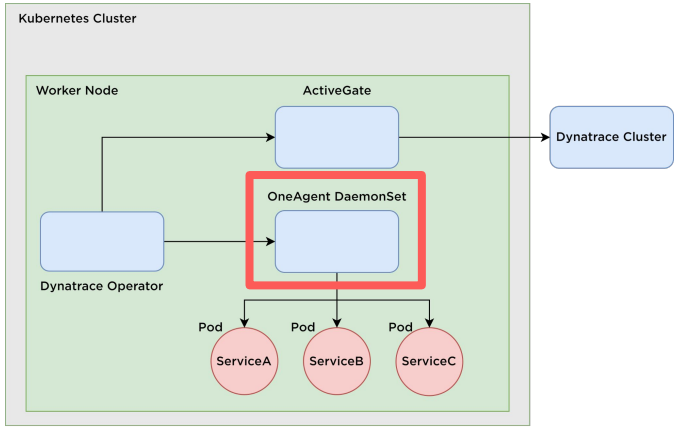


- automates **management** and **deployment** of Dynatrace in a Kubernetes cluster
- used to deploy and configure Dynatrace **OneAgent** and **ActiveGate**
- Enables easy and efficient management of Dynatrace monitoring environments



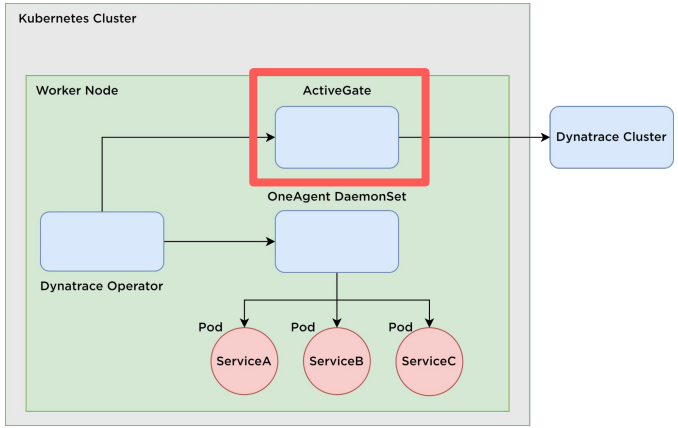
OneAgent

- a software agent that **collects** and **monitors performance data** from hosts, applications, and containers
- Provides operational and business performance metrics
- Discovers processes and activates instrumentation for unique application stack
- Injects tags for user-experience monitoring
- Running on **every node** in Kubernetes



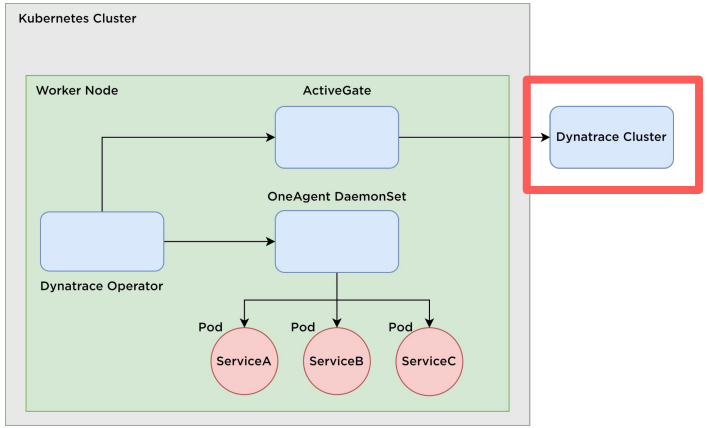
ActiveGate

- secure proxy between OneAgents and Clusters or other ActiveGates.
- Establishes Dynatrace presence in local networks
- Allows for traffic optimization, network complexity and cost reduction, and increased security
- Can perform monitoring tasks using APIs to query and monitor a wide range of technologies
- Can be deployed on physical or virtual hosts or as container

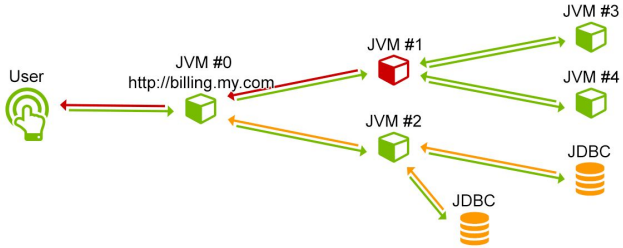


Dynatrace Cluster

- provide monitoring and performance analysis
- responsible for collecting data from the K8s
  - metrics and traces from pods, services and nodes
- Interface to users to interact with data
  - custom dashboards and visualizations



Distributed Tracing

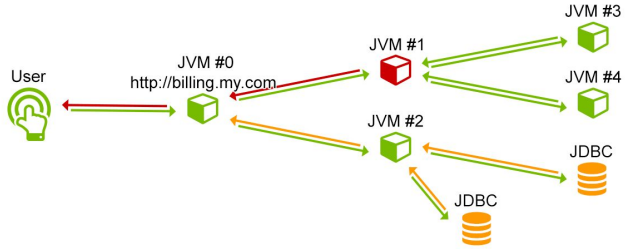


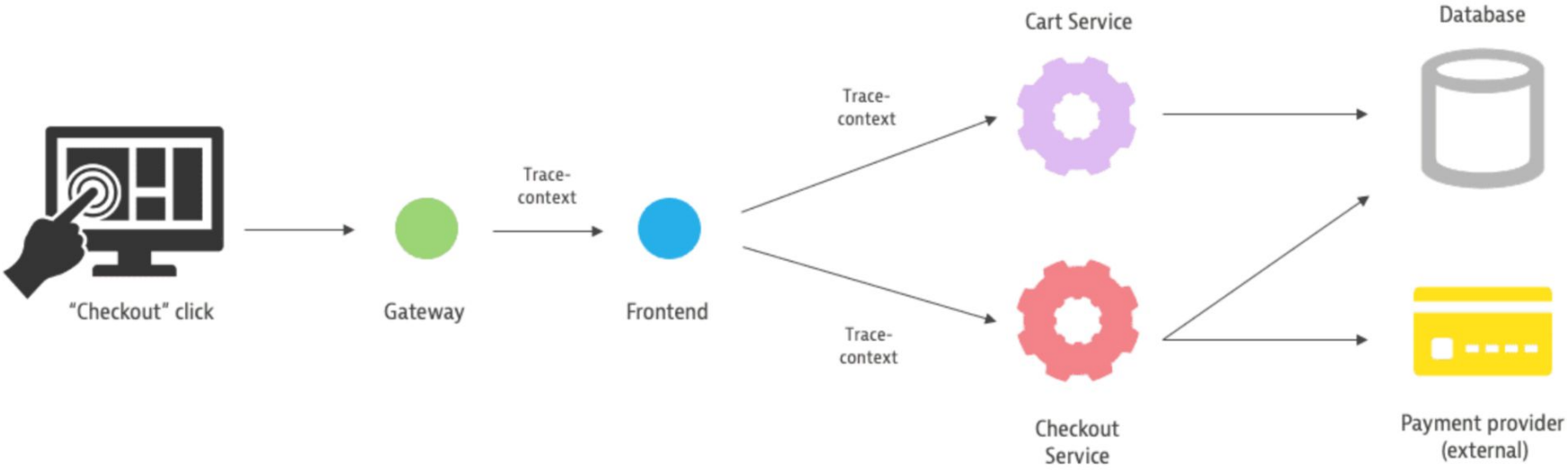
- Distributed Tracing: Track and analyze the flow of a request as it travels through multiple systems
- Understand performance and behavior of distributed applications
  - identify bottlenecks and errors

PurePath

- combine Distributed Tracing with Code-level visibility, topology information and metadata
- OneAgent captures PurePath traces across transactions
  - method-level analysis
  - detailed error analysis
  - database analysis
  - request attributes
  - multidimensional analysis

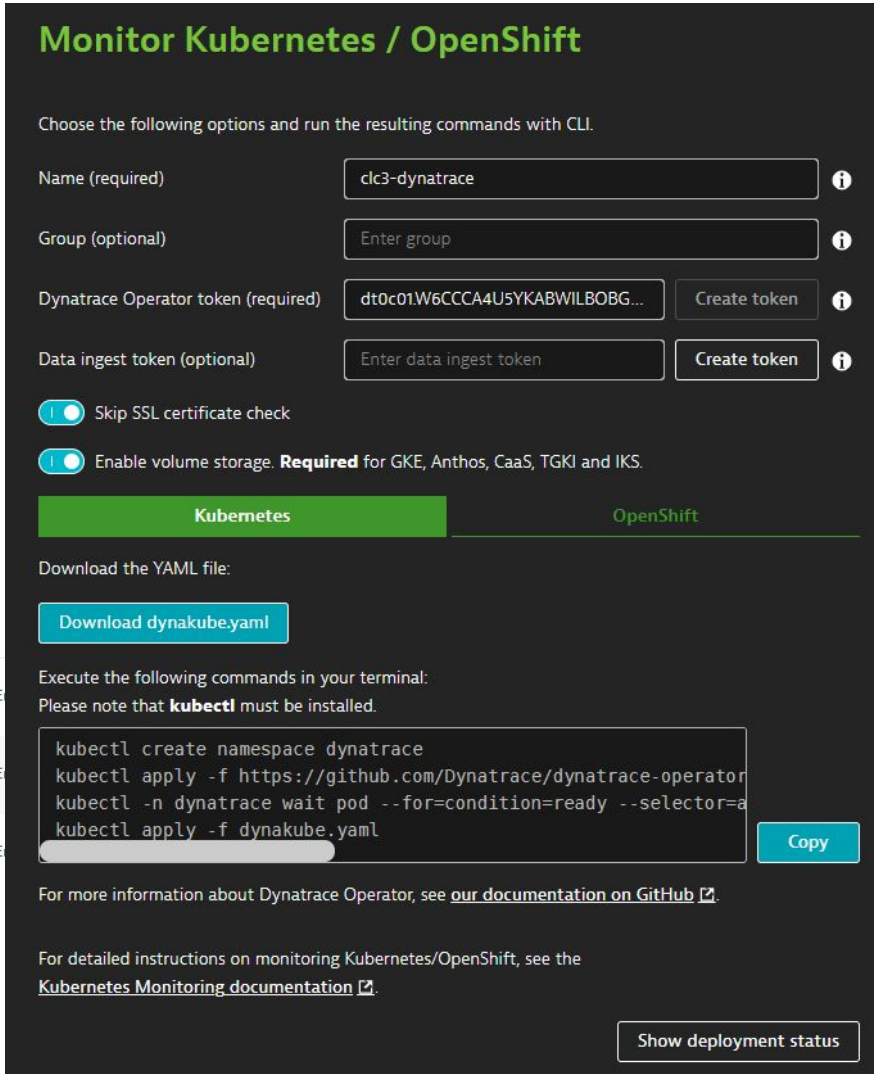
Distributed Tracing PurePath





Steps

1. implement Services in Java
2. create Docker Container
3. set up GKE (min. 3 Nodes)
4. run Services on GKE
5. deploy Dynatrace on Kubernetes
6. restart pods



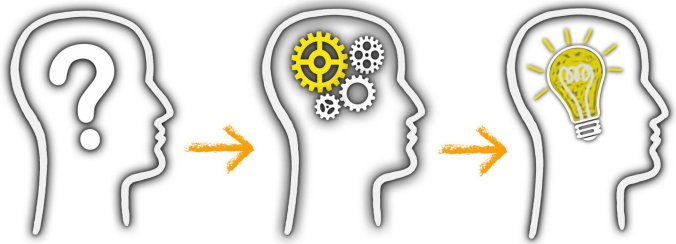
Live Demo | http://34.170.200.97/

1. Show Deployment Status
2. Metrics - Dashboard - System Behavior
3. Distributed Tracing (PurePath)

Repository

https://github.com/Phil91ip/clc3-dynatrace

Lessons Learned



- Docker hub is a powerful service for sharing container images
- Dynatrace has certain requirements for monitoring, such as a minimum number of nodes
- Running a Kubernetes cluster on Google Cloud could get expensive
- Running a Kubernetes cluster also requires maybe certain requirements in configuration
- Dynatrace is very powerful and offers many capabilities