Translucent

Developing Slack Bot for Kubernetes

Application Features

Slack Bot key features

- Ability to respond to events in a Slack Channel
- Ability to access Kubernetes resources
- Two-way communication with interactivity features

Interactions

- Use bot mentions to send messages to the bot -> @webinar Hi!
- Bot responds to messages -> **Sleeeeeping** 😴
- Provide trigger phrase to wake up the bot -> @webinar Time to wake up!
- Bot initiates to way-communication with interactive elements -> Uses buttons

- Based on the response it provides feedback ->

Let's see if the universe has something good to say.
"There is nothing permanent except change." by Heraclitus

Thread

patrykattc 2 minutes ago
@webinar what's up?

3 replies

webinar APP 2 minutes ago
Sleeeeeping 😴

patrykattc 2 minutes ago
@webinar time to wake up

webinar APP 2 minutes ago
Hi @patrykattc! How are you feeling!

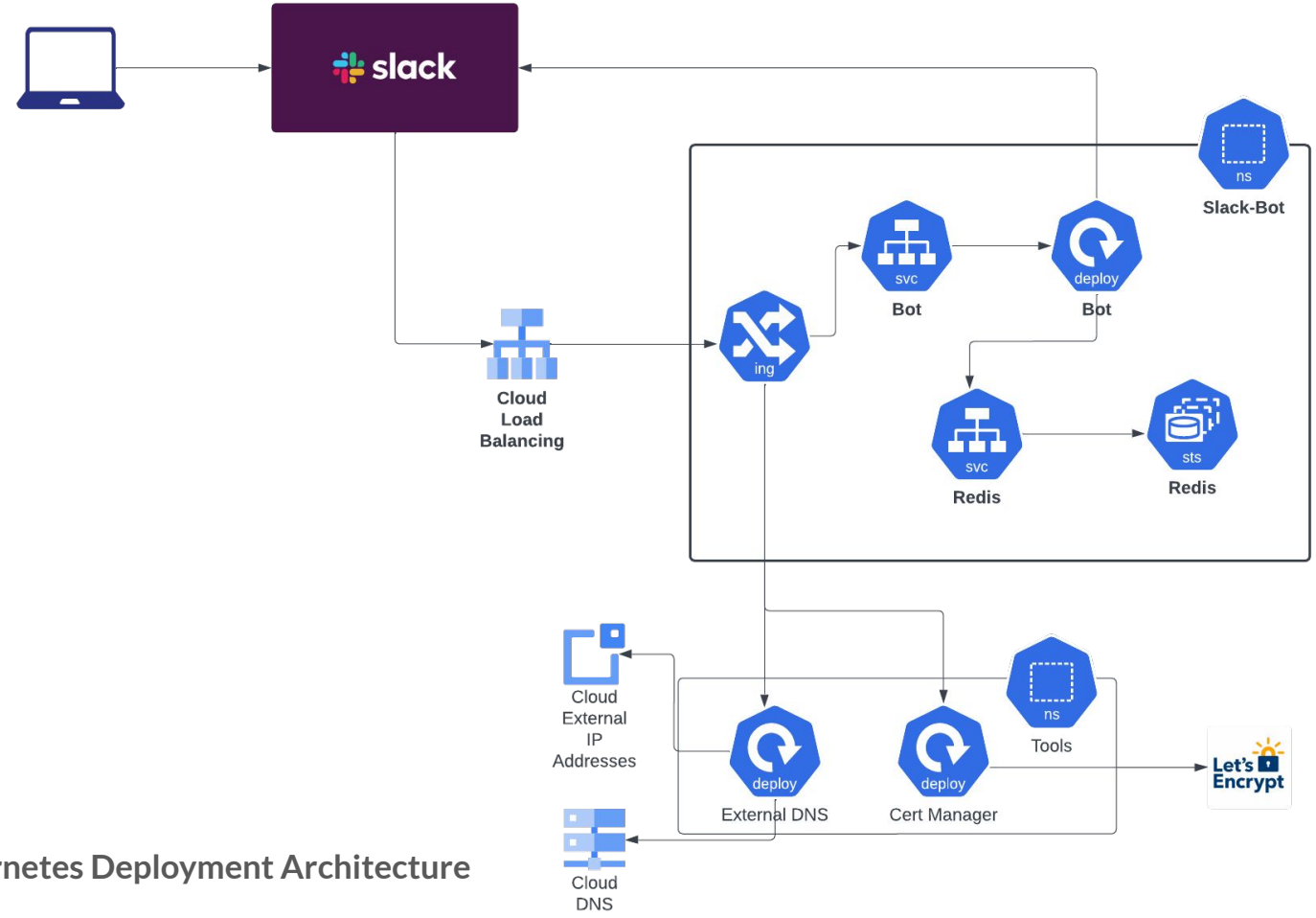
Good Bad

Reply...

Also send to # **webinar-slack-bot**

Kubernetes Deployment Architecture

- **Namespace**: A Kubernetes namespace (ns) dedicated to the Slack Bot for logical separation within the cluster.
- **Ingress**: The entry point for the Slack service, managed by an Ingress controller (ing) to route traffic to the bot service.
- **Bot Service**: A Kubernetes service (svc) that serves as an abstraction layer to expose the bot application.
- **Bot Deployment**: The deployment (deploy) managing the bot application pods in the Kubernetes cluster.
- **Redis**: A deployment of Redis with its service (svc) for storing bot data, with a StatefulSet (sts) for data persistence.
- **Cloud Load Balancing**: Distributes incoming Slack traffic efficiently to the Kubernetes services.
- **External Resources**: Includes Cloud External IP Addresses, Cloud DNS for domain name resolution, External DNS deployment, and a Cert Manager deployment for handling SSL certificates, with Let's Encrypt for certificate provisioning.

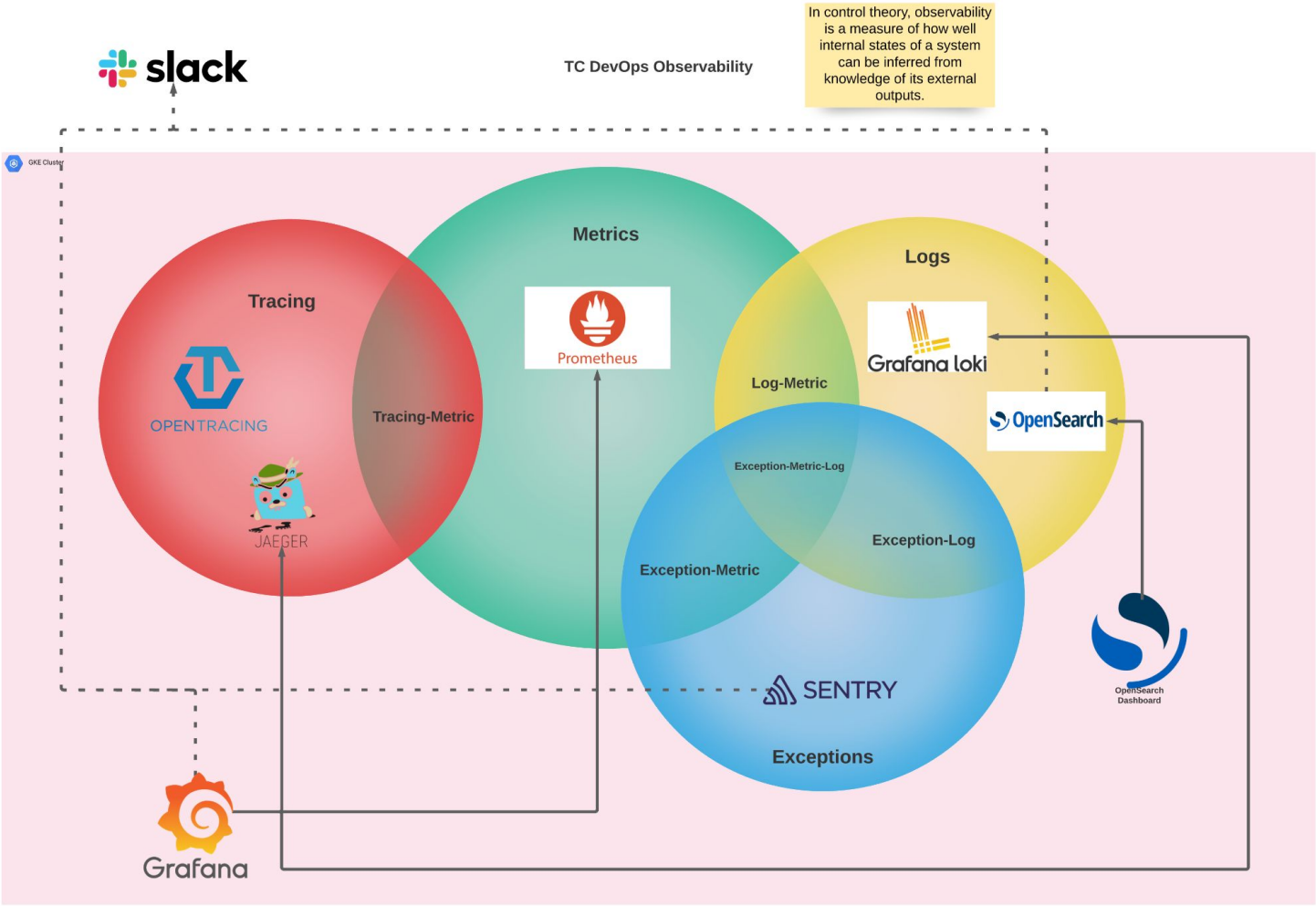


Kubernetes Deployment Architecture

Kubernetes Observability

Observability Stack to support Slack Bot Deployment

- Tracing
  - Jaeger: A tracing system used for monitoring and troubleshooting microservices-based distributed systems.
  - Integrated with Grafana for visualizing trace data.
- Metrics
  - Prometheus: An open-source monitoring system with a dimensional data model, flexible query language, and alerting functionality.
  - Metrics from both Prometheus and Jaeger are used to gain insights into the system's performance.
- Logs
  - Grafana Loki: A horizontally scalable, highly available, multi-tenant log aggregation system inspired by Prometheus.
  - OpenSearch (formerly Elasticsearch): An open-source search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured.
  - OpenSearch Dashboard: A visualization tool in the OpenSearch stack for analyzing log data.
- Exceptions
  - Sentry: An error tracking tool that helps developers monitor and fix crashes in real time.


slack
TC DevOps Observability
In control theory, observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs.
GKE Cluster
Tracing
OPENTRACING
JAEGER
Tracing-Metric
Metrics
Prometheus
Logs
Grafana loki
Log-Metric
OpenSearch
Exception-Metric-Log
Exception-Log
Exception-Metric
SENTRY
Exceptions
OpenSearch Dashboard
Grafana

Bot Technology Stack: FastAPI & Slack Bolt

- Web Application Framework
  - FastAPI: A modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints.
- Slack Integration
  - Slack Bolt Framework: A foundational framework for building Slack apps with minimal setup.
  - Capabilities: Simplifies the process of handling Slack events, actions, and commands.
- Redis Database
  - Used to support data needs for the bot.
- Containerization
  - Docker: Used to containerize the FastAPI application, ensuring consistency across various development and production environments.
  - Container Registry: Stores the Docker images, ready to be pulled into the Kubernetes cluster.

The architecture and deployment of the Slack Bot within a Kubernetes cluster showcases the symphony of modern technology. FastAPI and Slack Bolt framework come together to create a seamless and powerful user experience, while Docker and Kubernetes orchestrate the performance of a robust, scalable application.