INDONESIA
OpenInfra Days

02.11.2019 | Surabaya, Indonesia

Linkerd 2.0

Observability, Reliability, and Security. Ultralight Service Mesh for Kubernetes

Ananda Dwi Rahmawati – Cloud Eng, Linux IaaS

ananda@btech.id

Biznet GioCloud Biznet Mellanox TECHNOLOGIES BANK BRI OSF OpenStack Foundation

<>

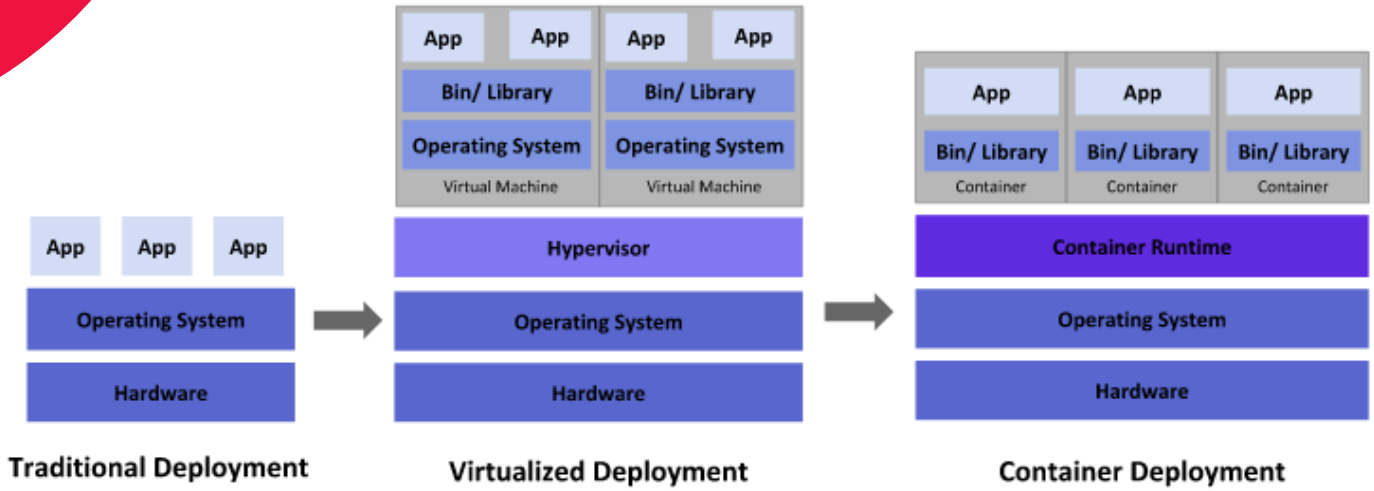
Who am i ?

- Cloud Engineer at PT Boer Technology (Btech)
- Infrastructur Team at BlankOn Linux Indonesia
- FLOSS Enthusiast
- Mahasiswa
- Tap me at :
  - +62 8132 6789 108
  - t.me/misskecupbung
  - ananda@btech.id
  - https://linkedin.com/in/anandadwir
  - https://misskecupbung.wordpress.com

Resource(s)

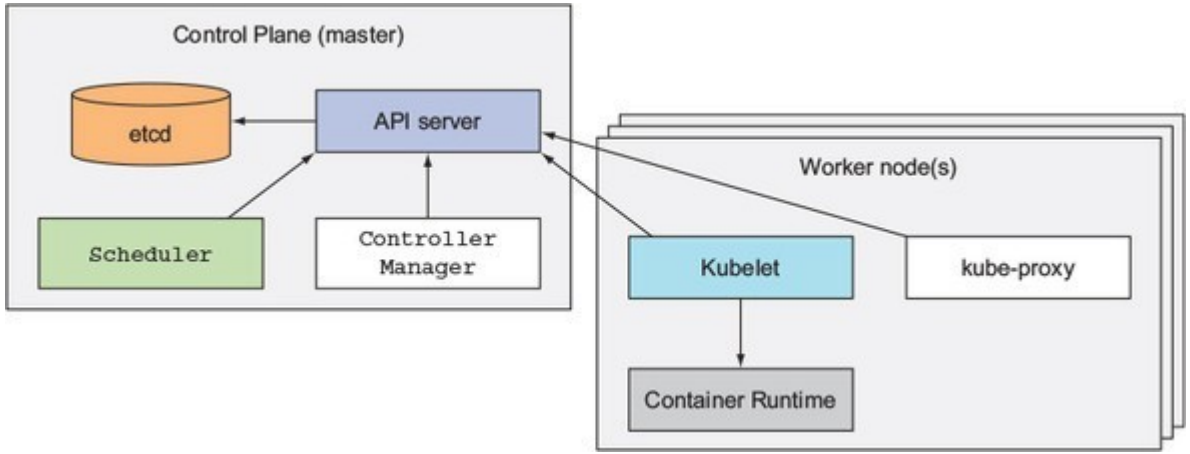
- Linkerd Documentation https://linkerd.io/2/overview/
- Kubernetes Documentation: https://kubernetes.io/docs/home/

Container Evolution

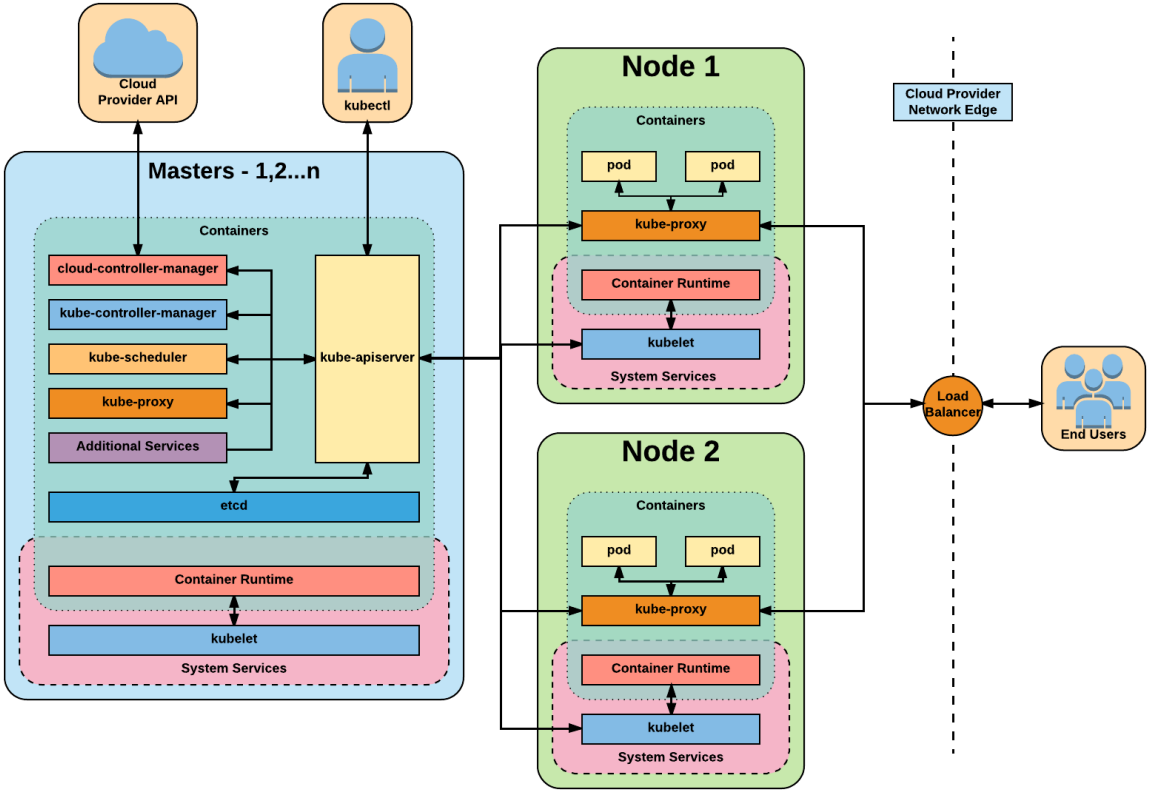


Kubernetes

"Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications."



Kubernetes Cluster

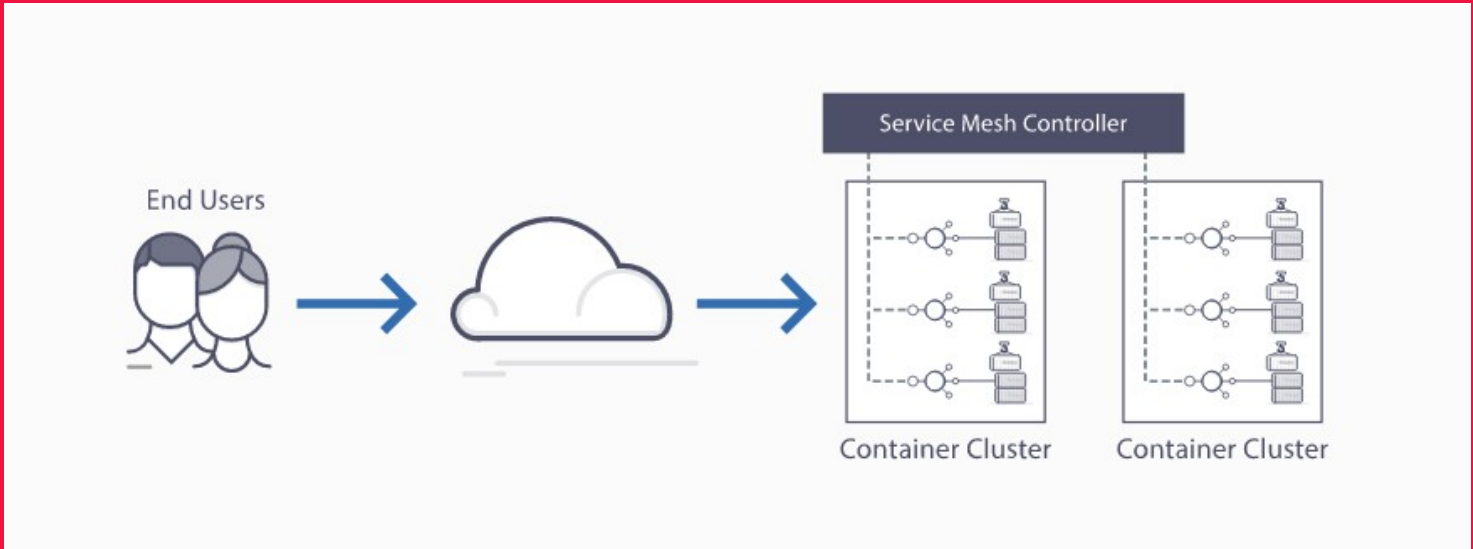


DO WE NEED A SERVICE MESH ?

The Term of ’Service Mesh’

A service mesh is a dedicated infrastructure layer for making service-to-service communication safe, fast, and reliable. If you’re building a cloud native application, you need a service mesh.

The Term of ’Service Mesh’



Service Mesh Comparison

Feature	Istio	Linkerd	Consul Connect
Traffic Redirection (Blue/Green deployment)	Yes	No	No
Traffic Splitting (Canary deployment)	Yes	No	No
Attribute based routing	Yes	No	No
Service Identification	Yes	No	Yes
Auto Proxy Injection	Yes	Yes	Yes
Non-Admin installation	No	Yes	No
Built-in Dashboard	Yes	Yes	No
Certificate Management	Yes	No	Yes

Service Mesh Comparison

Feature	Istio	Linkerd	Consul Connect
Metrics Collection	Yes	Yes	No
Built-In Dashboard	Yes	Yes	No
TLS	Yes	Yes	Yes
External Service Support	Yes	No	Yes
Rate Limiting	Yes	No	No
Tracing	Yes	No	No

Linkerd 2.0

Initially started as a network proxy (v1.0) for enabling service mesh.

Merged with Conduit to form Linkerd 2.0 in Sept 2018

LINKERD

<>

An open source service mesh and CNCF member project

- 24+ months in production
- 3,000+ Slack members
- 100+ contributors
- 10,000+ GitHub stars
- 40m+ DockerHub pulls

<>

An open source service mesh and CNCF member project

salesforce
OfferUp
CHASE
COMCAST
monzo
Expedia
planet.
STRAVA
credit karma
BIGCOMMERCE
Olark
FOX

Linkerd is a transparent proxy that adds service discovery, routing, failure handling, and visibility to modern software applications

- Integration service discovery
- Provides dynamic, scoped, logical routing rules, enabling blue-green deployments, staging, canarying, failover,and more.
- Multi-container orchestration supported
- Handles tens of thousands of requests per second per instance with minimal latency overhead. Scales horizontally with ease
- Zipkin, Prometheus and statsd integration

Why Linkerd ?

Thriving open source community

Linkerd is 100% Apache-licensed, with an incredibly fast-growing, active, and friendly community. Come join the fun!

Ultralight and ultra fast

Built in Rust, Linkerd's data plane proxies are incredibly small (<10 mb) and blazing fast (p99 < 1ms).

Simple, minimalist design

No complex APIs or configuration. For most applications, Linkerd will "just work" out of the box.

Installs in seconds with zero config

Linkerd's control plane installs into a single namespace, and services can be safely added to the mesh, one at a time.

Deep Runtime Diagnostics

Get a comprehensive suite of diagnostic tools, including automatic service dependency maps and live traffic samples.

Actionable service metrics

Best-in-class observability allows you to monitor golden metrics—success rate, request volume, and latency—for every service.

Linkerd Capabilities

No need to be a Platform admin to use linkerd

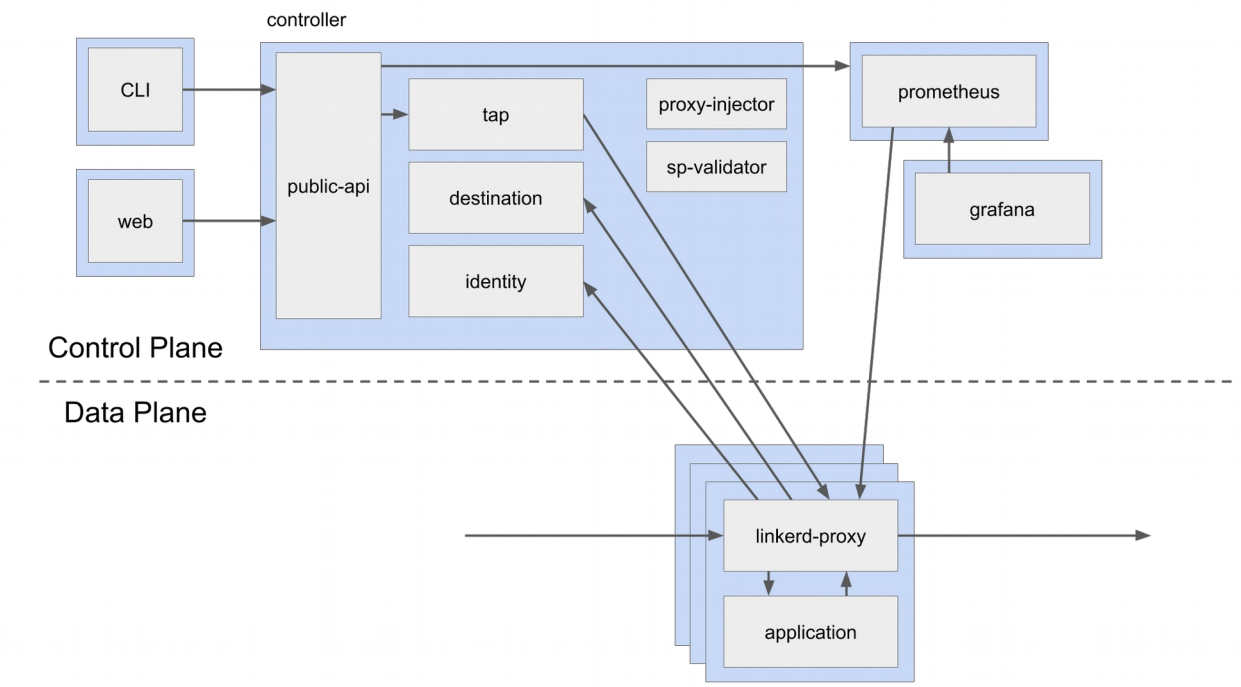
Simple installation and CLI tools to get started

Small sidecar proxy written in Rust

Can do end-to-end encryption and automatic proxy injection

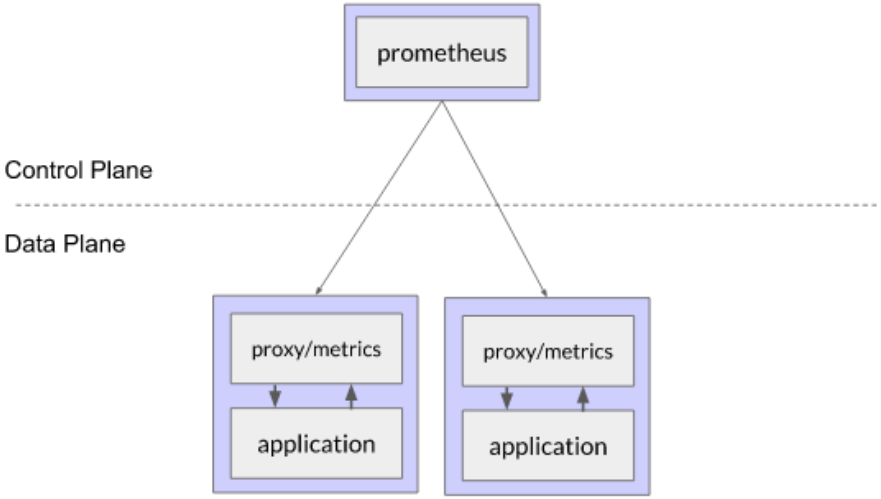
Lacks complex routing and tracing capabilities

Linkerd Architecture

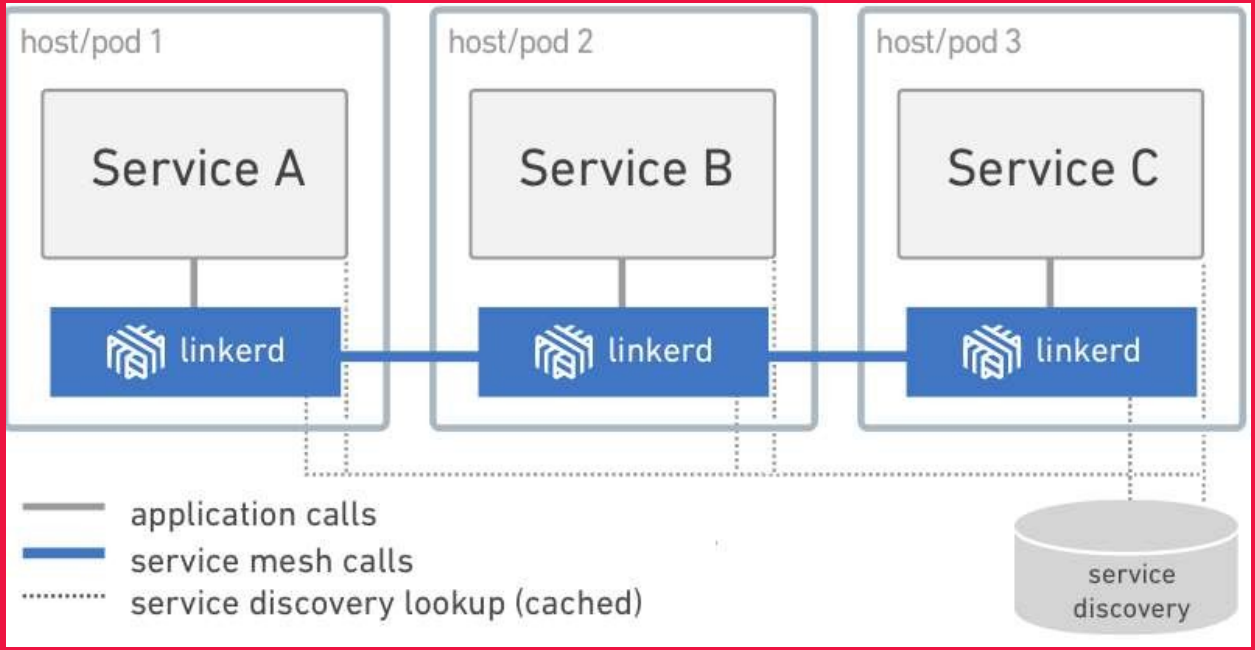


Linkerd Architecture

- Control Plane:
  - Controller
  - Grafana
  - Identity
  - Prometheus
  - Proxy Injector
  - Service Provide Validator
  - Tap
  - Web
- Data Plane



Example :


host/pod 1
Service A
linkerd
host/pod 2
Service B
linkerd
host/pod 3
Service C
linkerd
application calls
service mesh calls
service discovery lookup (cached)
service
discovery

Linkerd Architecture

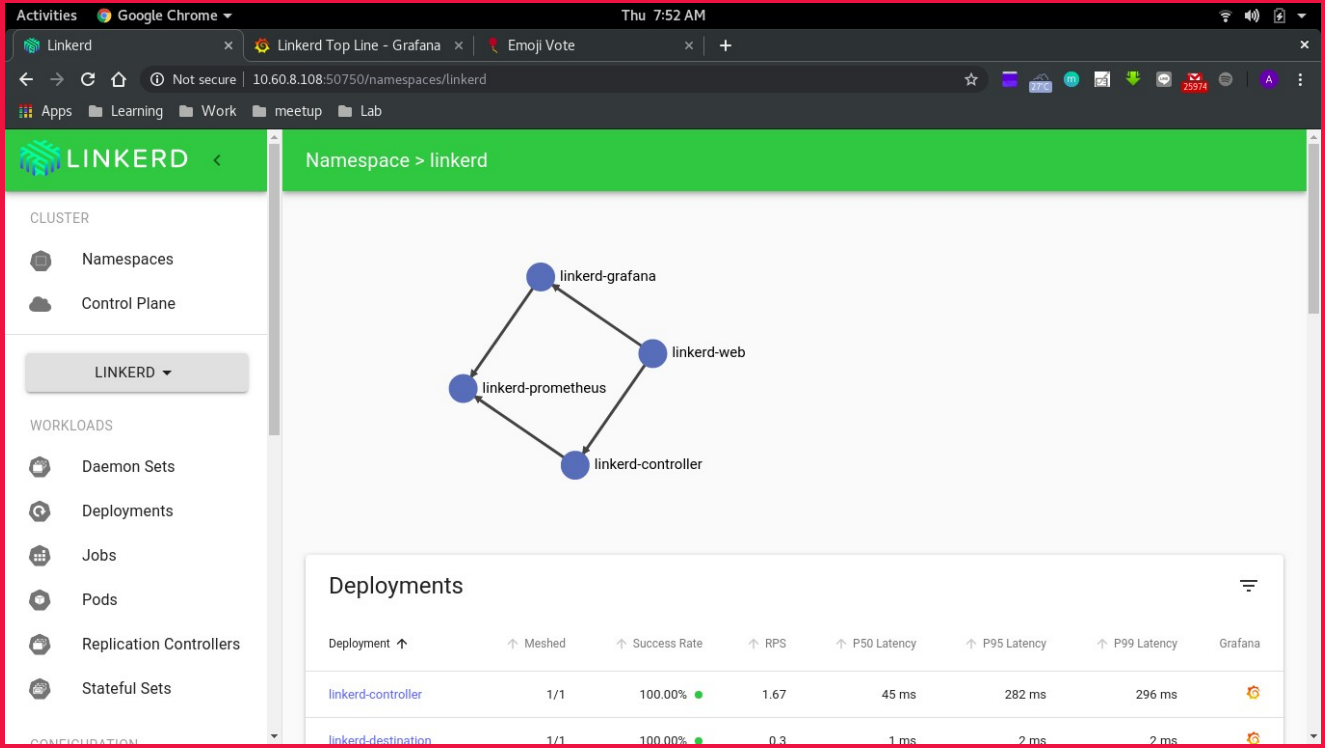
- Application in containers register to service Discovery as service
- Linkerd gets services from services Discovery
- Application communicate by linkerd through http_proxy variable or directly by node_name variable.
- Containers must connect to linkerd in your own host/ hypervisor.
- Linkerd balance or forward connection to another linkerd.

Linkerd Commands

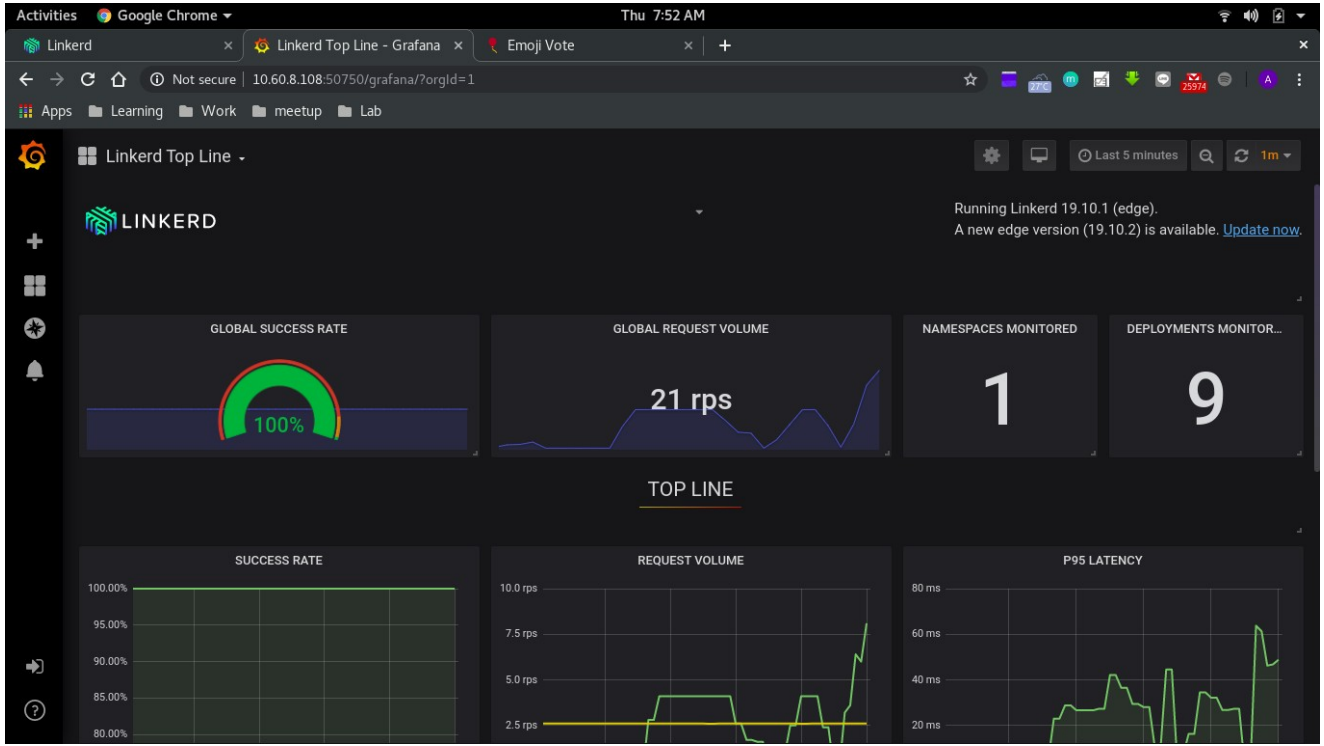
K9s Verify:
kubectl version --short
Install:
curl -sL https://run.linkerd.io/install | sh
export PATH=$PATH:$HOME/.linkerd2/bin
linkerd check --pre
linkerd install | kubectl apply -f -
Dashboard
linkerd dashboard &

Inject:
kubectl get -n emojivoto deploy -o yaml \
| linkerd inject - \ |kubectl apply -f -
Inspect:
linkerd -n emojivoto stat deploy
linkerd -n emojivoto top deploy
linkerd -n emojivoto tap deploy/web

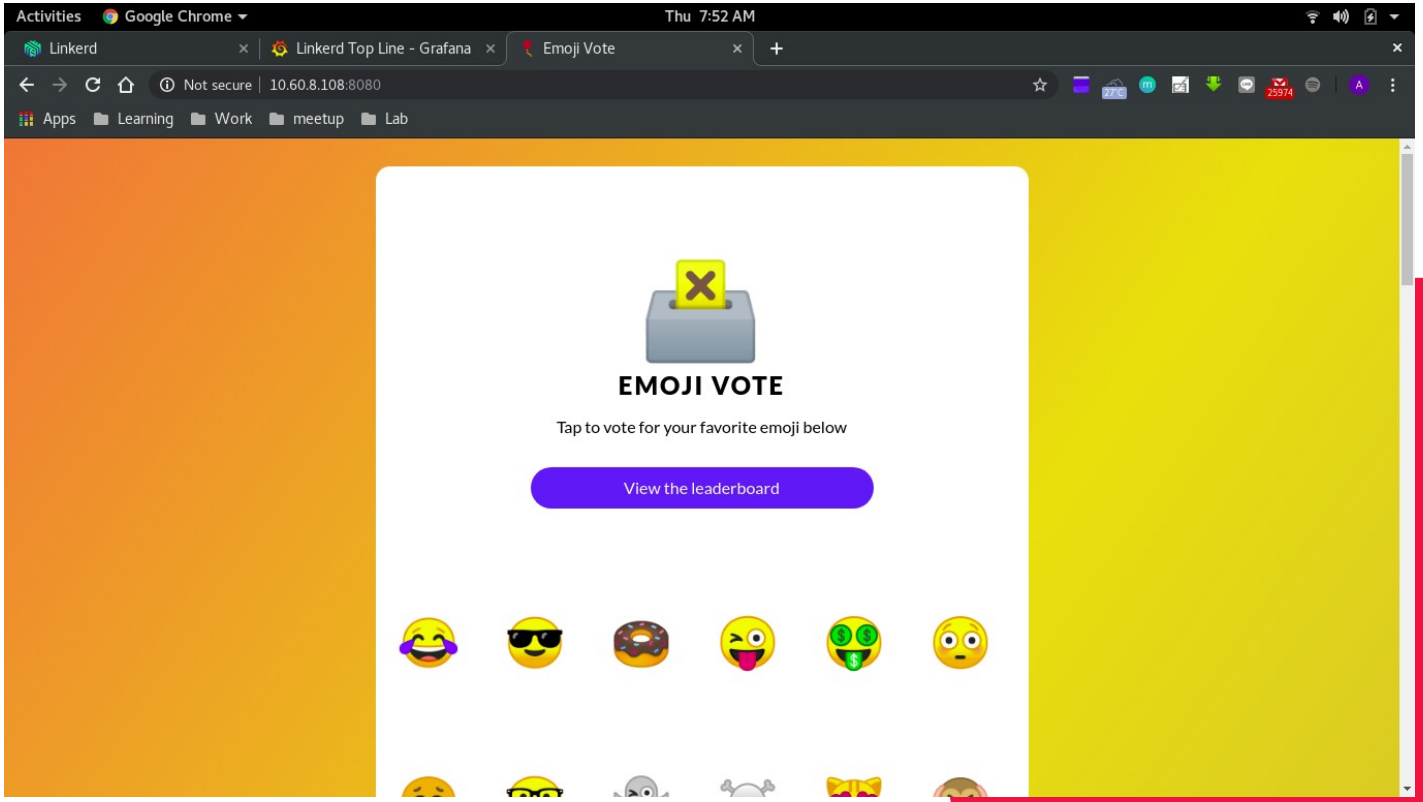
Linkerd Dashboard



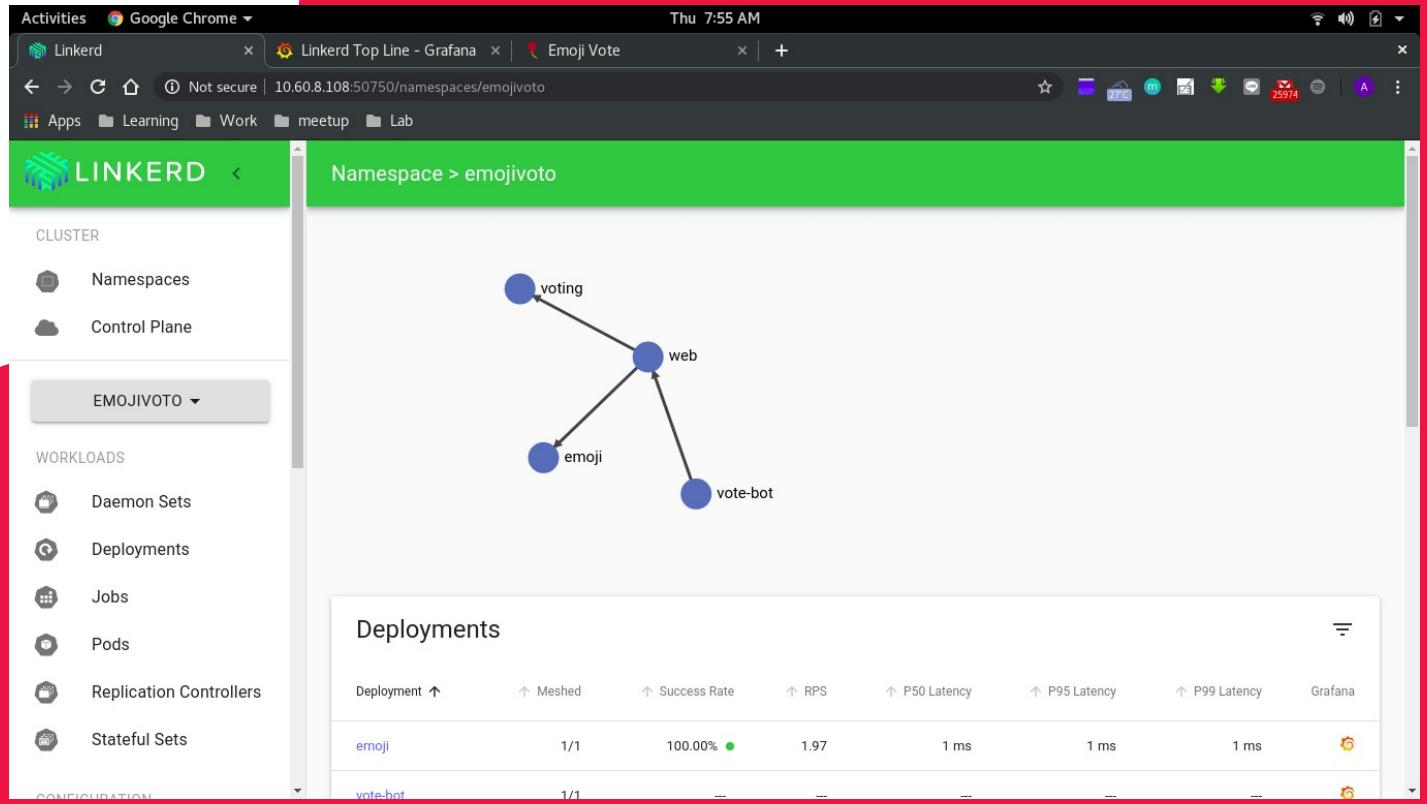
Linkerd Traffic Monitoring using Grafana



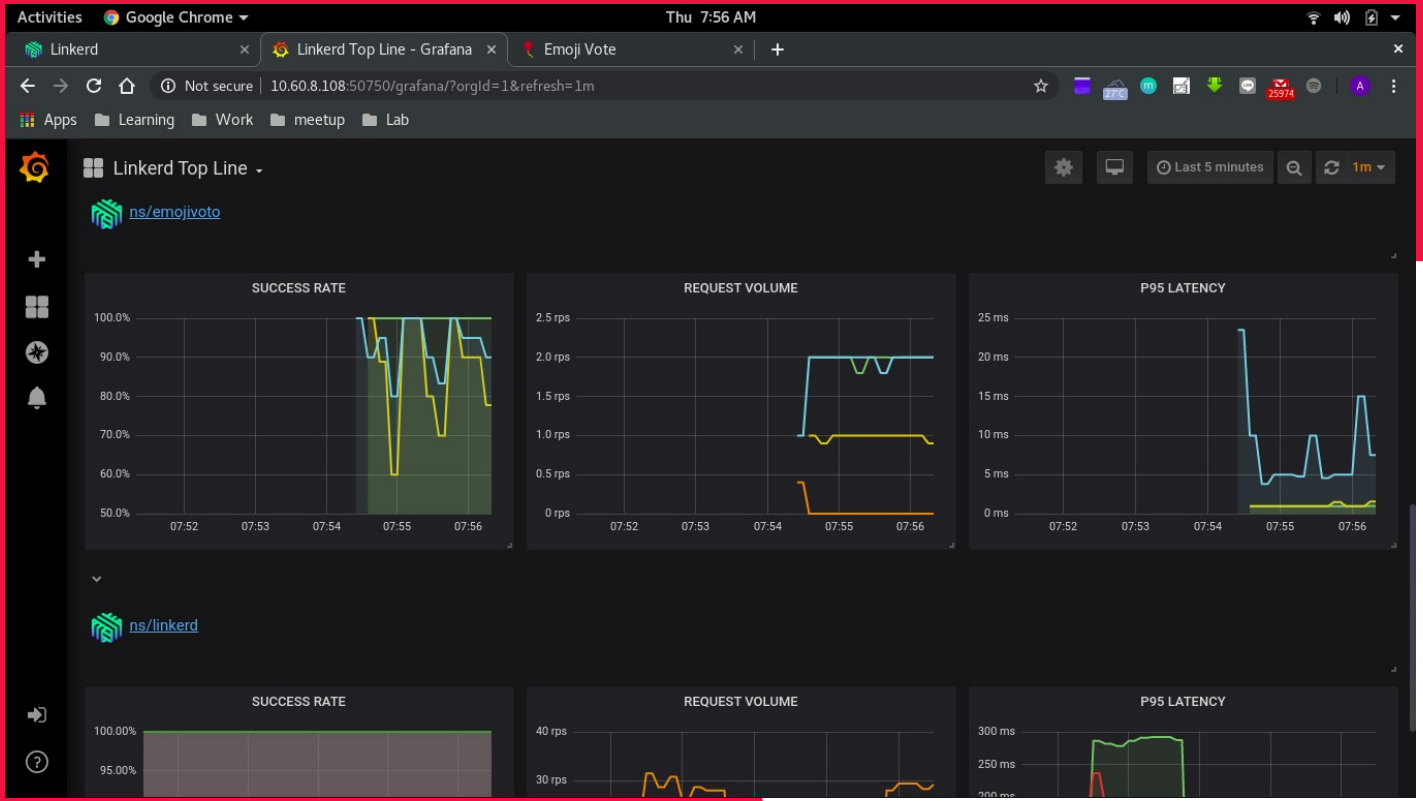
Test Deploy emoji-vote Apps



Test Deploy emoji-vote Apps



Test Deploy emoji-vote Apps



Thank you!