ADDO
ALL DAY DEVOPS

OCTOBER 28, 2021

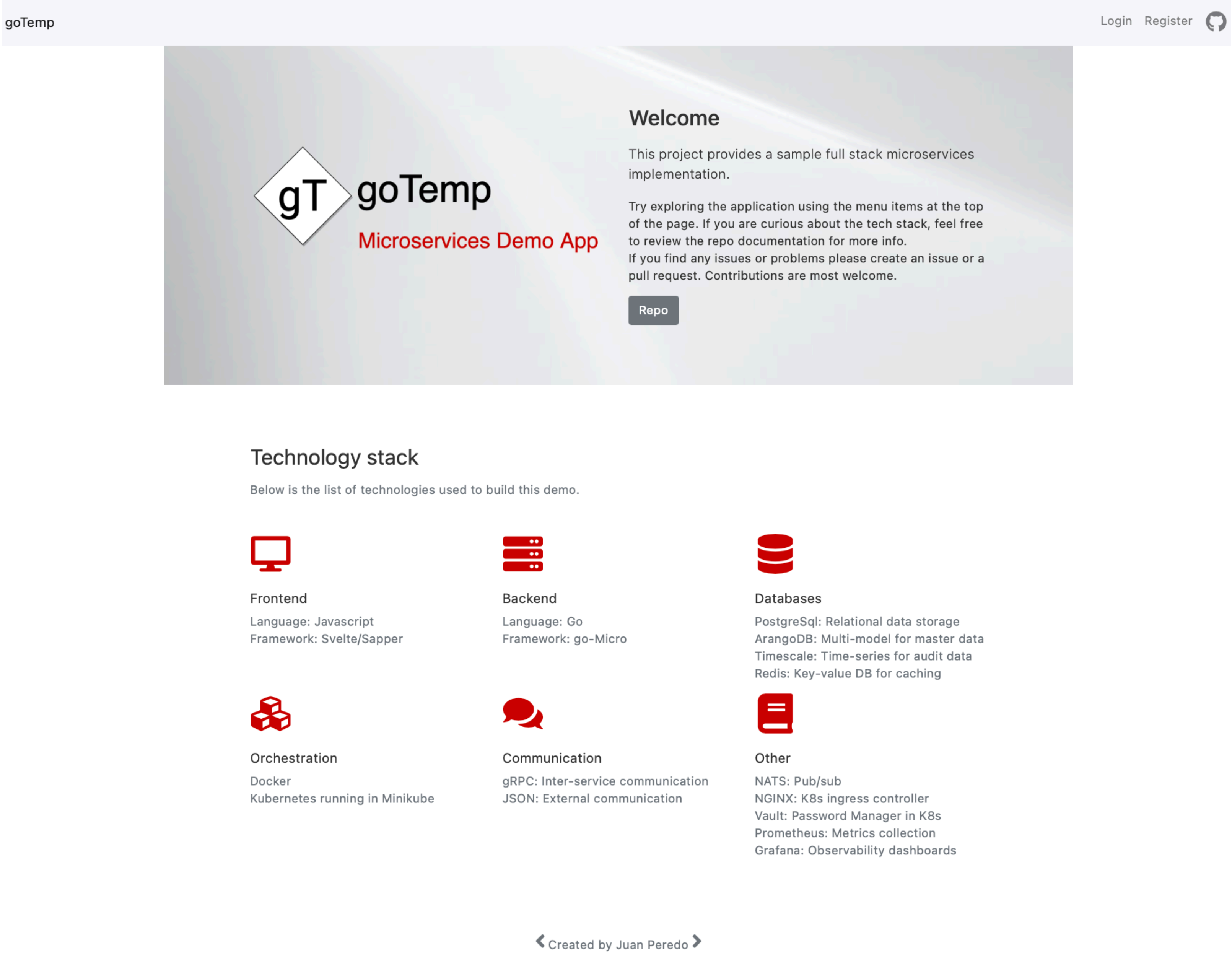
Juan Peredo

Fast-track microservices development with Docker Compose and Minikube

The promise

Microservices simplify application development

Testing the promise



goTemp

Login Register

Welcome

This project provides a sample full stack microservices implementation.

Try exploring the application using the menu items at the top of the page. If you are curious about the tech stack, feel free to review the repo documentation for more info.
If you find any issues or problems please create an issue or a pull request. Contributions are most welcome.

Repo

Technology stack

Below is the list of technologies used to build this demo.

Frontend
Language: Javascript
Framework: Svelte/Sapper

Backend
Language: Go
Framework: go-Micro

Databases
PostgreSql: Relational data storage
ArangoDB: Multi-model for master data
Timescale: Time-series for audit data
Redis: Key-value DB for caching

Orchestration
Docker
Kubernetes running in Minikube

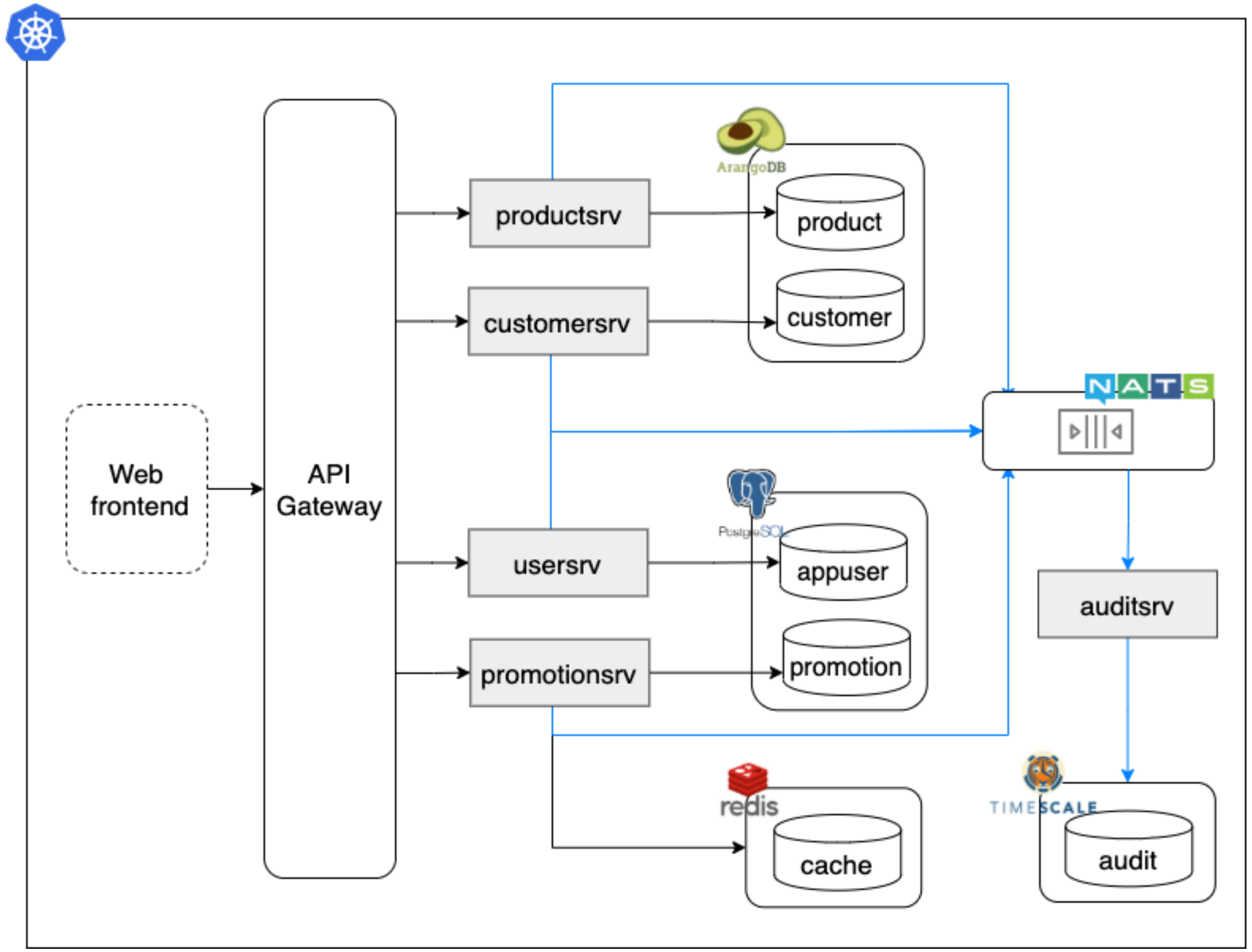
Communication
gRPC: Inter-service communication
JSON: External communication

Other
NATS: Pub/sub
NGINX: K8s ingress controller
Vault: Password Manager in K8s
Prometheus: Metrics collection
Grafana: Observability dashboards

Created by Juan Peredo

- Built small microservices application
- 5 services
- Multiple DBs
- Orchestrated with K8s

Reality

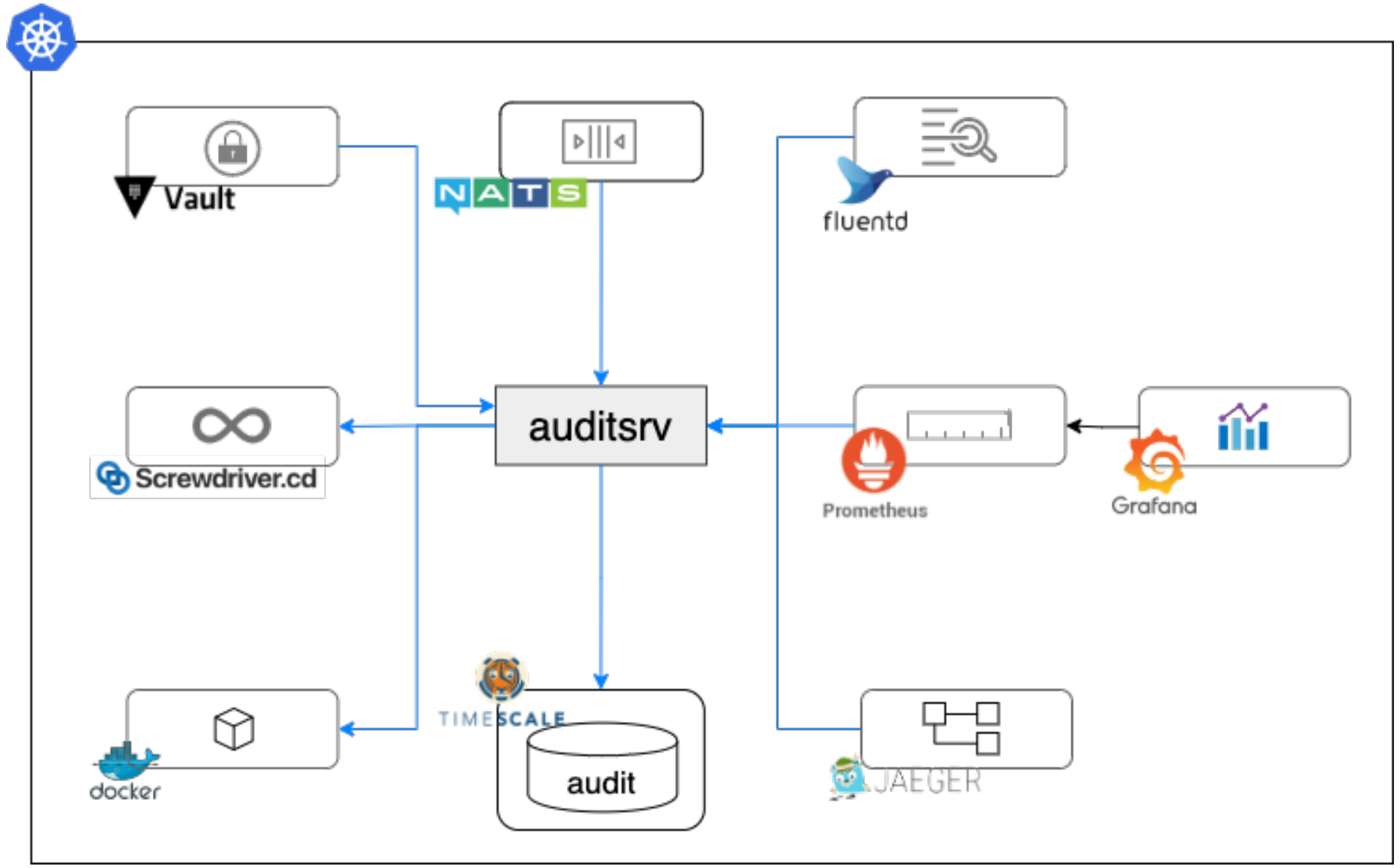


Complexity moves to the edges

Development Challenges:

- Many moving parts
- Debugging and tracing
- Orchestration
- Slow development cycle

One more thing....



- Each service requires an ecosystem to be effective and functional

Quest to simplify the process

Explore:

- Faster onboarding
- Manage dependencies
- Enhance microservices development
- Orchestrate application
- Faster feedback cycle

Your guide on this journey

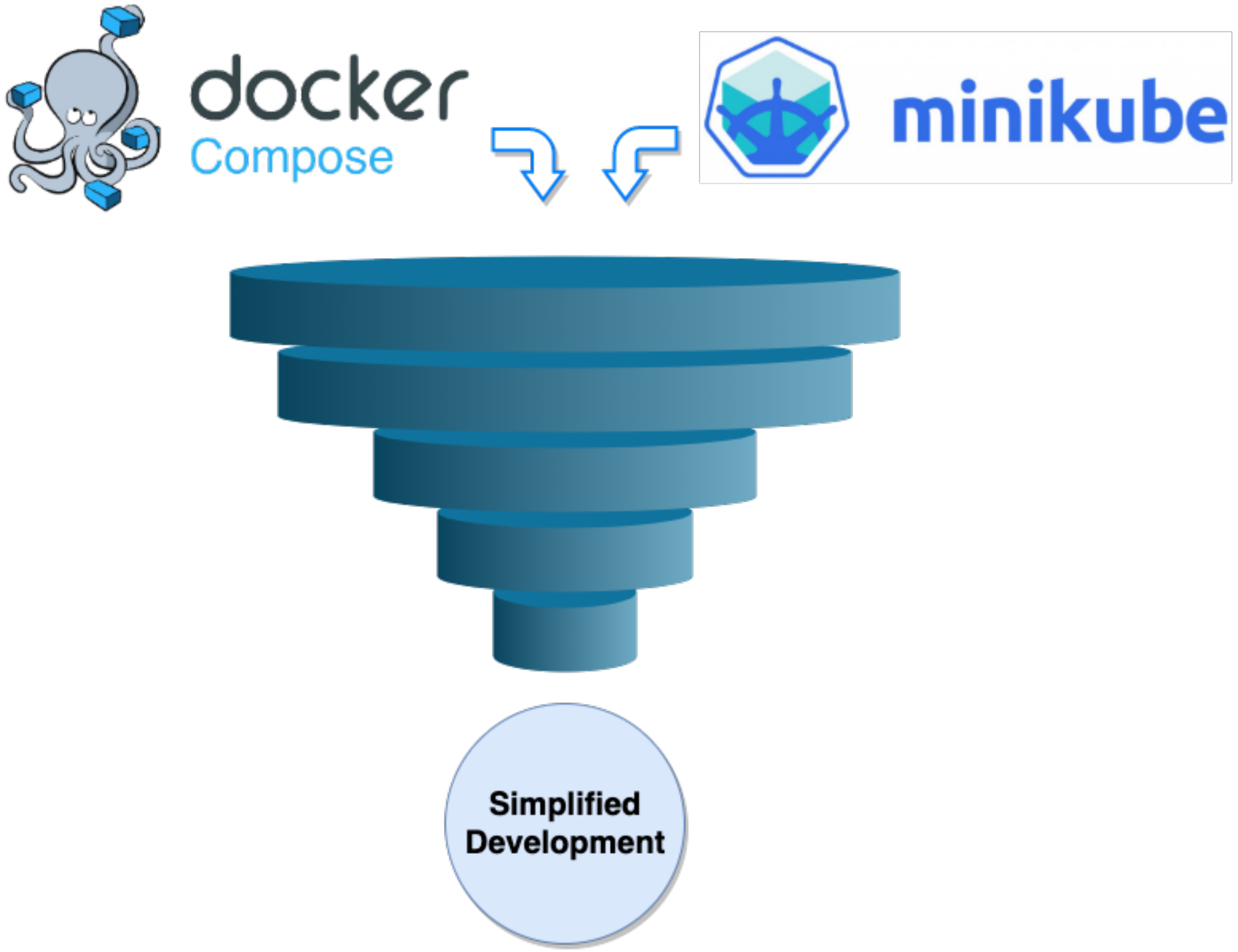
Cloud consultant / architect / developer and everything in between.

Linkedin: http://linkedin.com/in/juanperedotech

Twitter: @JuanPeredoTech

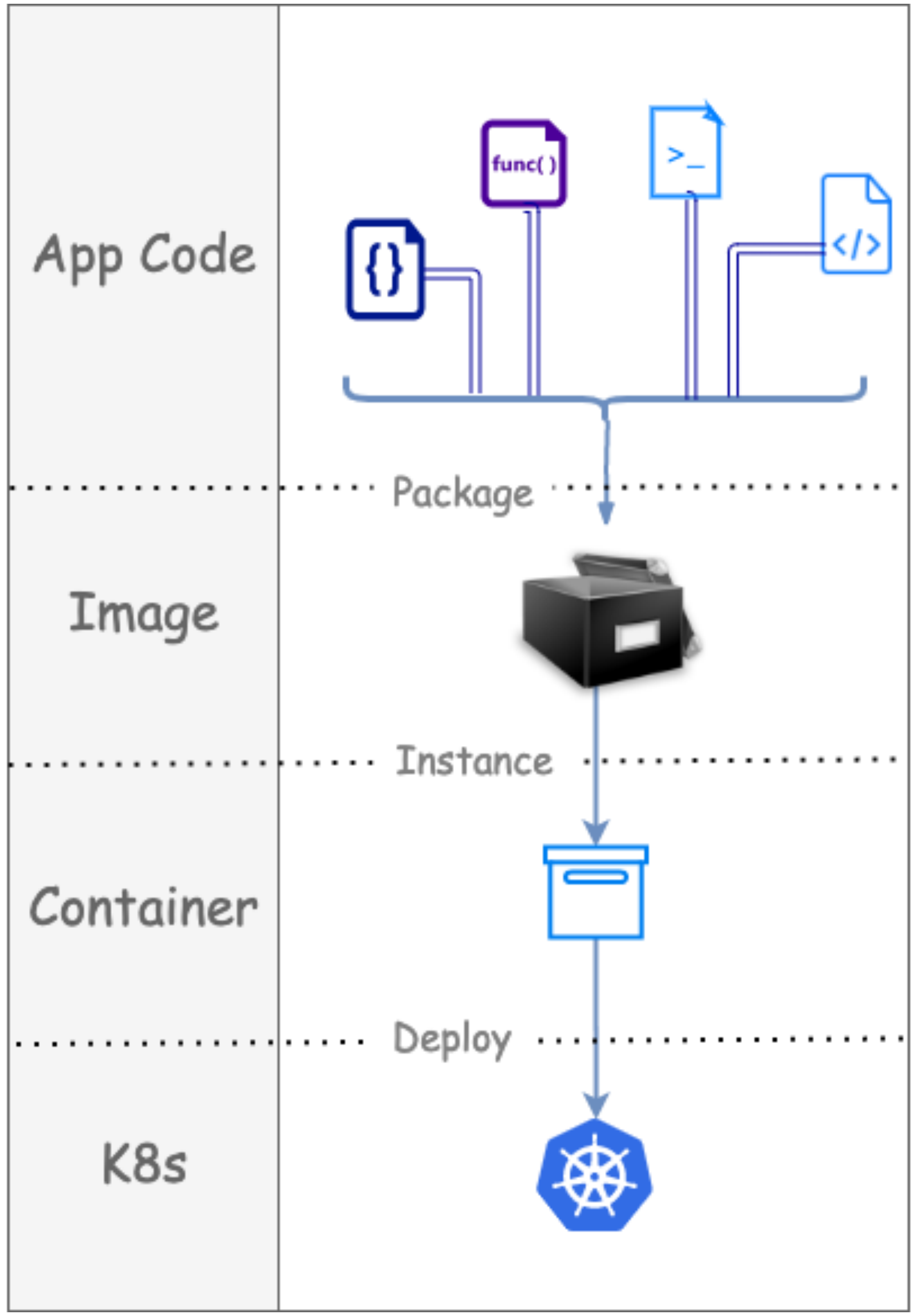
Github: https://github.com/camba1

Tools to wrangle our services



- Docker Compose:
  - Help us develop and manage services
- Minikube:
  - Simplify orchestration and speed feedback loop

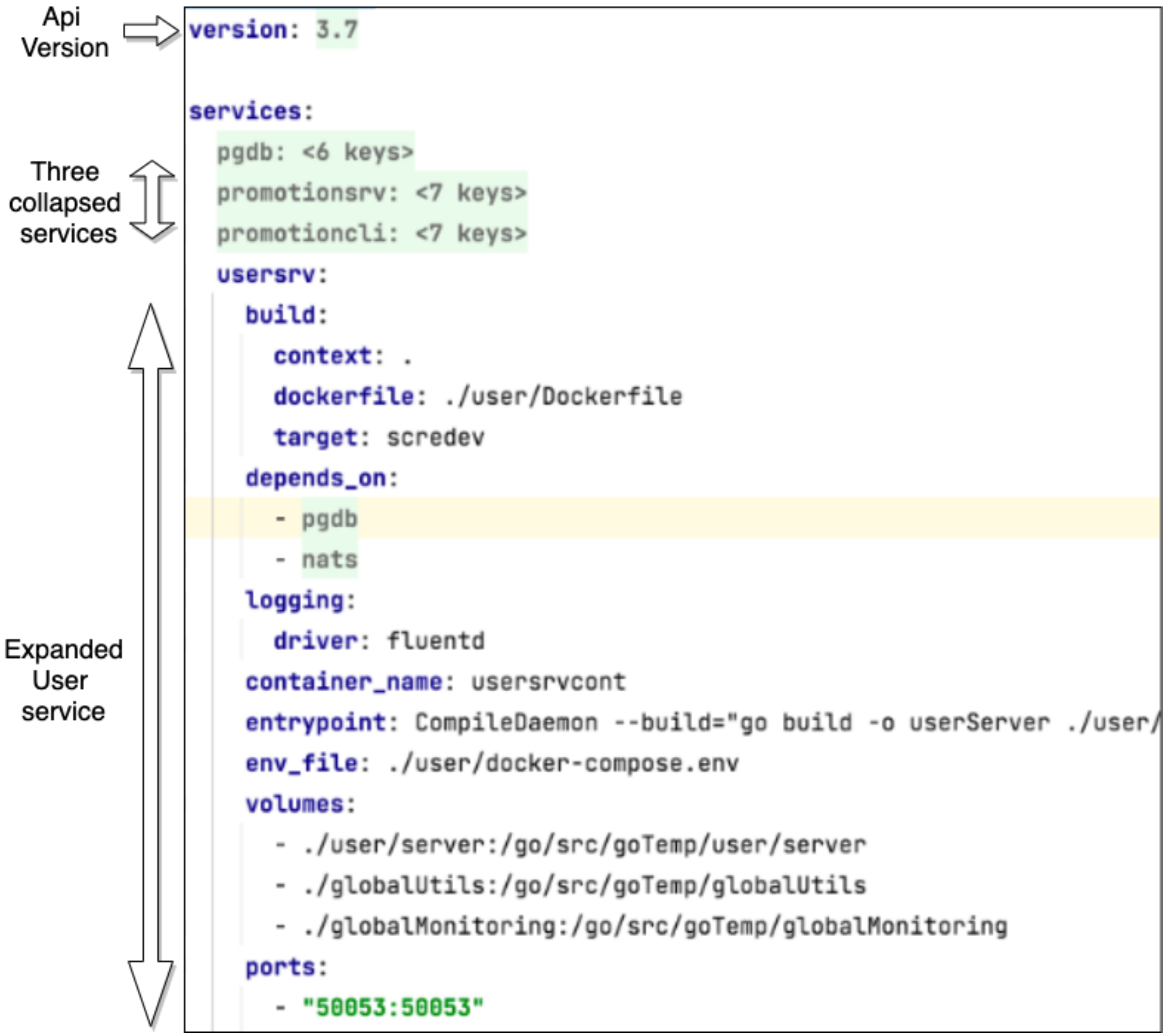
Images and containers



- Images package apps and all their dependencies
- Images can be shared using container registries like Docker Hub
- Containers are ephemeral image instances
- Kubernetes manages container lifecycle automatically

Docker Compose

- Run multiple container in services
- Each service defines how to run a container

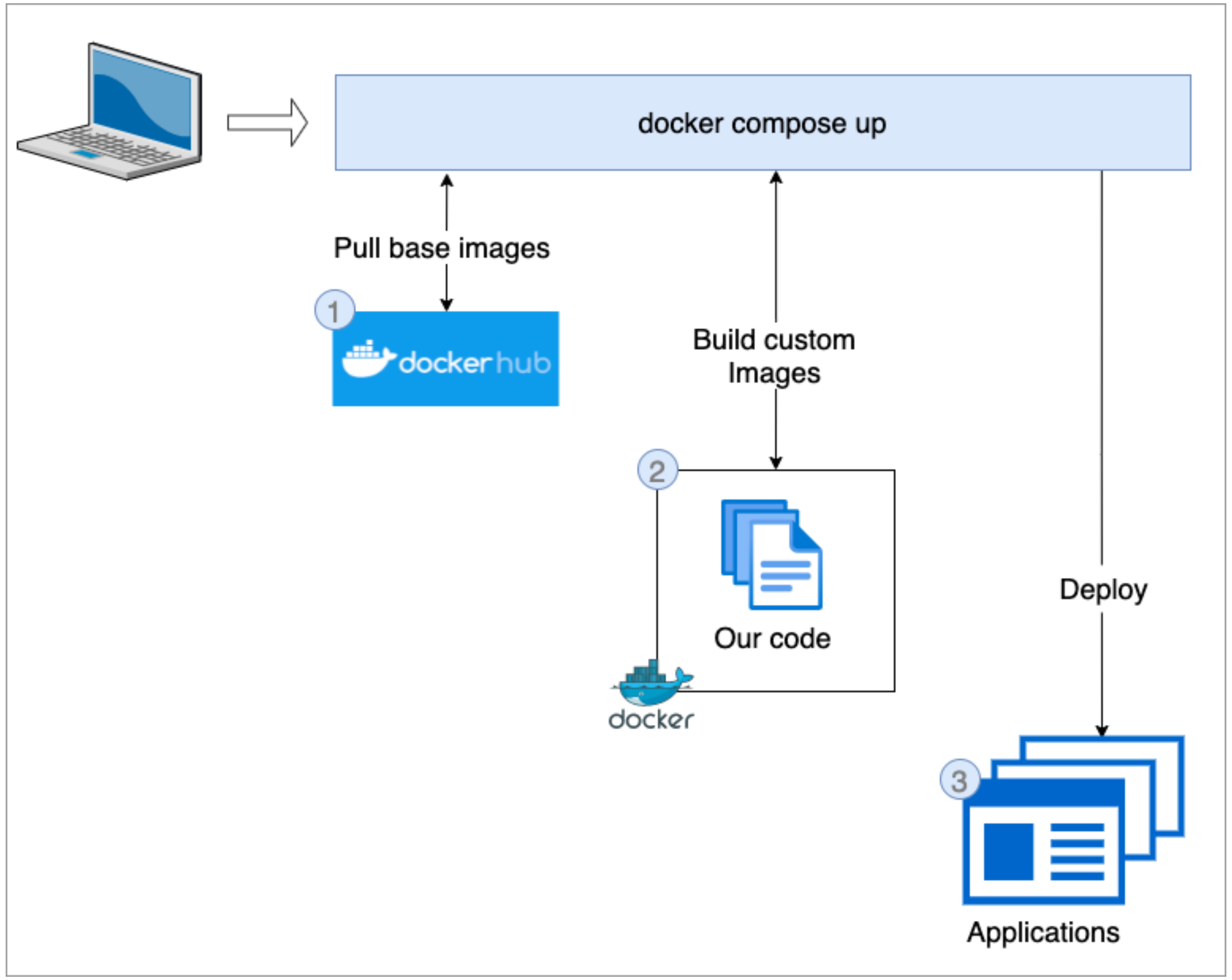


```
version: 3.7

services:
  pgdb: <6 keys>
  promotionsrv: <7 keys>
  promotioncli: <7 keys>
  usersrv:
    build:
      context: .
      dockerfile: ./user/Dockerfile
      target: scredev
    depends_on:
      - pgdb
      - nats
    logging:
      driver: fluentd
    container_name: usersrvcont
    entrypoint: CompileDaemon --build="go build -o userServer ./user/
    env_file: ./user/docker-compose.env
    volumes:
      - ./user/server:/go/src/goTemp/user/server
      - ./globalUtils:/go/src/goTemp/globalUtils
      - ./globalMonitoring:/go/src/goTemp/globalMonitoring
    ports:
      - "50053:50053"
```

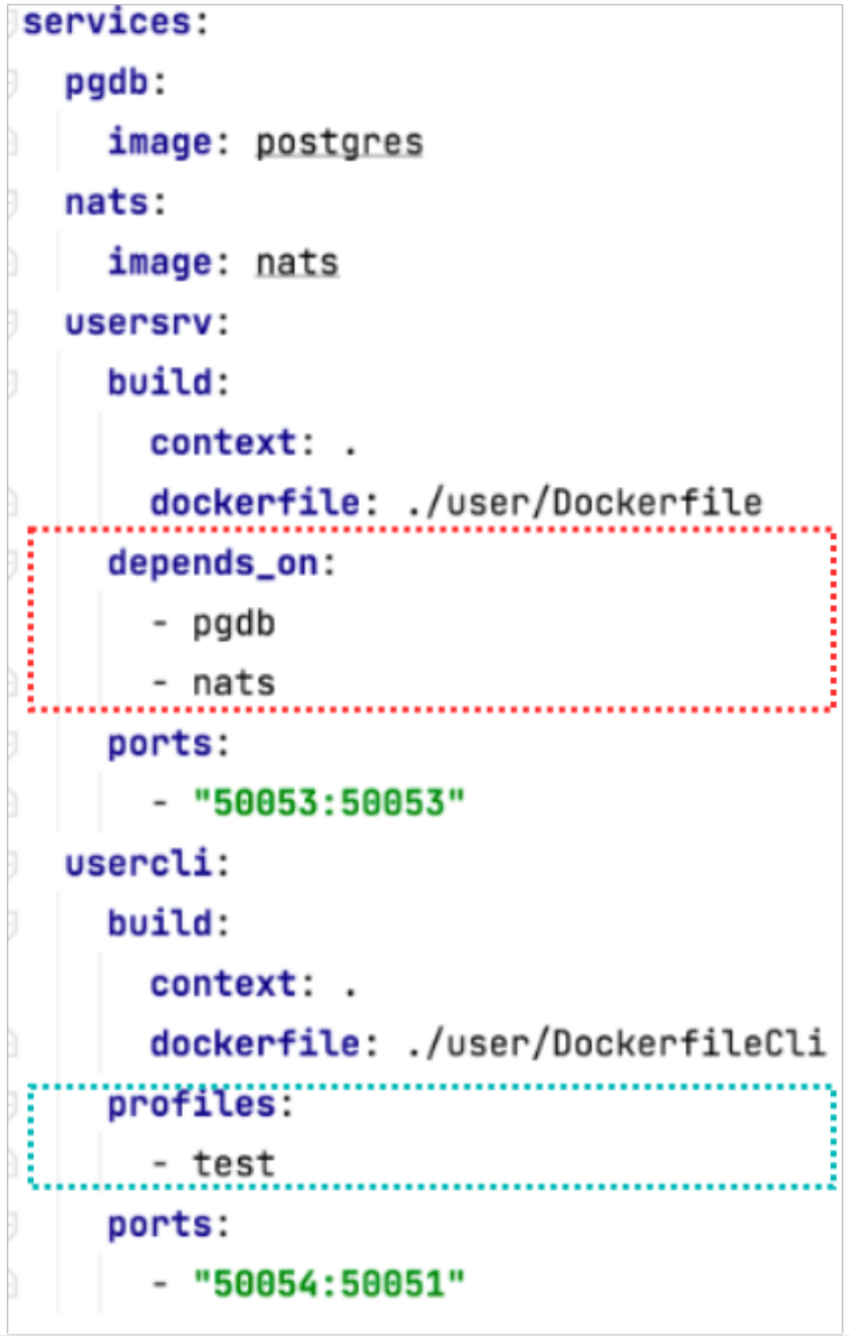
Faster start up & onboarding

- `"Docker compose up":`
  - Pulls & builds images
  - Starts application
- Much faster than manual installs
- Ensures app runs correct dependency versions



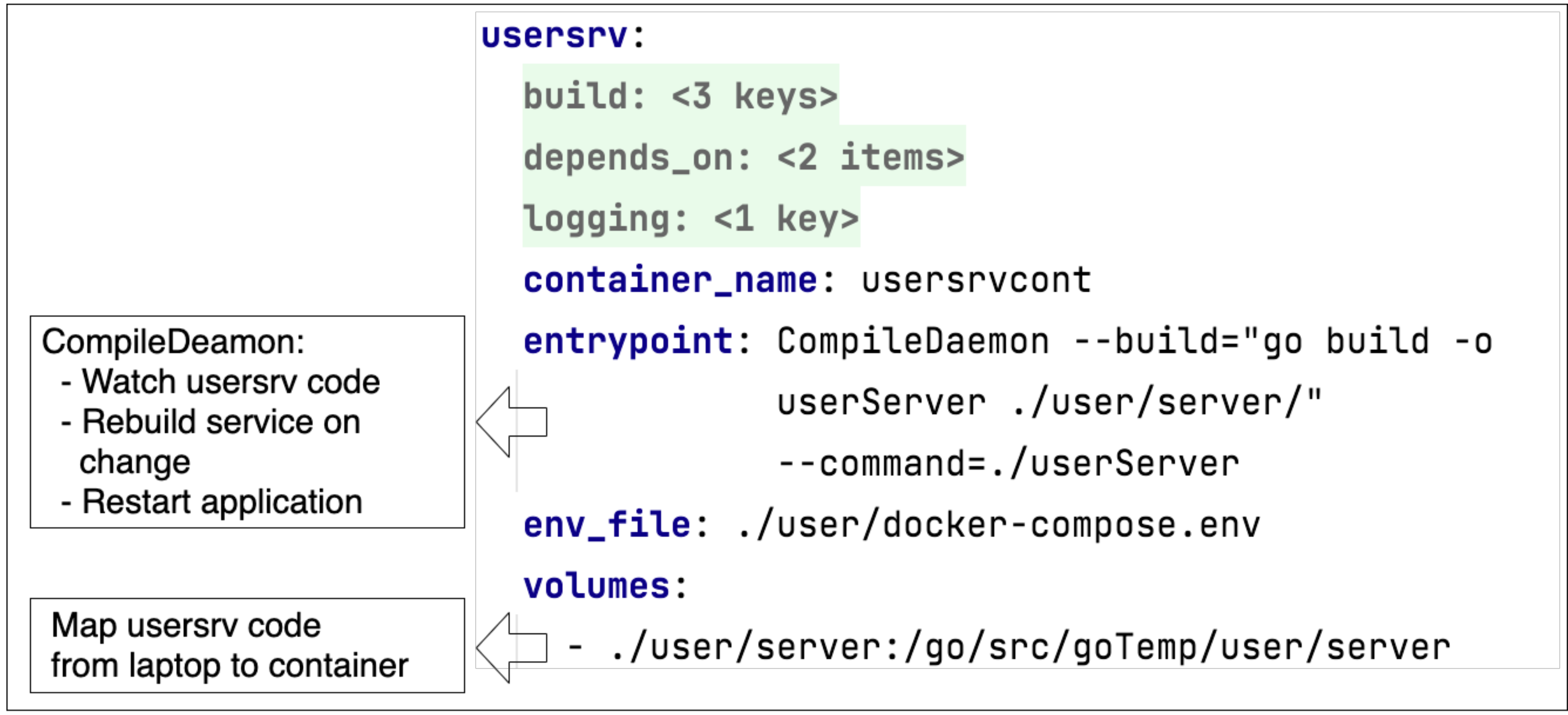
Manage service dependencies

- "depends_on":
  - Start service & dependencies
- "profiles":
  - Start selected services

<table>
<tr><th>Sample docker compose file (partial)</th><th>Sample commands</th></tr>
<tr><td rowspan="2">
<pre>
services:
  pgdb:
    image: postgres
  nats:
    image: nats
  usersrv:
    build:
      context: .
      dockerfile: ./user/Dockerfile
    depends_on:
      - pgdb
      - nats
    ports:
      - "50053:50053"
  usercli:
    build:
      context: .
      dockerfile: ./user/DockerfileCli
    profiles:
      - test
    ports:
      - "50054:50051"
</pre></td><td>docker compose up usersrv
Starts:
- user service
- nats
- postgreSql</td></tr>
<tr><td>docker compose --profile test up
Starts:
- user service test client
- Other test clients (not shown)</td></tr>
</table>

Debugging & hot reloads

- Entry point override:
  - Setup hot rebuilds
- IDE container debugging:
  - VS code Plug-in
  - PyCharm
  - IntelliJ



CompileDeamon:
- Watch usersrv code
- Rebuild service on change
- Restart application

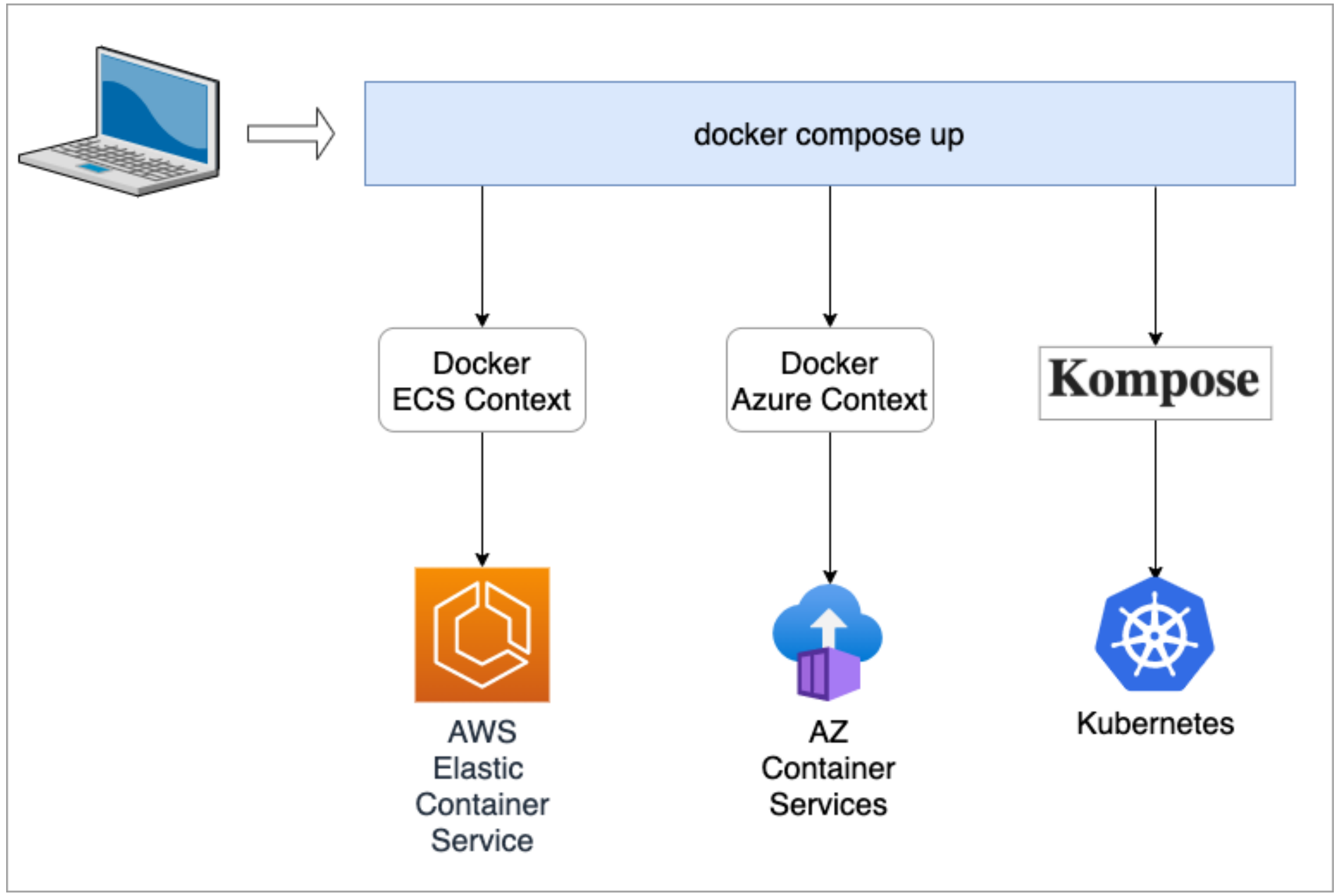
Map usersrv code from laptop to container

```
usersrv:
  build: <3 keys>
  depends_on: <2 items>
  logging: <1 key>
  container_name: usersrvcont
  entrypoint: CompileDaemon --build="go build -o
              userServer ./user/server/"
              --command=./userServer
  env_file: ./user/docker-compose.env
  volumes:
    - ./user/server:/go/src/goTemp/user/server
```

Note: Entry point formatted to fit slide

Quick migration to Orchestration Engines

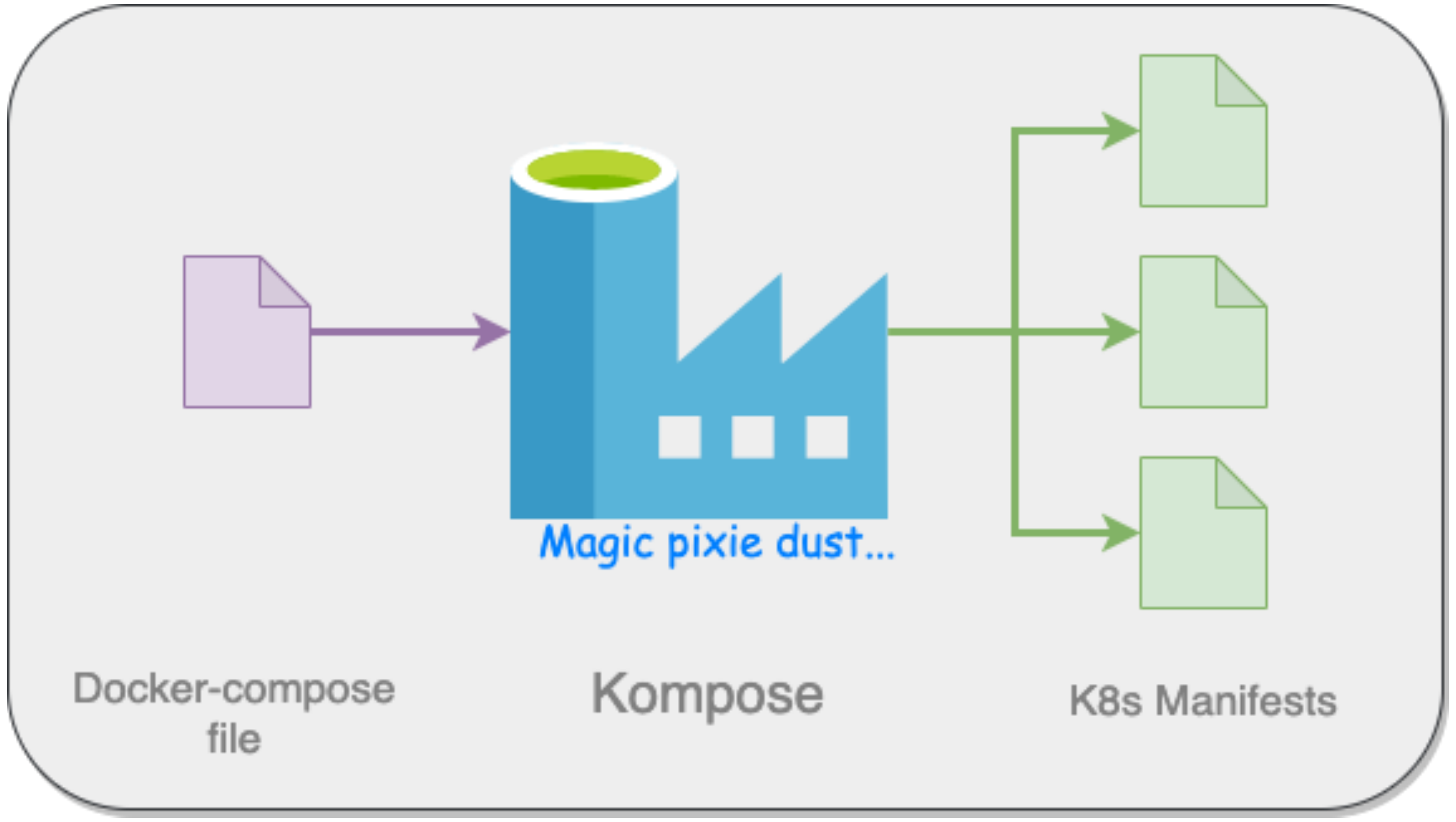
- Deploy locally, AWS or AZ
  - AWS Context
  - Azure Context
- Migrating to K8s
  - Kompose



Kompose

Kompose generates K8s manifests automagically! 😎

- Based on docker compose file
- `kompose -f <pathToComposeFile> convert`
- Deploy output to any K8s cluster

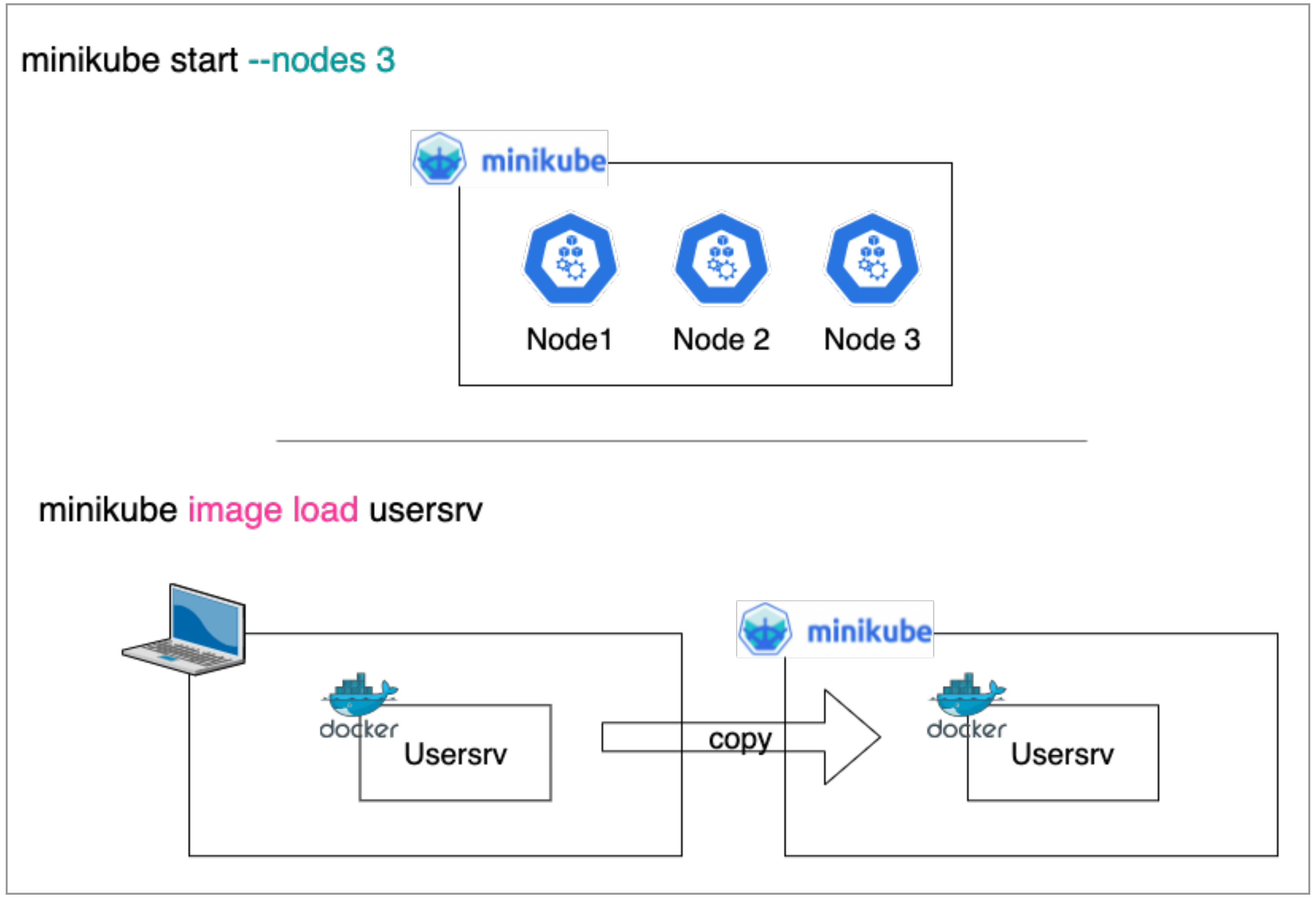


Running application in K8s with Minikube

- Runs official K8s locally
- Supports:
  - latest version
  - 6 previous minor versions
- Fast K8s cluster creation

Fast deployments to K8s using Minikube

- Multi-node clusters
- Use locally stored containers



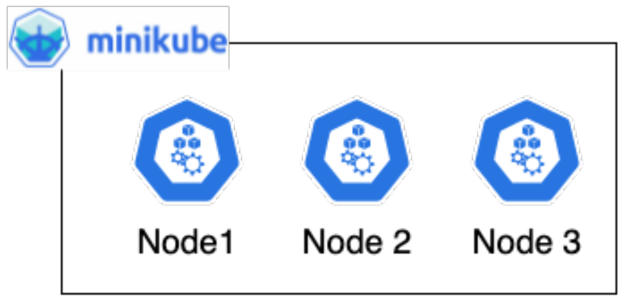
```
minikube start --nodes 3
```

```
minikube image load usersrv
```

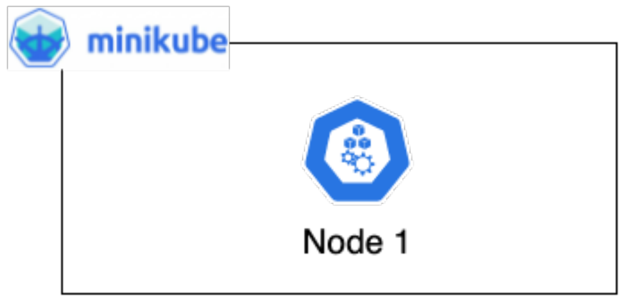
Cluster management

- Profiles
- Dashboard
- Addons

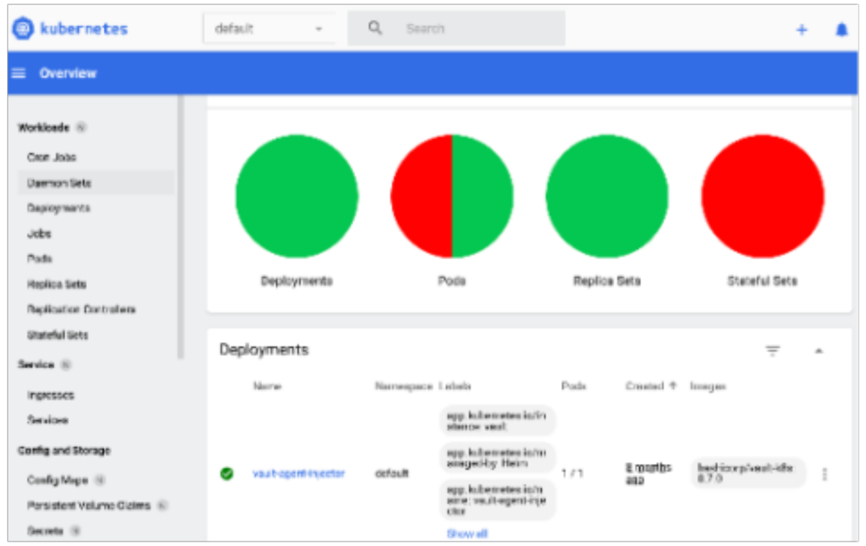
minikube start --nodes 3 --profile test



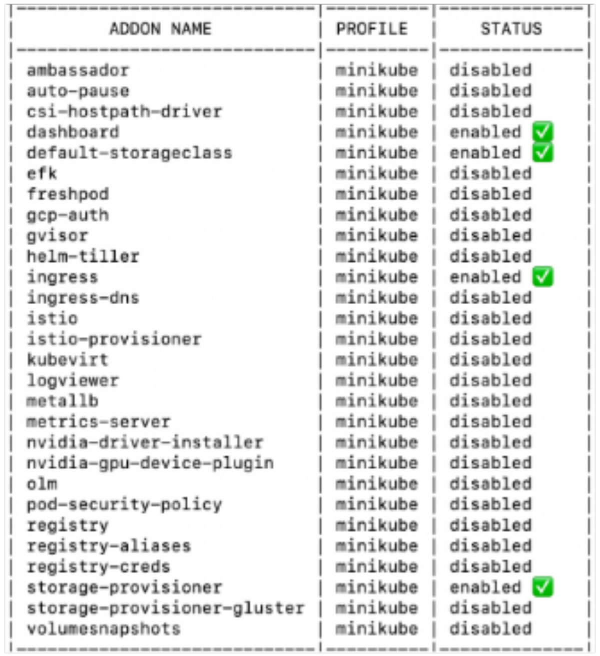
minikube start --profile dev



minikube dashboard



minikube addons list



```
|-----------------------------|----------|--------------|
|         ADDON NAME          | PROFILE  |    STATUS    |
|-----------------------------|----------|--------------|
| ambassador                  | minikube | disabled     |
| auto-pause                  | minikube | disabled     |
| csi-hostpath-driver         | minikube | disabled     |
| dashboard                   | minikube | enabled ✅   |
| default-storageclass        | minikube | enabled ✅   |
| efk                         | minikube | disabled     |
| freshpod                    | minikube | disabled     |
| gcp-auth                    | minikube | disabled     |
| gvisor                      | minikube | disabled     |
| helm-tiller                 | minikube | disabled     |
| ingress                     | minikube | enabled ✅   |
| ingress-dns                 | minikube | disabled     |
| istio                       | minikube | disabled     |
| istio-provisioner           | minikube | disabled     |
| kubevirt                    | minikube | disabled     |
| logviewer                   | minikube | disabled     |
| metallb                     | minikube | disabled     |
| metrics-server              | minikube | disabled     |
| nvidia-driver-installer     | minikube | disabled     |
| nvidia-gpu-device-plugin    | minikube | disabled     |
| olm                         | minikube | disabled     |
| pod-security-policy         | minikube | disabled     |
| registry                    | minikube | disabled     |
| registry-aliases            | minikube | disabled     |
| registry-creds              | minikube | disabled     |
| storage-provisioner         | minikube | enabled ✅   |
| storage-provisioner-gluster | minikube | disabled     |
| volumesnapshots             | minikube | disabled     |
|-----------------------------|----------|--------------|
```

Bottom line

Micro services simplify application development

Bottom line

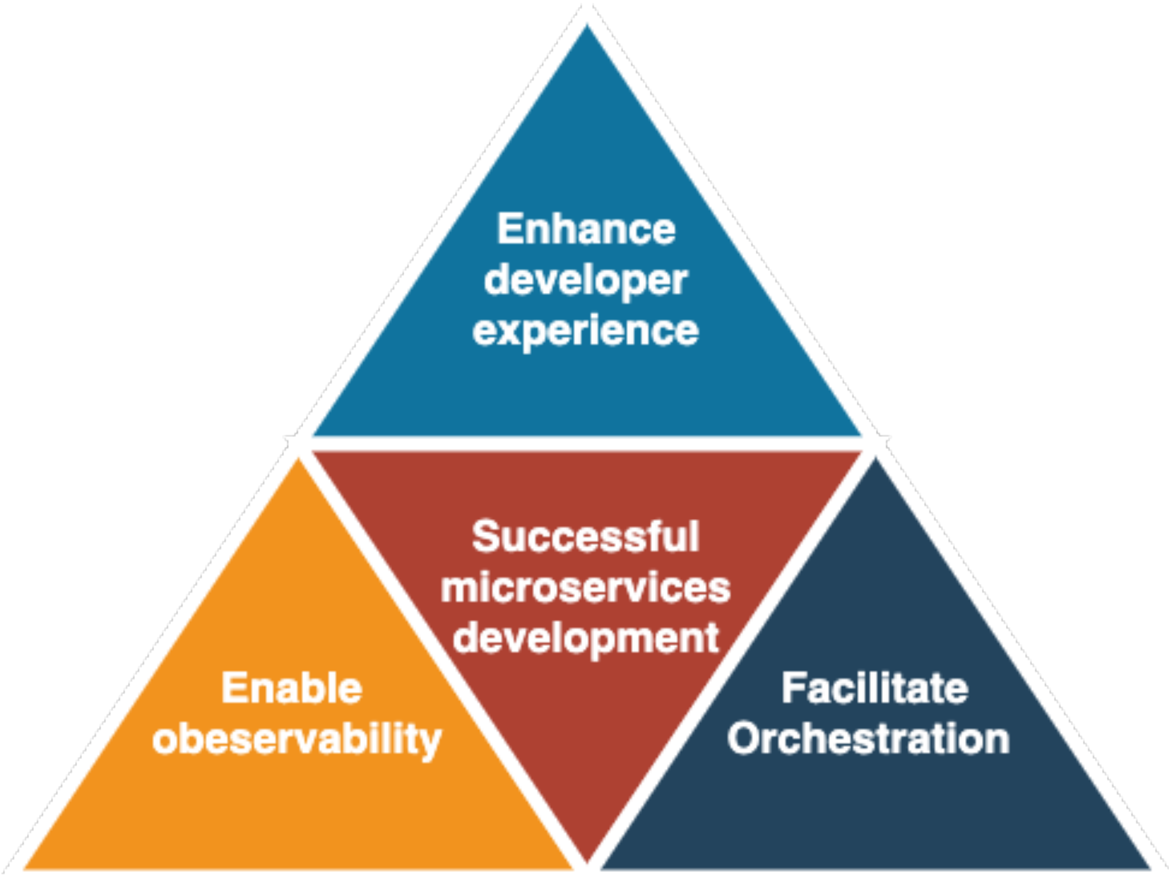
Micro services simplify application development

If you have the right tools

Bottom line

Micro services simplify application development

If you have the right tools

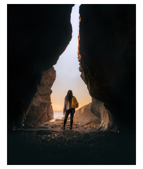

Enhance developer experience
Successful microservices development
Enable obeservability
Facilitate Orchestration

Questions?

Thanks to the ADDO organizers & all conference sponsors

Appendix

Photos

-  Photo by Ola Dybul on Unsplash
- Photo by Ben White on Unsplash