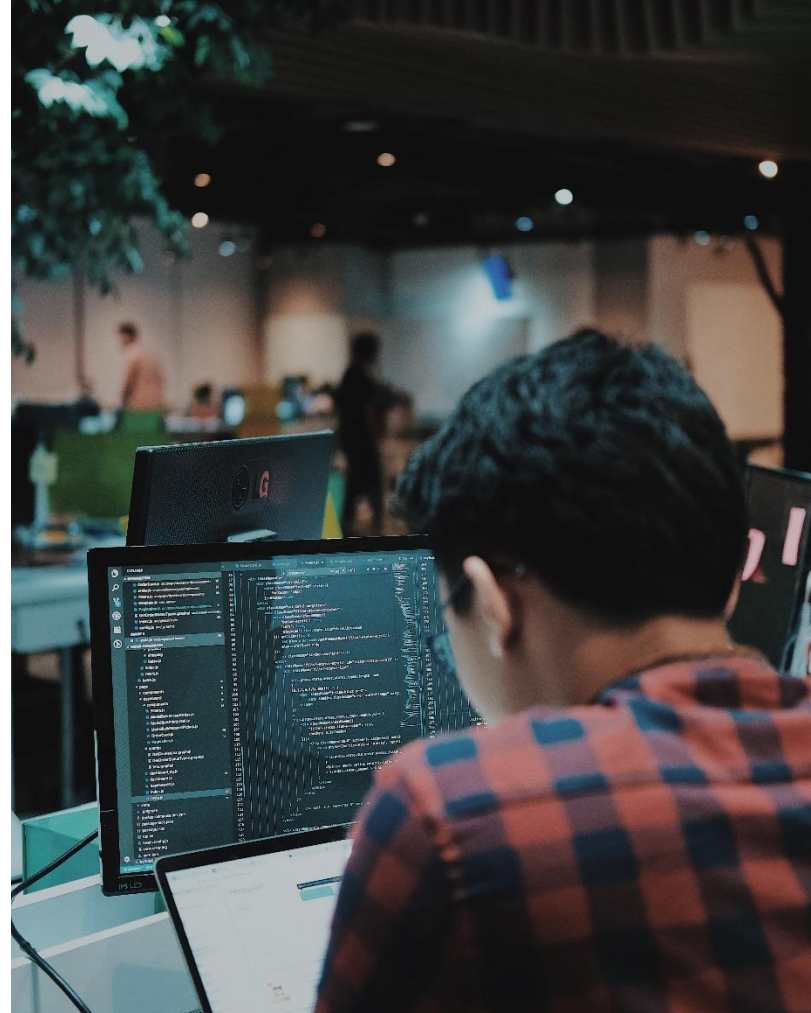# Deployment and configuration

SIARHEI SVILA

Senior Software Engineer

- 4+ years commercial experience in software development and design:
  - Java web development stack (Spring, Hibernate, Docker, Kubernetes)
  - Design of distributed systems with Messaging communication
  - Cloud stack (AWS, Azure, GCP)
  - Epam internal projects and mentoring programs contributor

Kubernetes for Software Engineers

1	**WORKSHOP: KUBERNETES OVERVIEW**

2	**WORKSHOP: KUBERNETES CONFIGURATION**

3	**WORKSHOP: HELM OVERVIEW**

4	**WORKSHOP: NETWORKING**

Agenda

1	**LIVENESS & READINESS PROBES**

2	**SECRETS AND CONFIG MAPS**

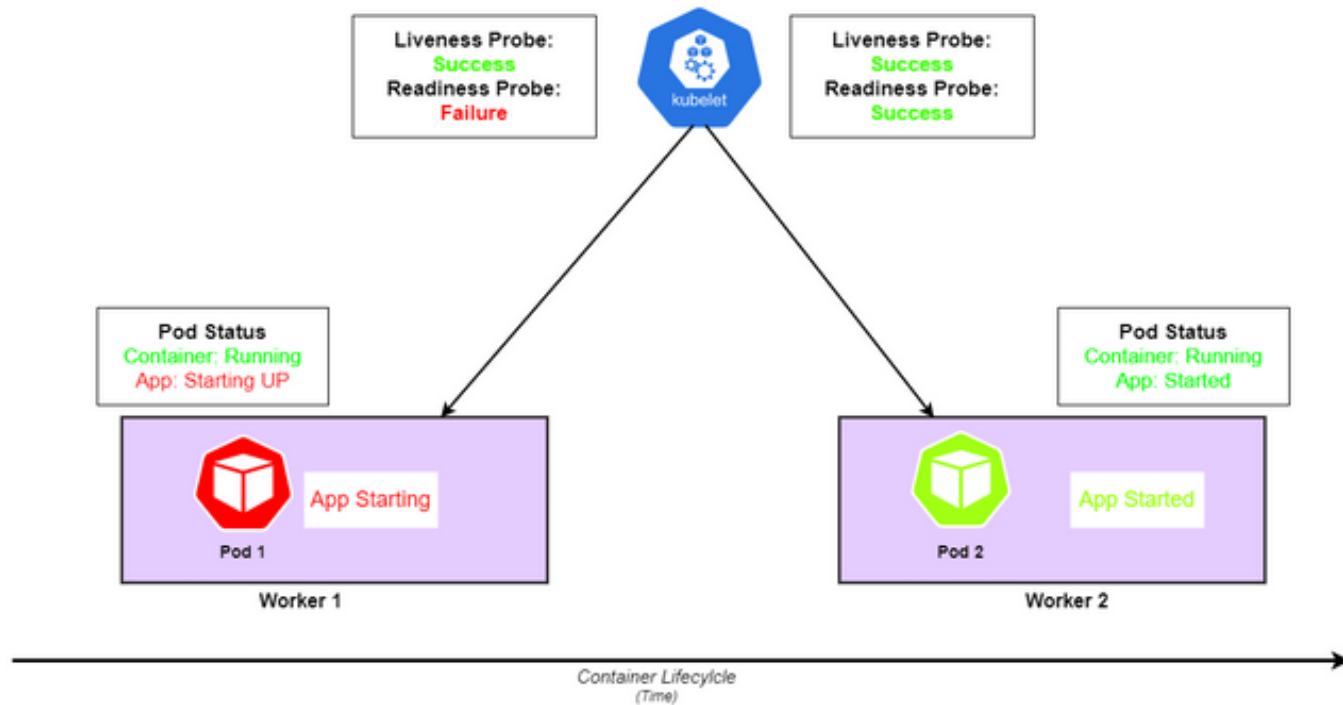| 3	**DEPLOYMENT STRATEGIES**

LIVENESS AND READINESS PROBES

Health checks
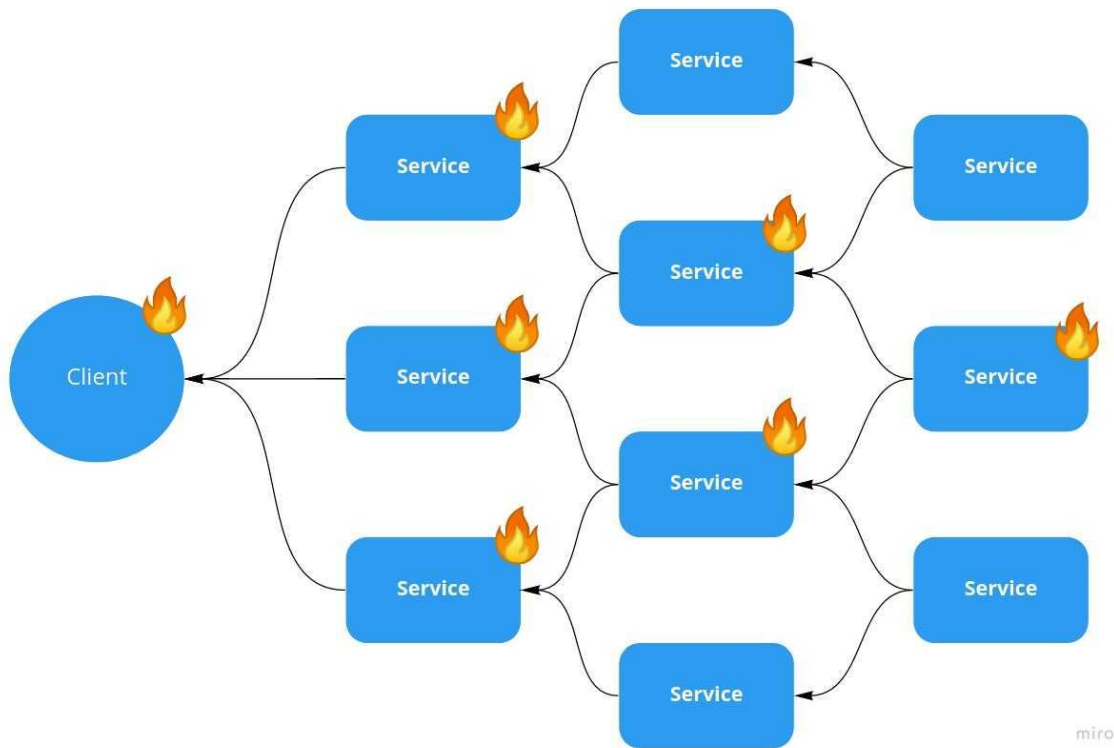
- Liveness:
  - Due to some error, memory leak, cpu usage, etc **application is not responding and stuck in error state**

- Readiness
  - **Application is available, but not able to serve traffic**

Readiness & Liveness probes

CONFIG-MAPS AND SECRETS

Config maps & secrets

```
apiVersion: v1
kind: Pod
metadata:
  name: dependent-envars-demo
spec:
  containers:
    - name: dependent-envars-demo
      args:
        - while true; do echo -en '\n'; printf UNCHANGED_REFERENCE=$UNCHANGE
      command:
        - sh
        - -c
      image: busybox:1.28
      env:
        - name: SERVICE_PORT
          value: "80"
        - name: SERVICE_IP
          value: "172.17.0.1"
```

```
apiVersion: v1
kind: Pod
metadata:
  name: dapi-test-pod
spec:
  containers:
    - name: test-container
      image: k8s.gcr.io/busybox
      command: [ "/bin/sh", "-c", "env" ]
      envFrom:
      - configMapRef:
          name: special-config
  restartPolicy: Never
```
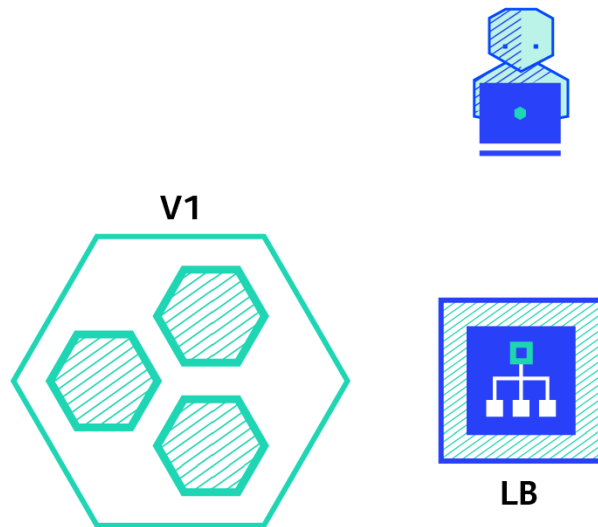
Config maps & secrets
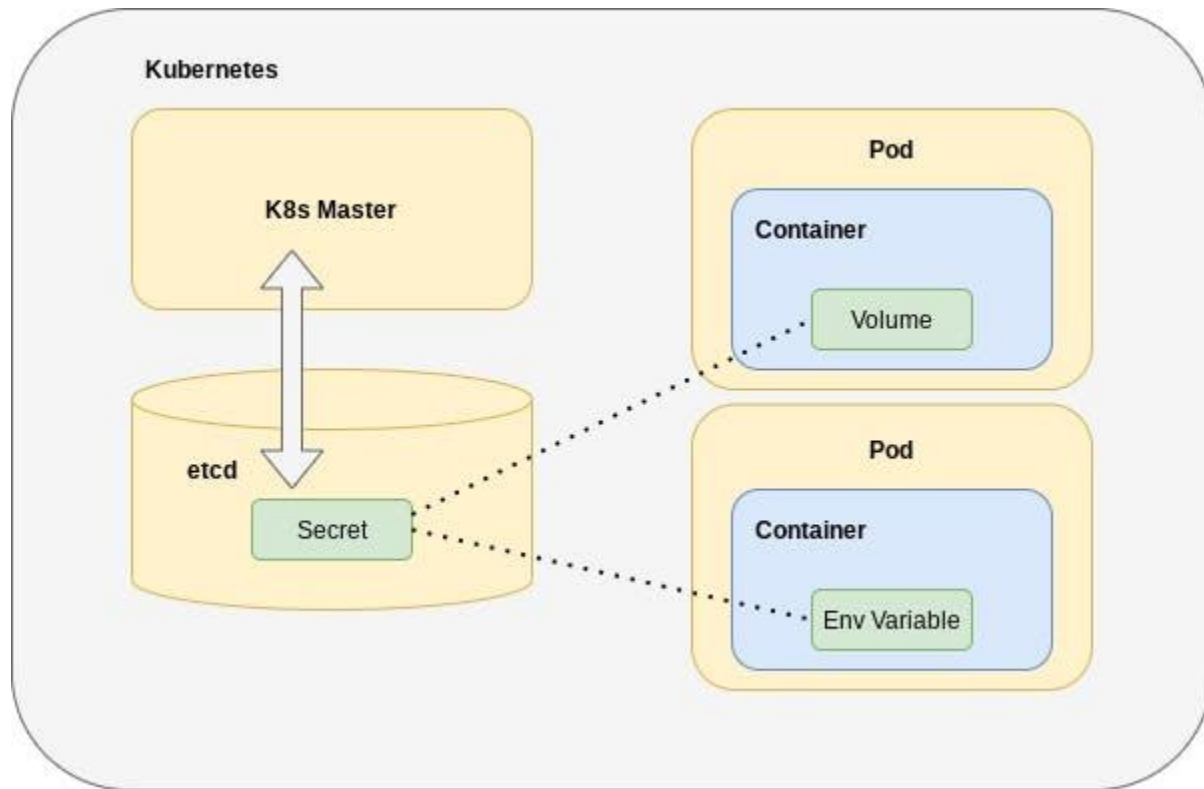
Secret types

- **Opaque Secrets** – The default Secret type if omitted from a Secret configuration file.

- **Service account token Secrets** – Used to store a token that identifies a service account. When using this Secret type, you need to ensure that the `kubernetes.io/service-account.name` annotation is set to an existing service account name.

- **Docker config Secrets** – Stores the credentials for accessing a Docker registry for images.

- **Basic authentication Secret** – Used for storing credentials needed for basic authentication. When using this Secret type, the `data` field of the Secret must contain the `username` and `password` keys

- **SSH authentication secrets** – Used for storing data used in SSH authentication. When using this Secret type, you will have to specify a `ssh-privatekey` key-value pair in the `data` (or `stringData`) field as the SSH credential to use.

- **TLS secrets** – For storing a certificate and its associated key that are typically used for TLS . This data is primarily used with TLS termination of the Ingress resource, but may be used with other resources or directly by a workload. When using this type of Secret, the `tls.key` and the `tls.crt` key must be provided in the data (or `stringData`) field of the Secret configuration

- **Bootstrap token Secrets** – Used for tokens used during the node bootstrap process. It stores tokens used to sign well known ConfigMaps.
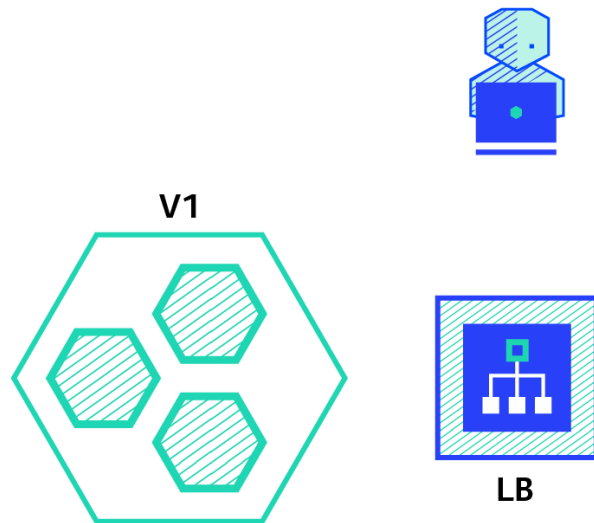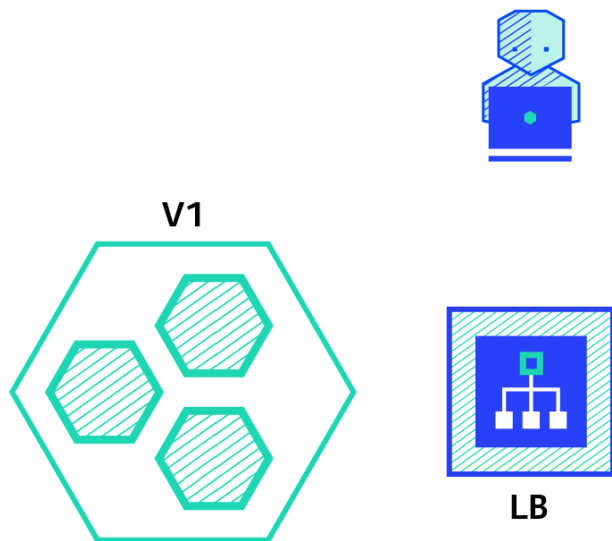
DEPLOYMENT STRATEGIES

Recreate

V1

LB
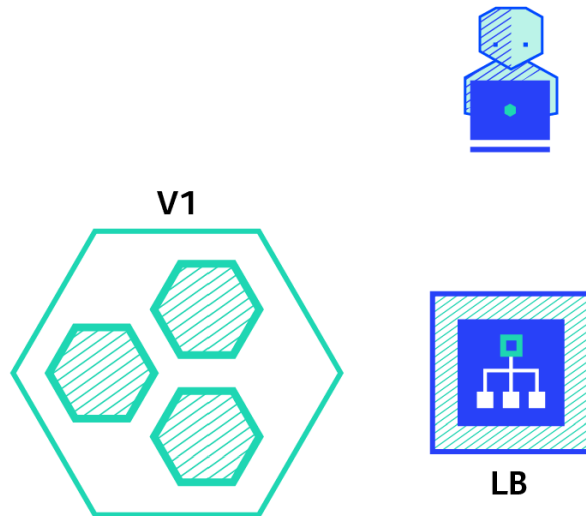
Polling-update (Ramped)

V1

LB

Blue/Green deployment

V1

LB

Canary deployment

V1

LB

Q&A