





NETWORKING

INGRES

POD

**SERVICE**



**LB**





```
1     apiVersion: v1
2     kind: Service
3     metadata:
4         name: kuard-service
5     spec:
6         selector:
7             app: kuard
8         ports:
9             - port: 80
10             targetPort: 8080
```

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: kuard-deployment
5    labels:
6      app: kuard
7  spec:
8    replicas: 3
9    selector:
10     matchLabels:
11       app: kuard
12   template:
13     metadata:
14       labels:
15         app: kuard
16     spec:
17       containers:
18         - image: gcr.io/kuar-demo/kuard-amd64:blue
19           name: kuard
20           ports:
21             - containerPort: 8080
22               name: http
```















**CLIENT**



**INGRESS**



```
1  ---
2  apiVersion: networking.k8s.io/v1
3  kind: Ingress
4  metadata:
5    name: tailnet-ingress
6    annotations:
7      ingress.kubernetes.io/rewrite-target: /
8      kubernetes.io/ingress.class: nginx
9      cert-manager.io/cluster-issuer: "letsencrypt-prod"
10 spec:
11   rules:
12     - host: test.cedi.dev
13       http:
14         paths:
15           - path: /
16             pathType: Prefix
17             backend:
18               service:
19                 name: kuard-service
20                 port:
21                   number: 80
22   tls:
23     - secretName: test-ts-secret
24       hosts:
25         - test.cedi.dev
```

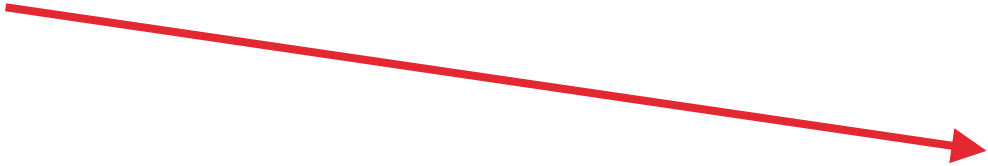










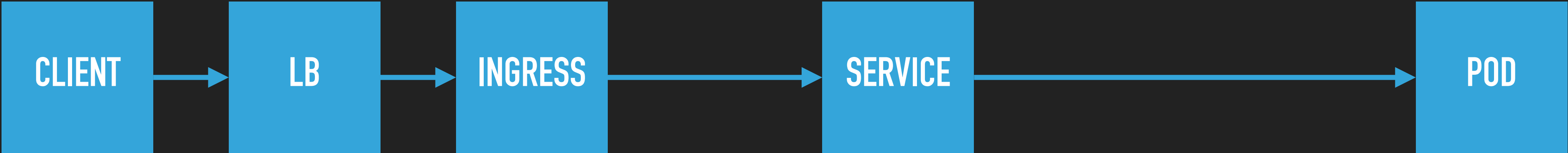


# INGRESS

```
1 ---
2 apiVersion: networking.k8s.io/v1
3 kind: Ingress
4 metadata:
5   name: tailnet-ingress
6   annotations:
7     ingress.kubernetes.io/rewrite-target: /
8     kubernetes.io/ingress.class: nginx
9     cert-manager.io/cluster-issuer: "letsencrypt-prod"
10 spec:
11   rules:
12     - host: test.cedi.dev
13       http:
14         paths:
15           - path: /
16             pathType: Prefix
17             backend:
18               service:
19                 name: kuard-service
20                 port:
21                   number: 80
22   tls:
23     - secretName: test-ts-secret
24       hosts:
25         - test.cedi.dev
```

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: kuard-service
5 spec:
6   selector:
7     app: kuard
8   ports:
9     - port: 80
10      targetPort: 8080
```

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: kuard-deployment
5   labels:
6     app: kuard
7 spec:
8   replicas: 3
9   selector:
10     matchLabels:
11       app: kuard
12   template:
13     metadata:
14       labels:
15         app: kuard
16   spec:
17     containers:
18       - image: gcr.io/kuar-demo/kuard-amd64:blue
19         name: kuard
20         ports:
21           - containerPort: 8080
22             name: http
```



# WRAP-UP

- ▶ Take a Pi, install K3s
- ▶ Deploy your workload with Helm and or Kustomize
- ▶ Secure your secrets with SOPS and don't leak them to Git
- ▶ Keep your Cluster reconciled with ArgoCD
- ▶ Use ingress-nginx to serve incoming requests