
CERN

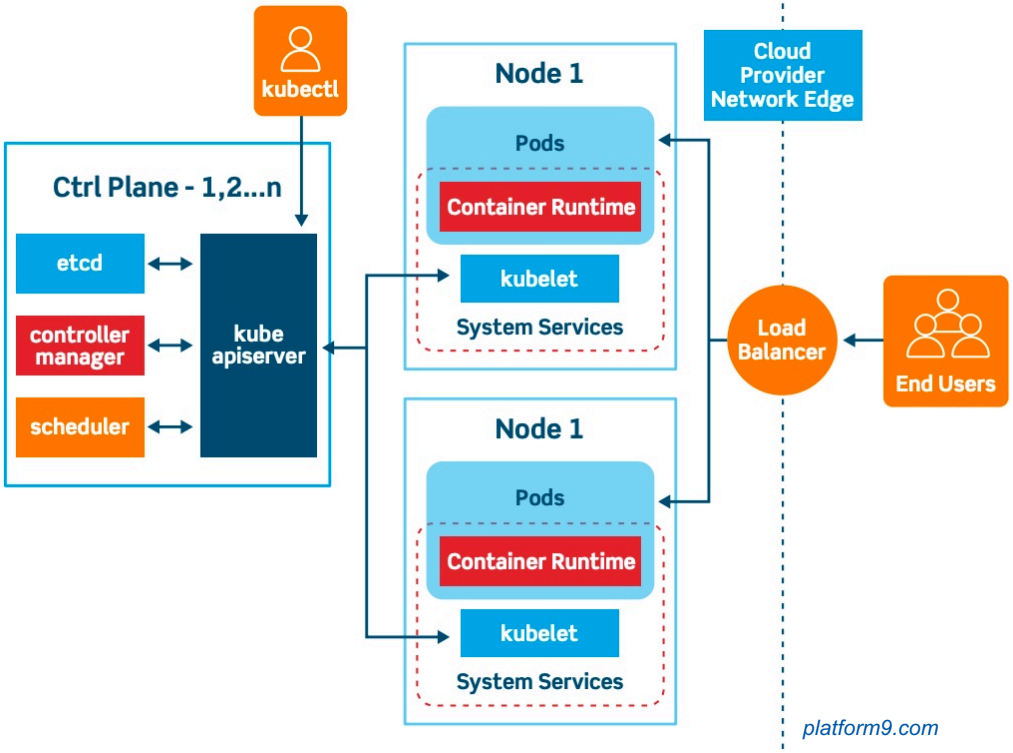
CS Kubernetes Infrastructure

Pablo García Miranda

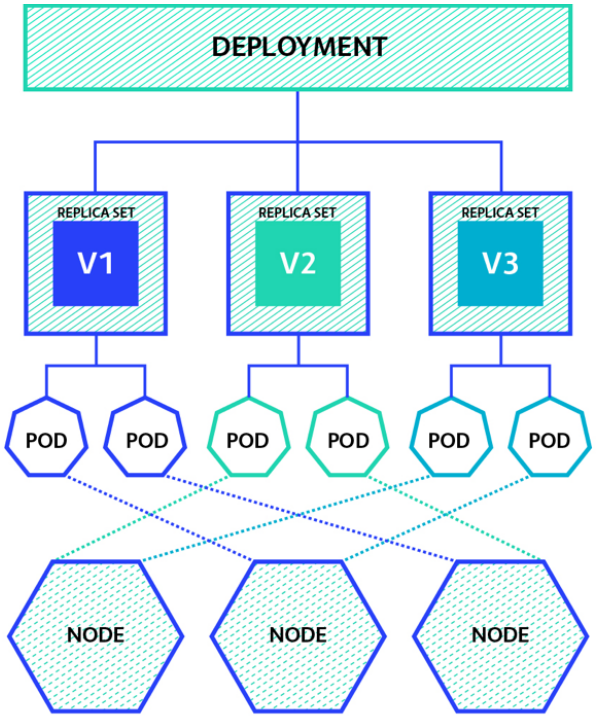
IT-CS-CT

- Kubernetes Architecture
- Workloads
  - Pods, ReplicaSets, Deployments
  - Secrets & Environment Variables
- Services: ClusterIP vs NodePort
- Load Balancing: Ingress Controller
- Scheduling
  - Clusters set-up
  - Assigning Pods to Nodes
- Milestones

Kubernetes Architecture



Workloads



thenewstack.io

Workloads: Pods, ReplicaSets, Deployments

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: network-microservice-infoream
spec:
  selector:
    matchLabels:
      app: network-microservice-infoream
  replicas: 2
  template:
    metadata:
      labels:
        app: network-microservice-infoream
    spec:
      containers:
        - name: network-microservice-infoream
          image: gitlab-registry.cern.ch/network/monolandb/microservices/infoream:master
          imagePullPolicy: Always
      imagePullSecrets:
        - name: gitlab-registry
```

Workloads: Pods, ReplicaSets, Deployments

```
 pgarciam@aiadm05  ~/kubernetes  kubectl apply -f ssl_enabled/infoream_basic.yaml
deployment.apps/network-microservice-infoream created
 pgarciam@aiadm05  ~/kubernetes  kubectl get all
NAME                                                READY   STATUS    RESTARTS   AGE
pod/network-microservice-infoream-6bdcd8b9cd-ntcz6   1/1     Running   0          7s
pod/network-microservice-infoream-6bdcd8b9cd-tw8gq   1/1     Running   0          7s

NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.254.0.1   <none>        443/TCP   29d

NAME                                           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/network-microservice-infoream   2         2         2            2           7s

NAME                                                      DESIRED   CURRENT   READY   AGE
replicaset.apps/network-microservice-infoream-6bdcd8b9cd   2         2         2       7s
```

Workloads: Secrets & Environment Variables

```
$ kubectl create secret generic key-store \
                --from-file=host.jks=host.jks \
                --from-literal=key_store_password=changeit
```

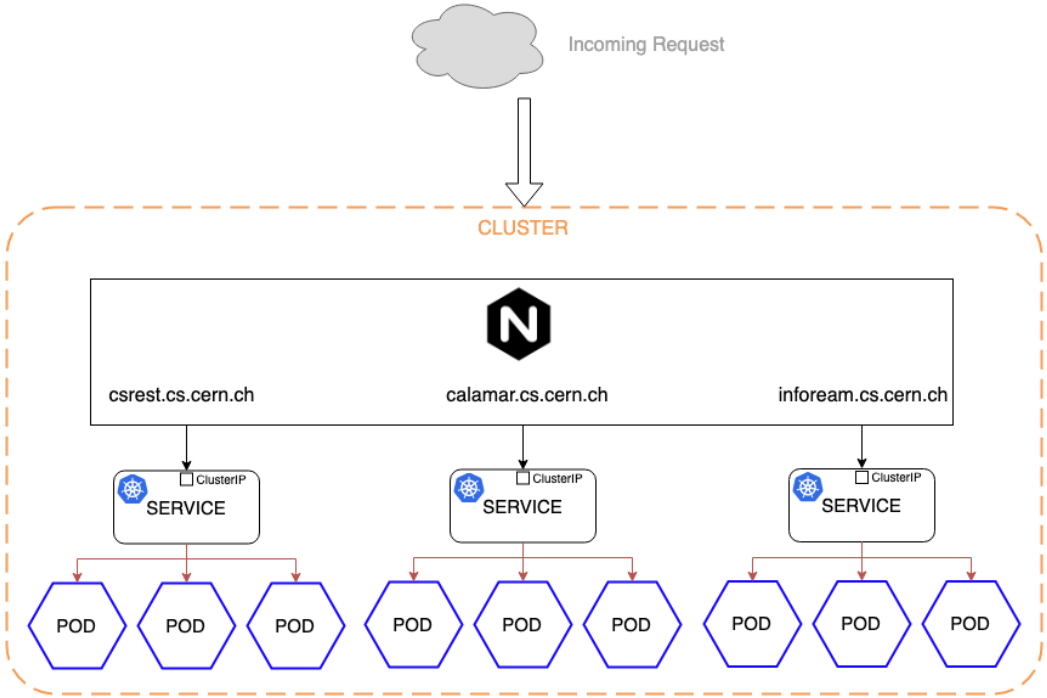
```
spec:
  containers:
    - name: network-microservice-infoream
      ...
      volumeMounts:
        - name: key-store-secret
          mountPath: /opt/service/config
          readOnly: true
      env:
        - name: "KEY_STORE_PASSWORD"
          valueFrom:
            secretKeyRef:
              name: key-store
              key: key_store_password
  volumes:
    - name: key-store-secret
      secret:
        secretName: key-store
        items:
          - key: host.jks
            path: application.jks
```

Services: ClusterIP vs NodePort

```yaml
apiVersion: v1
kind: Service
metadata:
  name: network-microservice-infoream
spec:
  selector:
    app: network-microservice-infoream
  ports:
    - name: https
      port: 443
      targetPort: 8080
  type: ClusterIP
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: network-microservice-infoream
spec:
  selector:
    app: network-microservice-infoream
  ports:
    - name: https
      port: 443
      targetPort: 8080
      nodePort: 30000
  type: NodePort
```

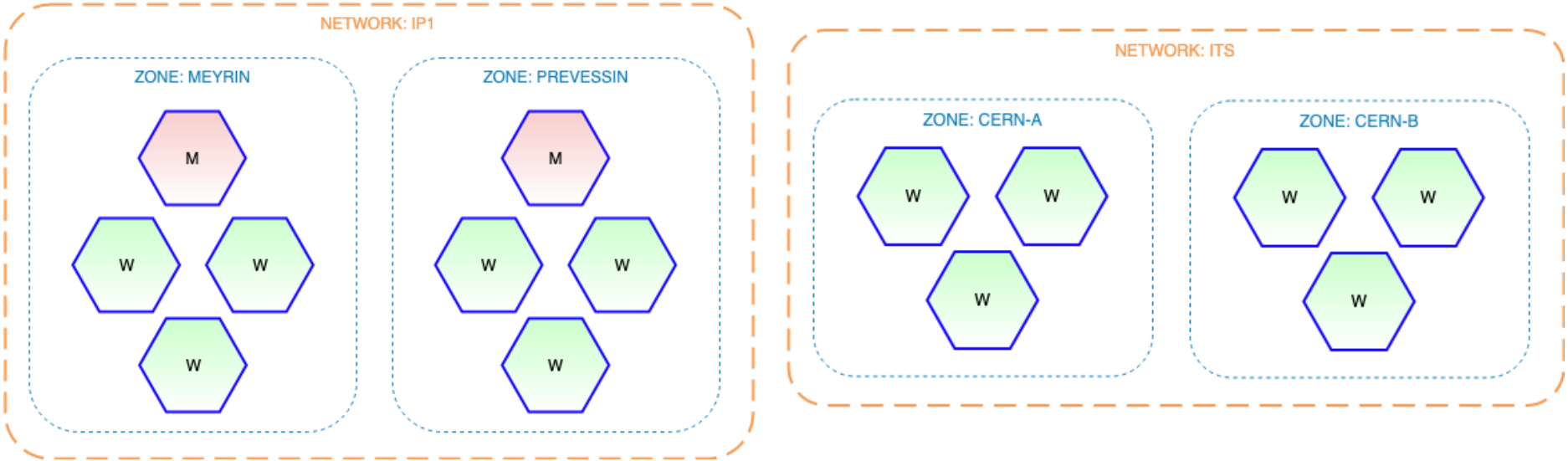
Load Balancing: Ingress Controller



Load Balancing: Ingress Controller

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: nginx-ingress
  namespace: default
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/ssl-passthrough: "true"
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
spec:
  rules:
    - host: infoream-network-microservice.cern.ch
      http:
        paths:
          - path: /
            backend:
              serviceName: network-microservice-infoream
              servicePort: 443
```

Scheduling: Clusters set-up

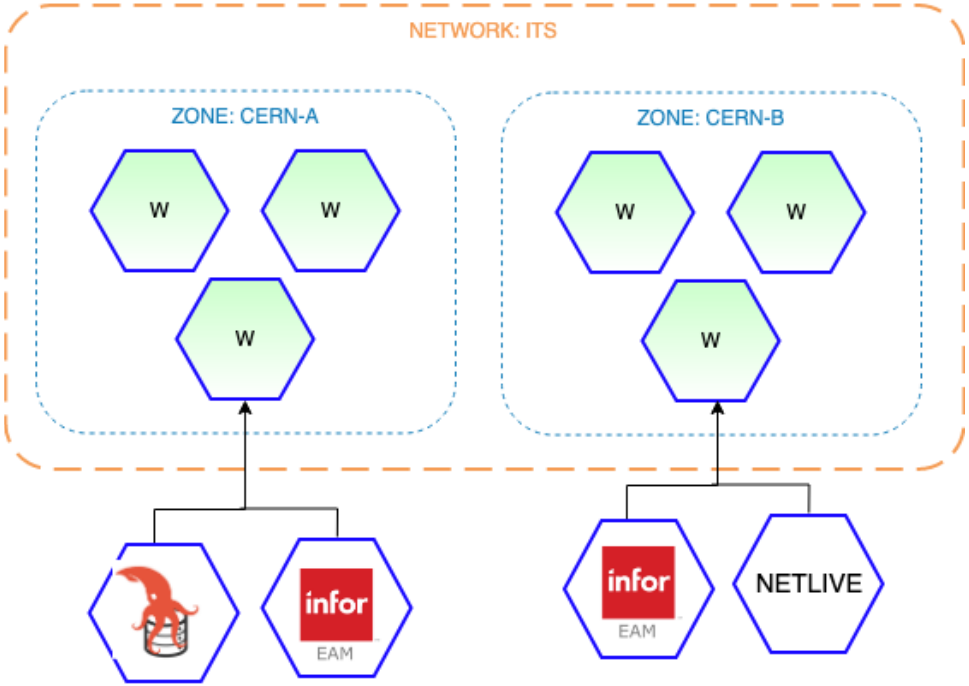


Scheduling: Clusters set-up

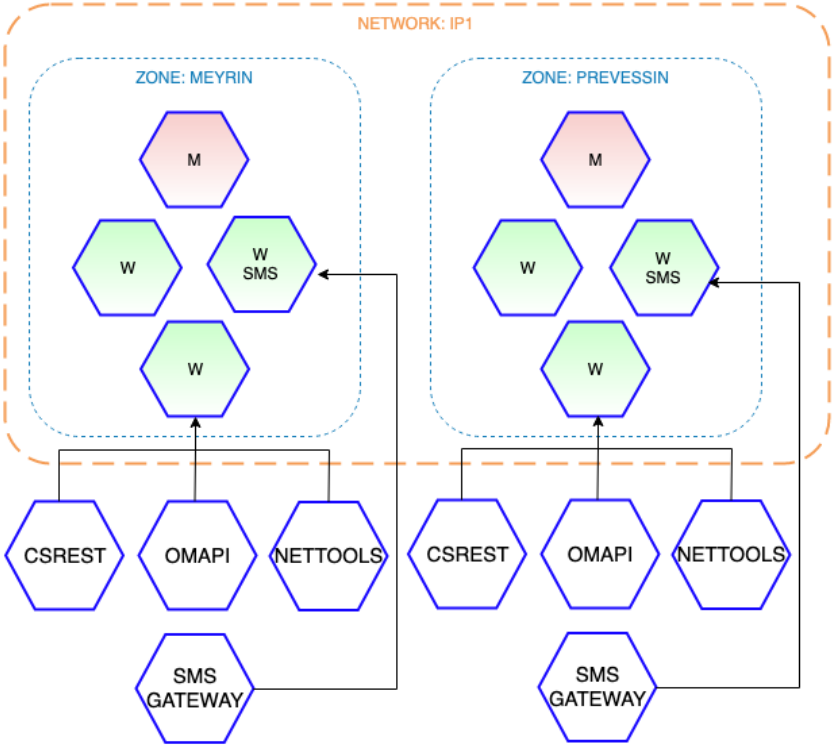
```
$ kubectl get nodes
NAME                                                STATUS   ROLES    AGE   VERSION
network-microservices-multinod-djyb6qfro67s-master-0   Ready    master   9m    v1.11.6
network-microservices-multinod-djyb6qfro67s-minion-1   Ready    <none>   9m    v1.11.6
network-microservices-multinod-djyb6qfro67s-minion-2   Ready    <none>   9m    v1.11.6
network-microservices-multinod-djyb6qfro67s-minion-3   Ready    <none>   9m    v1.11.6
network-microservices-multinod-djyb6qfro67s-minion-4   Ready    <none>   9m    v1.11.6
network-microservices-multinod-djyb6qfro67s-minion-5   Ready    <none>   9m    v1.11.6

$ kubectl label nodes network-microservices-multinod-djyb6qfro67s-minion-0 network=IP1 zone=meyrin
$ kubectl label nodes network-microservices-multinod-djyb6qfro67s-minion-1 network=IP1 zone=meyrin capability=sms
$ kubectl label nodes network-microservices-multinod-djyb6qfro67s-minion-2 network=IP1 zone=prevessin
$ kubectl label nodes network-microservices-multinod-djyb6qfro67s-minion-3 network=IP1 zone=prevessin capability=sms
$ kubectl label nodes network-microservices-multinod-djyb6qfro67s-minion-4 network=ITS zone=cern-a
$ kubectl label nodes network-microservices-multinod-djyb6qfro67s-minion-5 network=ITS zone=cern-b
```

Scheduling: Pod and Node Affinity



Scheduling: Pod and Node Affinity



Scheduling: Pod and Node Affinity

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: network-microservice-infoream
spec:
  selector:
    matchLabels:
      app: network-microservice-infoream
  replicas: 2
  template:
    metadata:
      labels:
        app: network-microservice-infoream
    spec:
      containers:
        ...
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: network
                    operator: In
                    values:
                      - ITS
```

```yaml
    spec:
      containers:
        ...
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            nodeSelectorTerms:
            - matchExpressions:
              - key: network
                operator: In
                values:
                - IP1
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchExpressions:
              - key: app
                operator: In
                values:
                - network-microservice-csrest
            topologyKey: zone
```

Milestones

1. Cluster creation and hostname-based routing with SSL pass-through
2. Assigning pods to nodes based on the requirements for each microservice
3. Demonstrate mounting/sharing of USB ports inside pods
4. Integration with Gitlab
5. Monitoring and alarms

Milestones

6. Secret management
7. High availability via DNS load-balancing
8. Dashboard with security
9. Migration of our microservices to the new infrastructure
10. Auto-scaling of deployments and of cluster*
11. Automatic host certificate renewal*

** Improvements*

Questions


CERN