KUBERNETES MASTERCLASS

- BY Praveen Singampalli founder of Heydevops
- Linkedin - https://www.linkedin.com/in/praveen-singampalli/
- Instagram - https://www.instagram.com/singam4devops/
- Download HeyDevOps App on Android

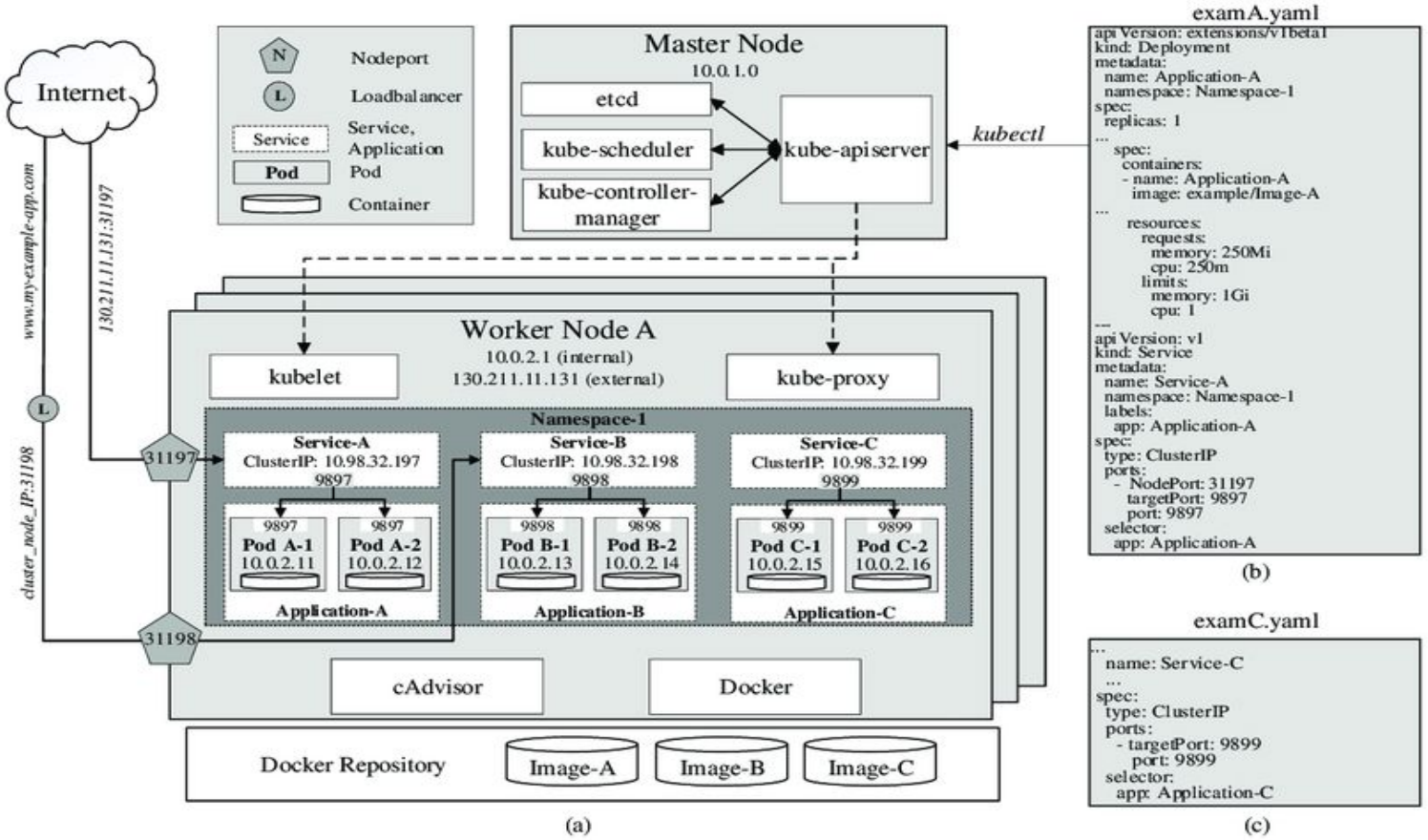
Why Kubernetes?

Docker is an image building, storing and container creation technology but kubernetes is container orchestration technology.
At one point of time it is very difficult to manage docker containers spanning over multiple hosts.
So we need Kubernetes for the below

- Auto Scaling Networking
- High Availability
- Reliability Self Healing
- Storage mechanism

HEY DEVOPS

KUBERNETES ARCHITECTURE



Service in Kubernetes

Pods are naturally ephemeral in Kubernetes.So Pod IP will be changed every time it is created. To achieve pod to pod communication we can't re on IP addresses. Solution is service.

Service can have name and Port when a pod is created it can go attached to service itself. It acts as a service mesh. Multiple pods can be attached to service based on label selectors. Service acts as Load balancer between pods.

Types of Service

Cluster IP: Default type. It can get one IP address. We can configure cluster IP service to achieve pod to pod communication, But it can't be accessed over the internet.

Node Port: Node Port by default creates cluster IP in background. When we say NodeRort a port will be opened on each and every node. This port will be redirected to the cluster IPNode Port can be accessed over the internet.

Load Balancer: We can create Load Balancer through cloud providers like AWS, GCP, Azure, etc. Load Balancer by default creates NodePort and Cluster IP in the background. Load Balancer -> NodePort -> Cluster IP

HEY DEVOPS

Replica Set

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx
  labels: # these are labels related to replica set resource
    app: nginx
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels: #this is the syntax replica set uses to find the pods
      tier: frontend
  template: # pod template, labels are related to pod
    metadata:
      labels:
        tier: frontend
    spec:
```

It guarantees the declared no of pods will run always. It is the most important feature of high availability and autoscale.

Deployment

HEY DEVOPS

Deployment creates Replica Set in background to maintain the desired number of replicas We can use deployment for stateless applications. Finally Deployment is the highest resource in Kubernetes to maintain Availability - Make sure desired no of replicas are always available Scalability - Can be used to scale at runtime based on traffic. Maintainability - We can do the updates with new image versions.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  annotations:
    deployment.kubernetes.io/change-cause: "Upgraded to version 2.0"
  labels:
    app: nginx
spec:
  replicas: 10
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 2 # we asked for 10, we are okay if 9 are running at the time of upgrade
      maxSurge: 2 # we asked for 10, we are okay at the time of upgrade 11 are running
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.22.1
        ports:
        - containerPort: 80
```

Daemon Set

The Daemon set is similar to Deployment. But there is only one difference. Deployment makes pods available in any node.

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
      - name: fluentd-elasticsearch
        image: quay.io/fluentd_elasticsearch/fluentd:v2.5.2
        resources:
          limits:
            memory: 200Mi
          requests:
            cpu: 100m
            memory: 200Mi
        volumeMounts:
        - name: varlog
          mountPath: /var/log
      volumes:
      - name: varlog
        hostPath:
          path: /var/log
```