Kubernetes의 기본적인 Networking 과 Service

10.0.0.0 – **10**.255.255.255
172.16.0.0 – 172.31.255.255
192.168.0.0 – **192.168**.255.255

- **Private IP Address** and **Public IP Address** are used to uniquely identify a machine on the internet. **Private IP address** is used with a local network and **public IP address** is used outside the network. ...
  Public IP Address is used to communicate outside the network.

참고: 기본지식: 인터넷 주소 기초, Switch/Bridge, Router,...Public IP address, Private IP address

Kubernetes Network Basic

1. Every Pod gets its own IP address – PoD IP address--Networking SW(CNI Pugin) assign this.
2. Pods on a node should be able to communicate with all Pods on all nodes without NAT. --- Networking SW(CNI Pugin) Do this.
3. Containers within a Pod share a network namespace (IP and MAC address), so they can communicate with each other using the loopback address.

Terminology related to IP addresses in Kubernetes

Services, **Pods**, **containers**, and **nodes** communicate using IP addresses and ports.

- **Node IP**: The IP address assigned to a given node.

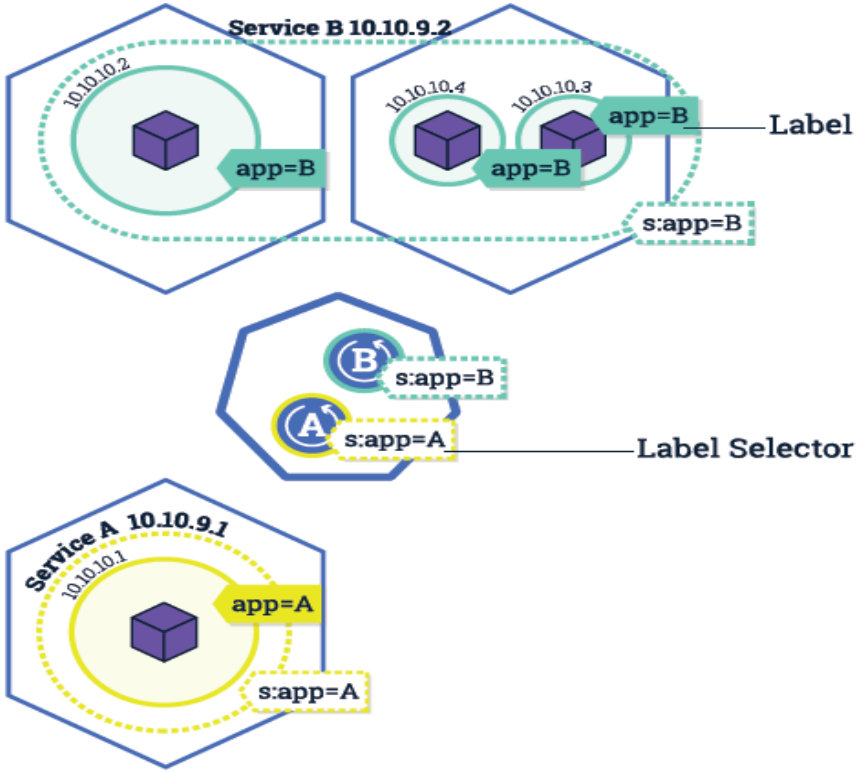
- **Pod IP**: The IP address assigned to a given Pod. This is ephemeral.(임시적이다) Each Pod gets its own IP address (network plugins(CNI plugin) do this). For a given Deployment in your cluster, the set of **Pods running in one moment in time could be different from the set of Pods running** that application a moment **later**.

- **ClusterIP**: The IP address assigned to a **Service**. This address is **stable for the lifetime of the Service.**

K8S에서 Service 란?

- A Service in Kubernetes is an abstraction which defines a logical set of Pods and a policy by which to access them.
- The set of Pods targeted by a Service is usually determined by a *LabelSelector*
- Although each Pod has a unique IP address, those IPs are not exposed outside the cluster without a Service. Services allow your applications to receive traffic.
- Services can be exposed in different ways by specifying a type in the ServiceSpec:
  - **ClusterIP** (default) - Exposes the Service on an internal IP in the cluster. This type makes the Service only reachable from within the cluster.
  - **NodePort** - Exposes the Service on the same port of each selected Node in the cluster using NAT. Makes a Service accessible from outside the cluster using <NodeIP>:<NodePort>. Superset of ClusterIP.
  - **LoadBalancer** - Creates **an external load balancer** in the current cloud (if supported) and assigns a fixed, external IP to the Service. Superset of NodePort.
  - **ExternalName** - Exposes the Service using an arbitrary name (specified by externalName in the spec) by returning a CNAME record with the name. No proxy is used. This type requires v1.7 or higher of **kube-dns.**

Service, Label, Label Selector



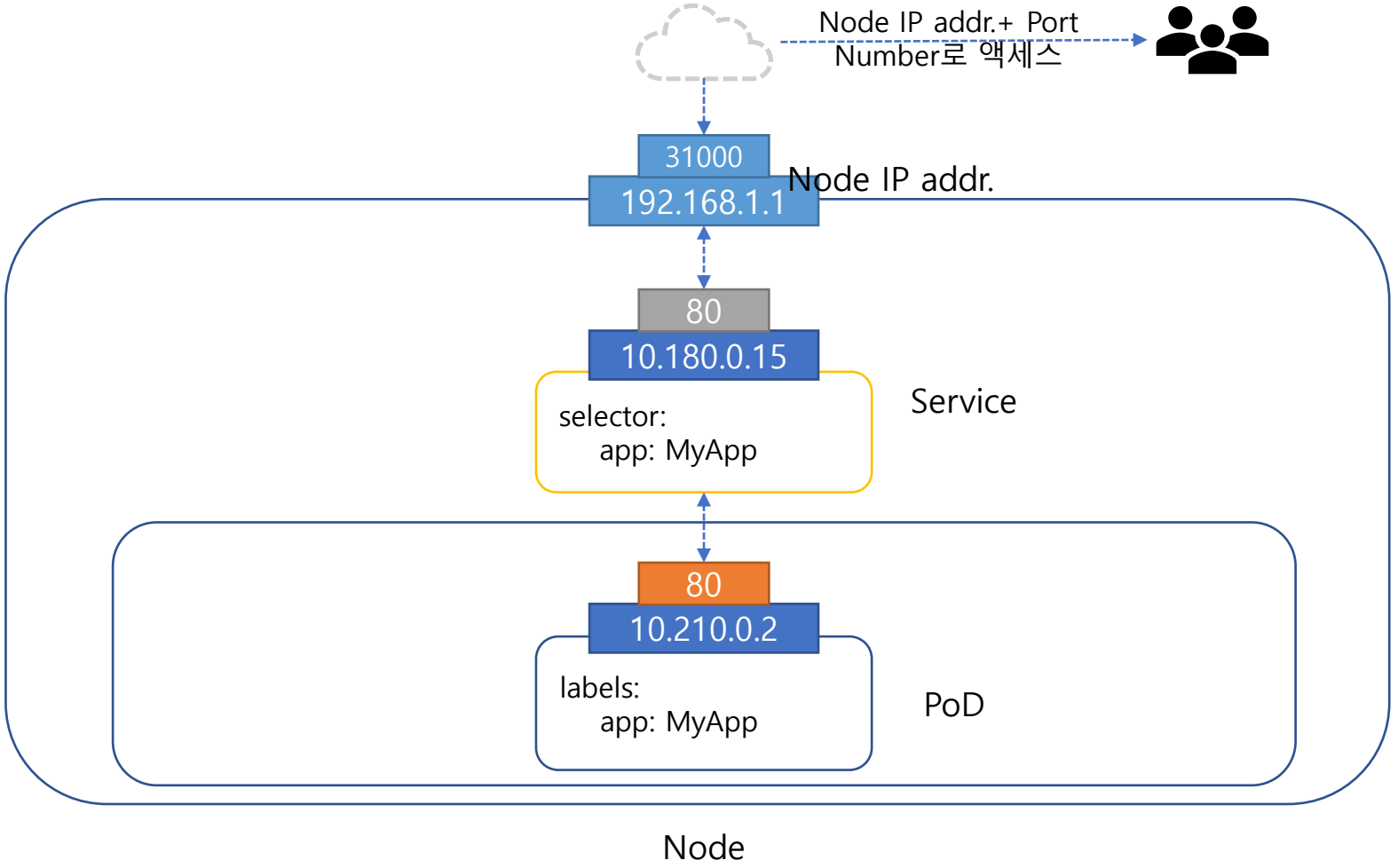
Cluster IP Service ..Cluster IP주소를 통해 access 가능한 service 란 이야기

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
     app.kubernetes.io/name: MyApp
  ports:
   - protocol: TCP
     port: 80
     targetPort: 9376
```

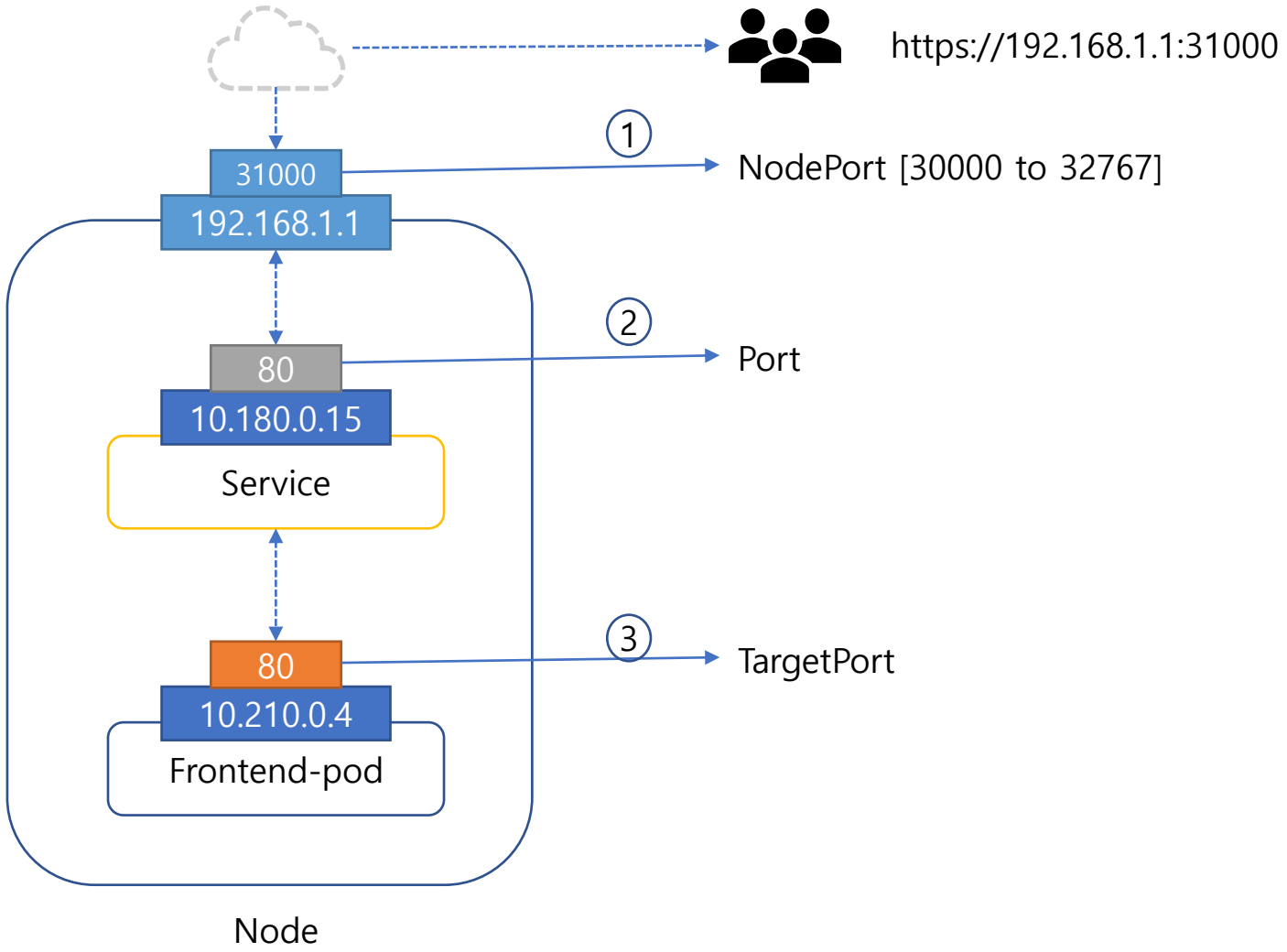
Nodeport Service .. Node의 IP주소와 별도 지정한 port값을 이용해서 access가능한 service란 이야기

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app.kubernetes.io/name: MyApp
  ports:
    # By default and for convenience, the `targetPort` is set to the same value as the `port` field.
    - port: 80
      targetPort: 80
      # Optional field
      # By default and for convenience, the Kubernetes control plane will allocate a port from a range (default: 30000-32767)
      nodePort: 30007
```

NodePort Service – PoD를 외부에 노출하기 위한 방법 중 하나

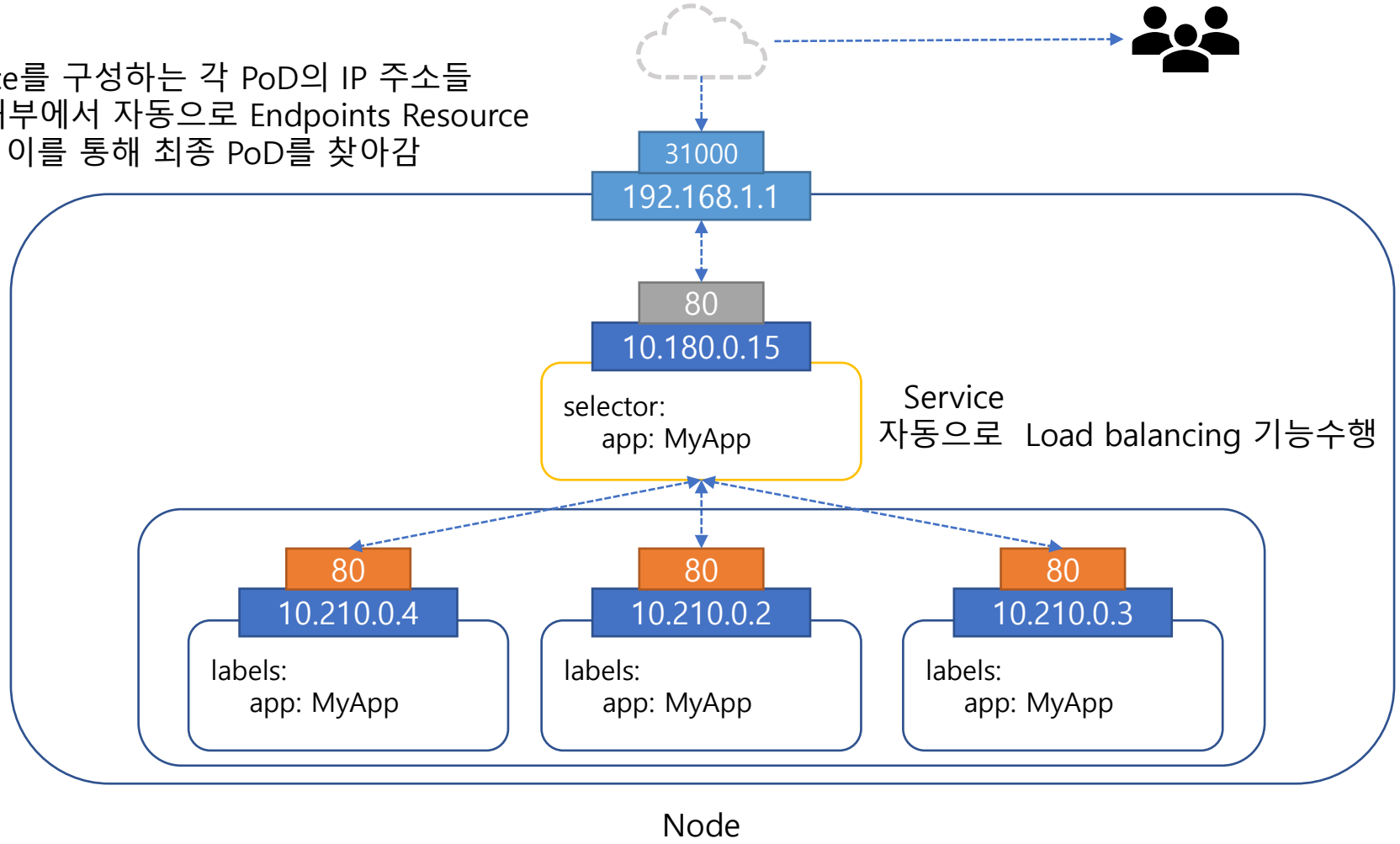


Port 종류

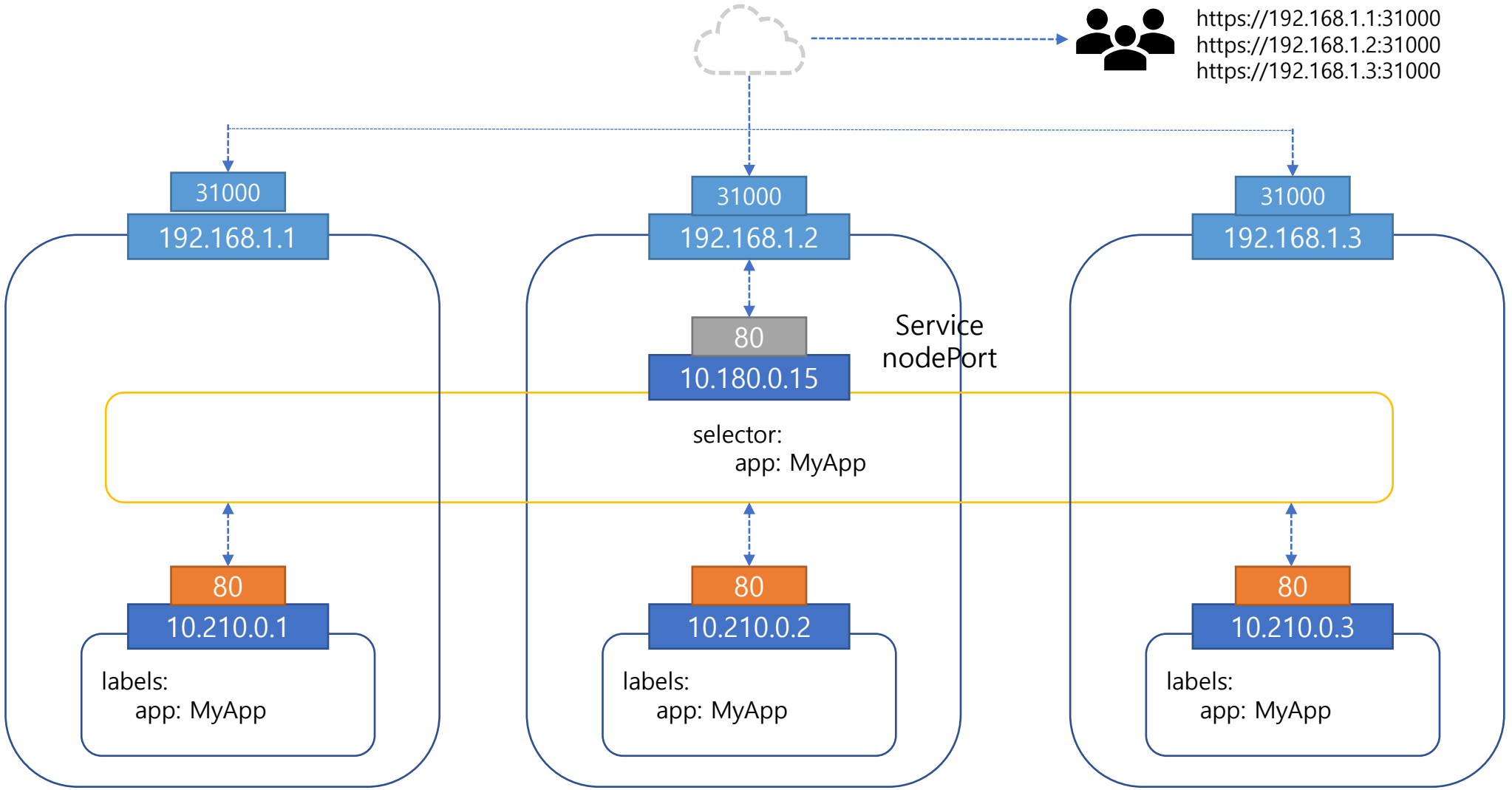

https://192.168.1.1:31000
1
NodePort [30000 to 32767]
31000
192.168.1.1
2
Port
80
10.180.0.15
Service
3
TargetPort
80
10.210.0.4
Frontend-pod
Node

단일 노드에 여러 개의 PoD로 구성된 서비스 예

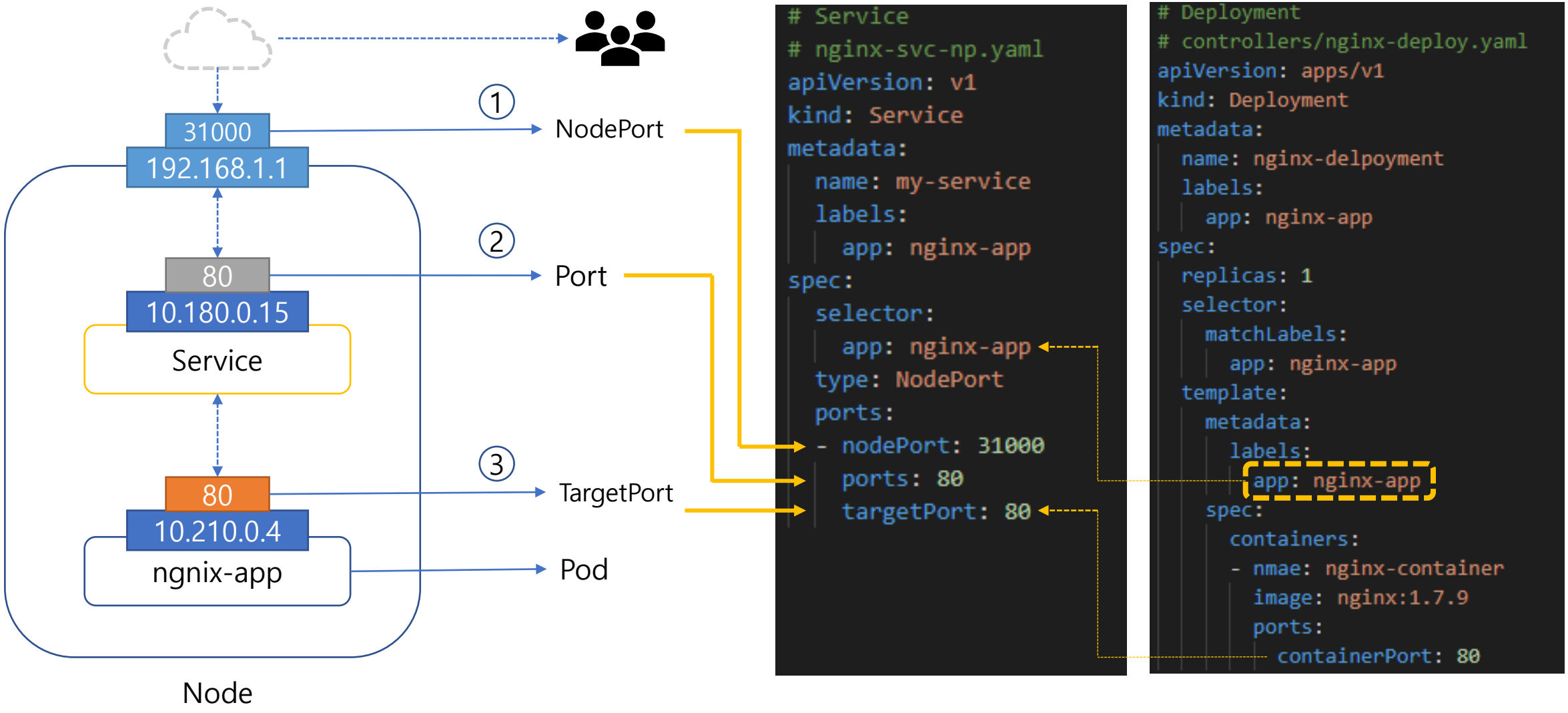
Endpoints : service를 구성하는 각 PoD의 IP 주소들
(Service를 생성하면 내부에서 자동으로 Endpoints Resource
를 생성시키고 이를 통해 최종 PoD를 찾아감



여러 노드에 분산된 PoD들로 구성된 서비스 예



NodePort Service 를 위한 Yaml 파일의 내용 예



Deployment with Service definition

- Apache-Deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: apache-deployment
spec:
  selector:
    matchLabels:
      app: webserver
  replicas: 1
  template:
    metadata:
      labels:
        app: webserver
    spec:
      containers:
      - name: php-apache
        image: k8s.gcr.io/hpa-example
        ports:
        - containerPort: 80
```

Continue (Don't forget ---)

```yaml
---

apiVersion: v1
kind: Service
metadata:
  name: webserver
  labels:
    app: webserver
spec:
  type: NodePort
  ports:
  - port: 8080
    targetPort: 80
    protocol: TCP
    name: http
  selector:
    app: webserver
```