Application Deployment

Kubernetes – PoD, ReplicaSet, Deployment

애플리케이션 배포방식

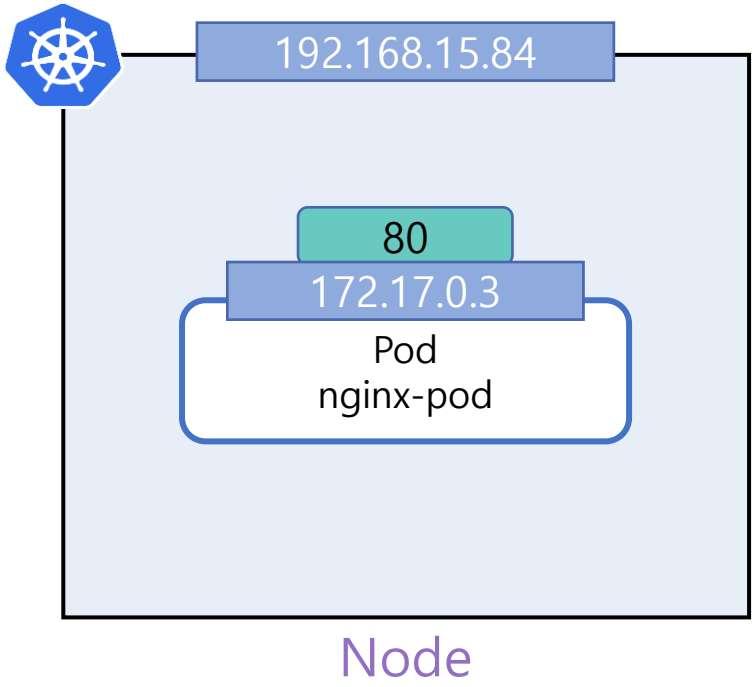
- Pod 타입으로 배포
  - 애플리케이션을 pod의 형태로 배포한다.

- ReplicaSet 타입으로 배포
  - 애플리케이션을 ReplicaSet 형태로 배포한다.
  - ReplicaSet은 실행되는 pod 개수에 대해 지정한 replicas 개수만큼 항상 실행될 수 있도록 관리한다.

- Deployment 타입으로 배포
  - Deployment: 1개 이상의 Pod를 관리하는 컨트롤러의 종류 중 하나로써 replicaset resourc를 생성시키고 그후 생성된 replicaset resourc정보에 따라 PoD resource가 해당 개수만큼 생성되어 최종적으로 실제 PoD가 생성되게 한다.

Pod 타입으로 배포

- run 명령어를 이용하여 Pod 배포하기
  - kubectl run [Pod name] --image=[container image name] --port=[port number]
  - ex) kubectl run nginx-pod --image=nginx --port=80
  - kubectl get pods 를 이용하여 배포된 Pod 확인하기

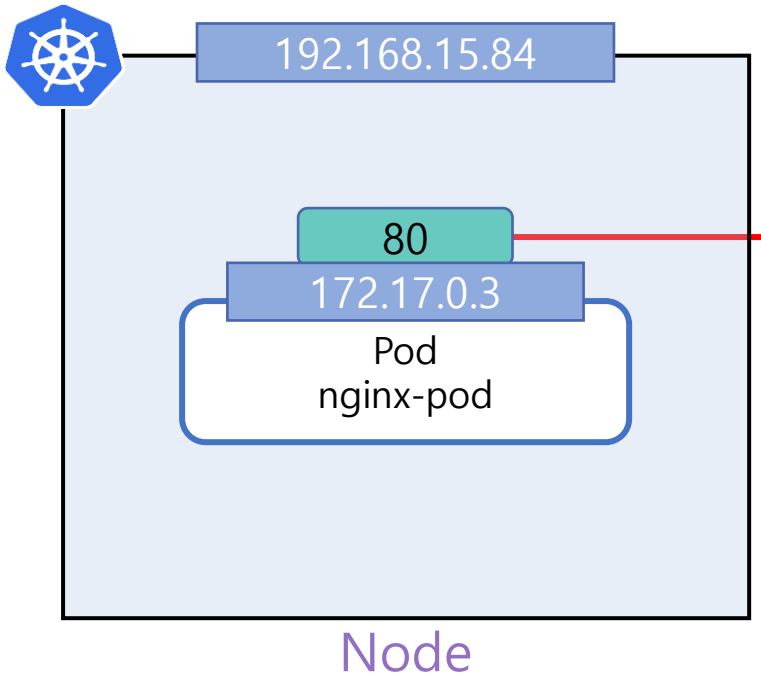



```
mini@minikube:~$ kubectl run nginx-pod --image=nginx --port 80
pod/nginx-pod created
mini@minikube:~$ kubectl get pods
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          9s
```

<실행결과>

Pod 타입으로 배포

- yaml 파일을 이용하여 Pod 배포하기
  - kubectl create -f [File name]
  - ex) kubectl create -f nginx-pod.yaml

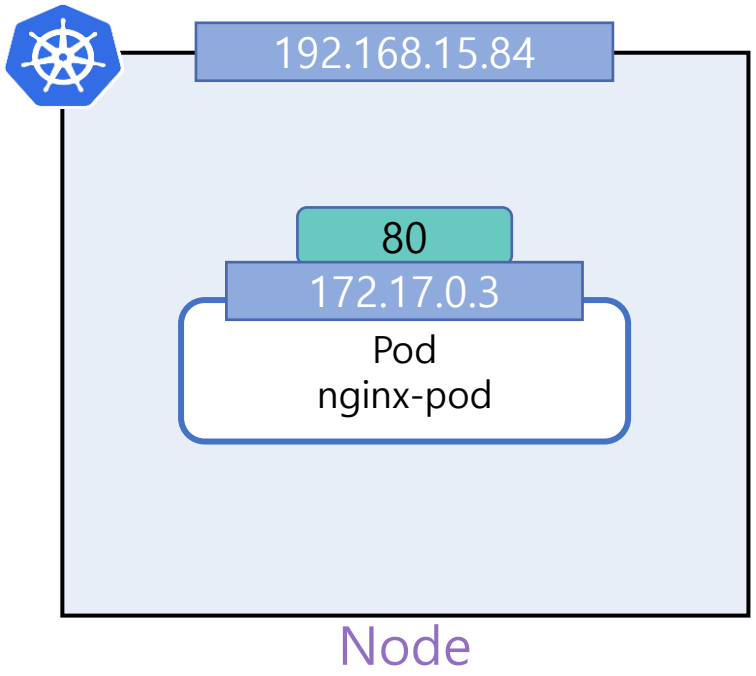



```yaml
# nginx-pod.yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx-container
    image: nginx
    ports:
    - containerPort: 80
      protocol: TCP
```

<nginx-pod YAML 파일>

Pod 타입으로 배포 – 상세 정보 출력

- namespace, label, IP address, container의 상세 정보 확인 가능
- kubectl describe pod [Pod name]
- ex) kubectl describe pod nginx-pod




```
mini@minikube:~$ kubectl describe pod nginx-pod
Name:         nginx-pod
Namespace:    default
Priority:     0
Node:         minikube/192.168.15.84
Start Time:   Tue, 20 Jul 2021 08:02:47 +0000
Labels:       <none>
Annotations:  <none>
Status:       Running
IP:           172.17.0.3
IPs:
  IP:  172.17.0.3
Containers:
  nginx-container:
    Container ID:   docker://2e1485ac168c7c1675a34d13e2b1ffb6f48486f4c0f543792310592
    Image:          nginx:1.14
    Image ID:       docker-pullable://nginx@sha256:f7988fb6c02e0ce69257d9bd9cf37ae20
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 20 Jul 2021 08:02:48 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-gncs8 (ro)
```

<출력결과>

Pod - exec 명령어

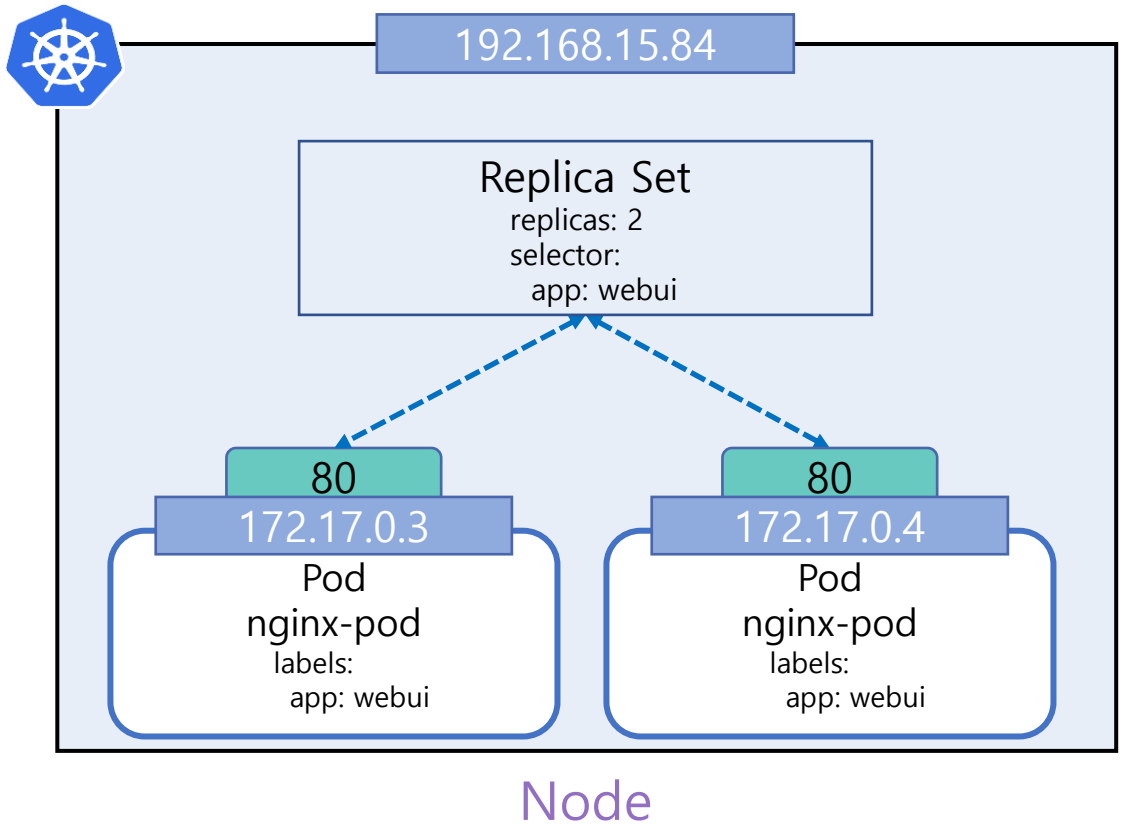
- kubectl exec [Pod name] -- [command]
  - 해당 Pod의 container에 [command] 명령 실행

- exec 명령어를 통해 nginx 컨테이너에 접속 후 index.html 파일 수정
  - kubectl exec -it nginx-pod -- /bin/bash
  - echo “Hello world” > /usr/share/nginx/html/index.html

```
mini@minikube:~$ kubectl exec -it nginx-pod -- /bin/bash
root@nginx-pod:/# echo "Hello world" > /usr/share/nginx/html/index.html
root@nginx-pod:/# exit
exit
mini@minikube:~$ curl 172.17.0.3:80
Hello world
```

ReplicaSet 타입으로 배포

- yaml 파일을 이용하여 배포하기
  - kubectl apply -f [File name]
  - ex) kubectl apply -f rs-nginx.yaml



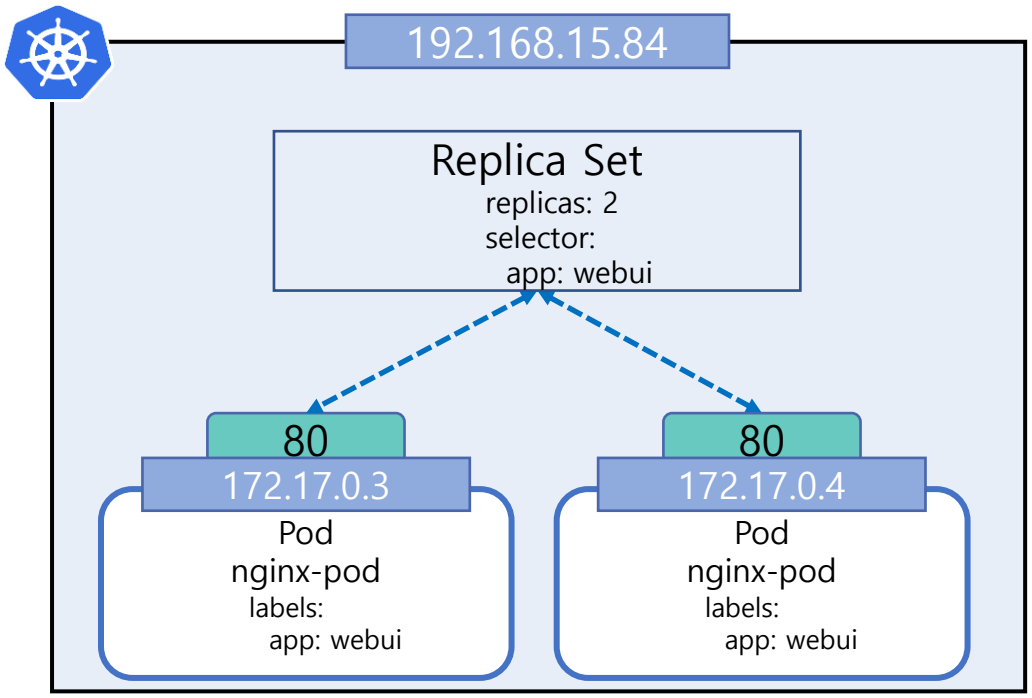

```yaml
# rs-nginx.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rs-nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: webui
  template:
    metadata:
      name: nginx-pod
      labels:
        app: webui
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
        ports:
        - containerPort: 80
          protocol: TCP
```

<rs-nginx yaml 파일>

ReplicaSet 타입으로 배포

ReplicaSet 상세 정보 출력

- kubectl describe replicasets [ReplicaSet name]
- ex) kubectl describe replicasets rs-nginx



Node

```
mini@minikube:~$ kubectl describe rs rs-nginx
Name:         rs-nginx
Namespace:    default
Selector:     app=webui
Labels:       <none>
Annotations:  <none>
Replicas:     2 current / 2 desired
Pods Status:  2 Running / 0 Waiting / 0 Succeeded / 0 Failed
Pod Template:
  Labels:  app=webui
  Containers:
   nginx-container:
    Image:        nginx:1.14
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
```

<출력결과>

ReplicaSet - replicas 수 유지

- 콘솔에서 2개의 세션을 열어
  - 하나의 세션에서는 watch 명령어를 이용하여 Pod 리스트를 변화 관찰
    - watch -n 1 kubectl get pods
  - 나머지 세션에서는 Pod를 인위적으로 삭제해봄
    - kubectl delete pod [Pod name]
- 지정한 Pod는 삭제되었지만 ReplicaSet에 의해 새로운 Pod가 생성되는 것을 확인 가능

```
NAME                 READY   STATUS              RESTARTS   AGE
pod/rs-nginx-sdm2b   0/1     ContainerCreating   0          0s
pod/rs-nginx-w65hv   1/1     Terminating         0          63s
pod/rs-nginx-x2hlr   1/1     Running             0          63s
```

- 지정한 replicas 수(2개)가 유지되는 것을 확인
  - kubectl get pods

```
NAME                 READY   STATUS    RESTARTS   AGE
pod/rs-nginx-sdm2b   1/1     Running   0          50s
pod/rs-nginx-x2hlr   1/1     Running   0          113s
```

ReplicaSet - replicas 수 유지

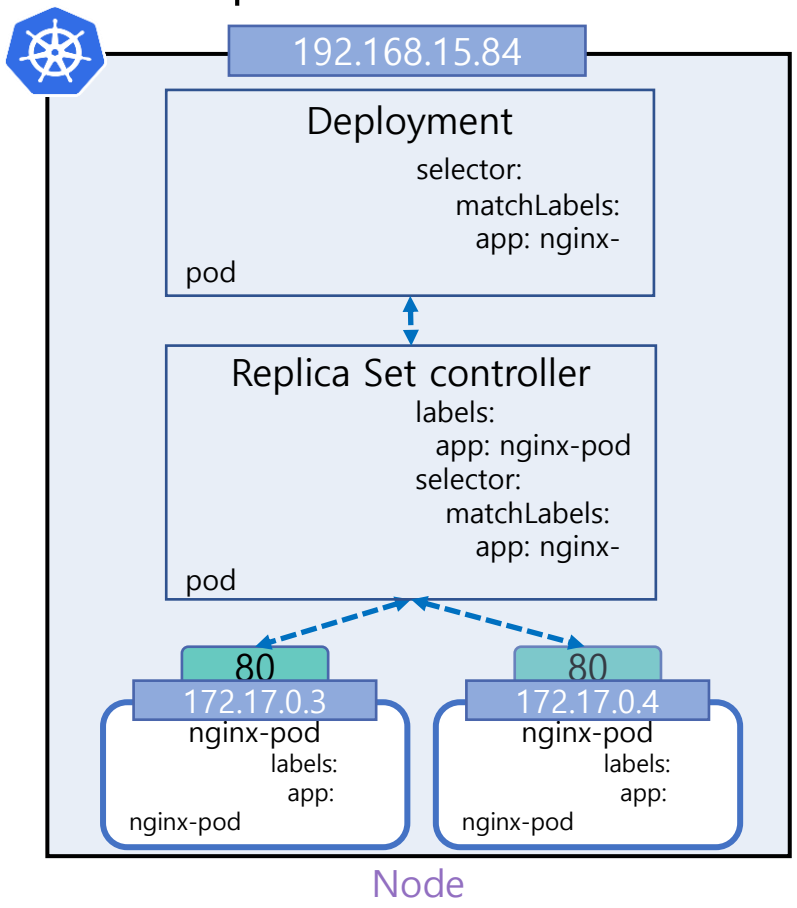
- 지정한 replicas 수를 초과하면 생성한 Pod를 다시 삭제
  - ex) kubectl run testpod --image=nginx:1.14 --labels=app=webui --port=80

```
Every 1.0s: kubectl get rs,po -o wide                                  minikube: Tue Jul 20 11:46:00 2021

NAME                        DESIRED   CURRENT   READY   AGE   CONTAINERS        IMAGES       SELECTOR
replicaset.apps/rs-nginx    2         2         2       32m   nginx-container   nginx:1.14   app=webui

NAME                 READY   STATUS        RESTARTS   AGE     IP            NODE       NOMINATED NODE   READINESS
GATES
pod/rs-nginx-fqcgg   1/1     Running       0          6m26s   172.17.0.4    minikube   <none>           <none>
pod/rs-nginx-rthm4   1/1     Running       0          8m39s   172.17.0.3    minikube   <none>           <none>
pod/testpod          0/1     Terminating   0          1s      <none>        minikube   <none>           <none>
```

```
Every 1.0s: kubectl get rs,po -o wide                                  minikube: Tue Jul 20 11:48:2

NAME                        DESIRED   CURRENT   READY   AGE   CONTAINERS        IMAGES       SELECTOR
replicaset.apps/rs-nginx    2         2         2       35m   nginx-container   nginx:1.14   app=webui

NAME                 READY   STATUS    RESTARTS   AGE     IP            NODE       NOMINATED NODE   READINES
S
pod/rs-nginx-fqcgg   1/1     Running   0          8m48s   172.17.0.4    minikube   <none>           <none>
pod/rs-nginx-rthm4   1/1     Running   0          11m     172.17.0.3    minikube   <none>           <none>
```

Deployment 타입으로 배포

- create 명령어를 이용하여 deployment 배포
  - kubectl create deployment [Deployment name] --image=[container image name] --replicas=2

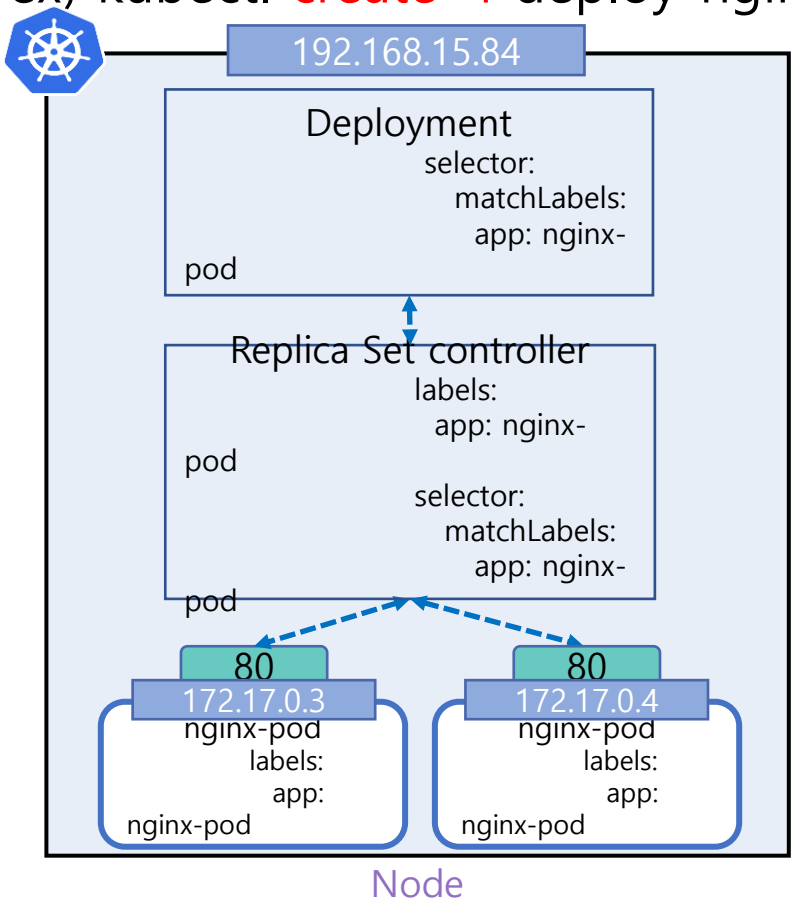


- Deployment로 배포 시 replicas 수를 관리하기 위한 ReplicaSet을 자동 생성
  - kubectl create deployment nginx-pod --image=nginx:1.14 --replicas=2
  - kubectl get deployment, replicaset, pod -o wide

```
Every 1.0s: kubectl get deployment,rs,po -o wide                                                    minikube: Wed Jul 21 02:04

NAME                        READY   UP-TO-DATE   AVAILABLE   AGE   CONTAINERS   IMAGES       SELECTOR
deployment.apps/nginx-pod   2/2     2            2           31s   nginx        nginx:1.14   app=nginx-pod

NAME                                  DESIRED   CURRENT   READY   AGE   CONTAINERS   IMAGES       SELECTOR
replicaset.apps/nginx-pod-66769c85bc   2         2         2       30s   nginx        nginx:1.14   app=nginx-pod,pod-template-hash=66769c85bc

NAME                            READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
pod/nginx-pod-66769c85bc-6cnv9   1/1     Running   0          30s   172.17.0.4   minikube   <none>           <none>
pod/nginx-pod-66769c85bc-xnksl   1/1     Running   0          30s   172.17.0.3   minikube   <none>           <none>
```

Deployment 타입으로 배포

- yaml 파일을 이용하여 deployment 배포
  - kubectl create -f [File name]
  - ex) kubectl create -f deploy-nginx.yaml

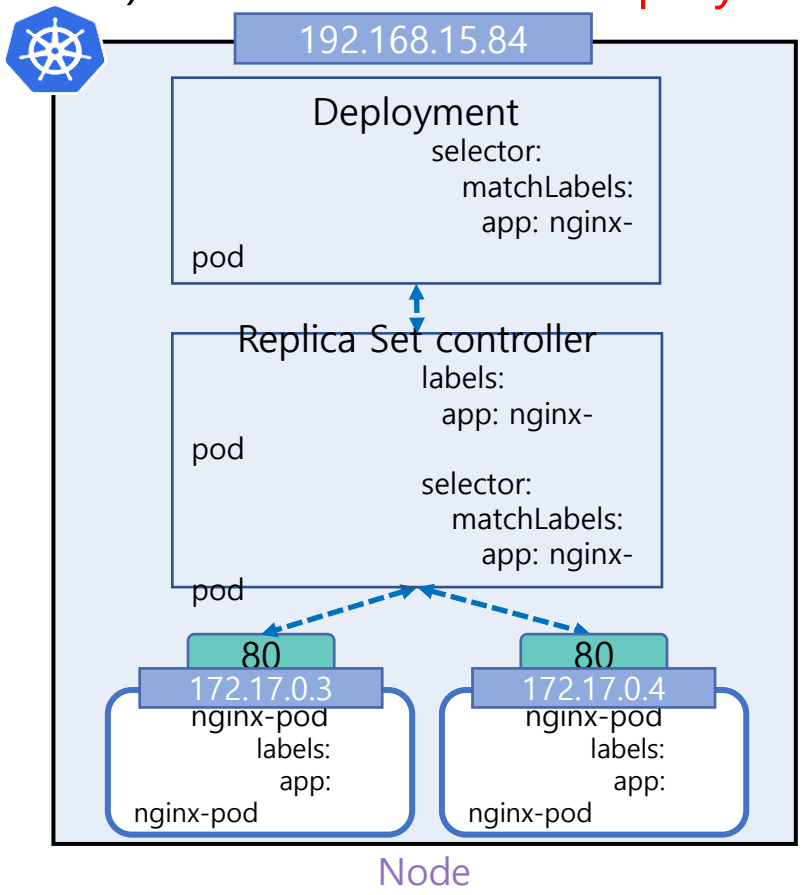



```yaml
# deploy-nginx.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx-pod
  template:
    metadata:
      name: nginx-pod
      labels:
        app: nginx-pod
    spec:
      containers:
      - name: nginx-container
        image: nginx:1.14
```

<deploy-nginx yaml 파일>

Deployment 타입으로 배포

- Deployment 상세 출력
  - kubectl describe deployments [Deployment name]
  - ex) kubectl describe deployments nginx-pod




```
Name:                   nginx-pod
Namespace:              default
CreationTimestamp:      Wed, 21 Jul 2021 02:03:34 +0000
Labels:                 app=nginx-pod
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx-pod
Replicas:               2 desired | 2 updated | 2 total | 2 ava
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx-pod
  Containers:
   nginx:
    Image:        nginx:1.14
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-pod-66769c85bc (2/2 replicas created)
```

<출력결과>

Deployment - ReplicaSet

- ReplicaSet에 문제가 발생했을 경우 자동으로 ReplicaSet을 복제하여 명시된 replicas 개수에 대한 가용성 보증
  - kubectl delete replicasets [ReplicaSet name]