



# **Kubernetes Ingress와 Helm을 이용해서 Application blue/green, canary 배포하기**

**SKT DaeSeong Kim**

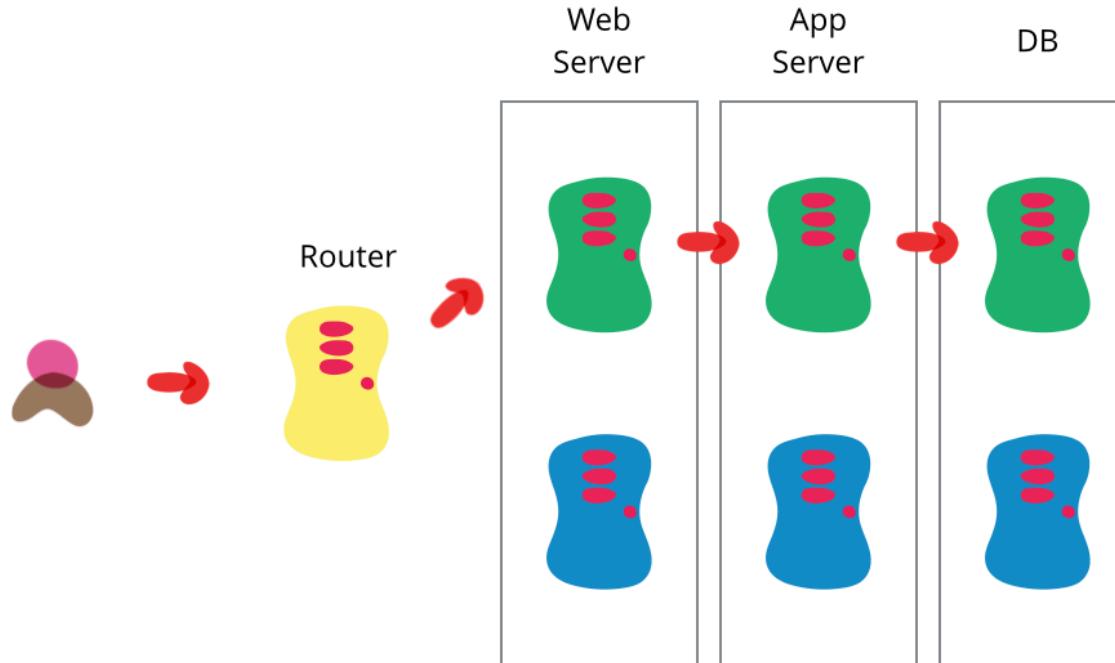
- **Hostway IDC**
- **Innogrid**
- **Samsung SDS**
- **SKT**
  - **5GX Cloud Labs**
    - **TACO 개발/구축**
    - **IT 혁신센터 IT DevOps 그룹 개발 파트**
      - **Kubernetes 플랫폼 서비스 개발 / 구성 / 정책 수립 담당**

### SK Broadband IPTV 어플리케이션들에 실제 적용된 모델

- 세계 최초 해당 국가 내 1위 IPTV 시스템을 Kubernetes로 구축한 사례
- SKB IPTV는 850만대가 넘는 셋탑박스가 실시간으로 접속하는 시스템
- 다양한 요구사항
  - ...
  - 운영 환경에 무중단 앱 배포
    - Blue/green
    - canary

## Blue/Green 배포란

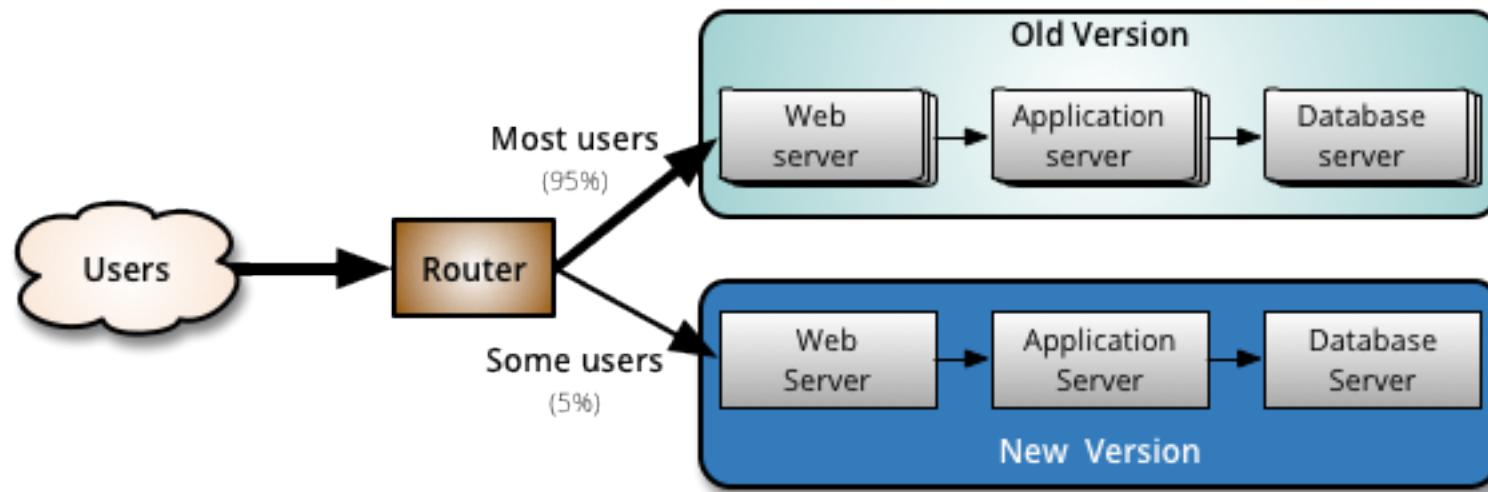
소프트웨어를 신속하게 운영 환경에 배포하고 연속성있게 관리하는 기법



- 출처 : <https://martinfowler.com/bliki/BlueGreenDeployment.html>

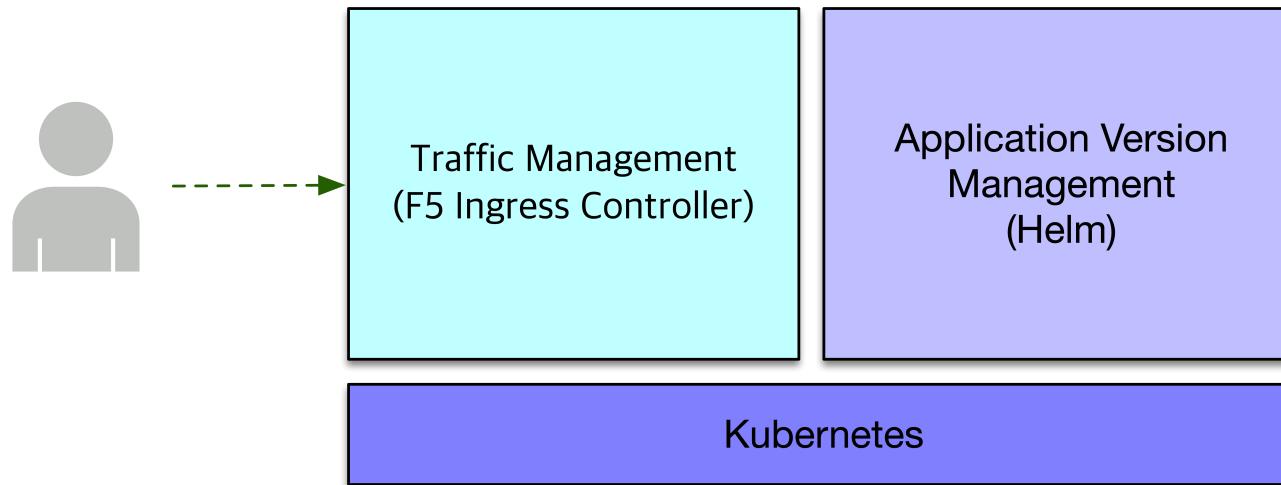
# Canary 배포란

새로운 버전의 소프트웨어를 운영 환경에 배포해서 테스트해보는 기법



- 출처 : <https://martinfowler.com/bliki/CanaryRelease.html>

### 사용자 트래픽 제어 + 어플리케이션 버전 관리



- The blue/green, canary deployment pattern is not supported natively in kubernetes or helm.

## Kubernetes Ingress 란

Kubernetes 클러스터 외부에서 트래픽이 들어올 수 있도록 해주는 경로



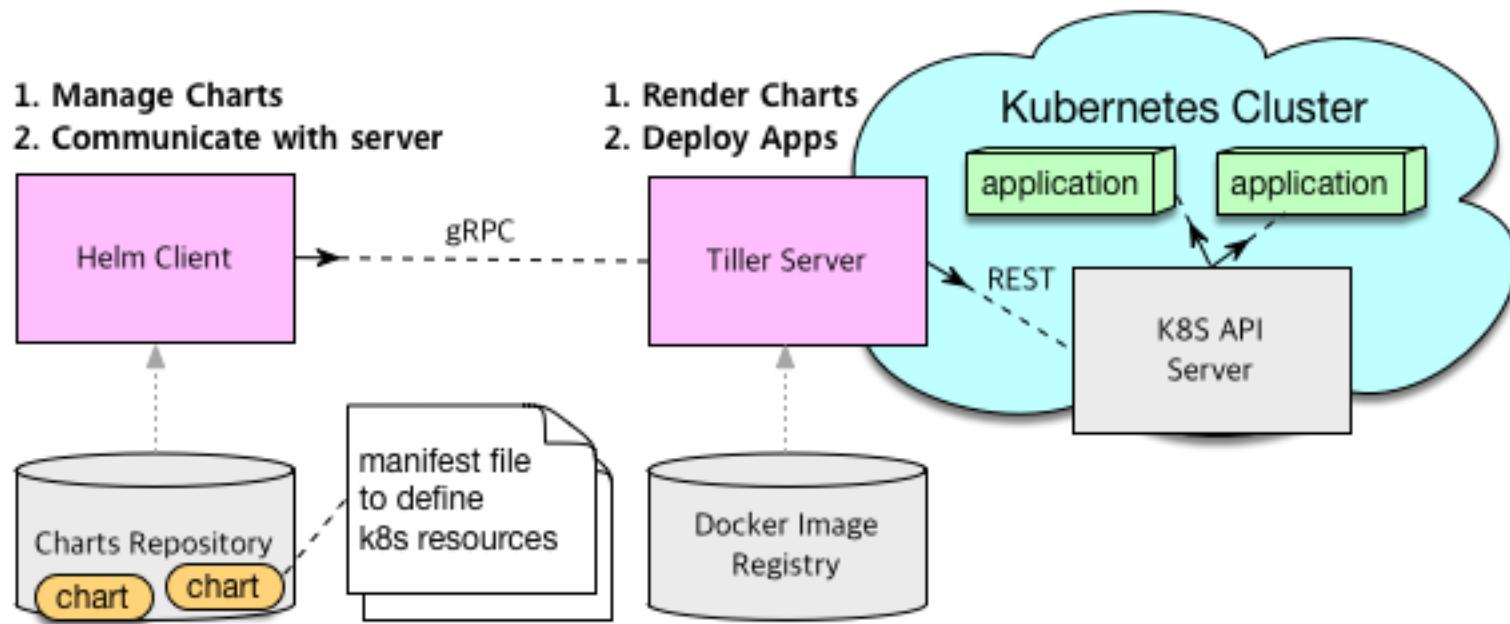
- 출처 : <https://kubernetes.io/docs/concepts/services-networking/ingress/>

## Ingress Controllers

- AKS Application Gateway Ingress Controller is an ingress controller that enables ingress to AKS clusters using the Azure Application Gateway.
- Ambassador API Gateway is an Envoy based ingress controller with community or commercial support from Datawire.
- AppsCode Inc. offers support and maintenance for the most widely used HAProxy based ingress controller Voyager.
- AWS ALB Ingress Controller enables ingress using the AWS Application Load Balancer.
- Contour is an Envoy based ingress controller provided and supported by VMware.
- Citrix provides an Ingress Controller for its hardware (MPX), virtualized (VPX) and free containerized (CPX) ADC for baremetal and cloud deployments.
- F5 Networks provides support and maintenance for the F5 BIG-IP Container Ingress Services for Kubernetes.
- Gloo is an open-source ingress controller based on Envoy which offers API Gateway functionality with enterprise support from solo.io.
- HAProxy Ingress is a highly customizable community-driven ingress controller for HAProxy.
- HAProxy Technologies offers support and maintenance for the HAProxy Ingress Controller for Kubernetes. See the official documentation.
- Istio based ingress controller Control Ingress Traffic.
- Kong offers community or commercial support and maintenance for the Kong Ingress Controller for Kubernetes.
- NGINX, Inc. offers support and maintenance for the NGINX Ingress Controller for Kubernetes.
- Skipper HTTP router and reverse proxy for service composition, including use cases like Kubernetes Ingress, designed as a library to build your custom proxy
- Traefik is a fully featured ingress controller (Let's Encrypt, secrets, http2, websocket), and it also comes with commercial support by Traefik Labs.

- 출처 : <https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/>

## The Kubernetes Package Manager



- 출처 : <https://sktelecom-oslabs.github.io/Virtualization-Software-Lab/Helm/>

### Blue/green, Canary not recommended in Helm

- Helm supports high level ( package level ) update technology.
  - Managing package versions
  - Rollback to previous versions
- <https://github.com/helm/helm/issues/3518>
  - The blue / green deployment pattern is not supported natively in kubernetes or helm.

## Jenkins + 네트워크 제어용 chart + 어플리케이션 버전 관리용 chart



# Jenkins

- \* general-deploy
- \* general-conform
- \* general-rollback
- \* green-deploy
- \* green-conform
- \* green-rollback
- \* canary-deploy
- \* canary-conform
- \* canary-rollback
- \* canary-scale

Traffic Management  
Helm Chart  
(ROUTER CHART)

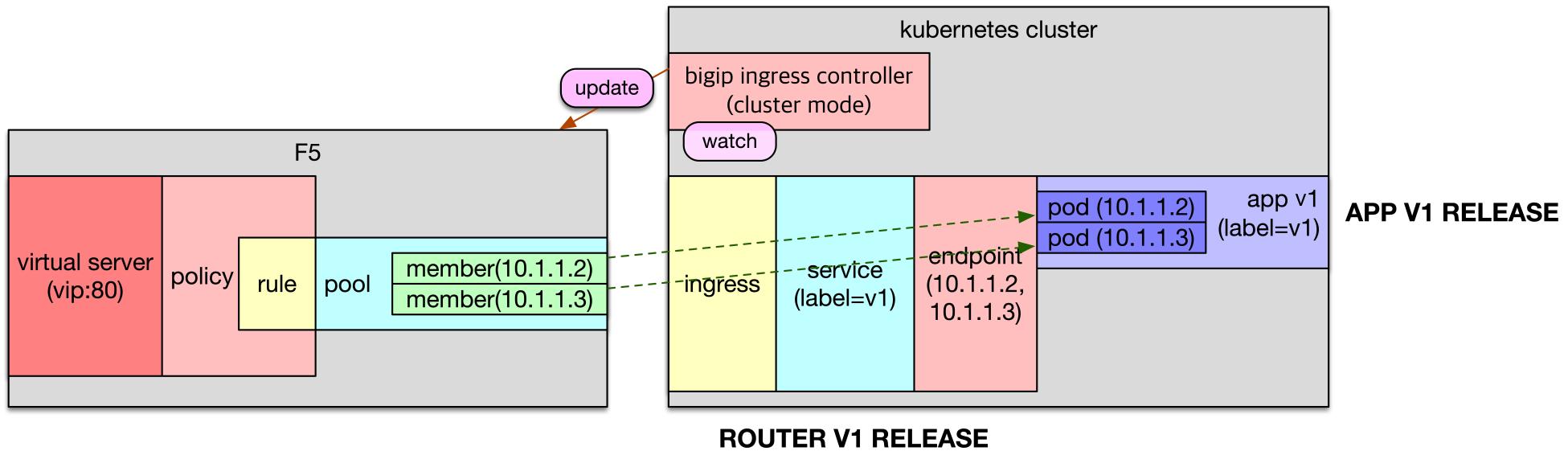
Application Helm Chart  
(APP CHART)

Kubernetes

- Router Chart : Ingress, Service
- App Chart : Deployment, ConfigMp, Secret, Persistent Volume, HPA...

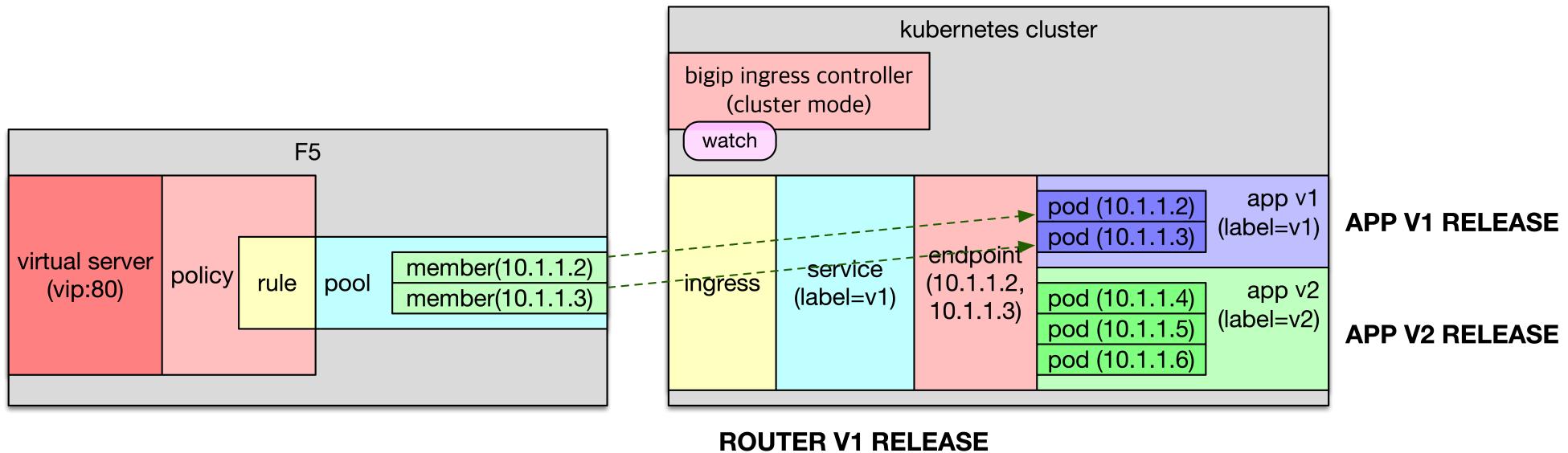
# Blue/Green 배포 Step 1

## Deploy Pipeline



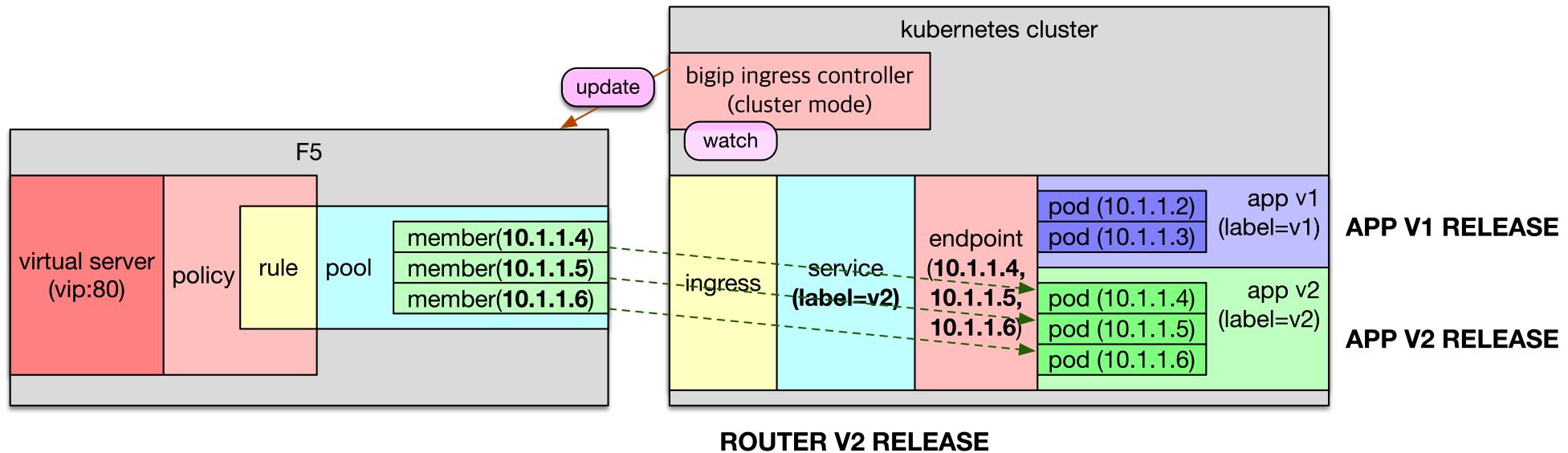
## Blue/Green 배포 Step 2

### Green Deploy Pipeline



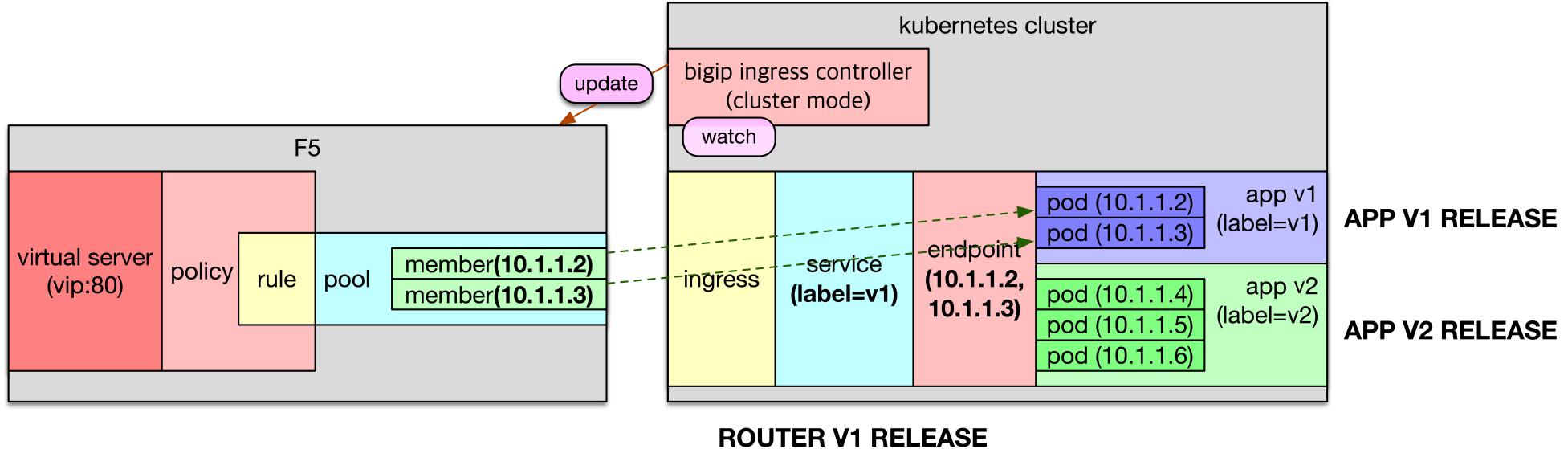
## Blue/Green 배포 Step 3

### Green Confirm Pipeline



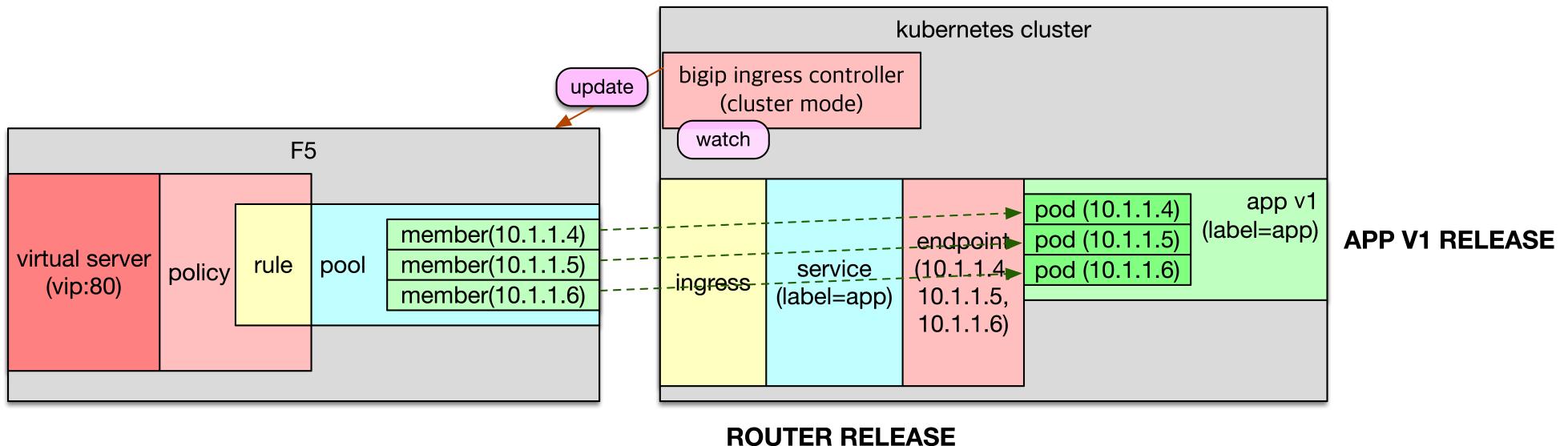
# Blue/Green 배포 Rollback

## Green Rollback Pipeline



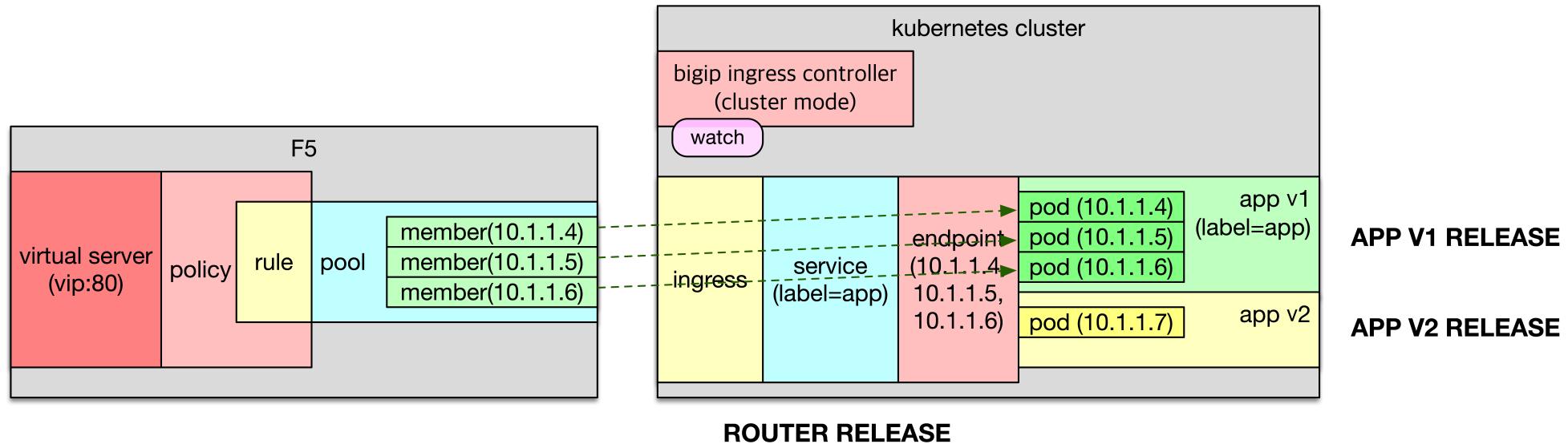
# Canary 배포 Step 1

## Deploy Pipeline



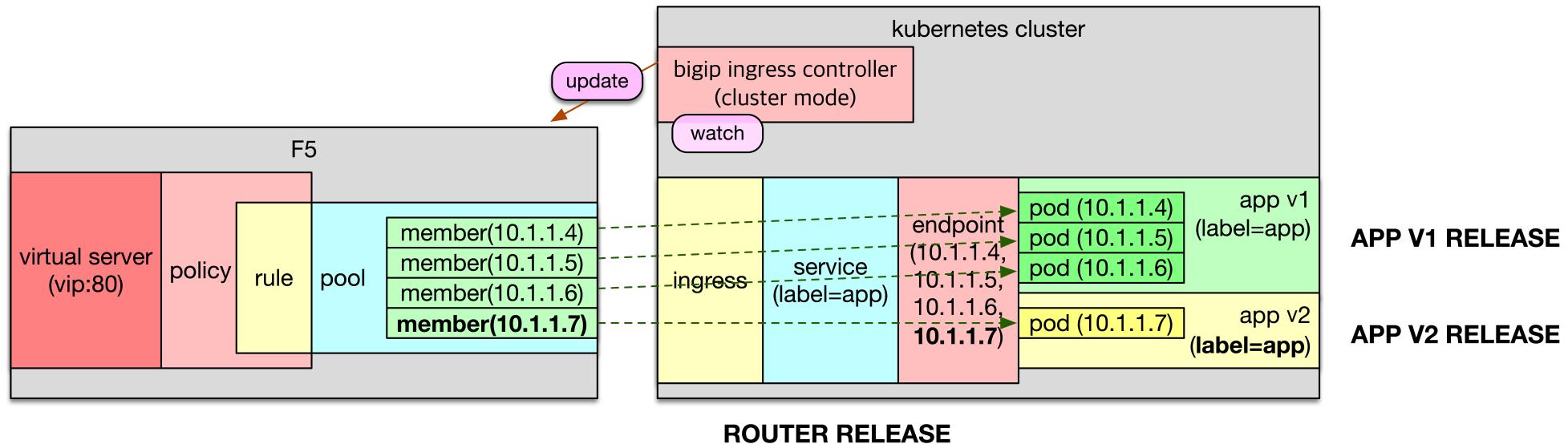
# Canary 배포 Step 2

## Canary Deploy Pipeline



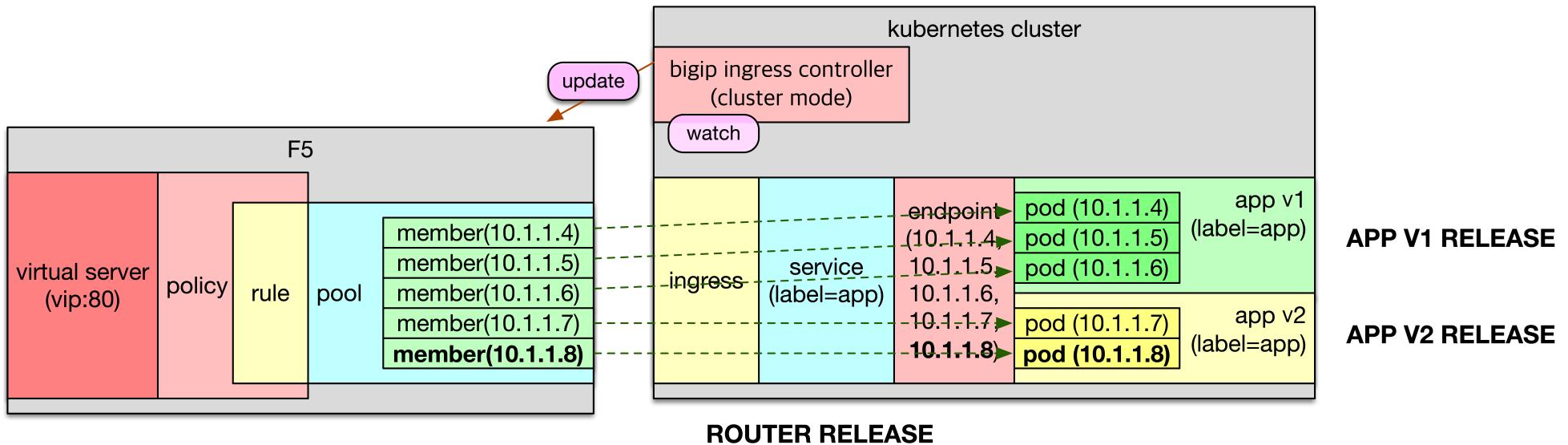
# Canary 배포 Step 3

## Canary Confirm Pipeline



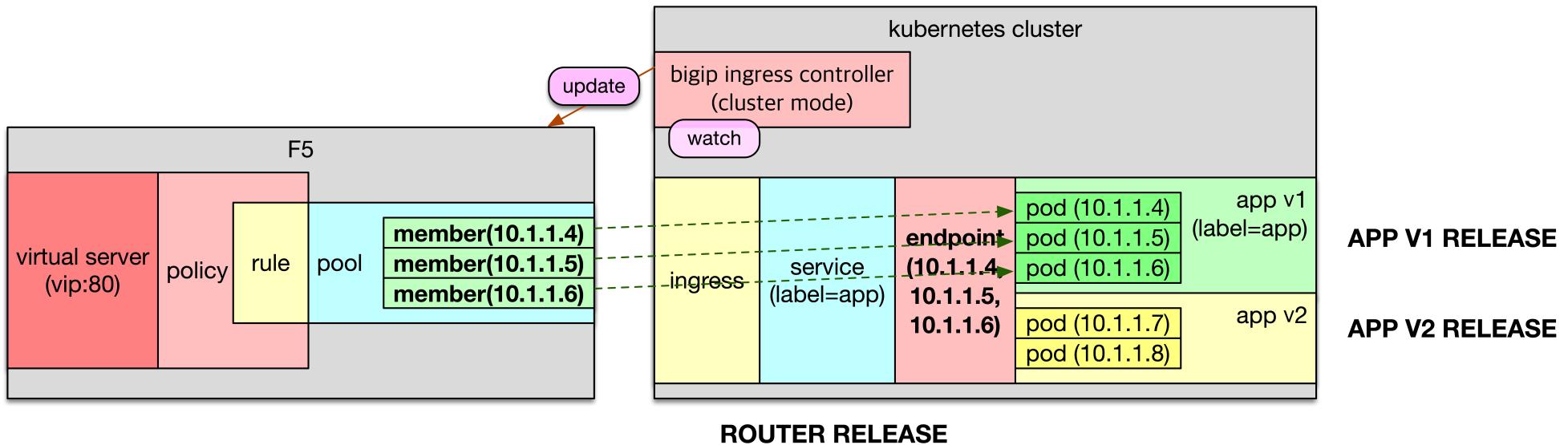
# Canary 배포 Step 4

## Canary Scale Pipeline



# Canary 배포 Rollback

## Canary Rollback Pipeline



## Memory 이슈 패치 사례

- 이미 배포된 릴리즈의 이슈 발견
- 패치 후 스테이징 테스트 완료
- **Canary Deploy**
- **Canary Rollback**
- **Green Deploy**
- **Green Confirm**
- **Blue Delete**

- 더 고민해보고 싶은거...
  - Service Mesh를 통한 보다 유연한 트래픽 제어...
  - 트래픽 비율 제어를 통한 canary 트래픽 제어...
  - 차트를 나누지 않고 통합 관리하는 설계...

## Thanks To

---

- 엑스투 이정민 소장님

# The End

---

- 감사합니다!