Kubernetes Gateway API: Getting Started

Ingress Essentials

Nigel Brown
Freelance Technical Author

@n_brownuk | @nigelb@fosstodon.org | windsock.io

Course Outline

What we'll cover in the course:

- Ingress abstraction options
- How the Gateway API facilitates ingress
- Define API objects to route ingress traffic
- Migrate from Ingress API to Gateway API

Ingress

Ingress traffic pertains to all network traffic and data that comes from outside a local network and is expected to land on a specific location within it.

Techslang, What is Ingress Traffic?, https://bit.ly/3R0hSdv

Ingress Traffic

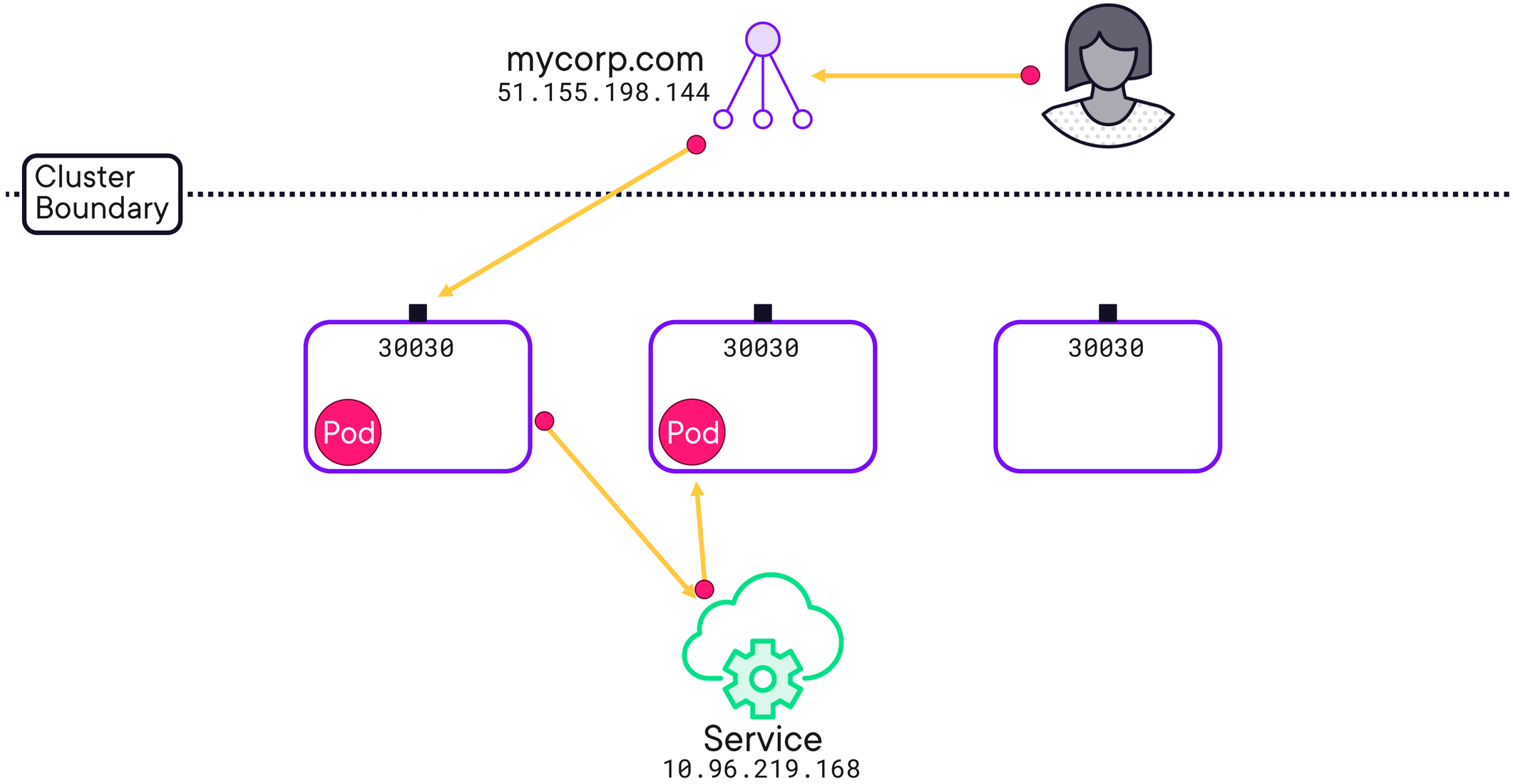
How is traffic that originates outside a Kubernetes cluster, routed to backend app services running in the cluster?

Service

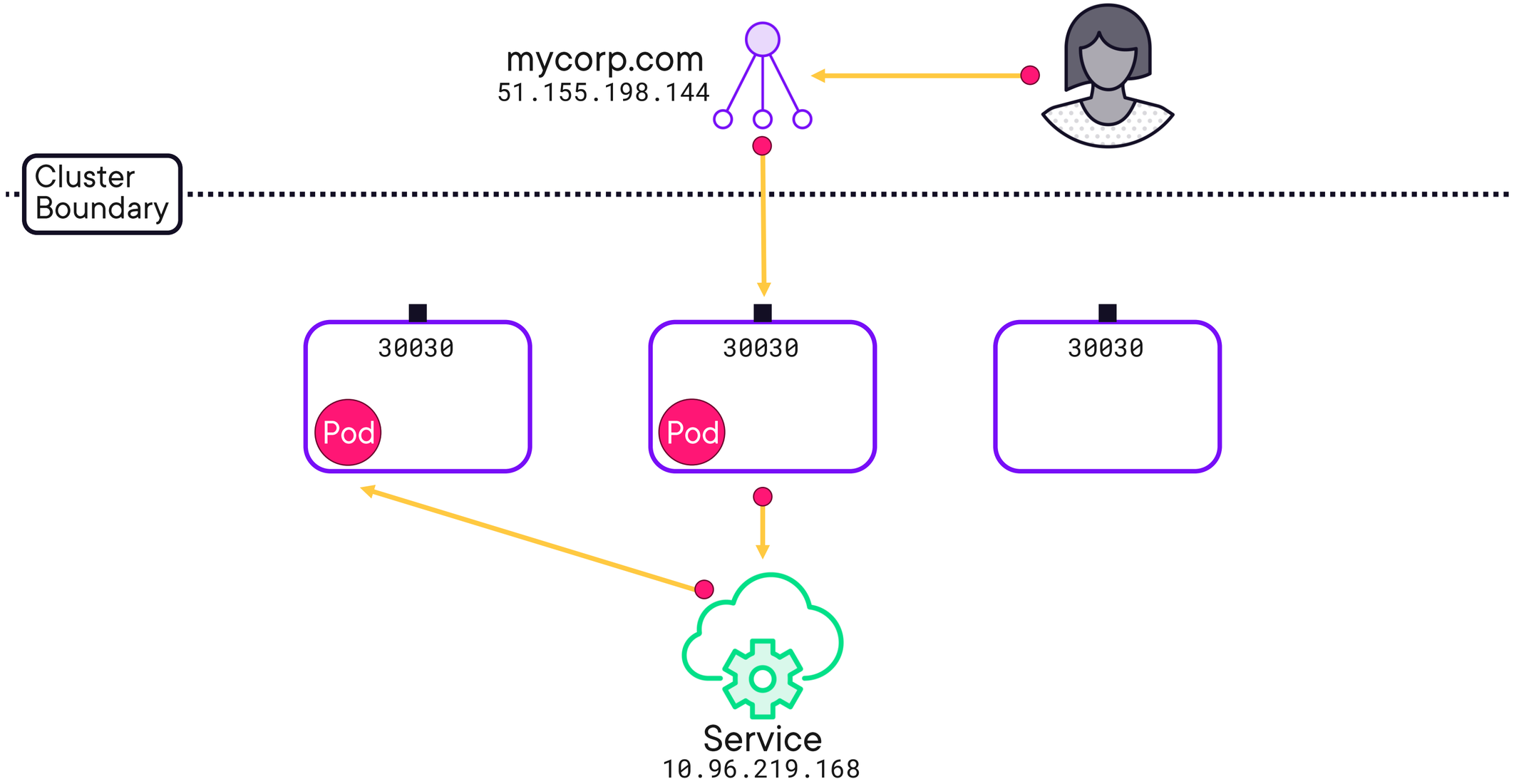
A method for exposing a network application that is running as one or more Pods in your cluster.

Glossary | Kubernetes, https://bit.ly/3SUAE8W

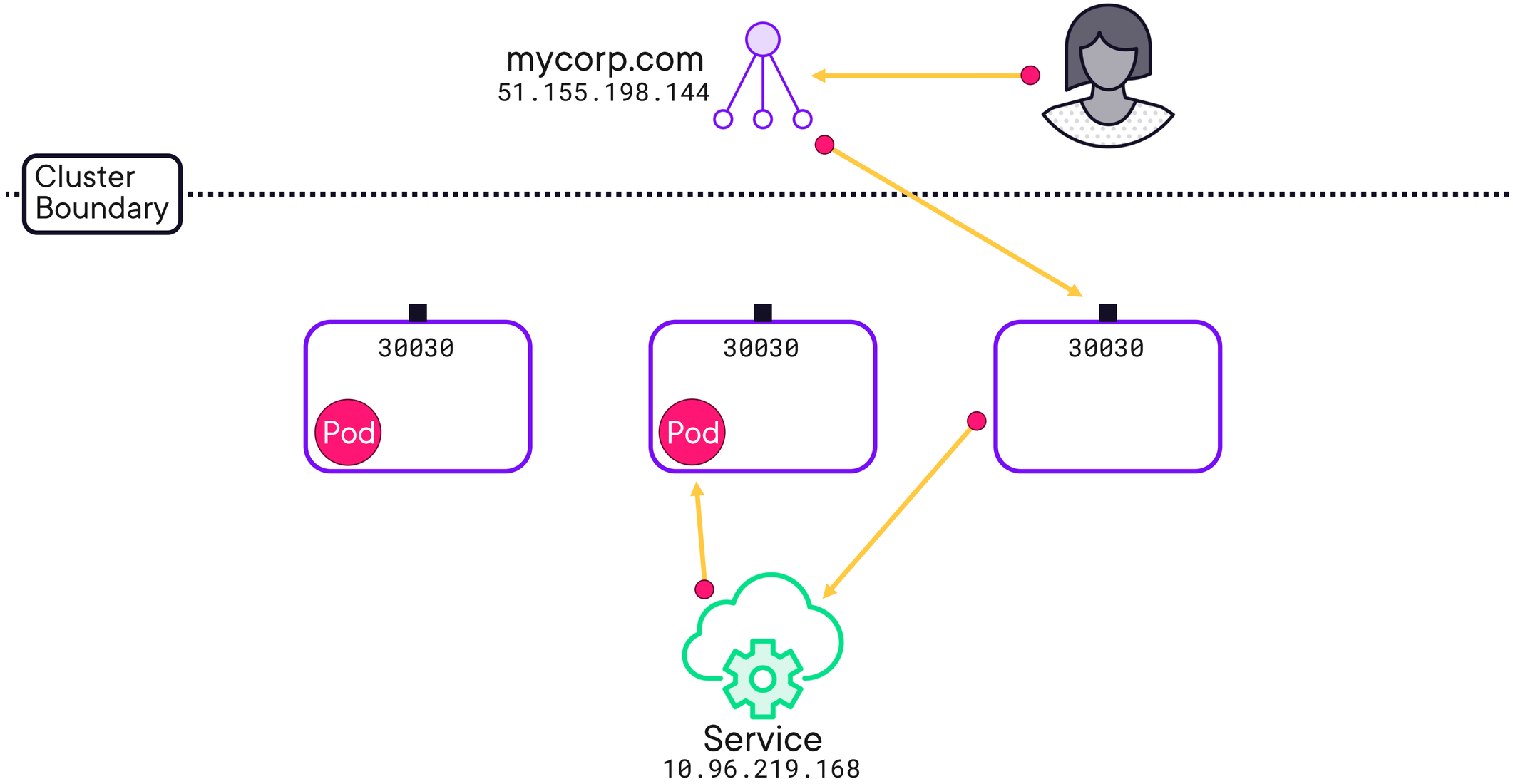
LoadBalancer Service



LoadBalancer Service



LoadBalancer Service



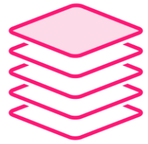
Ingress

An API object that manages external access to the services in a cluster, typically HTTP.

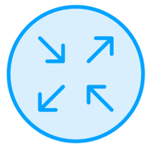
Glossary | Kubernetes, https://bit.ly/3GIPYDU

Ingress API Features

Load balances traffic across multiple backend services from a single IP address

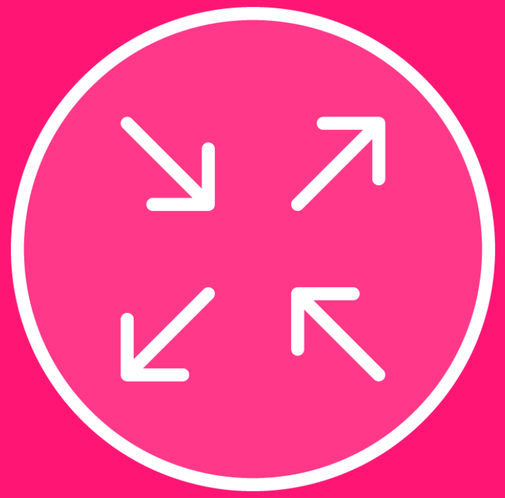


Manages traffic at the application layer of the network stack (HTTP/S)



Routes ingress traffic to backend services based on request host header and path

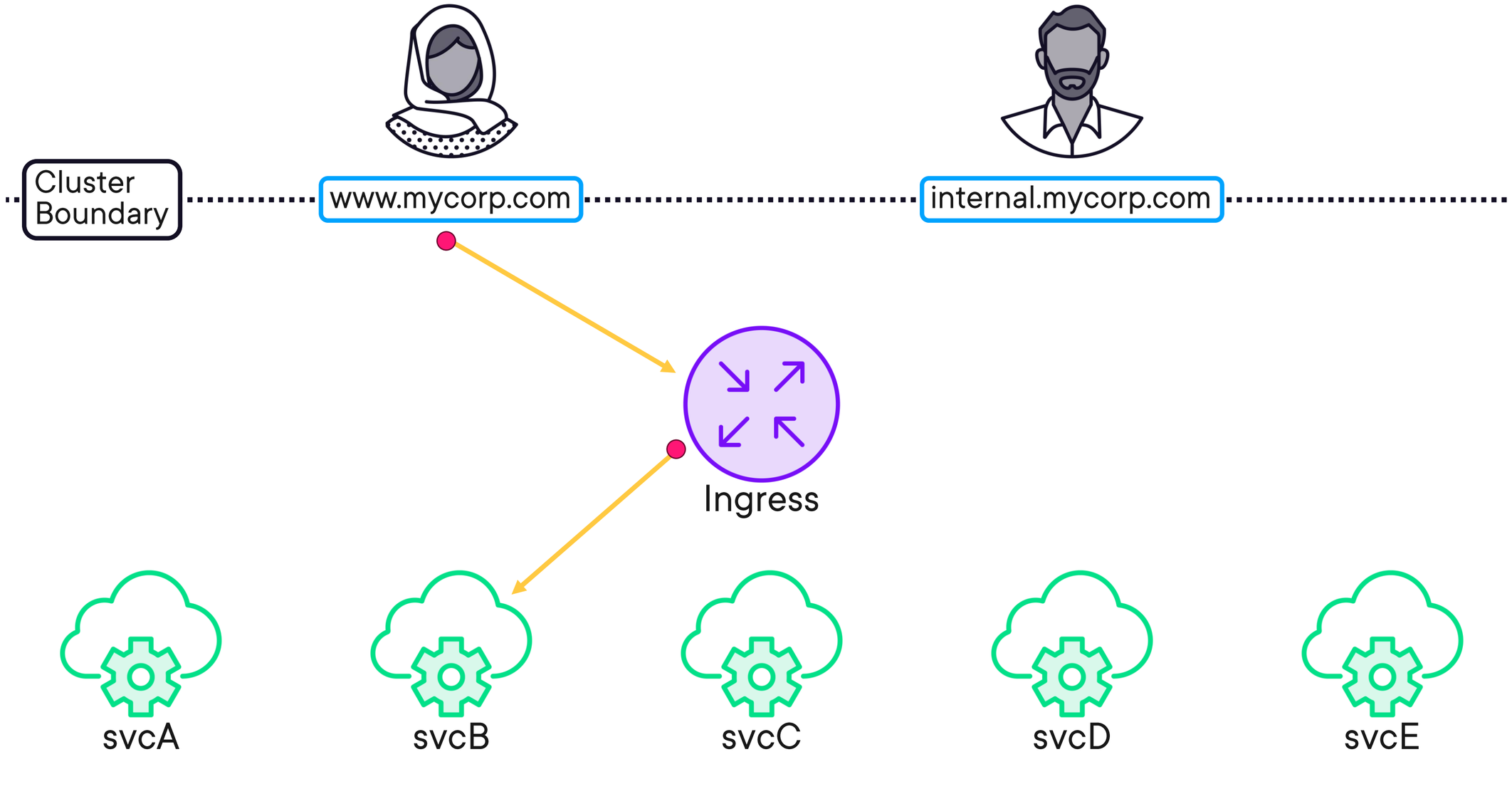
Requires the deployment of an ‘out-of-tree’ ingress controller in the cluster



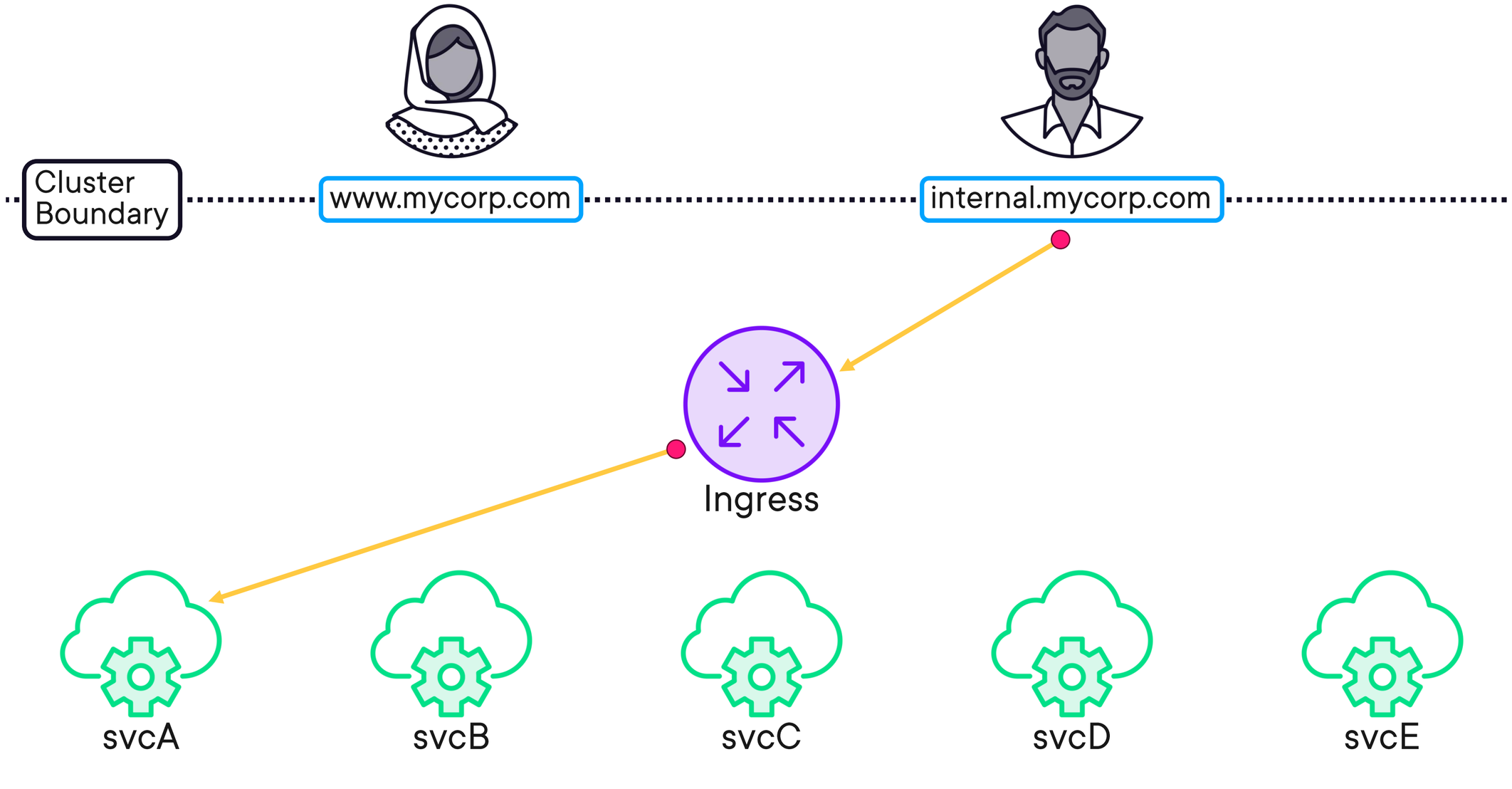
Ingress Controllers

Ingress controllers implement the Ingress API, but use existing proxy software tools to perform routing services.

Name-based Virtual Hosts



Name-based Virtual Hosts



Ingress Definition

ingress.yaml

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: host-rule-ingress
spec:
  rules:
    - host: mycorp.com
      http:
        paths:
          - backend:
              service:
                name: blog
                port:
                  number: 80
            pathType: ImplementationSpecific
```

Ingress Definition

ingress.yaml

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: host-rule-ingress
spec:
  rules:
    - host: mycorp.com
      http:
        paths:
          - backend:
              service:
                name: blog
                port:
                  number: 80
            pathType: ImplementationSpecific
```

Annotations

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: path-rule-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
    - host: mycorp.com
      http:
        paths:
          - path: /blog
            backend:
              service:
                name: blog
                port:
                  number: 80
            pathType: Prefix
```

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: web-app-ingress
  annotations:
    nginx.ingress.kubernetes.io/affinity: "cookie"
    nginx.ingress.kubernetes.io/session-cookie-name: "http-route"
    nginx.ingress.kubernetes.io/session-cookie-expires: "172800"
    nginx.ingress.kubernetes.io/session-cookie-max-age: "172800"
spec:

  <snip>
```

Multiple Annotations

It may be necessary to define multiple annotations to access proxy features

Service API

Advantages	VS	Disadvantages
Simple and straightforward		Requires an external IP per service
Automatically provisions an external load balancer		Expensive when there are multiple services to expose
		Limited traffic routing capabilities

Ingress API

Advantages	VS	Disadvantages
HTTP(S) routing based on content		Terse and inexpressive API
Caters for multiplicity of routes		Lack of formal means of extensibility
Implemented by best of breed proxies		Annotations not portable between ingress controllers
		Only caters for HTTP protocol
		Doesn’t allow for separation of concerns

Up Next:

Introducing the Gateway API