



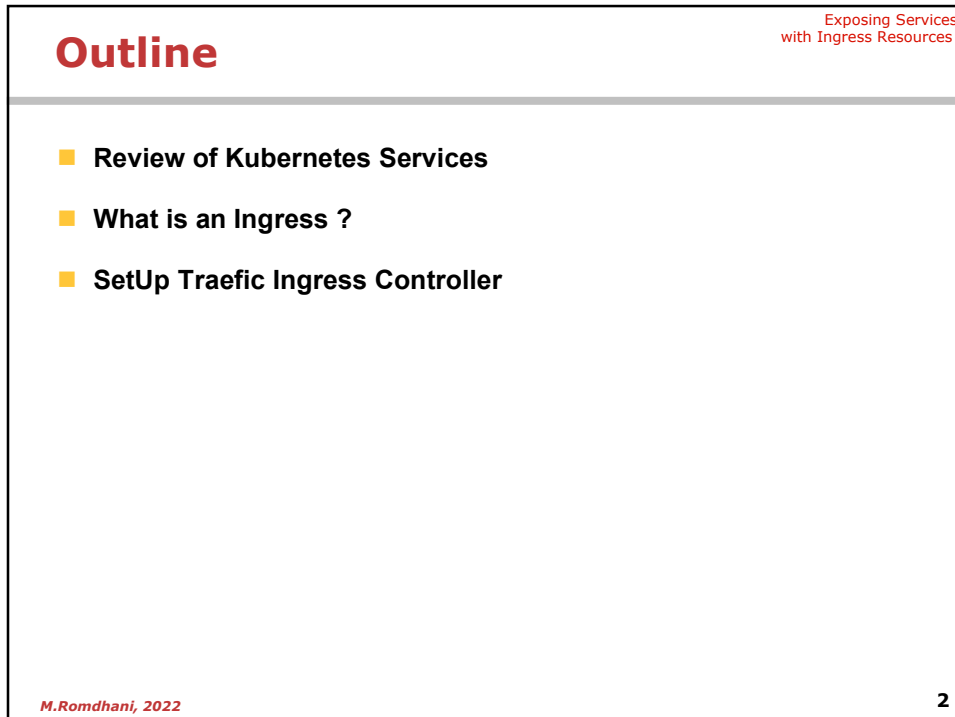
The slide features a purple header and footer bar. The header bar contains a small collage of images on the left. The main content area is white. At the top center is the Kubernetes logo (a blue shield with a white ship's wheel) and the text "Unit 5" in red. Below this is the main title "Exposing Services with Ingress Resources" in a large, bold, red font. In the bottom right corner, there is a logo for "Business Training" consisting of three blue icons (a circle, a square, and a triangle) above the text "Business Training". The footer bar also contains a small collage of images on the left.

Unit 5

# Exposing Services with Ingress Resources

Business Training

1



The slide has a white background. At the top left is the word "Outline" in a large, bold, red font. At the top right, in a smaller red font, is the text "Exposing Services with Ingress Resources". Below the title is a horizontal grey line. Underneath the line is a list of three items, each preceded by a yellow square bullet point. At the bottom left, in a small red font, is the text "M.Romdhani, 2022". At the bottom right is the number "2" in a bold black font.

## Outline

Exposing Services with Ingress Resources

- Review of Kubernetes Services
- What is an Ingress ?
- SetUp Traefic Ingress Controller

M.Romdhani, 2022

2

2

## Review of Kubernetes Services

3

### Service Types

Exposing Services  
with Ingress Resources

#### 1. ClusterIP (default)

- A virtual IP address is allocated for the service
- This IP address is reachable only from within the cluster (nodes and pods)
- Perfect for internal communication, within the cluster

#### 2. NodePort

- NodePort services extend the ClusterIP service.
- Exposes a port on every node's IP.
- Port can either be statically defined, or dynamically taken from a range between 30000-32767.

#### 3. LoadBalancer

- LoadBalancer services extend NodePort
- Works in conjunction with an external system to map a cluster external IP to the exposed service (typically a cloud load balancer, e.g. ELB on AWS, GLB on GCE ...)

M.Romdhani, 2022

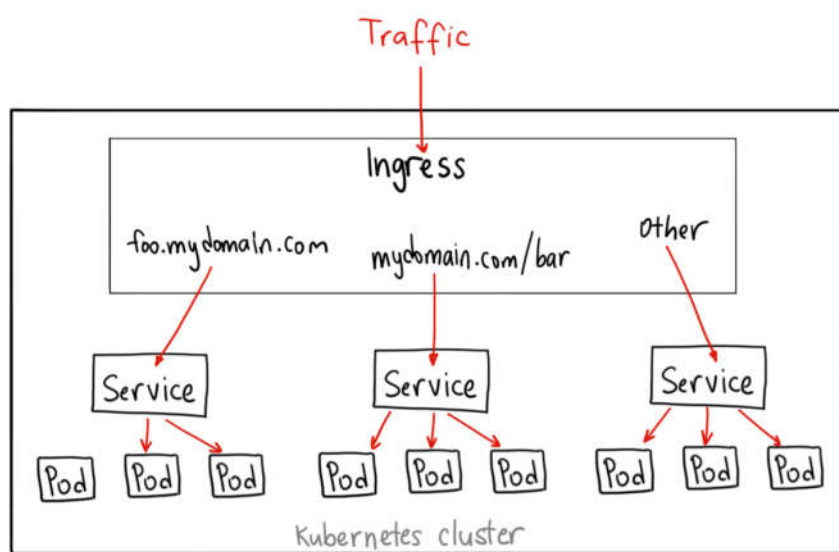
4

4

## What is an Ingress ?

5

### Ingress

Exposing Services  
with Ingress Resources

M.Romdhani, 2022

6

6

## The Ingress API Object

Exposing Services  
with Ingress Resources

- **An Ingress is an API object that manages external access to the services in a cluster**

- Provides load balancing, SSL termination and name/path-based virtual hosting
- Gives services externally-reachable URLs

- **They are specifically for HTTP services(not TCP or UDP)**

- **They can also handle TLS certificates, URL rewriting ...**

```
# Path based routing Example
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple-fanout-example
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: service1
          servicePort: 4200
      - path: /bar
        backend:
          serviceName: service2
          servicePort: 8080
```

M.Romdhani, 2022

7

7

## Ingress Controller

Exposing Services  
with Ingress Resources

- **The Ingress manifest doesn't actually do anything on its own; you must deploy an [Ingress Controller](#) into your cluster to watch for these declarations and act upon them.**

- **Ingress controllers are pods, just like any other application, so they're part of the cluster and can see other pods. They're built using reverse proxies that have been active in the market for years.**

- So, you have your choice of an **HAProxy**, **traefik**, **NGINX** Ingress Controller, and so on. The underlying proxy gives it Layer 7 routing and load balancing capabilities.

- **Being inside the cluster themselves, Ingress Controllers are susceptible to the same walled-in jail as other Kubernetes pods.**

- You need to expose them to the outside via a Service with a type of either NodePort or LoadBalancer.
- However, now you have a single endpoint that all traffic goes through: one Service connected to one Ingress Controller, which, in turn, is connected to many internal pods.
- The controller, having the ability to inspect HTTP requests, directs a client to the correct pod based on characteristics it finds, such as the URL path or the domain name.

M.Romdhani, 2022

8

8

## Set up Traefic Ingress controller

9

### Traefic Ingress Controller

Exposing Services  
with Ingress Resources

■ **Traefik is a modern HTTP reverse proxy and load balancer that makes deploying microservices easy.**

- Traefik integrates with your existing infrastructure components (Docker, Swarm mode, Kubernetes, Amazon ECS, ...) and configures itself automatically and dynamically. Pointing Traefik at your orchestrator should be the only configuration step you need.



■ **Features:**

- Continuous update of configuration (no restarts),
- Support for multiple load balancing algorithms,
- Web UI, metrics export,
- Support for various protocols, REST API, canary releases and so on.
- The support for Let's Encrypt certificates right out of the box is another nice feature.

M.Romdhani, 2022

10

10

## Setup Traefik

Exposing Services  
with Ingress Resources

[<https://doc.traefik.io/traefik/getting-started/install-traefik/>]

- Add Traefik's chart repository to Helm:

```
helm repo add traefik https://helm.traefik.io/traefik
```

- You can update the chart repository by running:

```
helm repo update
```

- And install it with the helm command line:

```
helm install traefik traefik/traefik
```

- Exposing the Traefik dashboard

- This HelmChart does not expose the Traefik dashboard by default, for security concerns. Thus, there are multiple ways to expose the dashboard. For instance, the dashboard access could be achieved through a port-forward :

```
kubect1 port-forward $(kubect1 get pods --selector  
"app.kubernetes.io/name=traefik" --output=name) 9000:9000
```

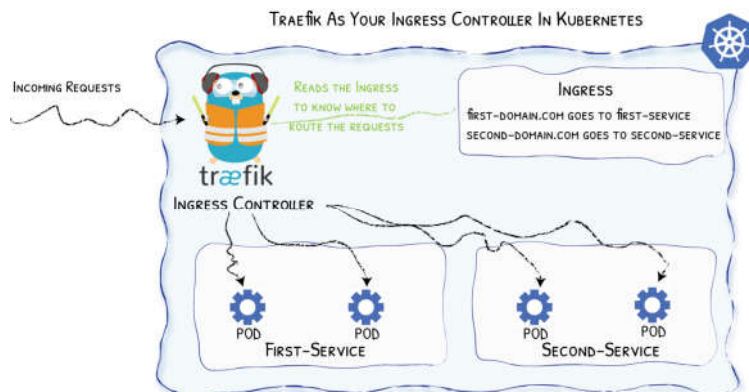
M.Romdhani, 2022

11

11

## How Traefik works ?

Exposing Services  
with Ingress Resources



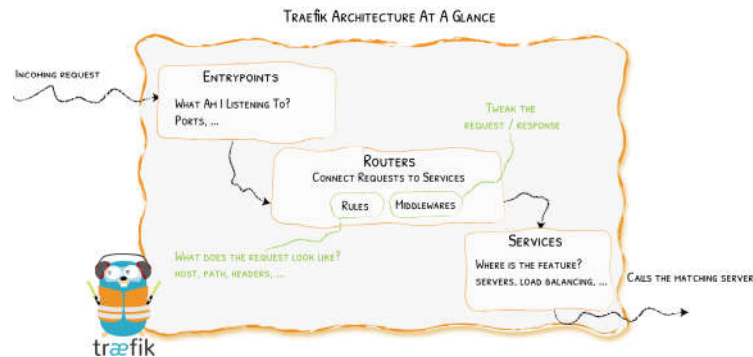
M.Romdhani, 2022

12

12

## Traefik 2 Architecture

Exposing Services  
with Ingress Resources



- **Providers** discover the services that live on your infrastructure (their IP, health, ...)
- **Entrypoints** listen for incoming traffic (ports, ...)
- **Routers** analyse the requests (host, path, headers, SSL, ...)
- **Services** forward the request to your services (load balancing, ...)
- **Middlewares** may update the request or make decisions based on the request (authentication, rate limiting, headers, ...)

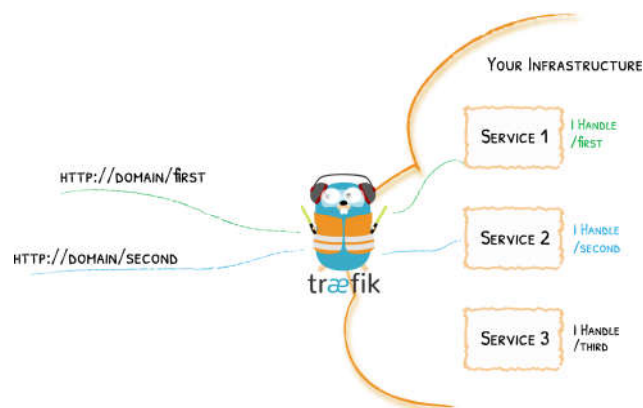
M.Romdhani, 2022

13

13

## Path-based Routing

Exposing Services  
with Ingress Resources



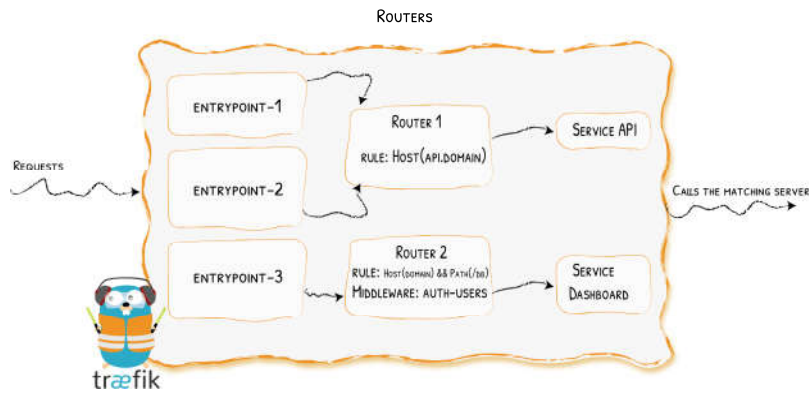
M.Romdhani, 2022

14

14

## Host-based Routing

Exposing Services  
with Ingress Resources



M.Romdhani, 2022

15

15

## The New Ingress v1 syntax

Exposing Services  
with Ingress Resources

### V1beta1 syntax

```
apiVersion:
networking.k8s.io/v1beta1
kind: Ingress
metadata:
  name: minimal-ingress-v1beta1
spec:
  rules:
    - http:
        paths:
          - path: /testpath
            pathType: Prefix
            backend:
              serviceName: test
              servicePort: 80
```

### V1 Syntax

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: minimal-ingress-v1
spec:
  rules:
    - http:
        paths:
          - path: /testpath
            pathType: Prefix
            backend:
              service:
                name: test
                port:
                  number: 80
```

M.Romdhani, 2022

16

16



## Traefik IngressRoute CRD object

Exposing Services  
with Ingress Resources

### Standard Ingress Syntax

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress-path-based
  annotations:
    // Stripped
spec:
  rules:
    - http:
        paths:
          - path: /apple
            pathType: Prefix
            backend:
              service:
                name: apple-service
                port:
                  number: 5678
          - path: /banana
            pathType: Prefix
            backend:
              service:
                name: banana-service
                port:
                  number: 5678
```

### Traefik IngressRoute Syntax

```
apiVersion: traefik.containo.us/v1alpha1
kind: IngressRoute
metadata:
  name: example-ingress-path-based-crd
spec:
  entryPoints:
    - web
  routes:
    - match: PathPrefix(`/apple`)
      kind: Rule
      services:
        - name: apple-service
          port: 5678
          strategy: RoundRobin
          weight: 10
      middlewares:
        - name: do-stripprefix
    - match: PathPrefix(`/banana`)
      kind: Rule
      services:
        - name: banana-service
          port: 5678
          strategy: RoundRobin
          weight: 10
```

M.Romdhani, 2022

17

17

## Ingress Definition for the BookStoreApp

Exposing Services  
with Ingress Resources

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: bookstore-ingress
  annotations:
    kubernetes.io/ingress.class: traefik
spec:
  rules:
    - host: bookstore.minikube
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: frontend
                port:
                  number: 80
          - path: /books
            pathType: Prefix
            backend:
              service:
                name: backend-api
                port:
                  number: 80
          - path: /categories
            pathType: Prefix
            backend:
              service:
                name: backend-api
                port:
                  number: 80
```

M.Romdhani, 2022

18

18