



# **MICRO-SERVICES SAMPLE PROJECT**

*Components Document & Tools Usage*

**VIJAYENDRA MUDIGAL**

<http://vijayendra.mudigal.com/>

# TABLE OF CONTENTS

---

API Gateway.....	3
Service Registration and Discovery .....	3
Monitoring AND Vizualization .....	5
Centralized Logging using elk .....	6

# API GATEWAY

---

Netflix Zuul acts as the API Gateway for accessing the micro services behind the gateway. It also acts as the load balancer and the reverse proxy server which routes the request to the respective service. Microservice's stay behind reverse proxy server and needs to be consumed via api gateway. The Docker profile of the api-gateway micro service runs on port 8080 and can be accessed by <http://localhost:8080> .

Configuration done in API Gateway for Routing:

```
zuul:
  ignoredServices: '*'
  routes:
    one:
      path: /service-one/**
      serviceId: Service-One
    two:
      path: /service-two/**
      serviceId: Service-Two
    three:
      path: /service-three/**
      serviceId: Service-Three
```

## SERVICE REGISTRATION AND DISCOVERY

---

Registration and discovery is taken care by the HashiCorp's Consul. Every micro service that has to be routed from api-gateway has to register itself to Consul so that it could be discovered by the API Gateway. Once the service is registered to the consul, consul checks for the health of the service by sending a heartbeat for the health check path and health check interval that has been registered with Consul.

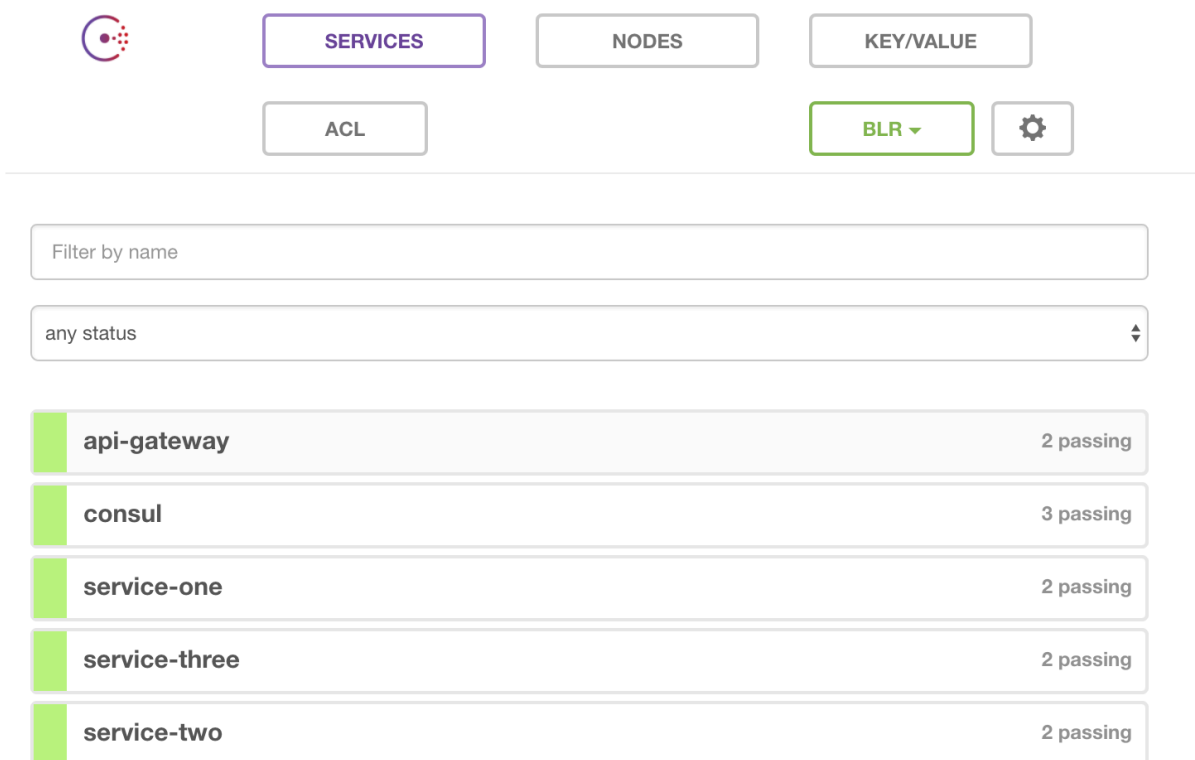
## Configuration done in micro services to register to Consul:

```
management:
  contextPath: /manage

spring:
  application.name: service-one
  cloud:
    consul:
      host: consul
      port: 8500
      discovery:
        hostname: service-one
        instanceId:${spring.application.name}:${spring.application.i
nstance_id:${random.value}}
        healthCheckPath: ${management.contextPath}/health
        healthCheckInterval: 15s
```

## Tools:

Consul Management console: <http://localhost:8500/ui/>



The screenshot shows the Consul Management console interface. At the top, there is a navigation bar with the Consul logo and several tabs: SERVICES (highlighted in purple), NODES, KEY/VALUE, ACL, BLR (highlighted in green with a dropdown arrow), and a settings gear icon. Below the navigation bar, there is a search bar labeled "Filter by name" and a status filter dropdown menu currently set to "any status". The main content area displays a list of registered services, each with a green status indicator, the service name, and the number of instances passing health checks.

Service Name	Status
api-gateway	2 passing
consul	3 passing
service-one	2 passing
service-three	2 passing
service-two	2 passing

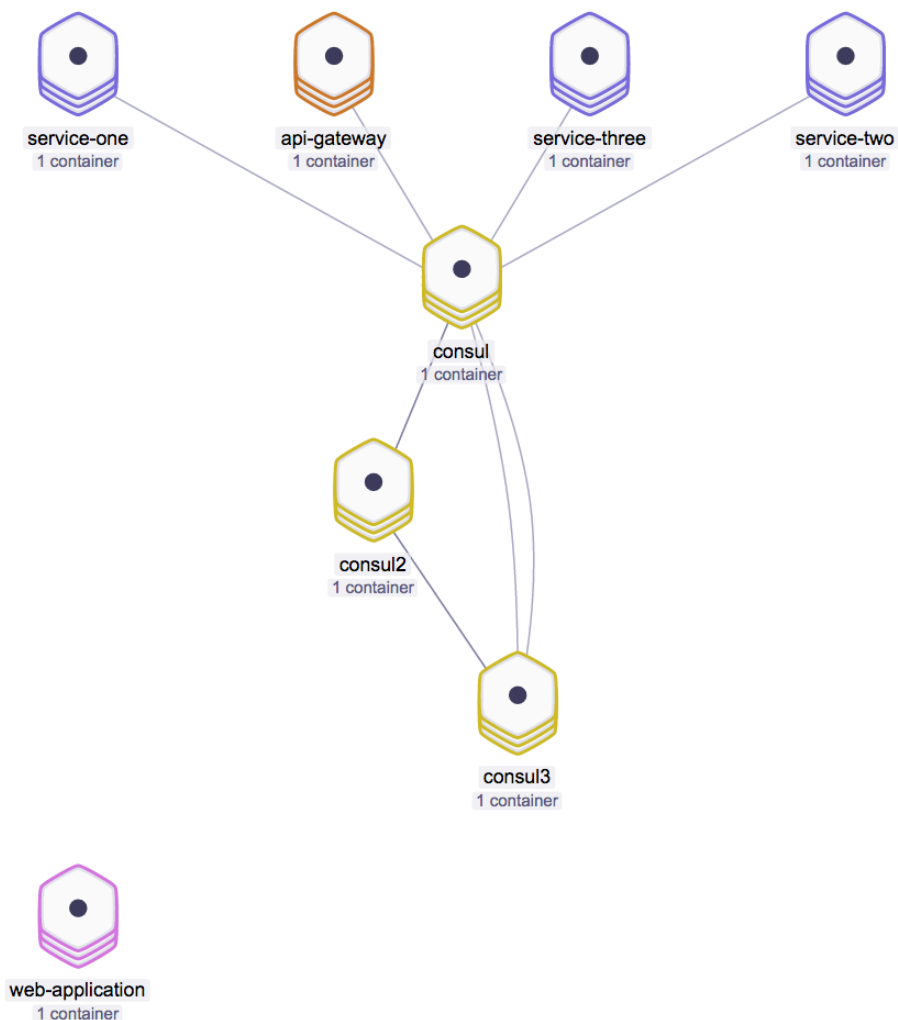
# MONITORING AND VIZUALIZATION

---

Monitoring, visualisation & management of the container in docker is done by weave scope.

## Tools:

Weavescope Management Console: <http://localhost:4040/>



# CENTRALIZED LOGGING USING ELK

Our services use Logback to create application logs and send the log data to the logging server (Logstash). Logstash formats the data and send it to the indexing server (Elasticsearch). The data stored in elasticsearch server can be beautifully visualized using Kibana.

## Tools:

Elasticsearch: [http://localhost:9200/\\_search?pretty](http://localhost:9200/_search?pretty)

Kibana: <http://localhost:5601/>

