

## Gateway Server Configuration

### Gateway Server Dependencies

- Gateway
- Config Client
- Eureka Discovery Client
- Actuators
- Devtools

1. Add Following properties in application.yml file –

```
spring.application.name="gateway"

spring.config.import = "optional:configserver:http://localhost:8071/"

spring.cloud.gateway.discovery.locator.enabled = false

management.endpoints.web.exposure.include = "*"

management.endpoint.gateway.enabled = true

management.info.env.enabled = true

info.app.name = "Gateway Server"

info.app.description = "Edge Server for Synergy Bank"

info.app.version = "1.0.0"
```

2. You can also include below properties; else these properties can be obtained from config server github repo –

```
server.port= 8072

eureka.instance.client.preferIpAddress = true

eureka.client.fetchRegistry = true

eureka.client.registerWithEureka = true

eureka.client.serviceUrl.defaultZone = http://localhost:8070/eureka/
```

3. Verify the Gateway Server running by visiting below URL-

<http://localhost:8072/actuator/gateway/routes>

4. Disable the additional routes available

```
spring.cloud.gateway.discovery.locator.enabled = true
```

5. Add Custom Routes for the application using **RouteLocatorBuilder**.

## Updating Docker Compose

1. Include health related probes in all other microservices.

```
management.health.readiness-state.enabled = true
```

```
management.health.liveness-state.enabled = true
```

```
management.endpoint.health.probes.enabled = true
```

2. Generate docker images for services

```
mvn compile jib:dockerBuild
```

3. Push generated images to docker hub-

```
docker image push docker.io/synergyhub/image_name:1.0.0
```

4. Add gatewayserver service in docker-compose.yml file
5. Add healthcheck properties to all other microservices, like configserver.
6. Gateway server service depends on – all the other services. Check for service healthy condition
7. After all the above mentioned steps, run the following command –

```
docker compose up
```