Today Topic
==========
1. StockProject(Using MS)
2. Api-Gateway(ZuulProxy Server)
3. Actuators
4. SpringBoot-AdminDashboard(To manages the services)

Topics Left
==========
Monday topics::
1. Circuit Breaker[hystrik-circuit-breaker]
2. MessageBroker[Apache-Kafka]

Tuesday topics::
3. RedisCache
4. ReactiveProgramming

Video part:: SpringSecurity and Junit with Mockito(will uploaded in dashboard)

refer::
Service-Registery-Eurekha-Server
Stock-Calcuation
Stock-Service
Stock-UI-APP

APIGateway
=========
It acts as entry point for all microservices in our project.
It is also called as "Edge Microservice".
If we don't have APIGateway, then we need to implement request filtering logic in every microservice,it increases the maintainence cost of the project.
Zuul Proxy we can use as "APIGateway",it is provided by spring-cloud-netflix library.
It is an open source gateway.
Apigee is a commercial APIGateway provided by google.

Steps to develop API Gateway
============================
1. Create spring boot application with below dependancies
    a. spring-boot-starter-web
    b. spring-cloud-netflix-zuul
    c. spring-cloud-netflix-eurekha-client

2. Configure @EnableZuulProxy and @EnableDiscoveryClient annotations at spring boot starter class
3. Configure API gateway with Eurekha server registration
4. Configure Routing to access backend services in application.properties/yml file
5. Configure ApplicationName and port no.

application.properties
======================
```
server.port= 4444
spring.application.name=STOCK-API-GATEWAY

zuul.prefix=/api

zuul.routes.price.path=/price/**
```

```
zuul.routes.price.service-id==STOCK-PRICE-SERVICE

zuul.routes.calc.path=/calc/**
zuul.routes.calc.service-id==STOCK-CALCUlATION-SERVICE


StockApplication using Microservice Architecture
================================================
 ServiceRegistry: http://localhost:8761
 STOCK-PRICE-SERVICE: http://localhost:1111/price/stockprice/{companyName}
 STOCK-CALC-SERVICE: http://localhost:2222/calc/calculate/{companyName}/{quantity}
 API-GATEWAY : http://localhost:4444/api/

URL's given to end user
=======================
  To get company price :
http://localhost:4444/api/price/price/stockPrice/{companyName}
  To get total cost    :
http://localhost:4444/api/calc/calc/calculate/{companyName}/{quantity}



Actuators
=========
 => We are developing our applications using springboot and those applications will
be deployed in servers.
 => After deploying our applications in server, those applciation will be used by
the clients(users).
 => when our application is running in production, It is very important to monitor
the application.
 => Monitoring the application corresponds to how the application is responding the
clients.

What is the meaning of Monitoring the application?
  a. HealthCheck
  b. BeansCheck[Gettign to know all the beans available for MS]
  c. configProps check[Getting to know all the END_POINTS]
  d. Headump
  e. Thread dump(information about threads)
  f. Http trace etc.....

=> To monitor our applications,Spring boot has provided actuators.
=> Actuators are used to provide "Production ready features" of our application.

Working with actuators
======================
1. use the following dependancies in our application

<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>

2. Actuators provided several predefined endpoints to monitor our application
      a. health
      b. info
      c. beans[To know all the beans loaded in our application]
      d. heapdump
```

e. Threaddump
f. shutdown(It is a special endpoints)
g. configProps
h. mappings[To know all the url patterns of our application]

3. To access actuator endpoints we should use /actuator as prefix(It is introduced from SpringBoot2.X)
eg: http://localhost:9999/actuator/health

4. In SpringBoot 1.x 'actuator' is optional in URL to access the endpoints.
5. In SpringBoot 2.x 'autuator' is compulsory in URL to access the endpoints.
6. By default 2 endpoints will be available(health,info)

Note: In order to include all the inbuilt endpoints
management.endpoints.web.exposure.include=*
management.endpoints.web.exposure.exclude=health,mappings,beans