Small autonomous services that work together.

Fault tolerant, scalability.

1. **Configuration Management:**

* Spring cloud config server (Keeps configuration in one place, that makes easy to maintain configuration for all micro-services)

1. **Dynamic Scale Up and Scale down:**

* Naming Server (Eureka)
* Ribbon (Client Side Load Balancing by using Naming server)
* Feign (Easier REST Clients)

1. **Visibility and Monitoring:**

* Zipkin distributed tracing (to trace request across multiple components)
* Netflix API gateway.

Spring cloud slouth will assign id to request across multiple components.

1. **Fault Tolerance:**

* Hystrix

To connect limits-server to spring-cloud-config-server we need to rename application.properties to bootstrap.properties then add

spring.cloud.config.uri=http://localhost:8888/

To connect spring-cloud-config-server to git repo add below property in application.properties.

spring.cloud.config.server.git.uri=file:\\E:\\GitRepo

if the active profile is dev and key is not available in dev properties file, then value will be picked up from default properties file.

Spring boot starter config client: is used connect the spring cloud configuration(config server).

Feign is used as rest client

Ribbon is used as load balancing. ( we are hard coding instance url’s)

If you use both Feign and Ribbon then only load balancing works.

Eureka Naming Server is used to dynamically increase and decrease instances.(we no need to hard code url’s)