29/07/24

Jenkins pipeline-script based- groovy and ui based

[Using Jenkins with Groovy scripts: A Practical Guide (with examples) - Sling Academy](https://www.slingacademy.com/article/using-jenkins-with-groovy-scripts-a-practical-guide-with-examples/)

Car manufacturing🡪

Pujshing your source code to repo🡪run quality gates(clean code)🡪testing all types of testing all types of reports🡪packaging🡪deploy/install🡪release

Jenkins

* Pipeline stage view
* -Dmaven.test.failure.ignore=true🡪even if it fails it builds

[CircuitBreaker (readme.io)](https://resilience4j.readme.io/docs/circuitbreaker)

Sure, here is a simplified explanation of the CircuitBreaker implementation:

### CircuitBreaker Basics

1. \*\*States\*\*:

- \*\*Normal States\*\*: CLOSED, OPEN, HALF\_OPEN.

- \*\*Special States\*\*: DISABLED, FORCED\_OPEN.

### Sliding Window

1. \*\*Types\*\*:

- \*\*Count-based\*\*: Tracks the last N calls.

- \*\*Time-based\*\*: Tracks calls within the last N seconds.

2. \*\*Count-based Sliding Window\*\*:

- Uses a circular array to store outcomes of N calls.

- Keeps a running total which updates with each new call and removes the oldest call's outcome.

- Snapshot retrieval is fast and memory usage is proportional to N.

3. \*\*Time-based Sliding Window\*\*:

- Uses a circular array of N buckets for each second.

- Each bucket aggregates outcomes for calls in that second.

- Snapshot retrieval is fast and memory usage is nearly constant.

### Aggregation Details

1. \*\*Data Stored\*\*:

- Number of failed calls.

- Number of slow calls.

- Total number of calls.

- Total duration of all calls.

### State Transitions

1. \*\*From CLOSED to OPEN\*\*:

- When the failure rate exceeds a threshold (e.g., >50% failed calls).

- When the slow call rate exceeds a threshold (e.g., >50% calls slower than a set duration).

2. \*\*From OPEN to HALF\_OPEN\*\*:

- After a wait period, allows a limited number of calls to test availability.

3. \*\*From HALF\_OPEN to OPEN or CLOSED\*\*:

- If failure or slow call rate exceeds threshold, goes back to OPEN.

- If below threshold, goes to CLOSED.

### Special States

1. \*\*DISABLED\*\*: Always allows access, no metrics recorded.

2. \*\*FORCED\_OPEN\*\*: Always denies access, no metrics recorded.

### Thread Safety

1. \*\*State Management\*\*:

- Uses atomic operations to manage state changes.

- Call recording and snapshot retrieval are synchronized.

2. \*\*Function Calls\*\*:

- Not synchronized to avoid performance bottlenecks.

- Multiple threads can call functions if the state is CLOSED.

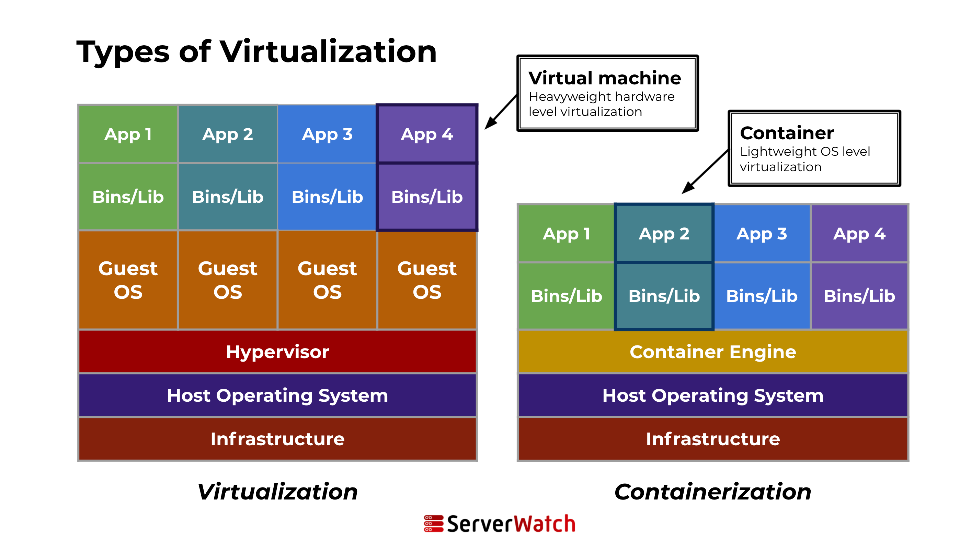
### Bulkhead Pattern

- To limit concurrent calls, use a Bulkhead pattern in combination with CircuitBreaker.\_

Client-rest template, feint, bean

* Docker is a platform designed to make it easier to create, deploy, and run applications by using containers. Containers allow developers to package an application with all the parts it needs, such as libraries and other dependencies, and ship it all out as one package. Here are the key points about Docker:

DOCKER

* [Ansible Documentation — Ansible Community Documentation](https://docs.ansible.com/ansible/latest/index.html)
* [Chef and Puppet | Chef](https://www.chef.io/puppet)
* 
* Container will have direct access to the kernel of the underlying OS.
* When we instantiate an image it becomes a container.(we can replicate the container)
  + There are many spring containers among which two are very important.
  + Bean factories defined by the **org.springframework.beans.factory.BeanFactory** interface are the simplest of containers, providing basic support for dependency injection (DI).  This BeanFactory is the super interface of all the spring containers.
  + Application contexts defined by the **org.springframework.context.ApplicationContext** interface build on the notion of a bean factory by providing application framework services.
  + These **BeanFactory** and **ApplicationContext**  are java interfaces, but in spring world these are called as containers.
  + [What is spring container –Types of Spring Container - java4coding](https://www.java4coding.com/contents/spring/spring-container)

0. create a new folder and switch to that folder.

1. Create a dockerfile

2. define the libs/bins

3. Build an image with dockerfile

4. rum the image file to instantiate the container-runtime instance of an image which based on docker file specification

Dockerfile:

FROM openjdk:11

RUN mkdir/app

COPY Hello.java/app

WORKDIR /app

RUN Javadoc Hello.java

CMD[“java”,”Hello”]

Hello.java:

{Write a java code}

docker build .

docker build -t myhello.v1 .

docker run myhello.v1

* docker ps -a
  + for showing the running of containers