

Deploy Kapua

Pre-requisites:

- 64-bit architecture
- Java VM Version 8
- Maven 3.6.0+
- Docker Version 1.2+

Cloudlab experiment setup

- Profile: [small-lan](#)
- Speed: 10Gpbs
- Machine Type: m510 or d710
- Chose max disk space before creating the experiment
- Volume management
 - This application requires more volume than the default 16GB allocated
 - Run the following command to use the mounted storage **/mydata**
 - **sudo chown -Rf \$USER /mydata**

Install JDK

`sudo apt install openjdk-8-jdk -y`

Docker setup

- Install Docker
- <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-18-04>
- Change the docker image download location to **/mydata**
 - Create a configuration file:
`/etc/docker/daemon.json`
 - Add this content in that daemon.json file

```
{  
  "data-root": "/mydata/docker-data"  
}
```
 - Restart docker
`sudo systemctl restart docker`

Maven setup

- Install maven
- **sudo apt install maven -y**
- Change default maven dependency download location
- Open the maven settings file
 - `sudo vi /usr/share/maven/conf/settings.xml`

- Update the local repository as
`<localRepository>/mydata/.m2/repository</localRepository>`

Repository

- <https://github.com/ssmtariq/kapua-test>

Branch

- `feature/development_1.0.0`

Build the application

- `mvn clean install -Pdev -DskipTests=true`
 Or
`mvn clean install -Pdev -DskipTests=true -Dcheckstyle.skip`

Build docker containers

- `mvn -f assembly -Pdocker`

Note: If there is any CheckStyle-related failure you can use the following command with maven to skip CheckStyle validation

`-Dcheckstyle.skip`

Run the integration test cases to execute the inefficient code block

```
mvn verify -Dcommons.db.schema=kapuadb -Dcommons.settings.hotswap=true
-Dbroker.host=localhost
-Dgroups='!org.eclipse.kapua.qa.markers.junit.JUnitTests'
-Dcucumber.options="--tags ~@rest"
```

You may also save the output in a log file to check the latency

```
mvn verify -Dcommons.db.schema=kapuadb -Dcommons.settings.hotswap=true
-Dbroker.host=localhost
-Dgroups='!org.eclipse.kapua.qa.markers.junit.JUnitTests'
-Dcucumber.options="--tags ~@rest" -Dcheckstyle.skip >
result/original-code-es-64Kbps-10000req.txt
```

Inefficiency detected

RunDatastoreTransportI9nTest (L#22) -> Datastore.feature (L#129)((L#177)) ->
 DatastoreSteps.java (L#700) -> MessageStoreServiceImpl.java (L#223) ->
 MessageStoreFacade.java (L#224)

Remove all local images

```
docker rmi -f $(docker images -aq)
```

Throttle Network Bandwidth using Wondershaper

- Install wondershaper
sudo apt install wondershaper
- Clone wondershaper and change the directory
git clone <https://github.com/magnific0/wondershaper.git>
cd wondershaper
- Get the network interface name
ifconfig
- Limit upload and download bandwidth
sudo ./wondershaper -a <adapter> -d <down_speed_kbps> -u <upload_speed_kbps>
i.e.
sudo ./wondershaper -a eno1 -d 64 -u 64
sudo wondershaper eno1 64 64
- Clear the download upload limits
sudo wondershaper -ca eno1
Or
sudo wondershaper clear eno1

Check Elasticsearch Indices

- http://128.110.217.143:9200/_cat/indices?v
- Check aliases: http://128.110.217.78:9200/_aliases?pretty=true

Allow wildcard Index Delete

Update el

Allow from kibana or elasticsearch console using the following request

After installing Kibana go to DevTools and from the console execute this command

```
PUT _cluster/settings
{"persistent": {"action.destructive_requires_name": "false"}}
```

Important Changed Files:

- org.eclipse.kapua.service.datastore.steps.DatastoreSteps.java
- org.eclipse.kapua.service.datastore.internal.MessageStoreFacadeTest.java
- org.eclipse.kapua.service.datastore.internal.MessageStoreFacade.java
- org.eclipse.kapua.service.elasticsearch.client.rest.RestElasticsearchClient.java
- org.eclipse.kapua.service.datastore.internal.client.DatastoreElasticsearchClientConfiguration.java (Change db to remote/local)
- qa/integration/src/test/resources/features/datastore/Datastore.feature (The feature to run)

To replace a whole line start with text use the following regex in IntelliJ Idea

`^text.*`

Find newline with white space

`^\s*$`

Remove all Empty lines from text file

`grep -v -x '[:blank:]*' file`