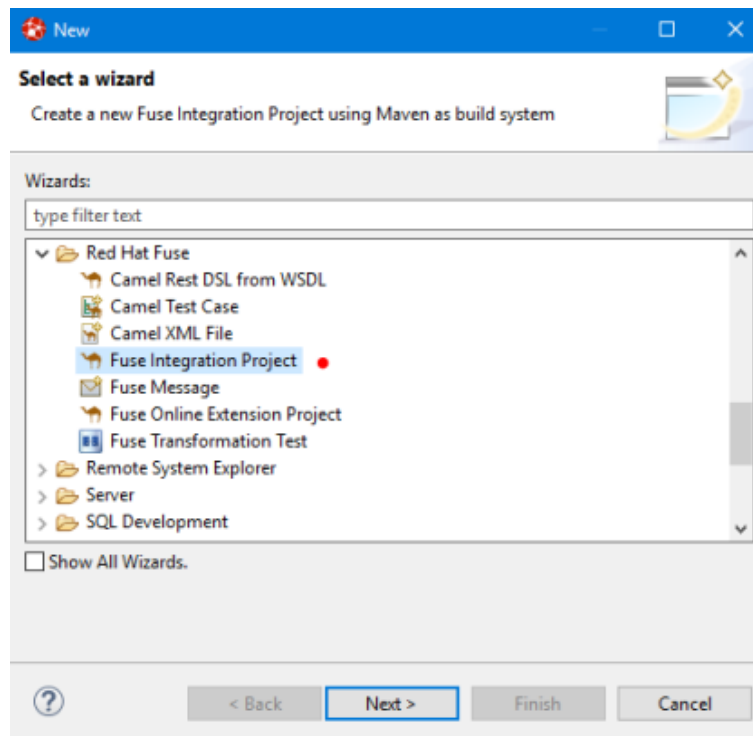


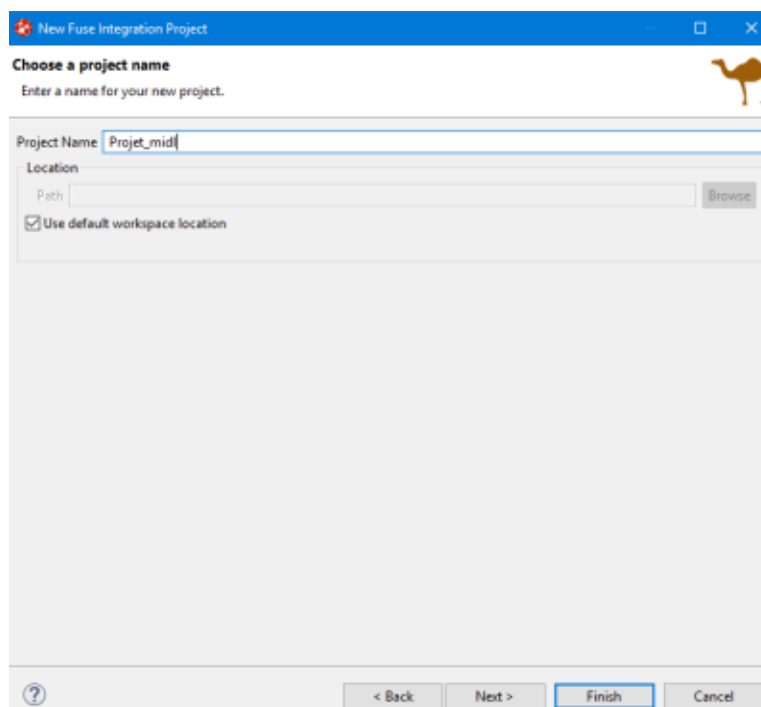
# Rapport

## Fuse REST API sur OpenShift

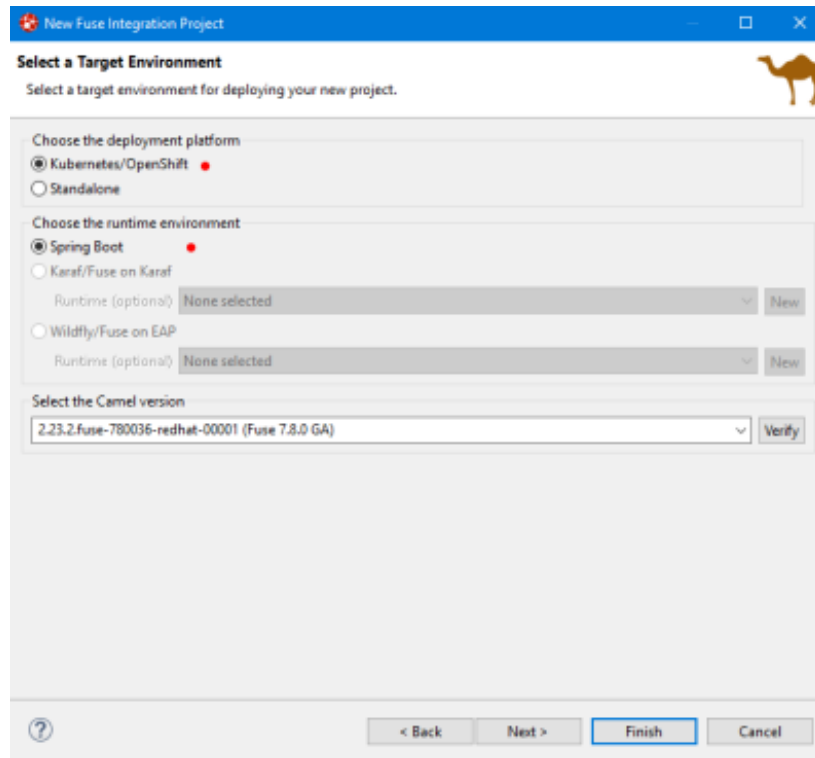
### Création de projet



Nom de projet :

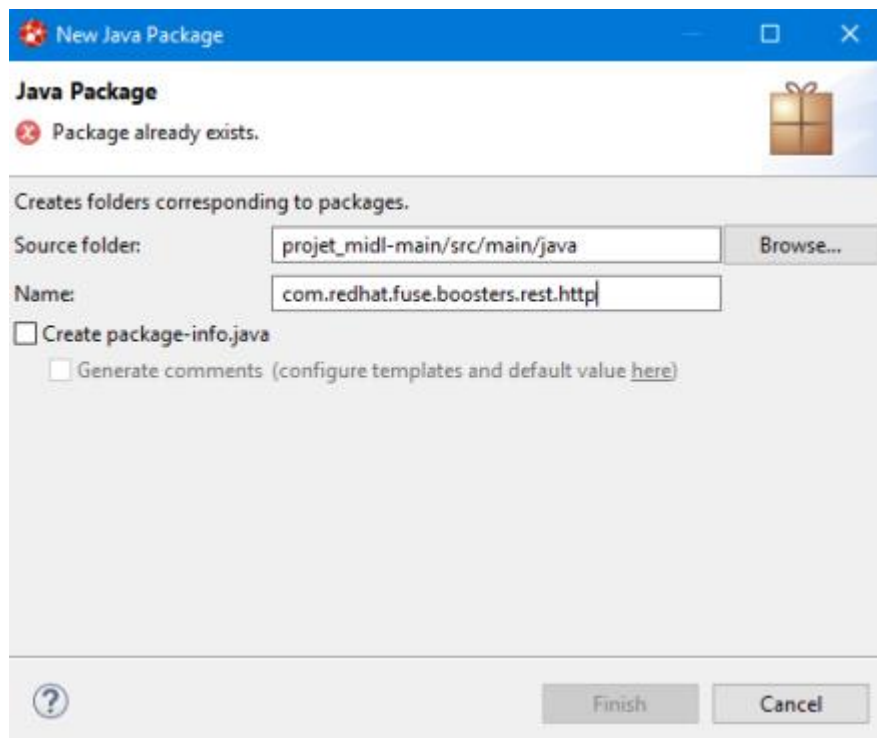


Choix de la plateforme de déploiement, l'environnement de runtime ainsi que la version de fuse :



The image shows a 'New Fuse Integration Project' dialog box. It has a blue title bar with the text 'New Fuse Integration Project'. Below the title bar, the main heading is 'Select a Target Environment' with a subtitle 'Select a target environment for deploying your new project.' and a camel icon. The dialog is divided into three sections: 'Choose the deployment platform' with radio buttons for 'Kubernetes/OpenShift' (selected) and 'Standalone'; 'Choose the runtime environment' with radio buttons for 'Spring Boot' (selected), 'Karaf/Fuse on Karaf', and 'Wildfly/Fuse on EAP', each with a 'Runtime (optional)' dropdown menu set to 'None selected' and a 'New' button; and 'Select the Camel version' with a dropdown menu showing '2.23.2.fuse-780036-redhat-00001 (Fuse 7.8.0 GA)' and a 'Verify' button. At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

Création de package dans le répertoire : **src/main/java**



The image shows a 'New Java Package' dialog box. It has a blue title bar with the text 'New Java Package'. Below the title bar, the main heading is 'Java Package' with a subtitle 'Package already exists.' and a gift icon. The dialog contains the text 'Creates folders corresponding to packages.' followed by two input fields: 'Source folder:' with the value 'projet\_midl-main/src/main/java' and a 'Browse...' button, and 'Name:' with the value 'com.redhat.fuse.boosters.rest.http'. There are two checkboxes: 'Create package-info.java' (unchecked) and 'Generate comments (configure templates and default value [here](#))' (unchecked). At the bottom, there are navigation buttons: a question mark icon, 'Finish' (disabled), and 'Cancel'.

Classe Application responsable a l'exécution du projet :

```
package com.redhat.fuse.boosters.rest.http;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Application {

    /**
     * Main method to start the application.
     */
    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }

}
```

Création de la classe CamelRouter pour la configuration et l'implémentation du  
Api router :

```
package com.redhat.fuse.boosters.rest.http;

import org.apache.camel.builder.RouteBuilder;
import org.apache.camel.model.rest.RestBindingMode;
import org.springframework.stereotype.Component;

@Component
public class CamelRouter extends RouteBuilder {

    @Override
    public void configure() throws Exception {

        restConfiguration()
            .apiContextPath("/api-doc")
            .apiProperty("api.title", "Taher's REST API")
            .apiProperty("api.version", "1.8")
            .apiProperty("cors", "true")
            .apiProperty("base.path", "camel/")
            .apiProperty("api.path", "/")
            .apiProperty("host", "")
            .apiContextRouteId("doc-api")
            .component("servlet")
            .bindingMode(RestBindingMode.json);

        rest("/Maamouri").description("Greeting to {name}")
            .get("/{name}").outType(Greetings.class)
            .route().routeId("greeting-api")
            .to("direct:greetingsImpl");

        from("direct:greetingsImpl").description("API route implementation")
            .streamCaching()
            .to("bean:greetingsService?method=getGreetings");
    }

}
```

Création de la class *Greetings* :

```
package com.redhat.fuse.boosters.rest.http;

public class Greetings {

    private String greetings;

    public Greetings() {
    }

    public Greetings(String greetings) {
        this.greetings = greetings;
    }

    public String getGreetings() {
        return greetings;
    }

    public void setGreetings(String greetings) {
        this.greetings = greetings;
    }
}
```

Création de l'interface *GreetingsService* :

```
package com.redhat.fuse.boosters.rest.http;

public interface GreetingsService {

    /**
     * @return a string greetings
     */
    Greetings getGreetings( String name);
}
```

Implémentation de l'interface *GreetingsService* avec la récupération du paramètre *nom* à l'aide de la méthode **GET**:

```
package com.redhat.fuse.boosters.rest.http;

import org.apache.camel.Header;
import org.springframework.stereotype.Service;

@Service("greetingsService")
public class GreetingsServiceImpl implements GreetingsService {

    private static final String THE_GREETINGS = "username: | ";

    @Override
    public Greetings getGreetings(@Header("name") String name ) {
        return new Greetings( THE_GREETINGS + name );
    }
}
```

Création du fichier **indexrootmapper** pour la redirection du client vers la page web :

```
package com.redhat.fuse.boosters.rest.http;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration
@ComponentScan
public class IndexRootMapper implements WebMvcConfigurer {

    public void addViewControllers(ViewControllerRegistry registry) {
        registry.addViewController("/").setViewName("forward:/rest-http-index.html");
    }
}
```

Choix du port pour le lancement de l'application dans le fichier :

**application.properties**

```
server.port=9999

endpoints.enabled = false
endpoints.health.enabled = true
management.health.defaults.enabled=false

camel.health.enabled=false
```

Création du fichier **deployment.yml** dans **src/main/fabric8** pour la configuration du déploiement sur openshift

```
spec:
  template:
    spec:
      containers:
      -
        resources:
          requests:
            cpu: "0.2"
            memory: 256Mi
          limits:
            cpu: "1.0"
            memory: 256Mi
        env:
        - name: SPRING_APPLICATION_JSON
          value: '{"server":{"undertow":{"io-threads":1, "worker-threads":2 }}}'
```

Création de l'interface du client (GUI) avec les fichiers **rest-http-index.html** et **style.css**

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="./style.css">
9 </head>
10
11 <body>
12   <div class="login-box">
13     <h2>API test :</h2>
14     <form>
15
16       <div class="user-box">
17         <input type="text" name="username" required="" id="id1">
18         <label>Username</label>
19       </div>
20
21       <a href="/test.html" onclick="location.href='/camel/maamouri/' + document.getElementById('id1').value ; return false;">
22         <span></span>
23         <span></span>
24         <span></span>
25         <span></span>
26         <span></span> Submit
27       </a>
28     </form>
29   </div>
30 </body>
31
32 </html>
```

L'ajout des « dependencies » nécessaires dans le fichier **Pom.xml** :

```
<dependency>
  <groupId>io.fabric8</groupId>
  <artifactId>openshift-client</artifactId>
</dependency>
```

```

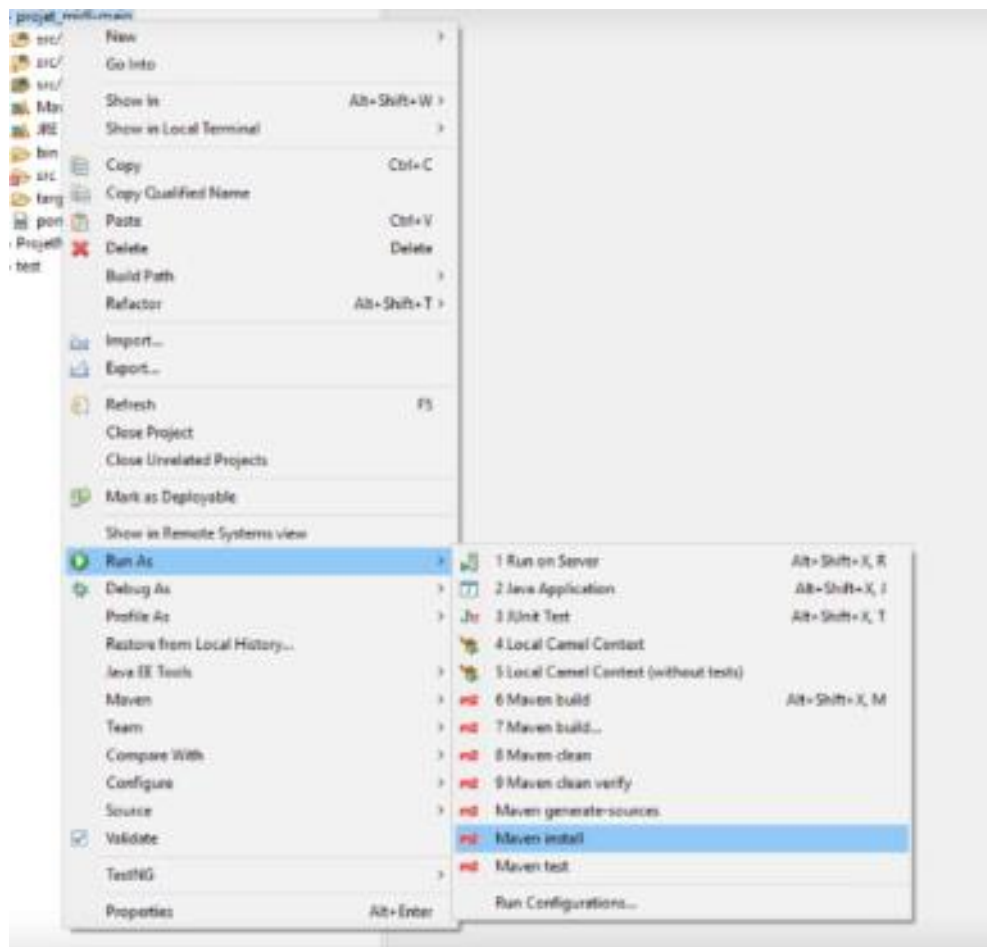
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-spring-boot-starter</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-http-starter</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-servlet-starter</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-jackson-starter</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-swagger-java-starter</artifactId>
</dependency>
<dependency>

```

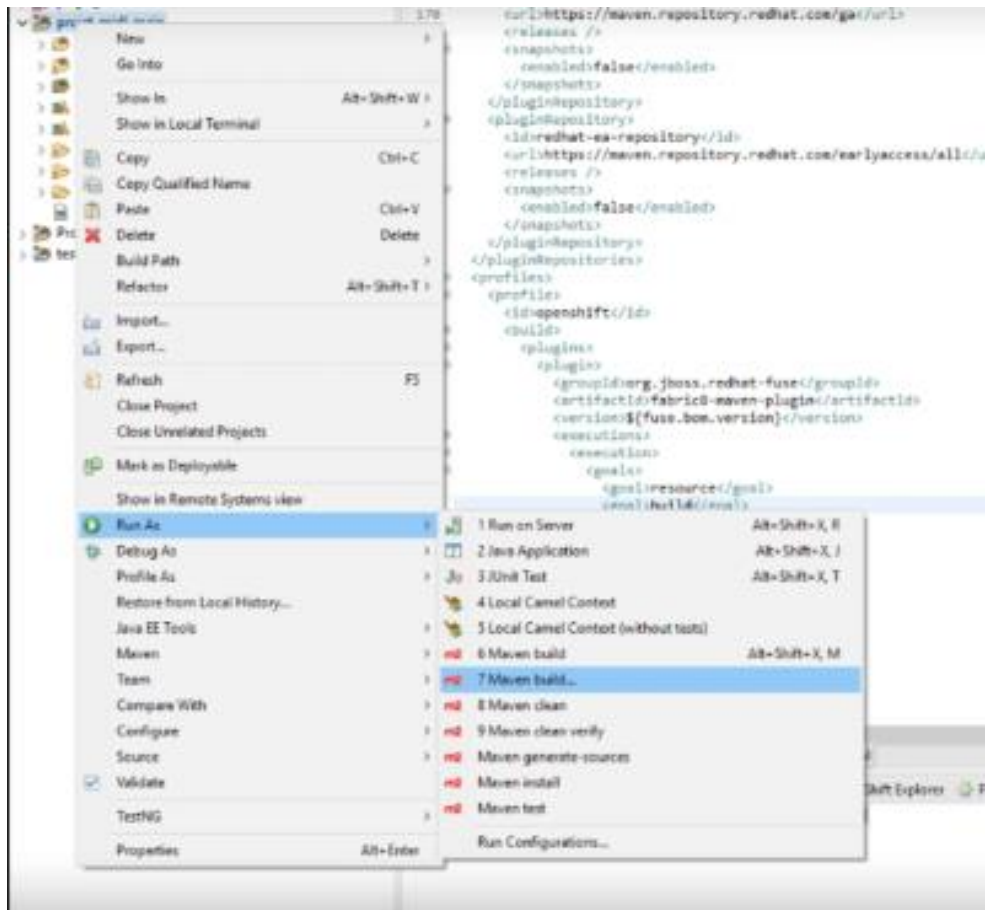
## Lancement de projet sur Fuse

On doit suivre ces étapes :

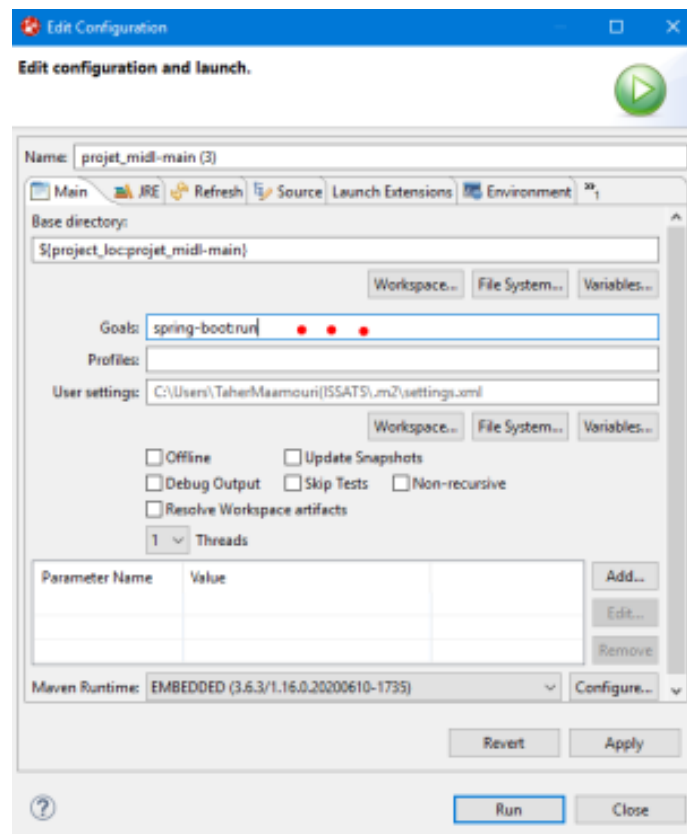
### Maven install



## Maven Build



On doit taper la commande suivante : « **spring-boot:run** »





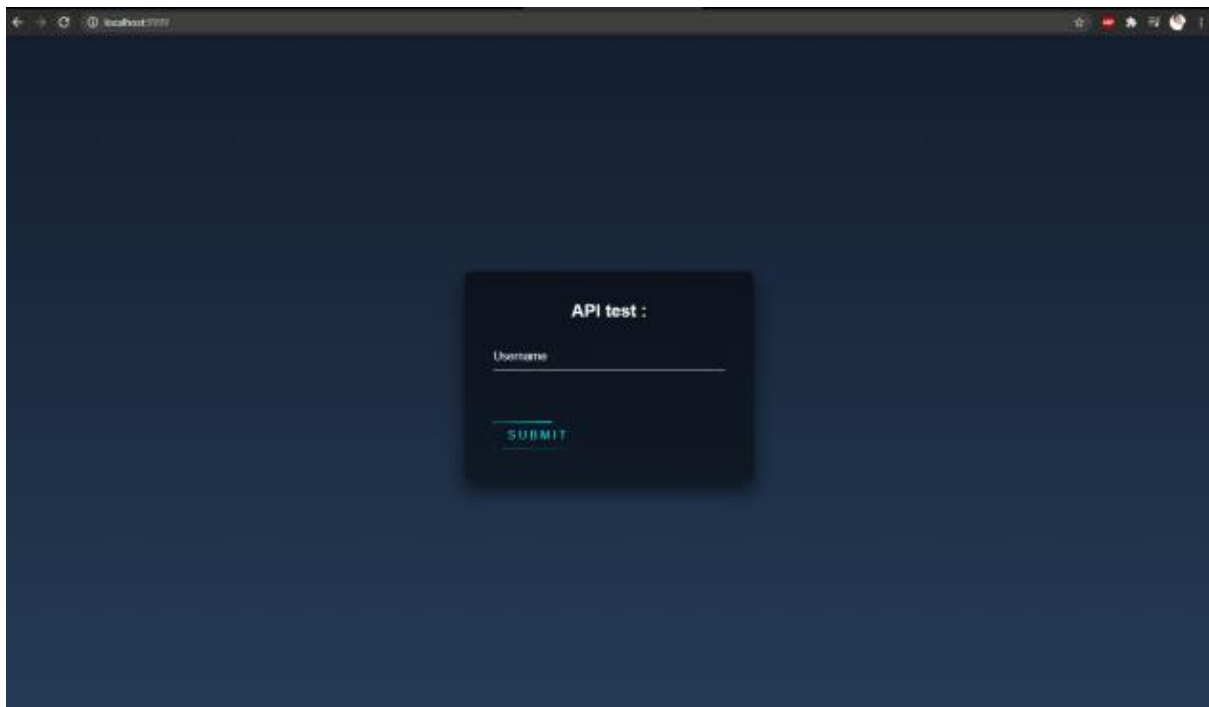
## L'application a été bien lancé sur le port 9999

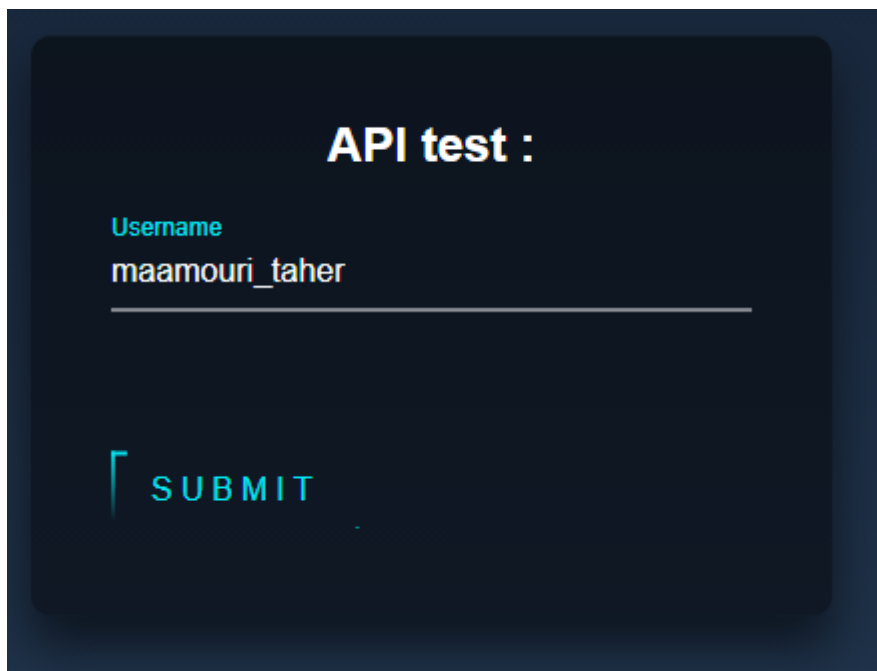
```
Message View | Servers | Console | Progress | OpenShift Explorer | Palette | Remote Systems | Console
projet_rndl-main [Maven Build] C:\Program Files\Java\jdk-8.0.771\bin\java.exe [29 Jan 2021, 20:47:53]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ fuse-demo-app ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\TaherHammouri\I55AT5\workspaceCR\projet_rndl-main\src\test\resources
[INFO] --- maven-compiler-plugin:3.7.0:testCompile (default-testCompile) @ fuse-demo-app ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- spring-boot-maven-plugin:1.5.9.fuse-ub2-760020-redhat-00001:run (default-cli) < test-compile @ fuse-demo-app ---
[INFO] --- spring-boot-maven-plugin:1.5.9.fuse-ub2-760020-redhat-00001:run (default-cli) @ fuse-demo-app ---

Spring Boot (v2.1.6.RELEASE)

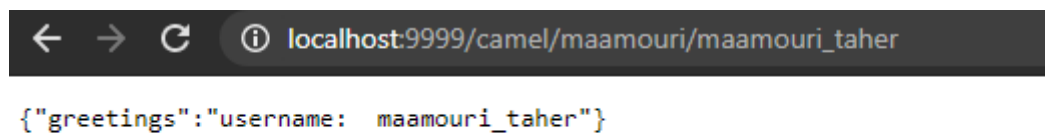
2021-01-29 20:48:00.255 INFO 5076 ---
2021-01-29 20:48:00.263 INFO 5076 ---
2021-01-29 20:48:00.744 INFO 5076 ---
2021-01-29 20:48:03.094 WARN 5076 ---
2021-01-29 20:48:03.117 INFO 5076 ---
2021-01-29 20:48:03.118 INFO 5076 ---
2021-01-29 20:48:03.924 INFO 5076 ---
2021-01-29 20:48:04.594 INFO 5076 ---
2021-01-29 20:48:04.878 INFO 5076 ---
2021-01-29 20:48:04.878 INFO 5076 ---
2021-01-29 20:48:04.895 INFO 5076 ---
2021-01-29 20:48:04.898 INFO 5076 ---
2021-01-29 20:48:05.219 INFO 5076 ---
2021-01-29 20:48:05.349 INFO 5076 ---
2021-01-29 20:48:05.351 INFO 5076 ---
2021-01-29 20:48:05.357 INFO 5076 ---
2021-01-29 20:48:05.359 INFO 5076 ---
2021-01-29 20:48:05.361 INFO 5076 ---
2021-01-29 20:48:05.363 INFO 5076 ---
2021-01-29 20:48:05.365 INFO 5076 ---
2021-01-29 20:48:05.373 INFO 5076 ---
2021-01-29 20:48:05.393 INFO 5076 ---
2021-01-29 20:48:05.413 INFO 5076 ---
2021-01-29 20:48:05.935 INFO 5076 ---
2021-01-29 20:48:05.948 INFO 5076 ---
2021-01-29 20:48:16.552 INFO 5076 ---
2021-01-29 20:48:16.553 INFO 5076 ---
2021-01-29 20:48:16.567 INFO 5076 ---

main| c.r.fuse.boosters.rest.http.Application : Starting Application on DESKTOP-V8IU35K with PID 5076 (C:\Users\TaherHammouri\I55AT5\w
main| c.r.fuse.boosters.rest.http.Application : No active profile set, falling back to default profiles: default
main| o.s.c.i.converter.DefaultTypeConverter : Type converters loaded (core: 195, classpath: 10)
main| io.undertow.websockets.jsr : UO020010: Buffer pool was not set on WebsocketDeploymentInfo, the default pool will be
main| io.undertow.servlet : Initializing Spring embedded WebApplicationContext
main| o.s.web.context.ContextLoader : Root WebApplicationContext: initialization completed in 2792 ms
main| o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
main| o.s.b.a.e.web.EndpointLinksResolver : Exposing 2 endpoint(s) beneath base path '/actuator'
main| o.s.camel.spring.boot.RoutesCollector : Loading additional Camel XML routes from classpath:camel/*.xml
main| o.s.camel.spring.boot.RoutesCollector : Loading additional Camel XML rests from classpath:camel-rest/*.xml
main| o.s.camel.spring.SpringCamelContext : Apache Camel 2.23.2.fuse-760020-redhat-00001 (CamelContext: camel-1) is starting
main| o.s.c.w.ManagedManagementStrategy : JMX is enabled
main| o.s.c.i.impl.DefaultStreamCachingStrategy : StreamCaching in use with spool directory: C:\Users\TaherH\I55AT5\AppData\Local\Temp\camel\
main| o.s.c.c.jackson.JacksonDataFormat : Found single ObjectMapper in Registry to use: com.fasterxml.jackson.databind.ObjectMapper
main| o.s.c.c.jackson.JacksonDataFormat : Found single ObjectMapper in Registry to use: com.fasterxml.jackson.databind.ObjectMapper
main| o.s.camel.spring.SpringCamelContext : Route: routes started and consuming from: direct://greetingsImpl
main| o.s.camel.spring.SpringCamelContext : Route: greeting-api started and consuming from: servlet:/api-doc?httpMethodRestrict=GET&rest
main| o.s.camel.spring.SpringCamelContext : Route: doc-api started and consuming from: servlet:/api-doc?httpMethodRestrict=GET&rest
main| o.s.camel.spring.SpringCamelContext : Total 3 routes, of which 3 are started
main| o.s.camel.spring.SpringCamelContext : Apache Camel 2.23.2.fuse-760020-redhat-00001 (CamelContext: camel-1) started in 0.470
main| o.s.c.s.CamelHttpTransportServlet : Initialized CamelHttpTransportServlet[name=CamelServlet, contextPath=]
main| org.xmlio : XMLIO version 3.3.8.Final
main| org.xmlio.nio : XMLIO NIO Implementation Version 3.3.8.Final
main| o.s.b.w.e.u.UndertowServletWebServer : Undertow started on port(s) 9999 (http) with context path ''
main| c.r.fuse.boosters.rest.http.Application : Started Application in 6.178 seconds (JVM running for 12.724)
XHIO-1 task-1| io.undertow.servlet : Initializing Spring DispatcherServlet 'dispatcherServlet'
XHIO-1 task-1| o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
XHIO-1 task-1| o.s.web.servlet.DispatcherServlet : Completed initialization in 14 ms
```



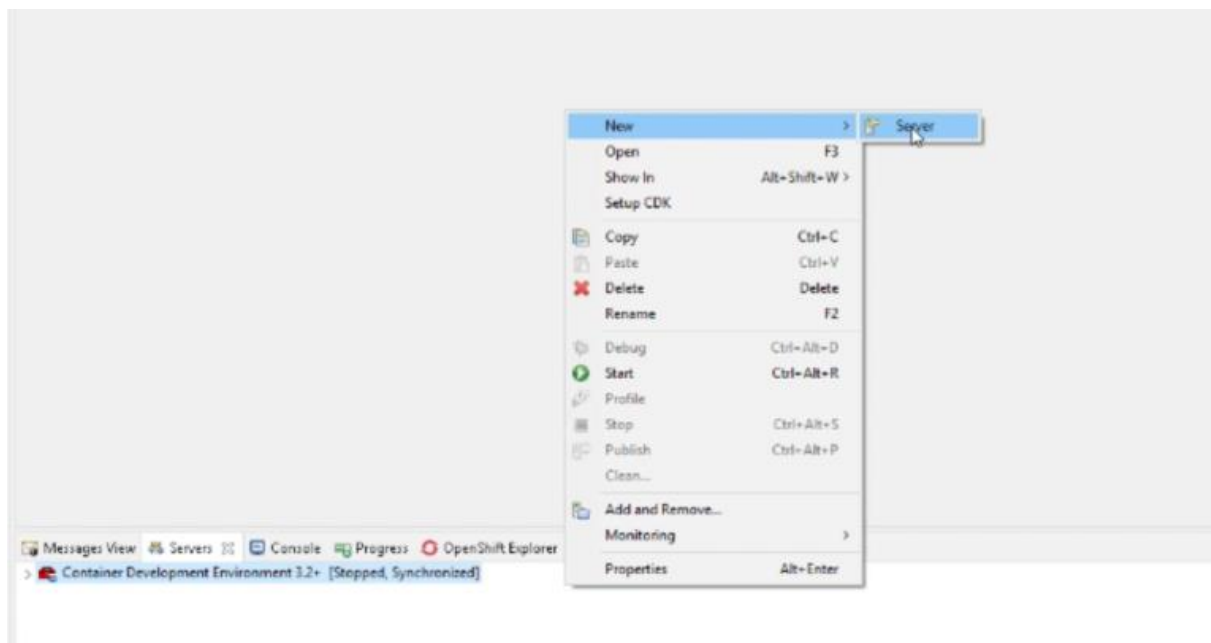


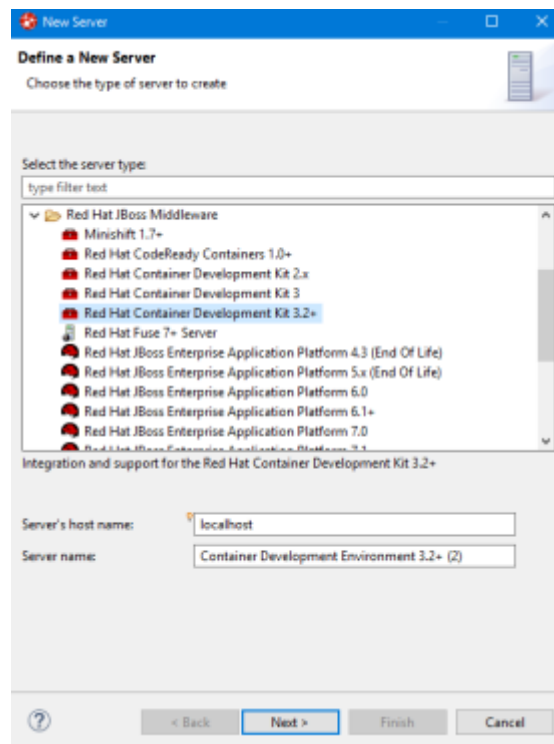
Le retour de l'Api sous forme **JSON**



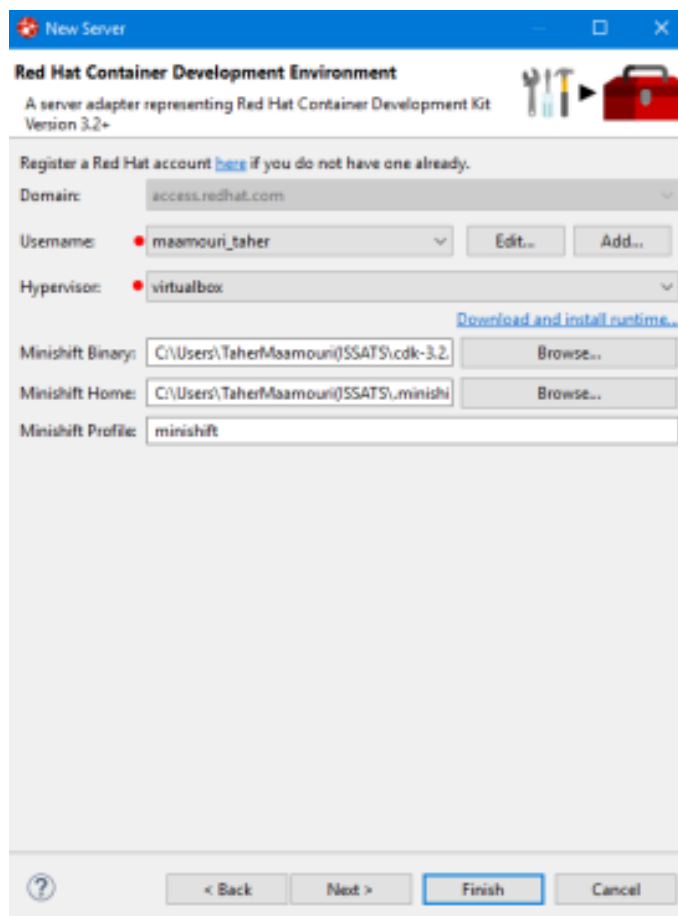
## Deployment sur OpenShift

Création de container Openshift

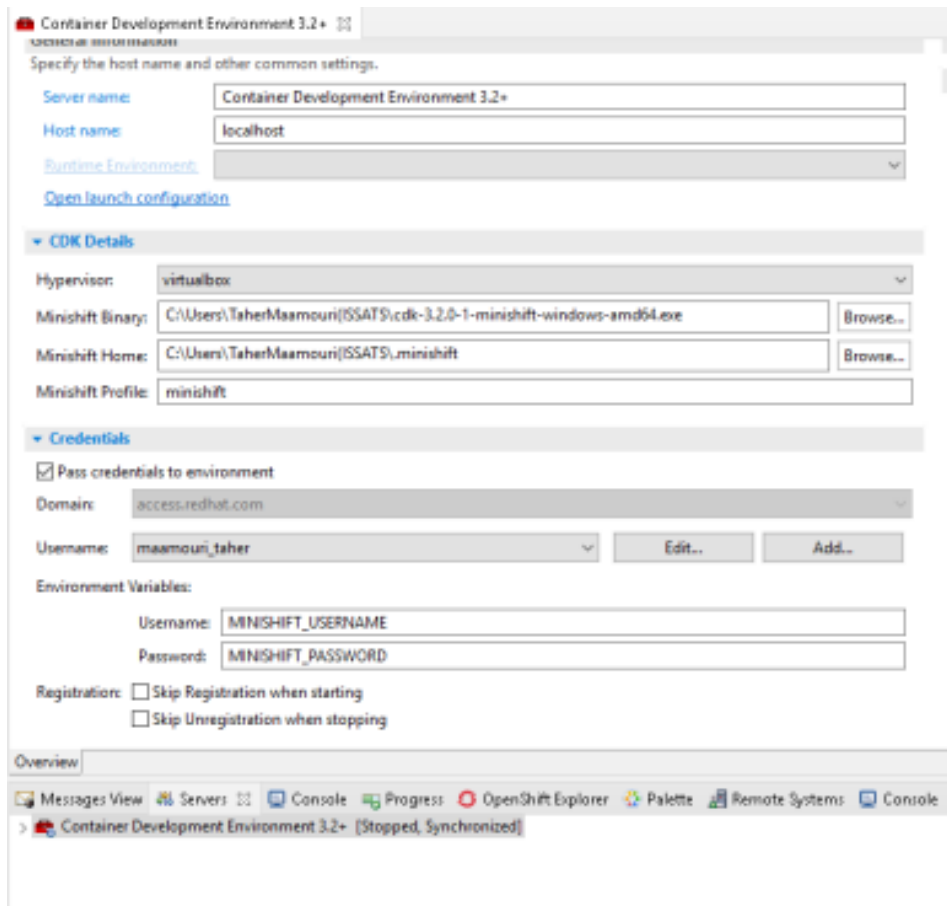




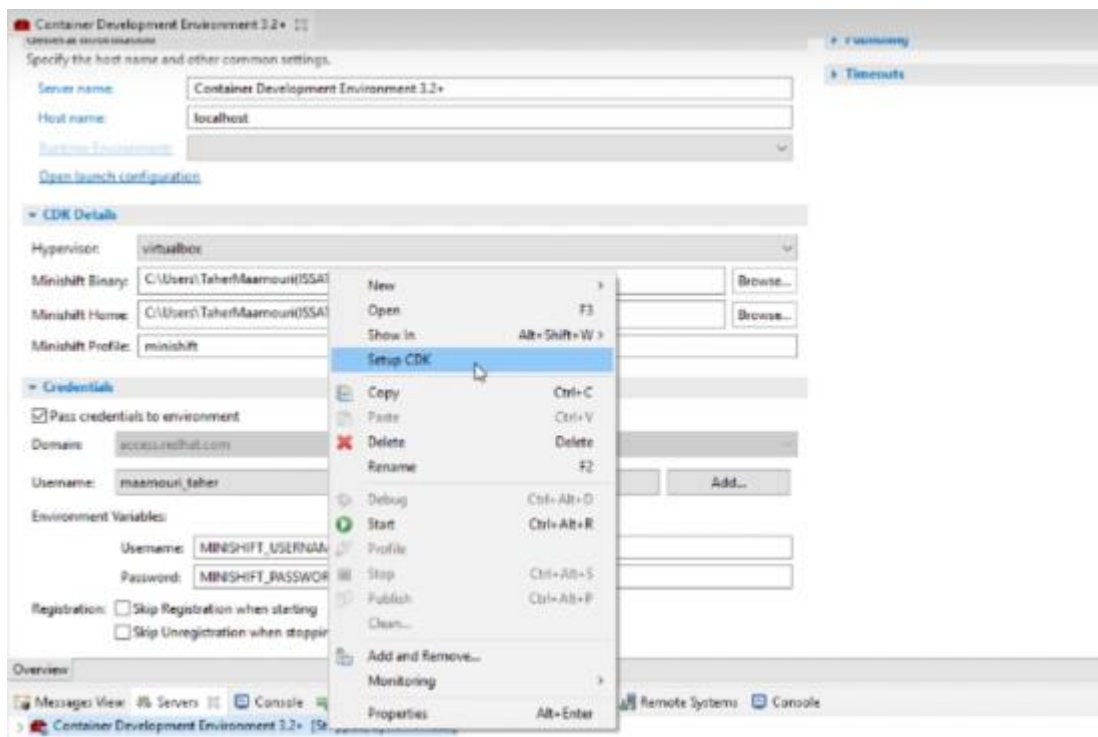
L'ajout du compte Redhat et le choix de l'hyperviseur de la machine virtuelle



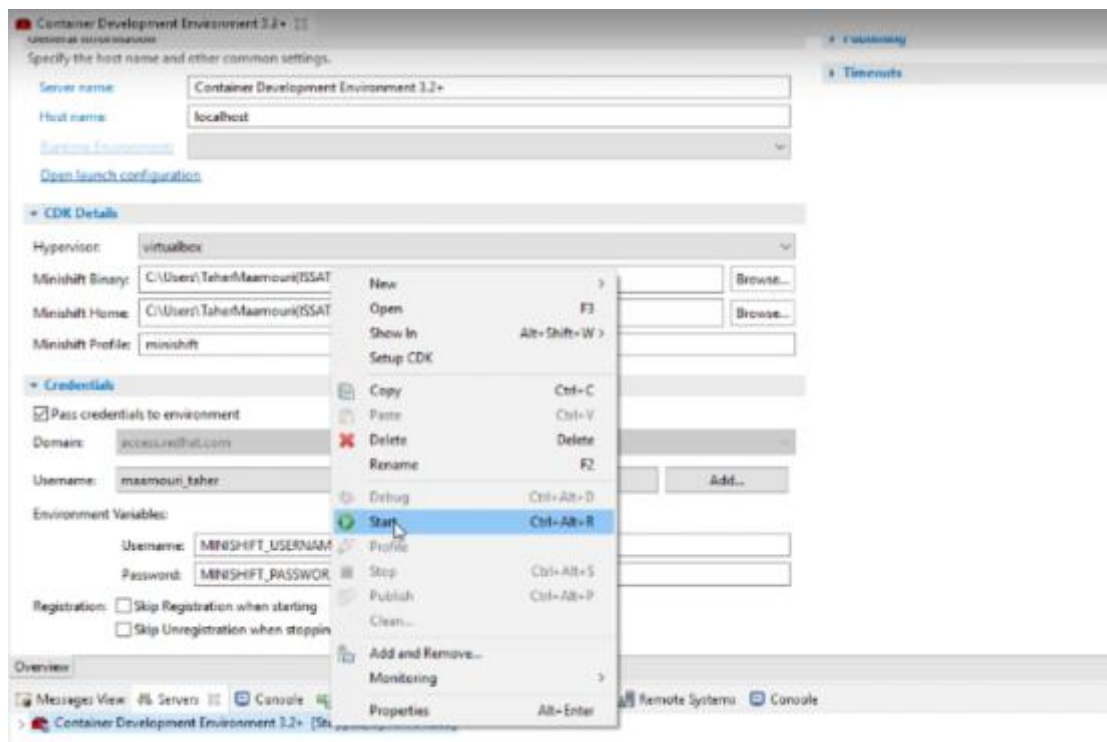
La configuration des variables d'environnement :



Setup CDK pour la création de l'image virtuelle minishift.



## Lancement de minshif :



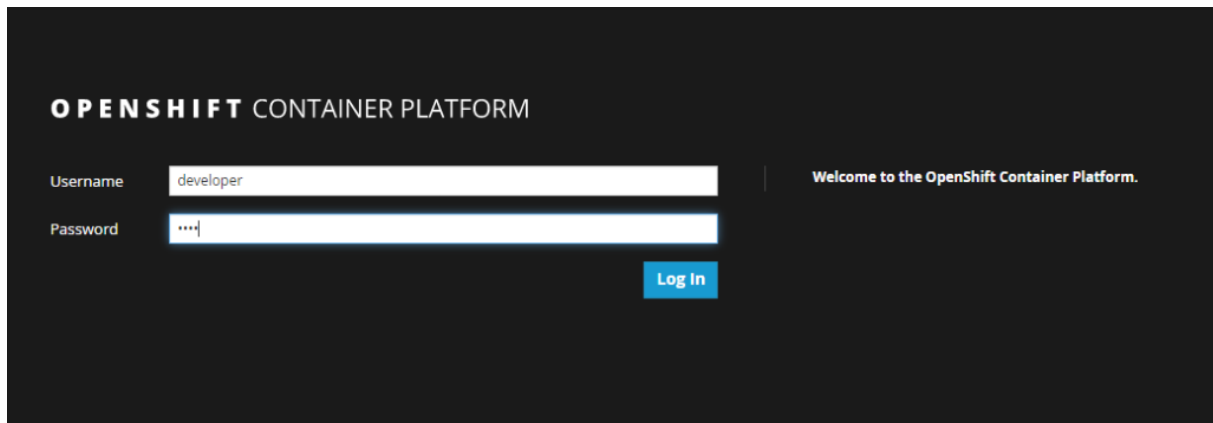
Messages View Servers Console Progress OpenShift Explorer Palette Remote Systems

<terminated> Container Development Environment 3.2+ [Red Hat CRC / CDK / Minishift Server Startup Configuration] C:\Users\T

```
-- Starting local OpenShift cluster using 'virtualbox' hypervisor ...
-- Starting Minishift VM ..... OK
-- Registering machine using subscription-manager
Registration in progress ..... OK [32s]
-- Checking for IP address ... OK
-- Checking if external host is reachable from the Minishift VM ...
Pinging 8.8.8.8 ... OK
-- Checking HTTP connectivity from the VM ...
Retrieving http://minishift.io/index.html ... OK
-- Checking if persistent storage volume is mounted ... OK
-- Checking available disk space ... 37% used OK
-- OpenShift cluster will be configured with ...
Version: v3.6.173.0.21
-- Checking 'oc' support for startup flags ...
host-data-dir ... OK
host-pv-dir ... OK
host-volumes-dir ... OK
routing-suffix ... OK
host-config-dir ... OK
Starting OpenShift using registry.access.redhat.com/openshift3/ose:v3.6.173.0.21 ...
OpenShift server started.
```

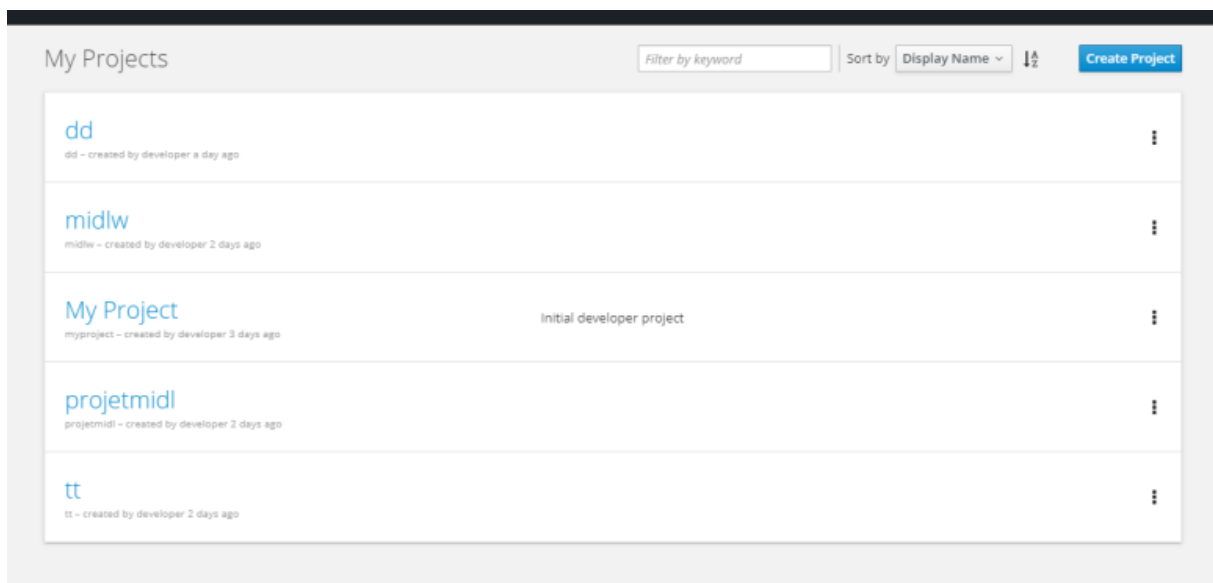
The server is accessible via web console at:  
<https://192.168.99.103:8443>

Connexion avec le compte développeur créé par défaut avec l'installation minishift



The image shows the OpenShift Container Platform login interface. It has a dark background with the title 'OPENSIFT CONTAINER PLATFORM' at the top. Below the title, there are two input fields: 'Username' with the value 'developer' and 'Password' with masked characters '\*\*\*\*'. A blue 'Log In' button is positioned below the password field. To the right of the login fields, a message reads 'Welcome to the OpenShift Container Platform.'

Création d'un projet en OpenShift :

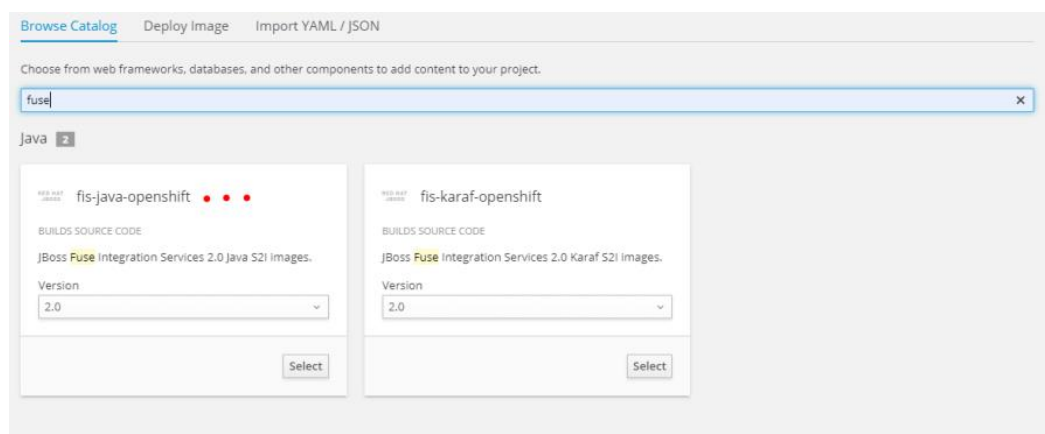
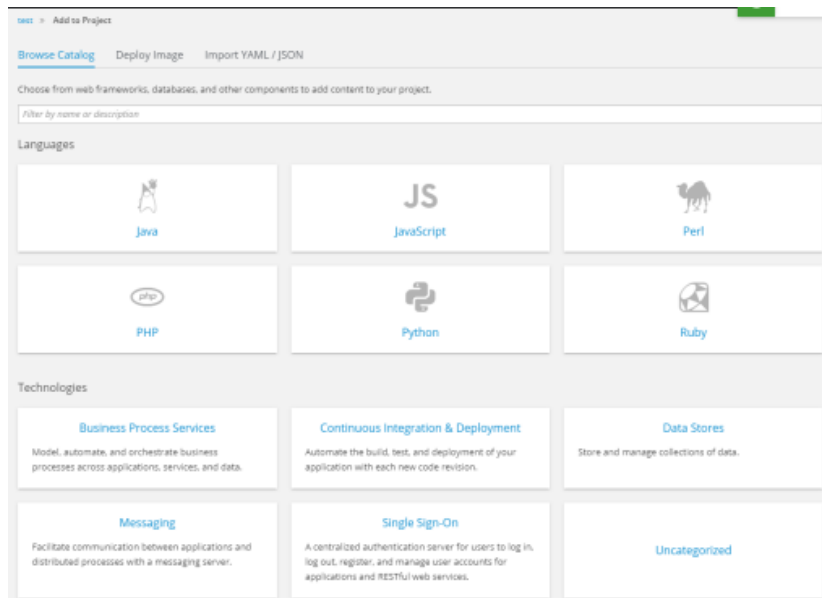


The image displays the 'My Projects' page in OpenShift. At the top, there's a header with 'My Projects', a search bar labeled 'Filter by keyword', a 'Sort by' dropdown set to 'Display Name', and a 'Create Project' button. Below the header is a list of five projects, each with a name, a description, and a three-dot menu icon. The projects are: 'dd' (created by developer a day ago), 'midlw' (created by developer 2 days ago), 'My Project' (myproject - created by developer 3 days ago, labeled as 'Initial developer project'), 'projetmidl' (created by developer 2 days ago), and 'tt' (created by developer 2 days ago).

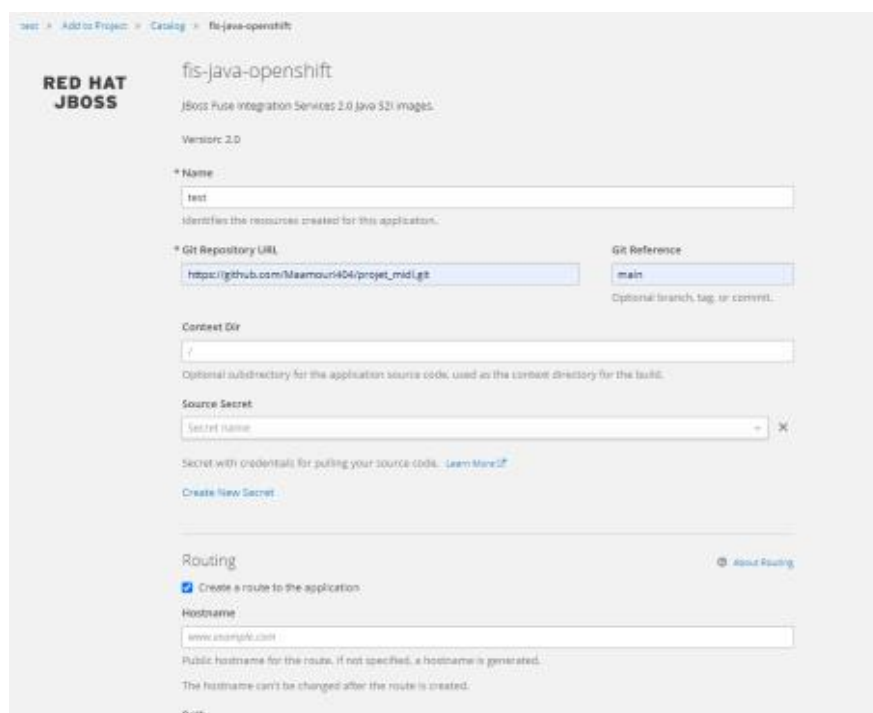


The image shows the 'Create Project' form in OpenShift. It has a title 'Create Project' and three main sections: '\* Name' with a text input field containing 'test' and a subtext 'A unique name for the project.', 'Display Name' with a text input field containing 'test', and 'Description' with a text area containing 'A short description.'. At the bottom, there are two buttons: 'Create' (blue) and 'Cancel' (grey).

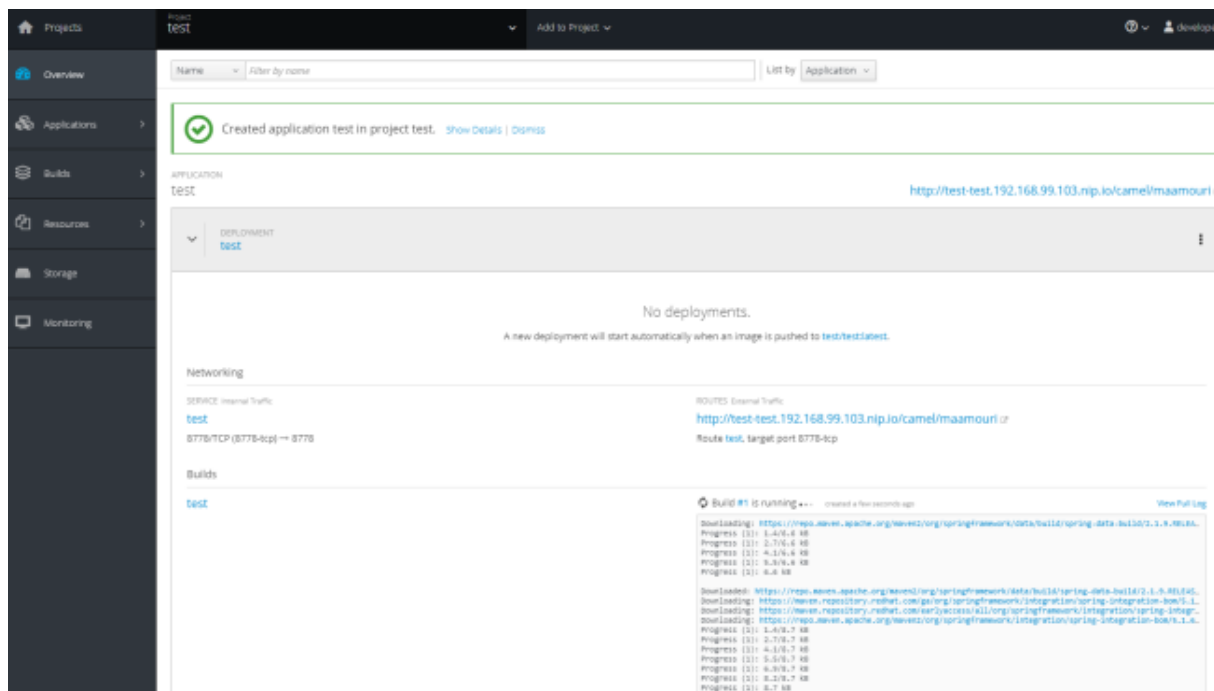
## Choix de type de l'image (l'enveloppe ) de notre projet :



## Configuration de l'image :



## Building de projet :



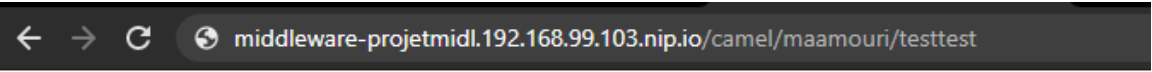
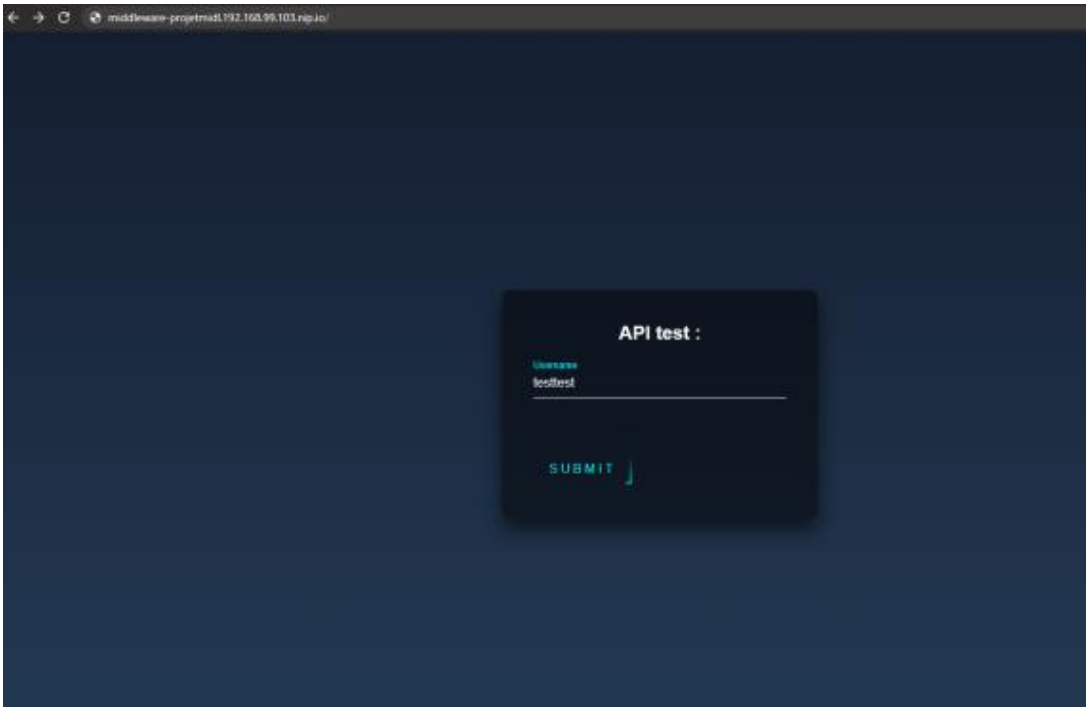
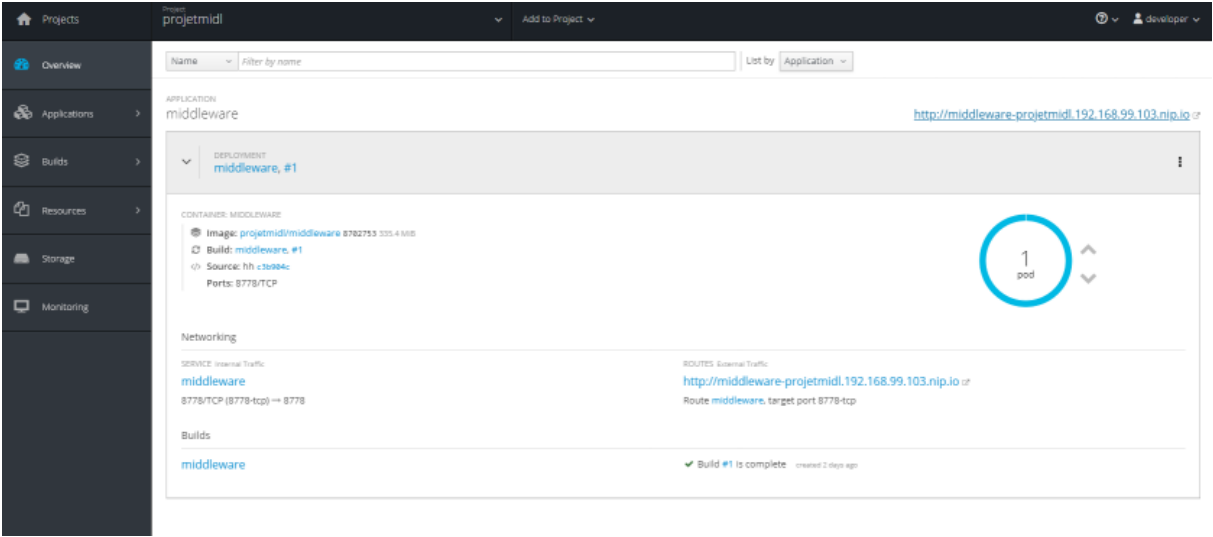
The screenshot shows the AWS CloudFormation console for a project named 'test'. The left sidebar contains navigation links for Overview, Applications, Builds, Resources, Storage, and Monitoring. The main content area displays the application details for 'test'. At the top, a green checkmark indicates 'Created application test in project test.' Below this, the application is shown in the 'DEPLOYMENT' state. The URL 'http://test-test.192.168.99.103.nip.io/camel/maamouri' is displayed. The deployment status is 'No deployments.' with a note: 'A new deployment will start automatically when an image is pushed to test:test:latest.' The Networking section shows the service 'test' with the endpoint '8778/TCP (8778-8cp) → 8778'. The Builds section shows 'Build #1' is running, with a list of dependencies and their download progress.

## la fin de build et le deploiemnt de projet :

```
Downloaded: https://repo.maven.apache.org/maven2/com/google/guava/guava/11.0.2/guava-11.0.2.jar (1.6 MB at 491 kB/s)
3632 [INFO] Replacing main artifact with repackaged archive
3633 [INFO] -----
3634 [INFO] BUILD SUCCESS
3635 [INFO] -----
3636 [INFO] Total time: 10:21 min
3637 [INFO] Finished at: 2021-01-29T20:54:36Z
3638 [INFO] Final Memory: 55M/2174M
3639 [INFO] -----
3640 Copying Maven artifacts from /tmp/src/target to /deployments ...
3641 Running: cp *.jar /deployments
3642 ... done
3643 Pushing image 172.30.1.1:5000/test/test:latest ...
3644 Pushed 0/6 layers, 0% complete
3645 Pushed 1/6 layers, 19% complete
3646 Pushed 2/6 layers, 38% complete
3647 Pushed 3/6 layers, 61% complete
3648 Pushed 4/6 layers, 91% complete
3649 Pushed 5/6 layers, 93% complete
3650 Pushed 6/6 layers, 100% complete
3651 Push successful
```



On va ouvrir le lien généré par OpenShift



```
{"greetings": "username: testtest"}
```