

Technologie utrzymania aplikacji 2020

Zadanie 1: Migracja

Mateusz Wasilewski 216913

Szymon Dobrowolski 216747

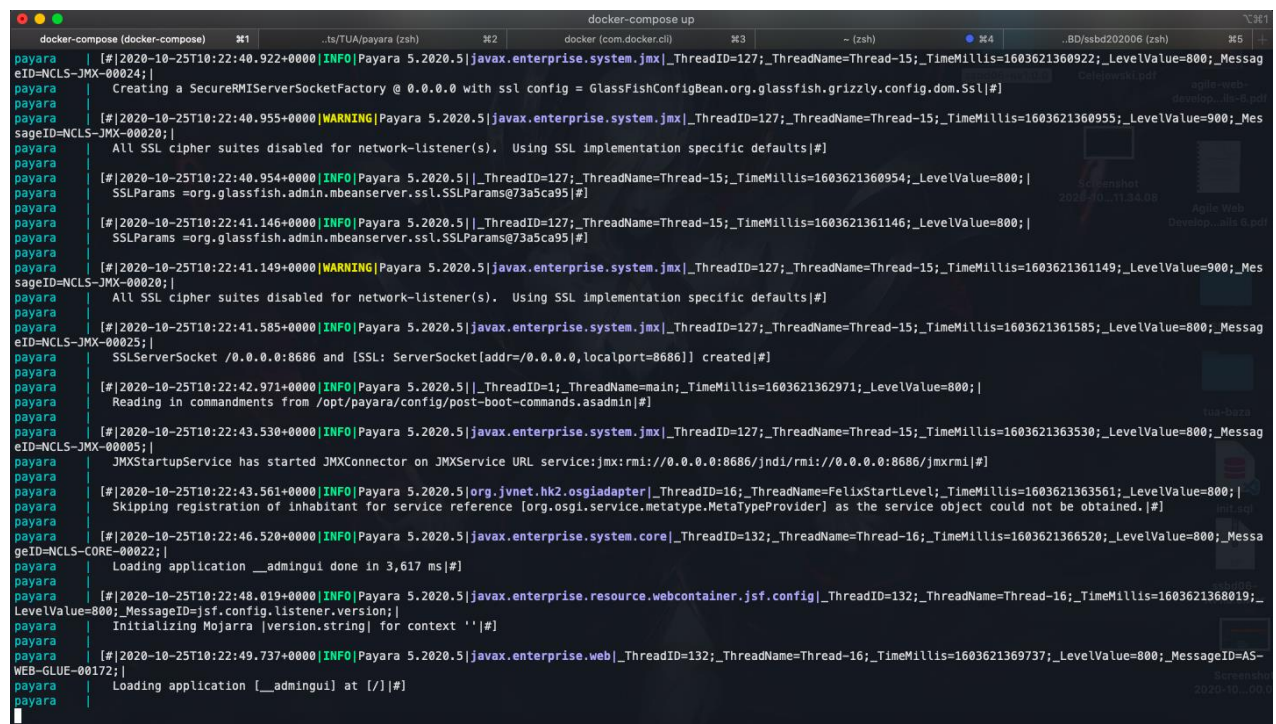
Jędrzej Dobrucki 216748

Kamil Celejewski 216733

Maciej Bartos 216719

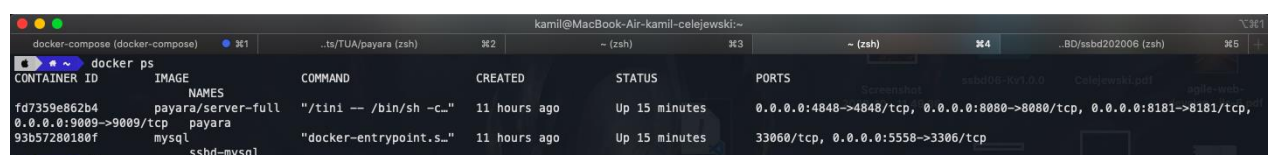
1. Pobranie i uruchomienie kontenerów za pomocą docker-compose:

docker-compose up



```
docker-compose (docker-compose) #1 ..ts/TUA/payara (zsh) #2 docker (com.docker.cli) #3 ~ (zsh) #4 ..BD/ssbd202006 (zsh) #5 +
payara [#|2020-10-25T10:22:40.922+0000|INFO|Payara 5.2020.5|javax.enterprise.system.jmx|_ThreadID=127;_ThreadName=Thread-15;_TimeMillis=1603621360922;_LevelValue=800;_MessageID=NCLS-JMX-00024;|
Creating a SecureRMIServerSocketFactory @ 0.0.0.0 with ssl config = GlassFishConfigBean.org.glassfish.grizzly.config.dom.Ssl|#]
payara [#|2020-10-25T10:22:40.955+0000|WARNING|Payara 5.2020.5|javax.enterprise.system.jmx|_ThreadID=127;_ThreadName=Thread-15;_TimeMillis=1603621360955;_LevelValue=900;_MessageID=NCLS-JMX-00020;|
All SSL cipher suites disabled for network-listener(s). Using SSL implementation specific defaults|#]
payara [#|2020-10-25T10:22:40.954+0000|INFO|Payara 5.2020.5|_ThreadID=127;_ThreadName=Thread-15;_TimeMillis=1603621360954;_LevelValue=800;|
SSLParams =org.glassfish.admin.mbeanserver.ssl.SSLParams@73a5ca95|#]
payara [#|2020-10-25T10:22:41.146+0000|INFO|Payara 5.2020.5|_ThreadID=127;_ThreadName=Thread-15;_TimeMillis=1603621361146;_LevelValue=800;|
SSLParams =org.glassfish.admin.mbeanserver.ssl.SSLParams@73a5ca95|#]
payara [#|2020-10-25T10:22:41.149+0000|WARNING|Payara 5.2020.5|javax.enterprise.system.jmx|_ThreadID=127;_ThreadName=Thread-15;_TimeMillis=1603621361149;_LevelValue=900;_MessageID=NCLS-JMX-00020;|
All SSL cipher suites disabled for network-listener(s). Using SSL implementation specific defaults|#]
payara [#|2020-10-25T10:22:41.585+0000|INFO|Payara 5.2020.5|javax.enterprise.system.jmx|_ThreadID=127;_ThreadName=Thread-15;_TimeMillis=1603621361585;_LevelValue=800;_MessageID=NCLS-JMX-00025;|
SSLServerSocket /0.0.0.0:8686 and [SSL: ServerSocket[addr=0.0.0.0,localport=8686]] created|#]
payara [#|2020-10-25T10:22:42.971+0000|INFO|Payara 5.2020.5|_ThreadID=1;_ThreadName=main;_TimeMillis=1603621362971;_LevelValue=800;|
Reading in commandments from /opt/payara/config/post-boot-commands.asadmin|#]
payara [#|2020-10-25T10:22:43.530+0000|INFO|Payara 5.2020.5|javax.enterprise.system.jmx|_ThreadID=127;_ThreadName=Thread-15;_TimeMillis=1603621363530;_LevelValue=800;_MessageID=NCLS-JMX-00005;|
JMXStartupService has started JMXConnector on JMXService URL service:jmx:rmi://0.0.0.0:8686/jndi/rmi://0.0.0.0:8686/jmxrmi|#]
payara [#|2020-10-25T10:22:43.561+0000|INFO|Payara 5.2020.5|org.jvnet.hk2.osgiadapter|_ThreadID=16;_ThreadName=FelixStartLevel;_TimeMillis=1603621363561;_LevelValue=800;|
Skipping registration of inhabitant for service reference [org.osgi.service.metatype.MetaTypeProvider] as the service object could not be obtained.|#]
payara [#|2020-10-25T10:22:46.520+0000|INFO|Payara 5.2020.5|javax.enterprise.system.core|_ThreadID=132;_ThreadName=Thread-16;_TimeMillis=1603621366520;_LevelValue=800;_MessageID=NCLS-CORE-00022;|
Loading application __admingui done in 3,617 ms|#]
payara [#|2020-10-25T10:22:48.019+0000|INFO|Payara 5.2020.5|javax.enterprise.resource.webcontainer.jsf.config|_ThreadID=132;_ThreadName=Thread-16;_TimeMillis=1603621368019;_LevelValue=800;_MessageID=jsf.config.listener.version;|
Initializing Mojarra [version.string] for context ''|#]
payara [#|2020-10-25T10:22:49.737+0000|INFO|Payara 5.2020.5|javax.enterprise.web|_ThreadID=132;_ThreadName=Thread-16;_TimeMillis=1603621369737;_LevelValue=800;_MessageID=AS-WEB-GLUE-00172;|
Loading application [_admingui] at [/]|#]
payara
```

docker ps



CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
fd7359e862b4	payara/server-full		"/tini -- /bin/sh -c."	11 hours ago	Up 15 minutes	0.0.0.0:4848->4848/tcp, 0.0.0.0:8080->8080/tcp, 0.0.0.0:8181->8181/tcp,
0.0.0.0:9009->9009/tcp	payara					
93b57280180f	mysql		"docker-entrypoint.s..."	11 hours ago	Up 15 minutes	33060/tcp, 0.0.0.0:5558->3306/tcp
	ssbd-mysql					

2. Zalogować się do panelu administracyjnego hasło admin user admin:

<https://localhost:4848/common/index.jsf>

3. Dodanie security realm:

The screenshot shows the Payara Community web console at <https://localhost:4848/common/index.jsf>. The left sidebar shows a tree view of the configuration, with 'ssbd06realm' selected under 'Realms'. The main panel displays the configuration for 'ssbd06realm' with the following fields:

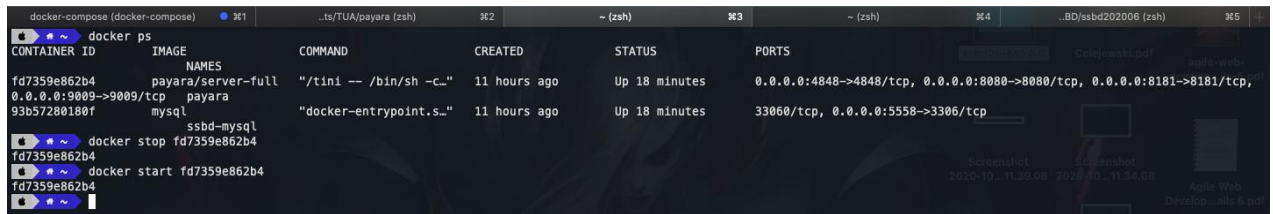
Configuration Name:	server-config
Realm Name:	ssbd06realm
Class Name:	com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm
Properties specific to this Class	
JAAS Context:	jdbcRealm <small>Identifier for the login module to use for this realm</small>
JNDI:	jdbc/ssbd06authDS <small>JNDI name of the JDBC resource used by this realm</small>
User Table:	glassfish_auth_view <small>Name of the database table that contains the list of authorized users for this realm</small>
User Name Column:	login <small>Name of the column in the user table that contains the list of user names</small>
Password Column:	haslo <small>Name of the column in the user table that contains the user passwords</small>
Group Table:	glassfish_auth_view <small>Name of the database table that contains the list of groups for this realm</small>
Group Table User Name Column:	 <small>Name of the column in the user group table that contains the list of groups for this realm</small>
Group Name Column:	poziom <small>Name of the column in the group table that contains the list of group names</small>
Assign Groups:	 <small>Comma-separated list of group names</small>
Database User:	 <small>Specify the database user name in the realm instead of the JDBC connection pool</small>
Database Password:	 <small>Specify the database password in the realm instead of the JDBC connection pool</small>
Digest Algorithm:	SHA-256 <small>Digest algorithm (default is SHA-256); note that the default was MD5 in GlassFish versions prior to 3.1</small>
Encoding:	 <small>Encoding (allowed values are Hex and Base64)</small>
Charset:	UTF-8 <small>Character set for the digest algorithm</small>

```
<auth-realm classname="com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm"
name="ssbd06realm">
  <property name="charset" value="UTF-8"></property>
  <property name="group-table" value="glassfish_auth_view"></property>
  <property name="password-column" value="haslo"></property>
  <property name="digest-algorithm" value="SHA-256"></property>
  <property name="user-table" value="glassfish_auth_view"></property>
  <property name="datasource-jndi" value="jdbc/ssbd06authDS"></property>
  <property name="user-name-column" value="login"></property>
  <property name="group-name-column" value="poziom"></property>
  <property name="jaas-context" value="jdbcRealm"></property>
</auth-realm>
```

4. Zatrzymanie i ponowne uruchomienie kontenera payara:

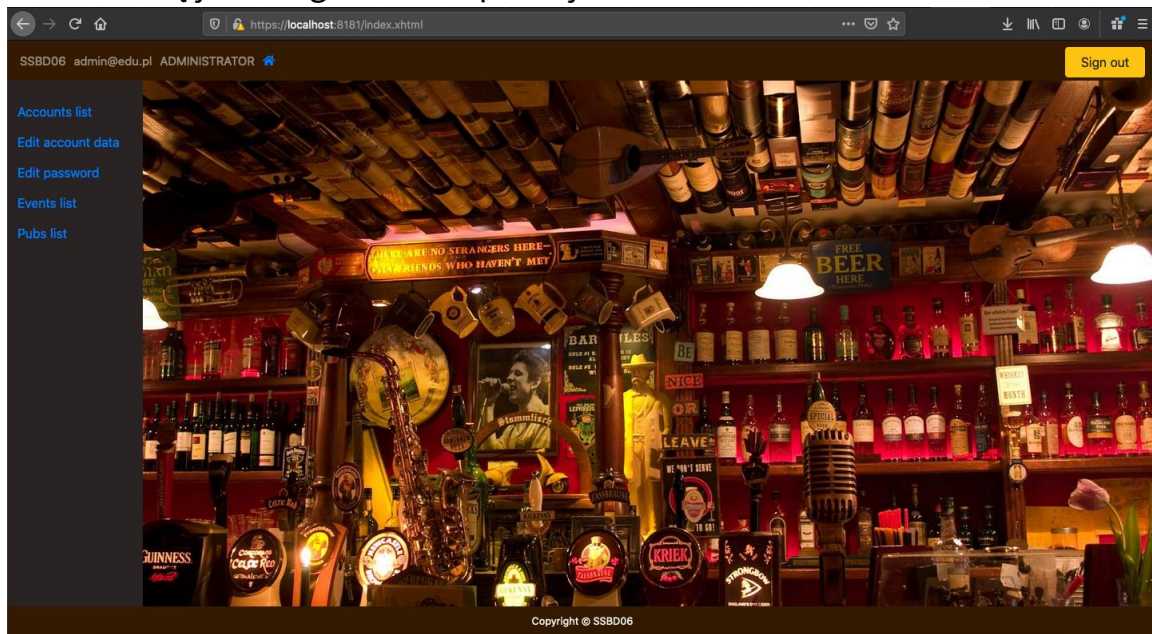
```
docker stop fd7359e862b4
```

```
docker start fd7359e862b4
```



5. Odczekać chwilę aż serwer uruchomi się ponownie

6. Można się już zalogować do aplikacji



7. Zakres własnego wkładu w realizację zadania

Kamil Celejewski:

- docker compose v1
- ustawienie przygotowanej konfiguracji w kontenerze serwera
- połączenie między kontenerem payary a kontenerem mysql
- dokumentacja/instrukcja

Maciej Bartos:

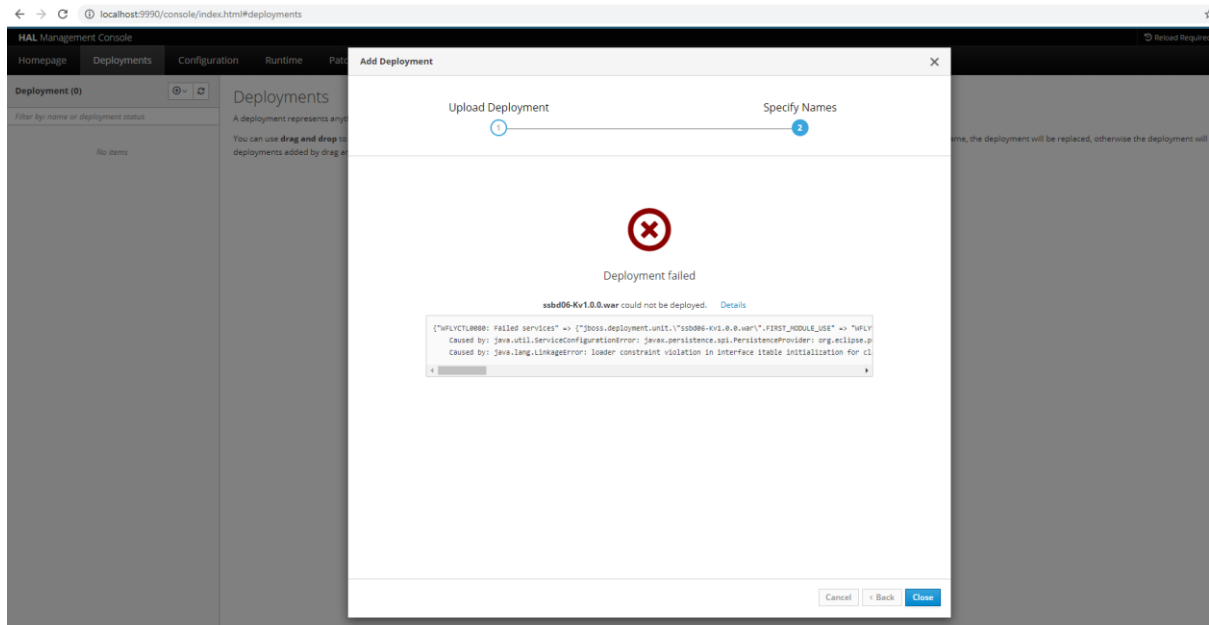
- Wstępna migracja bazy danych do kontenera dockera
- Przygotowanie konfiguracji zasobów (connection pool oraz resources) dla serwera aplikacyjnego
- Wstępne nawiązanie komunikacji między serwerem (lokalnie) a bazą danych (kontener), dostosowanie wersji sterowników mysql-connector

Szymon Dobrowolski:

- Przygotowanie skryptu tworzącego strukturę bazy danych oraz dane inicjalizacyjne (odpowiednie dla bazy mysql).
- Przygotowanie wstępnego DockerFile'a, automatyzującego tworzenie bazy danych wraz ze strukturami, danymi inicjalizacyjnymi oraz użytkownikami bazodanowymi.

Mateusz Wasilewski i Jędrzej Dobrucki:

- Przygotowanie DockerFile'a, automatyzującego cały proces wdrażania aplikacji.
- Modyfikacja Docker Compose, w celu zwiększenie automatyzacji wdrażania.
- Próba migracji na serwer aplikacyjny WildFly, finalnie zakończona niepowodzeniem.



```
{ "WFLYCTL0080: Failed services" => { "jboss.deployment.unit.\"ssbd06-Kv1.0.0.war\".FIRST_MODULE_USE" => "WFLYSRV0153: Failed to process phase FIRST_MODULE_USE of deployment \"ssbd06-Kv1.0.0.war\" Caused by: java.util.ServiceConfigurationError: javax.persistence.spi.PersistenceProvider: org.eclipse.persistence.jpa.PersistenceProvider Unable to get public no-arg constructor Caused by: java.lang.LinkageError: loader constraint violation in interface itable initialization for class org.eclipse.persistence.jpa.PersistenceProvider: when selecting method 'javax.persistence.spi.ProviderUtil javax.persistence.spi.PersistenceProvider.getProviderUtil()' the class loader 'javax.persistence.api@2.2.3' @4aa4fecb for super interface javax.persistence.spi.PersistenceProvider, and the class loader 'org.eclipse.persistence@2.5.2.v20140319-9ad6abd' @4e9af8e4 of the selected method's class, org.eclipse.persistence.jpa.PersistenceProvider have different Class objects for the type javax.persistence.spi.ProviderUtil used in the signature (javax.persistence.spi.PersistenceProvider is in unnamed module of loader 'javax.persistence.api@2.2.3' @4aa4fecb, parent loader 'app'; org.eclipse.persistence.jpa.PersistenceProvider is in unnamed module of loader 'org.eclipse.persistence@2.5.2.v20140319-9ad6abd' @4e9af8e4, parent loader 'app')"} }
```