

Automatisiertes Bereitstellen einer Java-Webapplikation

1. Voraussetzung

- AWS – Elastic Beanstalk
- Java Webapplikation
 - Java >= v.11
 - Maven

2. Aufgabe

Es soll eine automatisierte Bereitstellung eines Java Webservices geben welcher über online verfügbar ist. Dabei soll kein eigener Server angelegt oder Umgebung installiert werden müssen.

3. Ablauf Einrichtung Webservice (Java)

1. (optional) Öffnen des `webservice_example` in IDE der Wahl
2. Maven Befehl „mang clean package“ ausführen (IDE oder CMD)
3. Es wird eine neue .jar gebaut welche unter `webservice_example/target/` liegt



Neue REST-Interfaces können im File `webservice_example/src/main/java/com/cc3/webservice_example/controller/RestController.java` angelegt werden

Siehe: <https://spring.io/guides/tutorials/rest/>

4. Ablauf Einrichtung Elastic Beanstalk

1. Neue Applikation erstellen
 - Plattform: Java
 - Plattform branch: Corretto 11
 - Version: 3.1.6
2. Upload des eigenen Codes

Application code

☐ Sample application
Get started right away with sample code.

☒ Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

Source code origin

Version label
Unique name for this version of your application code.
1.0.0

Source code origin
Maximum size 512 MB

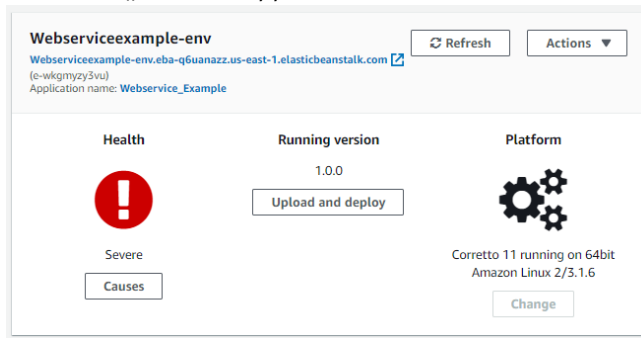
☒ Local file

☐ Public S3 URL

File name : `webservice_example-1.0.0.jar`

☒ File successfully uploaded

3. Drücke „Create Application“ (dauert etwas -> automatische Weiterleitung)



4. Konfiguration des Webservice-Ports (5000)

Webserviceexample-env

Go to environment [Go to environment](#)

Configuration

Logs

Health

Monitoring

Alarms

Managed updates

Events

Tags

Software

Environment properties: GRADLE_HOME, JAVA_HOME, M2, M2_HOME
Log streaming: disabled
Rotate logs: disabled
X-Ray daemon: disabled

Edit

Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

Name	Value
GRADLE_HOME	/usr/local/gradle
JAVA_HOME	/usr/lib/jvm/java-11-amazon-corretto.x86
M2	/usr/local/apache-maven/bin
M2_HOME	/usr/local/apache-maven
SERVER_PORT	5000

5. Nun sollte die Web-Applikation live geschaltet und über die URL aufrufbar sein

Webserviceexample-env

[Webserviceexample-env.eba-q6uanazz.us-east-1.elasticbeanstalk.com](#)

Application name: **Webservice_Example**

Health

Ok

Running version

1.0.0

Platform

Corretto 11 running on 64bit Amazon Linux 2/3.1.6

web-service-example-env.eba-q6uanazz.us-east-1.elasticbeanstalk.com

Web-service is running!

5. Ergebnis

Es ist möglich, ohne einen Server einzurichten eine Java-Webapplikation auf einen Server zu deployen. Dabei haben wir später viele Einstellungsmöglichkeiten, wie eine detaillierte Überwachung, Loadbalancer und Versionierung. Auch DB-Services sind möglich ohne größeren Aufwand zu verbinden.

Die Versionierung findet bei Upload einer neuen Version statt, womit man jederzeit schnell die Applikation auf der Environment auf einen neuen Stand hochziehen kann, bzw. auch wieder auf eine Vorversion, falls es Fehler gab.

▼ Webservice_Example

Application versions

Saved configurations

Application versions Actions Settings Upload Refresh

<input type="checkbox"/>	Version label	Description	Date created	Source	Deployed to
<input type="checkbox"/>	1.1.0		2021-03-16T12:22:03+01:00	20210752uk-webservice_example-1.1.0.jar	-
<input type="checkbox"/>	1.0.0		2021-03-16T12:01:13+01:00	2021075Tcb-webservice_example-1.0.0.jar	Webserviceexample-env

6. Stolperfallen

- Webservice ist nicht erreichbar

Problem	AWS arbeitet nicht über den Port 8080 wie standardmäßig Webapplikationen
Lösung	Server Port 5000 setzen

- Healthanzeige ist ständig rot

Problem	CloudWatch überwacht die Responses des Webservices und baut daraus einen Health-Report, es kann sein das die eigene Anzeige rot ist, trotz funktionierenden Server.
Lösung	1. Setzen der Response-Codes in der Webapplikation 2. Falls Default-Root (/) nicht genutzt wird trotzdem bei Aufruf Code 2xx zurückgeben

- Keine Pausierung möglich (Testumgebung – Produktion läuft meist durch)

Problem	Man kann Environments nicht pausieren
Lösung	Terminieren der Umgebung bei nicht nutzen <ul style="list-style-type: none">○ Jars bleiben erhalten und können schnell wieder in Env geladen werden○ Settings können ex- und importiert werden