POC_1 Deployment Documentation

The building process is optional - you will receive the last version of the app bundled as a .jar

Build the application

1. Download the app - it should contain two directories: back and front

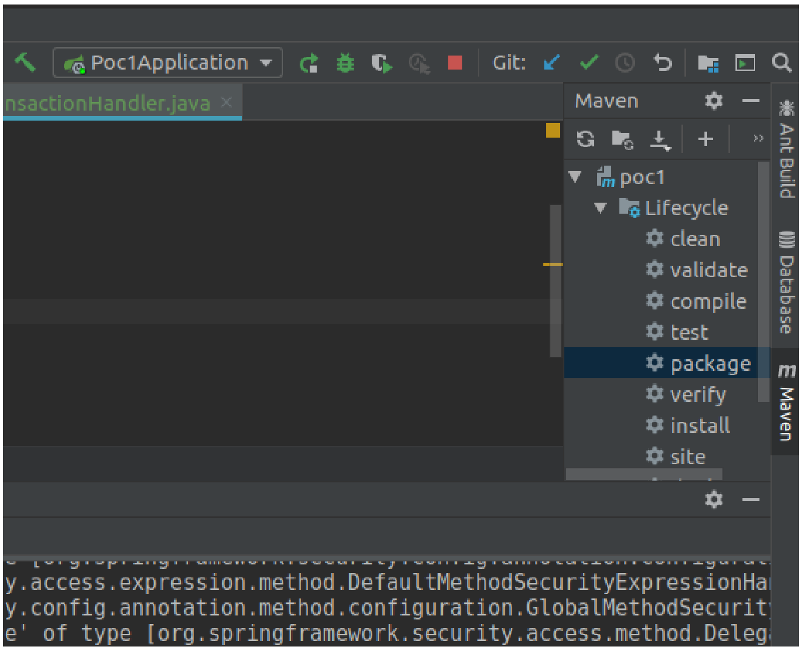
2. Install required tools
   a. for front-end: [Node.js]()
   b. for back-end: [Java]() & [Maven]()
      i. or [IntelliJ Community Edition]() that will take care of everything

3. Build front-end
   a. Navigate to front directory
   b. Open a terminal (bash/cmd/PowerShell)
   c. run “npm install”
   d. run “ng build --prod”

4. Navigate to back directory and create a new directory called “public” under the existing directory called “resources” (under “back/src/main/resources”)

5. Copy the content generated by the prod command from “front/dist/poc” to “back/src/main/resources/public”

6. Building the app
   a. Building the app with IntelliJ
      i. Open the “back” folder in IntelliJ and, after it loads up the project, open the “Maven” window (it should be present in the right side of the screen, if it’s not, press CTRL+E and choose it from the new opened window)
      ii. Select “Lifecycle” and double click on the “package” command



```
y.access.expression.method.DefaultMethodSecurityExpressionHa
y.config.annotation.method.configuration.GlobalMethodSecurit
e' of type [org.springframework.security.access.method.Deleg
```

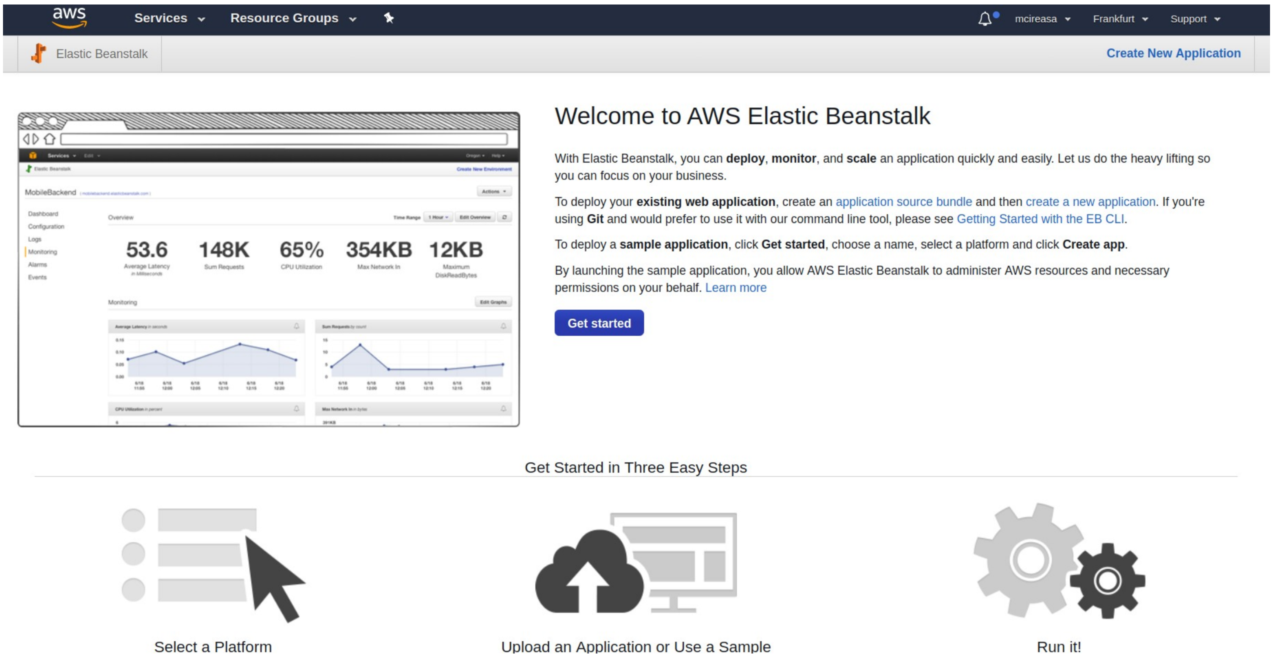
   b. ***Command Line***
      i. Open a terminal, navigate to “back” folder and run “mvn clean package”

7. Copy the resulted output “poc1-0.0.1-SNAPSHOT.jar” from back/target in an accessible place - this artifact includes both the back-end and the front-end of the application and will be used for deployment

Deploying the application on AWS using Elastic Beanstalk & AWS RDS

1. Log into your AWS account

2. Search “Elastic Beanstalk” service



Welcome to AWS Elastic Beanstalk

With Elastic Beanstalk, you can **deploy**, **monitor**, and **scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

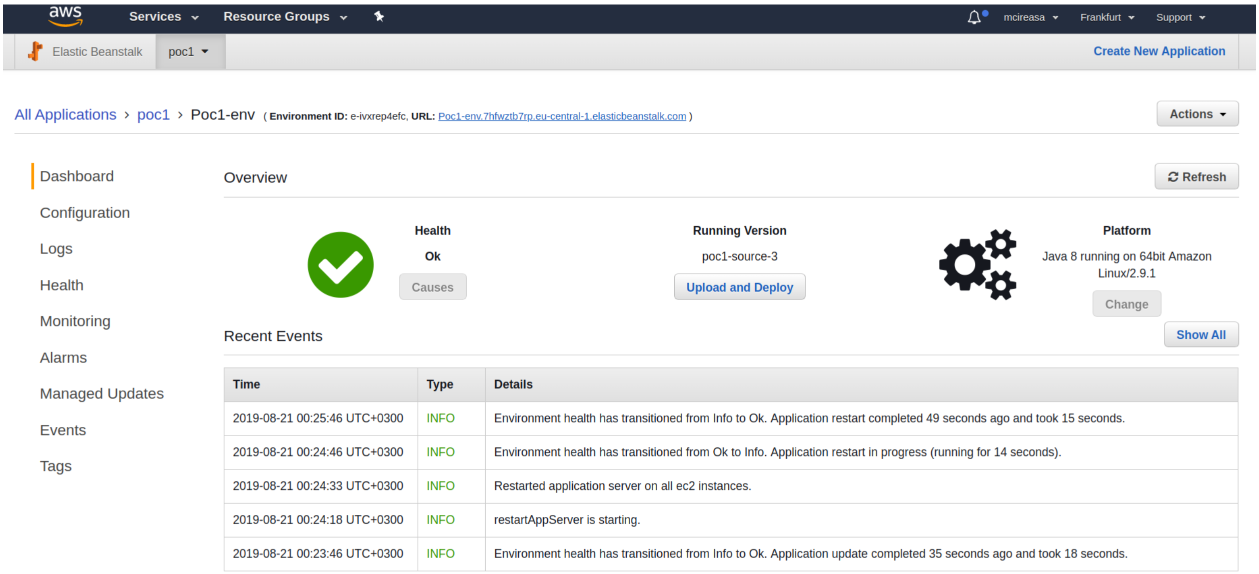
To deploy your **existing web application**, create an application source bundle and then create a new application. If you're using **Git** and would prefer to use it with our command line tool, please see Getting Started with the EB CLI.

To deploy a **sample application**, click **Get started**, choose a name, select a platform and click **Create app**.

By launching the sample application, you allow AWS Elastic Beanstalk to administer AWS resources and necessary permissions on your behalf. Learn more

3. Create elastic beanstalk
   a. set application name, platform Java,
   b. click on “Upload your code”, choose file and add the provided .jar file
   c. click on configure more options, scroll to bottom and select Database
   d. select sqlserver-ex as engine, set username and password, then click save
   e. click on create app

4. Wait for the application to be deployed. When everything is ready, you will see the following screen.



All Applications › poc1 › Poc1-env (**Environment ID:** e-ivxrep4efc, **URL:** Poc1-env.7hfwztb7rp.eu-central-1.elasticbeanstalk.com)

Overview

Health
Ok

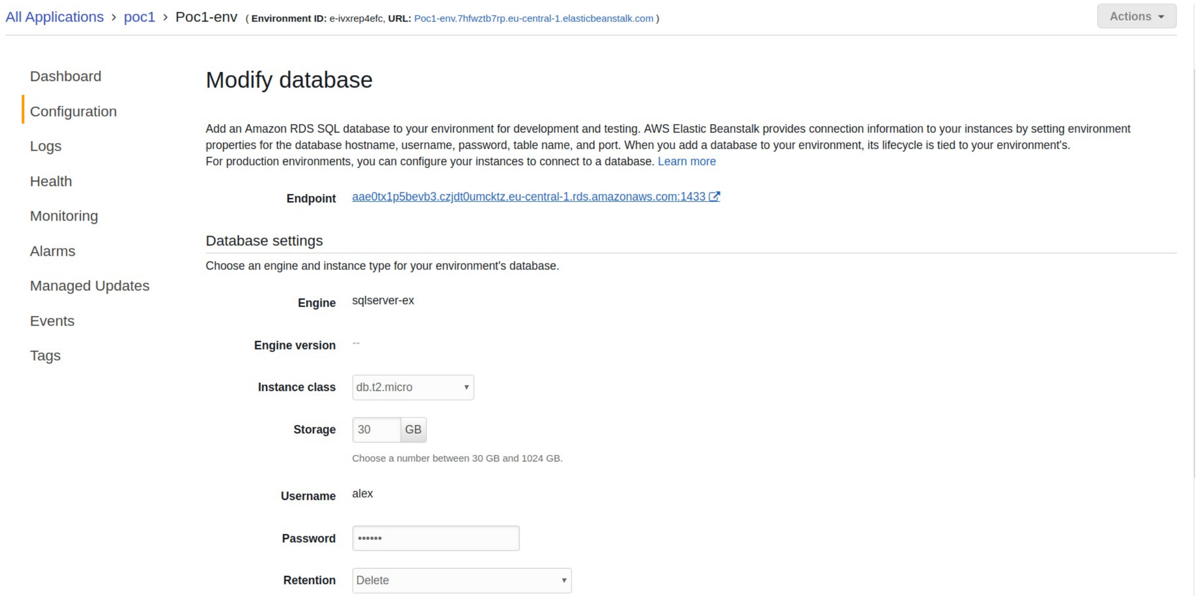
Running Version
poc1-source-3

Platform
Java 8 running on 64bit Amazon Linux/2.9.1

Recent Events

Time	Type	Details
2019-08-21 00:25:46 UTC+0300	INFO	Environment health has transitioned from Info to Ok. Application restart completed 49 seconds ago and took 15 seconds.
2019-08-21 00:24:46 UTC+0300	INFO	Environment health has transitioned from Ok to Info. Application restart in progress (running for 14 seconds).
2019-08-21 00:24:33 UTC+0300	INFO	Restarted application server on all ec2 instances.
2019-08-21 00:24:18 UTC+0300	INFO	restartAppServer is starting.
2019-08-21 00:23:46 UTC+0300	INFO	Environment health has transitioned from Info to Ok. Application update completed 35 seconds ago and took 18 seconds.

5. After the application was deployed, enter the database and add an inbound rule, accept all on MS SQL type connection, source anywhere
    a. Click on Configuration
    b. Scroll to bottom and click the “Modify” button on the “Database” row
    c. Click on the Endpoint URL



All Applications › poc1 › Poc1-env (**Environment ID:** e-ivxrep4efc, **URL:** Poc1-env.7hfwztb7rp.eu-central-1.elasticbeanstalk.com)

Modify database

Add an Amazon RDS SQL database to your environment for development and testing. AWS Elastic Beanstalk provides connection information to your instances by setting environment properties for the database hostname, username, password, table name, and port. When you add a database to your environment, its lifecycle is tied to your environment's. For production environments, you can configure your instances to connect to a database. Learn more

Endpoint aae0tx1p5bevb3.czjdt0umcktz.eu-central-1.rds.amazonaws.com:1433

Database settings

Choose an engine and instance type for your environment's database.

Engine sqlserver-ex

Engine version --

Instance class db.t2.micro

Storage 30 GB

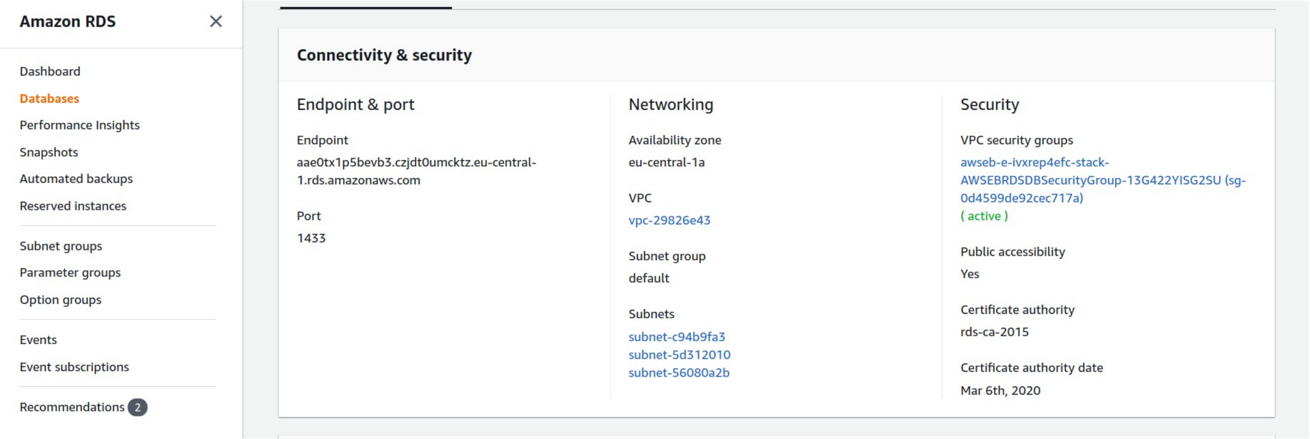
Choose a number between 30 GB and 1024 GB.

Username alex

Password ••••••

Retention Delete

    d. After the page loads, you should see the following dashboard
        i. be sure to copy the link under Endpoint, we’ll need it later



Connectivity & security

Endpoint & port

Endpoint
aae0tx1p5bevb3.czjdt0umcktz.eu-central-1.rds.amazonaws.com

Port
1433

Networking

Availability zone
eu-central-1a

VPC
vpc-29826e43

Subnet group
default

Subnets
subnet-c94b9fa3
subnet-5d312010
subnet-56080a2b

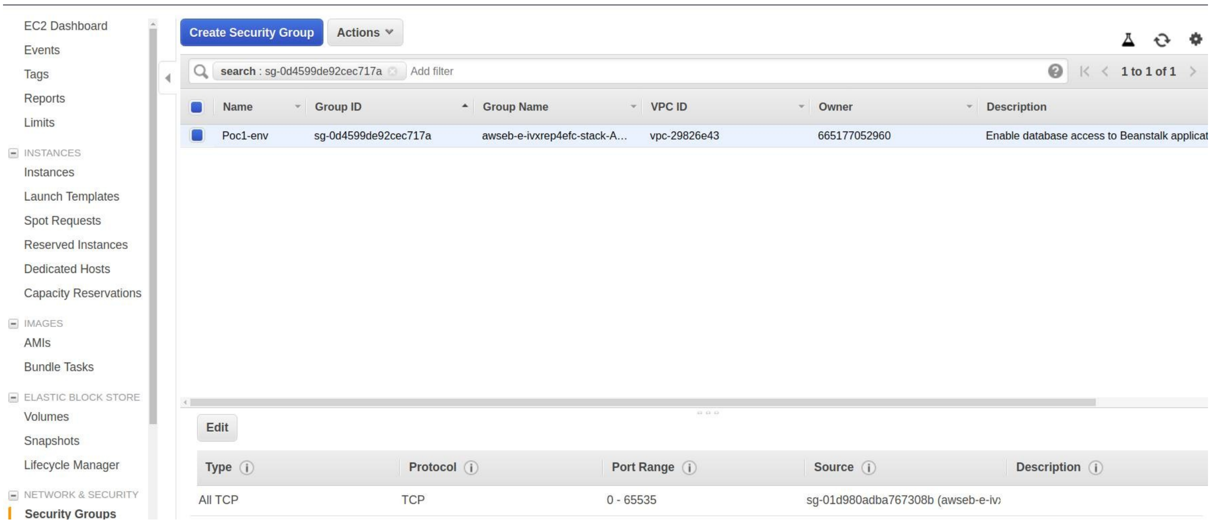
Security

VPC security groups
awseb-e-ivxrep4efc-stack-AWSEBRDSDBSecurityGroup-13G422YISG2SU (sg-0d4599de92cec717a)
(active)

Public accessibility
Yes

Certificate authority
rds-ca-2015

Certificate authority date
Mar 6th, 2020

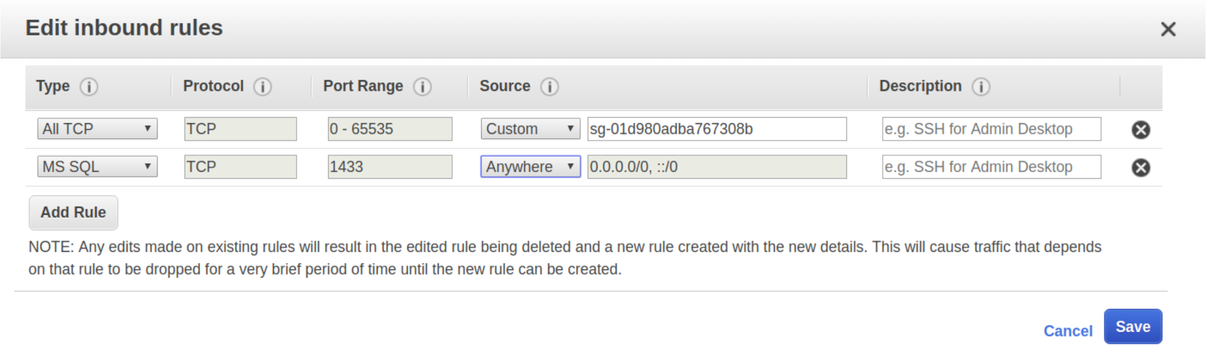


search : sg-0d4599de92cec717a

Name	Group ID	Group Name	VPC ID	Owner	Description
Poc1-env	sg-0d4599de92cec717a	awseb-e-ivxrep4efc-stack-A...	vpc-29826e43	665177052960	Enable database access to Beanstalk applicat

Type	Protocol	Port Range	Source	Description
All TCP	TCP	0 - 65535	sg-01d980adba767308b (awseb-e-iv:	

e. Click on the url under “VPC security groups”

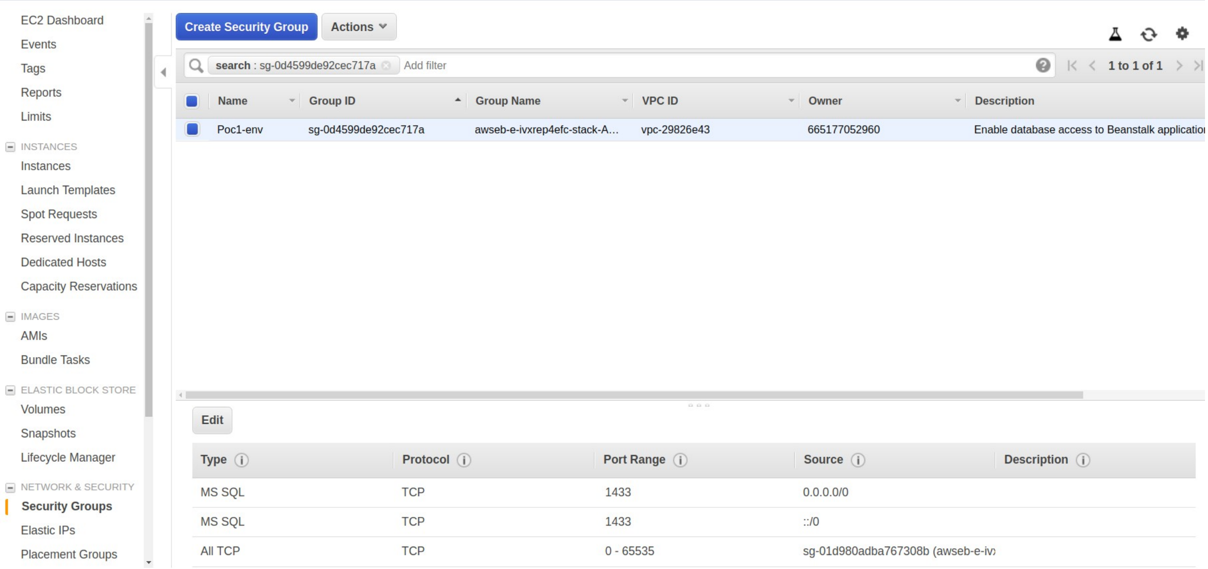

Edit inbound rules

Type	Protocol	Port Range	Source		Description
All TCP	TCP	0 - 65535	Custom	sg-01d980adba767308b	e.g. SSH for Admin Desktop
MS SQL	TCP	1433	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

f. Click on edit and add a new inbound security rule, like the one on the second row

g. After clicking save, the page should look like this

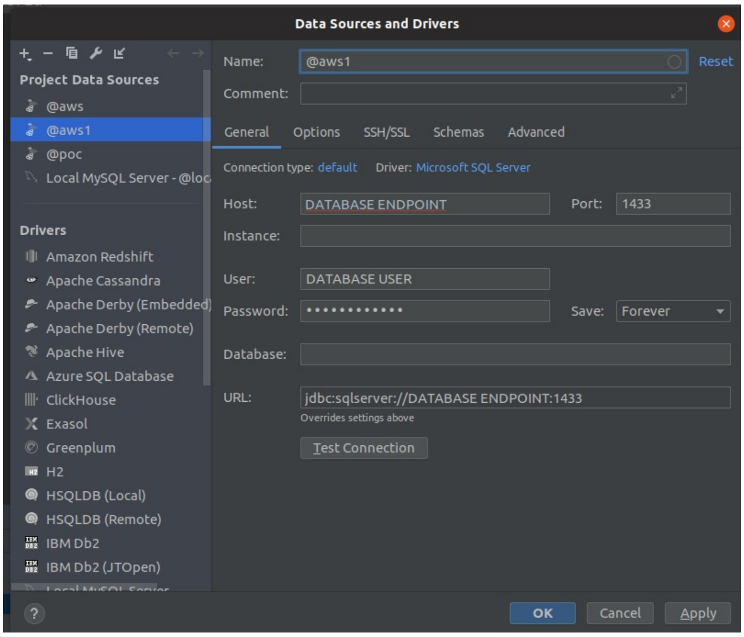


search : sg-0d4599de92cec717a

Name	Group ID	Group Name	VPC ID	Owner	Description
Poc1-env	sg-0d4599de92cec717a	awseb-e-ivxrep4efc-stack-A...	vpc-29826e43	665177052960	Enable database access to Beanstalk applicatio

Type	Protocol	Port Range	Source	Description
MS SQL	TCP	1433	0.0.0.0/0	
MS SQL	TCP	1433	::/0	
All TCP	TCP	0 - 65535	sg-01d980adba767308b (awseb-e-iv:	

6. Use a local client to create a new database
   a. Using DataGrip
      i. Open the app
      ii. Click on “File->New->Data Source->Microsoft SQL Server



Name: @aws1

Comment:

Connection type: default Driver: Microsoft SQL Server

Host: DATABASE ENDPOINT Port: 1433

Instance:

User: DATABASE USER

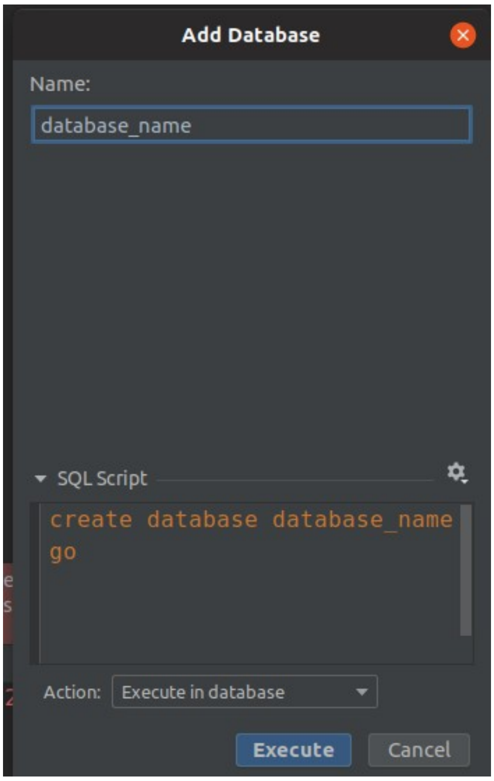
Password: •••••••••••• Save: Forever

Database:

URL: jdbc:sqlserver://DATABASE ENDPOINT:1433

Overrides settings above

iii. After adding the connection, create a new database by right clicking on the connection in the left panel -> New -> Database



Name: database_name

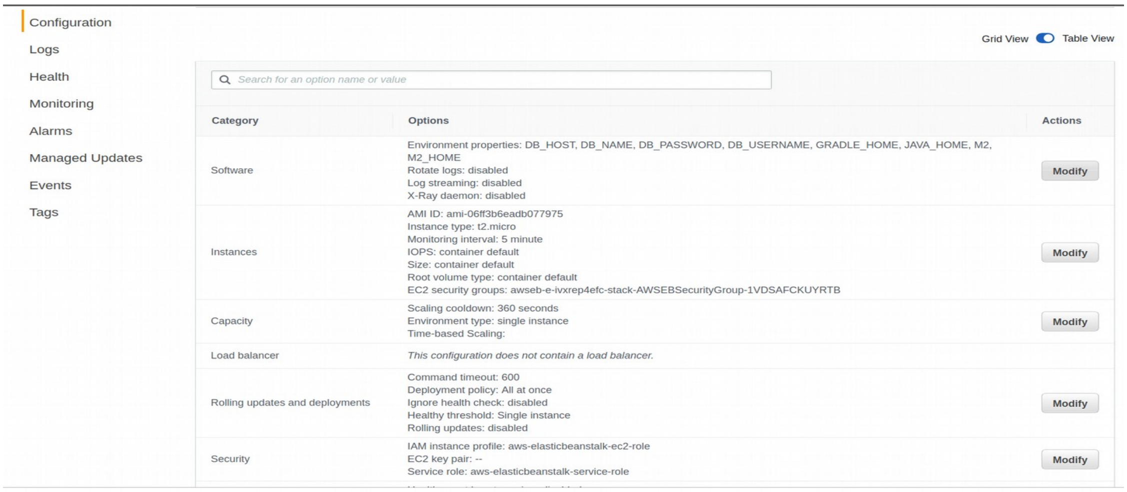
```
create database database_name
go
```

Action: Execute in database

7. **OPTIONAL BUT RECOMMENDED: remove the added inbound rule, so that the database will only be accessible from the AWS Cloud**

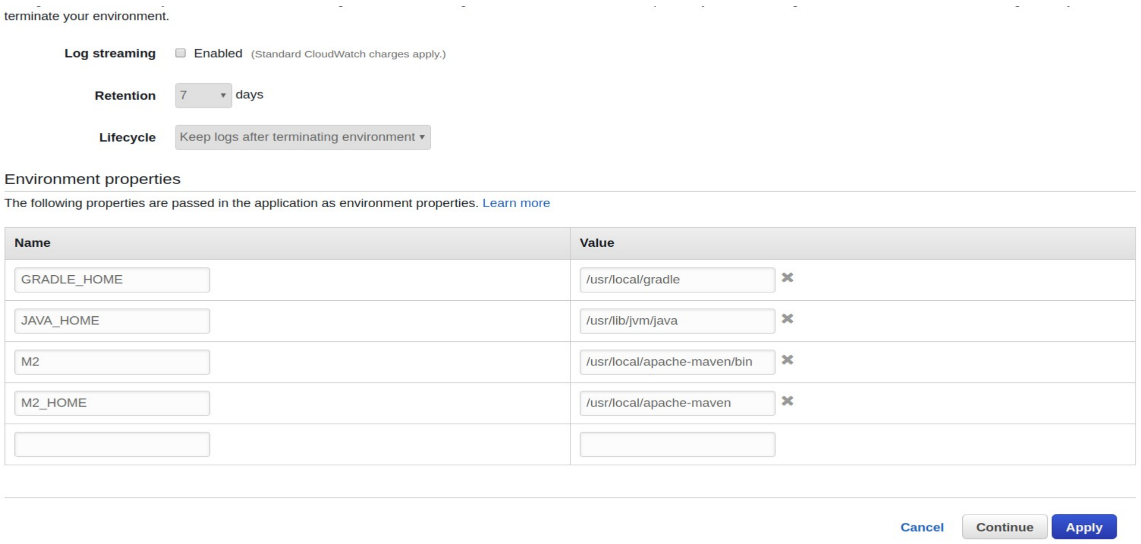
8. Override the environment variables on your app instance

   a. Go back to the Elastic Beanstalk dashboard, click on configurations and click on the “Modify” button in the Software row



Category	Options	Actions
Software	Environment properties: DB_HOST, DB_NAME, DB_PASSWORD, DB_USERNAME, GRADLE_HOME, JAVA_HOME, M2, M2_HOME Rotate logs: disabled Log streaming: disabled X-Ray daemon: disabled	Modify
Instances	AMI ID: ami-06ff3b6eadb077975 Instance type: t2.micro Monitoring interval: 5 minute IOPS: container default Size: container default Root volume type: container default EC2 security groups: awseb-e-ivxrep4efc-stack-AWSEBSecurityGroup-1VDSAFCKUYRTB	Modify
Capacity	Scaling cooldown: 360 seconds Environment type: single instance Time-based Scaling:	Modify
Load balancer	*This configuration does not contain a load balancer.*	
Rolling updates and deployments	Command timeout: 600 Deployment policy: All at once Ignore health check: disabled Healthy threshold: Single instance Rolling updates: disabled	Modify
Security	IAM instance profile: aws-elasticbeanstalk-ec2-role EC2 key pair: -- Service role: aws-elasticbeanstalk-service-role	Modify

b. Scroll down, to the environment variables section



terminate your environment.

Log streaming □ Enabled (Standard CloudWatch charges apply.)

Retention 7 days

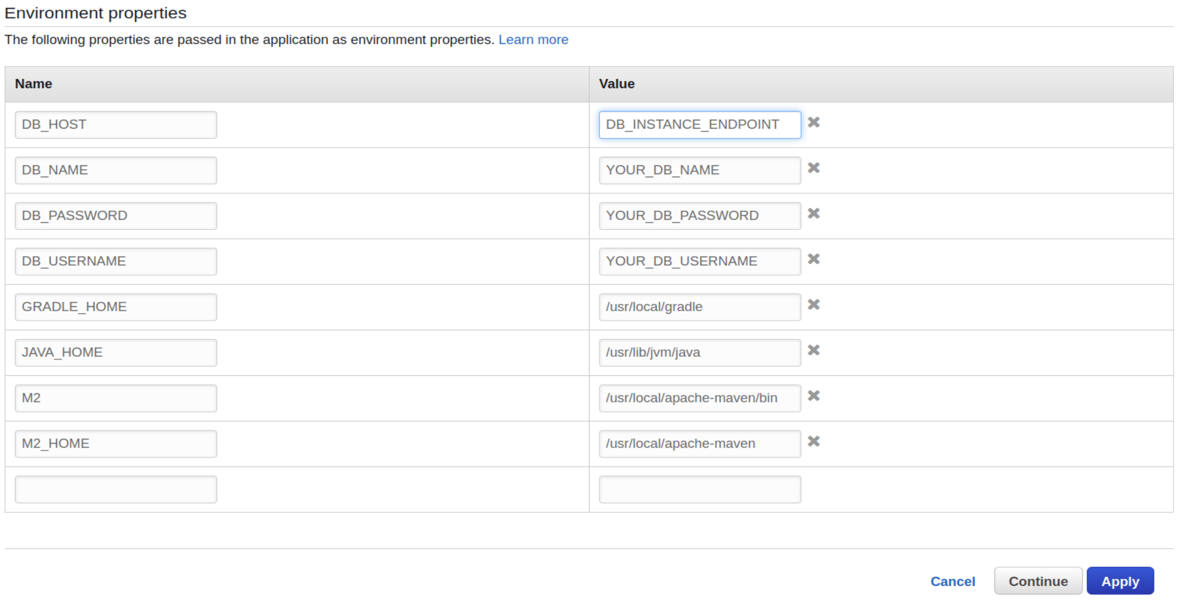
Lifecycle Keep logs after terminating environment

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
M2	/usr/local/apache-maven/bin
M2_HOME	/usr/local/apache-maven

c. Add your variables like in the following example, then click apply



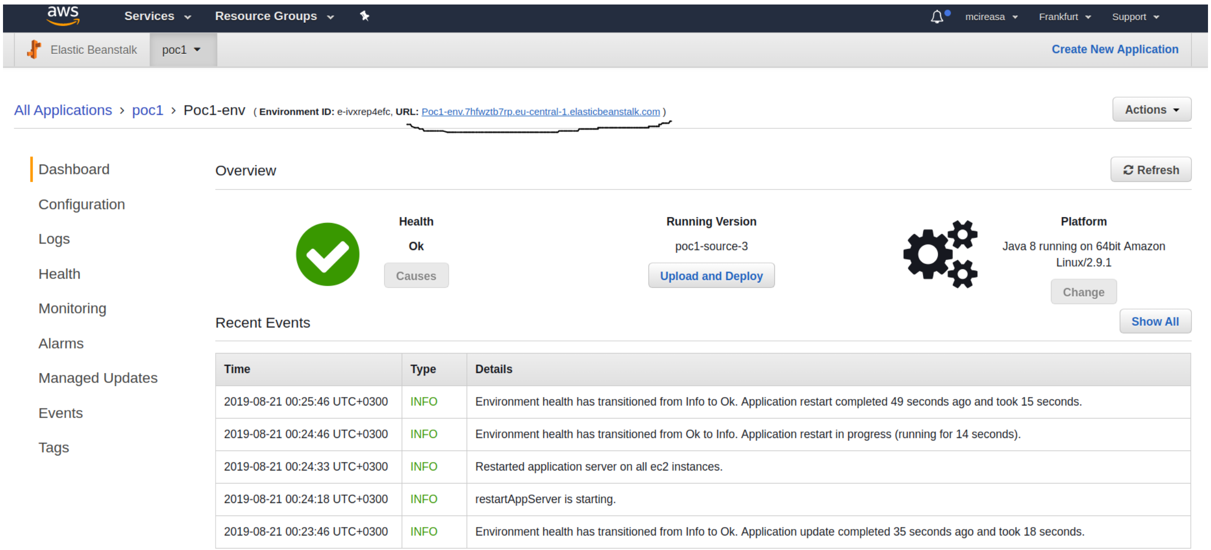
Environment properties

The following properties are passed in the application as environment properties. Learn more

Name	Value
DB_HOST	DB_INSTANCE_ENDPOINT
DB_NAME	YOUR_DB_NAME
DB_PASSWORD	YOUR_DB_PASSWORD
DB_USERNAME	YOUR_DB_USERNAME
GRADLE_HOME	/usr/local/gradle
JAVA_HOME	/usr/lib/jvm/java
M2	/usr/local/apache-maven/bin
M2_HOME	/usr/local/apache-maven

DB_USERNAME: the username provided at database creation
DB_PASSWORD: the password provided at database creation
DB_HOST: the ENDPOINT value copied from the database dashboard
DB_NAME: the name of the database you’ve just created using DataGrip

9. Return to the Elastic Beanstalk dashboard and after variables have been applied, click on the URL highlighted in the top of the page



All Applications › poc1 › Poc1-env (**Environment ID:** e-ivxrep4efc, **URL:** Poc1-env.7hfwztb7rp.eu-central-1.elasticbeanstalk.com)

Health
Ok

Running Version
poc1-source-3

Platform
Java 8 running on 64bit Amazon Linux/2.9.1

Time	Type	Details
2019-08-21 00:25:46 UTC+0300	INFO	Environment health has transitioned from Info to Ok. Application restart completed 49 seconds ago and took 15 seconds.
2019-08-21 00:24:46 UTC+0300	INFO	Environment health has transitioned from Ok to Info. Application restart in progress (running for 14 seconds).
2019-08-21 00:24:33 UTC+0300	INFO	Restarted application server on all ec2 instances.
2019-08-21 00:24:18 UTC+0300	INFO	restartAppServer is starting.
2019-08-21 00:23:46 UTC+0300	INFO	Environment health has transitioned from Info to Ok. Application update completed 35 seconds ago and took 18 seconds.

10. **Congrats! You've just deployed the web app on AWS.**
11. The default connection credentials are admin1, for both username and password **After the first login, create a new admin user and delete the default admin1 user**