# Installatie Ubuntu server

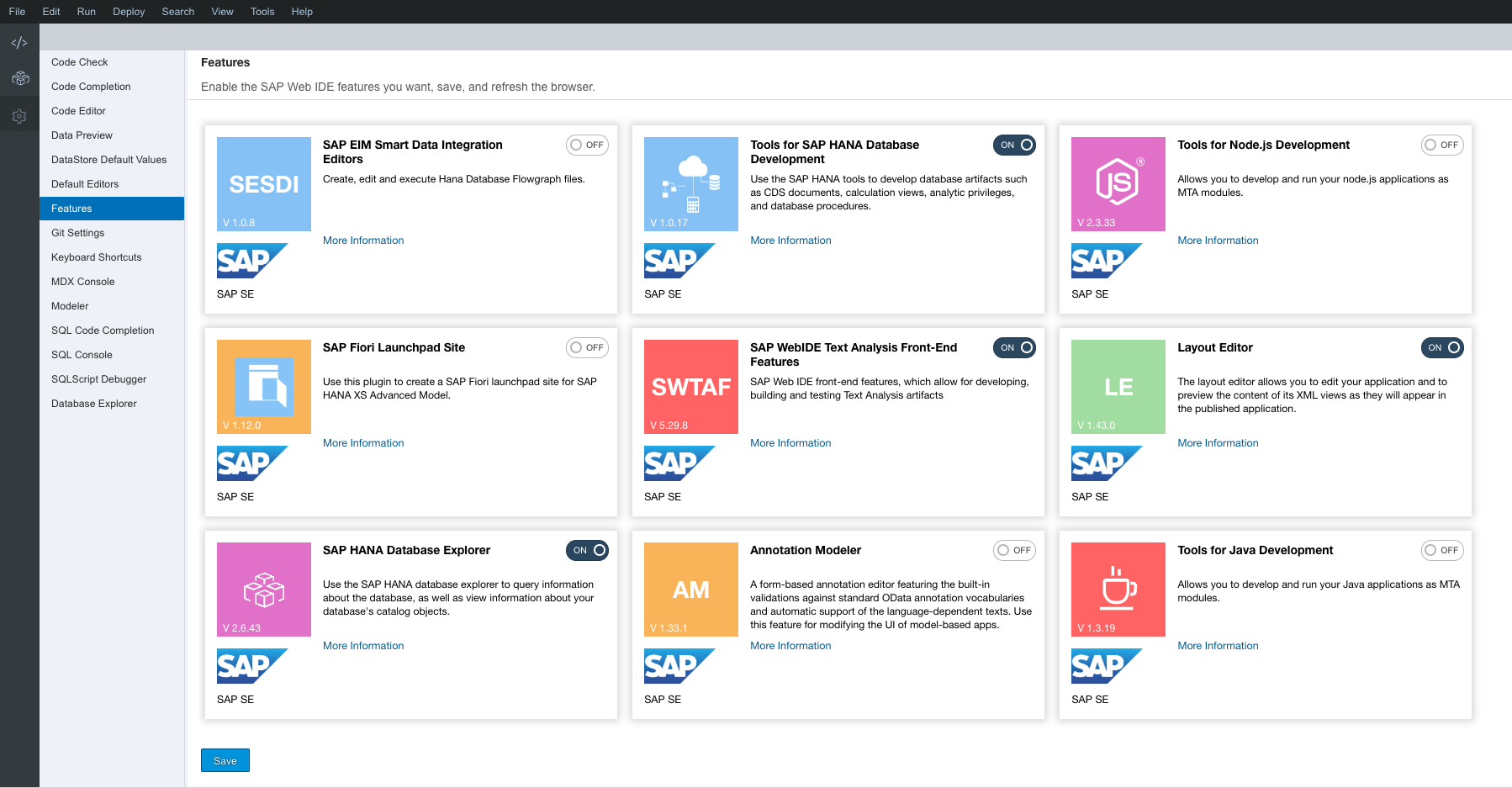
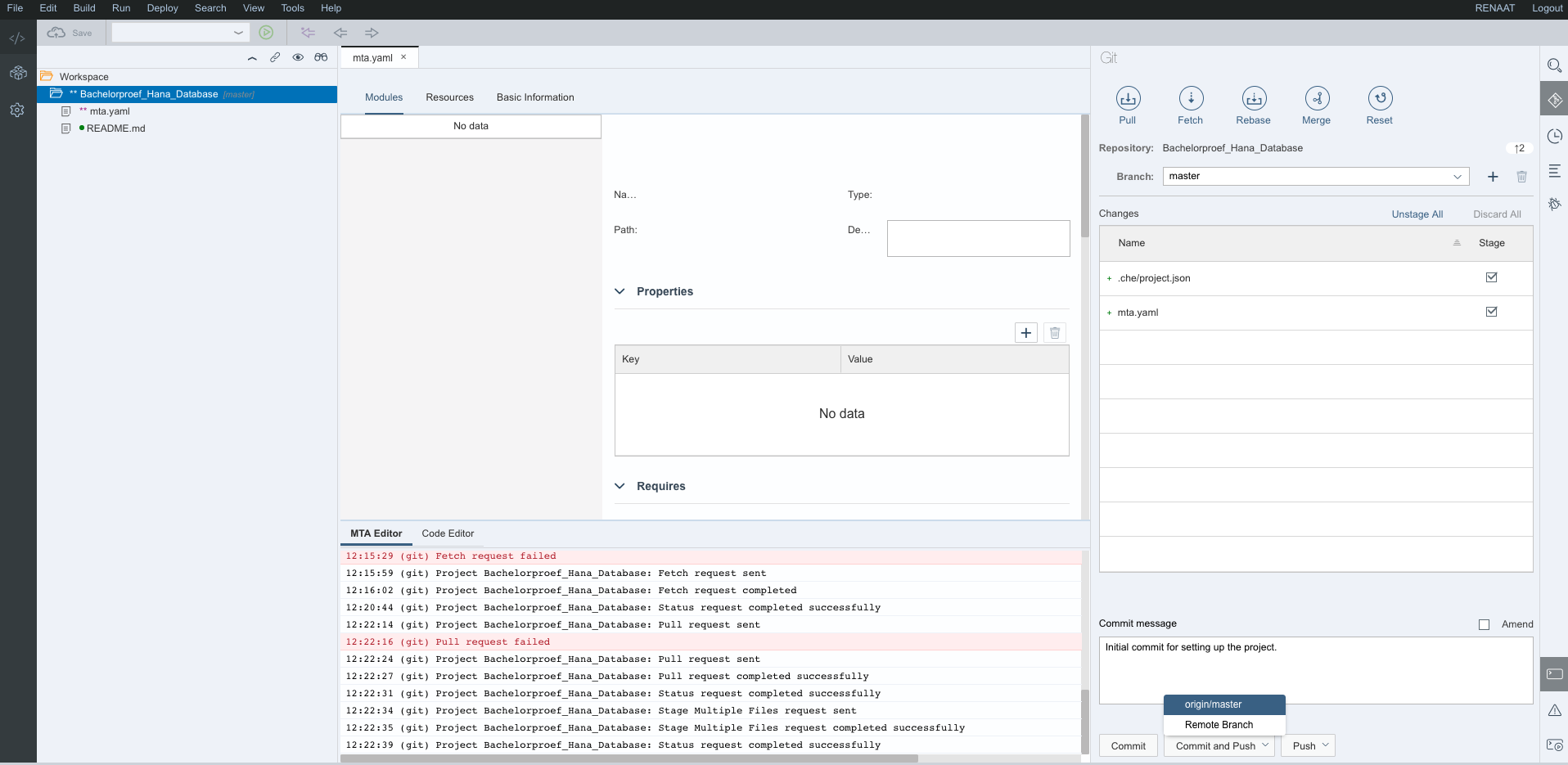
1. Typ **ssh root@{ip-adress van server}** van de server in de console om in te loggen in de server
2. Typ nadien het paswoord van de server. Er wordt meteen gevraagd om het te vernieuwen. Kies een nieuw paswoord.
3. Eens ingelogd op de sshd server moeten er enkele zaken aangepast worden aan de ssh configuratie in de file '/etc/ssh/sshd\\_config'. Open de file (via cat command) en zet de PermitRootLogin op yes zodat de root-gebruiker kan inloggen.
4. Zet StrictMode ook op yes zodat niemand kan inloggen als de authenticatie documenten leesbaar zijn voor iedereen.
5. Voor het verdere wordt verwacht dat er een SSH is opgezet op de client computer waarmee gewerkt wordt. Indien dit nog niet gebeurd is, neem een kijkje op: <https://www.ssh.com/ssh/key/>.
6. Kopieer de SSH key van de client computer naar de server door op de client computer een console te openen, naar de root folder te gaan (**cd ~**) en **ssh-copy-id root@{ip-adress van server}** te typen.
7. Ga terug naar de file '/etc/ssh/sshd\\_config' op de sshd server.
8. Verander PasswordAuthentication en ChallengeResponseAuthentication naar no en PubKeyAuthentication naar yes. Dit zorgt ervoor dat het niet meer mogelijk is om via een paswoord in te loggen, enkel via ssh.
9. Sla de ‘sshd\_config’ file op
10. Herstart de ssh daemon door **sudo systemctl restart ssh** te typen in de concole van de sshd server
11. Zet nu de UFW firewall op in de sshd server door **ufw app list** te typen.   
    Je ziet dat OpenSSH beschikbaar is als applicatie.
12. **Ufw allow OpenSSH** zorgt ervoor dat de firewall wordt opgezet.
13. Om gebruik te maken van de firewall, typ **ufw enable** en kies voor **yes**.
14. Om te controleren of alles daadwerkelijk gelukt is moet je **ufw status** typen

# Source Control

Maak een nieuwe private repository in BitBucket:

1. Kies een gepaste naam, vink ‘This is a private repository’ aan en kies voor Git.
2. Kopieer het adres dat gegeven wordt om een clone te maken van deze repository

# Database

1. Ga naar de Web IDE voor HANA development
2. Maak een nieuw project en kies voor de Multi-Target Application Project template
3. Ga naar de instellingen en kies voor het onderdeel ‘Features’
4. Vink dezelfde features aan zoals in onderstaande figuur
5. Ga naar de Git Settings en voeg daar je gegevens in zodat er verbinding gemaakt kan worden
6. Ga terug naar de Development omgeving, klik met je rechtermuisknop op het project en kies voor Build
7. Om het project aan het account op Bitbucket te linken klikt u op het project met de rechtermuisknop, gaat u naar 'Git' en dan 'Initialize Local Repository'
8. Link de gemaakte repository aan het project door via rechter muisklik op het project op Git en dan Set Remote te klikken
9. Plak de gekopieerde Bitbucket link in het veld voor URL
10. Geef je wachtwoord van Bitbucket in
11. Klik OK wanneer het 'Changes Fetched' window opent
12. Commit de changes naar de repository zoals in onderstaande figuur

### Database Module

1. Maak een HANA Database Module door op het project met de rechtermuisknop te klikken, bij ‘New’ voor ‘SAP HANA Database Module’ te kiezen
2. Geef de module de naam: **db**
3. Kies **Bachelor\_Hana\_Database.db** als Namespace, **Voorbeeldapplicatie** als Schema Name en **2.0 SPS 03** als SAP Hana Database Version
4. Klik op de src map in de db module met de rechtermuisknop en kies voor ‘New’ en nadien voor ‘HDB CDS Artifact’
5. Geef **data** in als naam
6. Open de gemaakte file met de Code Editor en voeg volgende data in

namespace Bachelorproef\_Hana\_Database.db.data;

context data {

/\*@@layout{"layoutInfo":{"x":-467,"y":-239.5}}\*/

entity Artiest {

key Id : Integer generated by default as identity(start with 1 increment by 1 no minvalue maxvalue 2999999999 no cache no cycle);

Naam : String(256);

JaarVanOorsprong : Integer;

}

technical configuration {

column store;

};

};

1. Maak de database door in de meest linkse tab op de knop 'Database Explorer' te klikken.
2. Klik op het +-teken om een database aan te maken. Kies uit de lijst de optie met jouw naam en geef een realistische naam bij het veld 'How to Show in Display'. Laat de rest van de setting zoals ze zijn.
3. Ga terug naar 'Development' omgeving (in meest linkse tab)
4. Klik met de rechter muisknop op de 'data.hdbcds' file kies de optie 'Build Selected Files'.
5. Er moet natuurlijk ook data in de database zitten om deze te gebruiken in de SAPUI5 applicatie. Een manier om dit te doen is via het uploaden van csv-bestanden waar de data gescheiden is door een komma. Maak een nieuwe file, 'load.hdbtabledata' genaamd, in de data map van de db-module en voeg volgende inhoud toe:

{

"format\_version": 1,

"imports": [{

"target\_table": "Bachelorproef\_Hana\_Database.db.data::data.Artiest",

"source\_data": {

"data\_type": "CSV",

"file\_name": "Bachelorproef\_Hana\_Database.db.data::Artiest.csv",

"has\_header": false,

"dialect": "HANA",

"type\_config": {

"delimiter": ","

}

},

"import\_settings": {

"import\_columns": ["Id",

"Naam",

"JaarVanOorsprong"]

},

"column\_mappings": {

"Id": 1,

"Naam": 2,

"JaarVanOorsprong": 3

}

}]

}

1. Maak een csv-bestand aan en voeg volgende data toe

1,Nirvana,1989

2,Steak Number Eight,2006

3,Placebo,1995

4,Tool,1992

5,Shame,2016

6,Queens Of The Stone Age,1996

7,Madensuyu,2005

8,Kyuss,1990

9,King Gizzard And The Lizard Wizard,2014

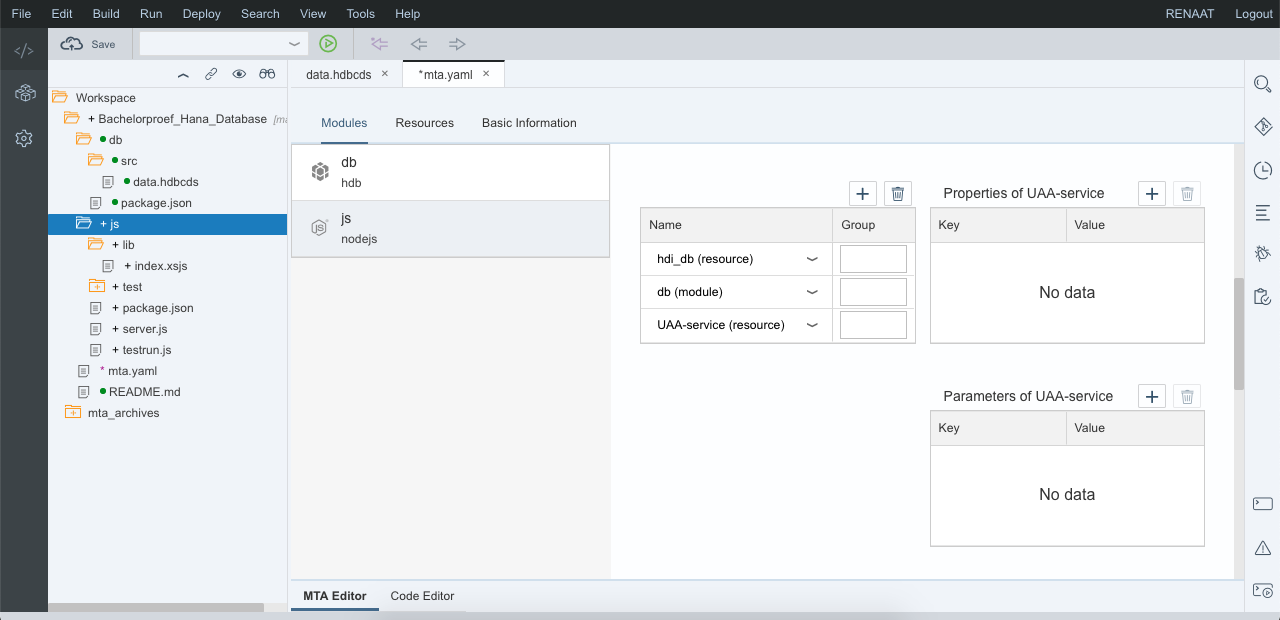
10,Kapitan Korsakov,2005

11,Jack Johnson,2004

1. Importeer de gemaakt file in de map ‘data’ van de ‘db-module’
2. Build de ‘db-module’ nogmaals
3. Controleer of er effectief data aanwezig is door naar de 'Database Explorer' te gaan
4. Open de map 'tables'
5. Links vanonder verschijnt een venster met de verschillende tabellen. Er rest ons enkel nog op de juiste tabel te klikken met de rechter muisknop en kies dan 'Generate SELECT statement'.
6. Klik op de groene play knop om het statement uit te voeren
7. Deze veranderingen kunnen best gepusht worden naar Bitbucket

### Node.js module

Maak een Node.js module om de data te exporteren als een OData service

1. Ga terug naar de instellingen van de Features en zorg ervoor dat de feature ‘Tools For Node.js Development’ beschikbaar is
2. Save de instellingen en laad het project opnieuw
3. De volgende stap is de module maken door op het project met de rechtermuisknop te klikken, new en dan Node.js Module kiezen. Geef het een goede naam (js is good practice) en zorg ervoor dat Enable XSJX support aangevinkt is alvorens op Finish te klikken
4. Databeveiliging is zeer belangrijk. Binnen HANA wordt er gebruik gemaakt van UAA (User Account en Authentication) om de Node.js module te beveiligen. Deze service moet samen met de database module en de HDI container, achter de db module, toegevoegd worden als resource zoals te zien is in onderstaande figuur
5. Maak in de ‘js-module’ in de ‘lib’ folder een nieuwe file en geef het volgende naam: **xsodata/services.xsodata**
6. Kopieer volgende code in deze file

service {

"Bachelorproef\_Hana\_Database.db.data::data.Artiest" as "Artiesten";

}

1. Maak een xsjs service door in de ‘lib’ folder een nieuwe file aan te maken met volgende naam: **xsjs/hdb.xsjs**
2. Kopieer de code die u hier ziet in deze file

/\*eslint no-console: 0, no-unused-vars: 0, dot-notation: 0\*/

/\*eslint-env node, es6 \*/

"use strict";

var conn = $.hdb.getConnection();

var query = "SELECT FROM Bachelorproef\_Hana\_Database.db.data::data.Artiest { " +

" Id as \"ArtiestId\", " +

" Naam as \"ArtiestenNaam\", " +

" JaarVanOorsprong as \"Jaar\" " +

" } ";

var rs = conn.executeQuery(query);

var body = "";

for (var item of rs) {

body += item.ArtiestenNaam + "\t" + item.Jaar + "\n";

}

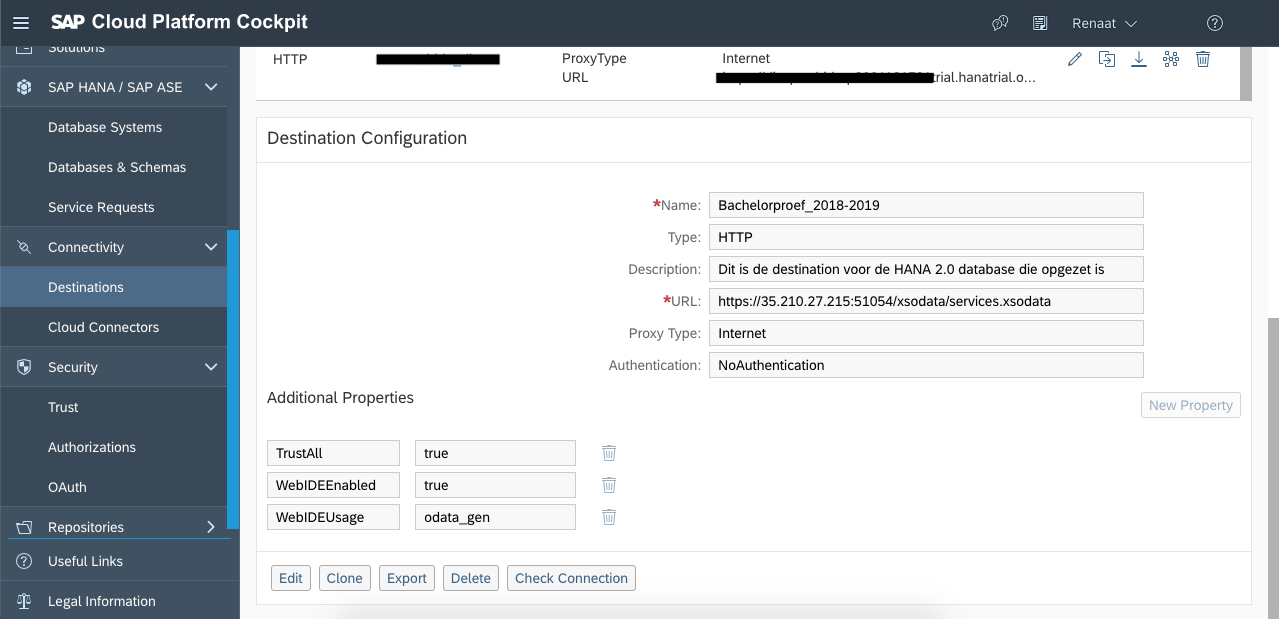
$.response.setBody(body);

$.response.contentType = "application/vnd.ms-excel; charset=utf-16le";

$.response.headers.set("Content-Disposition",

"attachment; filename=Excel.xls");

$.response.status = $.net.http.OK;

1. Build de nieuwe Node.js module door op de module met de rechtermuisknop te klikken, Run, Run As en dan Node.js Application te kiezen
2. Klik onderaan het scherm in het nieuwe venster op de link
3. De index.html wordt geopend in de browser. Het is mogelijk om de metadata te bekijken door in de url de '/index.html' te vervangen door '/xsodata/services.xsodata/$metadata'
4. Om de OData van de entiteit in de database te bekijken moet '/$metadata' vervangen worden door '/Artiesten'. Op deze manier verschijnt de OData van de tabel Artiest
5. Als voorgaande stappen gelukt zijn, kan men de OData-service gebruiken als een destination in SAP Cloud Platform. Om dit te doen zijn er enkele stappen nodig: ga naar SAP Cloud Platform
6. Ga naar destinations en voeg een nieuwe toe met zoals in onderstaande figuur

# UI5 applicatie

Maak een nieuwe applicatie, startend vanaf een template:

1. Kies SAPUI5 Application en geef het een correcte naam (Bachelorproef\_2018-2019)
2. Kies XML als formaat voor de eerste view en geef het een juiste naam (App bijvoorbeeld)
3. Klik op finish
4. In de neo-app.json file, voeg volgende code toe aan de routes array:

{

"path": "/xsodata/services.csodata ",

"target": {

"type": "destination",

"name": "Bachelorproef\_2018-2019"

},

"description": "Bachelorproef\_2018-2019"

}

1. Voeg volgende code toe aan het ‘sap.app’ object in manifest.json

"dataSources": {

"mainService": {

"uri": "/xsodata/services.xsodata/",

"type": "OData",

"settings":{

"odataVersion": "2.0"

}

}

}

1. In de manifest.json moet ook nog het default model toegevoegd worden aan het object ‘models’

“” : { “dataSource”: “mainService” }

1. Verander de code in App.view.xml door volgende code

<mvc:View controllerName="Bachelorproef\_2018-2019.Bachelorproef\_2018-2019.controller.App"

xmlns:mvc="sap.ui.core.mvc" displayBlock="true” xmlns="sap.m"

xmlns:core="sap.ui.core" xmlns:l="sap.ui.layout" xmlns:f="sap.ui.layout.form">

<Shell id="shell">

<App id="app">

<pages>

<Page id="page" title="{i18n>title}">

<content>

<IconTabBar id="idTopLevelIconTabBar"

class="sapUiResponsiveContentPadding">

<items>

<IconTabFilter text="Data Binding" key="db">

<List headerText="Artiesten"

width="auto" items="{/Artiesten}"

mode='SingleSelectMaster'

selectionChange="onItemSelected">

<items>

<ObjectListItem

title="{Naam}"

intro="{JaarVanOorsprong}">

</ObjectListItem>

</items>

</List>

</IconTabFilter>

<IconTabFilter>

<l:VerticalLayout

class="sapUiContentPadding" width="100%">

<l:content>

<MessageStrip

text="Als voorbeeld"

showIcon="true"

showCloseButton="true"

class="sapUiMediumMarginBottom">

</MessageStrip>

</l:content>

</l:VerticalLayout>

</IconTabFilter>

</items>

</IconTabBar>

</content>

</Page>

</pages>

</App>

</Shell>

</mvc:View>

1. Wanneer op een item geklikt wordt in de lijst verschijnt er een Message Toast met de leeftijd van de Artiest. Deze wordt berekend in de AppController zoals u hier kan zien.

sap.ui.define([

'sap/m/MessageToast',

"sap/ui/core/mvc/Controller"

], function (MessageToast, Controller) {

"use strict";

return Controller.extend("Bachelorproef\_2018-2019.Bachelorproef\_2018-2019.controller.App", {

onItemSelected: function(oEvent) {

var oSelectedItem = oEvent.getParameter("listItem");

var msg = oSelectedItem.getTitle() + " is " + this.calculateYearFromNow(oSelectedItem.getIntro()) + " jaar oud!";

MessageToast.show(msg);

},

calculateYearFromNow: function(jaar) {

return (new Date().getFullYear() - jaar);

}

});

});

1. U kan nu de app runnen en testen.

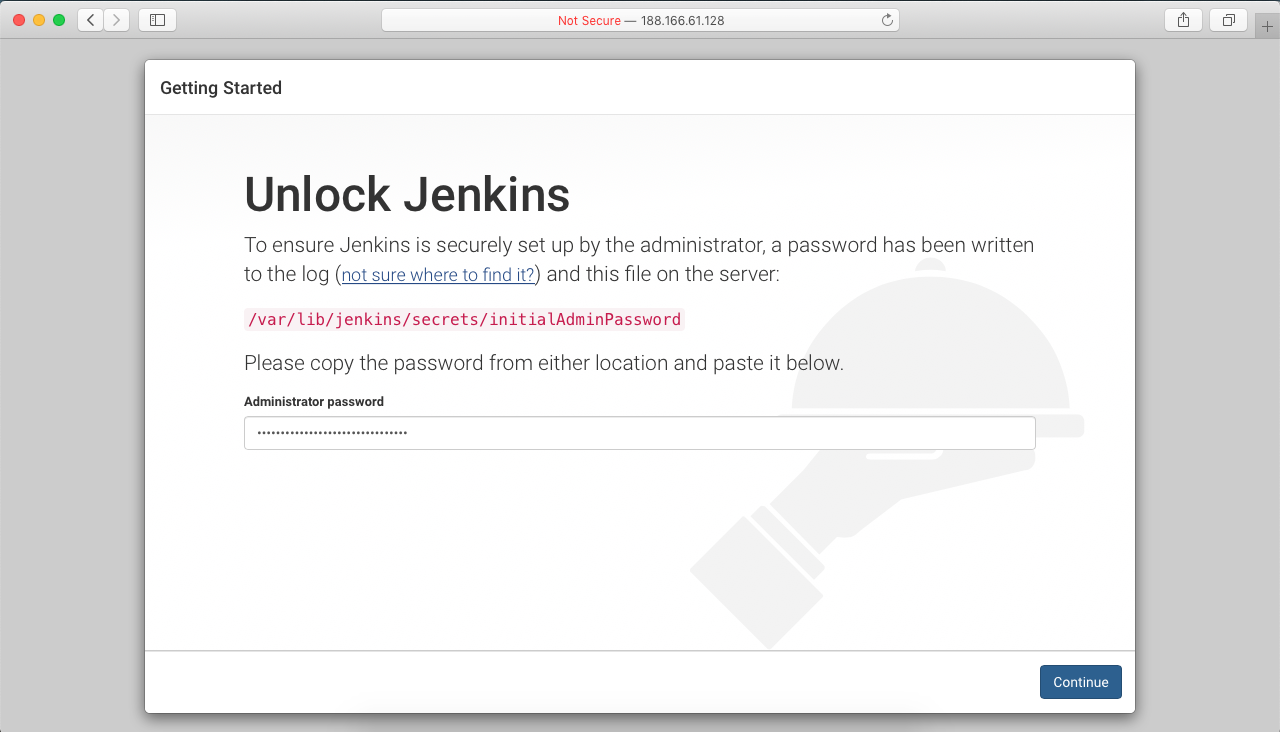
## Proof-of-concept

### Jenkins

Installatie van Jenkins op de server

1. Omdat Jenkins op Java draait is het nodig om Java 8 te installeren. Dit is dan ook de eerste stap in het proces.
2. Log in de server door ssh root@188.166.61.128 in de console op jouw computer te typen
3. Voer het commando **sudo apt install openjdk-8-jdk** intypen om JAVA 8 te installeren. De server vraagt bevestiging om te downloaden en te installeren, er moet enkel maar **y** worden getypt en op return gedrukt worden om te bevestigen
4. Nu kunnen we overgaan tot de installatie van Jenkins op de server. We maken gebruik van de packages die Jenkins onderhoud om de software te installeren op Ubuntu door **wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -** in te typen
5. Eens dit gebeurd is moeten we de Debian package repository toevoegen aan de server's sources.list door **sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'** te typen
6. Typ **sudo apt update**
7. Installeer Jenkins door **sudo apt install Jenkins** in te typen

Maak gebruik van Jenkins

1. Typ **sudo systemctl start jenkins** in
2. Wanneer na het invoeren van commando **sudo systemctl status jenkins** active verschijnt wil dit zeggen dat Jenkins runt op de Ubuntu server
3. Eens Jenkins gestart is moeten we die kunnen aanspreken vanuit een browser, daarom moeten we de regels van de firewall wat aanpassen. Standaard draait Jenkins op poort 8080. Deze moet geopend worden op de server door het commando **sudo ufw allow 8080** in te voeren
4. Om Jenkins in de browser te configureren moet je naar http://188.166.61.128:8080 surfen. De cijfers voor de :8080 staan voor het ip-adres van de Ubuntu server, de :8080 is de poort waar je naartoe wil gaan. Hier draait - zoals in de vorige stappen geconfigureerd - Jenkins op.
5. Het paswoord dat gevraagd wordt is terug te vinden in de 'initialAdminPassword' file dat zich in de map '/var/lib/jenkins/secrets/' bevindt op de server. Het is voldoende om op de server volgend commando in te voeren: **sudo cat /var/lib/jenkins/secrets/initialAdminPassword**. Het paswoord dat als uitvoer komt moet gekopieerd worden en geplakt worden in het invoerveld zoals in onderstaande figuur te zien is

1. Klik op ‘Install suggested plugins’ om de plugins te installeren
2. Klik op ‘Continue as admin’ om door te gaan als admin user
3. Klik op ‘Save and Finish' wanneer je de url gecontrolleerd hebt op zijn juistheid
4. Nadien kies je voor 'Start using Jenkins'
5. Het dashboard van Jenkins wordt geopend