BFFCORE Installation & Deployment Guide

*Detailed installation instructions for BFFCORE*

Table of Contents

1 PRE-REQUISITES/SOFTWARES TO BE INSTALLED 3

1.1 JAVA/JDK 11 INSTALLATION AND SET-UP 3

1.2 BUILD TOOL INSTALLATION AND SET-UP 3

1.3 DATABASE SERVER AND TOOLS INSTALLATION AND SET-UP 4

2 APPLICATION INSTALLATION AND SET-UP 7

2.1 DATABASE CREATION, CONFIGURATION AND SET-UP 7

2.2 APPLICATION RESOURCES CONFIGURATION AND SET-UP 8

2.3 CONFIGURING AND RUNNING APPLICATION ON HTTPS (OPTIONAL) 10

2.4 INSTALL LIBRARIES & GENERATE BFFCORE APPLICATION BOOTABLE JAR 11

2.5 INITIAL DATABASE SET-UP SCRIPTS 11

2.6 ACCESSING AND RUNNING THE APPLICATION 12

2.7 IMPORTING REGISTRIES AND API POST-PROCESSORS FOR PICK FLOW (OPTIONAL) 13

2.8 PROVIDING SUPER USER PRIVELEGES TO WMS USERS 15

2.9 ADDING LOMBOK SUPPORT TO STS/Eclipse IDE (OPTIONAL) 18

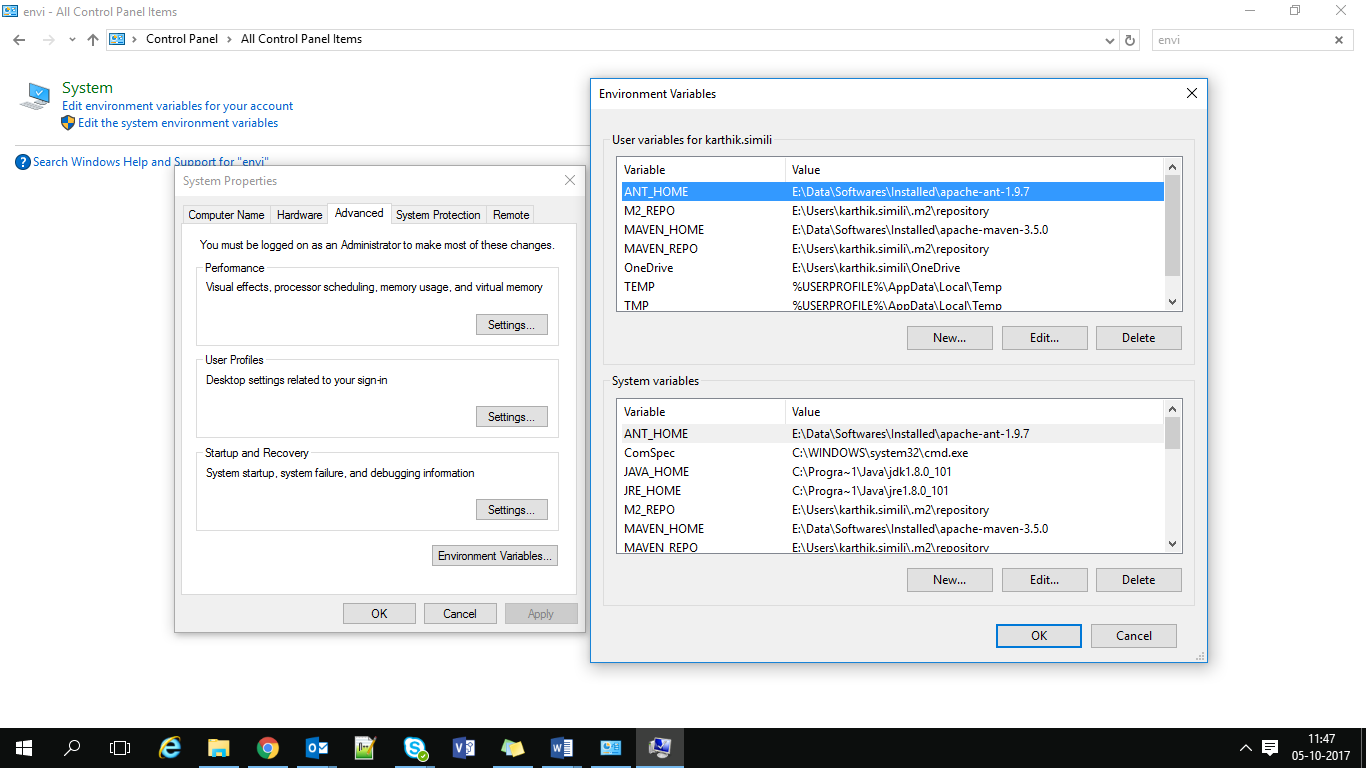
# PRE-REQUISITES/SOFTWARES TO BE INSTALLED

## JAVA/JDK 11 INSTALLATION AND SET-UP

1. Download the latest Java 11 / JDK 11 software from the location below –

<https://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase11-5116896.html>

1. Set *JAVA\_HOME* environment variable in system properties to the software installation path above. Screenshot below –

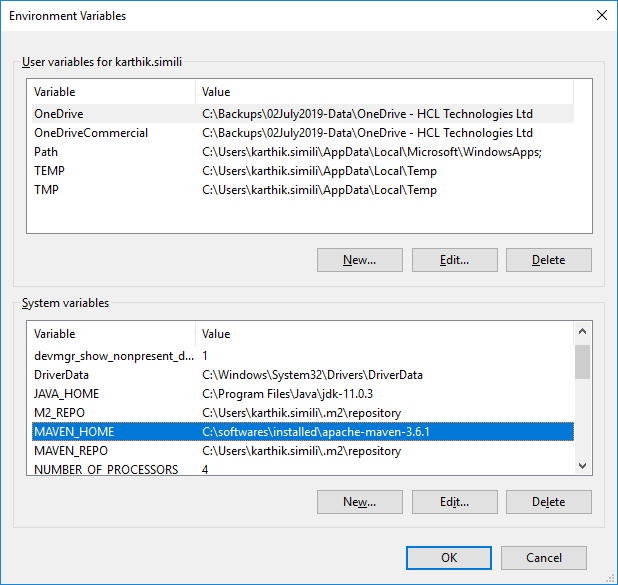


## BUILD TOOL INSTALLATION AND SET-UP

1. Download MAVEN 3.6.x latest version from the location below –

<https://maven.apache.org/download.cgi>

1. Set the following in SYSTEM properties – environment variables –
   1. MAVEN\_HOME and M2\_HOME environment variables in SYSTEM properties to the MAVEN installation directory/folder
   2. M2\_REPO to any folder where the application installation JARs/libraries will be stored as shown below –



## DATABASE SERVER AND TOOLS INSTALLATION AND SET-UP

1. Download SQL server database v2016 and above, preferrably *SQL Server Developer/Enterprise edition* from the location below –

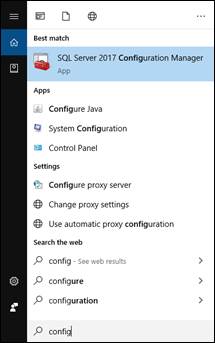
<https://go.microsoft.com/fwlink/?linkid=853016>

1. Download SQL Server management studio (SSMS) from the location below –

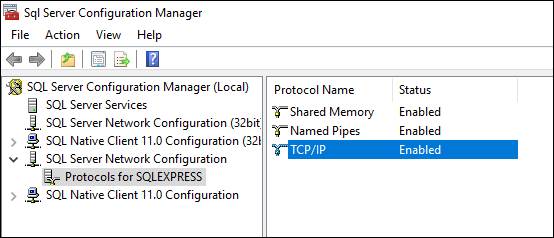
Link: <https://aka.ms/ssmsfullsetup>

Base link: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15#download-ssms>

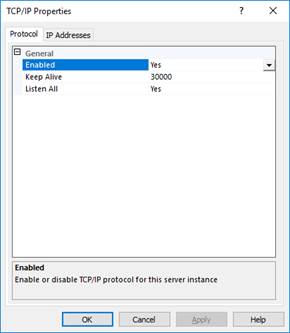
1. Open ‘SQL server configuration manager’ by searching using windows search as shown below –



1. In the resultant screen, select *‘SQL Server Network Configuration – Protocols for SQL Express’* as shown below –

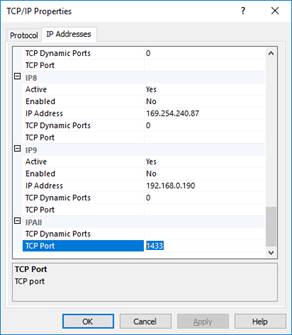


1. Right click on TCP/IP and select ‘Properties’, the pop-up below will be displayed –

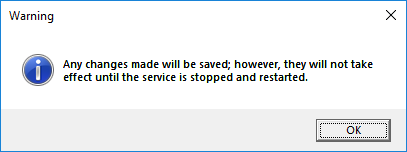


Select ‘Yes’ against the ‘Enabled’ flag as shown above.

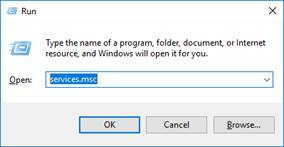
1. Now click on the ‘IP Addresses’ tab, scroll down and set TCP port to 1433 against the ‘*IPAll’* entry as shown below –



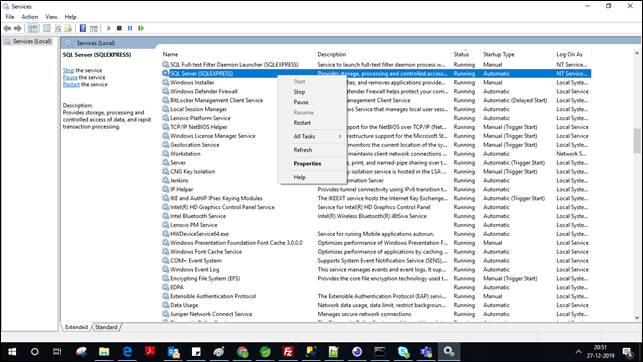
1. The following warning pop-up will be shown to restart SQL Server. Click **OK** in this pop-up and then again **OK** on main pop-up to accept the changes –



1. Now type *services.msc* in Start  Run as shown below –



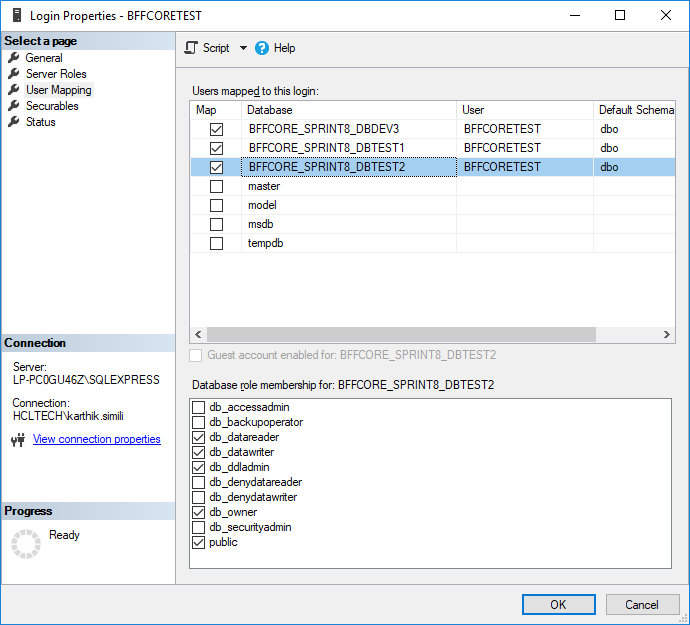
1. Search for the ‘SQL Server’ service and restart it as shown below –

`

# APPLICATION INSTALLATION AND SET-UP

## DATABASE CREATION, CONFIGURATION AND SET-UP

1. Open 'SQL Server Management Studio' downloaded as per steps outlined in previous section and perform the following steps -
2. Create a new schema with a name (say **'BFFCORE\_DB'**) which will serve as the BFFCORE datastore/database in the SQL server instance
3. Under "Security -> Logins" in the SQL Server instance, in the "General" tab/sub-menu, create a new user **'BFFCOREUSER'** and provide a valid password.
4. Under "User Mapping", in the "Users mapped to this login" section, provide the User as **'BFFCOREUSER’** and 'Default Schema' as 'dbo'. Additionally, check/select the 'public', 'db\_owner', 'db\_ddladmin', 'db\_datawriter', 'db\_datareader' roles in the 'Database role membership for: **'BFFCOREUSER’** section



1. Download the latest BFFCORE source code from SCM (GIT) repository.
2. Open *'application.properties'* file under *'bffWebManagement/src/main/resources*' folder of the BFFCORE source code and replace the placeholders below in **bold** appropriately –

spring.datasource.url=jdbc:sqlserver*://<BFF Core DB IP Address>:<BFFCORE DB port>*;databaseName=***<BFFCORE DB NAME>***

spring.datasource.driver-class-name=com.microsoft.sqlserver.jdbc.SQLServerDriver

spring.datasource.username=***<BFFCORE DB user id>***

spring.datasource.password=***<BFFCORE DB password>***

## APPLICATION RESOURCES CONFIGURATION AND SET-UP

1. Create OPEN ID / BASIC (NATIVE) AUTH configuration/profile -

**OPEN ID configuration/profile creation**

From the latest BFFCORE source code downloaded in previous section, open *'application-devoidc.properties'* under bffWebManagement/src/main/resources folder and make the following changes -

1. Configure application port. Replace the *placeholder* below in **bold** -

#Basic spring application properties

server.port=***<BFFCORE Application Server port>***

1. WMS product API server / docker instance. Replace the *placeholders* below -

productapi.base.url=***<WMS server IP address>****:****<port>***

1. OPEN ID configuration properties –

#WMS simple-oidc-provider Authentication properties

#Application authentication scheme: allowed Enum\_Values @AuthScheme {BASIC\_AUTH, OPENID;}

app.scheme=OPENID

app.openid.providerlocation=***<OIDC provider location>***

app.openid.clientId=***<OIDC Client id>***

app.openid.clientSecret=***<OIDC Client secret>***

app.openid.redirectUri=***<OIDC Redirect URI>***

app.openid.authenticationMethod=**client\_secret\_basic**

app.openid.audience=***<OIDC Audience>***

app.openid.scope=**openid**

Sample with Azure B2C sandbox provider –

app.openid.providerlocation=**https://jdacldsbxb2c.b2clogin.com/jdacldsbxb2c.onmicrosoft.com/B2C\_1A\_WMSTest\_signin/v2.0**

app.openid.clientId=**a63e1a4e-b790-4b7b-91a7-7a35e4e8a500**

app.openid.clientSecret=**yajvSU?(^06ro!9==4(8"T&c**

app.openid.redirectUri=**https://3.136.138.129/oauth**

app.openid.authenticationMethod=**client\_secret\_basic**

app.openid.audience=**a63e1a4e-b790-4b7b-91a7-7a35e4e8a500**

**BASIC/NATIVE AUTH configuration/profile file creation**

From the latest BFFCORE source code downloaded in previous section, open *'application-devbasicauth.properties'* under bffWebManagement/src/main/resources folder and make the following changes -

1. Configure application port. Replace the *placeholder* below in **bold** -

#Basic spring application properties

server.port=***<BFFCORE Application Server port>***

1. WMS product API server / docker instance and URL scheme (http/https). Replace the *placeholders* below –

productapi.url.scheme=***<http/https>*** *(WMS Docker is currently on* ***http****)*

productapi.base.url=***<WMS server IP address>:<port>***

1. Open *'log4j2.xml'* file under *bffWebManagement/src/main/resources* folder replace the *APP\_LOG\_ROOT* property value (*placeholder*) below to the *‘application logs files’* root path in local folder -

<Properties>

…….

<Property name="APP\_LOG\_ROOT">***<Applicaion Log file path>***</Property>

</Properties>

1. If analytics needs to be enabled, open *'application-<profile>.properties’* created above and set **google analytics server** properties –

app.analytics.enabled=**false** *(Flag to enable / disable analytics. Default is set to* ***false****)*

app.analytics.url.scheme=***<http/https>*** *(Default is http)*

app.analytics.server.api.host=www.google-analytics.com

app.analytics.server.api.contextpath=/collect

app.analytics.server.api.version=1

app.analytics.server.api.trackid=***<Tracking id>*** (Ex: ***UA-159043447-2)***

app.analytics.server.api.pagehittype=pageview

app.analytics.server.api.timinghittype=timing

A sample common google account for both Admin UI and BFFCORE has been created for recording analytics - User name / password: [*whmjda@gmail.com*](whmjda@gmail.com) */ Jda@1234*. Application properties to be used for this account are below –

app.analytics.enabled=true

app.analytics.url.scheme=**http**

app.analytics.server.api.host=www.google-analytics.com

app.analytics.server.api.contextpath=/collect

#analytics.server.api.contextpath=/batch

app.analytics.server.api.version=1

app.analytics.server.api.trackid=**UA-159043447-2**

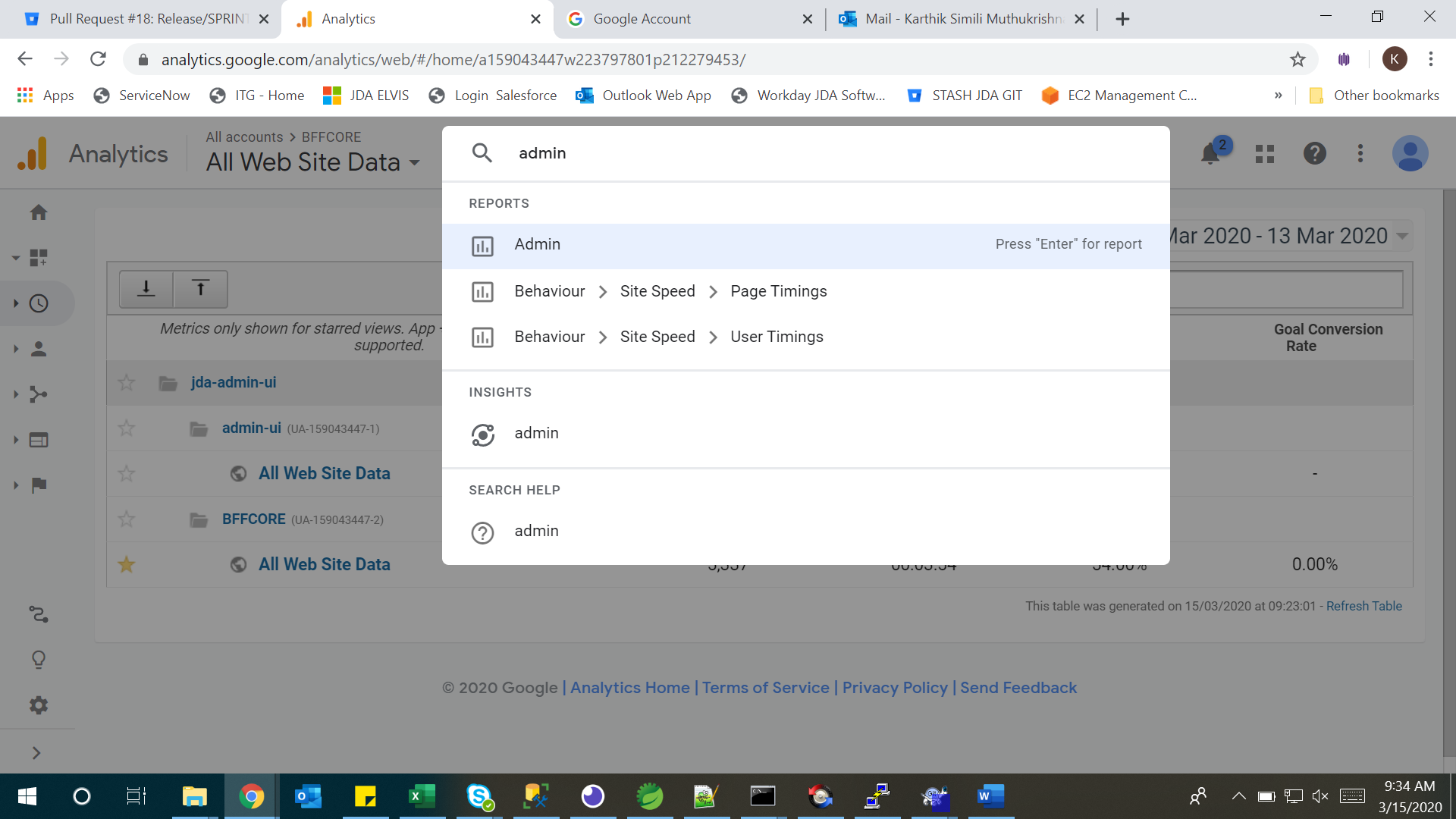
app.analytics.server.api.pagehittype=pageview

app.analytics.server.api.timinghittype=timing

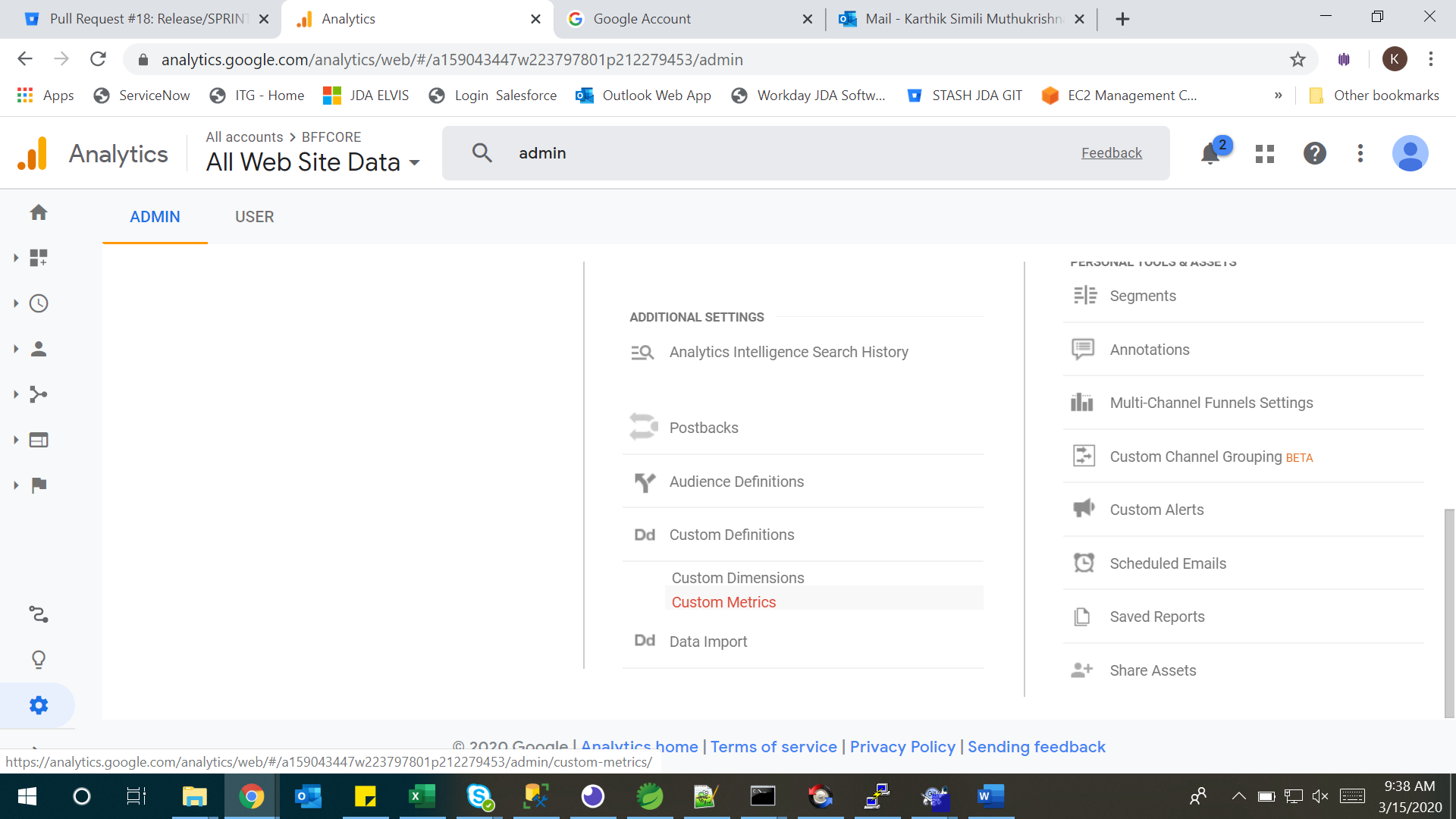
The relevant custom and calculated metrics have already been created in the google account above for recording BFFCORE API Hits and response times. In case a fresh google account is used to record BFFCORE metrics (API Hits and Response Times), some additional steps to add custom ‘*Response Time*’ metric and calculated metric for ‘*Average Response Time*’ of BFFCORE APIs re below:

**Step 1 – Add a new custom google analytics metric for API ‘Response Time’**

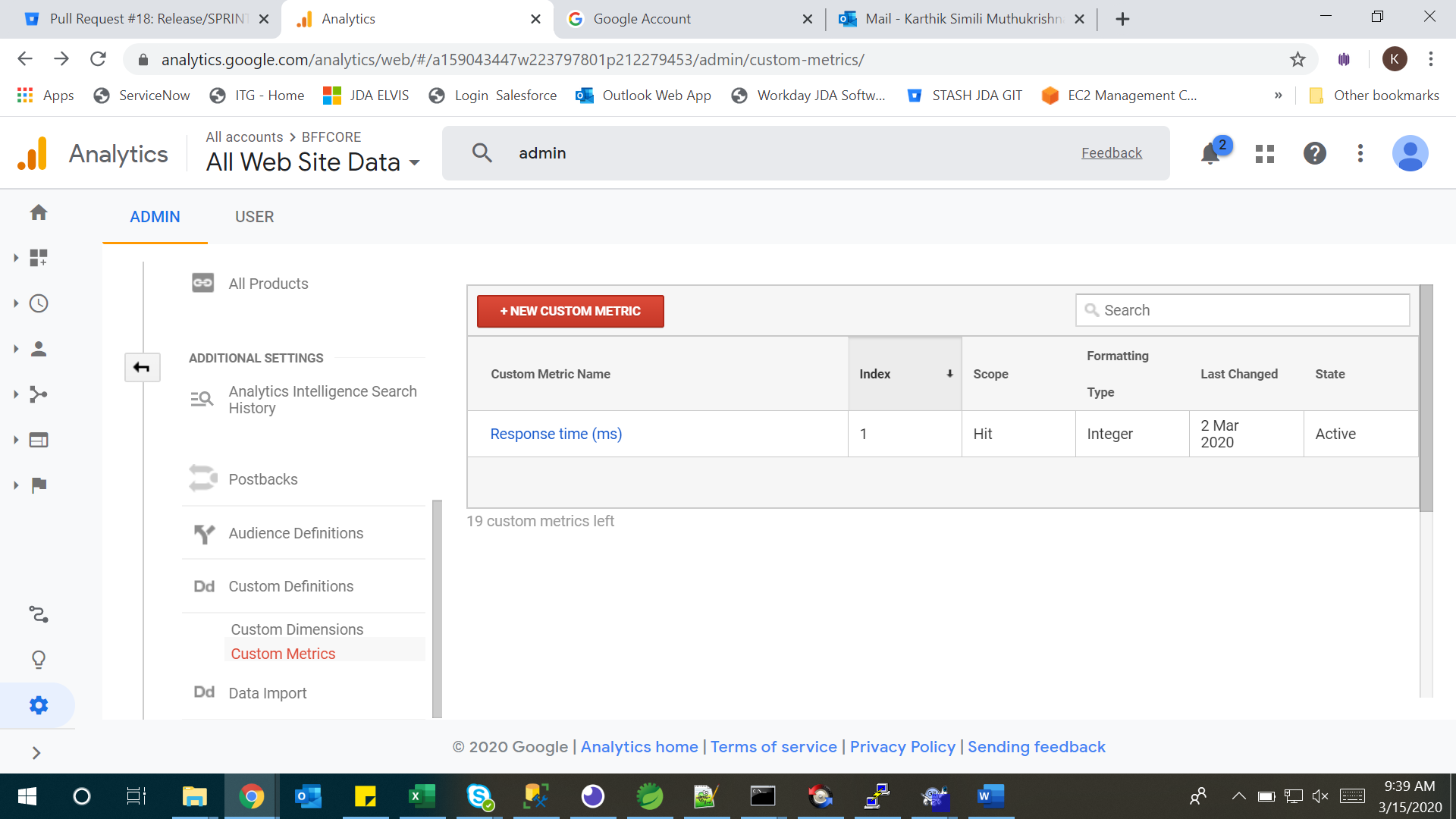
1. Create a new google accunt and login to the google analytics URL (<http://analytics.google.com>) using the new user id and password. Search for ‘**admin’** the top search bar as shown below –



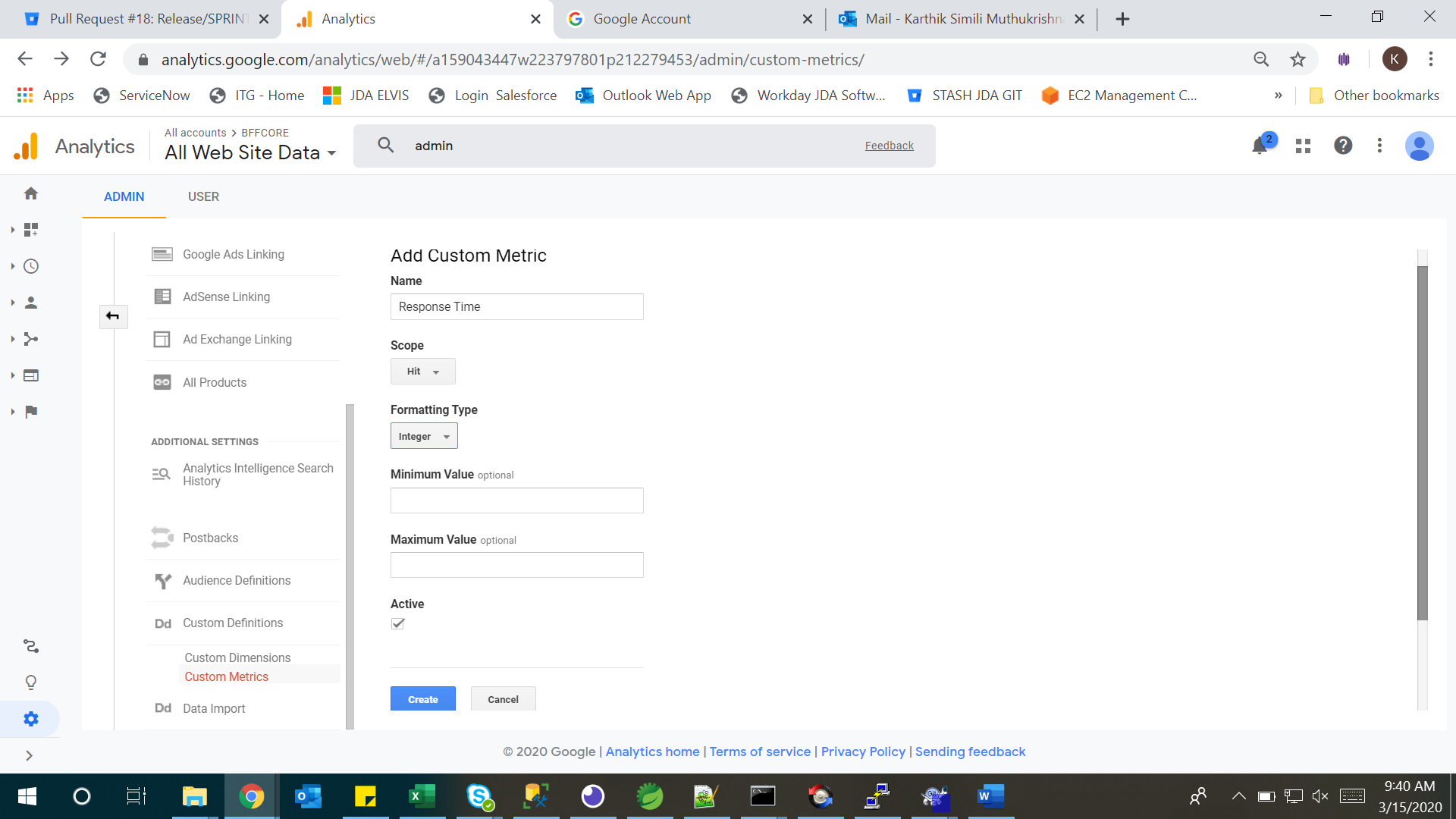
1. Click ‘Admin’ and scroll down to click on ‘*Custom Definitions 🡪 Custom Metrics*’ as shown below –



1. Click on ‘New Custom Metric’ button in the resultant page as shown below:



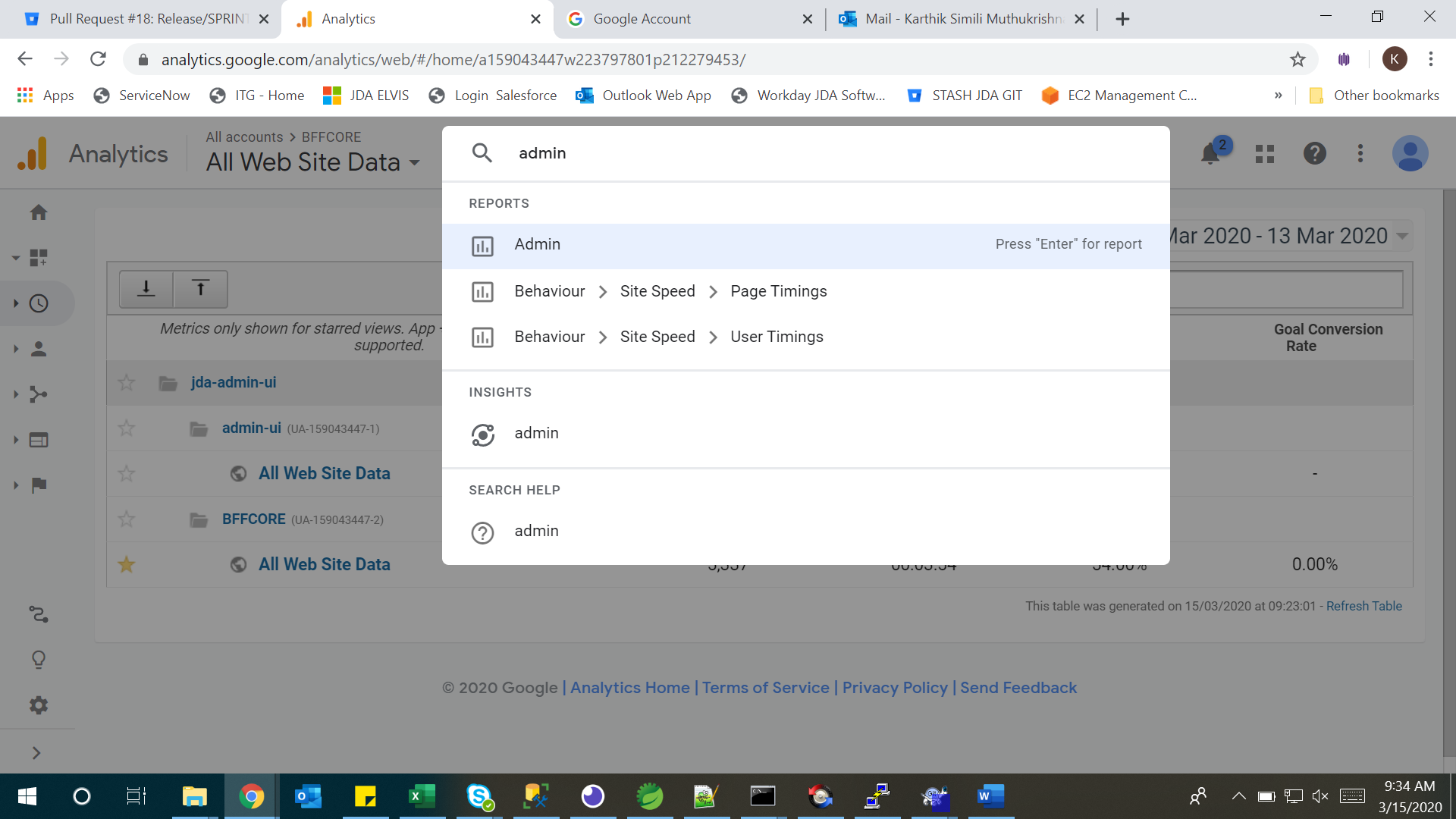
1. Enter details as shown and click on ‘*Create’* button



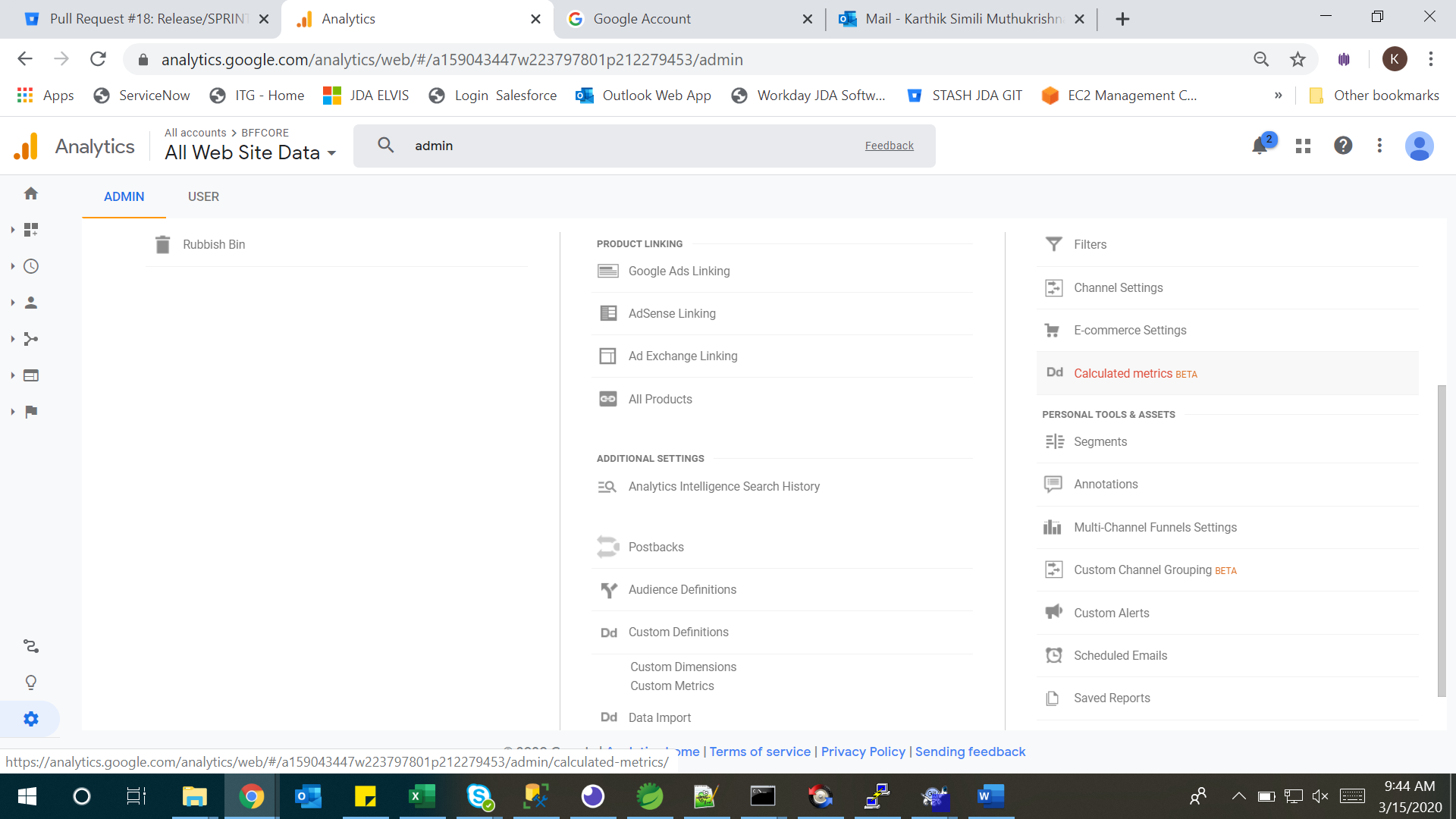
This will create a new custom metric called ‘Response Time’ which will be updated with the response time of every API hit coming in from BFFCORE server configured above (with analytics enabled).

**Step 2 – Add a new custom google analytics calculated metric for API ‘Average Response Time’**

1. Login to the google analytics URL using the new user id and password. Search for ‘**admin’** the top search bar as shown below –



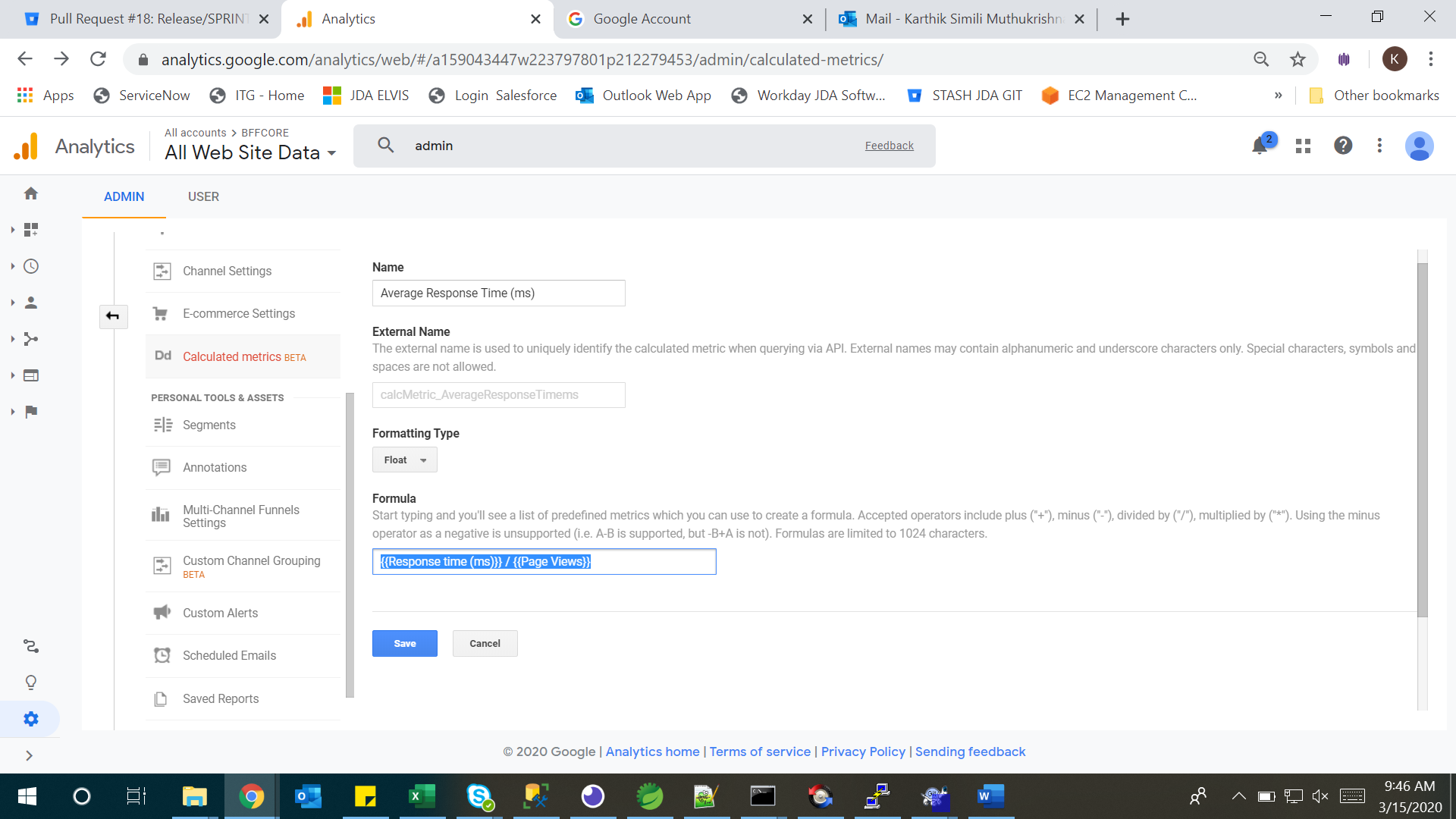
1. Click on ‘*Calculated Metrics*’ as shown below –



1. Click on ‘New Calculaed Metric’ button in the resultant page as shown below:

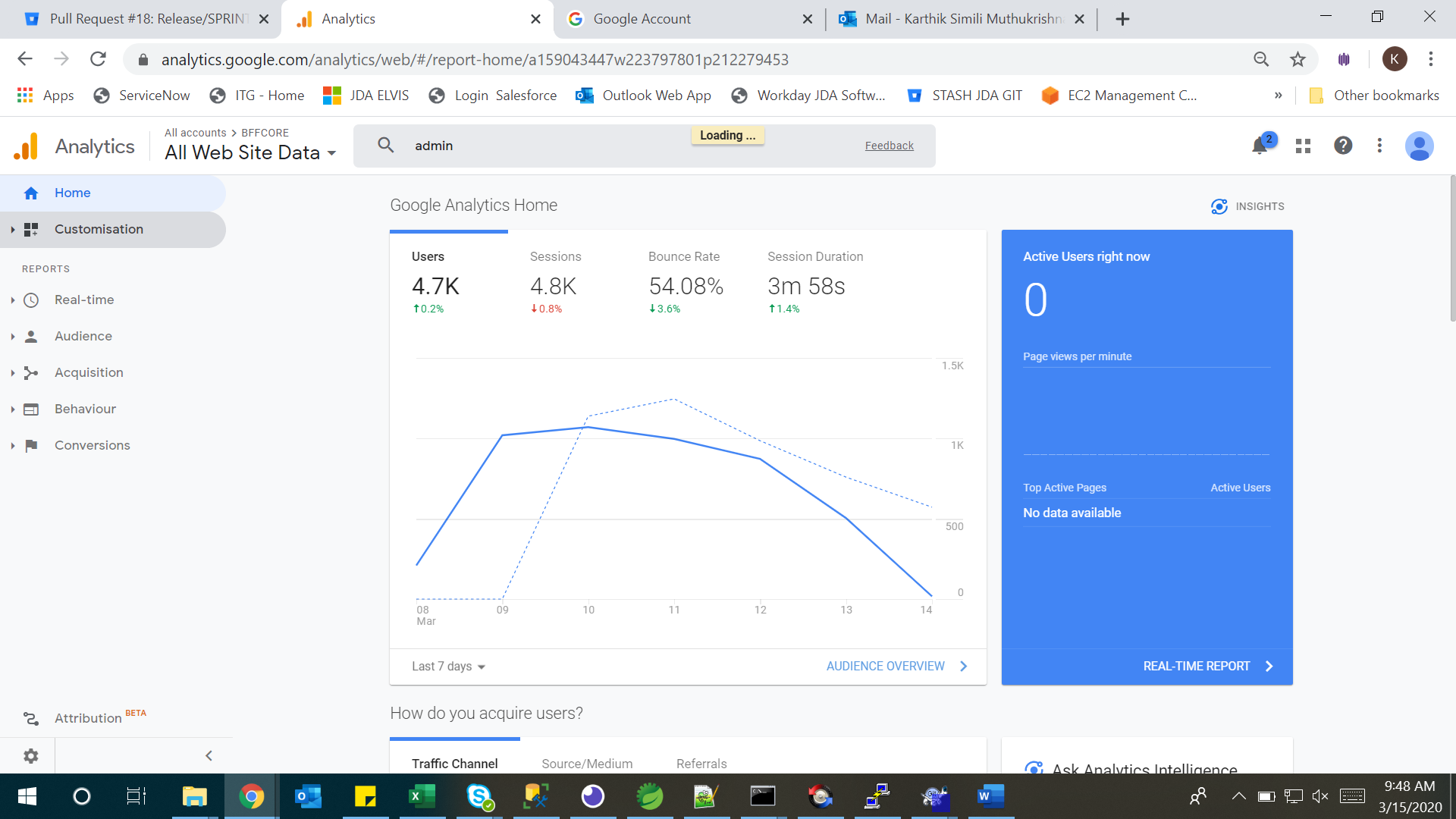
## 

1. In the resultant screen, provide the formula for Average Response time as shown below –



The *average response of each API per page view* is the final result of this calculated metric.

1. Custom reports and dashboards can be created using the out-of-the-box and calculated metrics using the Customizations menu as shown below –



## CONFIGURING AND RUNNING APPLICATION ON HTTPS (OPTIONAL)

This section will describe how to use run the BFFCORE springboot application on https using a **self-signed certificate**. However, in reality, a **trusted certificate** needs to be used to ensure that the Admin UI and Mobile Renderer can communicate with BFFCORE without handshake issues.

Steps -

1. Open the file *application-devhttps.properties* under *bffWebManagement/src/main/resources* folder and change the https port if required (default tomcat https port is **8443**). The following properties have already been added:

#Basic spring application properties

server.port=8443 **#*(Change port if required)***

#HTTPS enablement properties

security.require-ssl=true

server.ssl.key-store-type=PKCS12

server.ssl.key-store=classpath:keystore/jdabff.p12

server.ssl.key-store-password=changeit

server.ssl.key-alias=jda\_bff

1. Configure for OPEN ID / BASIC AUTH as per the instructions in the [previous section](#_APPLICATION_RESOURCES_CONFIGURATION) if required.

***Note****: This feature has not been tested end-to-end in an integrated manner yet. Currently, only instructions are being provided to test https support ONLY for BFFCORE application.*

## INSTALL LIBRARIES & GENERATE BFFCORE APPLICATION BOOTABLE JAR

1. Copy the OIDC IAM client JAR: *iam-java-client-3.1.3-SNAPSHOT-with-dependencies.jar* available in root folder of JDA GIT repository, to any folder/directory.
2. Further to previous step, run the following command from the same folder to add the IAM JAR with dependencies to the local maven repository -

*mvn install:install-file -Dfile=iam-java-client-3.1.3-SNAPSHOT-with-dependencies.jar -DgroupId=com.jda.iam -DartifactId=iam-java-client -Dversion=3.1.3-SNAPSHOT -Dpackaging=jar*

1. Run the command below from the root folder of the BFFCORE source code to build a spring boot JAR -

*mvn clean install -Dmaven.test.skip=true*

1. Run the executable application spring boot JAR generated above -
   1. Once jar is generated, navigate to 'target' folder under 'bffWebManagement' folder,copy the jar generated and place it in the folder from where the BFFCORE application should be running.
   2. Open command prompt and change the directory to the folder selected in the above step.
   3. Run the command below –

**OPEN ID**

*java -jar -Dspring.profiles.active=****devoidc*** *-Xms1024m -Xmx2048m bffWebManagement-0.0.1-SNAPSHOT.jar*

**BASIC / NATIVE AUTH**

*java -jar -Dspring.profiles.active=****devbasicauth*** *-Xms1024m -Xmx2048m bffWebManagement-0.0.1-SNAPSHOT.jar*

**Note:** BFFCORE Server needs to be restarted with either of the spring profiles above to switch between authentication schemes. There is no requirement to rebuild the spring boot JAR as per Steps 1 to 3 above.

## INITIAL DATABASE SET-UP SCRIPTS

From the root folder of JDA BFFCORE GIT repository, pick the following DB SCRIPTS and run them on the new MS SQL Server DB using SQL Server Management Studio tool downloaded (in [Section 1.3](#_Database_server_and)) above to create the database tables and populate the base and master data required to run the application -

1. *BffCore\_DBScript\_DDL\_DBSchema.sql* - The DDL to create the core BFFCORE schema - Replace all instances of <BFFCOREDB\_NAME> (with the angular braces) with the new DB created in SQL server.

**Path**: *<PROJECT ROOT>/bffDataManagement/src/main/resources/BffCore\_DBScript\_DDL\_DBSchema.sql*

1. *BffCore\_DBScript\_DDL\_Audit\_DBSchema.sql* - The DDL to create the BFFCORE audit schema - Replace all instances of <BFFCOREDB\_NAME> (with the angular braces) with the new DB created in SQL server.

**Path**: *<PROJECT ROOT>/bffDataManagement/src/main/resources/BffCore\_DBScript\_DDL\_DBSchema.sql*

1. *BffCore\_DBScript\_DML\_InitialSetup.sql* - The initial set-up scripts to populate the base and master data

**Path**: *<PROJECT ROOT>/bffDataManagement/src/main/resources/* *BffCore\_DBScript\_DML\_InitialSetup.sql*

1. Execute the UPDATE script below to update the PRODUCT (WMS) server IP address, URL scheme and port details in the BFFCORE database –

UPDATE product\_master SET

context\_path = '***<WMS server IP address>***',

port = '***<WMS Server port>***',

scheme = '***<WMS Server URL scheme>***' (Eg: http/https)

WHERE

name = 'WMS'

**Note**: Please use the same WMS server details provided in [Section 2.2](#_APPLICATION_RESOURCES_CONFIGURATION) – *Point #1 – b.*

1. *BffCore\_DBScript\_DML\_ResourceBundle.sql* - The master data script to populate the resource bundle keys for BFFCORE API success/error messages in English language

**Path**: *<PROJECT ROOT>/bffDataManagement/src/main/resources/* *BffCore\_DBScript\_DML\_ResourceBundle.sql*

1. *BffCore\_DBScript\_DML\_ResourceBundle\_fr.sql* - The master data script to populate the resource bundle keys for BFFCORE API success/error messages in French language.

**Path**: *<PROJECT ROOT>/bffDataManagement/src/main/resources/* *BffCore\_DBScript\_DML\_ResourceBundle\_fr.sql*

**Note:** To add support for more languages, please create a separate copy of this DML script and insert into the resource\_bundle table.

1. *BffCore\_DBScript\_DML\_LocalizedMobileVariables.sql* - The master data script to populate the resource bundle keys for ‘MobileRenderer’ app internal labels/texts in English language.

**Path**: *<PROJECT ROOT>/bffDataManagement/src/main/resources/* *BffCore\_DBScript\_DML\_LocalizedMobileVariables.sql*

1. *BffCore\_DBScript\_DML\_LocalizedMobileVariables\_fr.sql* - The master data script to populate the resource bundle keys for ‘MobileRenderer’ app internal labels/texts in French language.

**Path**: *<PROJECT ROOT>/bffDataManagement/src/main/resources/* *BffCore\_DBScript\_DML\_LocalizedMobileVariables\_fr.sql*

**Note:** To add support for more languages, please create a separate copy of this DML script and insert into the resource\_bundle table.

## ACCESSING AND RUNNING THE APPLICATION

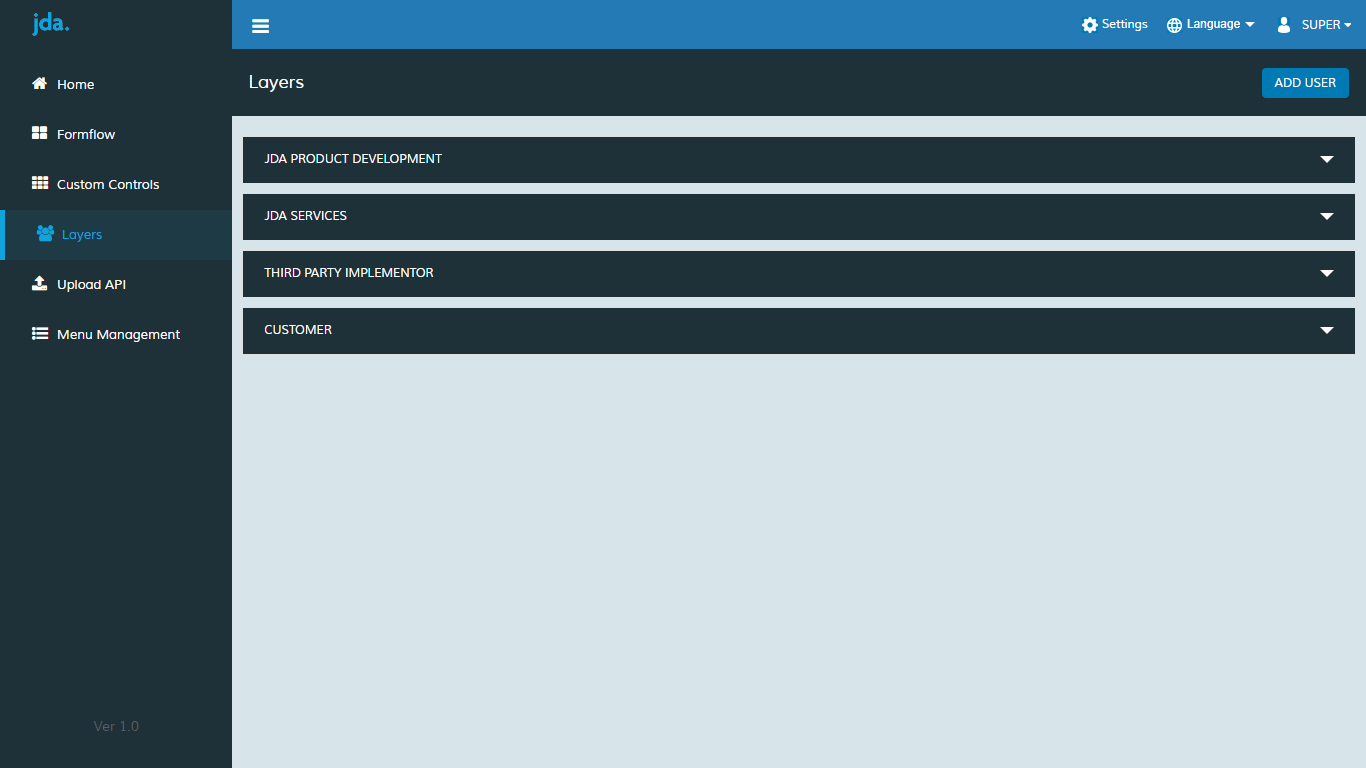
1. From a web browser, type the URL below –

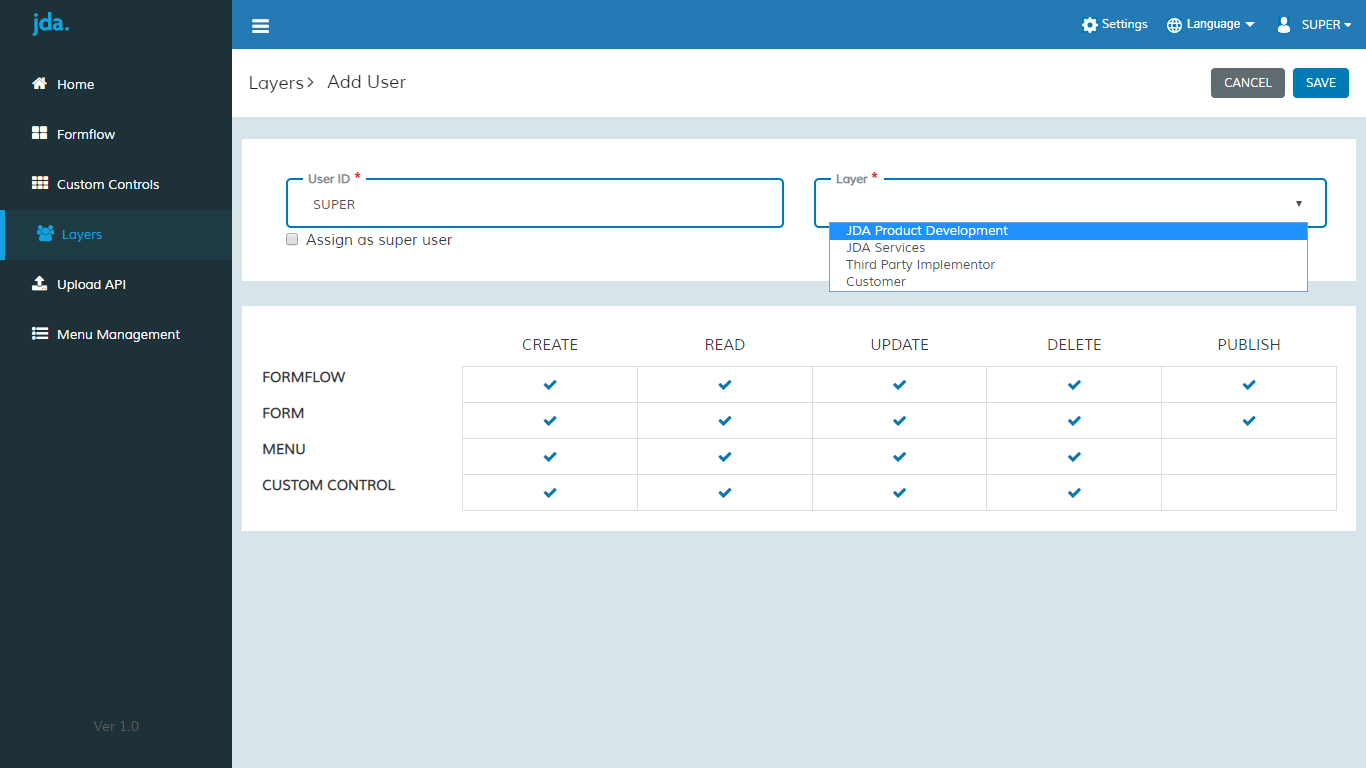
http://*<BFFCORE server machine IP address>:<BFFCORE Application Server port>* using the port configured in section. The following page will be displayed with the JSON response below (this is just to ensure BFFCORE is running and not to be used for browsing) -

{"timestamp":"Tue Mar 10 13:08:56 IST 2020","code":403,"message":"The requested endpoint requires proper authorization","errors":[{"errorCode":403,"userMessage":"Access token initial validation failed."}]}

1. Install latest Admin UI application (if not already installed) as per the installation instructions provided in the Admin UI README file.
2. Login to Admin UI using the user id / password – SUPER / SUPER and assign the user to a layer from the Layers screen as shown below –

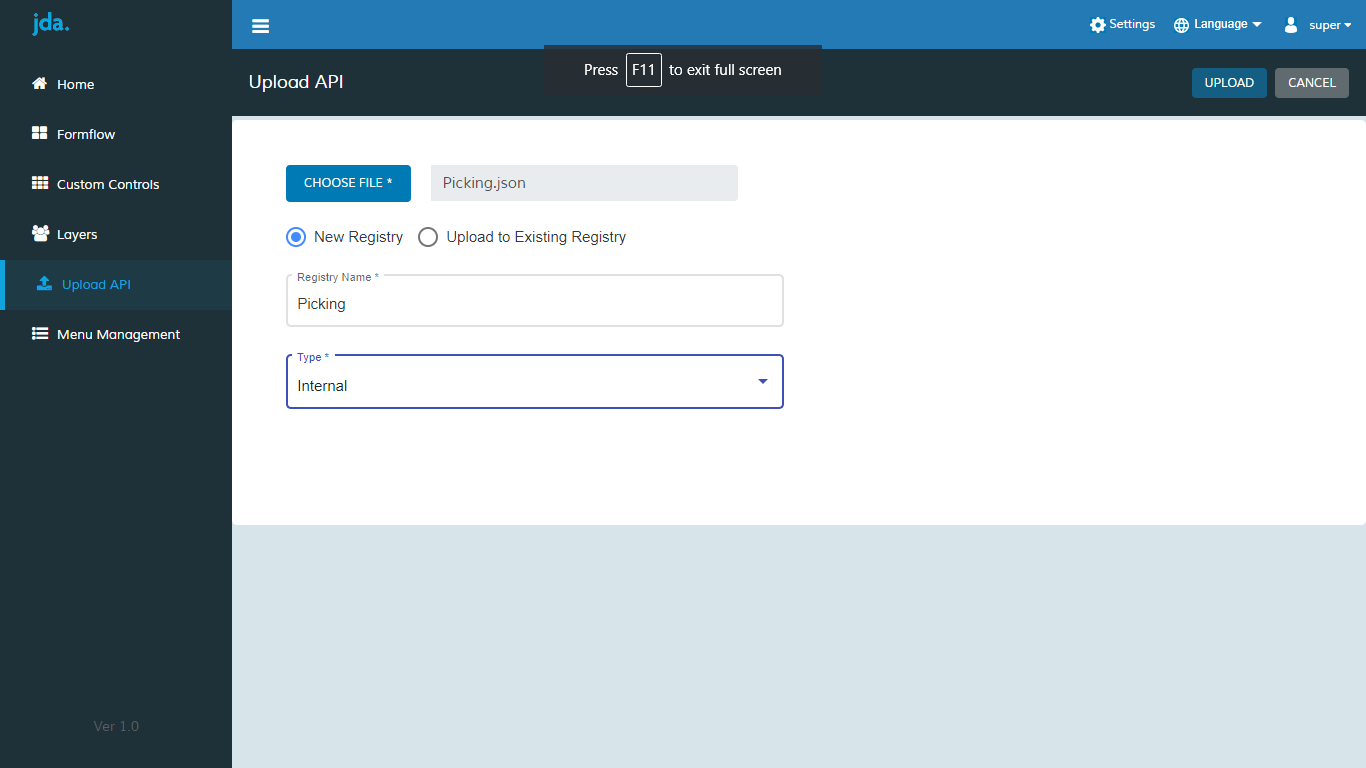
Step a).





## IMPORTING REGISTRIES AND API POST-PROCESSORS FOR PICK FLOW (OPTIONAL)

1. Login to Admin UI as detailed in the previous section using any valid WMS user (SUPER/SUPER)
2. Use the **Upload API** link on the left sidebar to import the WMS Picking registry in a new registry named **‘Picking’** as per the screenshot below –



1. Once the Picking registry is uploaded successfully, run the ‘*BffCore\_DBScript\_DML\_PrePostProcessor.sql*’ script on the new *BFFCORE MS SQL Server DB* using SQL Server Management Studio tool.

## PROVIDING SUPER USER PRIVELEGES TO WMS USERS

This is a mandatory step for all users logging into the Admin UI portal to be able to invoke WMS APIs like user roles, permissions and warehouse list in the menu management pages. Super user privleges need to be provided from the WMS Administration/Configuration portal which is automatically installed and hosted as part of the WMS docker package installation. Once the WMS instance is started and running on docker container, please follow the steps below to assign **super user priveleges** to a WMS user -

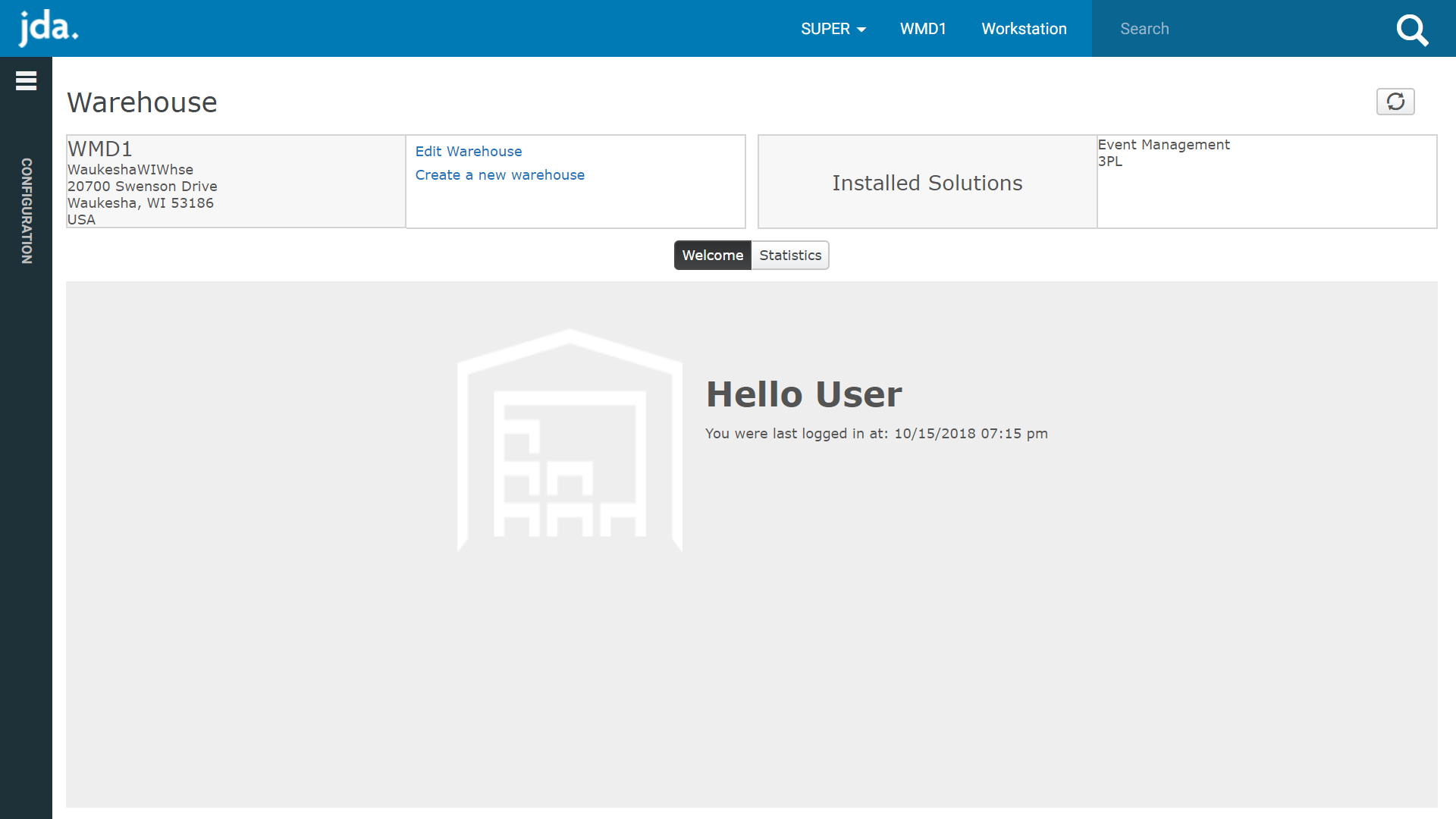
1. Open a web browser and access the WMS Administration/Configuration portal using the URL: **http://<WMS Server IP>:8090**

The following login page is displayed -

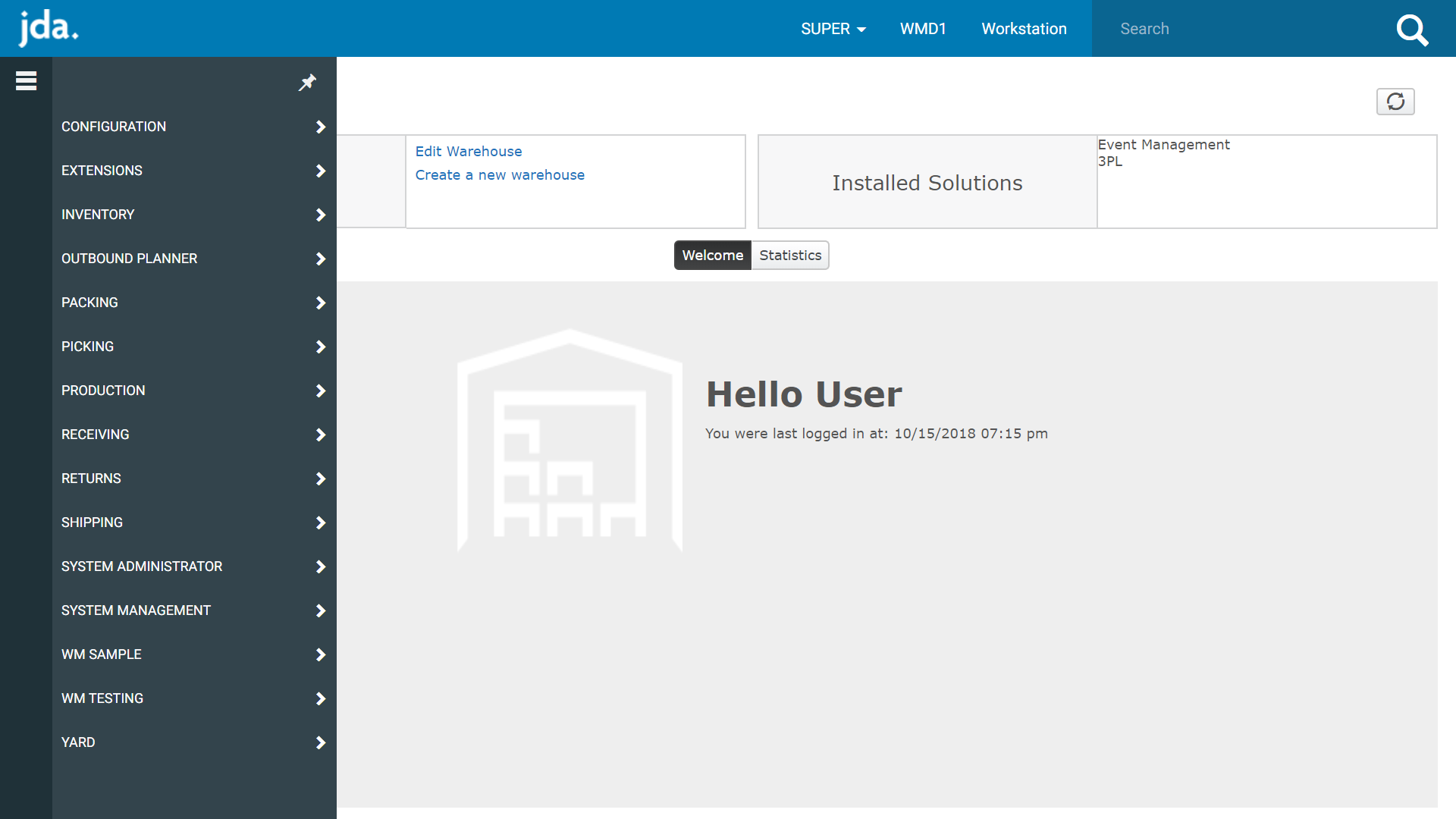


***Note:*** *This is normally hosted by default on port* ***8090*** *but if docker-compose.yml file has been changed, use the appropriate port*

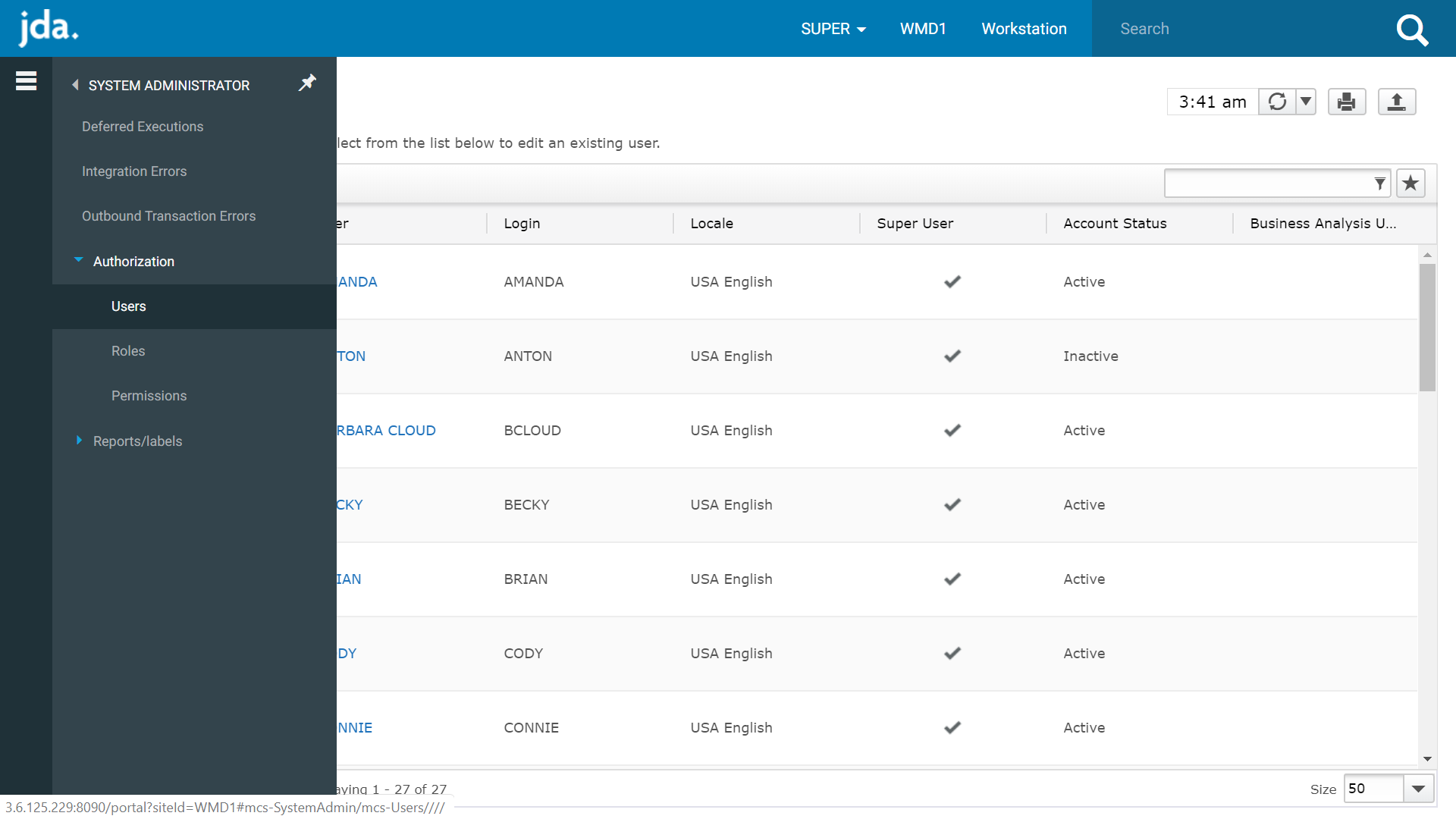
1. Login to the application above using user id / password as **SUPER/SUPER** who is already a WMS super user. The home page is displayed as shown below –



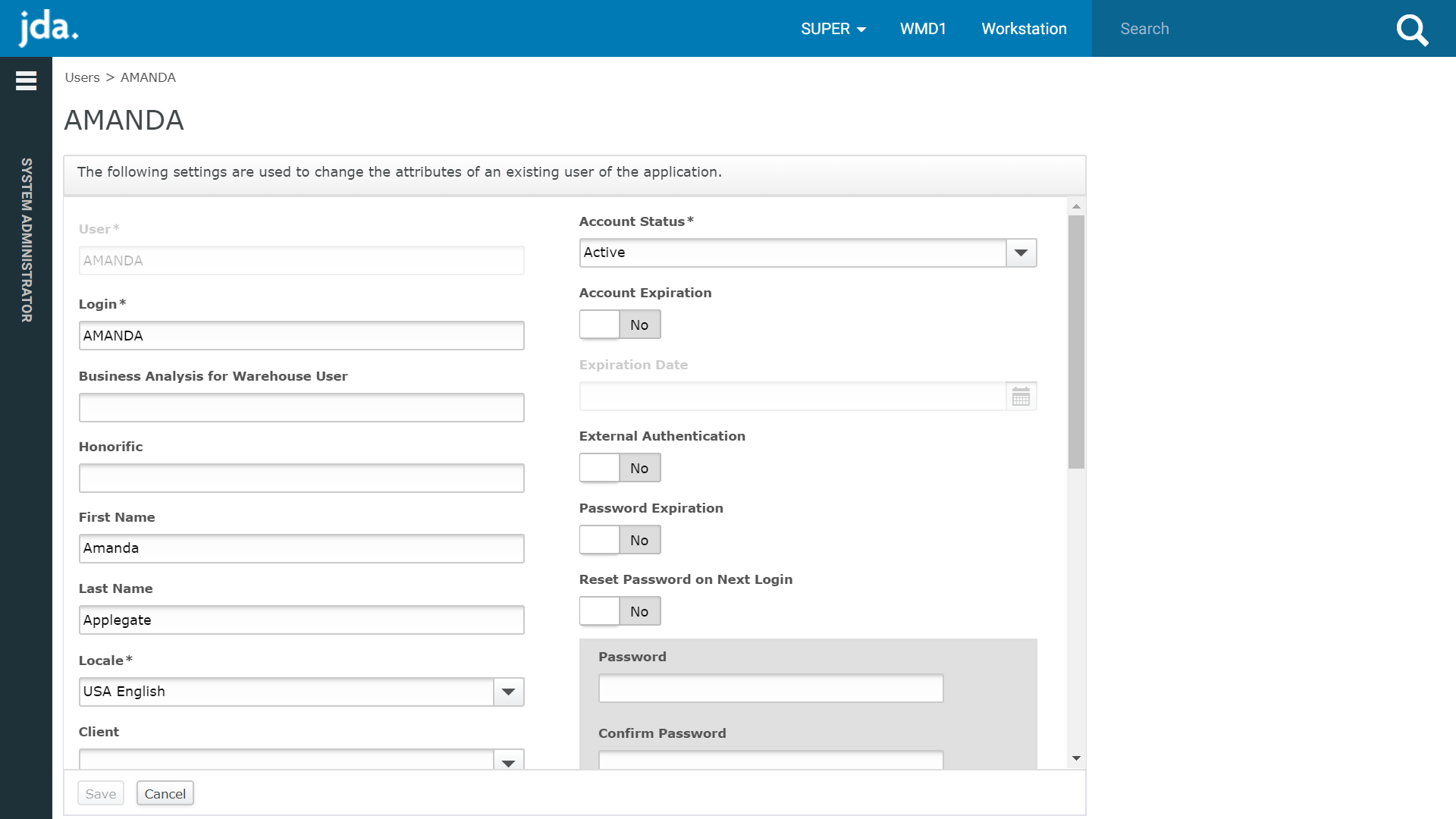
1. Click on the main menu icon () on the top left corner of the page and click on “**CONFIGURATION**” menu item. The following list of sub-menu items are displayed –



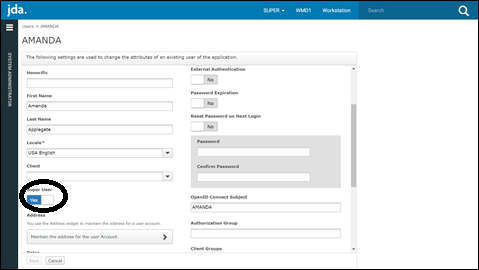
1. Click on the ‘**SYSTEM ADMINISTRATOR**’ sub-menu item and further drilldown to “Authorization 🡪 Users” sub menu item. The following screen is displayed –



1. Now click on any of the user links on the page in the previous step. The following user details page is displayed –



1. Switch on the Super User toggle to “Yes” as shown below to make the user a “WMS SUPER USER” –



## ADDING LOMBOK SUPPORT TO STS/Eclipse IDE (OPTIONAL)

**This section is optional and required only if BFFCORE source needs to be compiled and run using Eclipse/Spring Tools Suite IDE.**

Lombok library installation in Eclipse/Spring Tools Suite IDE to run BFFCORE application code -

1. After checkout of latest BFFCORE code from SCM / GIT, run

*mvn clean install -Dmaven.test.skip=true*

1. The above command will install jar in m2 folder under  M2\_REPO folder configured in SYSTEM properties. Open a new windows command prompt in the directory above to run -

*java – jar lombok-1.18.10.jar*

1. The above command will open a pop-up screen to scan eclipse installation directory. The search for eclipse installation directory will fail if using *Spring Tools Suite* (STS) as IDE . Hence, point it to your STS installation directory -

*<STS installation base path>\spring-tool-suite-4-4.3.0.RELEASE-e4.12.0-win32.win32.x86\_64\sts-4.3.0.RELEASE\ SpringToolSuite4.exe*

1. Validate that Lombok has been installed in STS by opening the IDE and clicking *Help 🡪 About* as shown below -

