GoldenGate Stream Analytics

Using GoldenGate Stream Analytics on Cloud Marketplace

F25909-07
May 2024

GoldenGate Stream Analytics Using GoldenGate Stream Analytics on Cloud Marketplace,

F25909-07

Contents

Preface

This book describes how to use Oracle Stream Anaytics on Oracle Cloud Marketplace.

Topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This document is intended for users of Oracle Stream Analytics on Oracle Cloud Marketplace.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

Documentation for Oracle Stream Analytics is available on Oracle Help Center.

Also see the following documents for reference:

- *Understanding Oracle Stream Analytics*
- *Developing Custom Jars and Custom Stages in Oracle Stream Analytics*
- *Quick Installer for Oracle Stream Analytics*
- *Known Issues in Oracle Stream Analytics*
- *Spark Extensibility for CQL in Oracle Stream Analytics*

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started with GGSA on Oracle Cloud

This chapter provides an introduction to GoldenGate Stream Analytics on the Oracle Cloud Marketplace.

1.3 Resources

The GGSA marketplace image includes the following components installed:

- GoldenGate Stream Analytics 19.1.0.0.*
- Spark 3.2.1
- Kafka 3.2.0
- Mysql Enterprise 8.0.18
- Java – JRE-8u281
- Oracle GoldenGate for Big Data 19.1.0.0.8 (Classic Architecture)

Except for Java and Mysql Enterprise, all the other components are installed under the `/u01/app` directory structure.

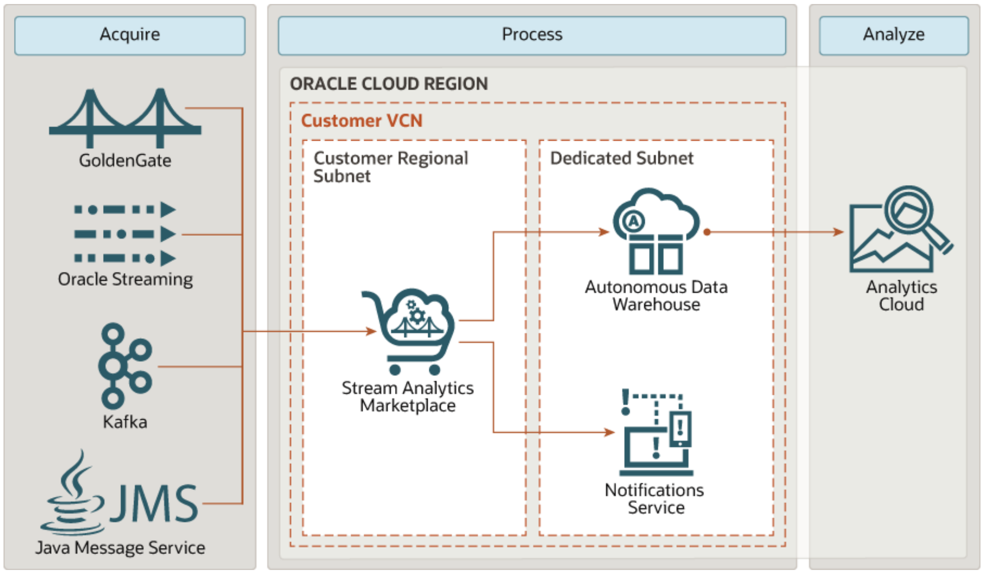
1.1 Overview

Oracle GoldenGate Stream Analytics (GGSA) on Oracle Cloud Marketplace enables customers to set up and run Oracle GGSA on Oracle Cloud. The cloud version provides the same functionality, scalability, security and support as the on-premise version. All sources and targets supported in the on-premise version are supported on cloud.

1.2 GGSA Architecture

Stream Analytics on Oracle Cloud Infrastructure Marketplace

This sample architecture diagram depicts a customer deployment of GGSA to use Autonomous Data Warehouse with Analytics Cloud as the visualization tool. GGSA supports many other target types and has its own charts and dashboards for real-time data viewing.



Acquiring data

Stream Analytics can acquire data from any of the following on-premises and cloud-native data sources:

- **GoldenGate**: Natively integrated with Oracle GoldenGate, Stream Analytics offers data replication for high-availability data environments, real-time data integration, and transactional change data capture.
- **Oracle Cloud Streaming**: Ingest continuous, high-volume data streams that you can consume or process in real-time.
- **Kafka**: A distributed streaming platform used for metrics collection and monitoring, log aggregation, and so on.
- **Java Message Service**: Allows java-based applications to send, receive, and read distributed communications.

Processing data

With Stream Analytics on Oracle Cloud Infrastructure Marketplace, you can filter, correlate, and process events in real-time, in the cloud. Marketplace solutions are deployed to Oracle Cloud Infrastructure Compute instances within your regional subnet.

After Stream Analytics processes the data, you can:

- store the data in Autonomous Data Warehouse, where it can be accessed by Analytics Cloud
- configure the Notifications service to broadcast messages to other applications hosted on Oracle Cloud Infrastucture
- send messages to OCI streaming service
- send the processed data to the OCI Object store

Analyzing data

You can connect Analytics Cloud to Autonomous Data Warehouse so that users can analyze the data in visualizations, analyses, dashboards, and pixel-perfect reports. Learn more.

1.4 Creating a GGSA Instance

You can create a GGSA instance and launch GoldenGate Stream Analytics on Oracle Cloud Marketplace using the Stack Listing, following the steps below:

Finding GGSA on Oracle Cloud Marketplace

1. Log in to the Oracle Cloud Marketplace.
2. From the Oracle Cloud Marketplace home page, use the search box under Applications and search for the keyword Stream Analytics .
3. From the search results, select GoldenGate Stream Analytics.

Deploying GGSA on Oracle Cloud Marketplace

1. From the Application page, select **Get App**.
2. Select **OCI Region** or **Log in using your Single Sign-On** credentials.
   - OCI Region – Select the desired region and click Create Stack.
3. On the **Oracle GoldenGate Stream Analytics** page, provide the following information:
   - **Select Version** - It provides a list of versions that are available
   - **Select Compartment** - Specifies the compartment where the compute node will be built. It is generally the location that you have access to build the compute node.
   - **Terms of Use** - This check box is selected by default. Oracle recommends to review the licenses before proceeding with the instance creation.
   - **Launch Stack** - It launches the stack in the OCI environment.

Creating GGSA Resource Manager Stack

1. Fill in the required **Stack** information:
   - **Name** - Name of the Stack. It has a default name and provides a date time stamp. You can edit this detail, if required.
   - **Description** - Description that you provide while creating the Stack.
   - **Create In Compartment** – It defaults to the compartment you have selected on the Oracle Marketplace page.
   - **Terraform Version** - It defaults to 0.12x.
   - **Tags** (optional) – Tags are a convenient way to assign a tracking mechanism but are not mandatory. You can assign a tag of your choice for easy tracking. You have to assign a tag for some environments for cost analysis purposes.
   - Click **Next**.
2. Fill in the required details to configure variables.
   - Name for New Resources -
     - **Display Name** – Display name used to identify all new OCI resources.
     - **Host DNS Name** – Name of the Domain Name Service for the new compute node.

> **Note:**
>
> Do not include a hyphen in the Host DNS Name, while provisioning a GGSA instance. You will see the following error if you include a hyphen: `No connection to server. Check your network connection and refresh page. If the problem persist contact administrator.`

- Network Settings -
  - **Create New** – Uncheck the check box **Use Existing Network** if you want to create new network infrastructure.
    a. **VCN Network Compartment** – Select the compartment in which you want to create the new VCN
    b. **Subnet Network Compartment** – Select the compartment in which you want to create the subnet.
    c. **New VCN DNS Name** – Name for the VCN
    d. **New VCN CIDR Block** – IP CIDR block for the VCN
    e. **New Subnet DNS Name** – Name for the new Subnet
    f. **New Subnet CIDR Block** – IP CIDR Block for the Subnet
  - **Use existing Network** – Check the **Use existing Network** check box, if you want to use existing network infrastructure.
    a. **VCN Network Compartment** – Select the compartment which has the existing VCN that you want to use
    b. **Subnet Network Compartment** – Select the compartment which has the existing Subnet that you want to use
    c. **VCN** – Select the VCN that you want to use.
    d. **Subnet** – Select the Subnet that you want to use.

> **Note:**
>
> If you are using an existing VCN and subnet, check the following if you want the instance to be accessible from public internet :
>
> - Your VCN must have an internet gateway
> - Your subnet should have security rules to allow ingress HTTP traffic for the following ports : 443, 7801, 7811-7830
>   - 443 – has ngnix running which fronts the OSA UI and spark console
>   - 7801 – is the port for the GG Manager when we install GoldenGate Big Data (GGBD)
>   - 7811-7830 –When we start GG Change data from OSA UI, it creates a GG Distribution path and starts the GG Replicat process. These are the ports used by Oracle GG processes to bind for communication with a remote Oracle GG process.
> - Your subnet should be a public subnet and should have route rule with target type set to Internet Gateway, and should use the Gateway of the VCN.

- Instance Settings –
  a. **Availability Domain** – It specifies the availability domain for the newly created GGSA Instance. It must match the Availability Domain of Subnet that you have selected in the Use Existing Network settings.
  b. **Compute Shape** – Shape of new compute instance. Supported shapes are VM.Standard2.4, VM.Standard2.8, VM.Standard2.16 and VM.Standard2.24
  c. **Assign Public IP** – This option indicates if the newly created VM should have a public IP address. This option is selected by default. If you clear this check box, no public IP address will be assigned preventing public access to the compute node. It is important to provide your RSA public key for accessing the instance using ssh. SSH is the only way to retrieve the passwords for GoldenGate Stream Analytics and MySQL Metadata store. RSA public key will start with `ssh-rsa`.
  d. **Custom Volume Sizes** – Select this check box to customize the size of the new block storage volumes that are built for the compute node.
     i. Boot Volume – 50 GB
     ii. Kafka Logs – 50 GB
     iii. Mysql Datafile – 50 GB
     iv. GG4BD (Golden Gate For Big Data) Datadir – 50GB

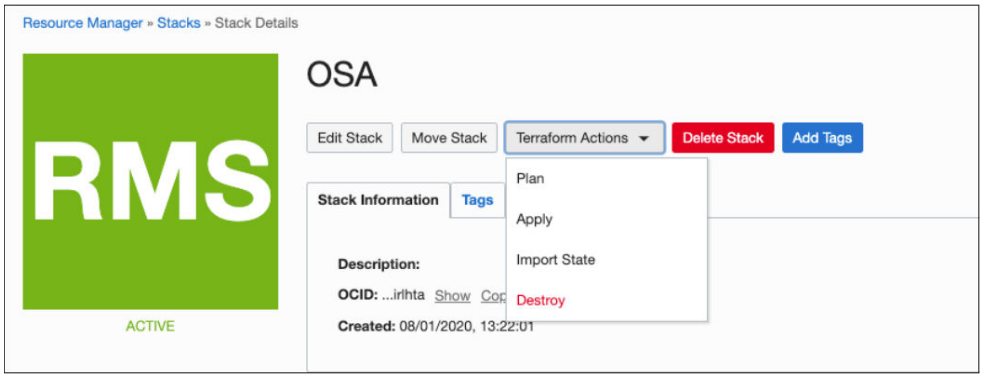
> **Note:**
>
> GGSA Resource Manager Stack will create 3 additional block volumes apart from the boot volume attached to the instance. The naming conventions for these block volume is :
>
> – <Display Name> (Kafka Logs)
>
> – <Display Name> (Mysql Datafile)
>
> – <Display Name> (GG4BD Datadir)
>
> The Display Name is what you specified earlier.
> You can configure backup for these block volumes by subscribing to the OCI Backup Policy. To apply the Backup Policy, to the Block Volume, go to **Block Storage** -> **Block Volume** and select the block volume for which you want to configure the backup. Click **Apply**.

- **Use Existing Block Volume** – Select this option if you already have block volumes from your previous install that has data that you want to reuse.
  a. Kafka Logs Block Volume – Provide the OCID of the block volume which was used for Kafka logs in your previous setup.
  b. MySql Datafile Block Volume – Provide the OCID of the block volume that was used for Mysql datafile storage in your previous setup
  c. GG4BD Datadir - Provide the OCID of the block volume that was used for Mysql datafile storage in your previous setup
- Shell Access
  - **SSH Public Key** – Public Key for allowing SSH access as the opc user. Enter the key and click **Next**.

3. Review -
   - On the **Review** page, review the information you provided and then click **Create**. The OCI Resource Manager Stack is created, and Terraform Apply action is initiated to create the OCI resources. You can see the created stack in the list of stacks in the **Resource Manager** - > **Stacks** section. You can manage the stack here.

Using Terraform Actions on Stack

Once the Stack is created, you can apply Terraform Actions on it to plan, create and destroy the OCI resources.



Description:
OCID: ...irlhta
Created: 08/01/2020, 13:22:01

- **Plan** – Terraform Plan performs a refresh, unless explicitly disabled, and then determines what actions are necessary to achieve the desired state.
- **Apply** – Use the Terraform Apply action to create the OCI resources required to reach the required state as per the user selection in the stack creation.
- **Destroy** – Use the Terraform Destroy action to destroy all the OCI resources created by the Stack during the Apply action.

> **Note:**
>
> All the resources created by the Apply action will be destroyed if you use the Destroy action, so be careful when using this action.

1.5 Validating a GGSA Instance

Checking the status of the GGSA Services

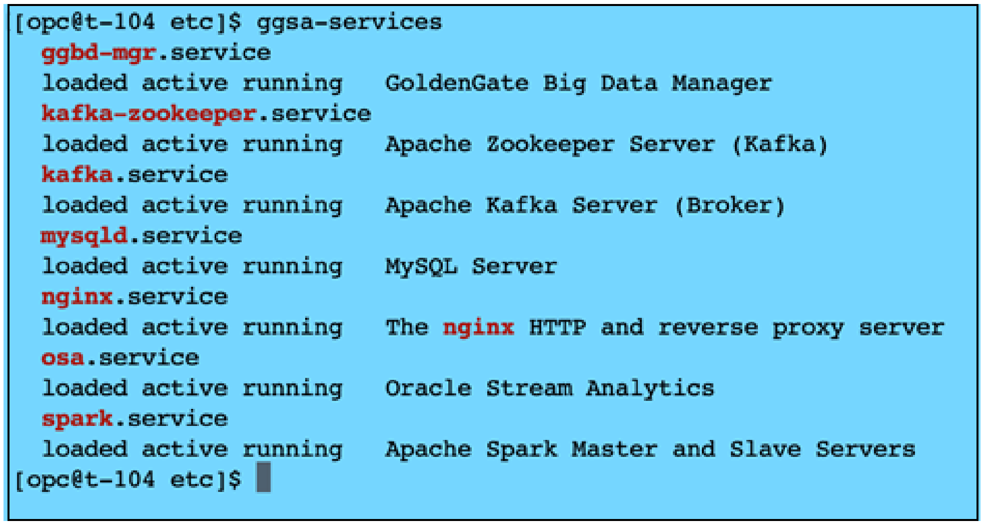
To check the status of the GGSA Services, connect to the newly created instance using SSH and run the command `ggsa-servcies`.

To connect to the instance, use the private key corresponding to the public key, provided at the time of stack creation.

You can find the Instance IP in the logs of the Apply job, or from **Outputs** under **Resources** on the **Job Details** page: **ggesp_public_ip**.

The following example illustrates how you can connect to the OSA compute node:`ssh -i <PRIVATE_KEY_FILE_PATH> opc@INSTANCE_IP`

You should see the following GGSAservices active and running:



```
[opc@t-104 etc]$ ggsa-services
  ggbd-mgr.service
  loaded active running   GoldenGate Big Data Manager
  kafka-zookeeper.service
  loaded active running   Apache Zookeeper Server (Kafka)
  kafka.service
  loaded active running   Apache Kafka Server (Broker)
  mysqld.service
  loaded active running   MySQL Server
  nginx.service
  loaded active running   The nginx HTTP and reverse proxy server
  osa.service
  loaded active running   Oracle Stream Analytics
  spark.service
  loaded active running   Apache Spark Master and Slave Servers
[opc@t-104 etc]$
```

> **Note:**
>
> If you see the OSA service to be dead or failed, wait for about 10 minutes for the services to start.

Logging in to the GGSA UI

1. Open the `README.txt` file in the home directory, and copy your OSA Admin password as shown in the line.

   ```
   ###### OSA UI

   Your osaadmin password to login to OSA UI is XXXXXXXXX
   ```

2. From your Chrome browser, type `https://<IP_of_Instance>` or `https://<IP_of_Instance/osa>`, to display the login page.
3. Login using **osaadmin** as username and the password copied from `README.txt` to display the **Home** page.

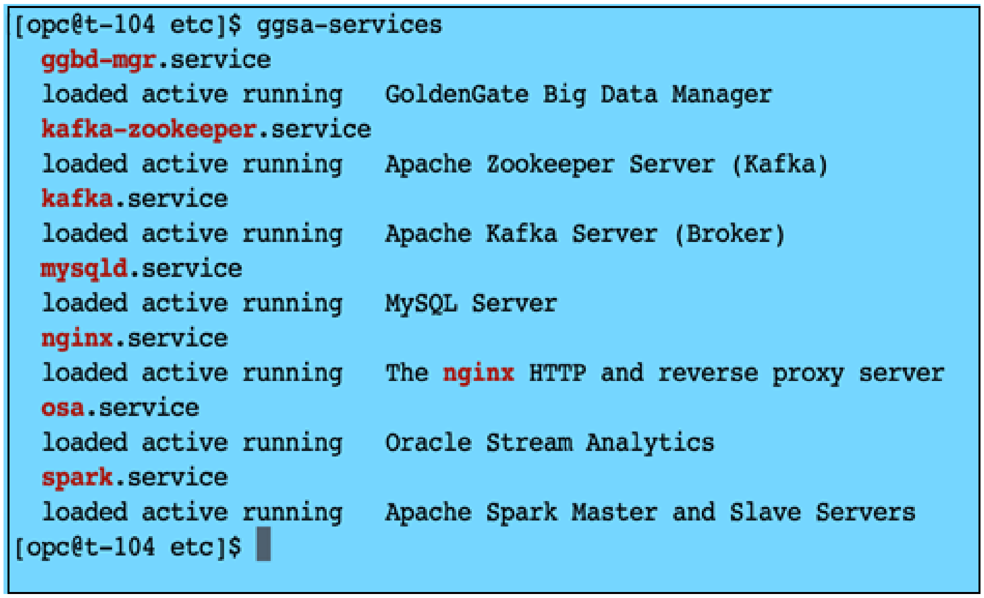
> **Note:**
>
> As mentioned in the previous section, wait for all the services to start, else the README file will have only placeholder passwords and not the actual passwords.

1.6 Starting and Stopping GGSA Services

All the GoldenGate Stream Analytics services listed above are registered as system services and are automatically started when an instance is successfully provisioned, and/ or the system is rebooted. Any published pipelines are also automatically restarted in case of a system reboot.

To start, stop, or restart the services, use the following commands from an ssh terminal:

```
sudo systemctl start osa /** starts OSA **/

sudo systemctl stop osa /** stops OSA **/

sudo systemctl restart osa /** restarts OSA **/
```



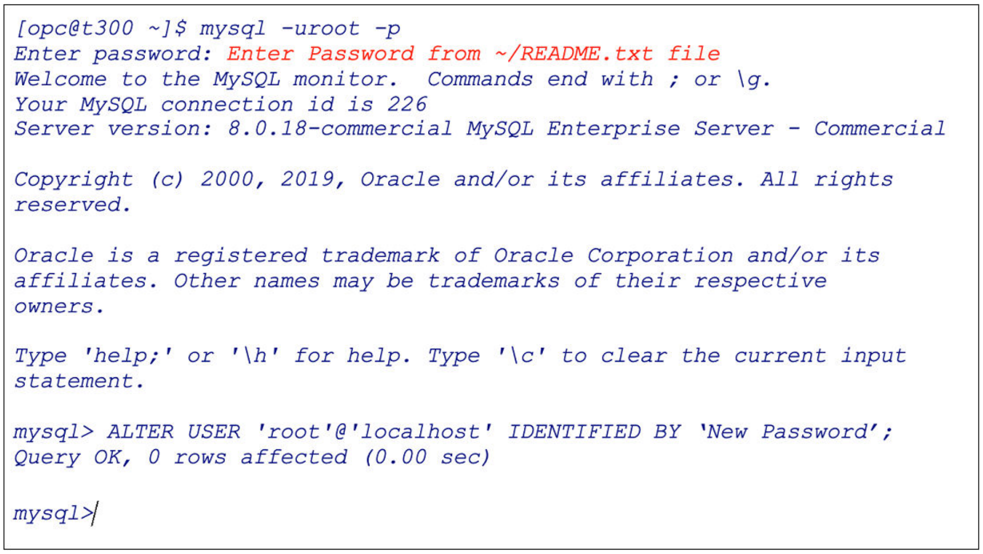
```
[opc@t-104 etc]$ ggsa-services
  ggbd-mgr.service
  loaded active running   GoldenGate Big Data Manager
  kafka-zookeeper.service
  loaded active running   Apache Zookeeper Server (Kafka)
  kafka.service
  loaded active running   Apache Kafka Server (Broker)
  mysqld.service
  loaded active running   MySQL Server
  nginx.service
  loaded active running   The nginx HTTP and reverse proxy server
  osa.service
  loaded active running   Oracle Stream Analytics
  spark.service
  loaded active running   Apache Spark Master and Slave Servers
[opc@t-104 etc]$
```

1.7 Securing your GGSA Instance

Connect to your newly created instance using ssh, and open the README.txt in the home directory. This file includes the passwords for MySQL root user and the GGSA Admin user. Additionally it includes the passwords for GGSA DB schema and a OSA Demo Database in MySQL, which you have the option to change. To change the password for GGSA DB schema, update the password in `/u01/app/osa/osa-base/etc/jetty-osa-datasource.xml` and restart GGSA.

Change MySQL Root/Admin Password

Note down or copy the MySQL admin password from README.txt file in home directory. You can change the MySQL root/admin password as shown in the commands below.



```
[opc@t300 ~]$ mysql -uroot -p
Enter password: Enter Password from ~/README.txt file
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 226
Server version: 8.0.18-commercial MySQL Enterprise Server - Commercial

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY ‘New Password’;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Note:

Similarly you can change OSA_DEMO user password with `mysql -u OSA_DEMO -p`.

Change GGSA Admin Password

Note down or copy the GGSA admin password to clipboard from README.txt file in home directory.

You can change the Admin password in the **User Management** tab of the **System Settings** screen on the UI.

Setup Spark Admin User and Password for Spark console

You can set the sparkadmin user and password in the **Environment** tab of the **System Settings** screen on the UI. You will need this to access the Spark Console.

2

Customizing Configurations

This chapter describes how to configure an external metadata store or a spark cluster.

For a list of other supported system configurations, see the latest certification matrix.

2.1 Configuring an external Metadata Store

You can use an Oracle database, an autonomous database, or MySQL as the external metadata store.

2.1.1 Customer Managed Oracle Database

A regular Oracle database can be used as a metadata store by editing the file `jetty-osa-datasource.xml`, in `/u01/app/osa/osa-base/etc` folder.

> **Note:**
>
> After configuring the OSA to a new metadata store, execute the below sql to the same user/schema.
> `UPDATE osa_system_property SET value="true" where mkey="osa.oci.instance".`

Follow the steps below to use Oracle database as a metadata store:

1. Stop GGSA by running the command `sudo systemctl stop osa`.
2. Comment the MySQL section in `jetty-osa-datasource.xml` file as shown below.

```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://
www.eclipse.org/jetty/configure_9_3.dtd">
<Configure id="Server" class="org.eclipse.jetty.server.Server">
<!--
 <New id="osads" class="org.eclipse.jetty.plus.jndi.Resource">
        <Arg>
            <Ref refid="wac"/>
        </Arg>
        <Arg>jdbc/OSADataSource</Arg>
        <Arg>
            <New class="com.mysql.cj.jdbc.MysqlConnectionPoolDataSource">
                <Set name="URL">jdbc:mysql://localhost:3306/OSADB</Set>
                <Set name="User">osa</Set>
                <Set name="Password">
                    <Call class="org.eclipse.jetty.util.security.Password"
name="deobfuscate">
                        <Arg>OBF:{OBFUSCATED_PASSWORD}</Arg>
                    </Call>
                </Set>
                <Set name="maxAllowedPacket">209715200</Set>
                <Set name="allowPublicKeyRetrieval">true</Set>
            </New>
        </Arg>
    </New>
-->
</Configure>
```

3. Add the following section for Oracle database right below the commented section.

```
<New id="osads" class="org.eclipse.jetty.plus.jndi.Resource">
         <Arg>
             <Ref refid="wac"/>
         </Arg>
         <Arg>jdbc/OSADataSource</Arg>
         <Arg>
             <New class="oracle.jdbc.pool.OracleDataSource">
                 <Set 
name="URL">jdbc:oracle:thin:@myhost.example.com:1521:XE</Set>
                 <Set name="User">osa</Set>
                 <Set name="Password">
                    <Call class="org.eclipse.jetty.util.security.Password" 
name="deobfuscate">
                        <Arg>OBF:{OBFUSCATED_PASSWORD}</Arg>
                    </Call>
                </Set>
                <Set name="connectionCachingEnabled">true</Set>
                <Set name="connectionCacheProperties">
                    <New class="java.util.Properties">
                        <Call name="setProperty"><Arg>MinLimit</
Arg><Arg>1</Arg></Call>
                        <Call name="setProperty"><Arg>MaxLimit</
Arg><Arg>15</Arg></Call>
                        <Call name="setProperty"><Arg>InitialLimit</
Arg><Arg>1</Arg></Call>
                    </New>
                </Set>
             </New>
         </Arg>
</New>
```

4. Decide on an OSA schema username and a plain-text password. For example, **osa** as schema user name and **alphago** as password.
   Change directory to `/u01/app/osa/lib` and execute the following command:

   ```
   java -cp ./jetty-util-9.4.17.v20190418.jar 
   org.eclipse.jetty.util.security.Password osa <your password>
   ```

   For example, `java -cp ./jetty-util-9.4.17.v20190418.jar org.eclipse.jetty.util.security.Password osa alphago`

   You should see results like below on console:

   ```
   2019-06-18 14:14:45.114:INFO::main: Logging initialized @1168ms to 
   org.eclipse.jetty.util.log.StdErrLogalphago
   OBF:obfuscated password
   ```

```
MD5:34d0a556209df571d311b3f41c8200f3
```

```
CRYPT:osX/8jafUvLwA
```

Note down the obfuscated password string that is displayed (shown in bold), by copying it to clipboard or notepad.

5. Change database host, port, SID, osa schema user name and osa schema password fields marked in bold in the code in Step 3.
6. The next step is to initialize the metadata store and set the password for user osaadmin.

> **Note:**
>
> You will need database administrator credentials with sysdba privileges to perform this step.

   a. Change directory to `/u01/app/osa/osa-base/bin`
   b. Execute `./start-osa.sh dbroot=<db sys user> dbroot_password=<db sys password> osaadmin_password=${OSA_ADMIN_PASSWORD}`
   c. Execute `./stop-osa.sh`
   d. Execute `sudo systemctl start osa`, to start GGSA as a system service

2.1.2 Autonomous Database

If you are using an Autonomous Database as a Metadata store, the Oracle Stream Analytics platform reads and writes all its meta-data to an Autonomous Database that is fully managed by Oracle.

1. Stop GGSA by running the command `sudo systemctl stop osa`
2. Comment the MySQL section in `jetty-osa-datasource.xml` file as shown below.

```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://
www.eclipse.org/jetty/configure_9_3.dtd">
<Configure id="Server" class="org.eclipse.jetty.server.Server">
<!--
 <New id="osads" class="org.eclipse.jetty.plus.jndi.Resource">
        <Arg>
            <Ref refid="wac"/>
        </Arg>
        <Arg>jdbc/OSADataSource</Arg>
        <Arg>
            <New class="com.mysql.cj.jdbc.MysqlConnectionPoolDataSource">
                <Set name="URL">jdbc:mysql://localhost:3306/OSADB</Set>
                <Set name="User">osa</Set>
                <Set name="Password">
                    <Call class="org.eclipse.jetty.util.security.Password"
name="deobfuscate">
                        <Arg>OBF:{OBFUSCATED_PASSWORD}</Arg>
                    </Call>
                </Set>
                <Set name="maxAllowedPacket">209715200</Set>
                <Set name="allowPublicKeyRetrieval">true</Set>
            </New>
        </Arg>
    </New>
-->
</Configure>
```

3. Add the following section for Oracle database right below the commented section.

```
<New id="osads" class="org.eclipse.jetty.plus.jndi.Resource">
         <Arg>
             <Ref refid="wac"/>
         </Arg>
         <Arg>jdbc/OSADataSource</Arg>
         <Arg>
             <New class="oracle.jdbc.pool.OracleDataSource" type="adw">
                 <Set name="URL">jdbc:oracle:thin:@{service_name}?
TNS_ADMIN={wallet_absolute_path}</Set>
                 <Set name="User">osa</Set>
                 <Set name="Password">
                    <Call class="org.eclipse.jetty.util.security.Password"
name="deobfuscate">
                        <Arg>OBF:{obfuscated_password}</Arg>
                    </Call>
                </Set>
                <Set name="connectionCachingEnabled">true</Set>
                <Set name="connectionCacheProperties">
                    <New class="java.util.Properties">
                        <Call name="setProperty"><Arg>MinLimit</
Arg><Arg>1</Arg></Call>
                        <Call name="setProperty"><Arg>MaxLimit</
Arg><Arg>15</Arg></Call>
                        <Call name="setProperty"><Arg>InitialLimit</
Arg><Arg>1</Arg></Call>
                    </New>
                </Set>
             </New>
         </Arg>
     </New>
```

4. Replace the following variables in above XML with actual values:
   - `service_name` – one of the service names listed in tnsnames.ora file inside the wallet
   - `wallet_absolute_path` – the absolute path of wallet folder on the machine where osa is installed
   - `obfuscated_password` – the obfuscated password for user ‘osa’. See the previous section on how to generate an obfuscated password
5. The next step is to initialize the metadata store and set the password for user osaadmin.

> **Note:**
>
> You will need database administrator credentials with sysdba privileges to perform this step.

a. Change directory to `/u01/app/osa/osa-base/bin`

b. Execute `./start-osa.sh dbroot=<db sys user> dbroot_password=<db sys password> osaadmin_password=${OSA_ADMIN_PASSWORD}`

c. Execute `./stop-osa.sh`

d. Execute `sudo systemctl start osa`, to start GGSA as a system service

2.2 Configuring an External Spark Cluster

Ensure network connectivity to hosts when configuring an external Spark cluster.

Spark on Oracle Big Data Service

You can configure GoldenGate Stream Analytics to use an external Spark cluster running in Oracle Big Data Service.

For configuration steps, see Setting up Runtime for GoldenGate Stream Analytics Server.

2.3 Configuring an External OCI Streaming Service

Ensure network connectivity to hosts when configuring an external OCI Streaming Service.

You can configure GoldenGate Stream Analytics to use an external OCI Streaming Service.

For configuration steps, see Configuring OCI Streaming Service .

3
Working with a GGSA Instance

This chapter describes how to create artifacts, data sources and targets, working with pipelines and monitoring the GGSA services.

3.1 Using the Local MYSQL Database, Spark, and Kafka Clusters

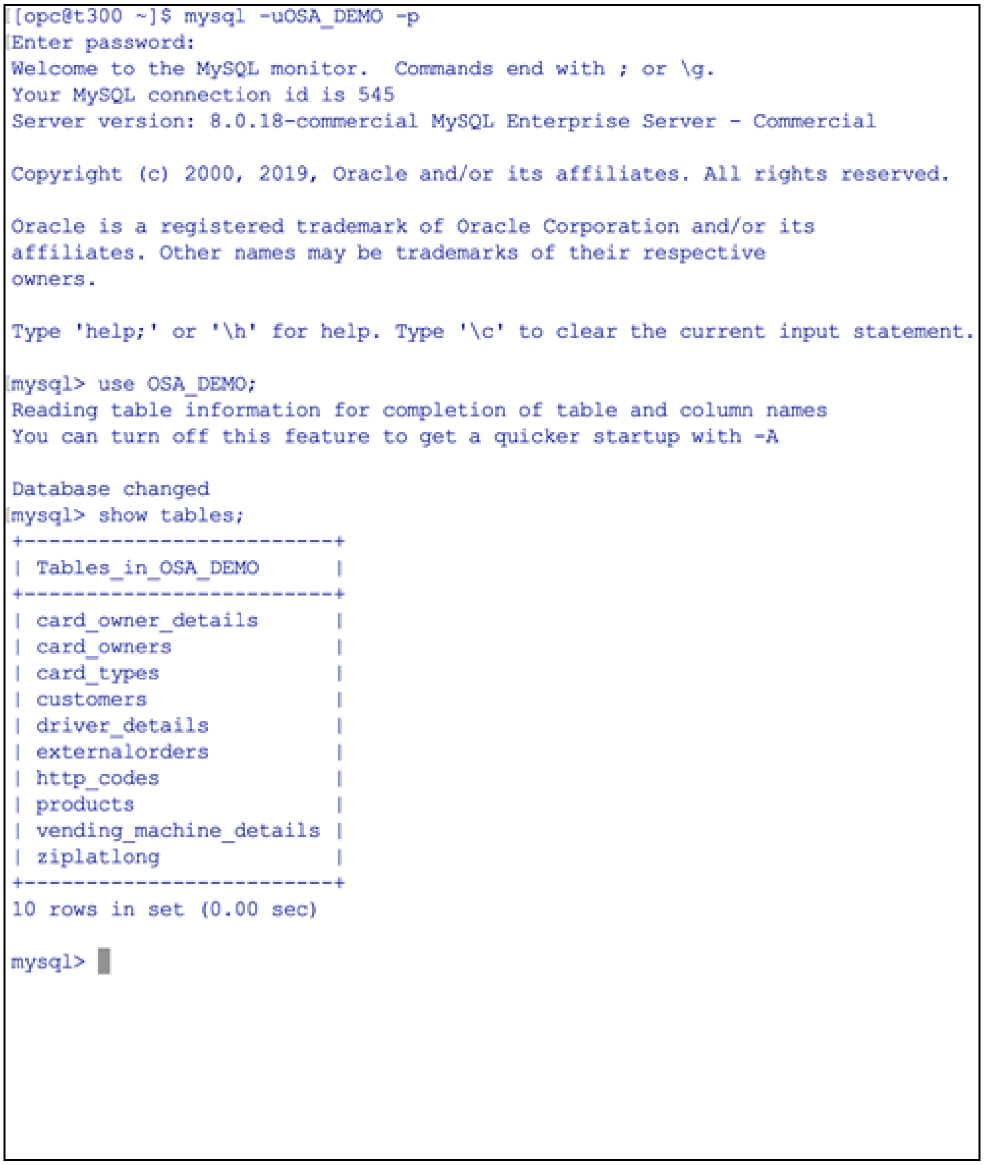
This chapter describes how to utilize the local MySQL database, the local Kafka Cluster and the local Spark Cluster.

3.1.1 Utilizing Local MySQL Database

You can use the local MySQL database, to store lookup and reference data and correlate the same with an event stream, while developing real-time ETL and Analytical pipelines.

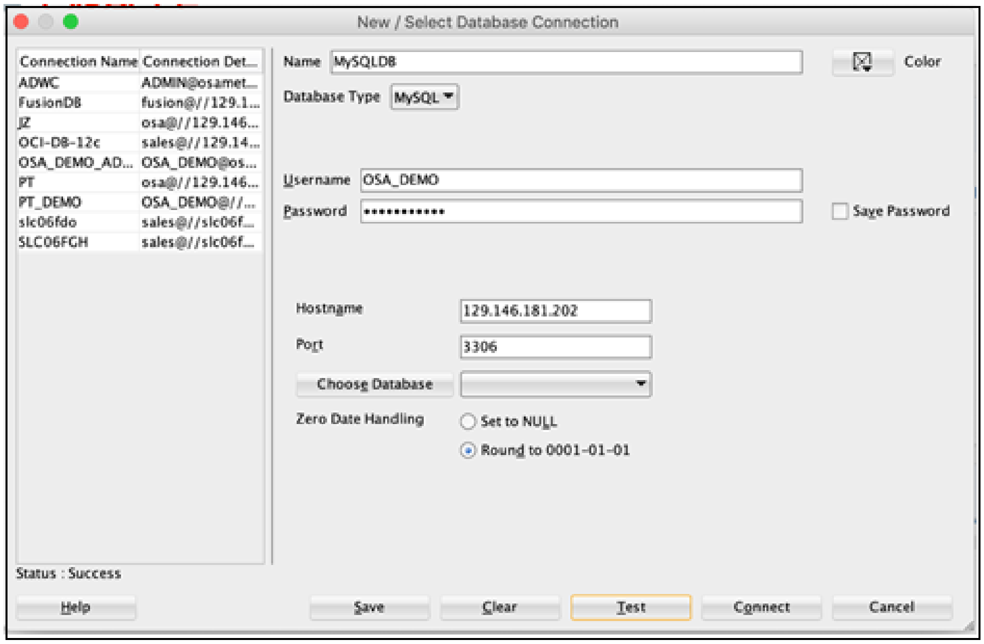
You can access the MySQL database on the instance from MySQL prompt or from SQL developer. To access from SQL developer running on a different host, ensure your VCN security rule allows TCP traffic to port 3306.

We recommend you to create database objects for your application in the existing OSA_DEMO database as shown in the screenshot below:



```
[opc@t300 ~]$ mysql -uOSA_DEMO -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 545
Server version: 8.0.18-commercial MySQL Enterprise Server - Commercial

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use OSA_DEMO;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------------+
| Tables_in_OSA_DEMO      |
+-------------------------+
| card_owner_details      |
| card_owners             |
| card_types              |
| customers               |
| driver_details          |
| externalorders          |
| http_codes              |
| products                |
| vending_machine_details |
| ziplatlong              |
+-------------------------+
10 rows in set (0.00 sec)

mysql> 
```

You can also connect from SQL Developer as shown in the screenshot below. Please note the setting for **Zero Date Handling**.



New / Select Database Connection

Connection Name	Connection Det...
ADWC	ADMIN@osamet...
FusionDB	fusion@//129.1...
JZ	osa@//129.146...
OCI-DB-12c	sales@//129.14...
OSA_DEMO_AD...	OSA_DEMO@os...
PT	osa@//129.146...
PT_DEMO	OSA_DEMO@//...
slc06fdo	sales@//slc06f...
SLC06FGH	sales@//slc06f...

Name: MySQLDB

Database Type: MySQL

Username: OSA_DEMO

Password: ************

□ Save Password

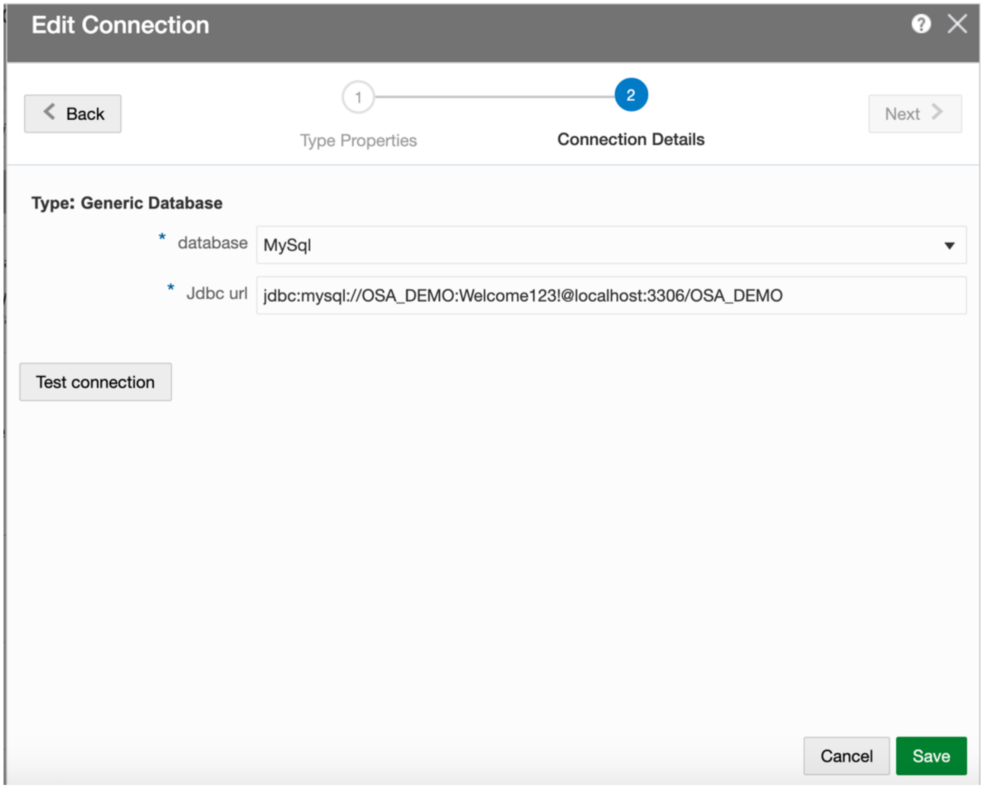
Hostname: 129.146.181.202

Port: 3306

Zero Date Handling: ○ Set to NULL ● Round to 0001-01-01

Status : Success

You can use the following connect string, when you create a connection from GGSA UI for MySQL database:



Edit Connection

Type: Generic Database

* database: MySql

* Jdbc url: jdbc:mysql://OSA_DEMO:Welcome123!@localhost:3306/OSA_DEMO

Note:

The password mentioned in the screen above is the password for OSA_DEMO schema, which holds the sample data for users to try, and is by default set to `Welcome123!`. You have an option to change it.

3.1.2 Utilizing the Local Kafka Cluster

You can use the local Kafka cluster as a source or target in GGSA pipelines. The instance provides a range of Kafka utilities to read and write data from local Kafka. These utilities are in the `README.txt` file in the `/u01/app/osa/utilities/kafka-utils` folder. Below is the description of the scripts for quick reference:

- For Kafka-related help, type `./kafka.sh help`.
- To list Kafka topics, run `./kafka-topics.sh`.
- To listen to a Kafka topic, type `./kafka-listen.sh {topic-name}`.
- To feed a csv file into Kafka, run

  ```
  cat {your-csv-file} | ./csv2json.sh | ./sampler.sh 1 1 | ./kafka.sh
  feed {your-topic-name}
  ```
- To feed a json file, run `cat {your-json-file} | ./sampler.sh 1 1 | ./kafka.sh feed {your-topic-name}`
- To loop through your csv file, run `./loop-csv.sh {your-csv-file} | ./csv2json.sh | ./sampler.sh 1 1 | ./kafka.sh feed {your-topic-name}`
- The command `sampler.sh {n} {t}` reads {n} lines in each {t} seconds from STDIN and writes to STDOUT. Default values: n=1 t=1
- The command `csv2json.sh` reads the CSV from the STDIN and writes JSON to STDOUT. First line must be the CSV header.
- There are data files to simulate GoldenGate capture:

  ```
  cat ./gg_orders.json | ./sampler.sh | ./kafka.sh feed goldengate
  ```
- There are data files to simulate nested json structures:

  ```
  ./loop-file.sh ./complex.json | ./sampler.sh 1 1 | ./kafka.sh feed
  complex
  ```
- There are data files to simulate movement of buses:

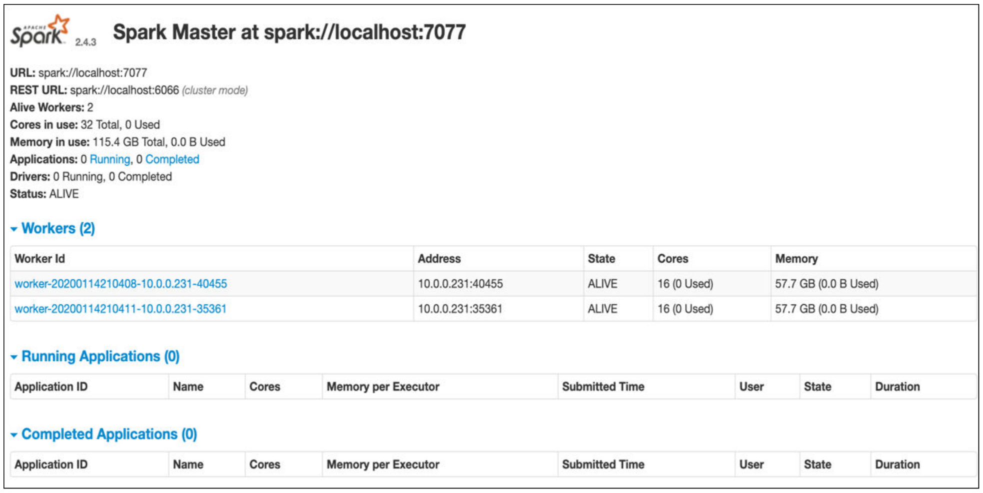
  ```
  cat ./30-min-at-50-rps.json | ./sampler.sh 20 1 | kafka.sh feed
  buses
  ```
- In general to feed test data to your running kafka installation, run `{your-shell-script} | kafka.sh feed {your-topic-name}`

> **Note:**
>
> `your-shell-script` generates data in JSON format.
>
> **Sample output:**
>
> ```
> {"price":284.03,"symbol":"USDHUF=x","name":"USD/HUF"}
>
> {"price":316.51,"symbol":"EURHUF=x","name":"EUR/HUF"}
>
> {"price":0.8971,"symbol":"USDEUR=x","name":"USD/EUR"}
> ```

3.1.3 Utilizing the Local Spark Cluster

The local Spark cluster is used to run your own Spark jobs, in addition to the GGSA pipelines. Spark jobs can be submitted to the REST endpoint using **localhost** and port 6066. Spark is installed in `/u01/app/spark` and you can change the configuration by referring the Spark documentation.

You can access Spark console by typing `https://<IP_of_Instance/spark>` and providing the username/password. By default the Spark cluster is configured with two workers, each with 16 virtual cores.



Spark Master at spark://localhost:7077

URL: spark://localhost:7077
REST URL: spark://localhost:6066 *(cluster mode)*
Alive Workers: 2
Cores in use: 32 Total, 0 Used
Memory in use: 115.4 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory
worker-20200114210408-10.0.0.231-40455	10.0.0.231:40455	ALIVE	16 (0 Used)	57.7 GB (0.0 B Used)
worker-20200114210411-10.0.0.231-35361	10.0.0.231:35361	ALIVE	16 (0 Used)	57.7 GB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

3.2 Working with a Sample Pipeline

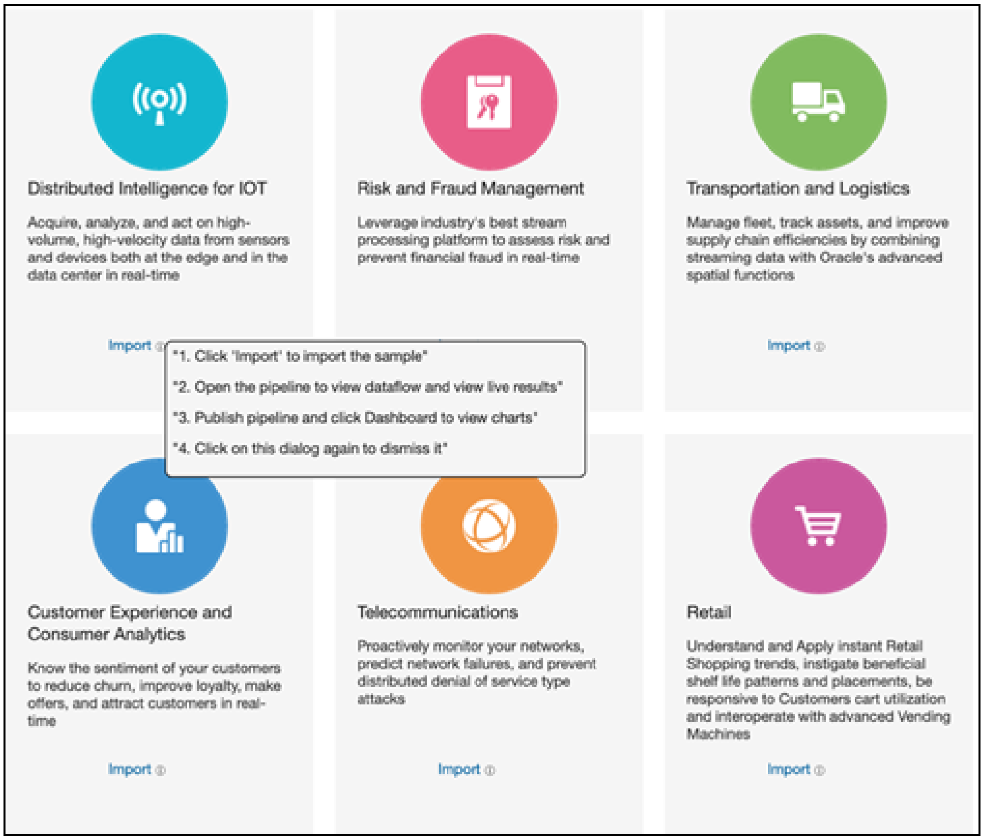
This chapter describes with example, how to import pipelines, publish them and view live results from a pipline.

3.2.1 IOT Sample

The VendingMachineManagement sample demonstrates a pipeline that ingests streaming IOT data from a farm of vending machines. The sample analyzes this data to check the health of these machines, and proactively manage the inventory in these machines.

To import and work with the VendingMachineManagement sample:

1. On the Home page, click the **Import** link on the **Distributed Intelligence for IOT** tile, to import the pipelines and dashboards.



Distributed Intelligence for IOT

Acquire, analyze, and act on high-volume, high-velocity data from sensors and devices both at the edge and in the data center in real-time

Import

Risk and Fraud Management

Leverage industry's best stream processing platform to assess risk and prevent financial fraud in real-time

Transportation and Logistics

Manage fleet, track assets, and improve supply chain efficiencies by combining streaming data with Oracle's advanced spatial functions

Import

"1. Click 'Import' to import the sample"

"2. Open the pipeline to view dataflow and view live results"

"3. Publish pipeline and click Dashboard to view charts"

"4. Click on this dialog again to dismiss it"

Customer Experience and Consumer Analytics

Know the sentiment of your customers to reduce churn, improve loyalty, make offers, and attract customers in real-time

Import

Telecommunications

Proactively monitor your networks, predict network failures, and prevent distributed denial of service type attacks

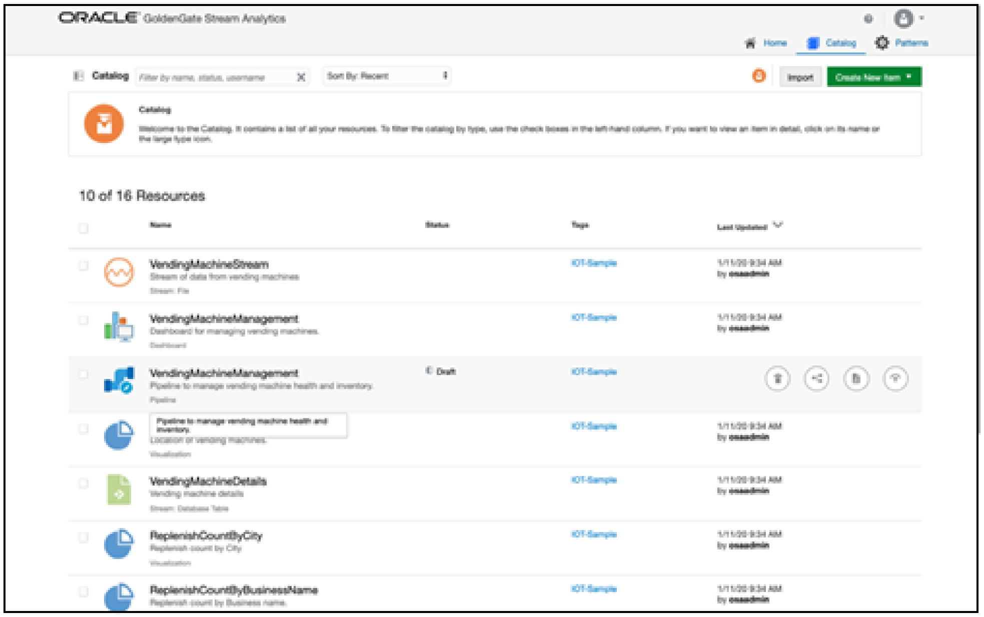
Import

Retail

Understand and Apply instant Retail Shopping trends, instigate beneficial shelf life patterns and placements, be responsive to Customers cart utilization and interoperate with advanced Vending Machines

Import

2. Click the IOT logo to navigate to the **Catalog** page, or click the **Catalog** tab. On the **Catalog** page, click the pipeline to view the live results from the pipeline.

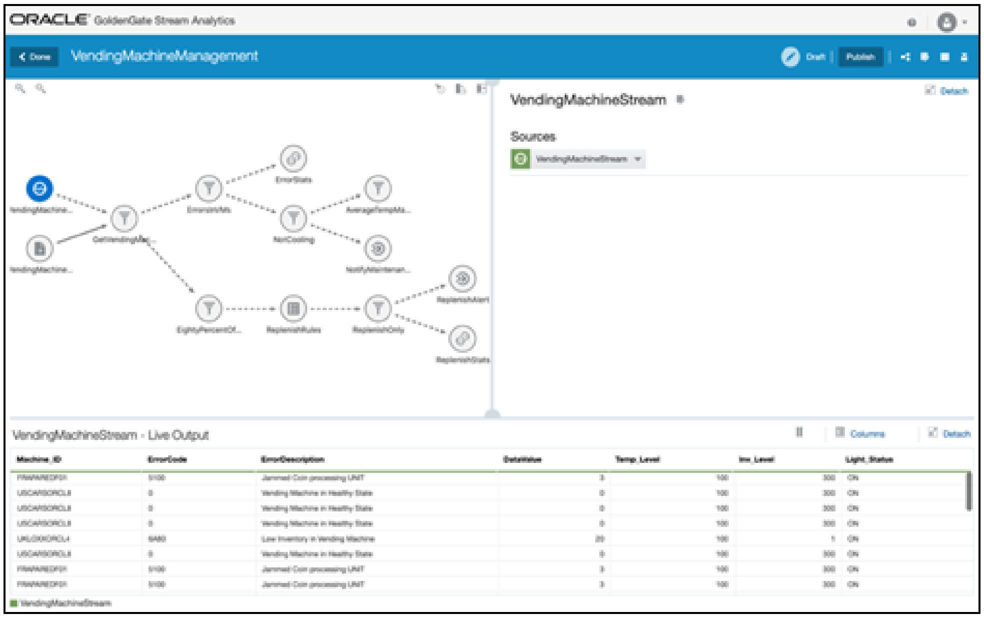

ORACLE GoldenGate Stream Analytics
Home
Catalog
Patterns
Catalog
Filter by name, status, username
Sort By: Recent
Import
Create New Item

Catalog

Welcome to the Catalog. It contains a list of all your resources. To filter the catalog by type, use the check boxes in the left hand column. If you want to view an item in detail, click on its name or the large type icon.

10 of 16 Resources

Name	Status	Tags	Last Updated
VendingMachineStream Stream of data from vending machines Stream: File		IOT-Sample	1/11/20 9:34 AM by osaadmin
VendingMachineManagement Dashboard for managing vending machines. Dashboard		IOT-Sample	1/11/20 9:34 AM by osaadmin
VendingMachineManagement Pipeline to manage vending machine health and inventory. Pipeline	Draft	IOT-Sample	
Pipeline to manage vending machine health and inventory. Location of vending machines. Visualization		IOT-Sample	1/11/20 9:34 AM by osaadmin
VendingMachineDetails Vending machine details Stream: Database Table		IOT-Sample	1/11/20 9:34 AM by osaadmin
ReplenishCountByCity Replenish count by City Visualization		IOT-Sample	1/11/20 9:34 AM by osaadmin
ReplenishCountByBusinessName Replenish count by Business name.		IOT-Sample	1/11/20 9:34 AM by osaadmin

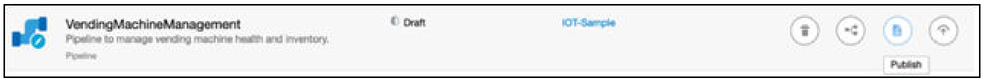


VendingMachineStream - Live Output

Machine_ID	ErrorCode	ErrorDescription	DataValue	Temp_Level	Inv_Level	Light_Status
FRAPAREDF01	5100	Jammed Coin processing UNIT	3	100	300	ON
USCARSORCL8	0	Vending Machine in Healthy State	0	100	300	ON
USCARSORCL8	0	Vending Machine in Healthy State	0	100	300	ON
USCARSORCL8	0	Vending Machine in Healthy State	0	100	300	ON
UKLOXXORCL4	6A80	Low Inventory in Vending Machine	20	100	1	ON
USCARSORCL8	0	Vending Machine in Healthy State	0	100	300	ON
FRAPAREDF01	5100	Jammed Coin processing UNIT	3	100	300	ON
FRAPAREDF01	5100	Jammed Coin processing UNIT	3	100	300	ON

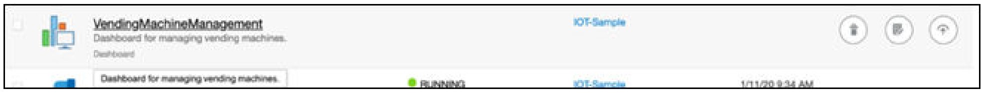
VendingMachineStream

3. Click **Done**, to exit the editor, and return to **Catalog** page. Click **Publish**, to publish the pipeline.



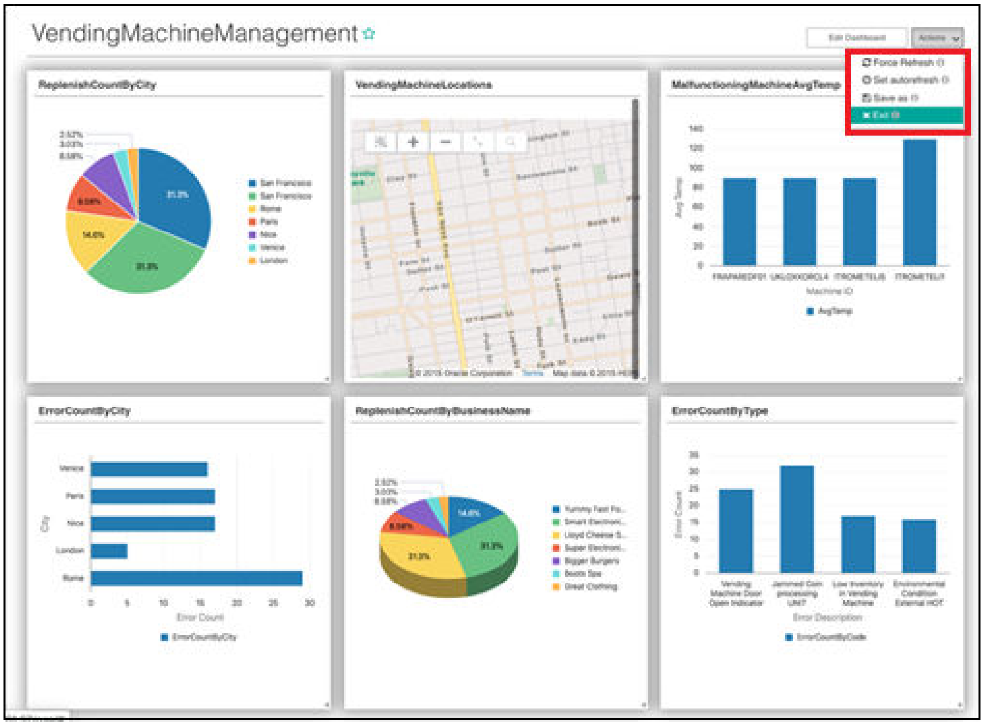
VendingMachineManagement
Pipeline to manage vending machine health and inventory.
Pipeline
Draft
IOT-Sample
Publish

4. Click the dashboard associated with the Pipeline, after successfully publishing the pipeline.



VendingMachineManagement
Dashboard for managing vending machines.
Dashboard
IOT-Sample
Dashboard for managing vending machines.
RUNNING
IOT-Sample
1/11/20 9:34 AM

The Dashboard displays the live results. Click **Exit**, to exit the Dashboard and return to the **Catalog** page.



3.2.2 Other Available Samples on Cloud Marketplace

This section lists all the available sample pipelines on the Cloud Marketplace.

Risk and Fraud Management Sample

The risk and fraud management sample analyzes a credit card transaction stream for potential fraud. In this example, in-store purchases that are 500 miles apart and within a span of few hours, from the same credit card number is deemed fraudulent. The dashboard displays fraud stats by card-type, location, etc.

Transportation and Logistics Sample

This sample analyzes data from city buses for driving violations. Consistent violations are sent to a downstream insurance pricing application, to mitigate risk and ensure public safety. The dashboard displays driving violations by driver age, highways where violations are occurring, etc.

Customer Experience and Consumer Analytics Sample

The pipeline in this example ingests service-request call data and applies rules to analyze the likelihood of a customer leaving the merchant. The rules consider facts like number of days a service request has been pending, the number of follow-up calls the customer made, previous customer purchases, etc., to predict churn. Potential churn can then be mitigated by proactively reaching out to customers with offers, rewards, etc.

Telecommunications Sample

The telecom sample analyzes logs from an Nginx reverse proxy to proactively alert on security violations, application outages, traffic patterns, redirects, etc.

Retail Sample

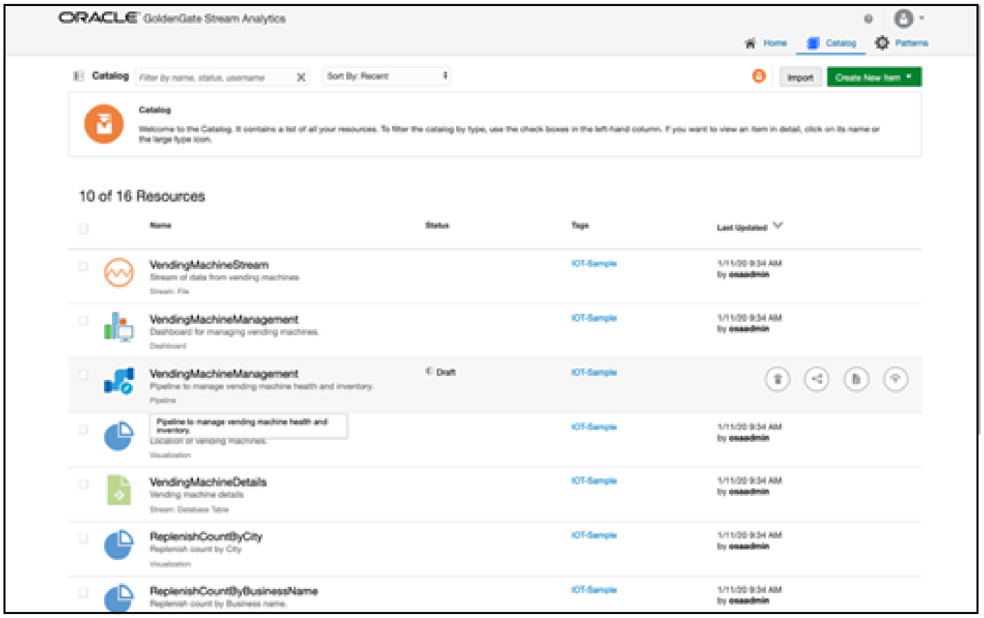
The retail sample analyzes credit card purchases in real-time and drives additional sales by making compelling offers. Offers are made based on customer location, past purchase history, etc. The sample also uses a Machine Learning model to score the likelihood of the customer redeeming the offer. Offers are dropped if the likelihood of redemption is low.

3.3 Monitoring Pipelines from the Spark Console

You can monitor the pipelines running in local Spark cluster from the Spark Console.

To open the Spark console, type `https://<IP_of_Instance/spark>` in a Chrome browser, enter the Spark user and password you used in the Securing your GGSA Instance section. You will see a list of Running and Completed applications.

Pipelines in a Draft state are automatically undeployed after exiting the Pipeline editor, while pipelines in Published state continue to run as shown in the screenshot below.


ORACLE GoldenGate Stream Analytics
Home
Catalog
Patterns
Catalog
Filter by name, status, username
Sort By: Recent
Import
Create New Item

Catalog

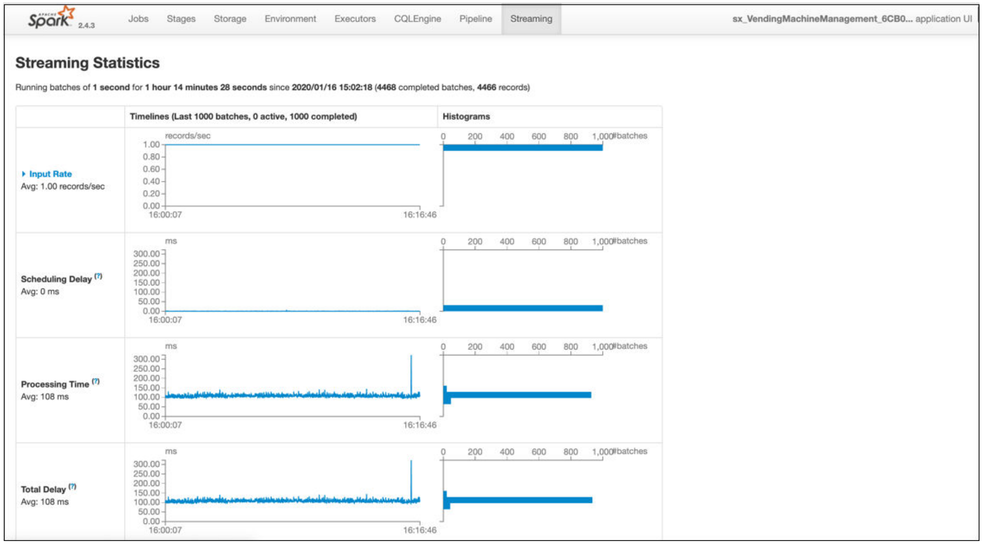
Welcome to the Catalog. It contains a list of all your resources. To filter the catalog by type, use the check boxes in the left-hand column. If you want to view an item in detail, click on its name or the large type icon.

10 of 16 Resources

Name	Status	Tags	Last Updated
VendingMachineStream Stream of data from vending machines Stream: File		IOT-Sample	1/11/20 9:34 AM by osaadmin
VendingMachineManagement Dashboard for managing vending machines. Dashboard		IOT-Sample	1/11/20 9:34 AM by osaadmin
VendingMachineManagement Pipeline to manage vending machine health and inventory. Pipeline	Draft	IOT-Sample	
Pipeline to manage vending machine health and inventory. Location of vending machines. Visualization		IOT-Sample	1/11/20 9:34 AM by osaadmin
VendingMachineDetails Vending machine details Stream: Database Table		IOT-Sample	1/11/20 9:34 AM by osaadmin
ReplenishCountByCity Replenish count by City Visualization		IOT-Sample	1/11/20 9:34 AM by osaadmin
ReplenishCountByBusinessName Replenish count by Business name.		IOT-Sample	1/11/20 9:34 AM by osaadmin

Pipelines in a Draft state end with the suffix _draft. The **Duration** column indicates the duration for which the pipeline has been running.

You can also view the Streaming statistics by clicking **App ID**-> **Application Detail UI** -> **Streaming**.



Streaming Statistics

Running batches of **1 second** for **1 hour 14 minutes 28 seconds** since **2020/01/16 15:02:18** (**4468** completed batches, **4466** records)

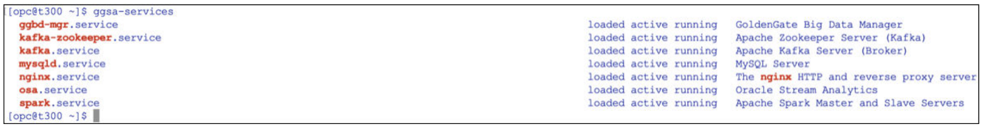
	Timelines (Last 1000 batches, 0 active, 1000 completed)	Histograms
Input Rate Avg: 1.00 records/sec		
Scheduling Delay (?) Avg: 0 ms		
Processing Time (?) Avg: 108 ms		
Total Delay (?) Avg: 108 ms		

The Streaming statistic page displays the data ingestion rate, processing time for a micro-batch, scheduling delay, etc. For more information on these metrics, please refer to the Spark documentation.

3.4 Monitoring GoldenGate Change Data

You can check the status of GoldenGate Big Data Manager by running `ggsa-services`. Status must be active and running.

To start the services, see Starting and Stopping GGSA Services.



```
[opc@t300 ~]$ ggsa-services
  ggbd-mgr.service                                                                      loaded active running   GoldenGate Big Data Manager
  kafka-zookeeper.service                                                               loaded active running   Apache Zookeeper Server (Kafka)
  kafka.service                                                                         loaded active running   Apache Kafka Server (Broker)
  mysqld.service                                                                        loaded active running   MySQL Server
  nginx.service                                                                         loaded active running   The nginx HTTP and reverse proxy server
  osa.service                                                                           loaded active running   Oracle Stream Analytics
  spark.service                                                                         loaded active running   Apache Spark Master and Slave Servers
[opc@t300 ~]$
```

Pipelines ingesting change data has a GG Analytics artifact created as **GG Change Data** from **Create New Item**.

To check if Replication is successful, you can navigate to `/u01/app/ggbd/OGG_BigData_Linux_x64_19.1.0.0.*.*` and run `./ggsci` to launch the command interface for GoldenGate.

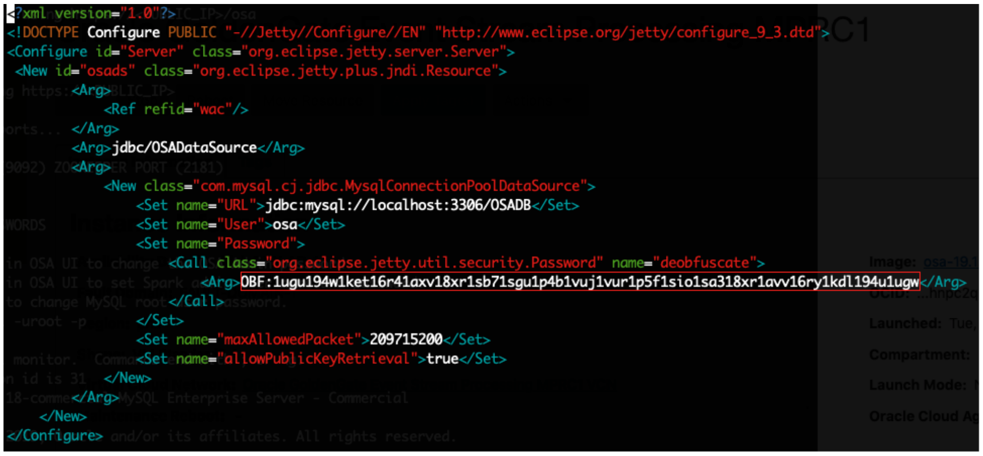
At the command prompt, check the status of Manager and Replicats, if any by typing `info all`.

4
Upgrading a GGSA Instance

This chapter describes how to Upgrade an existing GGSA instance.

4.1 Upgrading to newer GGSA version

1. Note down your current Mysql Root Password, Mysql OSA User password and the password for osaadmin user. If you have not changed these passwords, you can check these in `README.txt` home directory of opc user.
2. Connect to the instance through SSH, and copy the obfuscated password for OSA user in Mysql from `/u01/app/osa/osa-base/etc/jetty-osa-datasource.xml`.



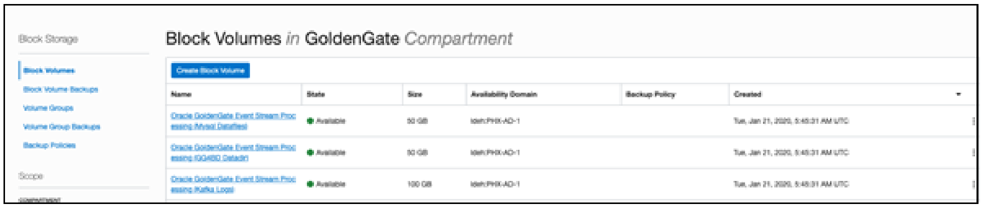
```
<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/configure_9_3.dtd">
<Configure id="Server" class="org.eclipse.jetty.server.Server">
 <New id="osads" class="org.eclipse.jetty.plus.jndi.Resource">
      <Arg>
          <Ref refid="wac"/>
      </Arg>
      <Arg>jdbc/OSADataSource</Arg>
      <Arg>
          <New class="com.mysql.cj.jdbc.MysqlConnectionPoolDataSource">
              <Set name="URL">jdbc:mysql://localhost:3306/OSADB</Set>
              <Set name="User">osa</Set>
              <Set name="Password">
                  <Call class="org.eclipse.jetty.util.security.Password" name="deobfuscate">
                      <Arg>OBF:1ugu194w1ket16r41axv18xr1sb71sgu1p4b1vuj1vur1p5f1sio1sa318xr1avv16ry1kdl194u1ugw</Arg>
                  </Call>
              </Set>
              <Set name="maxAllowedPacket">209715200</Set>
              <Set name="allowPublicKeyRetrieval">true</Set>
          </New>
      </Arg>
    </New>
</Configure>
```

3. Terminate the existing OCI VM on which the current GGSA instance is running. To terminate the instance, go to Compute > Instances. Select your instance running GGSA and in the **Instance Details** page, select **terminate** in the Actions drop down.

> **Note:**
>
> Ensure that you delete just the OCI VM running GGSA and not the complete infrastructure. If you use **Terraform Delete** action against the stack, it will delete all the artifacts including block volumes, network.

4. On the OCI Marketplace home page, click **GoldenGate Stream Analytics** to view the page below. Select the GGSA version, and compartment. Agree to the Oracle standard terms and restrictions, and click on **Launch Stack**.
5. In the Create Stack wizard, provide the essential information such as:
   - **Stack Information Screen**
   - **Configure Variables**
     a. In the **Instance Settings** section, select **Use Existing Block Storage Volumes**.
     b. For Block Volumes for Kafka Logs, Mysql Datafiles and GG4BD Datadir, provide the ID of the existing Block Volumes, that were attached to your GGSA instance

that you terminated earlier. To find the same, go to the top level menu, **Block Storage** and to **Block Volume**. You can identify the block volumes by their names, which are similar to the name of your instance host and will have Kafka Logs, Mysql Datafiles, GG4BD Datadir in brackets.



Block Volumes *in* GoldenGate *Compartment*

Name	State	Size	Availability Domain	Backup Policy	Created
Oracle GoldenGate Event Stream Processing (Mysql Datafiles)	● Available	50 GB	ideh:PHX-AD-1		Tue, Jan 21, 2020, 5:45:31 AM UTC
Oracle GoldenGate Event Stream Processing (GG4BD Datadir)	● Available	50 GB	ideh:PHX-AD-1		Tue, Jan 21, 2020, 5:45:31 AM UTC
Oracle GoldenGate Event Stream Processing (Kafka Logs)	● Available	100 GB	ideh:PHX-AD-1		Tue, Jan 21, 2020, 5:45:31 AM UTC

c. **Review**

.

6. Click **Create**. This will create the OCI Resource Manager Stack, and also trigger the Terraform Apply action on that stack to create the OCI resources.
7. After the **Terraform Apply** action is complete, go to the **Instances** page to find the details of the new instance created.
8. Connect to the new instance using SSH and update the MYSQL OSA User password in `/u01/app/osa/osa-base/etc/jetty-osa-datasource.xml`, using the password copied in Step 2.
9. Restart the OSA service using the commands

```
sudo systemctl reset-failed
```

```
sudo systemctl restart osa
```

10. If the existing pipelines are deployed on the internal spark cluster, republish them after the upgrade, failing which, they will be Killed when the existing GGSA instance is terminated.

4.2 Upgrading to GGSA Version 19.1.0.0.9

This chapter describes the prerequisites and the steps to upgrade to GGSA version 19.1.0.0.9.

To upgrade to GGSA 19.1.0.0.9:

1. Upgrade GGSA
2. Upgrade Spark
3. Upgrade Kafka

4.2.1 Prerequisites

1. Zip file for the latest GGSA to which you want to upgrade. Example,`osa-19.1.0.0.9.zip`.
2. Kakfa zip file. Example, `kafka 3.7.0`.

   .

3. Spark 3.5.0 zip file.

   Download the `spark-3.5.0-bin-hadoop3.tgz` file from https://archive.apache.org/dist/spark/spark-3.5.0/.

4.2.2 Upgrading GGSA

1. Stop the OSA services by running the command, `sudo systemctl stop osa.service`.

   > **Note:**
   >
   > You can check if the service is stopped by running the command `ggsa-services`.

2. Go to GGSA home by running the command, `cd /u01/app/osa`.
3. Create a backup of the old data, and move the backup folder into the OSA folder:
   - mkdir backup
   - mv * backup/
4. Copy the new OSA zip to OSA parent location, and extract the file. From the extracted folder, move all the contents to `/u01/app/osa` location, or the OSA parent folder.
5. Copy `sample` folder from backup to OSA parent folder.
6. Copy `deployedpipelines` folder under OSA parent folder from backup.
7. Copy `jetty-osa-datasource.xml` file from backup to newer version. The file is located in `osa/osa-base/etc/jetty-osa-datasource.xml`.

4.2.3 Upgrading Spark

1. Stop the spark worker and master services by running the following commands in the given order:
   a. `sudo systemctl stop spark-slave.service`
   b. `sudo systemctl stop spark-master.service`
2. cd /u01/app/spark
3. Create a backup folder and copy all the contents of spark inside it.
4. Copy the new spark zip file and extract it. Move all contents from the extracted folder to`spark_home` folder: `/u01/app/spark`
5. Copy `spark-defaults.conf` and `spark-env.sh` from backup `spark conf` folder to new `spark conf` folder
6. Copy jars from OSA to spark home:
   a. `cp ${OSA_HOME}/lib/jetty-util-9.4.53.v20231009.jar ${SPARK_HOME}/jars/jetty-util-9.4.53.v20231009.jar`
   b. `cp ${OSA_HOME}/osa-base/bin/osa.installer.jar ${SPARK_HOME}/jars/osa.installer.jar`
   c. `cp ${OSA_HOME}/osa-base/lib/ext/mysql-connector-java-8.0.12.jar ${SPARK_HOME}/jars/mysql-connector-java-8.0.12.jar`
   d. `cp ${OSA_HOME}/osa-base/lib/ext/ojdbc8.jar ${SPARK_HOME}/jars/ojdbc8.jar`

   e. `cp ${OSA_HOME}/osa-base/lib/ext/osa.spark-integration.model.jar ${SPARK_HOME}/jars/osa.spark-integration.model.jar`

7. Start the spark master and worker by running the following commands in the given order:
   a. `sudo systemctl restart spark-master.service`
   b. `sudo systemctl restart spark-slave.service`

If you encounter any issues, check the logs, or run the `ggsa-services` command.

4.2.4 Upgrading Kafka

Creating Backup for Kafka Log Files

1. Create a backup of the Kafka data:
   a. `cd /u02/kafka`
   b. `mkdir backup`
   c. `cp -r kafka-logs backup/`
   d. `cp -r zookeeper backup/`

Upgrading Kafka Broker

1. Go to Kafka home folder: `cd /u01/app/kafka`.
2. Stop Kafka broker services by running the command: `sudo systemctl stop kafka.service`.
3. Create a backup folder, by running the command, `mkdir backup`, and move all files into the backup folder.
4. Copy the Kafka 3.4.0 zip and extract the folder contents.
5. Move all the extracted files to Kafka parent folder: `/u01/app/kafka`.
6. Copy config folder files from the backup to the config folder of the new kafka.
7. Open the `server.property` file under the config folder and add the following parameters:

   - `inter.broker.protocol.version=<version_you_are_upgrading_from>`

     For example,

     ```
     inter.broker.protocol.version=2.1.1
     ```

   - `log.message.format.version=<version_you_are_upgrading_from>`

     For example,

     ```
     log.message.format.version=2.1.1
     ```

8. Start the broker by running the command, `sudo systemctl restart kafka.service`, and wait for a few minutes for the cluster to stabilize; check logs for any issues.

9. Stop the broker again by running the command, `sudo systemctl stop kafka.service`, and update the protocol again.
10. Stop the broker by running the command, `sudo systemctl stop kafka.service`, and update the message format.
11. Start the broker by running the command, `sudo systemctl restart kafka.service`, and wait for a few minutes for the cluster to stabilize; check logs for any issues.
12. Start the broker by running the command , `sudo systemctl restart kafka.service`, and wait for a few minutes for the cluster to stabilize; check logs for any issues.

Now you have the latest Kafka broker running.

Upgrading Zookeeper

1. Go to Kafka parent folder i.e /u01/app/kafka
2. Stop kafka broker and kafka-zookeeper by running the following commands:

   ```
   sudo systemctl stop kafka.service
   ```

   ```
   sudo systemctl stop kafka-zookeeper.service
   ```

3. Go to config folder and open **zookeeper.properties** file
4. Add `admin.serverPort` property in the file and assign it to some port number which is not being used by any other services:

   ```
   admin.serverPort=9876
   ```

   or to disable this feature, set

   ```
   admin.enableServer=false
   ```

5. Start the zookeeper by running the command, `sudo systemctl restart kafka-zookeeper.service`.
   - Check if the zookeeper services are up, by running the command, `ggsa-services`.
   - If the services are not up, check the zookeeper logs for errors.
   - If the error is related to snapshot, then download the snapshot file from https://issues.apache.org/jira/browse/ZOOKEEPER-3056, and copy it to the location, `/u02/kafka/zookeeper/version-2/`.
   - Open the zookeeper.properties file, add the property, `zookeeper.snapshot.trust.empty=true`, and restart the zookeeper by running the command, `sudo systemctl restart kafka-zookeeper.service`.
6. After the zookeeper gets started, start the Kafka broker by running the command, `sudo systemctl restart kafka.service`.
7. Check if the services `kafka.service` and `kafka-zookeeper.service` are running fine.

Now you have an upgraded Kafka.

5

Upscaling and Downscaling GoldenGate Stream Analytics Compute Shape Sizes

This chapter describes how to upscale or downscale the compute shape sizes of your GGSA instance on Oracle Cloud Marketplace compute node.

To change the size of the your GGSA Compute Shape:

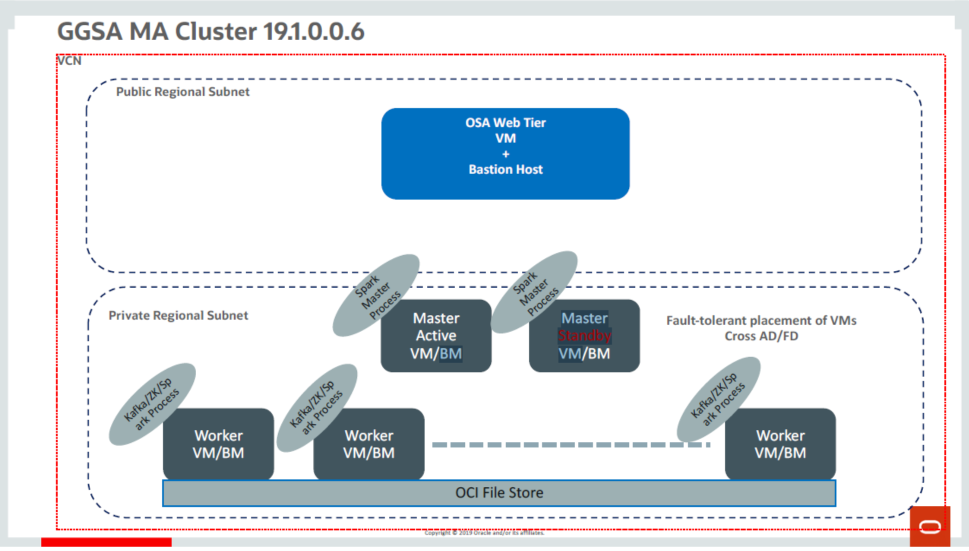
1. Login to your OCI Tenancy.
2. From the **Context** menu on the top left corner of your screen, select **Compute**, and then select **Instances**.
3. Select the GGSA Compute Node to edit.
4. From the **Instance Details** screen, locate the **Edit** drop down menu.
5. From the **Edit** drop down menu, select the option **Edit Shape**. A list of available shapes appear, allowing you to select the desired shape for your selected compute instance.
6. Select the required compute shape and click **Change Shape**.

This restarts your compute node.

6 Creating a Production Cluster for GGSA

This chapter describes how to create a production cluster for GGSA on OCI, with an example production cluster with six compute instances configured as follows:

A production cluster should have a minimum of 1 web-tier compute instance, 2 spark master compute instances, and a variable number of worker instances for Spark workers and Kafka. A minimum of 2 worker instances are required to run spark workers and Kafka.



1. A single compute instance running GGSA Web-tier. Minimal shape is VM 2.4. This web-tier instance also acts as a Bastion Host for the cluster and should be created in a public regional subnet.
2. Two spark master compute instances running spark master processes. Minimal shape is VM 2.2. Both instances can be of same shape and should be created in a private regional subnet.
3. Three worker instances running Spark worker, Kafka broker, and Zookeeper processes. Minimal shape is VM 2.2. All three instances can be of same shape and should be created in a private regional subnet.

All instances will use the GGSA image that comes packaged with GGSA, GGBD, Spark, Kafka, MySQL, and Nginx.

To provision a GGSA production cluster you must first provision GGSA compute instances, using the custom GGSA VM image. The GGSA VM image contains binaries for GGSA, GGBD, Spark, Kafka, MySQL, OSA, Nginx, etc. You do not require any additional software other than the GGSA image. The image packages the following scripts:

- `init-kafka-zk.sh`: Bash script to initialize Zookeeper and Kafka on worker VMs
- `init-spark-master.sh`: Bash script to initialize Spark master on Spark master VMs
- `init-spark-slave.sh`: Bash script to initialize Spark worker on worker VMs
- `init-web-tier.sh`: Bash script to initialize GGSA web-tier on the Web-tier VM

Infrastructure resources must be created and scripts must be run in the order as specified:

6.1 Creating Network Components

First version of Terraform could create new network components but subsequent versions should allow use of existing VCN, Subnet, etc.

1. Accept compartment as input. The above topology will be provisioned in compartment specified.
2. Create a new VCN with CIDR block 10.0.0.0/16, for example, `ggsa-ha`.
3. Create an Internet Gateway, for example, `ggsa-ig` for VCN `ggsa-ha`.
4. Add a route rule to the VCN’s route table with the following attributes:
   - Target Type = Internet Gateway
   - Destination CIDR Block = 0.0.0.0/0
   - Target Internet Gateway = `ggsa-ig`
5. Create a new public regional-subnet, for example, `your-vcn-name-public`, with CIDR block 10.0.0.0/24.
6. Create a new private regional-subnet, for example, `your-private-regional-subnet`, with CIDR block 10.0.1.0/24.

6.2 Creating Security Lists and Associating with Subnets

1. Create a new security list for the vcn `your-vcn-name`, for example, `ggsa_web_app_sec_list` and add the following stateful ingress rules with the following attributes:
   a. - Stateless = unchecked
      - Source CIDR = 0.0.0.0/0
      - Protocol = TCP
      - Source Port Range = All
      - Destination Port Range = 443
      - Description = Internet access to GGSA web application
   b. - Stateless = unchecked
      - Source CIDR = 10.0.0.0/16
      - Protocol = TCP
      - Source Port Range = All

  - Destination Port Range = All
  - Description = Traffic from GGSA web-tier to GGSA runtime

2. Associate security list `ggsa_web_app_sec_list` with subnet `your-vcn-name-public`.
3. Create another security list for the `your-vcn-name` vcn, for example `ggsa_runtime_sec_list` and add a stateful ingress rule with attributes:
   - Stateless = unchecked
   - Source CIDR = 10.0.0.0/16
   - Protocol = TCP
   - Source Port Range = All
   - Destination Port Range = All
   - Description = Traffic from GGSA web-tier to GGSA runtime
4. Associate security list `ggsa_runtime_sec_list` with subnet `your-private-regional-subnet`.

6.3 Creating Storage

1. Using OCI File Storage, create a MySQL file system with attributes.

   > **Note:**
   >
   > The export path names have to be exactly as defined below.

   - File System Name = `MYSQL`
   - Availability Domain = Any Domain
   - Export Path = `/u02/mysql`
   - Mount Target = `GGSMT`

     > **Note:**
     >
     > Mount Target IP address is required for subsequent steps. Note it down.

   - VCN = `your-vcn-name`
   - Subnet = `your-private-regional-subnet`
2. Create Kafka file system with attributes below. Use the same Mount Target as in **Step 1** because Mount Targets are expensive, and also the scripts assume a single Mount Target.
   - File System Name = KAFKA
   - Availability Domain = **Same domain as Step 1**
   - Export Path = `/u02/kafka`
   - Mount Target = `GGSMT` (**same as Step 1**)
   - VCN = `your-vcn-name`
   - Subnet = `your-private-regional-subnet`

3. Create Spark/Ignite file system with attributes below. Use the same Mount Target as in **Step 1** because Mount Targets are expensive, and also the scripts assume a single Mount Target.
    - File System Name = SPARK
    - Availability Domain = **Same domain as Step 1**
    - Export Path = `/u02/spark`
    - Mount Target = `GGSMT` (**same as Step 1**)
    - VCN = `your-vcn-name`
    - Subnet = `your-private-regional-subnet`
4. Create GoldenGate Big Data file system with attributes below. Use the same Mount Target as in **Step 1** because Mount Targets are expensive, and also the scripts assume a single Mount Target.
    - File System Name = GGBD
    - Availability Domain = **Same domain as Step 1**
    - Export Path = `/u02/ggbd`
    - Mount Target = `GGSMT` (**same as Step 1**)
    - VCN = `your-vcn-name`
    - Subnet = `your-private-regional-subnet`

6.4 Creating Worker VMs and Configuring ZK plus Kafka on each VM

1. Create requested number of Worker VMs in the private regional subnet, `your-private-regional-subnet` using the GGSA image. All Worker VMs should be of same compute shape. Minimum shape is VM Standard 2.2.

    > **Note:**
    >
    > IP addresses of all these Worker VMs are required for the next step. Save them as a space separated list, for example, `Worker VM IP List`.

2. SSH and execute the script `init-kafka-zk.sh` on each Worker VM. The usage for the script is as follows and all arguments to the script must be separated by space.

    > **Note:**
    >
    > The Server ID must be incremental starting with 1 and up to the number of workers. For example, if we have three Worker VMs, the ServerID will start with 1 and end at 3.

    ```
    $OSA_HOME/osa-base/bin/init-kafka-zk.sh <ServerId> <MountTargetIP> <IP Address
    List of Workers>
    ```

    Examples:

- When running on Worker VM1:
  `$OSA_HOME/osa-base/bin/init-kafka-zk.sh 1 10.0.1.73 10.0.0.45 10.0.0.46 10.0.0.47`
- When running on Worker VM2:
  `$OSA_HOME/osa-base/bin/init-kafka-zk.sh 2 10.0.1.73 10.0.0.45 10.0.0.46 10.0.0.47`

A successful run has Kafka and Zookeeper servers running on all Worker VMs.

You can use an OCI Streaming Service (OSS) connection instead of Kafka or Zookeeper. See Creating a Connection to OSS.

6.5 Creating Spark Master VMs and Configuing a Spark Master on each VM

1. Create 2 Spark Master VMs in the private regional subnet `your-private-regional-subnet` using the GGSA image. Spark Master VMs should be of same compute shape – Minimum shape is VM Standard 2.1.

   > **Note:**
   >
   > IP addresses of these Master VMs are required for subsequent steps. Save them as a space separated list, for example, `Master VM IP List`.

2. SSH and execute the script `init-spark-master.sh` on each Master VM. The usage for the script is as follows and all arguments to the script must be separated by space.
   `$OSA_HOME/osa-base/bin/init-spark-master.sh <MountTargetIP> < Worker VM IP List >`

   Example: `$OSA_HOME/osa-base/bin/init-spark-master.sh 10.0.1.73 10.0.0.45 10.0.0.46 10.0.0.47`

On successful run, Spark Master processes must be running on Spark Master VMs.

6.6 Initializing the Spark Worker on Worker VMs

1. SSH and Execute the script `init-spark-slave.sh` on each Worker VM. Please note this requires the master VM IP list.
   The usage for the script is as follows and all arguments to the script must be separated by space:

   `$OSA_HOME/osa-base/bin/init-spark-slave.sh <MountTargetIP> < Master VM IP List >`

   Example: `$OSA_HOME/osa-base/bin/init-spark-slave.sh 10.0.1.73 10.0.0.95 10.0.0.96`

On successful run, Spark Worker processes must be running on all Worker VMs.

6.7 Creating a Web-tier VM and configuring the Runtime Environment

1. Create Web-Tier VM in the public regional subnet `your-vcn-name-public` using the GGSA image. Minimum shape for VM is Standard 2.4.
2. SSH and execute the script `init-web-tier.sh` on Web-tier VM.

> **Note:**
>
> To run this script you require, the Mount Target IP, Spark master 1 IP, Spark Master 2 IP, and one or more Worker Node IPs. The usage for the script is as follows and all arguments to the script must be separated by space. `$OSA_HOME/osa-base/bin/init-web-tier.sh <MountTargetIP> < Master 1 IP>< Master 2 IP><Worker 1IP>...`
>
> Example: `$OSA_HOME/osa-base/bin/init-web-tier.sh 10.0.1.73 10.0.0.95 10.0.0.96 10.0.0.100`

6.8 Validating the Topology

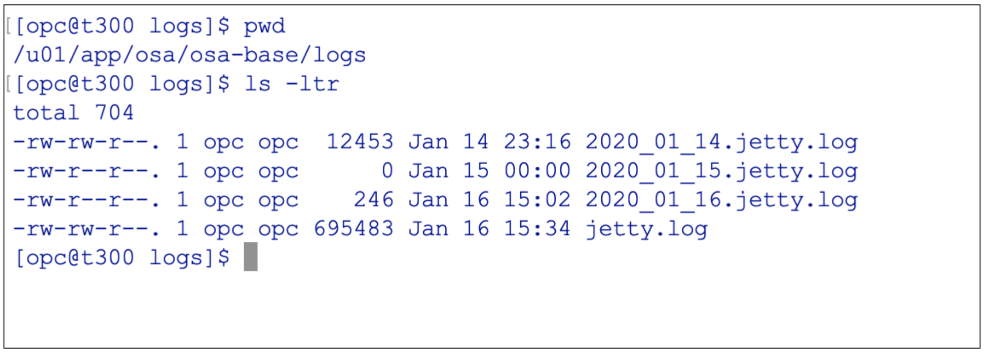
1. From the browser or via curl, open `https://<WebtierPublicIP>`, to see the login page.
2. SSH to the web-tier VM and retrieve the password for osaadmin user .
3. Login to OSA UI and import one of the samples
4. Ensure LocalKafka connection artifact points to zookeepers in system settings .
5. Publish pipeline and view dashboard.
6. Topology is valid if the dashboard is animated.

7 Troubleshooting GGSA on OCI

This chapter describes how to troubleshoot the issues you encounter while using GoldenGate Stream Analytics on OCI.

7.1 GGSA Web Tier

You can find the GoldenGate Stream Analytics Web-tier logs at `/u01/app/osa/osa-base/logs` as shown in the screenshot below:



```
[[opc@t300 logs]$ pwd
/u01/app/osa/osa-base/logs
[[opc@t300 logs]$ ls -ltr
total 704
-rw-rw-r--. 1 opc opc  12453 Jan 14 23:16 2020_01_14.jetty.log
-rw-r--r--. 1 opc opc      0 Jan 15 00:00 2020_01_15.jetty.log
-rw-r--r--. 1 opc opc    246 Jan 16 15:02 2020_01_16.jetty.log
-rw-rw-r--. 1 opc opc 695483 Jan 16 15:34 jetty.log
[opc@t300 logs]$ █
```

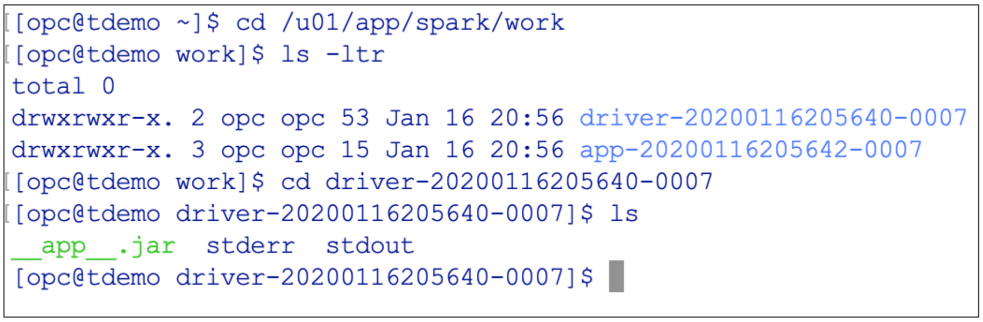
For troubleshooting specific UI issues, see Troubleshoot.

7.2 GGSA Pipeline

Pipelines are deployed to Spark cluster and you can retrieve logs from the Spark console or from the command line. Log into Spark console at `https://<IP_of_Instance/spark`. You are prompted for username and password which you can obtain from the `README.txt` file. Once you provide these credentials, you should see a list of Running Applications. Click the **Application ID** for which you want to retrieve logs.

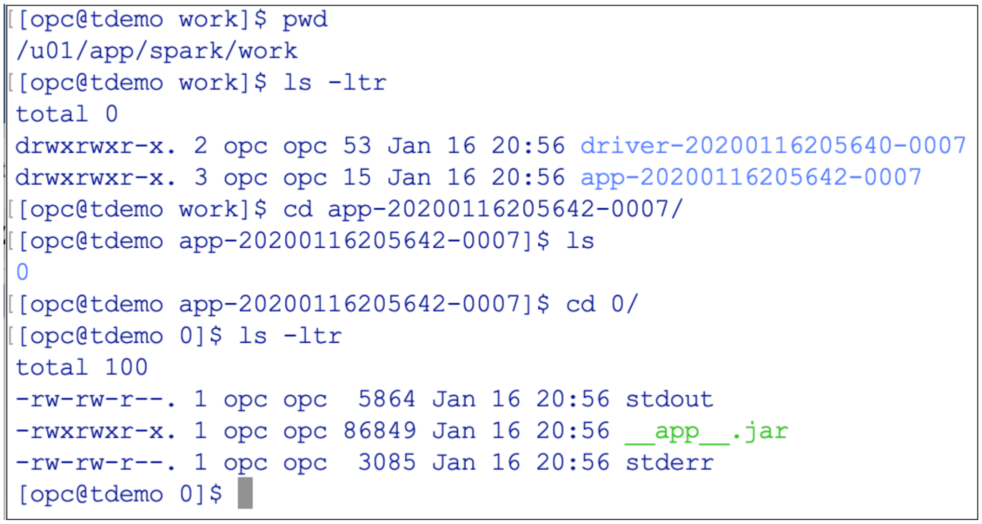
Click on stderr or stdout links in the **Log** column.

When running in the local Spark cluster, logs can also be accessed by navigating to `/u01/app/spark/work` folder. Run `ls -ltr` to view both application and driver logs. For example, you can navigate to Driver logs, as shown in the screenshot below.



```
[[opc@tdemo ~]$ cd /u01/app/spark/work
[[opc@tdemo work]$ ls -ltr
total 0
drwxrwxr-x. 2 opc opc 53 Jan 16 20:56 driver-20200116205640-0007
drwxrwxr-x. 3 opc opc 15 Jan 16 20:56 app-20200116205642-0007
[[opc@tdemo work]$ cd driver-20200116205640-0007
[[opc@tdemo driver-20200116205640-0007]$ ls
__app__.jar  stderr  stdout
[opc@tdemo driver-20200116205640-0007]$ █
```

Similarly you can navigate to application or pipeline logs, as shown in the screenshot below.

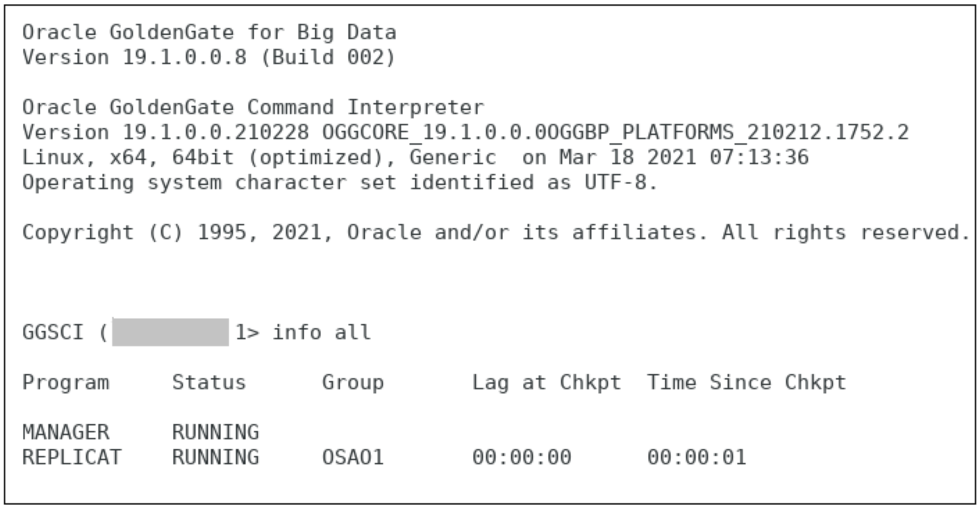


```
[opc@tdemo work]$ pwd
/u01/app/spark/work
[opc@tdemo work]$ ls -ltr
total 0
drwxrwxr-x. 2 opc opc 53 Jan 16 20:56 driver-20200116205640-0007
drwxrwxr-x. 3 opc opc 15 Jan 16 20:56 app-20200116205642-0007
[opc@tdemo work]$ cd app-20200116205642-0007/
[opc@tdemo app-20200116205642-0007]$ ls
0
[opc@tdemo app-20200116205642-0007]$ cd 0/
[opc@tdemo 0]$ ls -ltr
total 100
-rw-rw-r--. 1 opc opc  5864 Jan 16 20:56 stdout
-rwxrwxr-x. 1 opc opc 86849 Jan 16 20:56 __app__.jar
-rw-rw-r--. 1 opc opc  3085 Jan 16 20:56 stderr
[opc@tdemo 0]$
```

For troubleshooting specific UI issues, see Troubleshoot.

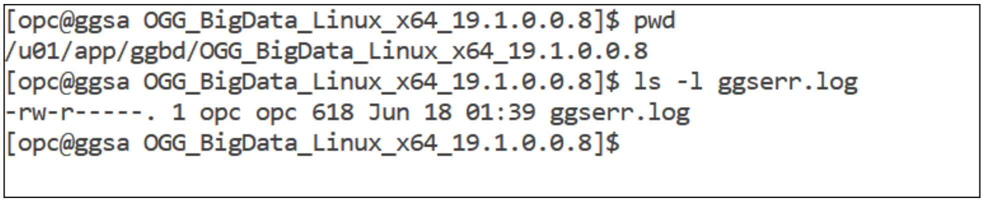
7.3 GoldenGate Change Data

If you see an issue with a pipeline that is processing change data from GoldenGate, check if Replicat processes are running as in the screenshot below.



```
Oracle GoldenGate for Big Data
Version 19.1.0.0.8 (Build 002)

Oracle GoldenGate Command Interpreter
Version 19.1.0.0.210228 OGGCORE_19.1.0.0.0OGGBP_PLATFORMS_210212.1752.2
Linux, x64, 64bit (optimized), Generic  on Mar 18 2021 07:13:36
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2021, Oracle and/or its affiliates. All rights reserved.



GGSCI ([redacted] 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     OSA01       00:00:00      00:00:01
```

If a Replicat is abended, you can check the GoldenGate for Big Data error log `ggserr.log` under the folder `/u01/app/ggbd/OGG_BigData_Linux_x64_19.1.0.0.8/`



```
[opc@ggsa OGG_BigData_Linux_x64_19.1.0.0.8]$ pwd
/u01/app/ggbd/OGG_BigData_Linux_x64_19.1.0.0.8
[opc@ggsa OGG_BigData_Linux_x64_19.1.0.0.8]$ ls -l ggserr.log
-rw-r-----. 1 opc opc 618 Jun 18 01:39 ggserr.log
[opc@ggsa OGG_BigData_Linux_x64_19.1.0.0.8]$
```