

HOL5221

Oracle GoldenGate: Faster Deployments Using Microservices Hands-on Lab

Bobby Curtis

Director, Product Management

Nick Wagner

Director, Product Management

Mack Bell



Senior Product Manager

# HOL5221 – Oracle GoldenGate: Faster Deployments Using Microservices

This hands-on lab is designed to demonstrate how Oracle GoldenGate 19c Microservices can be used to setup a replication environment by a mix of web page, shell scripts and Rest API interfaces. All labs will use shell scripts to facilitate the building of the environment, at the same time provide insight into how to use the web pages and AdminClient. What this hands- on lab will not demonstrate is how to install Oracle GoldenGate 19c and the required ServiceManager.

The labs will walk the end-user through how to add all components of Oracle GoldenGate replication. To do the instantiation of the target database, the end-user will be performing a data pump export and import. All replication process will be started as they are built.

[HOL5221 – ORACLE GOLDENGATE: FASTER DEPLOYMENTS USING MICROSERVICES 2](#_TOC_250010)

[OVERVIEW 3](#_TOC_250009)

Time to Complete 3

Prerequisites 3

Passwords 3

Ports 3

[TASK 1: PREPARATION STEPS 4](#_TOC_250008)

[TASK 2: CONFIGURE DATABASE AND GOLDENGATE USERS 8](#_TOC_250007)

[TASK 3: CREATING DEPLOYMENTS 13](#_TOC_250006)

[TASK 4: CONFIGURE REVERSE PROXY 18](#_TOC_250005)

[TASK 5: CREATE CREDENTIALS 20](#_TOC_250004)

[TASK 6: ADD EXTRACT AND DISTRIBUTION PATH 24](#_TOC_250003)

[TASK 7: INSTANTIATION 27](#_TOC_250002)

[TASK 8: CREATE THE REPLICAT 29](#_TOC_250001)

[EXTRA: GENERATE DATA LOAD 32](#_TOC_250000)

# Overview

**Time to Complete**

Perform all tasks – 45 - 60 Minutes

It is acceptable if you do not finish the complete lab, but we made every effort to ensure that you have the tools to build a fully functional replication environment.

**Prerequisites**

Before you begin this tutorial, you should

* Have a general understanding of database and data replication concepts using Oracle GoldenGate

**Passwords**

The passwords for all accounts are:

Database Accounts (sys/system, etc..): Welcome1 GoldenGate Users (c##ggate, ggate): ggate GoldenGate Admin (oggadmin): Welcome1

**Ports**

This HOL utilizes a lot of port. Many of the ports have been opened for you; however, you will need to open port 443. This ensures that the Reverse Proxy will be accessible when configured.

1. Within Oracle VM Virtualbox Manager, select **HOL5221**
2. Select **Settings**
3. Select **Network**
4. Select **Adaptor 1** -> **Advanced** -> **Port Forwarding**
5. Click the green **plus ( + ) sign**.
6. Enter:
   1. **Name:** https
   2. **Host Port:** 443
   3. **Guest Port**: 443
7. Click **OK** (twice)

**Notes:**

Lab3 – You will need to edit the create\_deployment.sh script to change Lab2 to Lab3

# Task 1: Preparation Steps

For this lab, Oracle GoldenGate and the Oracle Database sources and targets are contained within one VirtualBox image that you have access to at your Linux workstation. All user interactions will be through VNC Client (Remote Desktop Viewer) installed on the workstation, which in turn will be used to connect to your VirtualBox image.

Some steps, will highlight the remote capabilities of Oracle GoldenGate Microserivces. In these areas, you will use the host operating systems web browser to execute some commands/functions.

Start the VirtualBox Image

1. Log into your machine and open Oracle VM VirtualBox from the Desktop.



1. Click on “**HOL5221**” and then click **Start**.

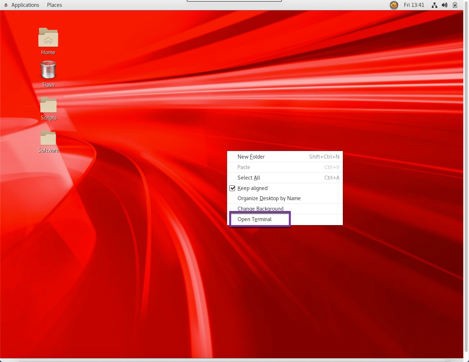
**Note:** Do NOT click on the one named ORI-HOL5221



1. Log in to the VirtualBox image on your workstation, using **Remote Desktop Viewer**.
   1. On your host, go to **Application** -> **Utilities** -> **Remote Desktop Viewer**.
   2. Once Remote Desktop Viewer opens, click Connect
      1. Change **Protocol** to **VNC**
      2. Set **Host** to **localhost:5901**
      3. Click **Connect**
2. Enter ***Welcome1*** in the password field and press **Authenticate**.
3. Once the Remote Desktop Viewer has connected, you should see a console that looks similar to this:



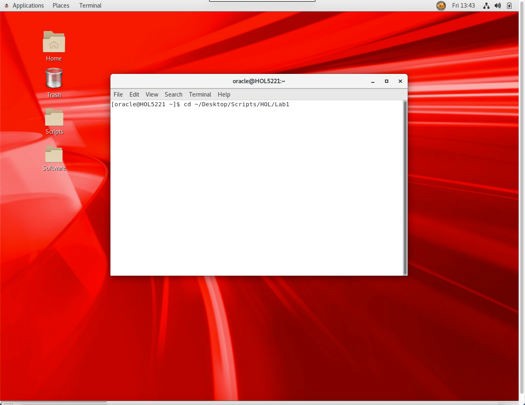
1. Open a Terminal window from the VNC Console.
   1. On the desktop, right-click and select “**Open Terminal**”.



1. After the Terminal window opens, move to the ~/Desktop/Scripts/HOL/Lab1

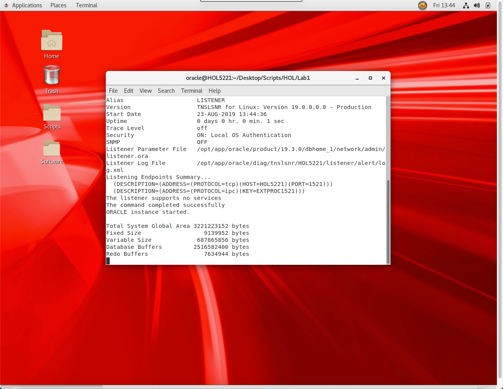
directory.

$ cd ~/Desktop/Scripts/HOL/Lab1



1. Once you are in the Lab1 directory, run the *startup.sh* shell script. This script will start the Oracle Database 19c (19.1) and Listener.

$sh ./startup.sh



Web browser on Linux Host

Part of this lab, you will interact with Oracle GoldenGate by using the local web browser. For this lab, the web brower will be Firefox.

To start Firefox and access the ServiceManager, follow these steps:

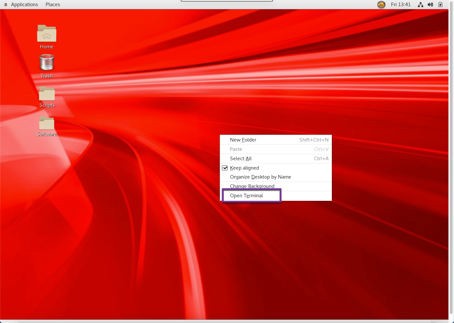
1. On you host machine, go to **Applications** -> **Firefox**
2. Once Firefox is open, navigate to **https://localhost:16000**
3. Accept the security notifications
4. Login to the ServiceManager using oggadmin

After you have successfully logged into the ServiceManager, this task is complete.

# Task 2: Configure Database and GoldenGate Users

In this task, you will configure the Oracle Database for GoldenGate replication and configure required users for Oracle GoldenGate. Perform the following steps:

1. Open a Terminal Window within the VirtualBox via Remote Desktop Viewer
   1. On the Desktop, Right-Click and select Open Terminal



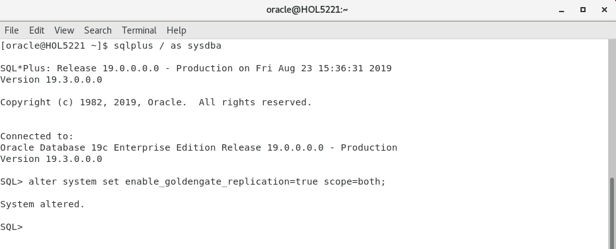
1. In the open Terminal Window, start SQL\*Plus.

$ sqlplus / as sysdba



1. Enable the database for GoldenGate replication

SQL> alter system set enable\_goldengate\_replication=true scope=both;



1. Enable Archive Log on the database. This will require you to shut down the database and restart it.
   1. Shutdown the database

SQL> shutdown immediate;



* 1. Start the database up in mount mode

SQL> startup mount;



* 1. Change database into Archive Log mode

SQL> alter database archivelog;



* 1. Open the database

SQL> alter database open;



* 1. Open the Pluggable Database

SQL> alter pluggable database all open read write;

1. Enable Minimal Supplemental Logging for the database. Additionally, enable Force Logging then switch the log file.

SQL> alter database add supplemental log data; SQL> alter database force logging;

SQL> alter system switch logfile;

1. Create the GoldenGate users needed at the Container Database and Pluggable Database Layers.
   1. From SQL\*Plus run the following SQL statements to create the Common User within the Container Database (CDB).

SQL> create user c##ggate identified by ggate quota unlimited on USERS account unlock;

SQL> grant connect, dba, resource to c##ggate; SQL> begin

SYS.DBMS\_GOLDENGATE\_AUTH.GRANT\_ADMIN\_PRIVILEGE('C##GGATE',

container=>'ALL');

end;

/

b. From SQL\*Plus, run the following SQL statements to create the GoldenGate users for the Pluggable database (PDB).

SQL> alter session set container = oggoow19; SQL> grant connect, dba to c##ggate;

SQL> create user GGATE identified by ggate quota unlimited on USERS account unlock;

SQL> grant connect, dba to ggate;

SQL> alter session set container = oggoow191; SQL> grant connect, dba to c##ggate;

SQL> create user GGATE identified by ggate quota unlimited on USERS account unlock;

SQL> grant connect, dba to ggate; SQL> exit

At this point, you have completed Task 2

# Task 3: Creating Deployments

In this task, you will create two deployments that will be used throughout the rest of the Hands-On Lab.

Deployments are a new concept in Oracle GoldenGate Microservices.

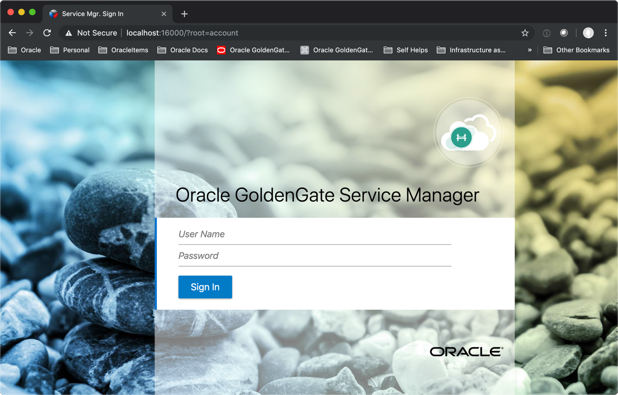
Deployments provide a siloed approach to splitting replication environments between applications, customers, or environments. This allows for greater control over the use of Oracle GoldenGate in larger environments.

**NOTE: The file create\_deployment.sh file needs to be edited. You will need to edit the file and change Lab2 to Lab3.**

1. Log in to ServiceManager’s HTML5 webpage. This is done by opening Firefox on your linux host machine.
2. From the browser, connect to port *16000* to access the ServiceManager login page:

## https://localhost:16000

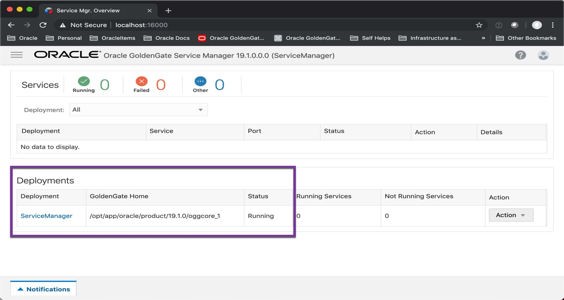
The URL should bring up the following login page.



1. Log in to the Service Manager using the following credentials: User Name: ***oggadmin***

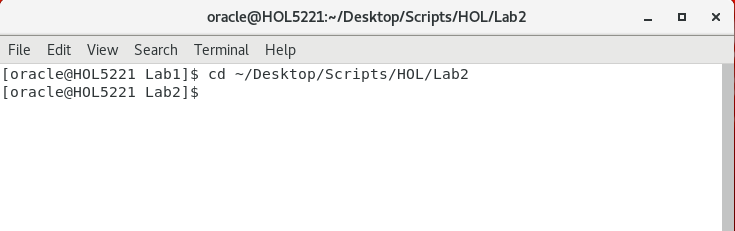
Password: ***Welcome1***

1. You should see “ServiceManager” under **Deployments** at the bottom of the page and the status should be set to “Running”.



1. Now switch back to the Remote Desktop Viewer and from the Terminal window, navigate to the Lab3 directory under ~/Desktop/Scripts/HOL.

$ cd ~/Desktop/Scripts/HOL/Lab3



1. At this point, by doing an “**ls**” in the directory; you will notice two files in the lab directory. The *rsp* file is a template file that will be copied and used by the *sh* file. In order to create two Deployments, the *sh* script file needs to be ran twice.

To run the *create\_deployment.sh* script, you will need to provide eight (8) command line parameters. Here is the template of the command:

*$ sh ./create\_deployment.sh <deployment\_name> <admin password>*

*<SMPort> <ASPort> <DSPort> <RSPort> <PMSPort> <PMSPortUDP>*

Each of the parameters will be used to replace items in the response file and build the Deployment and associated services. Each of the command line parameters corresponds to the following:

**<deployment\_name>** = Name of the deployment to be created

**<admin password>** = Password used by the Security Role user for the ServiceManager

**<SMPort>** = Port number of the ServiceManager (16000)

**<ASPort>** = Port number of the Administration Service (16001)

**<DSPort>** = Port number of the Distribution Service (16002)

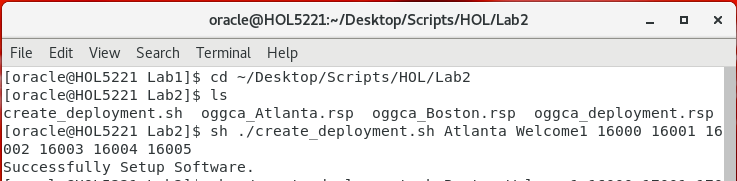
**<RSPort>** = Port number of the Receiver Service (16003)

**<PMSPort>** = Port number of the Performance Metric Service (16004)

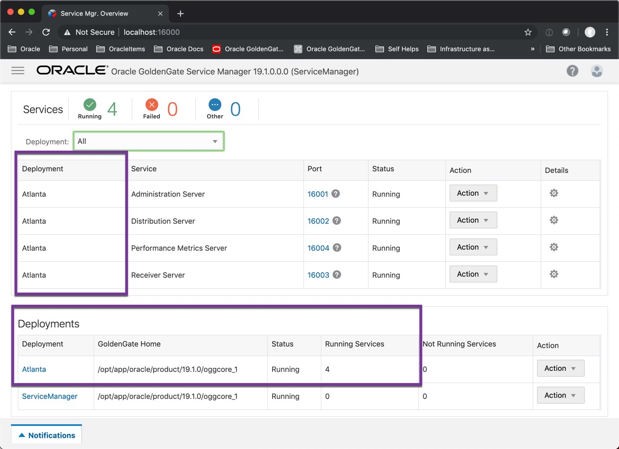
**<PMSPortUDP>** = UDP port number for Performance Metric Service NoSQL Database connection (16005)

Run the script using the following parameter values, to create the **Atlanta** Deployment:

$ sh ./create\_deployment.sh Atlanta Welcome1 16000 16001 16002 16003 16004 16005



1. Return to Firefox and refresh the ServiceManager page. You should have one (1) new Deployment called **Atlanta**, with four (4) services listed.

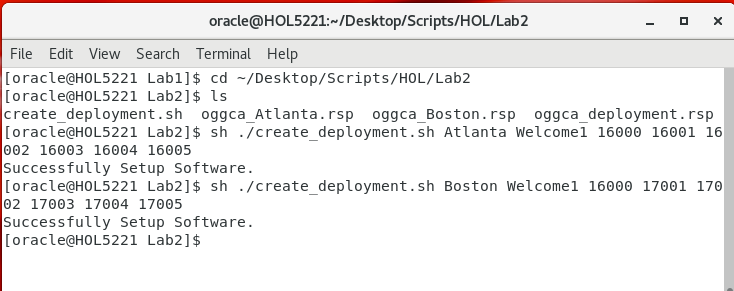


1. Return to the Terminal Window where you ran the *create\_deployment.sh* script and re-run the script again to create a 2nd Deployment (**Boston**), this time changing the Deployment name and all port numbers other than the ServiceManager (16000) port number.

To run the script:

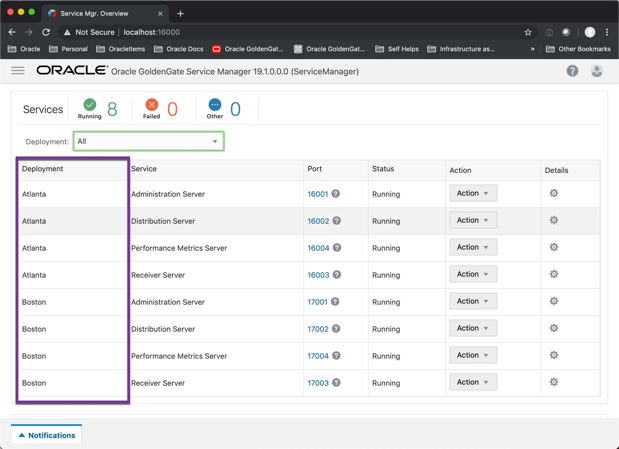
$ sh ./create\_deployment.sh Boston Welcome1 16000 17001 17002

17003 17004 17005



9. Return to Firefox and refresh the ServiceManager page again. You should now have two

(2) Deployments with a total of eight (8) services running.



At this point, you have completed Task 3.

# Task 4: Configure Reverse Proxy

In this Task, you will configure the NGINX Reverse Proxy.

**Note:** Prior to configuring the reverse proxy, you have to have a self-signed certificate**.** These certificates have already been created for you and are stored in ~/wallet. These are the same certificates that were used to setup the security for the ServiceManager and Deployments you created in an Task 3.

To complete this task, you will need to run the following script:

configureNginx.sh

After the completion of this task, accessing the Oracle GoldenGate Microservices HTML5 pages will be simpler.

To begin this task, follow the below steps:

1. From the Terminal window in the VNC Console, navigate to the Lab4 directory under

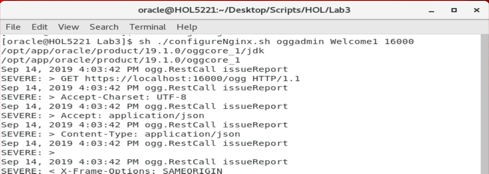
~/Desktop/Scripts/HOL/Lab4

$ cd ~/Desktop/Scripts/HOL/Lab4

1. Execute the script. This script will configure all items related to the Nginx Reverse Proxy

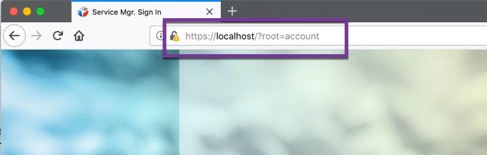
**Note:** The *configureNginx.sh* script is making a call to the ReverseProxySettings utility that can be found under $OGG\_HOME/lib/util/reverseproxy. We provide this script to make it easier for you to configure the Nginx Reverse Proxy in your environment.

$ sh ./configureNginx.sh oggadmin Welcome1 16000



1. Upon completion, return to your web browser. You should be able to access the ServiceManager page by only using the URL without a port number.

## https://localhost



Once you are able to access the ServiceManager by using the simpler URL, you have completed this task.

## Extra Information: Simplified URLs:

The benefit of using the Reverse Proxy is that is makes the URLs simpler to use. If you are so inclined, provide your browser a URL that models this:

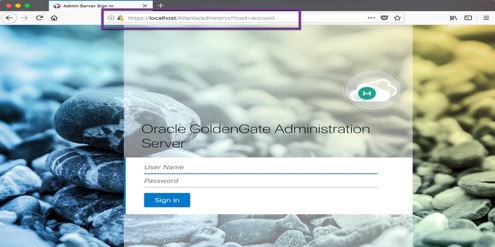
https://localhost/<deployment>/adminsrvr

**<deployment>** = the name of a deployment you build in Task 3.

In the example, if using the Atlanta deployment, the URL would look like this:

https://localhost/Atlanta/adminsrvr

This will take you directly to the Administration Service login page without using port numbers:



# Task 5: Create Credentials

In this Task, you will configure the database user credentials and tnsnames entries needed for replication. This requires running the following scripts:

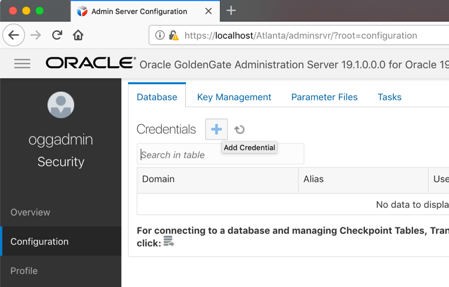
Edit\_tnsnames.sh Create\_credential\_GGAlias.sh Add\_SchemaTrandata.sh

After running these scripts, you will be able to establish connections for replication between the source and target pluggable database.

**Note:** The script create\_credential\_protocol.sh will not be used in this lab and can be ignored.

Before you work through this task, understand that there are multiple ways of doing this lab. The scripts provided are meant to speed up the lab process. If you are interested in creating credentials and adding schematrandata from the web pages or AdminClient the beginning steps are provided below for you.

For web page access, access the Administration Service (adminsrvr) using the simplified URL (https://localhost/<deployment>/adminsrvr). Login and go to Context Menu -> Configuration -> Credentials -> click on the plus ( + ) sign.



To access the AdminClient, open a Terminal Window and execute:

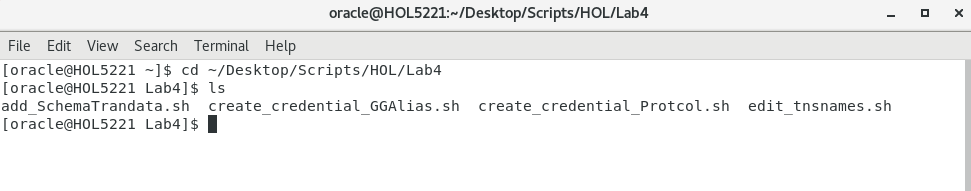
$OGG\_HOME/bin/adminclient

To begin this Task 5, follow the below steps:

1. From the Terminal window in the Remote Desktop Viewer, navigate to the Lab5

directory under ~/Desktop/Scripts/HOL.

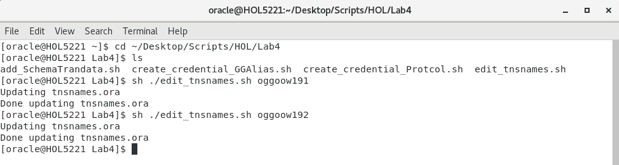
$ cd ~/Desktop/Scripts/HOL/Lab5



1. Edit the tnsnames.ora file using the *edit\_tnsnames.sh* script. This script will need to be ran twice to add entries for both pluggable databases (**oggoow19** & **oggoow191**).

$ sh ./edit\_tnsnames.sh oggoow19

$ sh ./edit\_tnsnames.sh oggoow191



1. Verify that the tnsnames.ora file has been updated.

$ cat $ORACLE\_HOME/network/admin/tnsnames.ora

1. With the tnsnames.ora file updated, you can now create the Oracle GoldenGate Credentials (create\_credential\_GGAlias.sh) needed to connect the capture process (Extract) to the Oracle Database.

In order to create the required credentials, run the following:

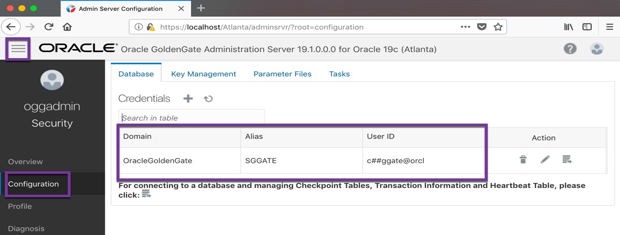
$ sh ./create\_credential\_GGAlias.sh Welcome1 16001 c##ggate@orcl ggate

After running this script, can go to your browser and that the credential was create.

* 1. Open a new browser tab and connect to

https://localhost/<deployment>/adminsrvr

* 1. Login with the following ***oggadmin/Welcome1***
  2. Click the **Context Menu** in the upper left, then select **Configuration** from the left pane



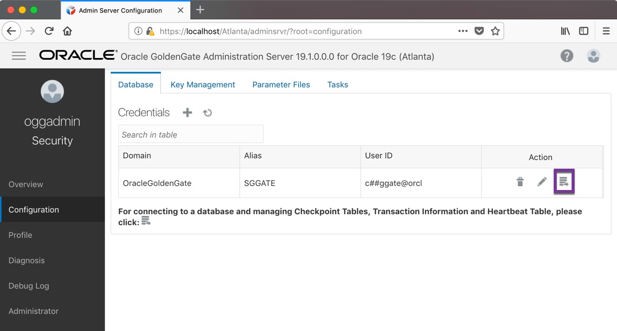
1. Next, we will enable schematrandata on the schema that we want to replicate. In order to do this, you will need to run the *add\_SchemaTrandata.sh* script.

To run this script, execute the following:

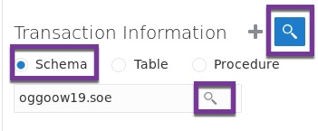
$ sh ./add\_SchemaTrandata.sh Welcome1 16001



You can also check that **SCHEMATRANDATA** has been added from the Administration Service -> Configuration page as well. Simply log in to the ***SGGATE*** alias.



Then, under “*Trandata*”, make sure that the magnifying glass and radio button for “*Schema*” are selected. Enter “***oggoow19.soe***” into the search box and then select the magnifying glass to the right of the search box to perform the search.



After the search is performed, you will see a column that provides the number of tables enabled for supplemental logging within the “*SOE*” schema.



You have now completed configuring schema level supplemental logging needed for use in the replication process.

At this point, you have completed Task 5.

# Task 6: Add Extract and Distribution Path

In this Task, you will create an Extract and a Distribution Path that will be used for replication. Unlike the Oracle GoldenGate Classic architecture, these processes will be created using the RESTful API end-points.

You will use the following two scripts to configure these processes:

Add\_Extract.sh Add\_DistroPath.sh

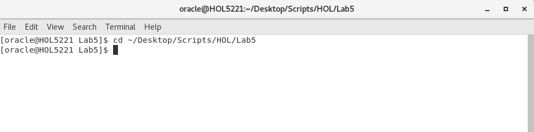
After you build the Extract and Distribution Path, you will be able to see them in the Administration Server and Distribution Server web pages of your deployment.

To begin this Task, follow the below steps:

1. From the Terminal Window in the VNC Console, navigate to the Lab6 directory under

~/Desktop/Scripts/HOL/Lab6.

$ cd ~/Desktop/Scripts/HOL/Lab6

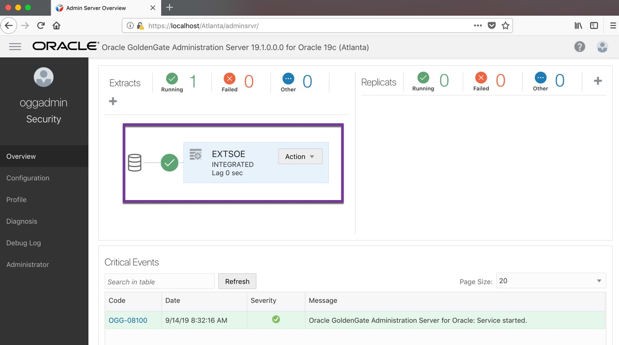


1. Next, to create the Extract you will run the *add\_Extract.sh* script as follows:

$ sh ./add\_Extract.sh Welcome1 16001 EXTSOE

After the script has completed, you can go to the Administration Server and see that the extract is there on the Overview page. Remember to use the short URL to access the Administration Server.

https://localhost/<deployment>/adminsrver



1. Now you will create the Distribution Path that will be used to ship trail files from the Deployment to the Deployment. In order to do this, you will need to run the *add\_DistroPath.sh* script.

The values used in the script correspond to the following:

**“Welcome1”** = OGGADMIN user password

**“16002”** = Atlanta Deployment’s Distribution Service port

**“SOE2SOE”** = Distribution Path name

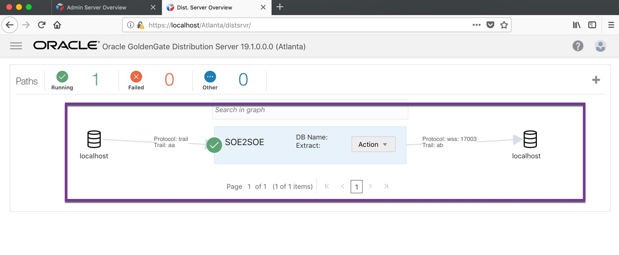
**“aa”** = Atlanta Deployment’s source trail file name prefix **“17003”** = Boston Deployment’s Receiver Service port **“ab”** = Boston Deployment’s remote trail file prefix

$ sh ./add\_DistroPath.sh Welcome1 16002 SOE2SOE aa 17003 ab

After running the add\_DistroPath.sh script, you will see the path created in the Distribution Service. Using the short URL approach, you can quickly see the

Distribution Path. Using your browser navigate to the Distribution Server and review the Distribution Path.

https://localhost/<deployment>/distsrvr



## Extra:

If you are curious about where the trail files are located, you can go look on the VM under the $DEPLOYMENT\_HOME. The default location for trail files is similar to this:

$DEPLOYMENT\_HOME/var/lib/data

This translates to the following for a deployment named Atlanta:

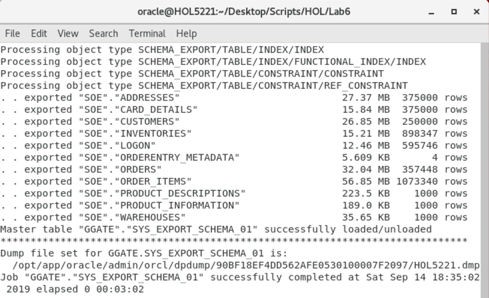
/opt/app/oracle/gg\_deployments/Atlanta/var/lib/data

# Task 7: Instantiation

In this Task, you will be exporting and importing the SOE schema from the source pluggable database (***oggoow19)*** to the target pluggable database (***oggoow191***). This process will use the Oracle Database Data Pump Export and Import procedures.

* 1. On the VM (via VNC) open a Terminal window
  2. Use Oracle Database Export Data Pump (expdb) to export you will export the SOE schema data from the oggoow19 pluggable database using the System user. Make a note of the directory when the export is done. You will have to copy/move this file to the /tmp directory

$ expdp system@oggoow19 directory=DATA\_PUMP\_DIR dumpfile=HOL5221.dmp schemas=soe



* 1. Once the export completes, you will need to import the data into the oggoow191 pluggable database. This will be done using the System user.
     1. Create the SOE data file and tablespace in oggoow191

SQL> alter session set container = oggoow191;

SQL> create tablespace SOE datafile ‘/opt/app/oracle/oradata/ORCL/oggoow191/soe01.dbf size 1024M;

* + 1. Create the SOE user in oggoow191

SQL> create user SOE identified by Welcome1 quota unlimited on SOE account unlock;

SQL> grant connect, resource to SOE;

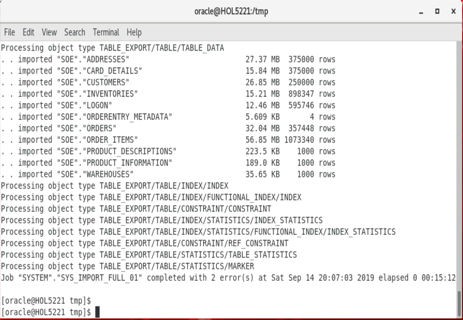
* + 1. Import the SOE data
       1. Identify where the HOL5221.dmp file is located and move the dump file to /tmp
       2. Create the import directory

SQL> create or replace directory IMP\_DUMP\_DIR as ‘/tmp’; SQL> grant read, write on directory IMP\_DUMP\_DIR to system;

SQL> exit

* + - 1. Import Data

$ impdp system@oggoow191 directory=IMP\_DUMP\_DIR dumpfile=HOL5221.dmp full=y



Once the data is imported, this Task 7 is complete.

# Task 8: Create the Replicat

In this Task, you will build the components needed to install the Replicat within the **Boston**

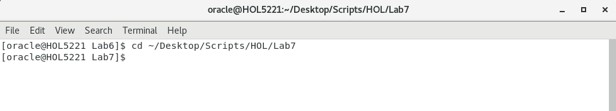
Deployment.

To begin this Task, follow the below steps:

1. From the Terminal window in the VNC Console, navigate to the Lab8 directory under

~/Desktop/Scripts/HOL.

$ cd ~/Desktop/Scripts/HOL/Lab8



1. While in the Lab8 directory, you will create a target database User Alias, Checkpoint Table, and the Replicat. In order to do these steps, there are three scripts that need to be ran:

create\_credential\_GGAlias.sh add\_CheckpointTable.sh add\_Replicat.sh

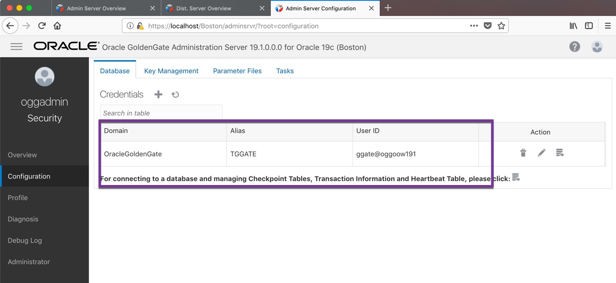
The *create\_credential\_GGAlias.sh* script is very similar to the one that you ran in an earlier Task, the difference being that this alias will be called ***TGGATE*** and will reside in the second Deployment (***Boston***).

To run the *create\_credential\_GGAlias.sh*, type the following in the Terminal window:

$ sh ./create\_credential\_GGAlias.sh Welcome1 17001 ggate@oggoow191 ggate

Upon a successful run, you can check the Administration Services for the **Boston** deployment from within the browser and verify the account was created. Log in with User name ***oggadmin*** and password ***Welcome1*** when prompted.

https://localhost/<deployment>/adminsrvr

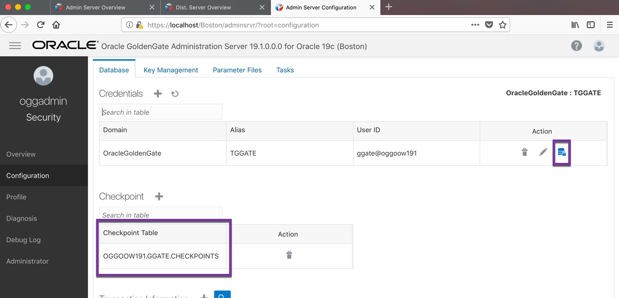


1. Next, you will create a Checkpoint Table to be used by the Replicat. This will be done by using the *add\_CheckpointTable.sh* script.

Run the following command from the Lab8 directory:

$ sh ./add\_CheckpointTable.sh Welcome1 17001

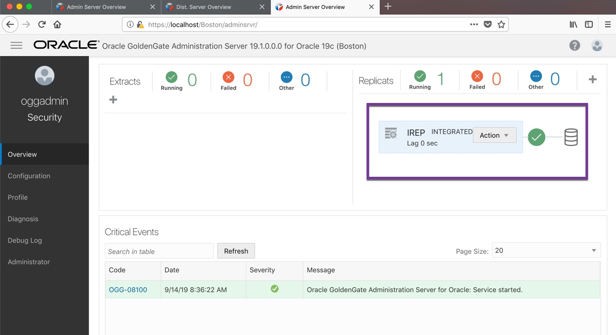
Upon a successful run, the Checkpoint Table can be viewed under the ***TTGATE*** alias from the Boston Deployment’s Administration Service > Configuration page. This is done by clicking the “**log in**” button and then looking under **Checkpoint**.



1. With the target database User Alias and Checkpoint Table created, you can now create the Replicat. In order to create the Replicat, you will need to run the *add\_Replicat.sh* script. Enter the following command to run the script:

$ sh ./add\_Replicat.sh Welcome1 17001 IREP

After the script is done running, you will see a running Replicat in the Administration Service for your deployment.



At this point, you have completed the lab and how-to setup a uni-direction replication using Oracle GoldenGate Microservices. Congrats!

Take some time to poke around and review the architecture and directory structure.

# Extra: Generate Data Load

In the extra task, you will start to generate load on the tables using Swingbench and looking at the Performance Metric Service as it runs. The script that will be used in this extra task is called *start\_swingbench.sh*.

From the HOL directory, you will run the start\_swingbench.sh from the Extra folder. The following scripts will be used:

$ sh ./start\_swingbench.sh

Once Swingbench starts to generate a load, you can view the **Boston** Deployment’s **Performance Metric Service** to view incoming transactions metrics within the Oracle GoldenGate configuration.

To get to the Performance Metric Service for your deployment use the short URL:

https://localhost/<deployment>/pmsrvr

And once logged in to the Performance Metric Service, you can click any of the GoldenGate Processes, such as the ***IREP*** Replicat process, and view more details about it.

