



Initial Configuration

- [New Account](#)
- [Single Sign On \(SSO\)](#)
- [Automatic user and role provisioning \(SCIM\)](#)
- [Operation Role](#)
- [Database, Schema and Stage](#)
- [Data Movement Roles](#)
- [Data Roles](#)
- [Function Roles](#)

⚠ When creating snowflake service user such as Fivetran, DBT ... ensure that each user, particularly the service users, has a unique email address.

We are using `EXTERNAL_OAUTH_SNOWFLAKE_USER_MAPPING_ATTRIBUTE 'EMAIL_ADDRESS'` in the configuration of the Snowflake OAUTH security integration for PowerBI, hence it means that the login process uses the "email" attribute from Snowflake users to map to the user attempting to log in.

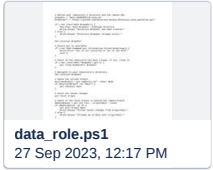
If multiple users or snowflake service users have the same email address, it could result in issues in apps integrated with Snowflake.

Scope	Step	Comment	Snowflake	Others
New Account	1	Create new account. naming convention: <ul style="list-style-type: none"> • DT_<BUSINESS_DOMAIN>_DW for PROD • DT_DEV<BUSINESS_DOMAIN>_DW for non PROD. 	<pre> 1 create account DT_MARKETING_DW 2 admin_name = MARKETING_DW_ADMIN 3 admin_password = 'XXXXX' -- password 4 first_name = THIERRY 5 last_name = KRUMEICH 6 email = 'thierry.krumeich@dynatrace.com' 7 edition = ENTERPRISE 8 region = AWS_US_EAST_1; </pre>	<div> i Save credential in Secret Server, path: EDE > BusinessSystems > Business Intelligence > Internal Accounts - Do Not Share > Snowflake BusSys > <New Subfolder>. Create a new subfolder for the new account. </div> <div> i Ensure sharing is limited to the appropriate user. </div>
	2	Create EDE Ticket to AD Team to configure SSO		Provide Snowflake URL to AD Team
Single Sign On (SSO)	1	Follow snowflake steps from: [Microsoft] Snowflake-AzureAD SSO and [Snowflake] Snowflake-AzureAD SSO	Identify Snowflake Account URL <pre> 1 use role ACCOUNTADMIN; 2 select t.value:type::varchar as type, 3 t.value:host::varchar as host, 4 t.value:port as port 5 from table(flatten(input => parse_json('{"account": "DT_MARKETING_DW", "admin_name": "MARKETING_DW_ADMIN", "admin_password": "XXXXX", "first_name": "THIERRY", "last_name": "KRUMEICH", "email": "thierry.krumeich@dynatrace.com", "edition": "ENTERPRISE", "region": "AWS_US_EAST_1"}'))) 6 where type like 'SNOWFLAKE_DEPLOYMENT'; </pre>	

		for the new Snowflake account.		Collect the SAML2_ISSUER, SAML2_SSO URL and SAML_X509_CERTIFICATE from AD Team.
				<div>  Save information in Secret Server path created step 1. </div>
	3	Complete SSO integration using Snowsight using information from step 2.	<pre> 1 use role ACCOUNTADMIN; 2 create security integration AZUREA 3 type = SAML2 4 enabled = TRUE 5 saml2_issuer = 'https://sts.wi 6 saml2_sso_url = 'https://login 7 saml2_provider = 'CUSTOM' 8 saml2_x509_cert = '<Azure Team 9 saml2_sp_initiated_login_page_ 10 saml2_enable_sp_initiated = TR 11 desc security integration AZUREADI 12 alter account set sso_login_page = </pre>	
	4	Customize SSO Login Page. Naming convention: "Dynatrace SSO Login in <ENV> BUSSYS"	<pre> 1 use role ACCOUNTADMIN; 2 alter integration AZUREADINTEGRATI </pre> <div> <div>▼ SSO Login Page</div>  </div>	
Automatic user and role provisioning (SCIM)	1	Follow snowflake steps from: [Microsoft] AzureAD SCIM and [Snowflake] AzureAD SCIM Enable SCIM integration in Snowflake.	<pre> 1 use role ACCOUNTADMIN; 2 create role if not exists AAD_PROVI 3 grant create user on account to rol 4 grant create role on account to rol 5 grant role AAD_PROVISIONER to role 6 create or replace security integrat 7 type = scim 8 scim_client = 'azure' 9 run_as_role = 'AAD_PROVISIONER' </pre>	
	2	Generate SCIM token	<pre> 1 select system\$generate_scim_access_ </pre>	
	3	Contact Azure team to complete SCIM integration.		Provide the generated SCIM Token.

				<div>  Save information in Secret Server path created step 1. </div>
	4	Validate the AD groups and users are provisioned in Snowflake		
Operation Role	1	Create role used by Snowflake Admin.	<pre> 1 use role ACCOUNTADMIN; 2 create role BUSSYS_OPERATION_ROLE; 3 grant role SECURITYADMIN to role BU 4 grant role SYSADMIN to role BUSSYS_ 5 grant role AAD_PROVISIONER to role 6 use role BUSSYS_OPERATION_ROLE; 7 show grants to role BUSSYS_OPERATIO </pre>	
	2	Create Operation warehouse	<pre> 1 use role BUSSYS_OPERATION_ROLE; 2 create warehouse if not exists BUSS </pre>	
	3	Grant operation to Snowflake Admin.	<pre> 1 use role BUSSYS_OPERATION_ROLE; 2 alter user "<email>" set default_wa </pre>	
Database, Schema and Stage	1	<p>Create SANDBOX database and UPLOAD schema.</p> <p>SANDBOX is the playground for developers with all permissions. UPLOAD schema hosts the tables used to upload semi-structure data for development. UPLOAD_SIGMA schema hosts the tables created by Sigma write back.</p> <div>  The schema UPLOAD_SIGMA is not visible in Sigma Computing by design. </div>	<pre> 1 use role BUSSYS_OPERATION_ROLE; 2 create database if not exists SANDB 3 create schema if not exists SANDBOX 4 create schema if not exists SANDBOX </pre>	
	2	Create STAGING in SANDBOX.UPLOAD to enable JSON file upload for migration purposes.	<pre> 1 use role BUSSYS_OPERATION_ROLE; 2 use database SANDBOX; 3 use schema UPLOAD; 4 create or replace stage JSON_STAGE 5 file_format = (type = 'JSON' compre </pre>	
	3	<p>Create RAW_<ENV> database and UPLOAD schema.</p> <p>naming convention for <ENV> is: PRD, UAT in</p>	<pre> 1 use role BUSSYS_OPERATION_ROLE; 2 create database if not exists RAW_P 3 create schema if not exists RAW_PRD </pre>	

		<p>PROD Account and HRD, QA, DEV in DEV Account.</p> <p>This database is used to replicate source data generated by Fivetran or custom Extract Load code.</p> <p>UPLOAD schema hosts the table used to upload semi-structure data, referenced in DBT models.</p>		
	4	<p>Create the <ENV>_DB used to store DBT models.</p> <p>naming convention for <ENV> is: PROD, UAT in PROD Account and HRD, QA, DEV in DEV Account.</p>	<pre>1 use role BUSSYS_OPERATION_ROLE; 2 create database if not exists <ENV></pre>	
Data Move ment Roles	1	[Optional] Go to Fivetran Create Destination and Connectors.		
	2	[Optional] Go to DBT, Set up Snowflake Connection		
	3	[Optional] Go to Sigma Create Connector.		
	4	[Optional] Go to Snowflake - PBI connectivity and User access management		

Data Roles	1	<p>Run <SOURCE>_<ENV>_ALL script for each Data Source environment</p> <div> <p>i Create the data role post data source replication.</p> </div> <p>This role grants permission to read <SOURCE>_<ENV> data, sensitive and non sensitive, current and future.</p>	<pre> 1 begin; 2 use role BUSSYS_OPERATION_ROLE; 3 create role if not exists <SOURCE>_<ENV>_ALL; 4 grant role <SOURCE>_<ENV>_ALL to <USER>; 5 grant usage on database RAW_<ENV> to <USER>; 6 grant usage on schema RAW_<ENV> to <USER>; 7 grant select on all tables in database RAW_<ENV> to <USER>; 8 grant select on all views in database RAW_<ENV> to <USER>; 9 grant select on future tables in database RAW_<ENV> to <USER>; 10 grant select on future views in database RAW_<ENV> to <USER>; 11 commit; 12 show grants to role <SOURCE>_<ENV>_ALL;</pre>	<p>Snowflake Admin can create data roles using the PowerShell script below:</p> <div> <p>⚠ Pre requisite:</p> <ul style="list-style-type: none"> user is granted usage to snowflake BUSSYS_OPERATION role. user is granted permission to clone the repo bussys_snow_operation. data_roles.json contains the data role information </div> <div> <p>Click here to expand...</p>  </div>
	2	<p>Run RAW_<ENV>_ALL script for each environment.</p> <p>This role grants permission to read all data in database RAW_<ENV>, current and future.</p>	<pre> 1 begin; 2 use role BUSSYS_OPERATION_ROLE; 3 create role if not exists RAW_<ENV>_ALL; 4 grant role <SOURCE>_<ENV>_ALL to <USER>; 5 commit; 6 show grants to role RAW_<ENV>_ALL;</pre>	
	3	<p>Run SANDBOX_ALL script .</p> <p>This role grants permission to read all data in database SANDBOX, current and future.</p>	<pre> 1 begin; 2 use role BUSSYS_OPERATION_ROLE; 3 create role if not exists SANDBOX_ALL; 4 grant usage, monitor on database SANDBOX to <USER>; 5 grant usage, monitor on all schemas in database SANDBOX to <USER>; 6 grant usage, monitor on future tables in database SANDBOX to <USER>; 7 grant select on all tables in database SANDBOX to <USER>; 8 grant select on all views in database SANDBOX to <USER>; 9 grant select on future tables in database SANDBOX to <USER>; 10 grant select on future views in database SANDBOX to <USER>; 11 commit; 12 show grants to role SANDBOX_ALL;</pre>	

Function Roles	1	<p>Run BUSSYS_DEVELOPER script</p> <p>This role grant permission to read all data in the account, create schema, tables and views in SANDBOX database, upload semi structure data in SANDBOX.UPLOAD, and upload semi structure data in RAW_<ENV>.UPLOAD.</p>	<pre> 1 begin; 2 create role if not exists BUSS 3 grant role BUSSYS_DEVELOPER to 4 create warehouse if not exists 5 grant usage, operate on wareho 6 grant role RAW_PRD_ALL, SANDBO 7 grant usage, monitor on databa 8 grant usage on schema SANDBOX. 9 grant create schema on databas 10 grant usage, monitor on databa 11 grant usage on schema RAW_PRD. 12 use database SANDBOX; 13 grant create table on schema U 14 use database RAW_PRD; -- adjus 15 grant create table on schema U 16 grant read, write on stage RAW 17 commit; 18 show grants to role BUSSYS_DEVELOP </pre>	
	2	<p>Run REVOPS ANALYST Role</p> <p>This role grant permissions to read SALESFORCE_ALL and NETSUITE_ALL data.</p>	<pre> 1 begin; 2 use role BUSSYS_OPERATION_ROLE; 3 set role_name = 'REVOPS_ANALYST' 4 set warehouse_name = 'REVOPS_ANA 5 create role if not exists identi 6 create warehouse if not exists i 7 grant usage, operate on warehous 8 grant role NETSUITE_PRD_ALL, SAL 9 commit; 10 show grants to role REVOPS_ANALYST </pre>	
	3	<p>Run BUSSYS_DQ_ADMIN script</p>	<pre> 1 begin; 2 use role BUSSYS_OPERATION_ROLE; 3 set role_name = 'BUSSYS_DQ_ADMI 4 create role if not exists ident 5 grant role identifier(\$role_nam </pre> <p>To Be Completed with DATA_QUALITY_ADMIN</p>	