Service-IQ Device Management Analytics

Installation Guide

Version: 3.1.0

Last Updated: 08-07-2022

Table of Contents

1. Introduction **1**

1.1 Understanding Service-IQ Device Management Analytics 1

2. System Requirements **4**

2.1 Cluster and Resources Requirements 4

2.2 Software Requirements 7

3. Pre- Installation Tasks **8**

3.1 Setup the Operating System 8

3.2 Set up a Service Account 8

3.3 Enable Network Time Protocol 8

3.4 Set Hostname and IP-to-Host Mapping 8

3.5 Verify Hostname Lookup 9

3.6 Disable Firewalld 9

3.7 Set Up SELINUX 9

3.8 Set Up Packages 9

3.9 Install Tableau 10

3.10 Before you Begin 10

3.11 Extracting the Provisioner 11

3.12 Set Up the Inventory File 12

3.13 Set Up vault.yml File 12

3.14 Set Up extra_vars.yml File 13

3.15 Set Up site.yml File 17

3.16 Synchronize LDAP Users and Groups (Optional) 17

3.17 Set Up dma.yml File 17

3.18 Set Up data-ingestion.yml File 21
3.19 Set Up Clickhouse Cluster Variables 25
4. Installation Procedure 27
4.1 Install Service-IQ DMA Components 28
4.2 Generate RSA Secrets 32
4.3 Sharing RSA Key 34
4.4 Install DMA Jobs 35
4.5 Install DMA UI 38
5. Uninstalling Service-IQ DMA Charts 39
6. Web UI Reference Table 40
7. Appendix A: Installing Tableau 41
7.1 Install Tableau Server 42
7.2 Install Tableau Desktop 59
7.3 Install ODBC Driver on Tableau Server 70
7.4 Updating Tableau Workbook and Tableau Data Source 74
7.5 Publish the Data Source Connection 78
7.6 Create Workbook using the Published Connection 80
7.7 Import Tableau Workbook 81
8. Appendix B: Kafka Topic 83
8.1 Create Kafka Topic for Kubernetes 83
8.2 Set Kafka SSL 84
8.3 View Kafka Topic for Kubernetes 84
8.4 Produce Kafka Topic for Kubernetes 85
8.5 Consume Kafka Topic for Kubernetes 85
9. Appendix C: Synchronize LDAP Users and Groups (Optional) 86

9.1 Set Up LDAP Users Sync 86

9.2 Set Up LDAP Group Sync 91

10. Appendix D: Creating Certificates 95

11. Appendix E: Encrypt and Edit vault.yml File 97

12. Appendix F: Parameters for CDR and DMC Data 98

12.1 Call Data Record (CDR) Data 98

12.2 Device Management Centre (DMC) Data 103

12.3 DMC Live Stream Data 106

12.4 Personally Identifiable Information Attributes Encryption 108

13. Appendix G: Data Ingestion Integration Requirements 110

14. Appendix H: SELINUX Guideline for Centos v7.x or RHEL v7.x 112

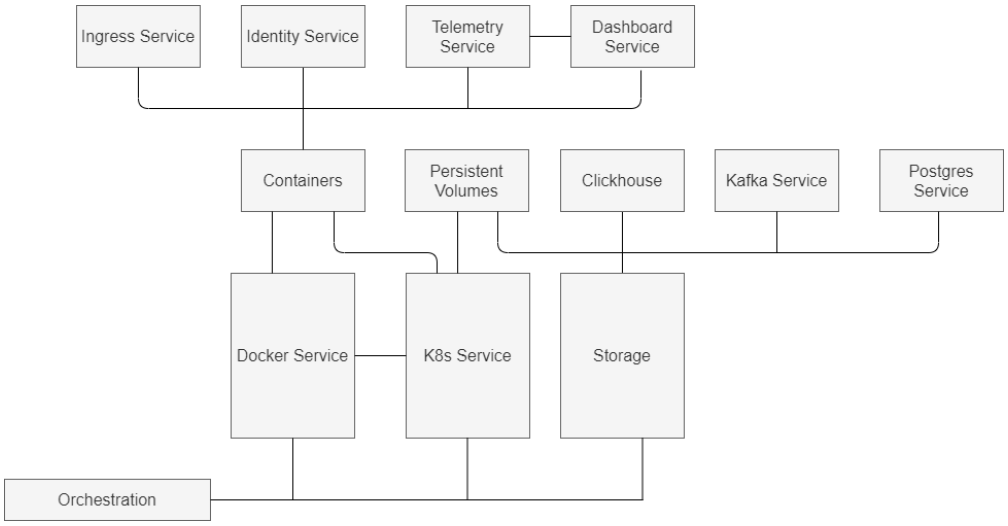
14.1 Set Up SELINUX 112

14.2 Set Up Packages 112

1. Introduction

This document describes how to install Service-IQ Device Management Analytics (DMA) on an on-premises cluster made of either bare metal nodes or virtual machines.

There are various components and services required to install Service-IQ DMA. The following image illustrates the Service-IQ DMA architecture.



1.1 Understanding Service-IQ Device Management Analytics

1.1.1 Data Ingestion Pipeline

The data ingestion pipeline is the entry point for the Service-IQ DMA data in the system. It ingests compressed data from a local or an SFTP server. In this release, Gzip compression is supported for compressed file ingestion, along with CSV format files. The following data formats are supported:

- Device Management Center (DMC)
- Call Detail Record (CDR)

The data ingestion pipeline performs the following logical functions:

1. Ingestion
2. Enrichment (CDR)

3. Kafka output

Data is ingested from the input location and parsed based on the value of *input_column_separator* variable in `data-ingestion.yml` file. The generated output is then published in a Kafka topic, which works as the input for the Service-IQ DMA jobs.

1.1.2 Service-IQ Device Management Analytics Jobs

The Service-IQ Device Management Analytics jobs runs at a configured interval to generate intermediary data to be used as the input for User Interface dashboards and reporting the services.

The Service-IQ DMA jobs contains the following processes:

- Device Change
- Device Library
- New User
- Reporting Services
- Whitelisting

The intermediary data is eventually used to generate output on the Service-IQ Device Management Analytics User Interface.

1.1.3 Service-IQ Device Management Analytics User Interface Module

This is the user interface that consumes device change output and provides the following analytics:

- **Device Adoption:** The Device Driver for this analytics are: Device OS, Category, Manufacturer or Name. Use this dashboard to visualize the percentage of users who changed and moved to another device driver and new users who showed up with device driver.
- **Device Loyalty:** The Device Driver for this analytics are: Device OS or Manufacturer. Use this dashboard to visualize the percentage of users who are

keeping the same device driver while changing the device.

- **Device Churn:** The Device Driver for this analytics are: Device OS, Category, Manufacturer or Name. Use this dashboard to visualize the percentage of users who are moving to another device driver.

For more information on the UI, refer to the *Service-IQ Device Management Analytics User Guide*.

Before you begin with the install procedure, users should be familiar with the following technologies and related terms:

- Basic Knowledge:
    - Kafka Cluster
    - Clickhouse
    - Docker, Containers and Kubernetes
    - Ansible
    - Helm Charts
    - Editing YAML (Yet Another Markup Language) Files
- Advanced Knowledge:
    - Linux OS administration and shell commands
    - IP Networking Configuration

2. System Requirements

This section provides a recommended list of hardware, network, and software requirements. The underlying Service-IQ DMA services must be installed for Service-IQ DMA to perform optimally. Contact the Technical Support team for more information.

Note: The installation procedure in the subsequent sections requires Internet connectivity to install software components. If your target cluster does not have Internet connectivity or is restricted by firewalls, you will have to fulfill the requirements for an offline install, such as creating a local registry and YUM mirrors.

2.1 Cluster and Resources Requirements

The following guidelines and recommendations are provided as a reference point to build the cluster.

Note: Ensure that you select the resource specifications based on the application workload as per your business requirement and these guidelines. Contact the Technical Support team for more information.

The following table lists the minimum node requirements to set up a high-availability (HA) cluster:

Note: Contact the Technical Support team on generating actual hardware requirements for installation based on your use case.

Node	Minimum Number
Master	03
Kafka	03
Compute	04
Load Balancer	02
Clickhouse	02

2.1.1 Master Nodes

The following table lists the minimum resource requirements for Master nodes:

Resource	Recommendation	Description
Nodes	3 Master Nodes for Kubernetes Control Plane	Master nodes run Kubernetes control plane services. Master nodes have different storage and memory requirements from Worker nodes.
Storage	The storage can be up to 1 TB of highly reliable RAID disks along with the unformatted disk space.	The Master nodes must have highly reliable storage for etcd database. It is recommended to provision the masters with RAID. **Note:** In a future release, options to provision such volumes on external storage will be available for more reliability.
CPU Core	12-24 CPU cores	The Master nodes significant communication between Name nodes and the clients, the Kubernetes Control Plane. This is the minimum requirement to provision for this volume of messaging traffic.
Memory	24-48GB RAM	This is the minimum requirement for memory. However, the amount of memory (RAM) depends on the number of objects in the Kubernetes space.
Network	1G Network Ports 10GB bandwidth	As all the communication flows through the Master nodes, this is the minimum requirement to ensure that there is no bottleneck in the network.

2.1.2 Worker Nodes

The following table lists the minimum resource requirements for Worker nodes:

Resource	Recommendation	Description
Memory	8G	The Worker nodes run the bulk of the workloads and also store data in clickhouse. This is the minimum requirement to run the services so that the overall workers do not slow down, but memory and processors will be based on the workload pattern.
Storage	Default clickhouse storage 2GB	Ingress nodes may be sized according to the workload expected in the data ingestion pipelines. Clickhouse Worker nodes need local storage to store data, having around 12GB to 16GB RAM per worker to benefit the overall performance. When co-locating multiple services, the overall performance will be much lower than dedicating nodes to the services.

2.1.3 Network

The network is a critical part of the cluster because it has the potential to slow down all operations. The main consideration is to avoid the possibility of congestion as far as possible. It is recommended to have a dedicated switch between the nodes that will purely handle this traffic. The network also provides a means to segregate the cluster into control plane and user plane traffic. The Control plane traffic, which includes both the Kubernetes control plane and the clickhouse data, should be provisioned with good bandwidth while the user plane that is only for accessing the UI and management can work well with lower bandwidth.

2.1.4 Load Balancer Nodes

A Load Balancer is provided through HAProxy to distribute external application requests across multiple servers. The nodes running the Load Balancer can be very low footprint. A 2 core and 4 GB RAM machine works well. The Load Balancer does not need local storage other than the OS.

2.1.5 Bastion, Ansible Controller Node (Optional)

A bastion and/or an Ansible controller node can be used to connect to the cluster to perform operations. This is useful in cases where the local network is private and there are no routable addresses in the cluster other than the Bastion and Load Balancer. Similar to the load Balancer nodes, this node has a low footprint in terms of CPU or Memory.

In case of unavailability of another node, one of the master nodes can also be used as ansible controller node.

2.2 Software Requirements

Service-IQ DMA is designed to run on a Linux Operating Systems. The following are the standard recommendations to utilize the deployment scripts that are included in the package:

- Software repositories to host RPMs, Docker and Helm Packages locally.
- RHEL v8 and apply YUM updates to the newest possible.
- Java - OpenJDK 1.8
- Python3

3. Pre- Installation Tasks

Before installing Service-IQ DMA application, perform the following tasks in the specified order to setup the environment.

3.1 Setup the Operating System

It is recommended that you start with at least RHEL v8 and update all your packages to the latest patches. Ensure that RHEL base and EPEL repository are available to all the nodes.

Run the following command to verify the version number of installed OS on your system:

```
cat /etc/redhat-release
```

The output of this command reflects the OS version number.

3.2 Set up a Service Account

To set up a service account for Service-IQ DMA installation, you must have a service account with 'sudo' privileges or you can use root user account.

3.3 Enable Network Time Protocol

It is important to synchronize all the nodes with Network Time Protocol (NTP). If the cluster does not have access to an internet time server, ensure that they are all synchronized with each other. Incorrect system clocks may lead to misinterpreted data during ingestion or enrichment.

3.4 Set Hostname and IP-to-Host Mapping

Set hostname with FQDN and add entry to `/etc/hosts` file for self IP to hostname mapping for all the nodes.

For example:

```
192.168.134.15  dma-mst-01.guavus.com dma-mst-01
```

3.5 Verify Hostname Lookup

Ensure that all the nodes can reach each other using Domain Name System (DNS). If there is no DNS, you can populate the hosts file to enable local hostname lookup for all the nodes and Virtual IP in the cluster.

3.6 Disable Firewalld

Disable firewalld during the installation to allow the services. You may enable firewalld later and allow the necessary services to communicate.

Run the following command to check the status of the firewalld service:

```
systemctl status firewalld.service
```

If the output of the preceding command is `Running` then run the following command to disable the service:

```
systemctl stop firewalld.service
```

3.7 Set Up SELINUX

Verify and ensure that the status of selinux is set to enforcing.

3.8 Set Up Packages

Ensure that:

- The `sshpass` package is installed on ansible-controller node.
- The `socat` package is installed on haproxy nodes.
- The following packages are installed on all the nodes:
    - python3
    - python3-pip
    - chrony
    - python3-libselinux (RHEL 8)
    - python3-libsemanage (RHEL 8)

- container-selinux
- selinux-policy
- libestr
- compat-openssl10
- libfastjson
- make
- rsyslog

Note: Refer to "Appendix H: SELINUX Guideline for Centos v7.x or RHEL v7.x" on page 112 to setup selinux and install packages.

Run the following command to verify if python3 is installed on all the nodes:

```
python3 -V
```

3.9 Install Tableau

Ensure that Tableau application is installed and integrated on your system. To determine the steps, refer to "Appendix A: Installing Tableau" on page 41

3.10 Before you Begin

Prior to installing the application, ensure that you have the following information readily available with you:

- The file name of SQLstream license must be known.
- The location for Service-IQ DMA IBs must be known.
- IP addresses of Load Balancer VIP and FQDN must be known.
- SFTP server must be available and details must be mapped in `extra_vars.yml` file.
- To configure LDAP, you must know the LDAP user and group information. For more information, refer to "Appendix C: Synchronize LDAP Users and Groups (Optional)" on page 86

3.11 Extracting the Provisioner

Perform the following steps to extract the provisioner:

1. Log into the ansible controller node as a SSH user.
2. Download `dma-v3.1.0-174.tar.gz`, `dma-backend-07-06-2022.tar.gz` and `dma-apps-v3.1.0.tar.gz` file from SFTP.

   Note: Contact Technical Support for the SFTP details.
3. Run the following command to extract the DMA ansible tarball:

```
sudo tar -zxvf dma-v3.1.0-174.tar.gz -C /data
```

4. Run the following command to provide ownership of `/data/dma` to sudo user:

```
sudo chown -R <sudo user name> /data/dma
```

5. Run the following command to make a copy of sample inventory file according to your name of the setup:

```
cp -r /data/dma/inventory/sample
/data/dma/inventory/<inventory site name>
```

6. Run the following commands to extract the artifacts:

```
sudo mkdir -p /data/guavus/software/data/dma/v3.1.0/
sudo tar -zxvf dma-backend-07-06-2022.tar.gz -C
/data/guavus/software/data/dma/v3.1.0/
sudo tar -zxvf dma-apps-v3.1.0.tar.gz -C
/data/guavus/software/data/dma/v3.1.0/
cd  /data/guavus
sudo ln -s software/data/dma/v3.1.0/charts charts
sudo ln -s software/data/dma/v3.1.0/containers containers
sudo ln -s software/data/dma/v3.1.0/repos repos
```

7. Run the following commands to execute Bootstrap installation:

```
cd /data/dma/
```

```
sudo sh /data/dma/scripts/bootstrap.sh

source /data/dma/.venv_ansible/bin/activate
```

3.12 Set Up the Inventory File

Edit the following files that are available at `/data/dma/inventory/<inventory site name>/hosts` location and add the hosts in ansible host groups to set up the inventory files.

- k8s_registry
- haproxy
- kafka
- zookeeper_nodes
- k8s_masters
- k8s_nodes
- k8s_infra
- clickhouse
- data_ingestion
- dma

3.13 Set Up vault.yml File

Update the configurations for the application in `vault.yml` file.

Run the following command to edit `vault.yml` file:

```
sudo vi /data/dma/inventory/<inventory-name>/group_
vars/all/vault.yml
```

Refer to the following table and update the variables as suggested:

Variable	Default	Description
site_default_password	''	Admin password for postgres & various User-Interfaces such as grafana, keycloak etc.
clickhouse_tableau_password	''	Clickhouse Tableau Password.

Note: It is mandatory to encrypt `vault.yml` file to proceed with installation process. For detailed steps, refer to "Appendix E: Encrypt and Edit vault.yml File" on page 97

3.14 Set Up extra_vars.yml File

Update the configurations for the application in `extra_vars.yml` file.

Run the following command to edit `extra_vars.yml` file:

```
sudo vi /data/dma/extra_vars.yml
```

Note: Update the variables as mentioned in the following sections. You can also update the values of other variables as per your requirement.

3.14.1 Define Site Variables

Variable	Default Value	Description
site_airgap_install	false	Change this value to `true` in case of airgap installation.
site_local_repo_dir	'/data/guavus'	Local directory path.
site_local_repo_ip_addr	''	Add this variable and set the value as IP address of ansible controller node, in case of airgap installation.
site_default_nw_interface	''	NW interface for platform components to run, For example `'eth0'`.

Variable	Default Value	Description
site_ext_nw_interface	''	NW interface for external services like UI, grafana etc to run, For example `'eth1'`. Required only in case of multi interface installations.
site_default_nw_gateway	''	Default gateway for `site_default_nw_interface`.
site_dns_nameservers	[]	List of dns nameservers for cluster. Its an optional item. In case of no dns servers available please make sure hosts file is populated on all nodes to enable local hostname lookup.
site_ntp_servers	[]	List of ntp servers for cluster. Its an optional item. In case of no ntp servers available please make sure to have ntpd service running on all nodes.
site_ext_lb_vip_ipaddr	''	HAProxy external VIP. VIP which is reachable from outside the cluster. Only required in case of multi interface installations.
site_ext_lb_vip_fqdn	''	HAProxy external VIP FQDN. FDQN for `site_ext_lb_vip_ipaddr`. Only required in case of multi interface installations.

3.14.2 Define Clickhouse Tableau Variables

Variable	Default Value	Description
clickhouse_tableau_username	"tableau"	This is the Tableau username for connecting to Clickhouse.
dma_ui_data_exploration_url	""	Data exploration for Tableau URL.

3.14.3 Define Kubernetes Variables

Variable	Default Value	Description
kubernetes_dns_domain	'kubernetes.local'	Kubernetes cluster dns domain.

3.14.4 Define Ingress Certificates

Variable	Default Value	Description
kubernetes_ nginx_ ingress_cert	file: '' key: ''	Path of SSL certificate and key for Ingress UIs. When creating certificate use site_lb_vip_fqdn as CN. If not provided ingress still works with HTTPS and a self signed certificate is generated for `ingress.local`. Refer appendix for the steps to create a self signed certificate and key.

3.14.5 Define Kafka Variables

Variable	Default Value	Description
kafka_operator_ kafka_stg_size	50Gi	Storage size for kafka.
kafka_operator_zoo-keeper_stg_size	5Gi	Storage size for kafka zookeeper metadata.
kafka_operator_ kafka_heap_size	6G	Kafka heap size.
kafka_data_dirs	- /data/kafka01/kafka-logs	List of kafka data directories. Path should be available and mounted.

3.14.6 Define Zookeeper Variables

Variable	Default Value	Description
zookeeper_data_dir	'/data/zookeeper'	This is the path for zookeeper data directory on kafka nodes. The paths must be available and mounted.
zookeeper_data_dir_size	10Gi	Zookeeper data directory Size in GBs.

3.14.7 Define Data Ingestion Variables

Variable	Default Value	Description
dmc_input_file_location	""	Provides the SFTP path of DMC data to be used by DI.
cdr_input_file_location	""	Provides the SFTP path of CDR data to be used by DI.

3.14.8 Define Keycloak Variables

Variable	Default Value	Description

keycloak_redirection_links	"[]"	You need to update this variable as per your requirement.
		For example:
		`keycloak_redirection_links:`
		`  - http://<tableau-url>/wg/saml/SSO/index.html`
		`  - http://<tableau-url>/wg/saml/SingleLogout/index.html`

3.15 Set Up site.yml File

Update the configurations for the application in `site.yml` file.

Run the following command to edit `site.yml` file:

```
sudo vi /data/dma/inventory/<inventory>/group_vars/all/site.yml
```

3.15.1 Define HAProxy Variables

Variable	Default Value	Description
site_lb_vip_ipaddr	''	HAProxy node IP (in case of single HAProxy node) or VIP (in case of multiple HAProxy nodes).
site_lb_vip_fqdn	''	HAProxy node FQDN (in case of single HAProxy node) or VIP FQDN (in case of multiple HAProxy nodes).
site_lb_kuber-netes_bind_port	"8443"	Default kubernetes bind port.
site_cluster_name	""	Default site cluster name.
site_local_repo_name	""	Local artifactory repo name.

3.16 Synchronize LDAP Users and Groups (Optional)

For detailed steps, refer to "Appendix C: Synchronize LDAP Users and Groups (Optional)" on page 86

3.17 Set Up dma.yml File

Update the configurations for the application in `dma.yml` file.

Run the following command to edit `dma.yml` file:

```
sudo vi /data/dma/inventory/<inventory_name>/group_vars/all/dma.yml
```

Note: Update the variables as mentioned in the following sections. You can also update the values of other variables as per your requirement.

3.17.1 Define DMA Variables

Variable	Default Value	Description
dma_imei_truncation_enabled	false	This variable specifies whether you need to truncate the IMEI column to 14 characters or not. Possible values are `true` and `false`.
dma_sv_extraction_enabled	false	This variable specifies whether we need to extract SV values from IMEI or not. Possible values are `true` and `false`.
dma_clickhouse_aggregated_ttl_interval	"1 DAY"	This variable specifies aggregated TTL interval for clickhouse. Example values are "1 DAY","1 MONTH","1 YEAR". **NOTE:** Recommended value for this variable is "180 DAY".
dma_hourAggregationLevel	true	This variable enables hourly aggregation. Possible values are `true` and `false`.

3.17.2 Define dma_deviceLibrary Variables

Variable	Default Value	Description
inputFileLocation	""	This variable specifies SFTP read location of device EaaS file.

Variable	Default Value	Description
filePattern	"Handsets_data_(\\d{8})\\.zip"	This variable specifies the file pattern in which the EaaS file would be expected at SFTP location.
sqlDumpPattern	"encrypted_wurfl-(\\d{8})-(\\d{6})\\.sql\\.tar\\.gz"	This variable specifies the dump pattern of EaaS file after extracting it from the ZIP format.

3.17.3 Define Whitelist Configurations

Variable	Default Value	Description
dma_whitelisting_enabled	false	This variable specifies that the Whitelisting job for DMA is enabled or not. Possible values are `true` and `false`.

3.17.4 Define dma_whitelisting Variables

Variable	Default Value	Description
ibType	"dynamic"	This variable specifies the IB type required for whitelisting. Possible values are `"static"` and `"dynamic"`.
ibPath	""	This variable specifies the SFTP path of IB in case of IB type:`"static"`
ttl	30	This variable specifies the TTL for whitelisting job.

Variable	Default Value	Description
threshold	5	This variable specifies threshold for whitelisting job.
inputFileType	"csv"	This variable specifies the INPUT FILE TYPE of IB. Possible values are `"csv"` and `"gzip"`.
whiteListField	"msisdn"	This variable specifies the field on the basis of which whitelisting is to be done. Possible values are `"msisdn"`,`"imei"` and `"imsi"`.
columnSeparator	","	This variable separates columns in IB file on the basis of given column separator.
whiteListColumnPosition	1	This variable specifies the position of WHITE LIST column. **NOTE:** This variable would only be applicable when ibType is set to dynamic.
lastVisibleTimeColumnPosition	2	This variable specifies the position of LAST VISIBLE TIME COLUMN. **NOTE:** This variable would only be applicable when ibType is set to dynamic.

Variable	Default Value	Description
chunkSize	50000	This variable specifies the CHUNK SIZE to be inserted via WHITE LISTING Job.

3.17.5 Define Parameters in dma_notification Variable

Variable	Default Value	Description
smtpHost	"192.168.104.25"	SMTP relay server host location.
smtpPort	"25"	SMTP relay server port.
sender	"dma-alerts@guavus.-com"	Default sender for the email notification of the generated reports.
recipient	""	Default recipient for the email notification of the generated reports.
cc	""	Default CC for the email notification of the generated reports.
bcc	""	Default BCC for the email notification of the generated reports.
location	""	Bucket name or directory to which the SFTP or local generated reports are added.
body	""	Body of the email notification for the generated report. If body configuration is given in the report JSON, then that will be picked.

3.18 Set Up data-ingestion.yml File

Update the configurations for the application in `data-ingestion.yml` file.

Run the following command to edit `data-ingestion.yml` file:

```
sudo vi /data/dma/inventory/<inventory_name>/group_vars/all/data-ingestion.yml
```

Note: You can also update the values of other variables as per your requirement.

3.18.1 Define Data Ingestion Variables

Variable	Default Value	Description
di_input_feed	"dmc"	This variable specifies the pipeline you need to run for DI-DMA. Possible values are `"dmc"` and `"cdr"`.
di_deploy-ment_type	"onPrem"	This variable specifies the deployment type of DMA application. By Default, it is set to onPrem and in case of cloud deployment it is set to cloud.

3.18.2 Define Parameters in input_feed_dmc Variables

Variable	Default Value	Description
file_pattern	.*(\\d{14}).*\\.csv	The Default Value is pattern of files to be read from input_file_location. Example for Default Value: `RMIDM_flow_REPORTOCS_02162021132909_006_000019877_009.csv`
column_ mapping	'msisdn,imei,imsi,last_ configuration,msc_gt,ic-cid,last_detection'	It specifies how to store each field from the record. The source column name is replaced by the configured destination column name enabling you to add value mappings to map the source values to target values for the column you require to map.

Variable	Default Value	Description
skip_header	false	Set this value to true if header line is present in input files.
file_type	"none"	Type of the files present on `input_file_location`. Set this value to `"gzip"` or `"none"` as required.
input_column_separator	";"	Separates values in the file based on the delimiter you provide in the file.
last_visible_time_in_record	false	Set this value to `true` to read last visible time from input file.
last_visible_time_column	"last_detection"	This variable specifies the data column on the basis of which last visible time would be fetched from the file if `last_visible_time_in_record` is set to `true`.
sort_field	"MODIFIED_FILE_TIME"	Sort the files on the basis of the file's modified time. If two files have the same modified time the files will be sorted lexicographically.
kafka_topic_partitions	1	This variable specifies the number of partition required for Kafka topic. **Note:** Ensure that number of kafka partitions are equal to number of clickhouse pods in case of containerized clickhouse.

Variable	Default Value	Description
kafka_ topic_con- fig_rep- lication	1	This variable specifies replication factor of the topic. In case of achieving HA for kafka, ensure that the value of `kafka_topic_config_ replication` is set to 3.

3.18.3 Define Parameters in input_feed_cdr Variables

Variable	Default Value	Description
file_pattern	.*(\\d{14}).*\\.csv	The Default Value is pat- tern of files to be read from input_file_location. Example for Default Value: `RMIDM_flow_REPORTOCS_ 02162021132909_006_ 000019877_009.csv`
skip_header	false	Set this value to `true` if header line is present in input files.
column_ mapping	',imsi,,,sgsn_ip,,,,,last_visible_time,- duration,,,,,,msisdn,,,imei,rat_ type,lac,ci,dataVolumeFBC'	It specifies how to store each field from the record. The source column name is replaced by the configured destination column name enabling you to add value mappings to map the source values to target val- ues for the column you require to map.

Variable	Default Value	Description
file_type	"none"	Type of the files present on `input_file_location`. change it to `"gzip"` or `"none"` as per your requirement.
kafka_ topic_partitions	1	This variable specifies the number of partition required for Kafka topic. **Note:** Ensure that number of kafka partitions are equal to number of clickhouse pods in case of containerized clickhouse.
kafka_ topic_config_replication	1	This variable specifies replication factor of the topic. In case of achieving HA for kafka, ensure that the value of `kafka_topic_config_replication` is set to 3.

3.19 Set Up Clickhouse Cluster Variables

Update the configurations for the application in `kubernetes.yml` file.

Run the following command to edit `kubernetes.yml` file:

```
sudo vi /data/dma/inventory/<inventory_name>/group_
vars/all/clickhouse/kubernetes.yml
```

3.19.1 Define Parameters in clickhouse_cluster Variable

Variable	Default Value	Description
clickhouse_clusters_shardsCount	2	Number of shard should be equal to number of clickhouse hosts.
clickhouse_clusters_replicasCount	2	Number of replica should be equal to number of clickhouse hosts.

3.19.2 Define Clickhouse Resources Requests and Limits (Optional)

Variable	Default Value	Description
clickhouse_resources	{}	

3.19.3 Define Clickhouse Storage Size

Variable	Default Value	Description
clickhouse_data_storage_size	50Gi	Size of clickhouse data storage.
clickhouse_zk_storage_size	2Gi	Size of clickhouse zk storage.

4. Installation Procedure

Refer to this section to install the Service-IQ DMA components in the prescribed order.

1. "Install Service-IQ DMA Components" on the facing page
2. "Generate RSA Secrets" on page 32
3. "Sharing RSA Key" on page 34
4. "Install DMA Jobs" on page 35
5. "Install DMA UI " on page 38

4.1 Install Service-IQ DMA Components

Log into the ansible controller node and run the following ansible playbooks to install the Service-IQ DMA components. Run playbooks in the following order. Order of execution is very important for core-services installation.

Note: Ensure that all the playbooks have run successfully and there are no failed tasks in the final summary report in the end of each run.

4.1.1 Install Bootstrap

Run the following ansible script:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/bootstrap/main.yml --extra-vars  "@extra\
_vars.yml" --ask-pass --ask-become-pass --ask-vault-pass\
 --user <sudo user name> --become --become-method=sudo
```

4.1.2 Install HAProxy

Run the following ansible script:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/haproxy/main.yml --extra-vars  "@extra\
_vars.yml" --ask-pass --ask-become-pass --ask-vault-pass\
 --user <sudo user name> --become --become-method=sudo
```

4.1.3 Install Kubernetes

Run the following ansible script:

```
cd /data/dma
```

```
source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/kubernetes/main.yml --extra-vars  "@extra\
_vars.yml" --ask-pass --ask-become-pass --ask-vault-pass\
 --user <sudo user name> --become --become-method=sudo
```

4.1.4 Install Kubernetes Infrastructure Applications

Run the following ansible script:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/kubernetes/infra-apps.yml --extra-vars  \
"@extra_vars.yml" --ask-pass --ask-become-pass --ask\
-vault-pass --user <sudo user name> --become --become-\
method=sudo
```

4.1.5 Install Guavus-IAM

Run the following ansible script:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/guavus-iam/main.yml --extra-vars  "@extra\
_vars.yml" --ask-pass --ask-become-pass --ask-vault-\
pass --user <sudo user name> --become --become-method=sudo
```

4.1.6 Install Kafka

Run the following ansible script:

```
cd /data/dma
```

```
source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/kafka/main.yml --extra-vars  "@extra_vars\
.yml" --ask-pass --ask-become-pass --ask-vault-pass -\
-user <sudo user name> --become --become-method=sudo
```

4.1.7 Install Clickhouse

Run the following ansible script:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/clickhouse/main.yml --extra-vars  "@extra\
_vars.yml" --ask-pass --ask-become-pass --ask-vault-\
pass --user <sudo user name> --become --become-method=sudo
```

4.1.8 Install Prerequisites for DMA

Run the following ansible script:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/dma/main.yml --extra-vars  "@extra_vars.yml" \
--ask-pass --ask-become-pass --ask-vault-pass --\
user <sudo user name> --become --become-method=sudo
```

4.1.9 Install Prerequisites DI

Note: Before installing the prerequisites for DI, ensure that SQLstream license file is placed in `/data/guavus/license` directory.

Run the following ansible script:

```
cd /data/dma
```

```
source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/data-ingestion/main.yml --extra-vars  \
"@extra_vars.yml" --ask-pass --ask-become-pass --\
ask-vault-pass --user <sudo user name> --become --\
become-method=sudo
```

4.2 Generate RSA Secrets

Perform the following steps to generate RSA secrets:

1. Run the following command to remove `/tmp/dma` directory and the existence files:

   ```
   sudo rm -rf /tmp/dma
   ```

2. Run the following command to create a `/tmp/dma` directory:

   ```
   mkdir /tmp/dma
   ```

3. Run the following command to generate a RSA key:

   ```
   ssh-keygen -m PEM -t rsa
   ```

   The preceding command generates the following content. Enter `/tmp/dma/id_rsa` file to save the RSA key.

   ```
   Generating public/private rsa key pair.
   Enter file in which to save the key (/home/guavus/.ssh\
   /id_rsa):/tmp/dma/id_rsa
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   ```

4. Run the following command to remove `/tmp/di` directory and the existence files:

   ```
   sudo rm -rf /tmp/di
   ```

5. Run the following command to create a `/tmp/di` directory:

   ```
   mkdir /tmp/di
   ```

6. Run the following command to generate a RSA key:

   ```
   ssh-keygen -m PEM -t rsa
   ```

   The preceding command generates the following content. Enter `/tmp/di/id_rsa` file to save the RSA key.

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/guavus/.ssh\
/id_rsa):/tmp/di/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

7. Run the following commands to create the new RSA key secrets:

```
kubectl delete secrets ssh-rsa-key-secret -n dma

kubectl delete secrets dma-ssh-rsa-key-secret -n dma

kubectl create secret generic ssh-rsa-key-secret \
-n dma --from-file=id_rsa=/tmp/di/id_rsa

kubectl create secret generic dma-ssh-rsa-key-secret \
-n dma --from-file=id_rsa=/tmp/dma/id_rsa
```

4.3 Sharing RSA Key

This section provides the ansible scripts to share the RSA Keys for DMA and Data Ingestion:

- "RSA Key for DMA" below
- "RSA Key for Data Ingestion" below

4.3.1 RSA Key for DMA

Run the following ansible script for keyless access of DMA with SFTP server:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible all -i <sftp_server_ip>, -m authorized_key \
-a "user=<sftp_user> state=present key={{ look\
up('file', '/tmp/dma/id_rsa.pub') }}" --ask-pass \
--ask-become-pass --user <sudo_user> --become \
--become-method=sudo
```

4.3.2 RSA Key for Data Ingestion

Run the following ansible script for keyless access of Data Ingestion (DI) with SFTP server:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible all -i <sftp_server_ip>, -m authorized_key \
-a "user=<sftp_user> state=present key={{ look\
up('file', '/tmp/di/id_rsa.pub') }}" --ask-pass \
--ask-become-pass --user <sudo_user> --become \
--become-method=sudo
```

4.4 Install DMA Jobs

Run the following ansible script to install DMA Job:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/dma/dma-jobs-app.yml --extra-vars  "@extra\
_vars.yml" --ask-pass --ask-become-pass --ask-vault-\
pass --user <sudo user name> --become --become-method=sudo
```

Note: Wait until Device Library Job is executed and the data is loaded into device table successfully. Ensure that the device table in Clickhouse have records and the count of these records are stable.

Once you have installed DMA Jobs, execute the following ansible scripts for `dma_data_processing_mode`:

- Batch

  ```
  cd /data/dma

  source /data/dma/.venv_ansible/bin/activate

  ansible-playbook -i inventory/<inventory_name> \
  playbooks/data-ingestion/di-7s-app.yml --extra-\
  vars  "@extra_vars.yml" --ask-pass --ask-become-\
  pass --ask-vault-pass --user <sudo user name> --\
  become --become-method=sudo
  ```

- Streaming

  1. Run the following command on kubernetes node to create kafka topics for DMC (dmc_output) and CDR (cdr_output) data:

     Note: Ensure that the count of the partitions is equal to number of nodes in the clickhouse.

```
cat <<EOF | kubectl apply -f -
---
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaTopic
metadata:
  name: dmc
  namespace: kafka
  labels:
    strimzi.io/cluster: kafka
spec:
  topicName: dmc_output
  partitions: 1
  replicas: 1
  config:
    retention.ms: 259200000
---
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaTopic
metadata:
  name: cdr
  namespace: kafka
  labels:
    strimzi.io/cluster: kafka
spec:
  topicName: cdr_output
  partitions: 1
  replicas: 1
  config:
    retention.ms: 259200000
EOF
```

2. Run the following command to view the kafka topic:

```
kubectl -n kafka get kt
```

Refer to "Appendix B: Kafka Topic" on page 83 for more information on kafka topic.

4.5 Install DMA UI

Run the following ansible script to install DMA UI:

```
cd /data/dma

source /data/dma/.venv_ansible/bin/activate

ansible-playbook -i inventory/<inventory_name> \
playbooks/dma/dma-apps.yml --extra-vars  "@extra\
_vars.yml" --ask-pass --ask-become-pass --ask-vault-\
pass --user <sudo user name> --become --become-method=sudo
```

5. Uninstalling Service-IQ DMA Charts

Perform the following steps to delete the helm charts from your system:

1. Run the following command on master node to get the list of helm charts in the namespace `<dma_namespace>`:

   ```
   sudo helm list -n <dma_namespace>
   ```

   For example if the value of `<dma_namespace>` is `dma`, then the command is as follows:

   ```
   sudo helm list -n dma
   ```

2. Run the following command on master node to uninstall the charts such as DMA jobs, DMA-UI and so on as per your requirement:

   ```
   sudo helm uninstall <CHART_RELEASE_NAME> -n <dma_namespace>
   ```

   For example,

   ```
   sudo helm uninstall dma-jobs -n dma
   sudo helm uninstall dma-ui -n dma
   sudo helm uninstall dmc-0 -n dma
   sudo helm uninstall cdr-0 -n dma
   ```

6. Web UI Reference Table

The following table lists the web service, URLs, and credentials to log into the services.

Web Service	URL
Grafana	https://<site_lb_vip_fqdn>/grafana
Guavus - IAM Admin Console	https://<site_lb_vip_fqdn>/auth/admin
DMA UI	https://<site_lb_vip_fqdn>/dma

Note: Contact your system administrator for the credentials.

7. Appendix A: Installing Tableau

This section describes the tasks and steps to install Tableau application on your system.

1. "Install Tableau Server" on the facing page
2. "Install Tableau Desktop " on page 59
3. "Install ODBC Driver on Tableau Server" on page 70
4. "Updating Tableau Workbook and Tableau Data Source" on page 74
5. "Publish the Data Source Connection" on page 78
6. "Create Workbook using the Published Connection" on page 80
7. "Import Tableau Workbook" on page 81

7.1 Install Tableau Server

Perform the following steps to install Tableau Server on a bare metal machine:

7.1.1 Prerequisites

Ensure that you meet the minimum hardware requirement prior to installing the Tableau Server on your system, refer to Minimum Tableau Hardware Requirement

7.1.2 Before You Begin

Perform the following steps:

1. Run the following command to stop firewall to access Tableau from a web browser:

```
sudo systemctl stop firewalld
```

2. Run the following command to install OpenSSL:

```
sudo yum install openssl
```

3. Run the following command to install wget:

```
sudo yum install wget
```

7.1.3 Adding Tableau User

1. Run the following commands as a root user to add a RHEL user:

```
sudo useradd dma
sudo passwd dma
```

2. Run the following command as a root user to add the previously created RHEL user to wheel group:

```
sudo gpasswd -a dma wheel
```

7.1.4 Installation Steps

Perform the following steps to install Tableau Server with Active Directory (AD) as the same user as created in the previous section:

1. Download `tableau-server-<TABLEAU_VERSION>.rpm` RPM from SFTP server.
2. Run the following command to install the RPM:

   ```
   sudo yum install tableau-server-<TABLEAU_VERSION>.rpm
   ```

3. Run the following command to move to `scripts.<SCRIPT_VERSION>/` directory:

   ```
   cd /opt/tableau/tableau_\
   server/packages/scripts.<SCRIPT_VERSION>/
   ```

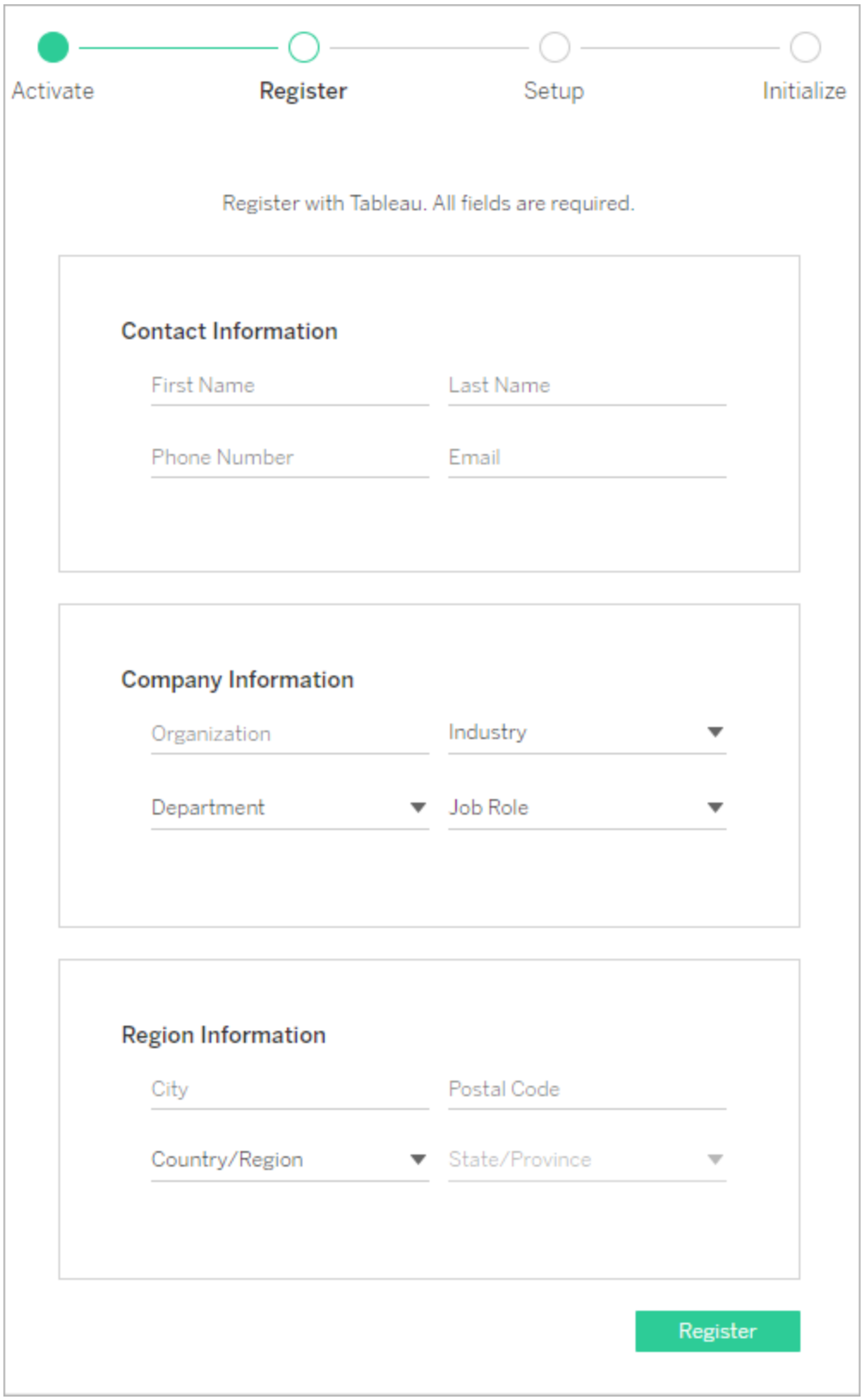
4. Execute the following ansible script to start Tableau Service Manager (TSM):

   ```
   sudo ./initialize-tsm --accepteula
   ```

   One the script runs successfully, close the command prompt.

5. Type the *URL - https://<machine_ip>:8850/#/* to log into TSM. These credentials are same as used for installing Tableau.
6. On the landing page, either select *Start Tableau Server Trial* option available in the bottom of the screen or enter your license key.

7. On the **Register** page, fill in your required details.



8. On **Setup** page, select either of the **Identity Store**. If you select *Local* then you are not required to fill in any details and if you select *Active Directory* then enter the details for the following fields as required:

Field	Sample Value
Domain	in.abc.com
Hostname	ui.in.abc.com
NetBIOS	ABCad
Port	389
Username	abc
Password	***
Gateway Port	80
Product Usage Data	Disabled
Include Samples	Enabled

   Once you have entered all the details, click **Initialize** button.

   This may take several minutes to complete.

9. On the **Initialize** page, *Initialization Complete* will be displayed. Click **Continue** button.

   Tableau Server is setup on your system successfully.

10. Run the following command to add an Admin user for Tableau:

    Ensure that no httpd services are running on the node where you intend to install Tableau Server.

```
tabcmd initialuser --server 'localhost:80' \
--username 'dma-admin' --password '<password>'
```

Note: You need to create your credentials if you have selected Local Identity Store (LIS). If you have selected Active Directory, then you must use the existing AD users' credentials.

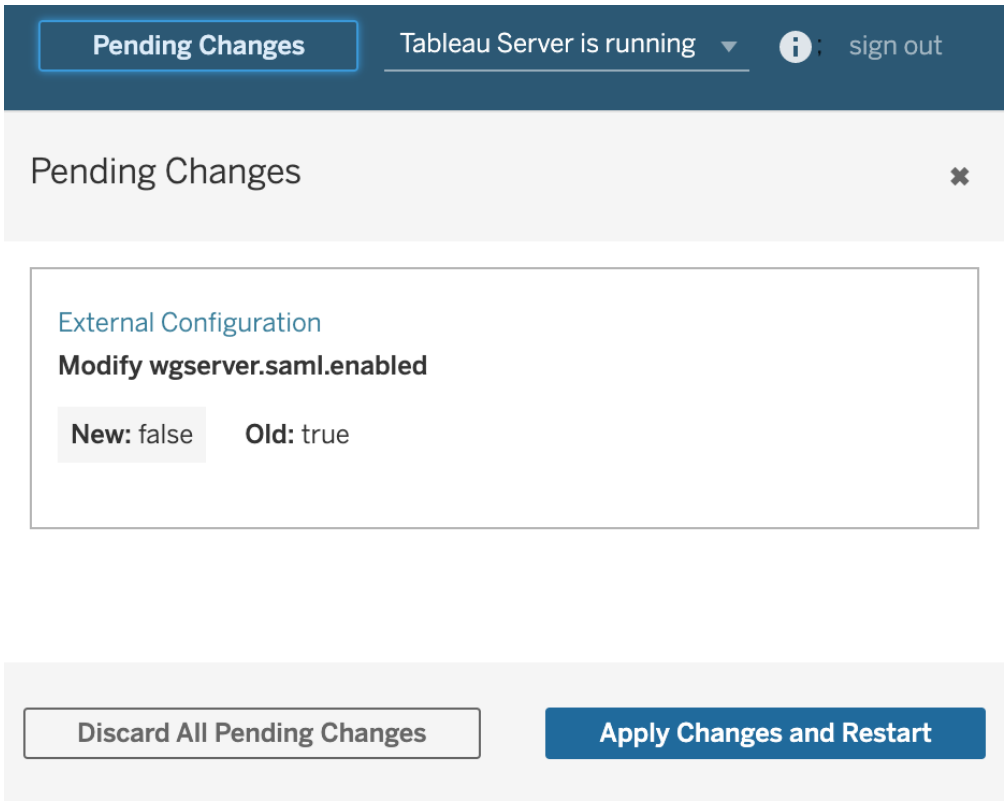
11. Type the *URL - http://<machine_ip>* in any web browser.
12. Log into the Tableau Server using your LIS or AD credentials, as required.

7.1.5 SSL Configuration for Tableau Server

Refer to this section to determine the steps for configuring SSL:

1. Enter the *URL: http://<tableau-server>:<port>* on your web browser.
2. Log into Tableau Service Manager with your valid credentials and navigate to **Configuration** > **Security** > **External SSL**.
3. Select the files generated in Start SAML Setup in TSM for the following fields respectively:
    - SSL certificate file
    - SSL certificate key file
    - SSL certificate chain file
4. Click **Save Pending Changes** button.
5. From the navigation bar, **Pending Changes** option and **Apply Changes**

and Restart to reflect all the changes.



7.1.6 Start SAML Setup in Tableau Service Manager

Perform the following two steps in the same sequence as mentioned:

1. "Start SAML Setup in TSM" below
2. "Creating a New Service Provider in KeyCloak" on page 51

Start SAML Setup in TSM

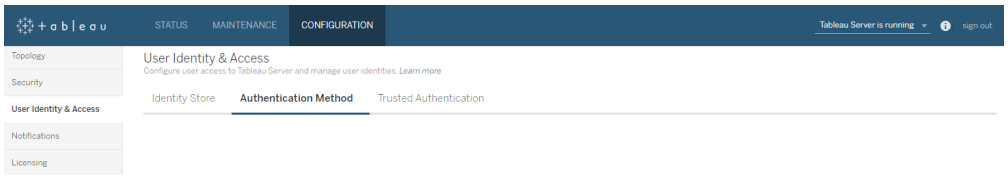
Perform the following steps on the Tableau Server node:

1. Run the following commands to generate *.crt* and *.key* files:

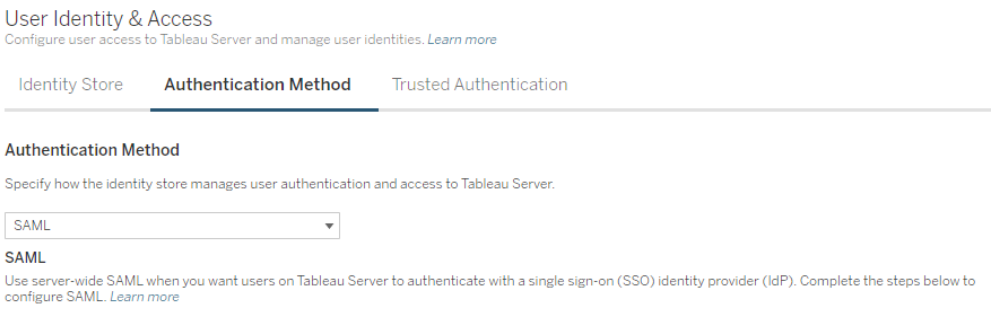
```
sudo yum install wget

openssl genrsa -des3 -passout pass:password -out \
server.pass.key 2048
```

```
openssl rsa -passin pass:password -in server.pass.\
key -out server.key

openssl req -new -key server.key -out server.csr

openssl x509 -req -sha256 -days 3650 -in server.\
csr -signkey server.key -out server.crt
```

2. Enter the *URL: http://<tableau-server>:<port>* on your web browser.
3. Log into Tableau Service Manager with your valid credentials and navigate to **Configuration** > **Authentication Method**.



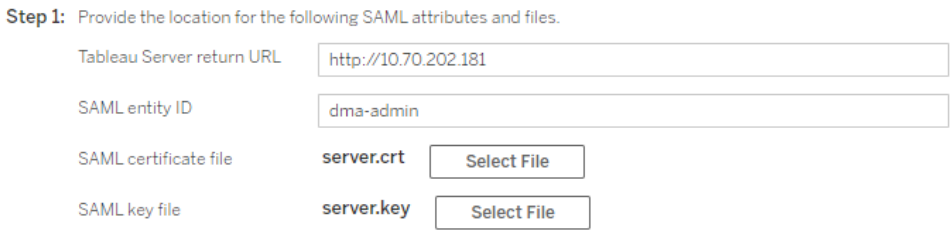
1. Select **SAML** option from the drop down.



2. Keep *Enable SAML authentication for the server* **unchecked**.
3. **Step 1** - Refer to the following table to determine the values for each field available on the UI:

Field Name	Sample Value	Description
Tableau Server return URL	http://<IP>:<port>	Enter the URL of Tableau Server Return. Ensure to not use forward slash at the end of the URL. Port is optional.
SAML entity ID	dmatableau	Enter the Entity ID of SAML. This same name is used in Keycloakclient configuration. Ensure that the name has no space.
SAML certificate file		Select the file for SAML certificate file (*server.crt* previously generated)
SAML file key		Select the file for SAML key file (*server.key* previously generated)

The following image illustrates the sample values:



4. **Step 2** - Click the **Download XML Metadata File** button to download the file and register it with your IdP.

Note: Check the content of the xml file as it must contain the certificate or key information. If it does not contain the information, then your certificate or key is incorrect. Contact your Technical Customer Support for further details.

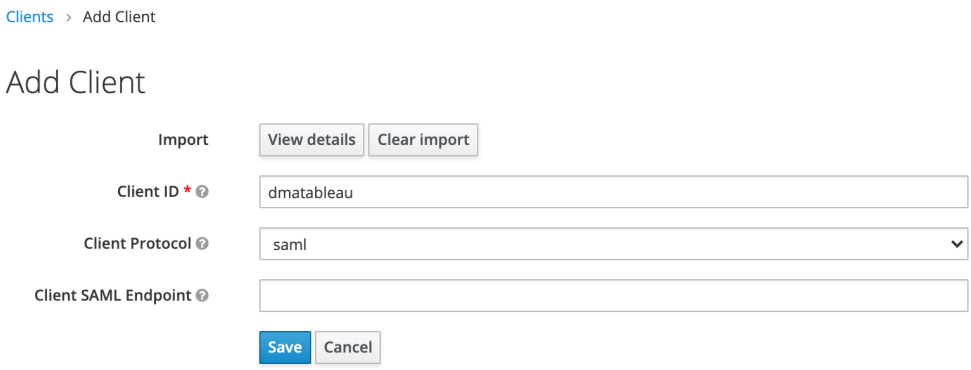
Creating a New Service Provider in KeyCloak

Perform the following steps to create a new client in keycloak:

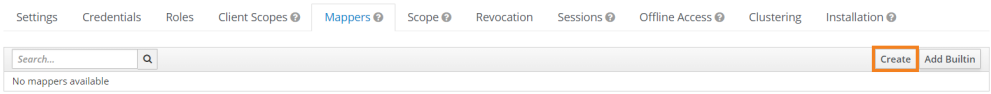
1. Enter the *URL: https://<LB-VIP>/auth/admin/master/console/#/realms/dma* on your web browser.
2. Log into Keycloak console with your valid credentials and navigate to **Clients** > **Create** button available on the right side of the screen.
3. On **Add Client** page, add the following values for the respective fields:

Field Name	Sample Value	Description
Import		Browse and add the metadata file downloaded previously in TSM.
Client ID	dmatableau	Enter your ID.
Client Protocol	saml	Select your protocol as **saml**.

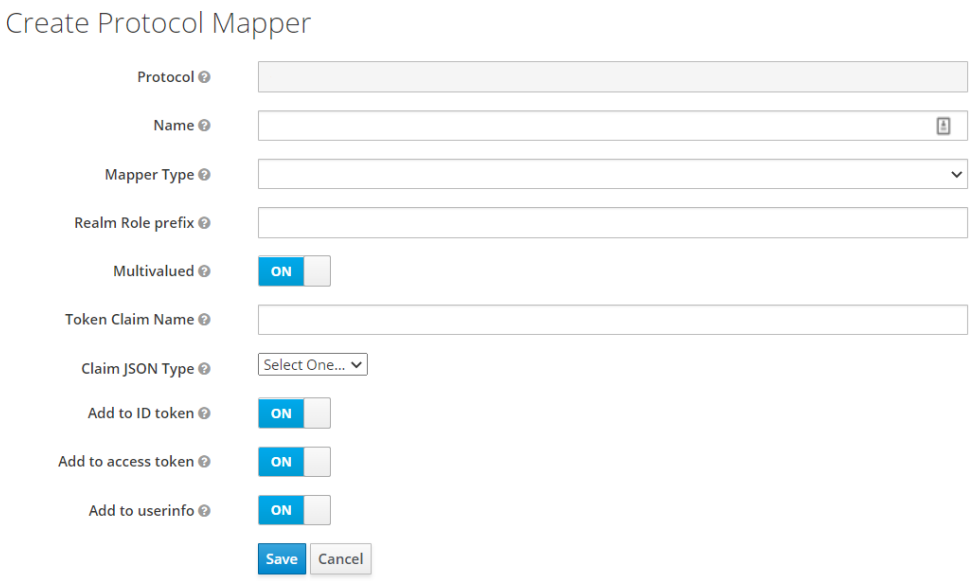
The following image illustrates the UI screen:



4. Once you have added values, click the **Save** button to keep the changes.
5. On the **Settings** tab, keep the values as is and move to **Mappers** tab.
6. On the Mappers tab, click the **Create** button to create a new protocol mapper. Mappers are mandatory to shape the SAML message from KeyCloak to Tableau Server.



7. Enter the required values for each field on the Create Protocol Mappers page.

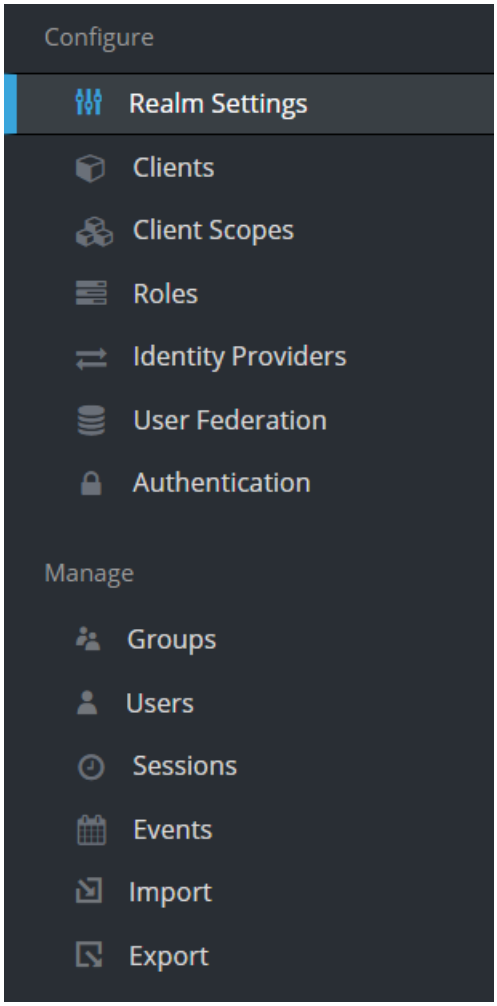


Refer to the following table to determine the values for each field available on the UI:

Protocol	Name	Mapper Type	Property	Friendly Name	SAML Attribute Name	SAML Attribute NameFormat
saml	lastname	User Property	lastName	lastname	lastname	Basic
saml	firstname	User Property	firstname	firstname	firstname	Basic
saml	email	User Property	email	email	email	Basic
saml	username	User Property	username	username	username	Basic

8. Once you have added values, click the **Save** button to keep the changes.
9. You cannot add all the values at once, therefore you have to re-create mappers several times to add the values for different fields.

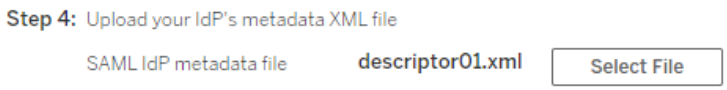
   Click the **Save** button to keep the changes.
10. Select and navigate to **Realm Settings** Option available on the left side of the screen.



11. In the **Endpoints** field, select **SAML 2.0 Identity Provider Metadata** and save this file on your system.

12. Go to TSM configuration page and check the *Enable SAML authentication for the server* option.

13. **Step 4** - Click **Select File** button and upload the IdP's metadata XML file, saved in the preceding step.

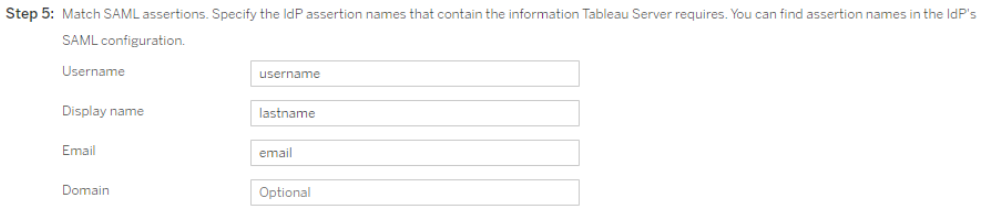


Here, for example the name of the downloaded file is *descriptor01.xml*

14. **Step 5** - Add the following values for the respective fields:

Field Name	Sample Value	Description
Username	"username"	Enter the <field name> as defined in Mappers in step 7.
Display Name	"lastname"	
Email	"email"	
Domain		Enter your domain name. However it is optional.

The following image illustrates the UI screen:

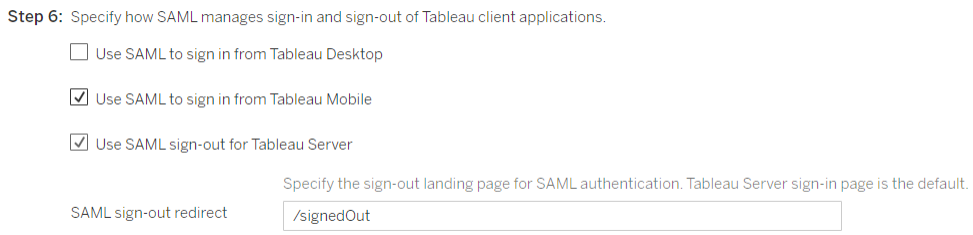


15. **Step 6** - Select the check boxes for the following parameters:

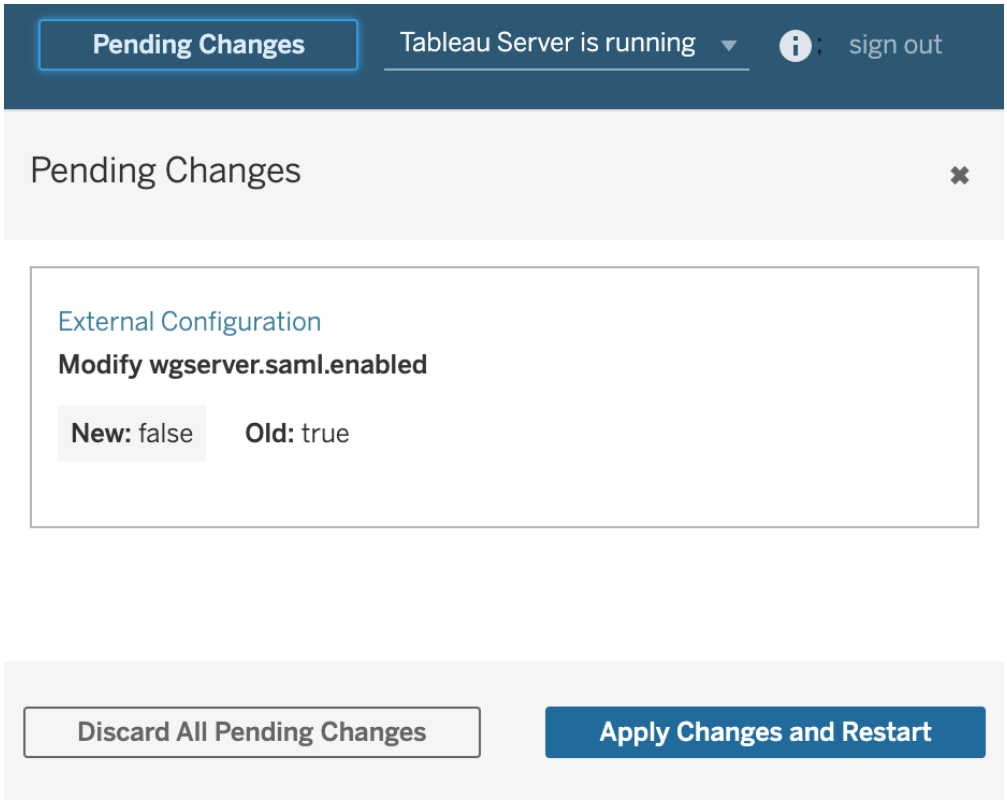
    - Use SAML to sign in from Tableau Mobile.
    - Use SAML sign-out for Tableau Server.

Enter `/signedOut` in **SAML sign-out redirect** field to sign out of the application.

The following image illustrates the UI screen:



16. Click **Save Pending Changes** button.
17. From the navigation bar, **Pending Changes** option and **Apply Changes and Restart** to reflect all the changes.



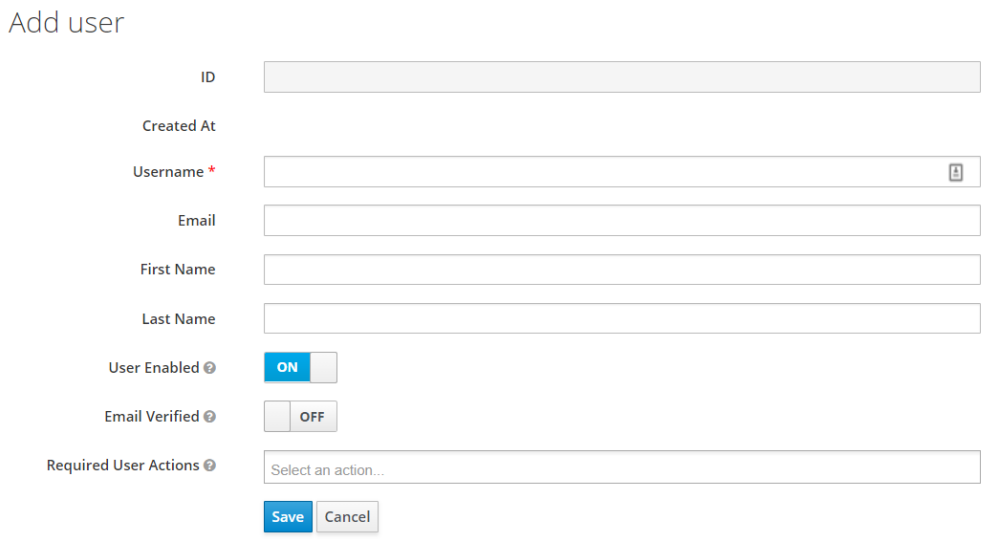
7.1.7 Create Tableau Admin User

Perform the following steps to create a tableau admin user in keycloak:

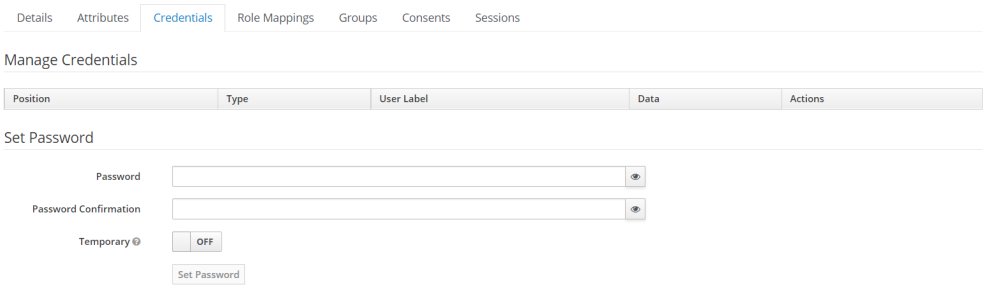
1. Log into Keycloak UI using *URL: https://<LB-VIP>/auth*
2. Navigate to **Manage** > **Users** > **Add User** to create a new user.
3. On **Add User** screen, fill in the details. Refer to the following table to determine the values for each available field:

Field Name	Description
Username	This should be same as your tableau server admin user
Email	The tableau admin user created on Keycloak/Guavus IAM must have an email associated with it.

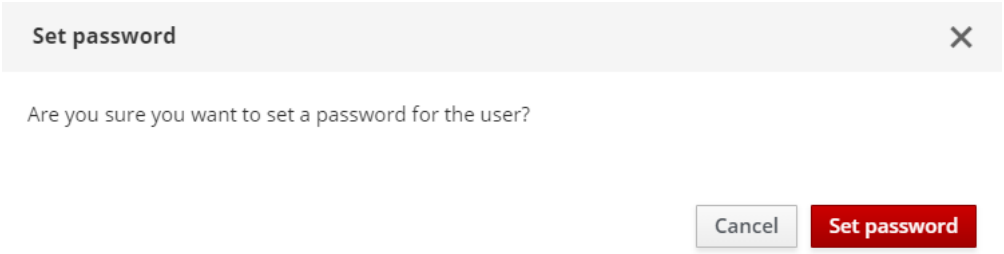
The following image illustrates the UI screen:



4. Click the **Save** button to keep the changes. A new user is now created.
5. Once you have created the user, the following screen appears on UI:



6. In **Set Credentials** section, enter your password details and click **Set Password** button. Ensure that this password is as same as provided in Tableau Server and *disable* the **Temporary Slider** option.
7. On **Set Password** dialog box, select **Set Password** option to keep the changes.



A new admin user is created successfully.

7.2 Install Tableau Desktop

This section describes the steps to install Tableau Desktop on your machine.
Tableau Desktop is supported in the following environments:

- "Install Tableau Desktop on Mac" below
- "Install Tableau Desktop on Windows" on page 65

7.2.1 Downloading Tableau Desktop

Download and Install tableau Desktop from *https://www.tableau.com/products/desktop/download*

7.2.2 Install Tableau Desktop on Mac

This section describes the tasks to install Tableau Desktop on your Mac.

Task 1 - Connecting to Clickhouse

To create Clickhouse connection from Tableau Desktop on Mac Operating System.

Prerequisites

Ensure that you have ODBC dylib file. To generate this file, refer to *https://github.com/ClickHouse/clickhouse-odbc*.

1. Run the following command to navigate to `libclickhouseodbc` file:

   ```
   cd clickhouse-odbc/build/clickhouse-odbc-1.1.9-Darwin/lib/
   ```

2. Run the following command to copy the contents of OBDC file to local Mac Library:

   ```
   cp libclickhouseodbc* /usr/local/lib/
   ```

Task 2 - Configuring Data Source Name

Perform the following steps for configuring Data Source Name (DSN) through command Prompt:

1. Run the following command to edit `odbc.ini` file:

```
vi ~/.odbc.ini
```

Update the values for the highlighted parameter as mentioned in the following content of `odbc.ini` file:

```
; Insert the content of this file into ~/.odbc.ini or
/etc/odbc.ini files.
[ODBC Data Sources]
ClickHouse DSN (ANSI)    = ClickHouse ODBC Driver (ANSI)
ClickHouse DSN (Unicode) = ClickHouse ODBC Driver (Unicode)
[ClickHouse DSN (ANSI)]
Driver      = ClickHouse ODBC Driver (ANSI)
Description = DSN (localhost) for ClickHouse ODBC Driver
(ANSI)
;## New all-in one way to specify connection with [optional]
settings:
; Url =
https://default:password@localhost:8443/query?database=default
&max_result_bytes=4000000&buffer_size=3000000
; ...or minimal (will connect to port 8443 if protocol is
"https://" or 8123 if it is "http://"):
; Url = https://localhost
;## Old way:
Server      = 192.168.192.206
Database    = dma_perf29
UID         = clickhouse
PWD         = admin123
Port        = 8123
Proto       = http
trace       = 1
tracefile   = /tmp/clickhouse-odbc.log
debug       = 1
```

```
debugfile   = /tmp/debugClickhouse-odbc.log
; Timeout for http queries to ClickHouse server (default is 30
seconds)
; Timeout=60
; SSLMode:
;   allow   - ignore self-signed and bad certificates
;   require - check certificates (and fail connection if
something wrong)
; SSLMode = require
; PrivateKeyFile =
; CertificateFile =
; CALocation =
; DriverLog = yes
; DriverLogFile = /tmp/chlickhouse-odbc-driver.log
[ClickHouse DSN (Unicode)]
Driver      = ClickHouse ODBC Driver (Unicode)
Description = DSN (localhost) for ClickHouse ODBC Driver
(Unicode)
; ...
;## Old way:
Server      = 192.168.192.206
Database    = dma_perf29
UID         = clickhouse
PWD         = admin123
Port        = 8123
Proto       = http
trace       = 1
tracefile   = /tmp/clickhouse-odbc_unicode.log
debug       = 1
debugfile   = /tmp/debugClickhouse-odbc_unicode.log
[ODBC]
Trace        = 1
TraceFile    = trace.log
```

```
TraceLibrary = /Library/ODBC/
```

2. Run the following command to edit `odbcinst.ini` file:

   ```
   vi ~/.odbcinst.ini
   ```

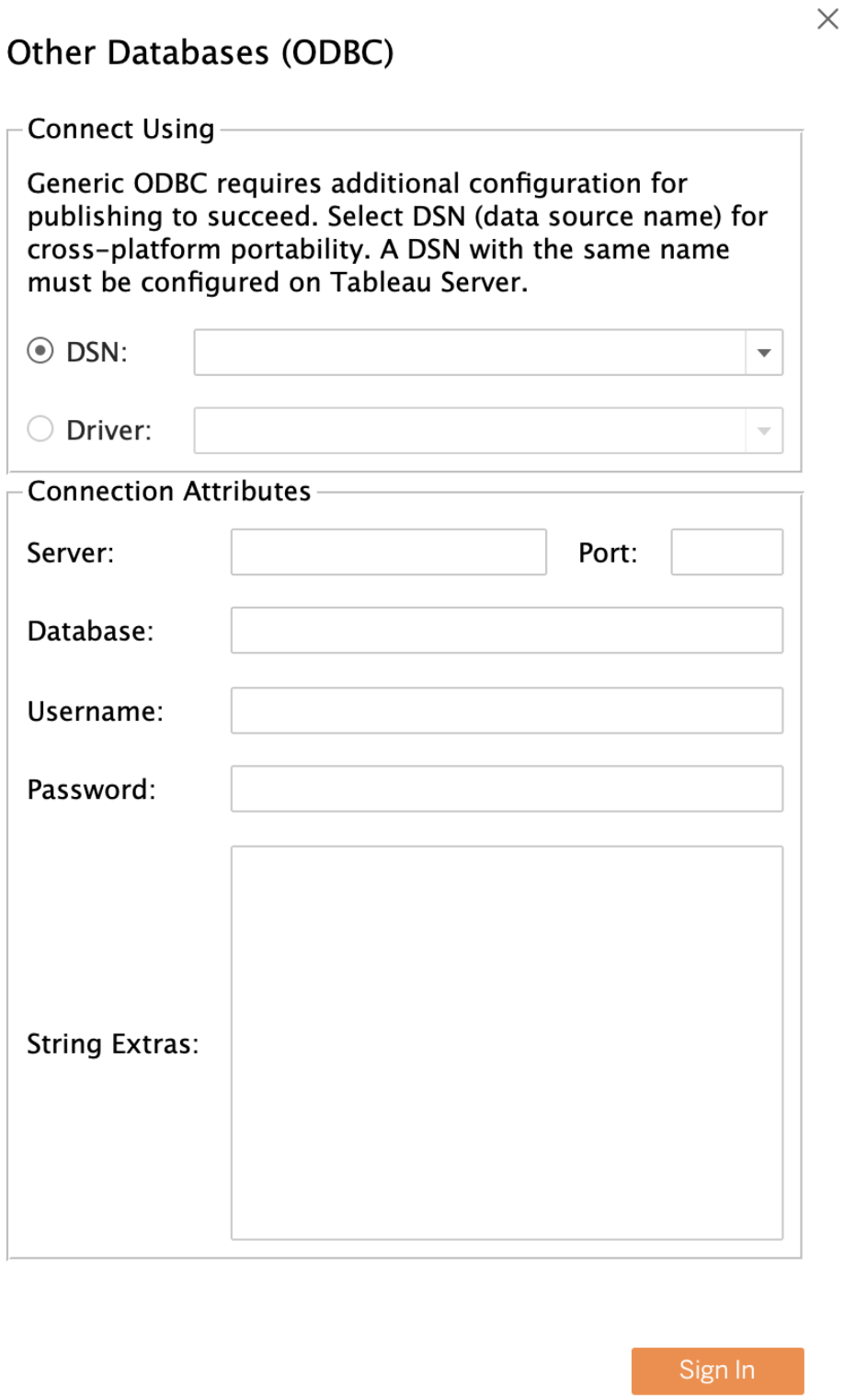
   Update the values as mentioned in the following content of `odbcinst.ini` file:

   ```
   [ODBC Drivers]
   ClickHouse ODBC Driver (ANSI)    = Installed
   ClickHouse ODBC Driver (Unicode) = Installed
   [ClickHouse ODBC Driver (ANSI)]
   Driver      = /usr/local/lib/libclickhouseodbc.dylib
   Setup       = /usr/local/lib/libclickhouseodbcw.dylib
   Description = ODBC Driver (ANSI) for ClickHouse
   UsageCount  = 1
   [ClickHouse ODBC Driver (Unicode)]
   Driver      = /usr/local/lib/libclickhouseodbc.dylib
   Setup       = /usr/local/lib/libclickhouseodbc.dylib
   Description = ODBC Driver (Unicode) for ClickHouse
   UsageCount  = 1
   ```

Task 3 - Connecting to Database

Perform the following steps to connect to the ODBC database:

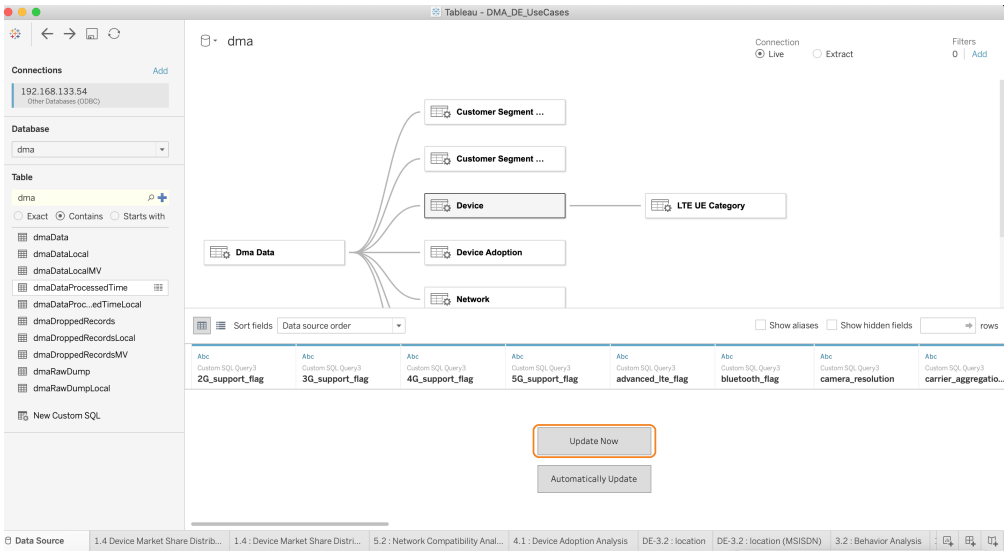
1. Navigate to **New Data Source** > **Connect** > **To a Server** > **More** > **Other Database (ODBC)**



2. From the **Other Database (ODBC)** dialog box, select the required **DSN**.
3. In the **Connection Attributes** section, fill in your required details and click **Sign In** button.

   The connection will be established.
4. Select the **Database** name that you want to access.
5. Type the name of the required **Table**. You can select any of the available options to search for the table:

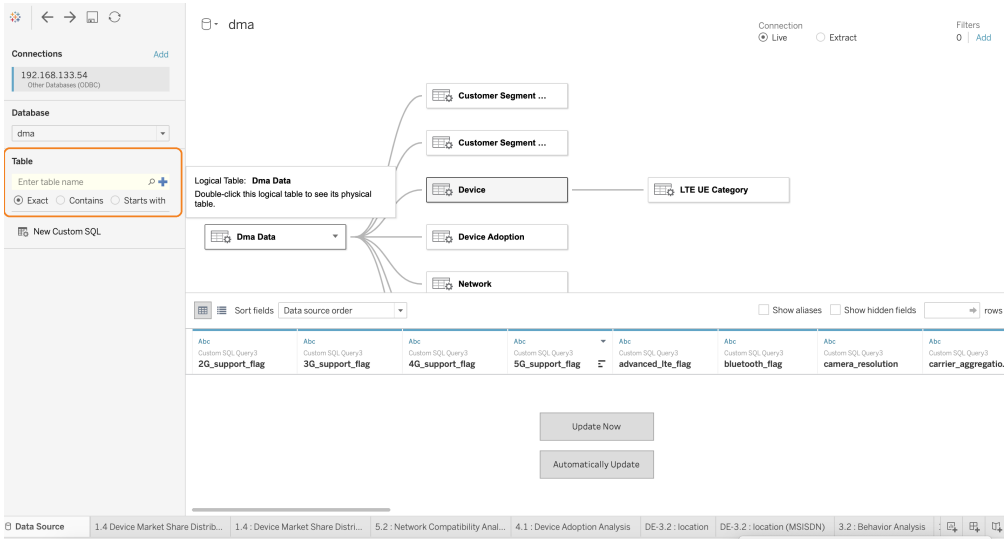
Option	Description
Exact	Select this when you are typing the exact name of the table.
Contains	Select this when you are typing a few key words of the table name.
Starts With	Select this when you are typing the initials of the table name.



6. To use the table schema, pick the required table and drag it to the working pane.

7. To fetch the data from the table schemas, click **Update Now** button. It will populate the sample data.

   The following image illustrates the UI screen:



8. Select the sheet available in the bottom of the page to create a new work-sheet:

Once you have executed all the steps, you can use the connected datasource and create your dashboards.

7.2.3 Install Tableau Desktop on Windows

Task 1 - Connecting to Clickhouse

To create clickhouse connection from Tableau Desktop on Windows Operating System.

Prerequisite

Download and install Clickhouse ODBC Driver from *https://-github.com/ClickHouse/clickhouse-odb-c/releases/download/v1.1.9.20201226/clickhouse-odbc-1.1.9-win64.msi*

Task 2 - Configuring Data Source Name

- for 32-bit applications (and drivers) execute %systemdrive%\Windows\SysWoW64\Odbcad32.exe
- for 64-bit applications (and drivers) execute %systemdrive%\Windows\System32\Odbcad32.exe

To configure ODBC Manager through dialog box:

1. On the **User DNS** tab, click **Add..** button to add the User DSN with the following details:

Field	Sample Value	Description
Name		Clickhouse
Description		DSN (localhost) for ClickHouse ODBC Driver (Unicode)
URL	https://localhost	It is the Clickhouse node where the database is stored.
Host	192.168.192.206	
Port	8123	
Database	dma_perf29	
SSLMode	require	The value can be either of the two: • allow - ignore self-signed and bad certificates • require - check certificates (and fail connection if something wrong)
User		It is the database user name.

Field	Sample Value	Description
Password		It is the password of database user name.
Timeout	60	It is for http queries to ClickHouse server. The default value is 30.

Task 3 - Connecting to Database

To connect to the ODBC database.

1. Navigate to **New Data Source** > **To a Server** > **More** > **Other Database (ODBC)**
2. From the **Other Database (ODBC)** dialog box, select the required **DSN** and

click **Connect** button.

3. In the **Connection Attributes** section, fill in your required details and click **Sign In** button.

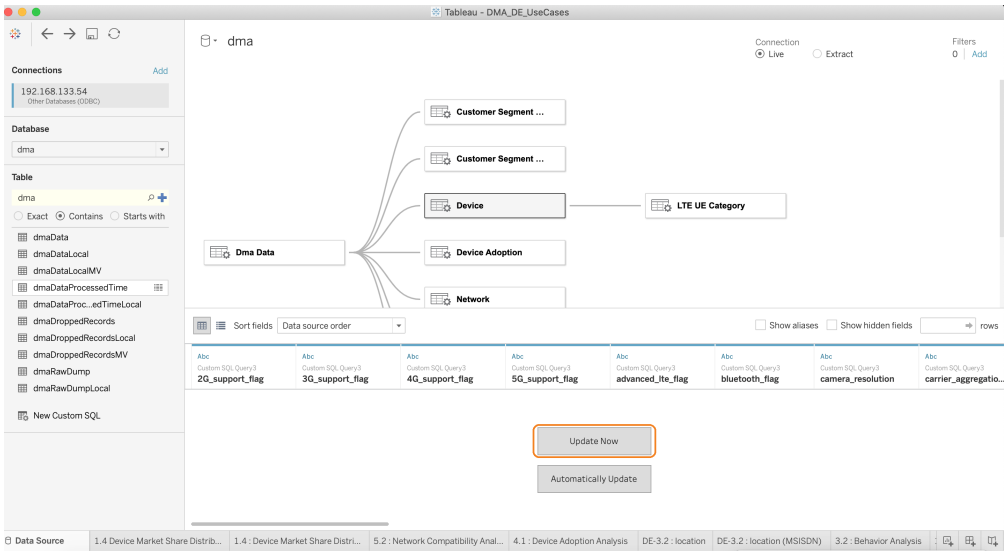
   The connection will be established.

4. Select the **Database** name that you want to access.

5. Type the name of the required **Table**. You can select any of the available options to search for the table:

Option	Description
Exact	Select this when you are typing the exact name of the table.
Contains	Select this when you are typing a few key words of the table name.
Starts With	Select this when you are typing the initials of the table name.

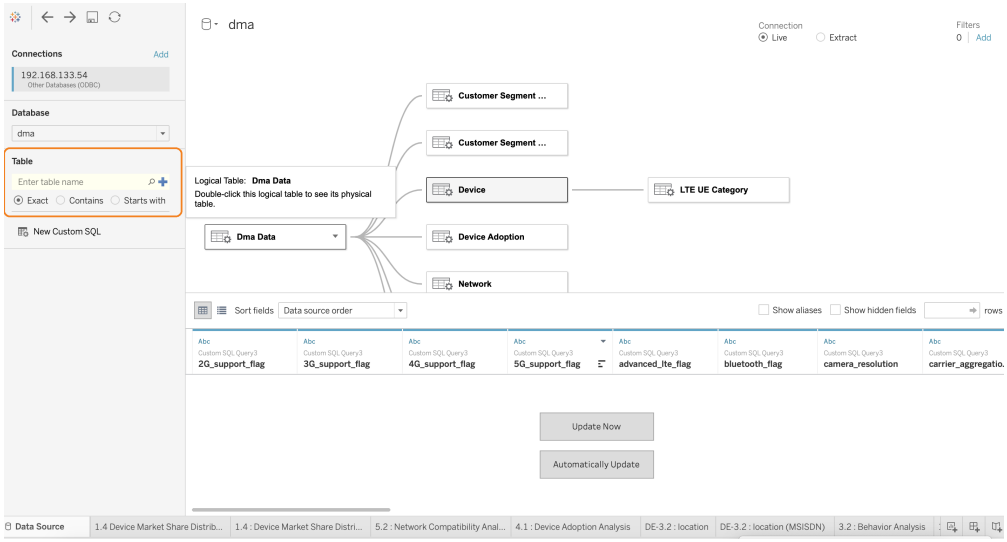
The following image illustrates the UI screen:



6. To use the table schema, pick the required table and drag it to the working pane.

7. To fetch the data from the table schemas, click **Update Now** button. It will populate the sample data.

   The following image illustrates the UI screen:



8. Select the sheet available in the bottom of the page to create a new work-sheet:

Once you have executed all the steps, you can use the connected datasource and create your dashboards.

7.3 Install ODBC Driver on Tableau Server

This section descries the steps to install and configure ODBC Driver to connect to Clickhouse from the Tableau server.

Perform the following steps:

1. Log into the master node and run the following commands to get the listed Clickhouse certificates from the clickhouse-tls secret.
    - tls.key
    - tls.crt
    - ca.crt

```
kubectl -n clickhouse get secret clickhouse-tls \
-o jsonpath='{.data.ca\.crt}'  | base64 -d > ca.crt
kubectl -n clickhouse get secret clickhouse-tls \
-o jsonpath='{.data.tls\.crt}'  | base64 -d > tls.crt
kubectl -n clickhouse get secret clickhouse-tls \
-o jsonpath='{.data.tls\.key}'  | base64 -d > tls.key
```

2. You need to copy all three files to Tableau server in required directory and use the same paths in `odbc.ini` file while configuring values for PrivateKeyFile, CertificateFile, and CALocation.
3. Download `clickhouse-odbc-1.1.9-1.el7.x86_64.rpm` RPM from SFTP server.
4. Run the following command to install clickhouse ODBC driver:

```
sudo yum install clickhouse-odbc-1.1.9-1.el7.x86_64.rpm
```

5. Run the following command to create a `/data/clickhouse/lib64/` directory:

```
sudo mkdir -p /data/clickhouse/lib64/
```

6. Run the following command to copy the contents of OBDC files to your local Library:

```
sudo cp /usr/local/lib64/libclickhouseodbc.so \
/data/clickhouse/lib64/

sudo cp /usr/local/lib64/libclickhouseodbcw.so \
/data/clickhouse/lib64/
```

7. Run the following command to update the permissions for the previously copied files:

   ```
   sudo chown -R <tableau_admin_user> /data/clickhouse/lib64/
   ```

8. Update the following content in `/etc/odbcinst.ini` file to add the Clickhouse drivers:

```
[guavus@siqperf01-mst-02 ~]$ cat /etc/odbcinst.ini
[ODBC Drivers]
PostgreSQL Unicode=Installed
Cloudera ODBC Driver for Impala = Installed
ClickHouse ODBC Driver (ANSI)    = Installed
ClickHouse ODBC Driver (Unicode) = Installed



[PostgreSQL Unicode]
Description=ODBC for PostgreSQL
Driver=/data/tableau/tableau_driver/postgresql-
odbc/psqlodbcw.so
FileUsage=1

[ClickHouse ODBC Driver (ANSI)]
Description = ODBC Driver (ANSI) for ClickHouse
Driver      = /data/clickhouse/lib64/libclickhouseodbc.so
Setup       = /data/clickhouse/lib64/libclickhouseodbc.so
UsageCount  = 1

[ClickHouse ODBC Driver (Unicode)]
```

```
Description = ODBC Driver (Unicode) for ClickHouse
Driver      = /data/clickhouse/lib64/libclickhouseodbcw.so
Setup       = /data/clickhouse/lib64/libclickhouseodbcw.so
UsageCount  = 1
```

9. Update the following content in `~/.odbc.ini` file to add the Clickhouse drivers:

   Ensure that you are logged in as a Tableau Admin user.

```
[guavus@siqperf01-mst-02 ~]$ cat ~/.odbc.ini
# Insert the content of this file into ~/.odbc.ini or
/etc/odbc.ini files.

[ODBC Data Sources]
ClickHouse DSN (ANSI)    = ClickHouse ODBC Driver (ANSI)
ClickHouse DSN (Unicode) = ClickHouse ODBC Driver (Unicode)

[ClickHouse DSN (ANSI)]
Driver      = ClickHouse ODBC Driver (ANSI)
Description = DSN (localhost) for ClickHouse ODBC Driver
(ANSI)

### New all-in one way to specify connection with [optional]
settings:
# Url =
https://default:password@localhost:8443/query?database=default
&max_result_bytes=4000000&buffer_size=3000000

# ...or minimal (will connect to port 8443 if protocol is
"https://" or 8123 if it is "http://"):
# Url = https://localhost

### Old way:
```

```
Server = 192.169.192.212
Database = default
# UID = default (use this value from topic - Define Clickhouse
Tableau Variables)
# PWD = password (use this value from topic - Define
Clickhouse Tableau Variables)
Port = 8123
Proto = http

# Timeout for http queries to ClickHouse server (default is 30
seconds)
# Timeout=60

# SSLMode:
#   allow   - ignore self-signed and bad certificates
#   require - check certificates (and fail connection if
something wrong)
# SSLMode = require
# PrivateKeyFile =
# CertificateFile =
# CALocation =

# DriverLog = yes
# DriverLogFile = /tmp/chlickhouse-odbc-driver.log

[ClickHouse DSN (Unicode)]
Driver      = ClickHouse ODBC Driver (Unicode)
Description = DSN (localhost) for ClickHouse ODBC Driver
(Unicode)
# ...
```

Note: While connecting to secure (SSL or TLS) Clickhouse, ensure to update the value for all the following parameters in SSLMode property:

- SSLMode
- PrivateKeyFile (this file is already generated in section Install Tableau Server, you can use the same file)
- CertificateFile (this file is already generated in section Install Tableau Server, you can use the same file)
- CALocation

7.4 Updating Tableau Workbook and Tableau Data Source

Based on the upgrade of Service-IQ DMA v3.1.0, you must make certain changes to the Tableau workbook.

Perform the following steps to access the Tableau Workbook and Tableau Data source file from the provided tar:

1. Log into the Tableau Server terminal using SSH.
2. Download `DMA_DE_Usecases.tar.gz` file from the SFTP server.

   Note: Refer to the Release Notes for the SFTP location. Contact your Technical Support team for SFTP credentials.
3. Run the following command to extract the tar file:

   ```
   tar -xzvf DMA_DE_Usecases.tar.gz
   ```

   This tar file contains the following two Tableau workbook files:

   - `DMA_DE_DMA_UseCases.twb`
   - `dma.tds`
4. Edit the workbook and data source files.
5. Search for `name='Custom SQL Query4'` string in the files. Replace the existing query with the following.

   ```
   SELECT
     dma.`dmaData`.`firstVisibleTime` AS `first_time_seen`,
   ```

```
  dma.`dmaData`.`imei` AS `imei`,
  dma.`dmaData`.`imsi` AS `imsi`,
  dma.`dmaData`.`lastVisibleTime` AS `last_time_seen`,
  dma.`dmaData`.`msisdn` AS `msisdn`,
  dma.`dmaData`.`sv` AS `SVN`,
  dma.`dmaData`.`tac` AS `tac`,
  dma.`dmaData`.`ratType` AS `RAT_name`
FROM dma.`dmaData`
```

The existing query may resemble as follows:

```
SELECT
  dma.`dmaData`.`firstVisibleTime` AS `first_time_seen`,
  dma.`dmaData`.`imei` AS `imei`,
  dma.`dmaData`.`imsi` AS `imsi`,
  dma.`dmaData`.`lastVisibleTime` AS `last_time_seen`,
  dma.`dmaData`.`msisdn` AS `msisdn`,
  dma.`dmaData`.`tac` AS `tac`,
  dma.`dmaData`.`ratType` AS `RAT_name`
FROM dma.`dmaData`
```

6. Search for the following block in the files.

```
          <metadata-record class='column'>
            <remote-name>msisdn</remote-name>
            <remote-type>129</remote-type>
            <local-name>[msisdn]</local-name>
            <parent-name>[Custom SQL Query4]</parent-name>
            <remote-alias>msisdn</remote-alias>
            <ordinal>5</ordinal>
            <local-type>string</local-type>
            <aggregation>Count</aggregation>
            <width>150</width>
            <contains-null>false</contains-null>
```

```
            <collation flag='0' name='binary' />
            <attributes>
              <attribute datatype='string'
name='DebugRemoteType'>&amp;quot;SQL_VARCHAR&amp;quot;</attribute>
              <attribute datatype='string'
name='DebugWireType'>&amp;quot;SQL_C_CHAR&amp;quot;</attribute>
              <attribute datatype='string'
name='TypeIsVarchar'>&amp;quot;true&amp;quot;</attribute>
            </attributes>
            <_.fcp.ObjectModelEncapsulateLegacy.true...object-
id>[dmaData_
CBE2AE6B426641239C24F47F508D0B42]</_.fcp.ObjectModelEncapsulat
eLegacy.true...object-id>
          </metadata-record>
```

7. Add the following meta-data block after the preceding block.

```
<metadata-record class='column'>
            <remote-name>SVN</remote-name>
            <remote-type>129</remote-type>
            <local-name>[SVN]</local-name>
            <parent-name>[Custom SQL Query4]</parent-name>
            <remote-alias>SVN</remote-alias>
            <ordinal>1</ordinal>
            <local-type>string</local-type>
            <aggregation>Count</aggregation>
            <width>10</width>
            <contains-null>false</contains-null>
            <collation flag='0' name='binary' />
            <attributes>
              <attribute datatype='string'
name='DebugRemoteType'>&amp;quot;SQL_VARCHAR&amp;quot;</attribute>
              <attribute datatype='string'
```

```
name='DebugWireType'>&amp;quot;SQL_C_CHAR&amp;quot;</attribute>
              <attribute datatype='string'
name='TypeIsVarchar'>&amp;quot;true&amp;quot;</attribute>
            </attributes>
            <_.fcp.ObjectModelEncapsulateLegacy.true...object-
id>[dmaData_
CBE2AE6B426641239C24F47F508D0B42]</_.fcp.ObjectModelEncapsulat
eLegacy.true...object-id>
          </metadata-record>
```

8. Delete the previously uploaded workbook and TDS from the Tableau Server UI. It is recommended to download the workbook to your local before you delete from the server.
9. Perform the following two tasks to upload the Tableau Workbook and Tableau Data Source files to Tableau Server.
    - "Publish the Data Source Connection" on the next page
    - "Import Tableau Workbook" on page 81

7.5 Publish the Data Source Connection

Note: Before executing the steps in Publish the Data Source Connection section, ensure that you have already performed the steps mentioned in "Updating Tableau Workbook and Tableau Data Source" on page 74

This section describes the steps to publish a data source connection:

1. Log into the Tableau Server terminal using SSH.
2. Open `dma.tds` file in editor to update the values for the following parameters of database connection as required:
    - **source-platform:** Change this value to 'win' in case of Windows machine.
    - **id:** This value should be as same as used in workbook.
    - **caption:** Change this value as required.
    - **dbname:** Change this value with the database name of your choice.
    - **server:** Change this value as required.
    - **port:** Change this database port value to 8123.
    - **odbc-dsn:** Change this value as per your Data Source Name.

```
<datasource formatted-
name='federated.0i1y68x0zlrl4217cdvpw1ddksul (copy 5)'
inline='true' source-platform='mac' version='18.1'
xml:base='http://dma-tableau01.cloud.in.guavus.com'
xmlns:user='http://www.tableausoftware.com/xml/user'>
```

```
<repository-location id='dma' path='datasources'
revision='1.0' />
  <connection class='federated'>
    <named-connections>

<named-connection caption='dma05-wk-02.cloud.in.guavus.com'
```

```
name='genericodbc.12l13vg023fkka100lp170sjc7dh (copy 4)'>
        <connection class='genericodbc' dbname='dma' odbc-
connect-string-extras='' odbc-dbms-name='ClickHouse' odbc-
driver='ClickHouse ODBC Driver (Unicode)' odbc-dsn='ClickHouse
DSN (Unicode)' odbc-suppress-connection-pooling='' odbc-use-
connection-pooling='' port='8123' server='dma05-wk-
02.cloud.in.guavus.com' username='clickhouse'>
                <connection-customization class='genericodbc'
enabled='false' version='18.1'>
```

3. Run the following command to publish the data source connection:

```
tabcmd publish "<path to dma.tds>" -n "dma" --db-\
username "<db_username>" --db-password "<db_\
password>" --no-certcheck
```

Note: Use the values for username and password from topic - Define Clickhouse Tableau Variables.

You can now create a new workbook or upload an existing workbook using this data source connection.

Note: If you want to upload a workbook, ensure that the name of the id entered in the previous command is same as the id of repository.

```
<repository-location id='dma' path='datasources'
revision='1.0' />
```

7.6 Create Workbook using the Published Connection

Once you have uploaded the Clickhouse connection on the Tableau server, we will be using the same on the server to connect to the clickhouse and create Workbooks.

1. Type the *URL - http://<ip>/#/home* to open Tableau Server UI.
2. Log into the Tableau Server using the credentials created in Section - "Install Tableau Desktop " on page 59. If you have enabled Keycloak AD user, use it's credentials.
3. Navigate to **Create** > **Workbook**
4. In the **Workbook**, use the uploaded clickhouse connection to create your workbook on Tableau server.

7.7 Import Tableau Workbook

Note: Before executing the steps in Import Tableau Workbook section, ensure that you have already performed the steps mentioned in "Updating Tableau Workbook and Tableau Data Source" on page 74

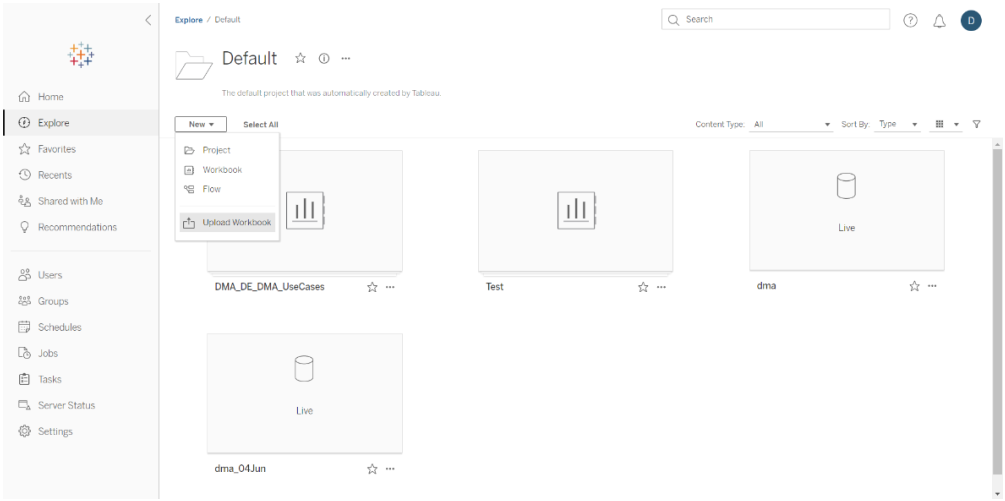
This section describes the steps for importing Tableau workbook in Tableau Desktop.

1. Open `DMA_DE_DMA_UseCases.twb` file in editor to update the values for the following parameters of database connection as required:
    - **source-build:** Change this value as per the installed version of Tableau Desktop.
    - **source-platform:** Change this value to 'win' in case of Windows machine.
    - **caption:** Change this value as required.
    - **dbname:** Change this value with the database name of your choice.
    - **server:** Change this value as required.
    - **odbc-dsn:** Change this value as per your Data Source Name.
    - **Username:** Change this value as per the value defined in topic - Define Clickhouse Tableau Variables.

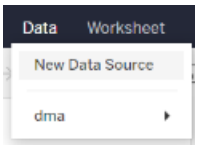
```
<datasource caption='dma' inline='true'
name='federated.0i1y68x0zlrl4217cdvpw1ddksul (copy 5)'
version='18.1'>
      <repository-location id='dma' path='datasources'
revision='1.0' />

<workbook source-build='2020.4.3 (20204.21.0315.1000)' \
source-platform='mac' version='18.1' \
xml:base='http://dma-tableau01.cloud.in.guavus.com' \
xmlns:user='http://www.tableausoftware.com/xml/user'>
```

```
<named-connection caption='192.168.133.54' \
name='genericodbc.12l13vg023fkka100lp170sjc7dh'>
<connection class='genericodbc' dbname='dma' \
odbc-connect-string-extras='' odbc-dbms-name='ClickHouse' \
odbc-driver='ClickHouse ODBC Driver (Unicode)' \
odbc-dsn='ClickHouse DSN (Unicode)' \
odbc-suppress-connection-pooling='' \
odbc-use-connection-pooling='' port='8123' \
server='192.168.133.54' username='clickhouse'>
```

2. Once the changes are done, save the file.
3. Import the Tableau workbook on Tableau Server.



4. Navigate to **Data** > **New Data** Source to change the data source connection, if required.



5. On the **Connect to Data** dialog box, select the required datasource and click **Connect** button.

8. Appendix B: Kafka Topic

8.1 Create Kafka Topic for Kubernetes

Run the following command to create kafka topic:

```
cat <<EOF | kubectl apply -f -
---
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaTopic
metadata:
  name: dmc
  namespace: kafka
  labels:
    strimzi.io/cluster: kafka
spec:
  topicName: dmc_output
  partitions: 1
  replicas: 1
  config:
    retention.ms: 259200000
---
apiVersion: kafka.strimzi.io/v1beta1
kind: KafkaTopic
metadata:
  name: cdr
  namespace: kafka
  labels:
    strimzi.io/cluster: kafka
spec:
  topicName: cdr_output
  partitions: 1
  replicas: 1
  config:
```

```
    retention.ms: 259200000
EOF
```

8.2 Set Kafka SSL

Run the following command to set kafka SSL:

```
kubectl -n kafka get kt

kubectl -n kafka get secret kafka-cluster-ca-cert \
-o jsonpath='{.data.ca\.password}'  | base64 -d
kubectl -n kafka get secret kafka-user -o \
jsonpath='{.data.user\.password}'  | base64 -d
kubectl -n kafka get secret kafka-user -o \
jsonpath='{.data.user\.p12}'  | base64 -d >user.p12
kubectl -n kafka get secret kafka-cluster-ca-cert \
-o jsonpath='{.data.ca\.p12}'   | base64 -d > \
kafka-cluster-ca.p12

vi ssl.properties
security.protocol = SSL
ssl.keystore.location = /tmp/user.p12
ssl.keystore.password = <update this>
ssl.truststore.location = /tmp/kafka-cluster-ca.p12
ssl.truststore.password = <update this>

kubectl cp ssl.properties kafka-kafka-0:/tmp/ -n kafka
kubectl cp user.p12 kafka-kafka-0:/tmp/ -n kafka
kubectl cp kafka-cluster-ca.p12 kafka-kafka-0:/tmp/ -n kafka
```

8.3 View Kafka Topic for Kubernetes

Run the following command to view kafka topic:

```
kubectl -n kafka  exec -it kafka-kafka-0 -- bin/kafka\
-topics.sh --bootstrap-server kafka-kafka\
```

```
-bootstrap.kafka.svc.kubernetes.local:9092 --list -\
-command-config /tmp/ssl.properties
```

8.4 Produce Kafka Topic for Kubernetes

Run the following command to produce kafka topic:

```
kubectl -n kafka  exec -it kafka-kafka-0 -- bin/kafka\
-console-producer.sh --bootstrap-server kafka-kafka\
-bootstrap.kafka.svc.kubernetes.local:9092 --topic \
<topic-name> --producer.config /tmp/ssl.properties
```

8.5 Consume Kafka Topic for Kubernetes

Run the following command to consume kafka topic:

```
kubectl -n kafka  exec -it kafka-kafka-0 -- bin/kafka\
-console-consumer.sh --bootstrap-server kafka-kafka\
-bootstrap.kafka.svc.kubernetes.local:9092 --topic \
dmc_output --from-beginning --consumer.config \
/tmp/ssl.properties
```

9. Appendix C: Synchronize LDAP Users and Groups (Optional)

9.1 Set Up LDAP Users Sync

You can create a user federation by providing details of your Active Directory (AD) server. The users are synced from your AD to Guavus-IAM. By default, this functionality is disabled. If you want to use this feature, perform the following steps:

1. Navigate to the ansible directory:

   ```
   cd /data/dma
   ```

2. Run the following command to edit the `guavus-iam.yml` file present in the inventory:

   ```
   vi /data/dma/inventory/<inventory site name>/group_
   vars/all/guavus-iam.yml
   ```

   Here, replace `<inventory site name>` with the value of site variable.

3. Enable user federation by updating the value of the variable as follows:

   `guavus_iam_util_sync_ldap_users: "true"`

   The following table lists the variables that must be updated to provide connection details of your AD server.

   Note: Sample values are for your reference. Update these variables as per your environment. Contact your system administrator for these values.

Variable	Sample Value	Description
guavus_iam_util_ldap_name	"ldap"	Name of the user federations that will be displayed on Guavus-IAM.

Variable	Sample Value	Description
guavus_iam_util_ldap_connection_url	"ldap://<local_host>:389"	Connection URL to the LDAP server.
guavus_iam_util_ldap_bind_dn	"CN=Administrator,CN=Users,DC=in,DC=guavus,DC=com"	DN of the LDAP admin that will be used by Guavus IAM to access the LDAP server.
guavus_iam_util_ldap_bind_credential	"password"	Password of the LDAP admin.

Variable	Sample Value	Description
guavus_iam_util_ldap_username_attribute	"cn"	Name of the LDAP attribute that is mapped as Guavus IAM username. For multiple LDAP server vendors, this value can be 'uid'. For the Active directory, this value can be 'sAMAccountName' or 'cn'. The attribute must be filled in for all the LDAP user records that you want to import from LDAP to Guavus IAM.
guavus_iam_util_ldap_rdn_attribute	"cn"	Name of the LDAP attribute that is used as RDN (top attribute) of a typical user DN. Usually, it is the same as Username LDAP attribute, however it is not required. For example, in case of Active directory, it is common to use 'cn' as the RDN attribute when the username attribute might be 'sAMAccountName'.

Variable	Sample Value	Description
guavus_iam_util_ldap_uuid_attribute	"objectGUID"	Name of the LDAP attribute that is used as the unique object identifier (UUID) for objects in LDAP. For multiple LDAP server vendors, the value is 'entryUUID'; however for some, this value is different. For example, in case of Active directory, this value must be 'objectGUID'. If the LDAP server does not support the notion of UUID, use any other attribute that is supposed to be unique among LDAP users in a tree. For example, 'uid' or 'entryDN'.

Vari-able	Sample Value	Description
guavu-s_iam_util_ldap_user_object_classes	"person, organizationalPerson, user"	All values of LDAP objectClass attribute for users in LDAP divided by comma. For example, 'inetOrgPerson, organ-izationalPerson' . Newly created Guavus IAM users will be written to LDAP with all those object classes and existing LDAP user records are found if they contain all those object classes. guavus_iam_util_ldap_user_object_classes: "per-son, organizationalPerson, user"
guavu-s_iam_util_ldap_users_dn	"ou=UserIDs, dc=in,D-DC=guavus,DC=com"	Full DN of LDAP tree where users exist. This DN is the parent of LDAP users. For example, 'ou=use-ers,dc=example,dc=com' assuming that a typical user will have a DN as 'uid-=john,o-u=use-ers,dc=example,dc=com'.

Vari-able	Sample Value	Description
guavu-s_iam_util_ldap_change-d_sync_period	"300"	Time period denoting how frequently do you want to sync users from LDAP. The default value is 300 seconds. Omit this vari-able, if you want to con-tinue with the default value.

9.2 Set Up LDAP Group Sync

The groups are synced from your AD to Guavus-IAM. By default, this functionality is disabled. If you want to use this feature, perform the following steps:

1. Enable LDAP Groups synchronization by updating the value of the variable as follows:

```
guavus_iam_util_sync_ldap_groups: "true"
```

The following table lists the variables that must be updated in the `guavus-iam.yml` file.

Note: Sample values are for your reference. You must update the variables as per your environment. Contact your system administrator for these values.

Variable	Sample Value	Description
guavus_iam_util_ldap_map-per_name	"ldap_mapper"	Name of the LDAP mapper that will appear for your user fed-eration in Guavus IAM. The default value is ldap_mapper. Omit this variable, if you want to continue with the default value.

Variable	Sample Value	Description
guavus_iam_util_ldap_mapper_groups_dn	"OU=Groups,D-DC=in,DC=guavus,DC=com"	LDAP DN where groups of this tree are saved. For example, 'ou=groups,-,dc=example,dc=org'.
guavus_iam_util_ldap_mapper_group_name_ldap_attribute	"cn"	Name of the LDAP attribute that is used in group objects for name and RDN of the group. Usually, it is 'cn' . In this case, typical group/role object may have DN as 'cn=Group1,o-u=groups,-,dc=example,dc=org'.
guavus_iam_util_ldap_mapper_membership_attribute_type	"DN"	DN or UID. DN means that LDAP group has its members declared in form of their full DN. For example 'member: uid-=john,o-u=users,dc=example,dc=com' . UID means that LDAP group has its members declared in form of pure user uids. For example 'memberUid: john'.

Variable	Sample Value	Description
guavus_iam_util_ldap_mapper_membership_user_ldap_attribute	"cn"	Used only if Membership Attribute Type is UID. It is the name of LDAP attribute on user that is used for membership mappings. Usually, it is 'uid' . For example, if the value of 'Membership User LDAP Attribute' is 'uid' and LDAP group has 'memberUid: john', then that particular LDAP user is expected to have attribute 'uid: john'.
guavus_iam_util_ldap_mapper_groups_ldap_filter	""	LDAP Filter adds additional custom filter to the whole query for retrieving LDAP groups. Leave this as empty, if no additional filtering is required and you want to retrieve all groups from LDAP. Otherwise, ensure that the filter starts with '(' and ends with ')'.
guavus_iam_util_ldap_mapper_group_object_classes	"group"	Object class (or classes) of the group object. It is divided by comma if more classes are required. In a typical LDAP deployment, it can be 'groupOfNames' . In Active Directory, it is 'group'.

Variable	Sample Value	Description
guavus_iam_util_ldap_mapper_mode	"READ_ONLY"	Flag used to sync back the changes from your Guavus IAM to your LDAP. The default value is READ_ONLY. Omit this variable, if you want to continue with the default value.

2. Refer to the following table to set the LDAP Roles for Admin and Standard UI Application user:

   Note: This is applicable only if LDAP details have been provided in the previous sections.

Variable	Default Value	Description
dma_admin_role_ldap_group_name	This value to be provided by the user.	LDAP group for UI Admin Roles.
dma_standard_user_role_role_ldap_group_name	This value to be provided by the user.	LDAP group for UI Standard Roles.
add_default_keycloak_users_for_ldap	true	Set it false to restrict UI access for LDAP users only.

10. Appendix D: Creating Certificates

Refer to this section to determine the steps for creating and signing the certificate for both Ingress and minio.

Note: Don't add `alt_names` value, if the certificates are not used for minio other than Ingress.

Task 1 : Create the openssl Configuration File

Create `openssl.conf` file and add the following content in this file:

```
[req]
distinguished_name = req_distinguished_name
x509_extensions = v3_req
prompt = no

[req_distinguished_name]
C = <country_code>
ST = <state_code>
L = <location>
O = <org_name>
OU = <org_unit_name>
CN = <site_lb_vip_fqdn>

[v3_req]
subjectAltName = @alt_names

[alt_names]
IP.1 = <site_lb_vip_ipaddr>
DNS.1 = <site_lb_vip_fqdn>
DNS.2 = *.minio-svc.minio.svc.<kubernetes_dns_domain>
DNS.3 = *.minio.svc.<kubernetes_dns_domain>
```

Perform either of the following two tasks:

Task 2 : Generate Self Signed Certificate

Run the following command only if you want to generate self-signed certificate and its key:

```
openssl req -new -x509 -nodes -days <certificate_\
validity_period> -newkey rsa:4096 -keyout private\
.key -out public.crt -config openssl.conf
```

OR

Task 2 : Generate CSR and Signed by CA

Perform the following steps only in case, you wish not to use self-signed certificate and want to generate a CSR and get it signed by Certifying Authority (CA).

1. Run the following command to generate the certificate request:

   ```
   openssl req -new -nodes -days <certificate_\
   validity_period> -newkey rsa:4096 -keyout private\
   .key -out public.csr -config openssl.conf
   ```

2. Get signed certificate from CA and root CA certificate, intermediate certificates (if any) and signed certificate into a `.crt` file.

11. Appendix E: Encrypt and Edit vault.yml File

Non-encrypted `vault.yml` file is available with DMA product bundle, the user or operator must encrypt the file after updating the required variables.

To encrypt the `vault.yml` file, run the following commands on ansible controller or management node.

```
cd /data/dma

source .venv_ansible/bin/activate

ansible-vault encrypt inventory/<inventory-name>\
/group_vars/all/vault.yml
```

Enter the new password for ansible vault in the textbox.

To edit the default values of the variables in `vault.yml` file, run the following commands on ansible controller or management node.

```
cd /data/dma

source .venv_ansible/bin/activate

ansible-vault edit inventory/<inventory-name>/group\
_vars/all/vault.yml
```

Enter the password for ansible vault if prompted.

12. Appendix F: Parameters for CDR and DMC Data

12.1 Call Data Record (CDR) Data

CDR Dump data sent to the Service-IQ DMA software must contain the following attributes in the same position as mentioned in the table:

Input Field Position	Field Name	Mandatory	Description	Format	Sample Value
	Timestamp	Y	The time record generated in the network.	Standard time formats e.g., yyyyMMdd HHmmss	20150306 054550
2	IMSI	Y	International mobile subscriber identity.	First 6 characters must be decimal	236166453828376

Input Field Position	Field Name	Mandatory	Description	Format	Sample Value
	IMEISV	Y	International Mobile Equipment Identity including the Software Version Number.	First 8 characters must be decimal	3585450566759500

Input Field Position	Field Name	Mandatory	Description	Format	Sample Value
17	MSISDN	Y	Mobile Subscriber Integrated Services Digital Network number.	Alphanumeric characters	19437715841919
21	RAT	Y	Radio Access Type.	Integer value. See 3GPP ref. 29.061 version 17.0.0	0
5	SGSN IP Address	N	Serving gateway IP address.	IPv4 or IPv6 format	list of IPs. Example: (123.153.229.130,123.153.229.132,123.153.229.131)

Input Field Position	Field Name	Mandatory	Description	Format	Sample Value
	PGW TAC	N	Packet gateway Tracking Area Code of the PGW serving the device. Depending on the CDR source can be any identifier able to be mapped to a location.	Integer	12307

Input Field Position	Field Name	Mandatory	Description	Format	Sample Value
	ECI	Y	E-UTRAN Cell Identifier of the cell serving the device. Depending on the CDR source can be any identifier able to be mapped to a location.	Integer	26218616

Input Field Position	Field Name	Mandatory	Description	Format	Sample Value
24 [4]	dataVolumeFBCUplink	N	4th value from the list of data.		(100,,1,20150306 054550,34245,23423,00c400-0680,8-0-0-0-0-96,6778,17785,20150306 054627,,) - Uplink Volume in Bytes.
24 [5]	dataVolumeFBCDownlink	N	5th value from the list of data.		(100,,1,20150306 054550,34245,23423,00c400-0680,8-0-0-0-0-96,6778,17785,20150306 054627,,) - Downlink Volume in Bytes.

12.2 Device Management Centre (DMC) Data

DMC Dump data sent to the Service-IQ DMA software must contain the following attributes in the same position as mentioned in the table:

Input Field Position	Field Name	Mandatory Field Name
1	msisdn	Y
2	imei	Y
3	imsi	Y
4	last_detection	Y
5	msc_gt	Y

Input Field Position	Field Name	Mandatory Field Name
6	iccid	Y
7	last_configuration	Y
8	billing	N
9	brand_name	N
10	model_name	N
11	device_category	N
12	os_vendor	N
13	os_name	N
14	os_version	N
15	ota	N
16	data	N
17	weight	N
18	screen_size	N
19	screen_reso	N
20	mms	N
21	wap	N
22	wapversion	N
23	internet	N
24	email	N
25	gprs	N
26	edge	N

Input Field Position	Field Name	Mandatory Field Name
27	hsdpa	N
28	hsupa	N
29	java	N
30	camera	N
31	camera_reso	N
32	memorycard	N
33	gps	N
34	video	N
35	streaming	N
36	radiofm	N
37	audio_amr	N
38	audio_mp3	N
39	bluetooth	N
40	wlan	N
41	ptt	N
42	instmsg	N
43	syncmlds	N
44	omadm	N
45	2g	N
46	3g	N
47	3gp	N

Input Field Position	Field Name	Mandatory Field Name
48	lte	N
49	nfc	N
50	sim_form_factor	N
51	frequencies	N
52	number_of_sim	N
53	tac	N
54	display_touchscreen	N

The following is the sample for all field names:

```
+2560772475065;352419081093077;641102949593114;-;-;-;2017-02-20
17:29:26;UNKNOWN;TECNO;L8
LITE;5;Google;Android;5.1;1;1;0;0x0;0;1;1;2.0;1;0;1;1;0;0;0;1;0;0;0
;1;1;0;0;1;1;1;0;0;0;0;0;1;0;0;0;0;-;1;35241908;1
```

12.3 DMC Live Stream Data

DMC livestream data is live data received from DMC system in kafka topic. The data is received is JSON format.

The following is the sample for mandatory field names:

```
{"imei":"353074041595011","imsi":"642700000000263","msisdn":"+25645
1000000263","timestamp":"1617820463"}

{"imei":"354053042631824","imsi":"642550000001325","msisdn":"+25633
4000001325","timestamp":"1617820465"}

{"imei":"010912002119430","imsi":"642550000001326","msisdn":"+25633
4000001326","timestamp":"1617820465"}
```

```
{"imei":"010913007803386","imsi":"642550000001327","msisdn":"+25633
4000001327","timestamp":"1617820465"}

{"imei":"010914002591817","imsi":"642550000001328","msisdn":"+25633
4000001328","timestamp":"1617820465"}

{"imei":"010915009459035","imsi":"642550000001329","msisdn":"+25633
4000001329","timestamp":"1617820465"}

{"imei":"358864031086061","imsi":"642700000000265","msisdn":"+25645
1000000265","timestamp":"1617820465"}

{"imei":"863482033656379","imsi":"642550000001335","msisdn":"+25633
4000001335","timestamp":"1617820467"}

{"imei":"010922006251720","imsi":"642550000001336","msisdn":"+25633
4000001336","timestamp":"1617820467"}

{"imei":"010923001670737","imsi":"642550000001337","msisdn":"+25633
4000001337","timestamp":"1617820467"}

{"imei":"010924009201744","imsi":"642550000001338","msisdn":"+25633
4000001338","timestamp":"1617820467"}

{"imei":"010925002745709","imsi":"642550000001339","msisdn":"+25633
4000001339","timestamp":"1617820467"}

{"imei":"867387033219716","imsi":"642700000000267","msisdn":"+25645
1000000267","timestamp":"1617820467"}
```

12.4 Personally Identifiable Information Attributes Encryption

Service-IQ Device Management Analytics supports the encrypted data for Personally Identifiable Information (PII) attributes such as IMSI, MSISDN and IMEI.

Refer to the following table to determine the supported encryption:

Encryption	Output Bits	Encrypted Output Char Length
SHA-1	160 bits	41
SHA-256	256 bits	65
SHA-384	384 bits	97
SHA-512	512 bits	129

The data must be encrypted based on the following conditions:

Field Name	Conditions
MSISDN	The value for complete MSISDN must be encrypted.
IMSI	MCC and MNC must not be encrypted. • if separator is used - first seven digits must not be encrypted. • if separator is not used - first six digits must not be encrypted.
IMEI	The value must be encrypted only after skipping first eight digits containing TAC.

13. Appendix G: Data Ingestion Integration Requirements

Refer to this section to determine the integration requirements for Service-IQ DMA data ingestion.

1. **Supported Data Ingestion Types**
    - Thales Device Management Center (DMC)
        - Batch or streaming ingestion of DMC Automatic Device Detection (ADD) records. Built in integration between DMC and Service-IQ Device Management Analytics (Service-IQ DMA)
2. **EIR Automatic Device Detection (ADD) records ingestion**
    - Required fields: timestamp, IMSI, IMEISV, MSISDN
    - Configurable column order
    - Secure communication is recommended between the EIR vendor and Service-IQ DMA for ADD record exchange
        - Batch requirements
            - Service-IQ DMA User Interface: SFTP repository
            - Daily batches
            - Input format: CSV or TSV
            - Compression: Gzip
            - Header information is not required
        - Streaming requirements
            - Service-IQ DMA User Interface: Kafka Topic
            - Input format: JSON
            - Compression: Gzip
3. **Call Data Record (CDR) / Event Data Record (EDR) Ingestion**

- Same requirements for CDR and EDR ingestion
- Required fields: timestamp, IMSI, IMEISV, MSISDN, RAT_ID, SGSN_IP_ADDRESS, PGW_TAC, ECI
- Configurable column order
- Secure communication is recommended between the CDR source and Service-IQ DMA
    - Batch requirements
        - Service-IQ DMA User Interface: SFTP repository
        - 5-min batches
        - Input format: CSV or TSV
        - Compression: Gzip
        - Header information is not required.

14. Appendix H: SELINUX Guideline for Centos v7.x or RHEL v7.x

14.1 Set Up SELINUX

Verify and ensure that the status of selinux is set to disabled.

14.2 Set Up Packages

Ensure that:

- The `sshpass` package is installed on ansible-controller node.
- The `socat` package is installed on haproxy nodes.
- The following packages are installed on all the nodes:
    - python3
    - python3-pip
    - chrony
    - ntp
    - python3-libselinux
    - container-selinux
    - selinux-policy
    - libestr
    - libfastjson
    - make
    - rsyslog