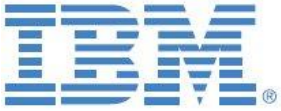


Accessing on-premises data from IBM Planning Analytics Cloud with *IBM Secure Gateway*:

What is it, how does it work, and how can it be configured?

V1.51

Last Updated:
November 2019

Created By:

Andreas Kugelmeier
Executive Consultant
Planning Analytics Architect
IBM Data and AI Expert Labs
Mobile Phone: +1-215-384-7302
Email: kugelmeier@us.ibm.com

Mike Bender
Solution Architect
IBM Analytics, Data Science, and Business Analytics

Notices & Disclaimers

Information in these presentations and papers (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS document is distributed "AS IS" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.
Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.
Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.
The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Document Version History

Date	Version	Author	Description
3/7/2016	0.9	Andreas Kugelmeier	1st Draft/Pre-Release Version
3/7/2016	0.95	Andreas Kugelmeier	2st Draft/Pre-Release Version
3/15/2016	0.96	Andreas Kugelmeier	3rd Draft/Pre-Release Version
3/18/2016	1.0	Andreas Kugelmeier	1st Version
3/31/2016	1.01	Andreas Kugelmeier	added information on TLS Version(s) and cipher suite
4/5/2016	1.02	Andreas Kugelmeier	Add Oracle Example
4/8/2016	1.03	Andreas Kugelmeier	Update Secure Gateway Client Installer Download Link
4/25/2016	1.04	Andreas Kugelmeier	Add material on Docker technology
5/17/2016	1.05	Andreas Kugelmeier	Addtl. information on ACL file requirements; addtl. information on Docker client
6/2/2016	1.1	Andreas Kugelmeier	Misc. updates related to SGW client version 1.5
8/8/2016	1.15	Andreas Kugelmeier	Add commentary on time needed to install new ODBC drivers in Cloud env.
2/3/2017	1.16	Andreas Kugelmeier	Edits to section on Firewall/Load-Balancer Node
3/6/2017	1.17	Andreas Kugelmeier	Update SGW architecture & data flow diagram
7/14/2017	1.18	Andreas Kugelmeier	Minor edits to section 4.4.2
11/26/2018	1.2	Andreas Kugelmeier	Misc. updates
2/22/2019	1.3	Andreas Kugelmeier	Add section on SGW vs VPN
04/02/2019	1.4	Andreas Kugelmeier	Update screenshots
04/04/2019	1.45	Andreas Kugelmeier	Update misc. IBM Cloud (Bluemix) SGW links
04/05/2019	1.46	Andreas Kugelmeier	Add SGW analogy graphics, misc. updates
04/05/2019	1.47	Andreas Kugelmeier	Add DeveloperWorks link
04/10/2019	1.48	Andreas Kugelmeier	Remove reference to welcome kit logon credentials for SGW administration (no longer needed with PAW Administration Panel
11/25/2019	1.5	Andreas Kugelmeier	Misc. updates re laod balancer / authentication server target URLs/Addresses and Proxy Configuration of SGW Client (authored by Mike Bender)
11/26/3019	1.51	Andreas Kugelmeier	Addl. Information re Proxy setup

Table of Contents

1. About this document

IBM Secure Gateway provides customers with a secure method to access on-premises or cloud data from within IBM Planning Analytics. This document introduces the IBM Secure Gateway technology and provides information on infrastructure requirements, configuration options, authentication & security as it pertains to IBM Planning Analytics.

2. Important Links

- Documentation for Secure Gateway in IBM Planning Analytics:
  https://www.ibm.com/support/knowledgecenter/en/SSD29G_2.0.0/com.ibm.swg.ba.cognos.tm1_prism_gs.2.0.0.doc/c_paw_administer_secure_gateway.html

- Secure Gateway (IBM Cloud / Bluemix):
  https://www.ibm.com/cloud/secure-gateway
  Online doc: https://cloud.ibm.com/docs/services/SecureGateway?topic=securegateway-getting-started-with-sg&locale=en#getting-started-with-sg
  API docs: https://cloud.ibm.com/apidocs/secure-gateway
  DeveloperWorks Recipe: https://developer.ibm.com/recipes/tutorials/how-to-integrate-ibm-secure-gateway-to-a-solution/

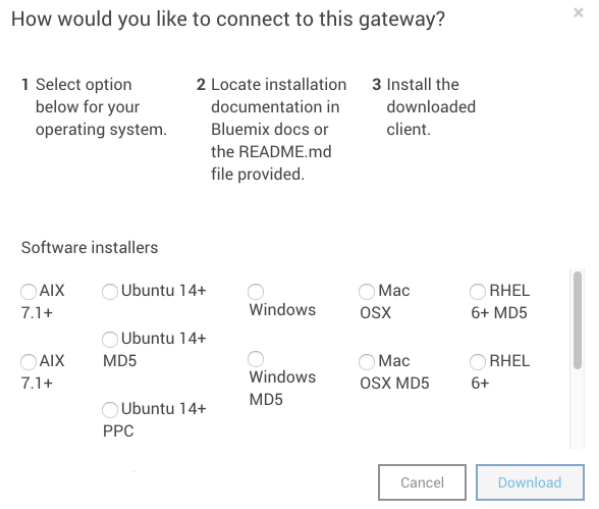
- Secure Gateway Clients:

  - IBM Secure Gateway Client**:**

    From the Planning Analytics Secure Gateway Panel (go to https://*<yourPlanningAnalyticsEnvironment>*.planning-analytics.ibmcloud.com/monitor/, then click on 'Secure Gateway), click the download button

    And follow the instructions on the screen:

    To directly download the Native Secure Gateway Clients you can also use the following URL:
    https://sgmanager.ng.bluemix.net/v1/getClientList

  - Docker (Windows, Mac OS X, most Linux flavours/platforms):
    https://docs.docker.com/engine/installation/#installation

For windows, see https://docs.docker.com/windows/step_one/ for instructions and https://www.docker.com/products/docker-toolbox to download

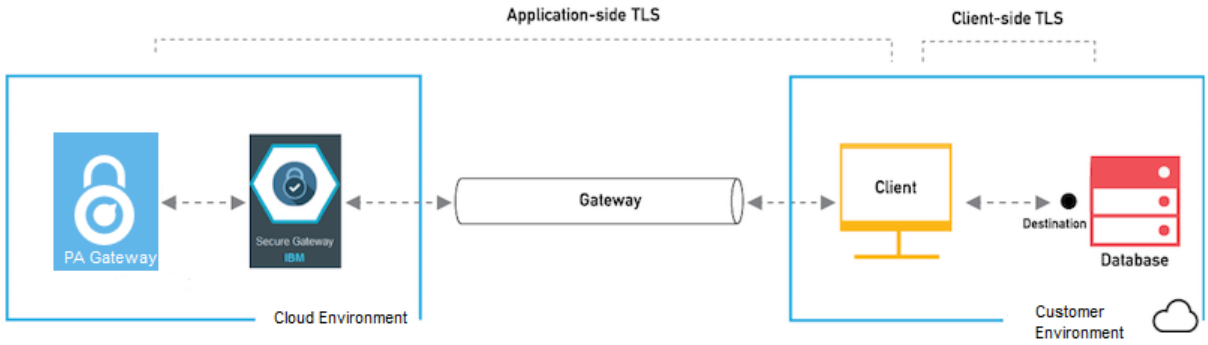
3. Secure Gateway – Introduction

3.1 What Is It?

Secure Gateway – via its underlying tunnelling protocol - allows providing a network service that the underlying network does not or should not support or provide directly. The secure gateway tunnel hence can be thought of as a secure tunnel that connects networks without requiring the networks to open their source and destination addresses and protocols. Only the secure gateway tunnel needs to be open.
The Secure Gateway service hence provides you with a secure method to access your on-premises or cloud data from your IBM Planning Analytics application through a secure passageway (a secure tunnel).

3.2 How does it work?

The Secure Gateway Service works by using a *Secure Gateway Client* to establish a secure tunnel to a *Secure Gateway Service* on the IBM Bluemix cloud from where the IBM Planning Analytics cloud environment (on IBM Cloud) can be accessed.



The secure gateway tunnel is established via a secure tunneling protocol connection similar to SSH[1]. The tunneling protocol provides an encrypted shell for information transfer via the gateway tunnel, such that even unencrypted data can travel in safety because the data is re-packaged and sent over the encrypted shell through the tunnel.
In other words: via the secure gateway shell, other protocols such as TCP, TLS, HTTP or HTTPS are 'tunneled' from the gateway service on the cloud to the destination (the gateway client on-premises). Once the packets reach the end of the tunnel (they are 'on-premises'), they are 'unpacked' to their original protocol and will 'proceed' as per their original configuration.

The gateway tunnel connection (the tunnel) needs to be opened by the on-premises gateway client (it cannot be opened by the gateway service itself). At the same time, while the gateway client opens the connection, it cannot request information from the cloud via the gateway service connection: The gateway service connection is uni-directional in the sense that all requests have to be initiated by the IBM Planning Analytics (Cloud) environment. This means the cloud environment – similar to a database client - has to initiate the request against the on-premises system. Requests can be of types READ/WRITE of course, but they have to be initiated by the Cloud environment.[2]

[1] https://en.wikipedia.org/wiki/Tunneling_protocol#Secure_Shell_tunneling
[2] The IBM Secure Gateway Client itself as of version 1.4.0 does support bi-directional connectivity. For more information, see https://developer.ibm.com/bluemix/2015/12/15/ibm-secure-gateway-updates-version-140/. Yet note that bi-directional connectivity is at this time not enabled for Planning Analytics, providing an additional safety measure in that the Planning Analytics environment cannot be queried via the secure gateway.

3.3 How is a Secure Gateway different from a VPN?

https://developer.ibm.com/recipes/tutorials/ibm-cloud-secure-gateway-service-vs-vpns-whats-the-difference/

4. IBM Secure Gateway Specs & Facts

4.1 Supported Protocols

The IBM Secure Gateway tunnel supports the following protocols for the cloud to destination data connection: TCP, HTTP, HTTPS, TLS[3]. See <Security: Authentication & Encryption> below for more information on TLS.

4.2 Tunneling Protocol

The secure gateway tunnel is established via a secure tunnelling protocol connection similar to SSH in that it is established from one side, is encrypted, is long-lived, and can flow data both up and down stream. The specific technology is a TLS secure websocket that starts as an HTTPS connection and gets upgraded to a WSS connection. This results in a long-lived, bi-directional tunnel that is secured using the same TLS layer protocol as an HTTPS connection.[4]

4.3 Secure Gateway Clients

4.3.1 Client Options

Secure Gateway client installers are available for Linux, Windows & Mac OS. Options are
a) Native IBM Secure Gateway Client
b) Docker Client (which runs the IBM Secure Gateway Client in a Docker Container[5])
c) DataPower Client
See <Important Links> for installation information and download links.

4.3.2 Differences

The IBM Secure Gateway native installation package has slightly better performance as Docker virtualizes the container. Both packages can run several clients. The Docker client on the other hand updates more easily as it will download the Docker IBM Secure Gateway package on start-up.
The DataPower option is an appliance optimized solution (for WebSphere SOA Data Power Appliances) with the same base features as the Docker client but with more security enforcements.

---

[3] https://en.wikipedia.org/wiki/Transport_Layer_Security

[4] IBM Secure Gateway TLS is using the native node processes, which uses OpenSSL. OpenSSL supports up to TLS 1.3. Encryption algorithms are using the following corresponding default cipher suite:
ECDHE-RSA-AES128-GCM-SHA256:
ECDHE-ECDSA-AES128-GCM-SHA256:
ECDHE-RSA-AES256-GCM-SHA384:
ECDHE-ECDSA-AES256-GCM-SHA384:
DHE-RSA-AES128-GCM-SHA256:
ECDHE-RSA-AES128-SHA256:
DHE-RSA-AES128-SHA256:
ECDHE-RSA-AES256-SHA384:
DHE-RSA-AES256-SHA384:
ECDHE-RSA-AES256-SHA256:
DHE-RSA-AES256-SHA256:
HIGH:
!aNULL:
!eNULL:
!EXPORT:
!DES:
!RC4:
!MD5:
!PSK:
!SRP:
!CAMELLIA

[5] See https://www.docker.com/what-docker. The Docker Container includes the IBM Secure Gateway Client software application and all of its dependencies. Compared to a VM, a Docker Container does not have a 'Guest OS', but would share its Docker kernel with other containers that may be running on the same computer. Docker Containers run as an isolated process in userspace on the host operating system and will run on any computer.

4.4 Security: Authentication & Encryption

4.4.1 Authentication

The gateway service can be configured to generate a JSON Web Token (JWT) and to then require the client to use this security token as an authentication mechanism.[6] Tokens can be set to expire, i.e. to have to be renewed. The token expiration timeframe can be specified in days.

4.4.2 Encryption

The data travelling through the Secure Gateway tunnel is 'wrapped' in an encrypted secure shell, i.e. the tunnel itself is encrypted and hence protected. The specific technology is a TLS secure websocket that starts as an HTTPS connection and gets upgraded to a WSS connection. This results in a long-lived, bi-directional tunnel that is secured using the same TLS layer protocol as an HTTPS connection.

In the Future, optional additional protocol, authentication and encryption methods for the Planning Analytics Secure Gateway will be HTTPS or TCP over TLS[7]:

A) Client-Side TLS to secure the on-premises connection between the database and Gateway Client:

- **TLS Client-Side Authentication**: In IBM Cloud, will allow the customer to upload certificates if desired. Note that certificates would not have to be uploaded if the TLS certificate is not self-signed, in which case select 'Enable Client TLS' on the data source destination and then any connection to the data source destination succeeds once the destination's certificate is verified against known certificate authorities.

B) Application-Side TLS to secure the connection between Gateway Client and Planning Analytics:

- **TLS Server-Side authentication:** Only the server needs to present its certificate
- **TLS Mutual Authentication**: Both Server and Client need to present their certificates. TLS Mutual Authentication is the best choice for any connections that handle sensitive data transfers. In IBM Cloud, Certificates and keys can be generated by the Gateway Service by checking the "Auto generate cert and private key" box. The Certificate can them be downloaded to be used by the Client. Alternatively, one can upload their own certificates.

C) HTTPS:

- **HTTPS:** Only the server needs to present its certificate
- **HTTPS Mutual Authentication**: Both Server and Client need to present their certificates.

Note that TLS and HTTPS are supported by the IBM Cloud Secure Gateway technology, the corresponding configuration for **TLS and HTTPS to upload certificates is currently <u>not yet</u> supported by the Planning Analytics Secure Gateway Configuration UI. The Planning Analytics Secure Gateway will support such TLS and HTTPS configuration in the future.** Once available, corresponding documentation on UI-related configuration options and examples will be provided in an update to this document.

[6] See https://jwt.io/introduction/ & https://en.wikipedia.org/wiki/JSON_Web_Token for additional information on JWTs.
[7] https://console.bluemix.net/docs/services/SecureGateway/securegateway_destination.html#adding-a-destination

4.5 Availability, Scalability & Load Balancing

4.5.1 Availability, Scalability & Load Balancing: The Secure Gateway Service on IBM Cloud

The IBM Secure Gateway Service Platform consists of a **multi-tenant, highly available and load balanced bare metal cluster of several static servers that share the hosting of all the gateways**. Each gateway destination (=endpoint) gets a unique host/port combination from one of the available servers, and - using mutual authentication - prevents any other applications from utilizing that connection. The IP addresses of the secure gateway servers do not change, even in case of failover.

4.5.2 Availability & Scalability: The Secure Gateway Client

As of release 1.4, the **Secure Gateway Client supports High Availability**: Multiple gateway clients can connect to a single gateway by starting them using the same gateway ID. The connection to these clients is load balanced by using round robin to prevent a single point of failure in the connections. If one of the clients goes down, you'll still be able to connect to the on-premises resource through another client that is still connected. Assuming you have two separate gateway clients, each of them on its own machine. To get the round robin working, all you need is to start the client on each machine (using the same command)

As of release 1.4, the secure gateway client can run in so-called 'multi-gateway' mode, allowing multiple gateway connections to be created from a single client instance rather than requiring a new client for each gateway connection, thereby improving usability and configurability. An example of how to run the client in multi-gateway mode with Planning Analytics is shown in section <Multi-Gateway Mode>.

The client HW requirements are strongly related to the number of concurrent connections that the client will be handling. Also, depending on data volume, memory usage for buffering can grow relatively high on a single client if it's handling a large amount of throughput and a lot of connections. CPU utilization will be higher when using TLS encryption. Examples: A 2 CPU - 4GB machine would be sufficient for one or two client instances with moderate traffic (concurrency). A 4CPU - 8GB machine would be recommended for hosting three clients with heavier traffic.

Note that in a Planning Analytics Context, concurrency on the Secure Gateway tends to be relatively low, as end-users typically will not often interact directly with an on-premises database.

4.6 Communication Ports & Network/Firewall Configuration Requirements

The Secure Gateway Client uses outbound ports 443 (SSL) & 9000 to connect to the IBM Cloud environment.

On client start-up, the

- first call goes to the DataPower proxy load-balancer on port 443 for authentication. Outbound target:
    - For SG client v180fp9 and former
        - US South: sgmanager.ng.bluemix.net
        - US East: sgmanager.us-east.bluemix.net
        - United Kingdom: sgmanager.eu-gb.bluemix.net
        - Germany: sgmanager.eu-de.bluemix.net
        - Sydney: sgmanager.au-syd.bluemix.net
    - For SG client v181 and later
        - US South: sgmanager.us-south.securegateway.cloud.ibm.com
        - US East: sgmanager.us-east.securegateway.cloud.ibm.com
        - United Kingdom: sgmanager.eu-gb.securegateway.cloud.ibm.com
        - Germany: sgmanager.eu-de.securegateway.cloud.ibm.com
        - Sydney: sgmanager.au-syd.securegateway.cloud.ibm.com

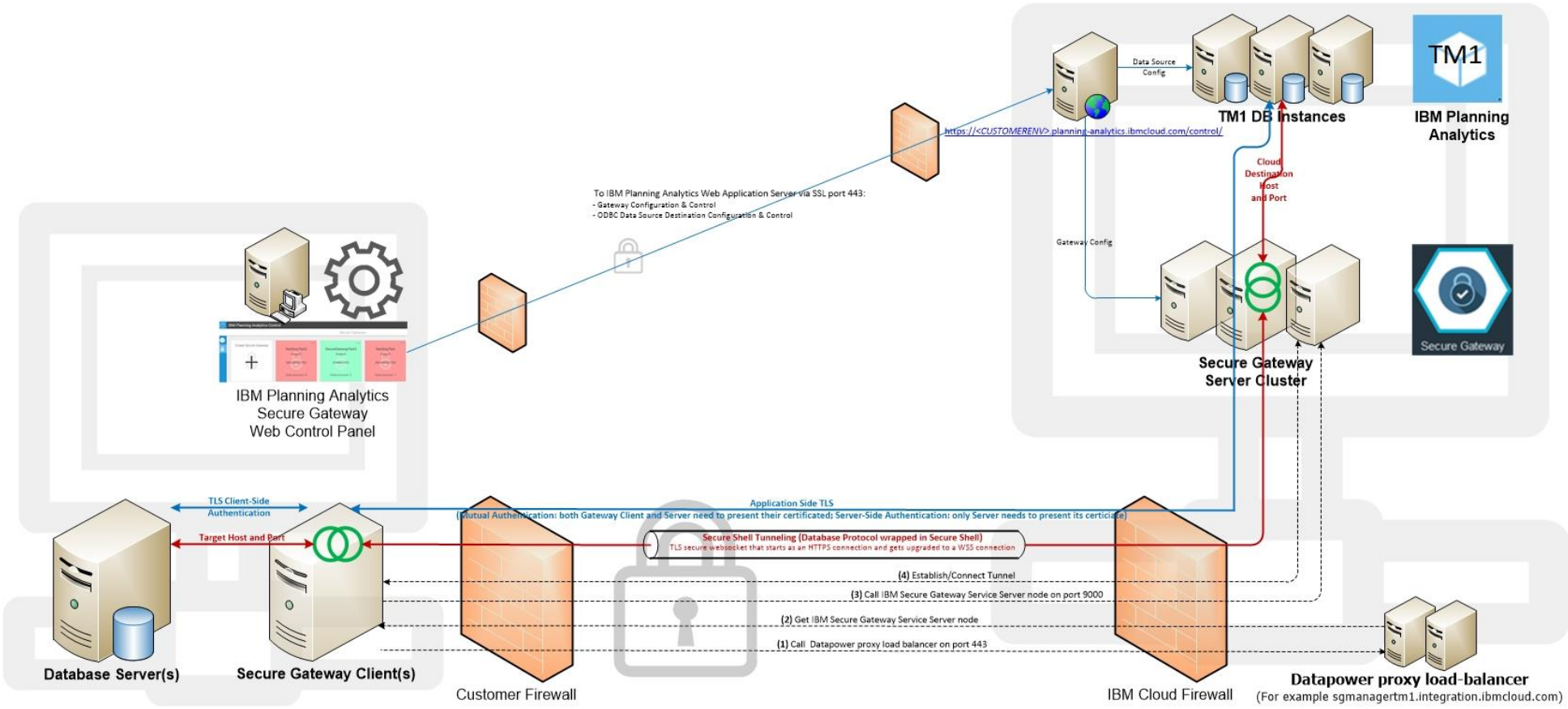
- The second call goes directly to one of the IBM Secure Gateway Service Server nodes on port 9000. The IP addresses for those Secure Gateway Servers are not published, but are available upon request if needed (for firewall configuration).

The tunnel connection between the Secure Gateway Client and the Secure Gateway Service does send ping-pong messages back and forth. If disconnected, the client will automatically attempt to reconnect to the server every 5 seconds.

4.7 Provisioning, Installation & Configuration Process

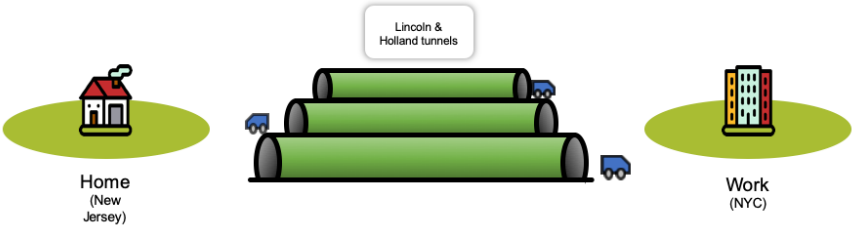
1. **Gateway Service**: Create one or more Gateway Services by using your IBM Planning Analytics Control Panel. The Gateway Services are thereby provisioned for you on the IBM Cloud. Provisioning a Secure Gateway Service in the IBM Planning Analytics Control Panel will provide you with the gateway unique gateway ID to use when connecting a Secure Gateway Client.
2. **Gateway Client Installation**: install the gateway client, which can be a Docker container (with Docker Engine), a virtual DataPower container/engine, or IBM Secure Gateway native clients for Linux, Mac & Windows.
3. **Start the Gateway Client**
4. **Establish Gateway tunnel connection**: using the gateway ID and security authentication token, establish the secure gateway tunnel connection from the client.
5. **Create Gateway Destinations**: Once you have created the gateway service, you need to configure gateway destinations (data sources). The Gateway Cloud destinations (hostname/IP + port) get created when creating a new data source. Each destination (=endpoint) created in the Secure Gateway service will have a unique cloud host and Port. This unique cloud host is mapped to a unique IP address that is independent from the application itself. This IP address is static and can be filtered at the firewall level. Authentication and Data Encryption is configured per Gateway Destination.
6. **Gateway Client Configuration**: Configure the gateway client(s) as needed/desired, configuring access control by using access control lists etc.
7. Test the gateway destinations/data sources

4.8 Secure Gateway Architecture and Data Flow Diagram



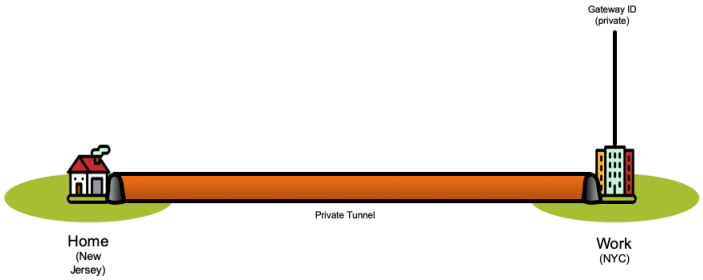
5. Secure Gateway Analogy: Secured & Ultra-Private Traffic Tunnels

Let's say you work in NYC and live in New Jersey. Unfortunately, you have to commute by car:

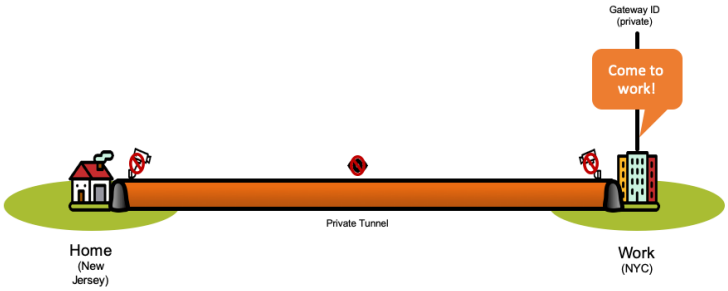


Now, in order to alleviate traffic congestion in the Lincoln & Holland tunnels the Port Authority of New York and New Jersey are offering a futuristic, new tunneling service, called the Gateway Service, connecting NYC (the cloud) with New Jersey (on-premises).

As a New York and New Jersey resident you would get your own private tunnel. Tunnels are created as if by magic (they do not need to be dug) in that they are established almost immediately. It's like platform 9¾ at King's Cross, but you do not have to run into a wall. The port authority is customer-friendly.



That's pretty nice you would say, I'll never be stuck in Lincoln Tunnel traffic again! Well, it gets even better: each private tunnel is configured to lead straight home, connecting your company's NYC car garage (fortunately, your company provides free parking) to the garage of your home in New Jersey. In other words, you 'tunnel right-through' from your apartment building to your garage, privately and unseen by other vehicles & drivers. It follows that with this tunnel you never have to worry about inner city traffic, freeways, dangerous neighborhoods and so forth.[8] And you do not even have to drive yourself: The tunnel itself provides its own transport vehicle (it has a rather strange name in that it is named 'the secure shell', but it is more comfortable than it suggests):



Once in the tunnel and for as long as being in the tunnel, vehicles of different kinds and with different cargo are transported by and within the tunnel's transport vehicle. The tunnel's transport vehicle supports the commonly used vehicle kinds (trucks, limos with tinted windows, protected motorcades, you name it). The port authority gives them cryptic names like TCP, HTTP, HTTPS, TLS.

[8] For the sake of maintaining the integrity of our analogy, let us just assume that unlimited parking is available in your garage as well as for your NYC apartment.

Each tunnel has its own unique address on the NYC side. They call it the gateway ID. Tunnel addresses are not shared and not made public. To actually use the tunnel and get to a room within your house, you need to provide additional information: Each room needs to be mapped via a distinct route, using a distinct and specific call name (the cloud destination host name and port) and secured individually. The dispatcher (some call him the Turbo-Integrator or TI - he closely works with a guy referred to as the ODBC driver[9], & together they make up the dispatcher team) on the NYC side only gets to know the tunnel- & destination-specific call name; it does not need to get to know the final destination's name (the room). The final destination (the room) is conveyed to the vehicle once the vehicle is on the New Jersey side, i.e. in the house where the destination address 'Living Room' makes sense and is unique (the Port authority has some funky names for their living rooms , they are given names like 192.168.1.15). To put it differently, the dispatcher (that TI guy with the driver support) thinks he speaks to destination Andy:12345. But Andy:12345 is only an intermediary call name (for a destination on the NYC side) that at the end of the tunnel (in the NJ Garage) will be mapped to room destination 129.168.1.15:5000 for example.

You may use one tunnel to access multiple rooms (destinations). Or you may use multiple tunnels to multiple rooms. It depends on how you would like to monitor/handle and govern traffic to your house. If for example you would like to be able to shut down traffic to one room in particular, it may be more useful to create a separate tunnel for that room.

The Port Authority takes your security and safety very seriously and furthermore provides several options for you to increase security and safety:

- If a tunnel collapses, other tunnels are on standby to immediately fill in. The new tunnel will take on the address of the old tunnel. Traffic gets re-routed immediately.
- You can secure your tunnel by requiring vehicles to identify themselves via a private token (the Port Authority calls it the JSON Web Token). If you do not have the token key, forget about using the tunnel. If you easily forget your keys, keep the tokens at home in a secure place or – if you did not do that - contact your tunnel service to issue you a new key. The keys can be set up to automatically stop functioning (to expire) in which case you have to ask the tunnel service to issue you a new key.
- Each tunnel is uni-directional, meaning traffic requests must be dispatched from NYC (the cloud) only. In other words: you cannot use the tunnel to drive to NYC without being invited (called on). You can also not use the tunnel network to try to break into a tunnel towards NYC to enter someone's NYC apartment. You have to be invited by the folks in your NYC apartment to use the tunnel (well, you could invite yourself too). NYC must initiate the request for information/traffic. Traffic can go both ways, i.e. NYC may request a packet to be delivered from home and NYC may send something back upon request of the packet. But tunnel-use requests cannot be made from home.
- You can explicitly block traffic from reaching a particular room or allow traffic to reach it. This is done via so-called Access Control Lists (ACL, not related to Anterior Cruciate Ligaments). By default, a tunnel connection is blocked at the garage door and not allowed to enter any room. You can block and/or allow rooms per tunnel connection, i.e. you can use a particular tunnel only for a particular room etc.
- One cannot put a camera inside the tunnel to see what and who is travelling. The tunnel itself is secured using camouflage (encryption technology); the vehicles and vehicle content that are being tunneled will be made unrecognizable while travelling through the tunnel.
- In case you are concerned about being seen while travelling to and from the tunnel: You may have your tunnels require the use of additional camouflage (encryption technology) to disguise the vehicle content entering and exiting the tunnel.

[9] Not to be confused with The Driver.

6. Secure Gateway Example: Connecting to SQL Server, DB2, Oracle

Now let's apply what we learned about Secure Gateway to setting up an ODBC database connection between the cloud and an on-premises database: Let's assume you have a SQL Server database on your premises. The Database is hosted on <mySQLserver>.<yourcompany>.com, port 49244. You would like to connect your TM1 Server on Planning Analytics with this database. You also have a DB2 Database. It is hosted on the same server (in our example case) and runs on port 50000. In the following examples, we will use the Database Server's IP address, but the machine name can also be used. The IP address of the machine in our example is 192.168.1.15.

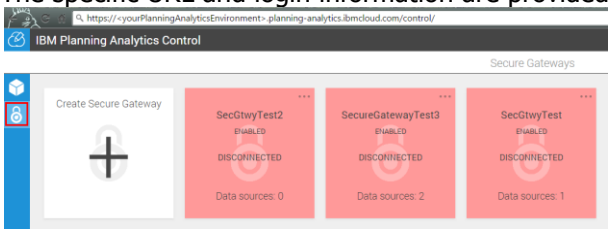
6.1 Configure and Test ODBC access on premises

Test ODBC access: On a client machine within your network domain (on-premises), ensure you can access the SQL Server and DB2 Server databases via ODBC, using the desired network protocol and authentication mechanism. This step is important. The secure gateway ODBC connection will only work if the destination server accepts the ODBC connection and if you provide it with the proper parameters. Do not test ODBC on the cloud, test it on premises first.

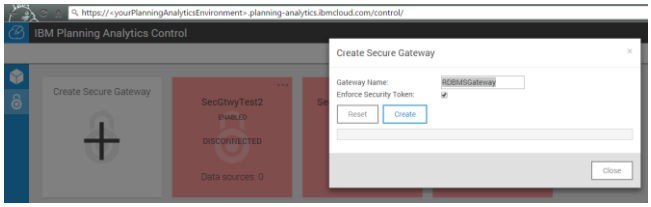
6.2 Create the Gateway Service

6.2.1 Earlier PA Cloud releases

Create a Secure Gateway Service: Now that you have tested and validated ODBC access, you create a secure gateway service. You do so via Planning Analytics control, which is at
https://<yourPlanningAnalyticsEnvironment>.planning-analytics.ibmcloud.com/control/
The specific URL and login information are provided with your welcome kit.[10]



Once you have logged into the control panel, click the lock icon (the icon below the cube icon) and click + to create a new gateway. Let's call it RDBMSGateway. We will have the gateway enforce a security token, meaning we will want the Gateway clients to authenticate themselves with a security token generated by the Gateway service. This way, no unauthorized client can connect to the gateway:



Click 'Create'. The information in the grey window is important:

[10] For the user name and pw, look for the following information in your welcome kit:

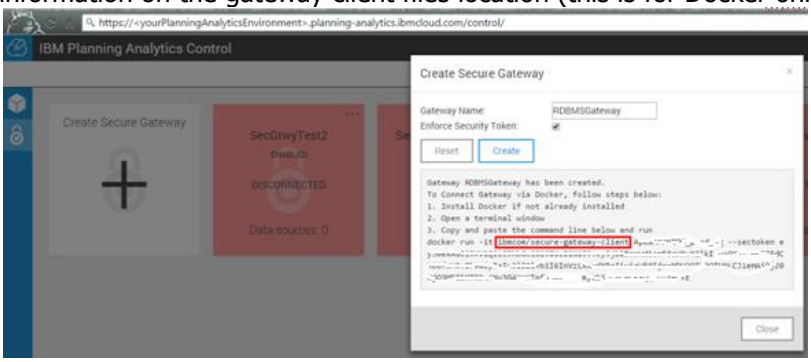
Use the following login credentials to control TM1 services:
User name *Password*
control ***<Password>***

For the URL for the Planning Analytics control panel, look for the following information in your welcome kit:

You can stop and start TM1 services and configure your connection to on-premises ODBC data sources using this web application.
https://<yourPlanningAnalyticsEnvironment>.planning-analytics.ibmcloud.com/control/

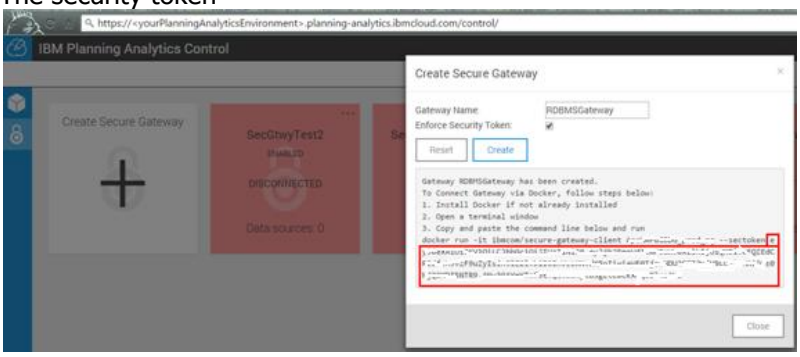
It contains

- information on the gateway client files location (this is for Docker only):



- The gateway ID:

- The security token



If you use Docker as the 'Client' platform to run the secure gateway client, you can just copy and paste the command

docker run -it ibmcom/secure-gateway-client *<GatewayID>* --sectoken *<JWT>*

into a Docker command line window. If you do use the native IBM Secure Gateway Client to run the Secure Gateway client, all you will need is the gateway ID and the security token. We will cover this later in the gateway client section <Secure Gateway Client via Docker>.

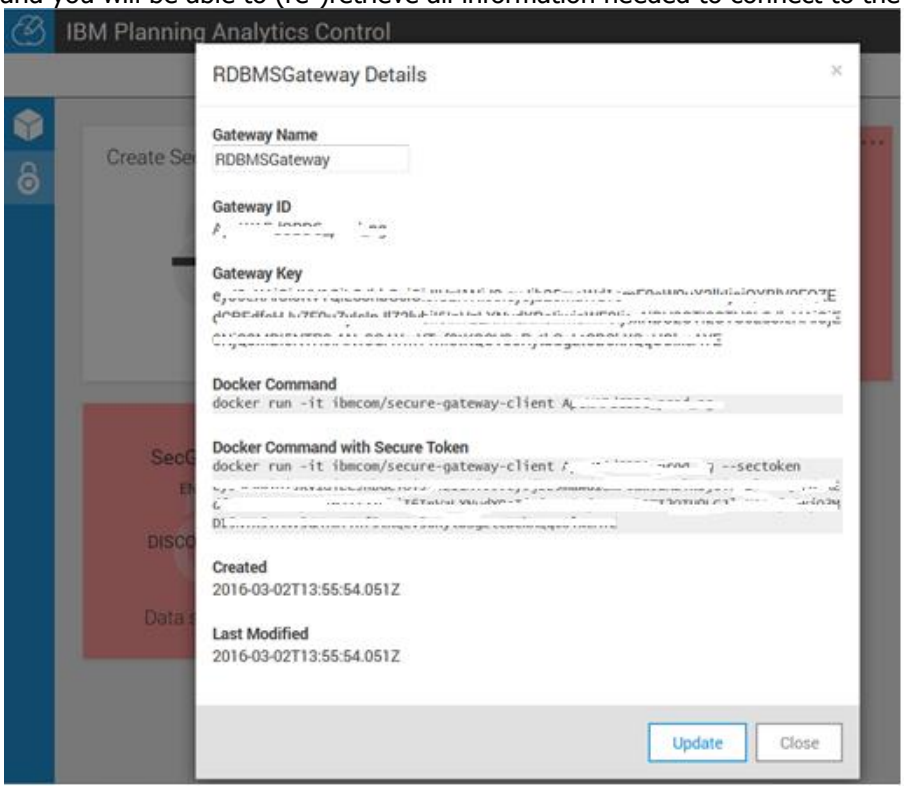
In any case, at this time, for now just copy this text into your clipboard:

Then click 'close' to exit the window.

About the data in the clipboard: Don't worry about accidentally overwriting/losing it, because you can always get it back: click on the ••• at the top right of the gateway tile:

then click on 'Edit/View':

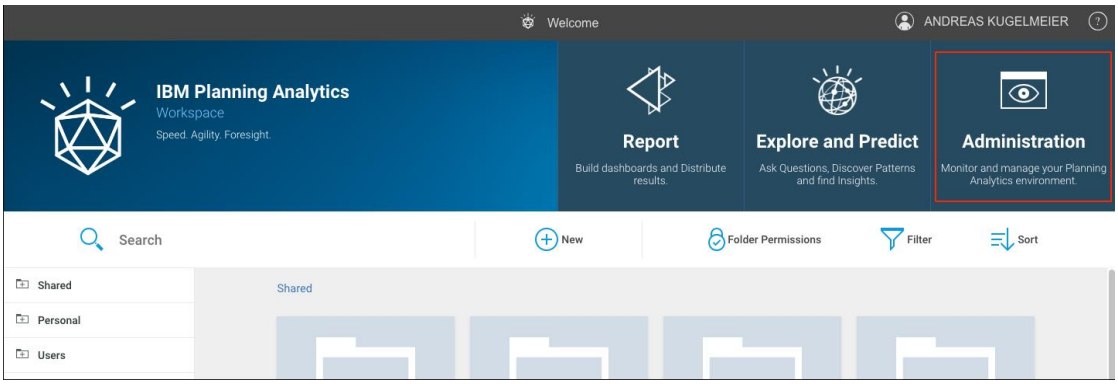
and you will be able to (re-)retrieve all information needed to connect to the gateway service:



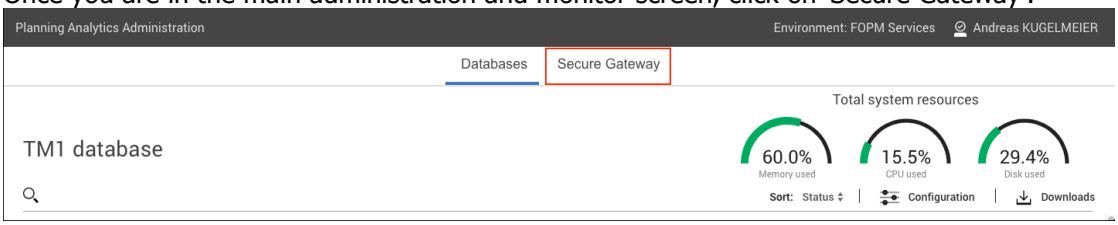
6.2.2 Current PA Cloud Release

Create a Secure Gateway Service: Now that you have tested and validated ODBC access, you create a secure gateway service.

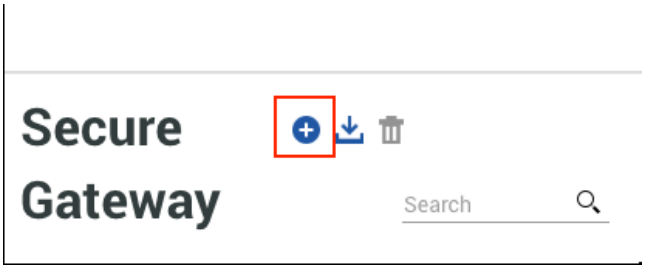
You do so via Planning Analytics Administration which is at
https:// *<yourPlanningAnalyticsEnvironment>*.**planning-analytics.ibmcloud.com/monitor/**
You can also go to
https:// *<yourPlanningAnalyticsEnvironment>*.**planning-analytics.ibmcloud.com**
and then click on 'Administration':



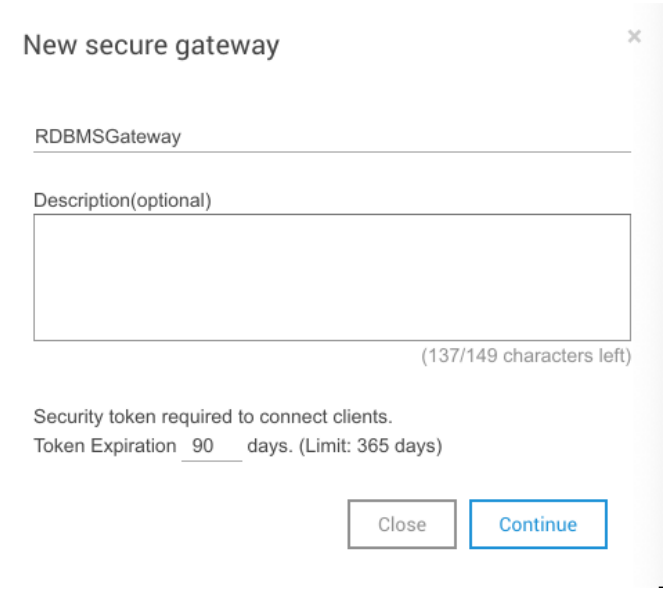
Once you are in the main administration and monitor screen, click on 'Secure Gateway':



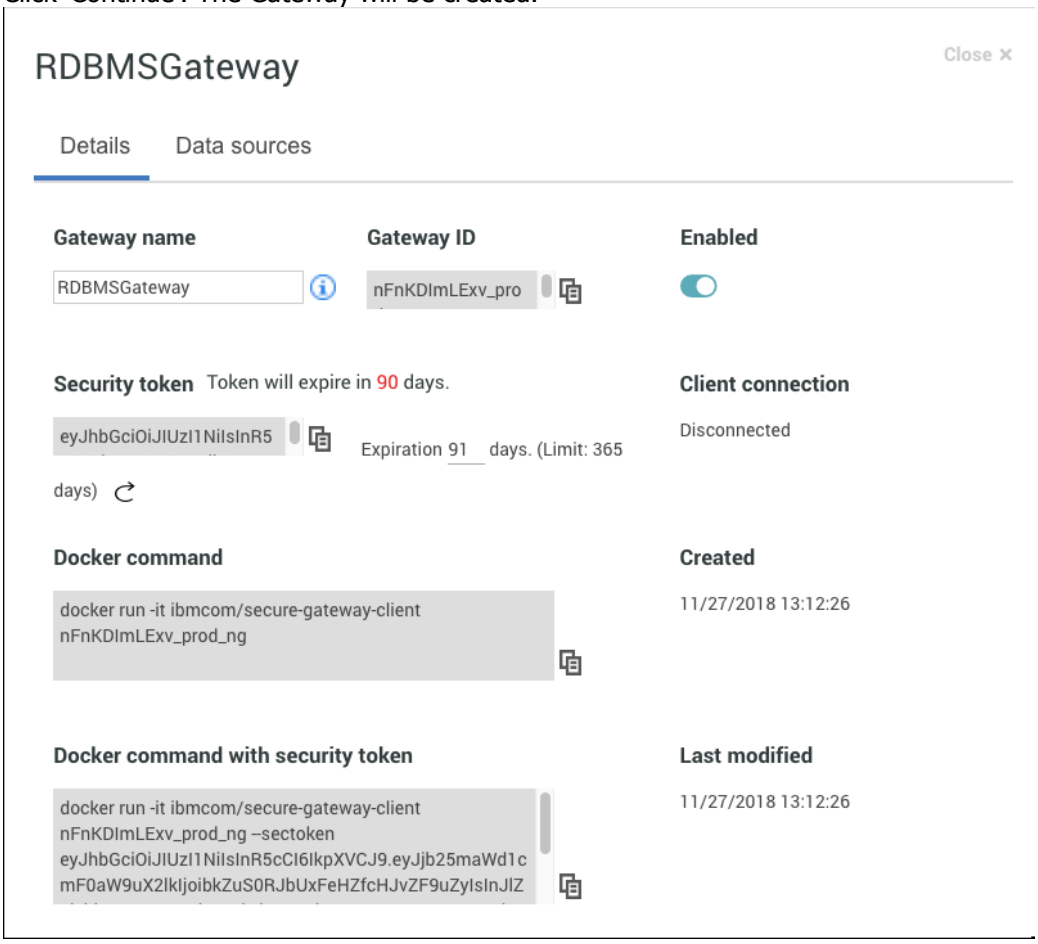
Once you are in the Secure Gateway panel, click + to create a new gateway.



Let's call it RDBMSGateway. We will have the gateway enforce a security token, meaning we will want the Gateway clients to authenticate themselves with a security token generated by the Gateway service. This way, no unauthorized client can connect to the gateway:

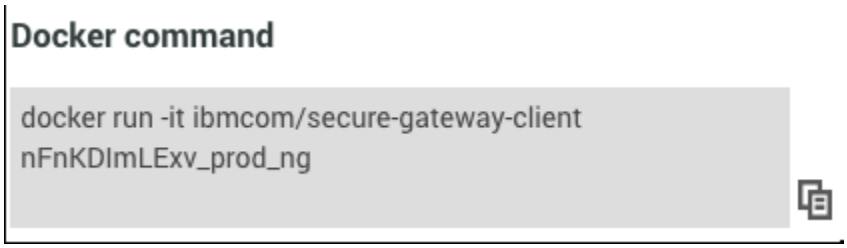


Click 'Continue'. The Gateway will be created.

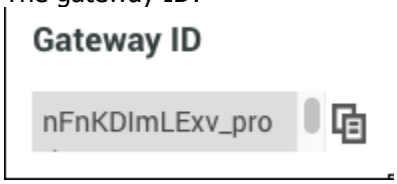


The information in the grey windows is important: It contains

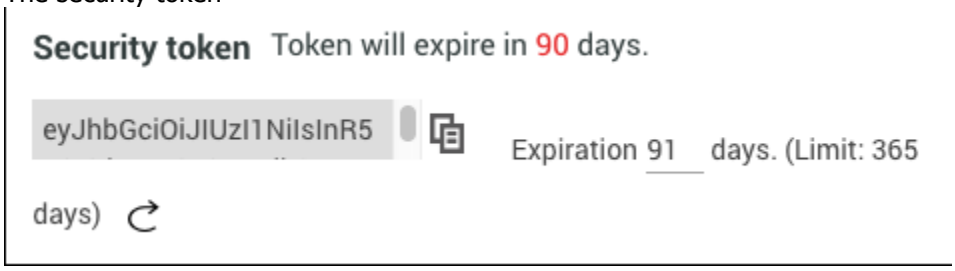
- information on the gateway client files location (this is for Docker only):



- The gateway ID:



- The security token

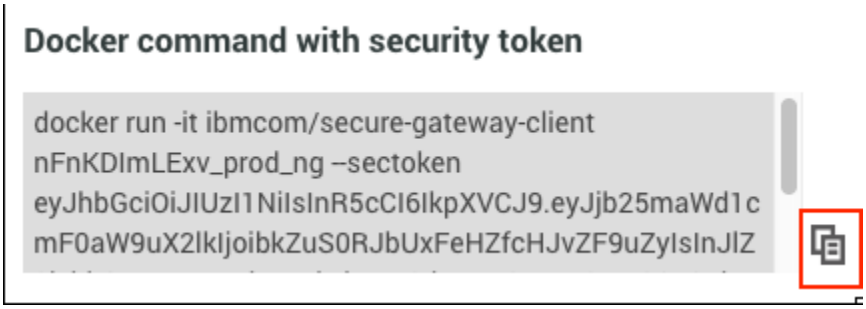


If you use Docker as the 'Client' platform to run the secure gateway client, you can just copy and paste the command

docker run -it ibmcom/secure-gateway-client *<GatewayID>* --sectoken *<JWT>*

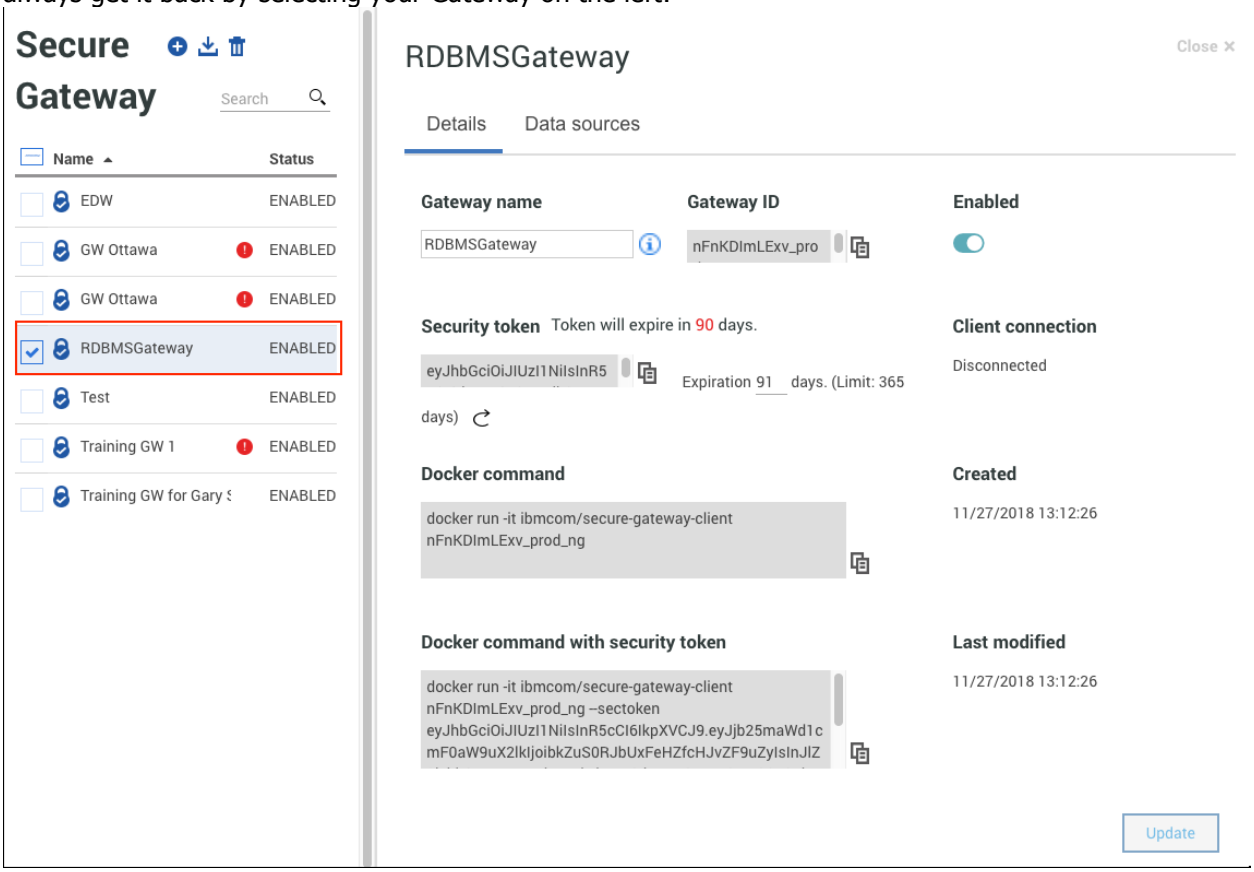
into a Docker command line window. If you do use the native IBM Secure Gateway Client to run the Secure Gateway client, all you will need is the gateway ID and the security token. We will cover this later in the gateway client section <Secure Gateway Client via Docker>.

In any case, at this time, for now just copy this text into your clipboard by clicking the copy button:



Then click 'close' to exit the window.

About the data in the clipboard: Don't worry about accidentally overwriting/losing it, because you can always get it back by selecting your Gateway on the left:



6.3 Install and configure the secure gateway client

In this example, we will be covering two options for secure gateway client installation:

a) Native IBM Secure Gateway client
b) IBM Secure Gateway client via Docker container

Note that all clients will run/use the same IBM Secure Gateway client software technology to connect and run the client. The difference between the clients lies in how the client gets initiated. Docker will retrieve (and on startup, run) the IBM client from the web:

Docker Command with Secure Token

```
docker run -it ibmcom/secure-gateway-client [illegible] --sectoken
```

With the native client installer, the client is physically installed on the on-premises machine.

Configuration options and methods are primarily discussed under the section <Native IBM Secure Gateway Client> (right below). This is because the Docker client essentially runs the same client commands and provides analog command line and configuration capabilities.

Note that during installation, Port 443 (default SSL) needs to be open for outgoing communication in order to facilitate npm installation: During the installation, the installer will connect to npm registry and run npm install to install the dependencies required by Secure Gateway Client. Before the installation, please make sure the machine which the client will be installed on can connect to a npm registry website. npm is configured to use npm, Inc.'s public registry at https://registry.npmjs.org by default. If there is npm Enterprise server in your environment, please whitelist all of the dependencies of Secure Gateway Client on the npm Enterprise server. For the list of dependencies, please refer to <Installation_directory>\ibm\securegateway\client\package.json file.

To validated that the machine that is hosting the Secure Gateway client can connect to the npm registry "ping registry.npmjs.org". If this command fails means that you don't have access to internet or your firewall rules are blocking access to this registry. You may need a proxy server to gain access:

6.3.1 Proxy Setup

If a proxy is required for the install, open a command prompt *after initial install* and navigate to the nodex.x.x directory (as an example C:\Program Files (x86)\Secure Gateway Client\ibm\node6.9.4). Run the following command to set npm to route through proxys:

```
npm config set proxy http://url:port
```

or

```
npm config set proxy http://username:password@url:port
```

Then -re-run the client installer. This will let npm pull the files it needs through the proxy. Now, edit the startup config file -%Installation_directory%/ibm/securegateway/client/securegw_service.config:

```
SECGW_ARGS="-xhttp://proxyserverip:proxyport—service"
http_proxy=http://proxyserverip:proxyport
```

Note: 1stline is for the CAP* servers via outbound port 9000. These are the only options you should need. In the SECGW Args, ignore the examples found in online docs. 2ndline is used for connecting to sgmanager serverviaoutbound port443.

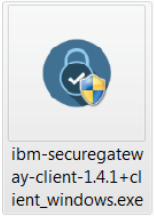
Ensure the SGW service is NOT running and start the SGW client to test. Example for output in case of a correct connection:

6.3.2 Native IBM Secure Gateway Client

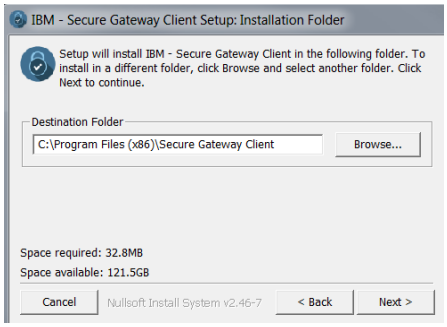
6.3.2.1 Installation

Download your client as per the links in section <Important Links> above. Following is a sample installation on Windows,

1. Run the installer

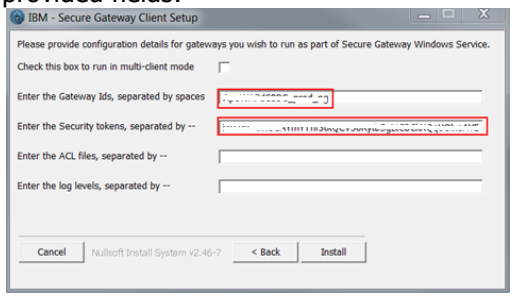


2. Pick an installation folder:



3. Check the box to run the client as a windows service (recommended) and have it restart automatically:

4. In the next screen, you may (you do not have to) provide gateway service connection & (destination) configuration information. You can paste the Gateway ID and token into the provided fields:



   Or: leave the fields empty and configure after the installation (see <Startup & Configuration> below).

5. Click install

6.3.2.2 Startup & Configuration

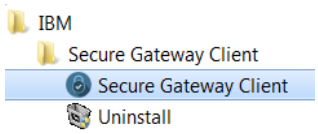
We will look at two options:

(A) manual startup using the secure gateway client console with
   (i) command line configuration and
   (ii) config file configuration
(B) configuration as per config file and start of the secure gateway client as a service

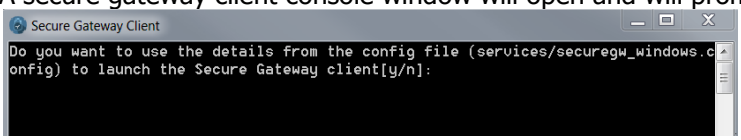
We will cover essential configuration options under section (A) below (because they are identical for both the manual and the service based gateway client). For further information on configuration options please refer to https://console.ng.bluemix.net/docs/services/SecureGateway/sg_021.html#sg_021.

6.3.2.2.1 Manual Startup & Configuration Essentials

1. **Start the client**:



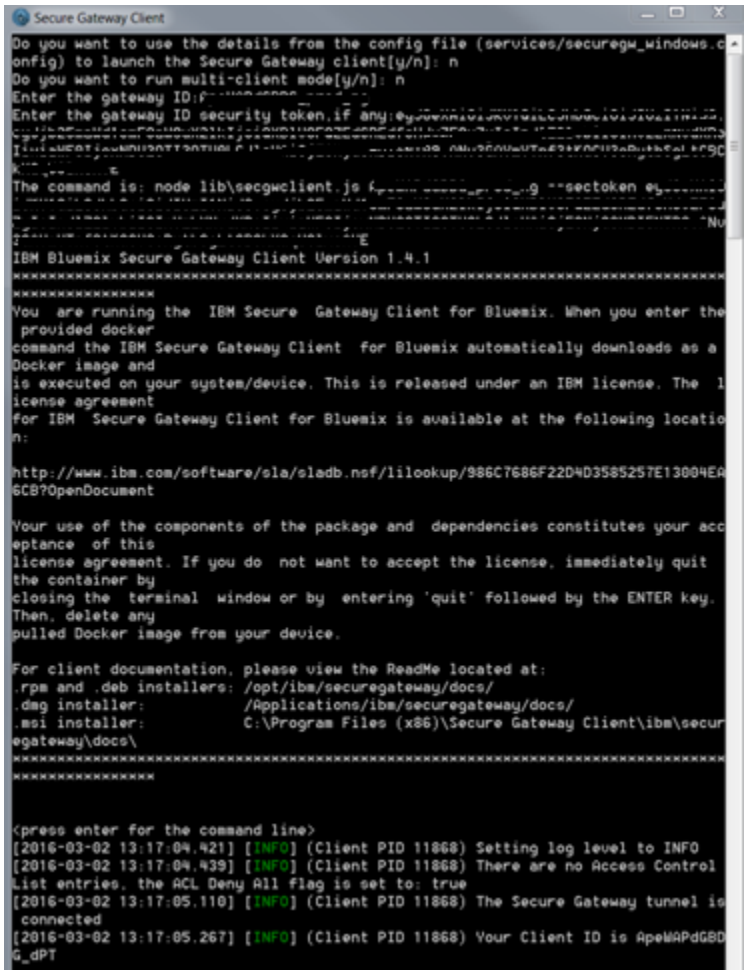
A secure gateway client console window will open and will prompted you with a question:



2. **Gateway Client Startup as per config file**

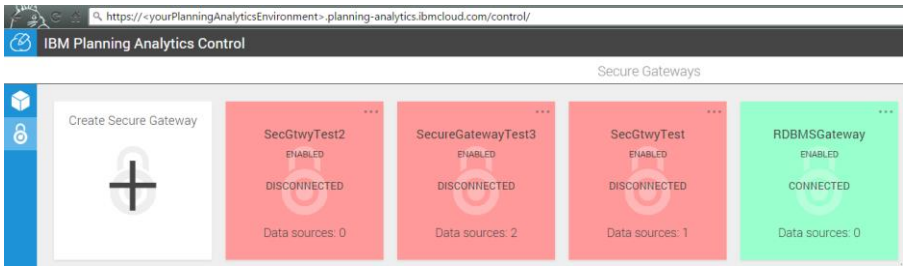
Start of gateway service connection as per connection, access and logging parameters entered during gateway setup (and setup-generated config file): enter Y if you did enter the Gateway ID and token before (enter N otherwise, we will show what you can do in section 4 if you did not enter Gateway ID and token).

If you hit Y and had entered correct Gateway ID and token during installation, the gateway will then automatically start with the corresponding parameters:



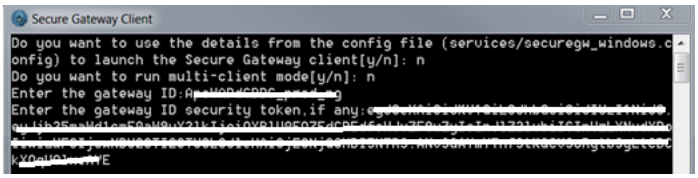
-> **Success, your gateway connection is established!**

If you go to the Planning Analytics control panel now, you will also see that the gateway is connected (it says 'connected' and is also in green):

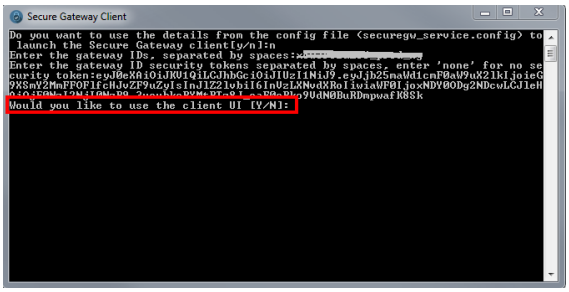


3. Gateway Client Startup with manual configuration

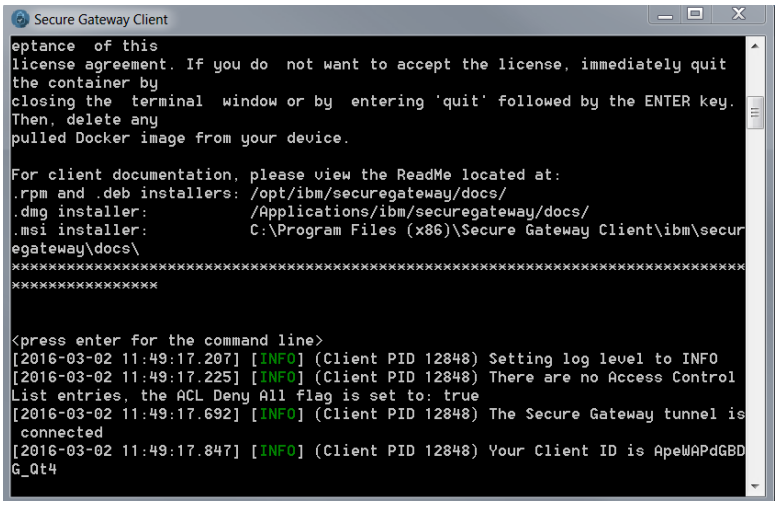
Manual entry of gateway service connection, access and logging parameters: If you hit N after the prompt from (2) above, you will be prompted for gateway ID and token:



If you are using Secure Gateway Client Version 1.5 or higher, you will be prompted if you want to use the Client UI:



We will look at the new Client UI functionality in section <Secure Gateway Client UI (as of Client Version 1.5)>, for the time being, please enter **N** (because we want to stay within the terminal environment for the time being):



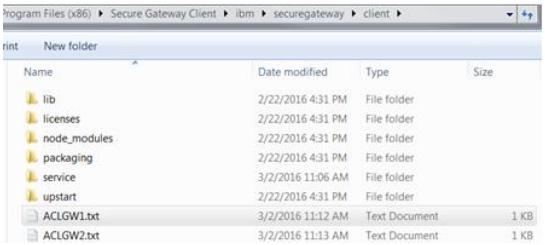
-> **Success, your gateway connection is established!**

4. **Access Control**

We have not (yet) configured an Access Control List (ACL) or provided ACL entries. **The access control list 'opens' the Database Server destinations to the Gateway Service on the Cloud. By Default, the Gateway Service on the cloud is DENIED Access to ALL, meaning while the tunnel is not established, the Gateway service cannot get anywhere from here.** We will need to allow access to your SQL Server & DB2 machine(s) as per the IP address or the machine name as well as the port. The machine name or FQDN is being resolved on the Secure Gateway Client (not on the cloud).

Entering
```
cli> acl allow 192.168.1.15:49244
cli> acl allow 192.168.1.15:50000
```
into the secure gateway command window hence will open ports 49244 (SQL Server) and 50000 (DB2) on 192.168.1.15. Again please note: you do not have to specify an IP address. A machine name is fine too. The name is resolved on the Secure Gateway client (so testing with ping for example will tell you if the client can resolve the destination name).

Instead of manually providing access control information, it is best to have the Secure Gateway client use an ACL file when starting up. Place the ACL file in the client directory[11], like the two files ACLGW1.txt & ACLGW2.txt below



The ACL files just need to contain the acl allow or deny entries as per your destination requirements and security needs, with the format:

acl allow *<hostname>*:*<port>*
acl deny *<hostname>*:*<port>*
no acl *<hostname>*:*<port>*

[11] **ACL file requirements:**
- Do not leave any residual white spaces.
- ACL files must be placed in the *<Installation_directory>*/ibm/securegateway/client directory or relative to that path. Where: *<Installation_directory>* is the name of the client installation directory that you chose.

For example:

acl allow :6666
acl deny localhost:22

or, in our example case:

acl allow 192.168.1.15:49244
acl allow 192.168.1.15:50000

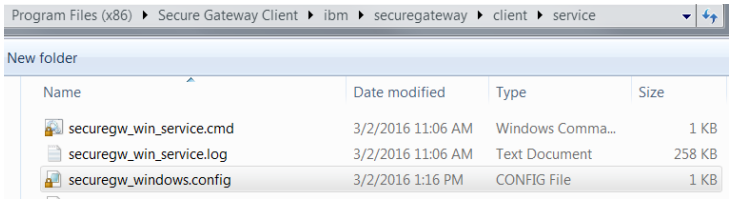
The *<hostname>* and *<port>* are optional, but you must enter one or the other. If the host name is omitted, all host names reachable by the client are affected, the same is true when you omit the port number. All port numbers are affected by the rule. For example, an 'acl allow *<hostname>*:' command allows connections to all ports on that host name and denies all other connections. Allow rules are mutually exclusive while deny rules are specific. Lines that are not understood or have an unrecognizable format are ignored. Shortcut commands within this file are not accepted.

Once you have an ACL file, you can configure the gateway client to use the ACL file on startup by including it into the startup config file:

5. **The gateway client config file**

The gateway client config file allows to start up one or multiple gateway service connection as per pre-specified startup parameters.

Version 1.4.*: The gateway config file 'securegw_windows.config' can be found at 'C:\Program Files (x86)\Secure Gateway Client\ibm\securegateway\client\service':



Version 1.5.*: The gateway config file 'securegw_service.config' can be found at 'C:\Program Files (x86)\Secure Gateway Client\ibm\securegateway\client'

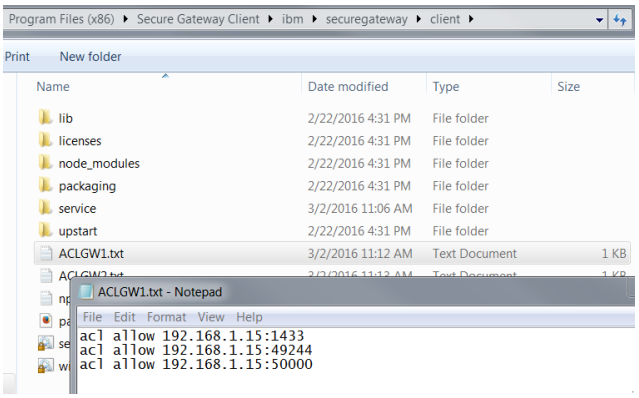
Open 'securegw_windows.config'/'securegw_service.config' with Wordpad. You will see that in case you provided Gateway ID and token (and possibly even ACL file(s), Log Levels, Multi- vs. Single gateway mode) during the gateway client installation[12], the config file will already contain those entries. You can modify the entries as per comments in the config file. Your secure gateway client will then use the config to start up and configure itself accordingly. This is also the file that will be used when starting up the secure gateway client as a service:

```
#Config file for Secure Gateway Client, to start as a Windows Service.
#PLEASE AVOID ANY RESIDUAL WHITE SPACES

#The value of MULTI flag needs to be y or Y if client needs to be launched in multi mode.
MULTI=N

#Enter the gateway id. If launching in multi mode, separate gateway ids by single spaces.
GATEWAY_ID=<GatewayID>

#Enter the security tokens. If launching in multi mode, separate security tokens by --.
```

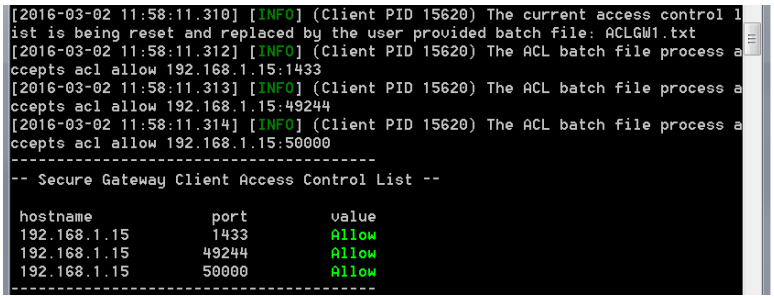
[12] see the screenshot from <In the next screen, you may (you do not have to) provide gateway service connection & (destination) configuration information. You can paste the Gateway ID and token into the provided fields:>

```
SECTOKEN=<JWT>

#Enter the ACL files. If launching in multi mode, separate ACL files by --.
ACL_FILE=ACLGW1.txt

#Enter the log levels. If launching in multi mode, separate log levels by --. NONE,
ERROR, INFO, DEBUG, TRACE
LOGLEVEL=INFO
```

In the above config, we specified an ACL file called ACLGW1.txt. Its contents are:



In the following screenshot, you can see that the ACL file ACLGW1.txt will be read and applied on startup as per config file:



6. Multi-Gateway Mode

If I were to configure the Gateway client in so-called multi-gateway mode, the gateway client would connect to multiple gateway services on the cloud. Here's an example configuration for two gateways:

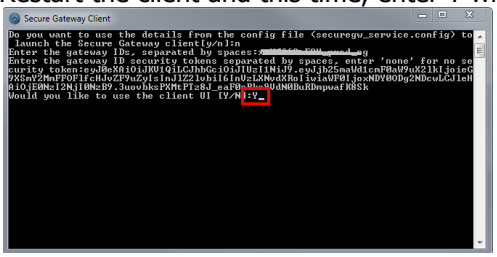
```
#Config file for Secure Gateway Client, to start as a Windows Service.
#PLEASE AVOID ANY RESIDUAL WHITE SPACES

#The value of MULTI flag needs to be y or Y if client needs to be launched in multi mode.
MULTI=Y

#Enter the gateway id. If launching in multi mode, separate gateway ids by single spaces.
GATEWAY_ID=<GatewayID1> <GatewayID2>

#Enter the security tokens. If launching in multi mode, separate security tokens by --.
SECTOKEN=<JWT1>--<JWT2>

#Enter the ACL files. If launching in multi mode, separate ACL files by --.
ACL_FILE=ACLGW1.txt--ACLGW2.txt

#Enter the log levels. If launching in multi mode, separate log levels by --. NONE,
ERROR, INFO, DEBUG, TRACE
LOGLEVEL=INFO—INFO
```

6.3.2.2.2 Secure Gateway Client UI (as of Client Version 1.5)

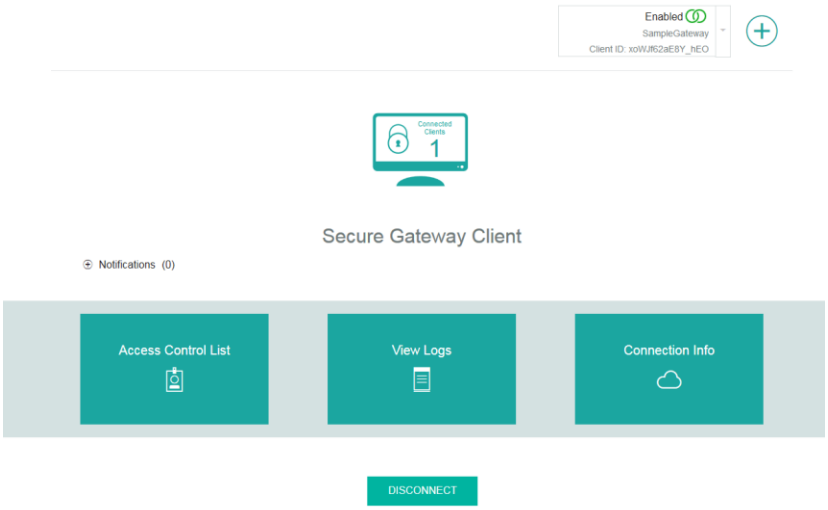
We will now explore the new Client UI available as of version 1.5:

1) Stop the current client (enter 'quit')

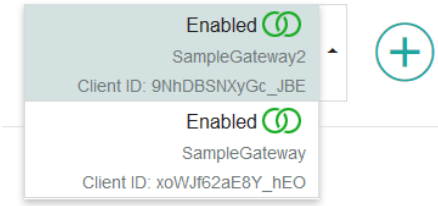
2) Restart the client and this time, enter Y when prompted if to use the client UI:



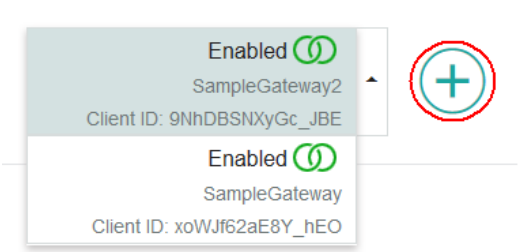
3) The UI will open in your default browser. The URL: http://localhost:9003/dashboard

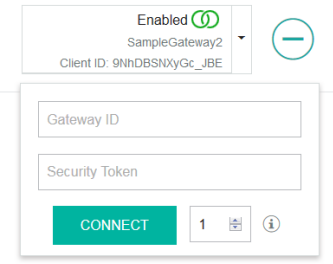


4) The UI will allow you to:
   a) View configuration, status & logs for multiple gateways:

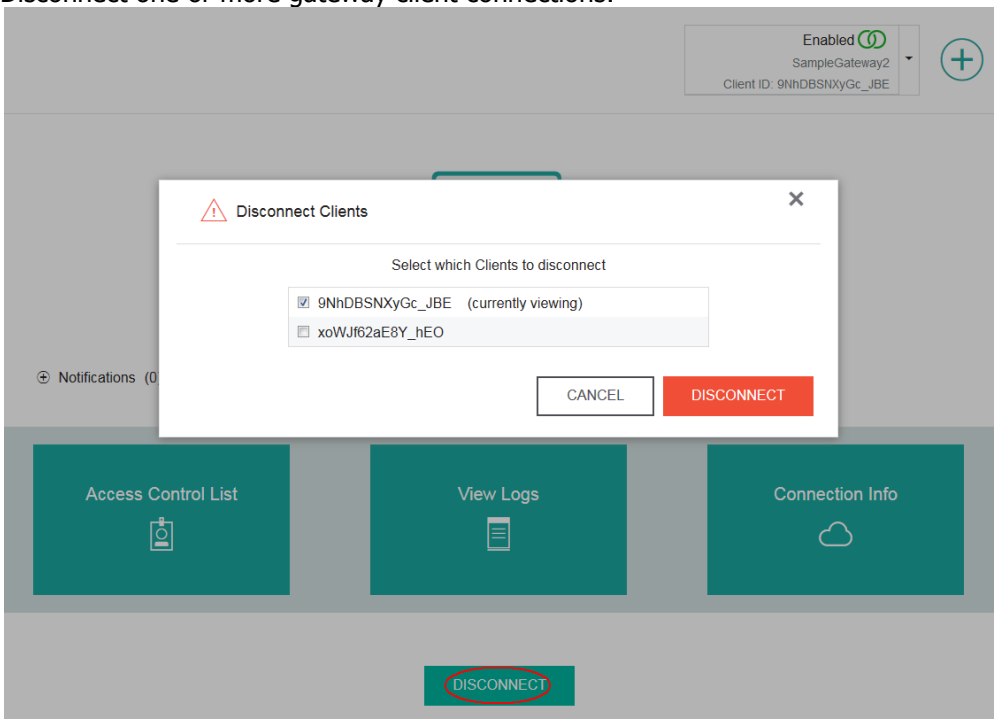


   b) Add gateways:

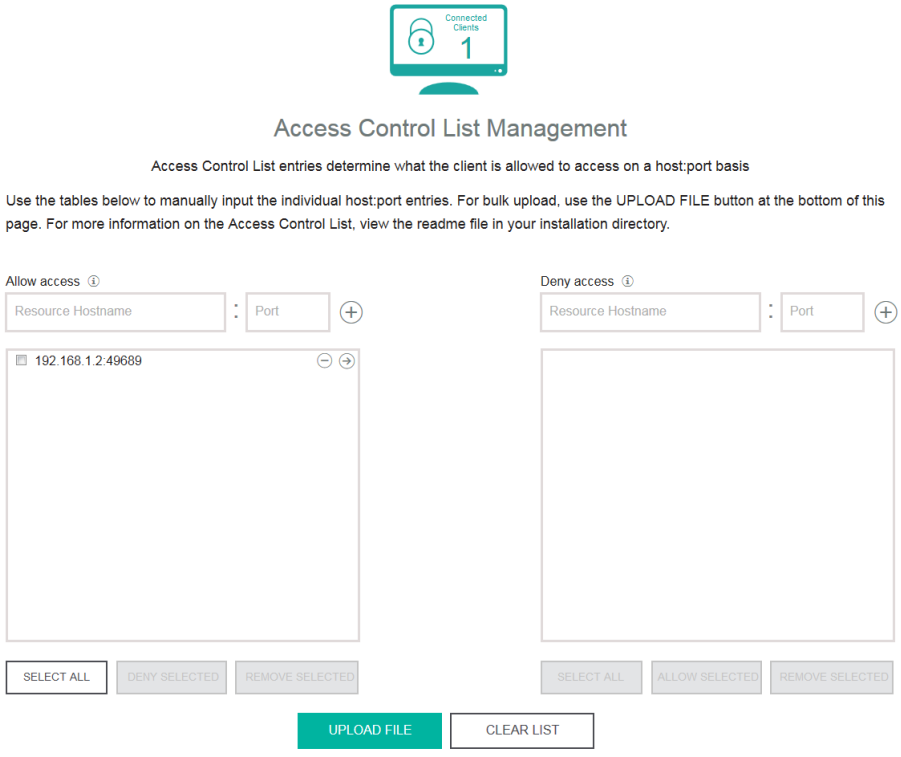




c) Disconnect one or more gateway client connections:

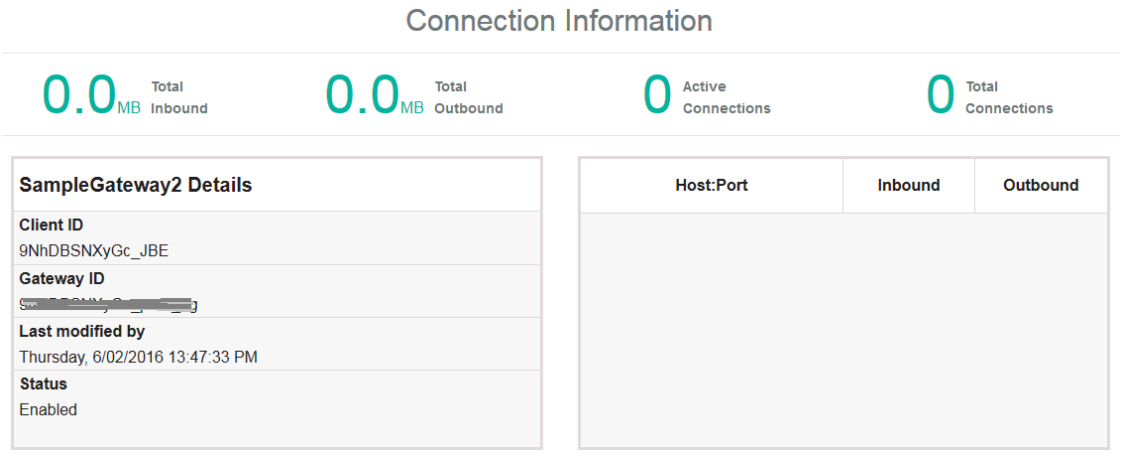


d) View ACL access credentials and restrictions & to upload an ACL file:



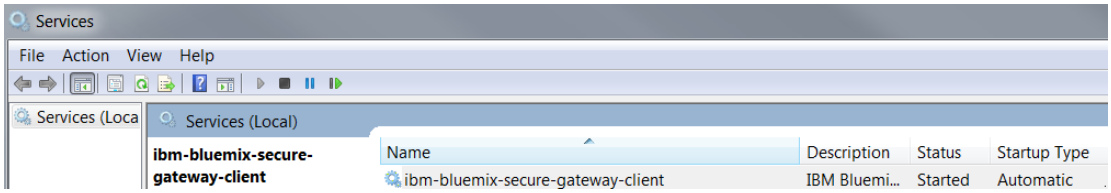
e) View logs, change log levels, export logs:

f) View Connection Info:

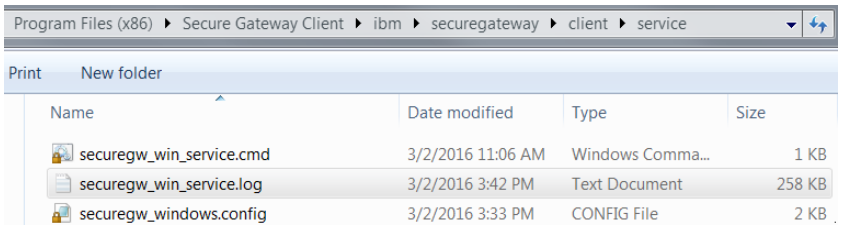


6.3.2.2.3 Startup as a Windows Service

If the Secure Gateway Client was configured to run as a service, it will be available under windows services:



Logging output – mirroring the type of output you get when using the Secure Gateway Client console in our prior section (the black console screenshots) – will now be generate in the securegw_win_service.log file in C:\Program Files (x86)\Secure Gateway Client\ibm\securegateway\client\service (V 1.4) or C:\Program Files (x86)\Secure Gateway Client\ibm\securegateway\client (V 1.5):

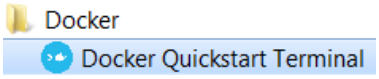


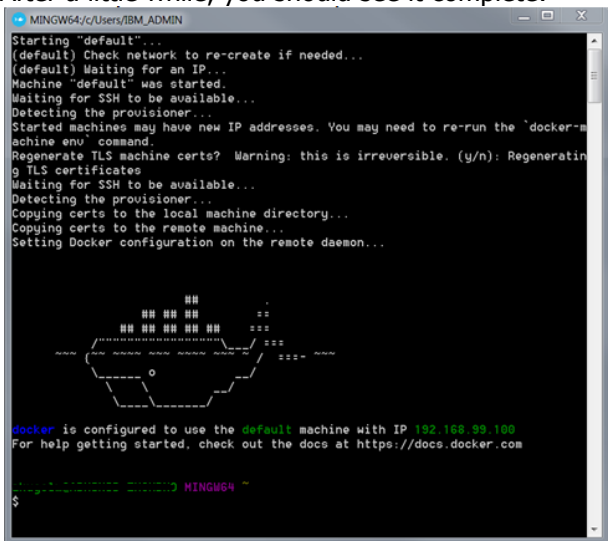
6.3.3 Secure Gateway Client via Docker

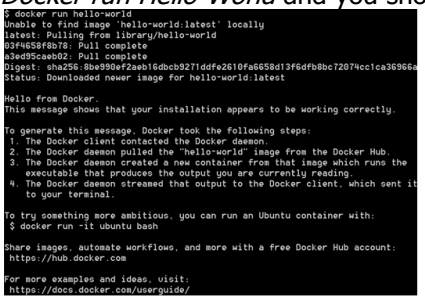
6.3.3.1 Installation

Check compatibility of your client machine & install the Docker client (via Cocker toolbox) as described in https://docs.docker.com/windows/step_one/. Note: contrary to some documentation on Bluemix, Docker toolbox 1.8.0 and higher **is** supported.
Docker will install and configure a lunix virtual machine. The commands that you will be running on Docker will run in that VM. That includes the IBM Secure Gateway Client. In other words: If you use Docker as your means for running the IBM Secure Gateway client, the Gateway client will run in a Docker Container.

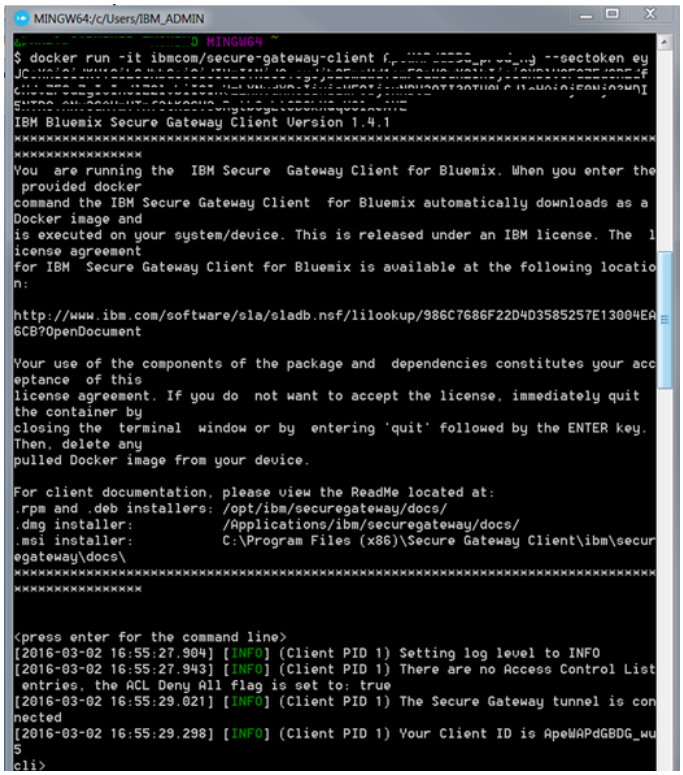
6.3.3.2 Startup and configuration

1. Click to open a Docker Terminal: 

2. You will see a command/terminal window pop up. Docker will initiate and start up the Container. After a little while, you should see it complete:


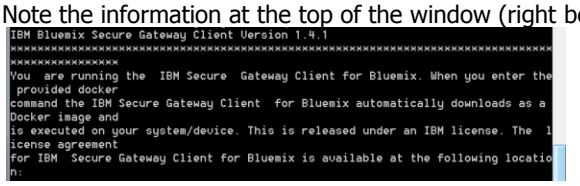
3. If you want to test your Docker installation as per Docker installation page suggestion, enter *Docker run Hello-World* and you should see something like


4. You can now **start the IBM gateway client in Docker**. You can use the Docker command that you copied to the clipboard in section <Create the Gateway Service>. If you do not have it anymore, retrieve it as describe in that same section. So we will paste
`docker run -it ibmcom/secure-gateway-client` *<GatewayID>* `--sectoken` *<JWT>*
into the window it and hit enter, and we will see this:



-> As you can see from the terminal window, you've been **successful; your gateway connection is established!**

Note the information at the top of the window (right below the Docker run command we entered):



It says: "You are running the IBM Secure Gateway Client for Bluemix. When you enter the provided docker command the IBM Secure Gateway Client for Bluemix automatically downloads as a Docker image and is executed on your system/device. This is released under an IBM license. The license agreement for IBM Secure Gateway Client for Bluemix is available at the following location:"
=> on first run, Docker will automatically pull and run the latest available IBM Secure Gateway client version.[13],[14],[15],[16]

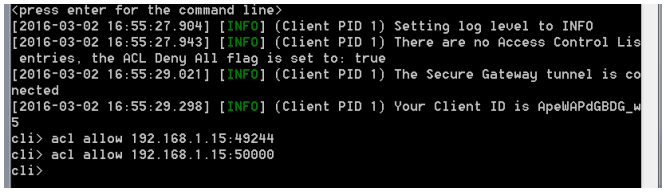
5. Just as with the Windows client, the Docker client was started wtih ACL Deny All. So we have to provide Access to the SQL Server and DB2 destinations via commands
   acl allow 192.168.1.15:49244

[13] the following two docker commands are supported: pull, run
[14] The Secure Gateway Docker client does not support the --multi option. The intent behind the high-availability support from the Docker client is supported by the idea of creating an individual container for each client that you are running, rather than creating a multi-process client in a single container.
[15] Updating the Docker client: Every so often you need to update the Secure Gateway client in order to get the latest version, which can include important security updates and fixes. You are notified when an update is required.

1. To update the Secure Gateway client, issue the following docker command.
   `docker pull ibmcom/secure-gateway-client`
2. Restart the client by issuing the Docker run command:
   `docker run -it ibmcom/secure-gateway-client…`

[16] At https://docs.docker.com/windows/step_two/ you can read a little more about Docker images and containers

acl allow 192.168.1.15:50000



You can also upload ACL files to Docker. Please refer to the aforementioned URL on Gateway Client configuration for detailed information.

6.4 Secure Gateway Destinations: Data Source Setup and Configuration

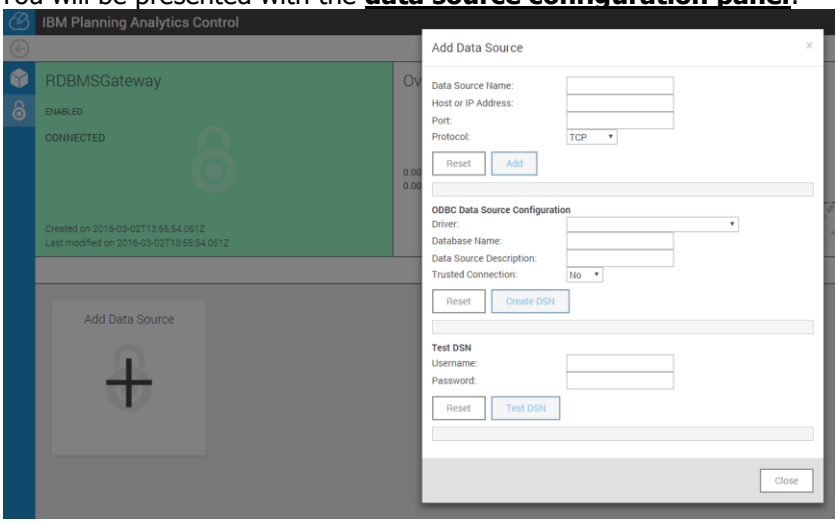
Now that we have established a Secure Gateway Tunnel between on-premises and the Planning Analytics Cloud we need to create and configure the destinations (= data sources) for the tunnel:

6.4.1 DSN Setup and Configuration: The Configuration Panel

1. In the Planning Analytics Control Panel, click on the secure gateway icon (the lock) and then click on the Tile for the Secure Gateway you want to configure destinations / data sources for. In our example, this was 'RDBMSGateway'.

2. You will see the Secure Gateway Dashboard for 'RDBMSGateway'. Click on 'Add Data Source':

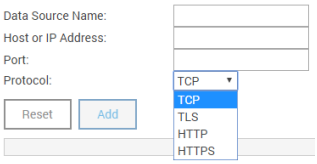
3. You will be presented with the **data source configuration panel**:



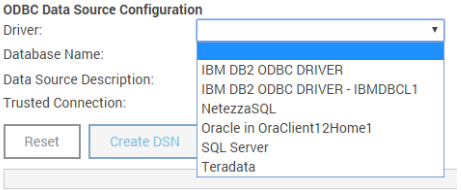
- **Data Source Name**: The name for your data source. **This will be the** DSN that you will be using in Turbo-Integrator when connecting to the on-premises database

- **Host Name or IP Address**: The Host **Name or IP address of the on-premises machine running the RDBMS database**. This will be the destination machine that the Gateway client will direct the tunnel packet to once the query request has arrived on premises.

- **Port**: The Port on the on-premises machine that the RDBMS database is running on.
- **Protocol**: The Protocol by which you want to communicate with the database (TCP, HTTP, HTTPS, TLS):



- **Driver**: the ODBC driver you want to use. The following drivers are available by default:



Other drivers may be installed per request. Due to required legal procedures, the installation of such ODBC drivers may take approximately 2-3 weeks.

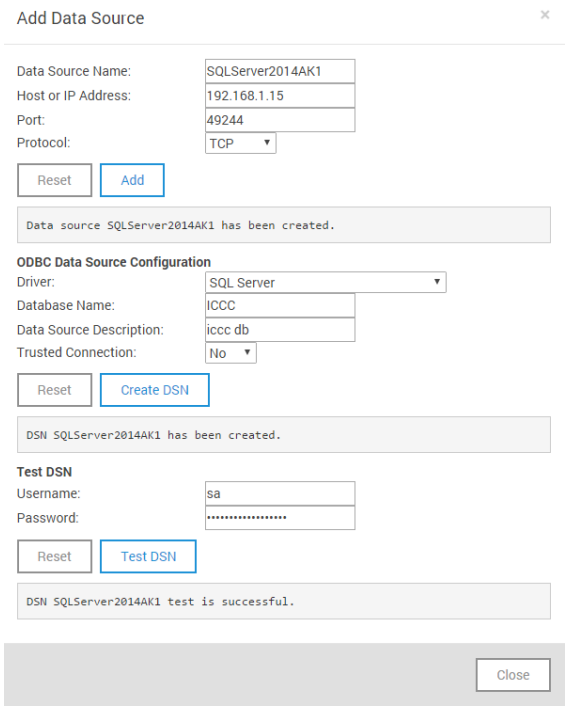
- **Data Source Description**: just a description for the DSN (required)

- **Trusted Connection**: ODBC Driver configuration; specifies if the driver is to establish a 'trusted connection' to the database. Functionalities, features etc. depend on the database. Consult your database-specific documentation for details.

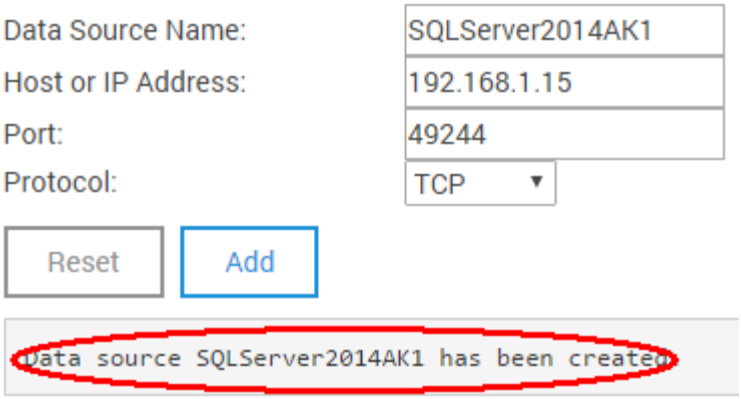
- **User Name**: a user name to test the connection. Only used for testing.
- **Password**: the password for the above user to test the connection. Only used for testing.

6.4.2 DSN Setup and Configuration: SQL Server Example

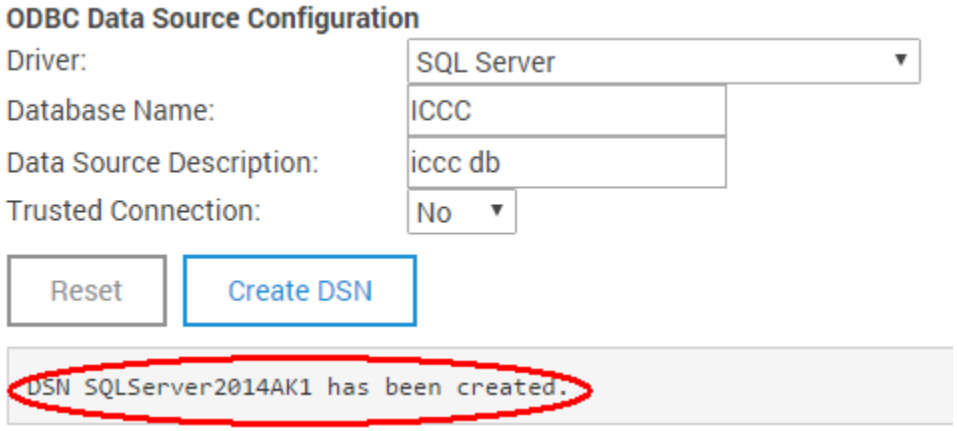
Configure the Data Source as follows:



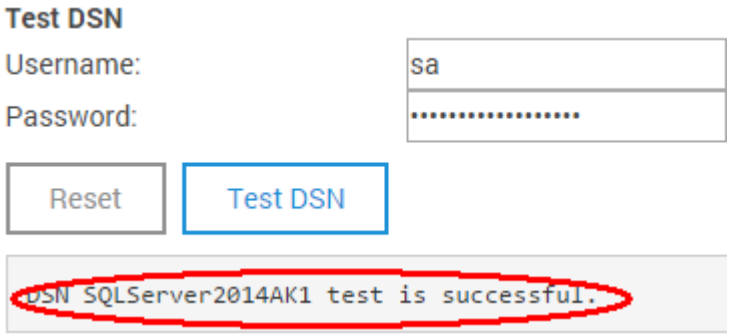
1. Enter DSN name, destination, port and communication protocol. Then click 'Add':



2. Specify the Driver, Database, DB Description and Trusted Connection, then click 'Create DSN':



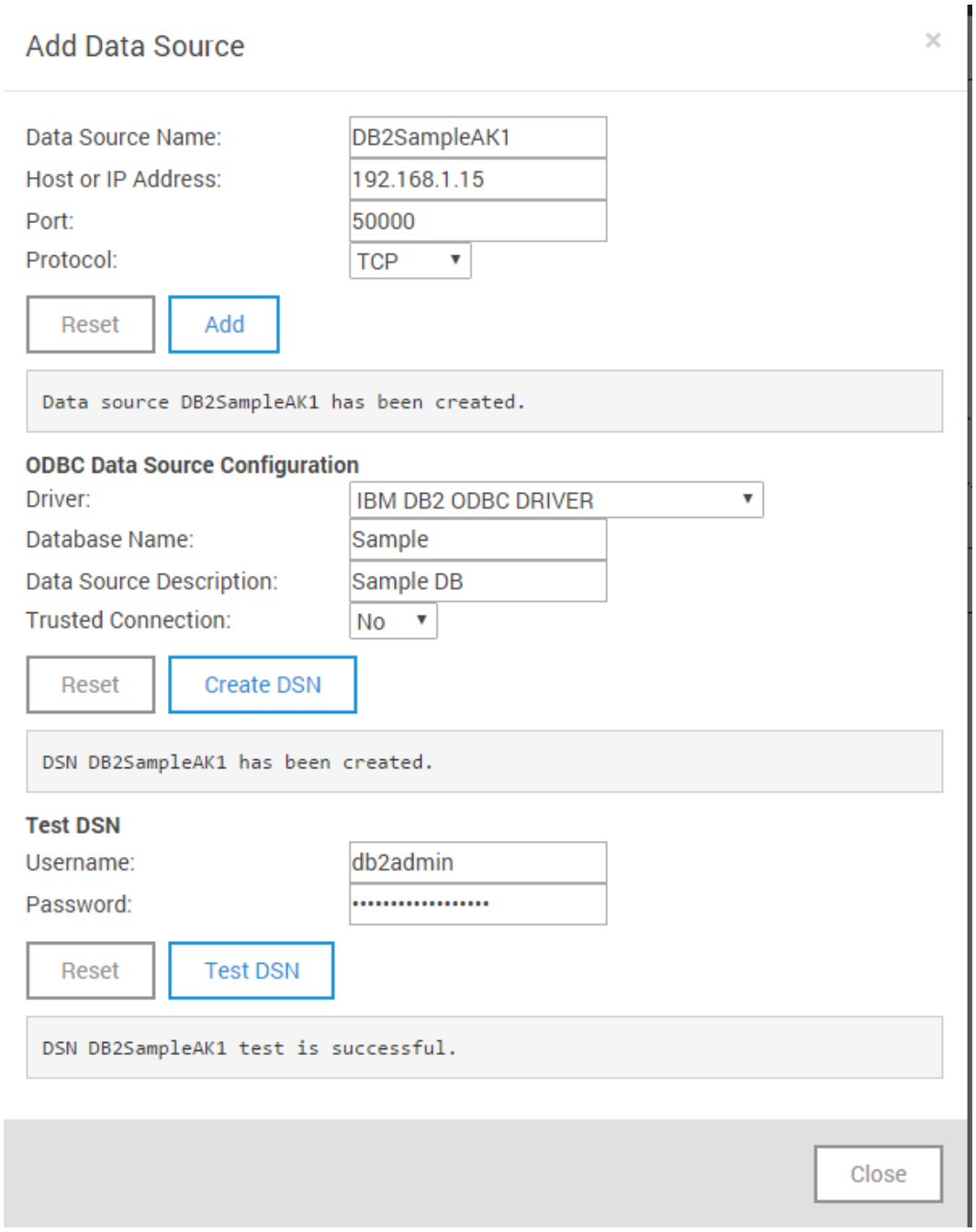
3. Enter a user name and pw and click 'Test DSN'. You should see:



Use the RESET Button when making changes to a DSN

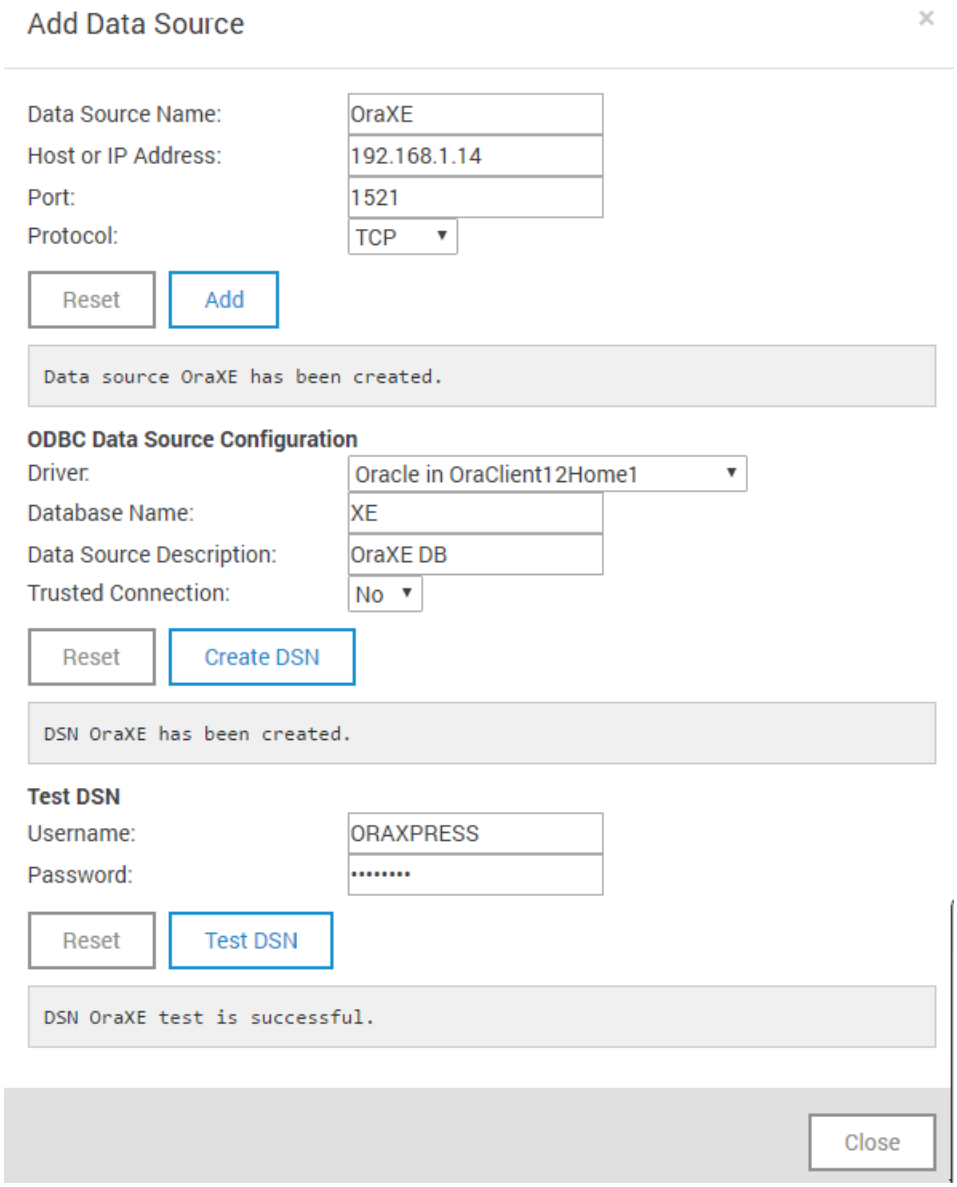
6.4.3 DSN Setup and Configuration: DB2 Server Example

For DB2, the DSN Setup and configuration will be very similar to SQL Server. The port is different and so is the test user, but apart from that the configuration is the same:



6.4.4 DSN Setup and Configuration: Oracle Example

Below is a sample configuration for an Oracle Database (Express Edition). Use the IP address or machine name of the Oracle Server machine (here 192.168.1.14) and the Oracle Database port (1521 is the default port for OracleXE). For the database name, use the TNS Service name (without the IP address / machine name).

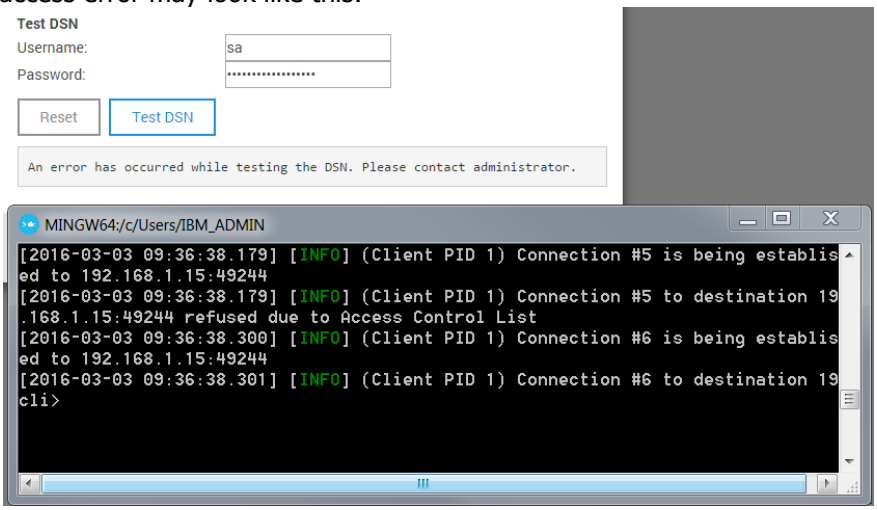


6.4.5 DSN Troubleshooting

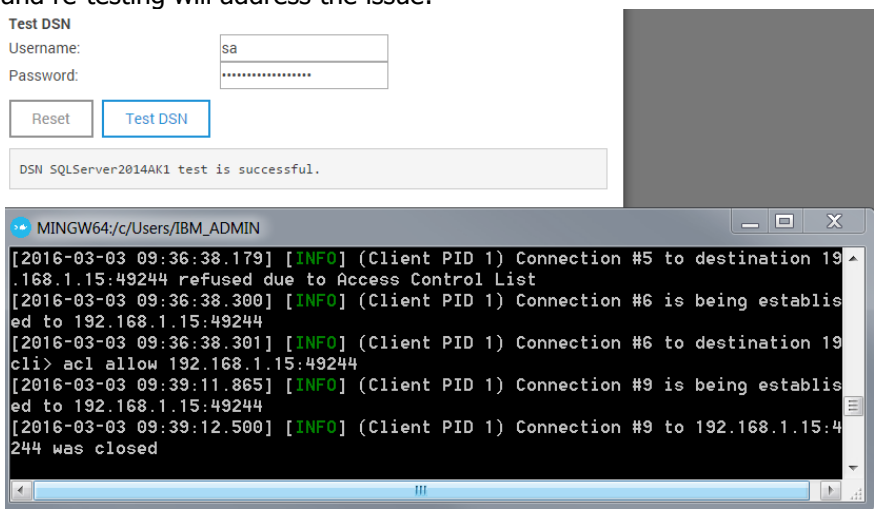
1. Ensure that you can access the Database with the selected protocol and user/pw via on-premises ODBC. Use the same protocol for your on-premises test. I.e. do not test with Named Pipes if you are using TCP via the gateway.

2. If you have trouble getting ODBC to work (on prem), consult the customer's DB admin team to assist. ODBC setup for some databases can be more involved than for SQL Server for example. DB2 will require you to bind the Database to ODBC and the ODBC driver installation is not as automated as for SQL Server for example.

3. Take a look at the logs from the gateway console. Do you get Access control errors? You can also increase the log levels (see config file for the gateway client) to get more information. An ACL access error may look like this:



You see that 'an error has occurred while testing the DSN'. In the log (file), you can see that the connection to 192.168.1.15:49244 was refused due to ACL. Therefore, issuing the command
acl allow 192.168.1.15:49244
and re-testing will address the issue:



4. Changing Log levels: For detailed troubleshooting, Log levels NONE, ERROR, INFO, DEBUG, TRACE are available. In our example configuration using the IBM Secure Gateway Client, we used log level INFO. To change the log level in the IBM client, use the following syntax:

```
loglevel <ERROR | INFO | DEBUG | TRACE> <process ID>
logpath <file> <process ID>
```

In the following example, we are changing the log level for PID 12200 to DEBUG:

```
cli> loglevel DEBUG 12200
[2016-03-03 16:52:01.173] [INFO] (Client PID 12200) Setting log level to DEBUG
```

To change the log level to debug on the Docker client, hit enter in the command line and set the log level to DEBUG or TRACE by entering l DEBUG or l TRACE, like in

```
<press enter for the command line>
[2015-09-21 04:04:40.414] [INFO] The Secure Gateway tunnel is connected

cli> l DEBUG
[2015-09-21 04:04:45.775] [INFO] Setting log level to DEBUG
```

Note that for reconnecting to a running Docker container, you can simply type docker ps to get a list of running container IDs, and then you can type Docker attach $ID to get back into the container. Pressing the Enter key will get you back to the CLI prompt at any time once you are inside the container, and you can use the 'C' and 's' commands to check the current status of the gateway and its connections.

5. IP address may change. IF you are using IP addresses, make sure it is still the correct one and that it is in line with ACL allow entries. Better: use Machine Names and have DNS lookup take care of the resolution so you do not have to worry about IP's changing.

6. ACL commands are case sensitive. If you use machine names in your acl lists, be aware that the machine name may also be case sensitive.

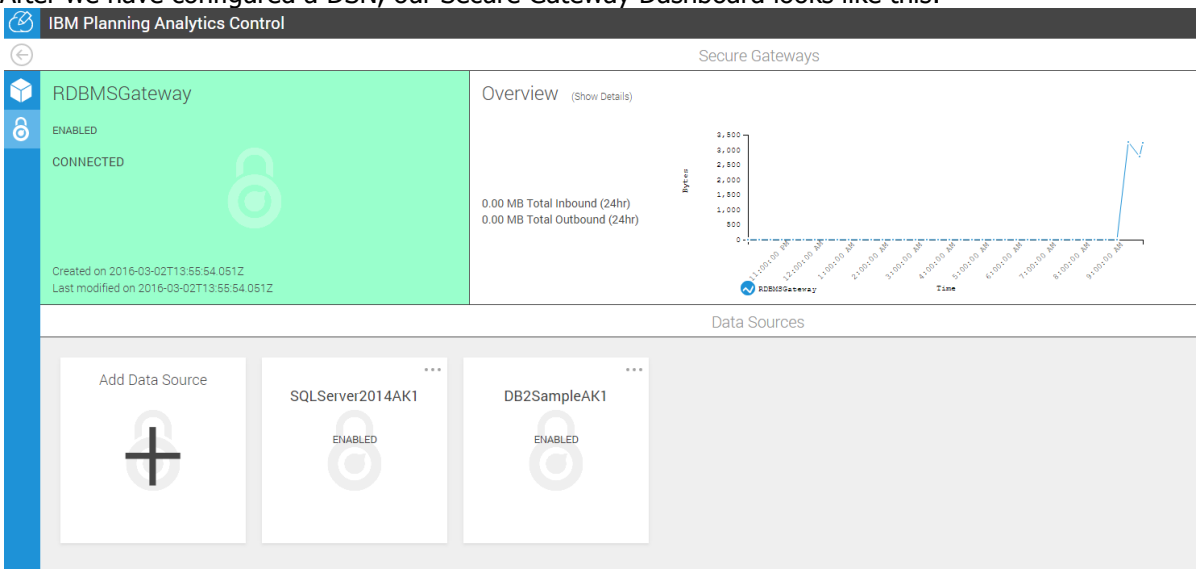
7. Enter Gateway IDs and Security Tokens exactly as provided by the Gateway UI.

8. Ports may change too. SQL Server for example allows ports to be dynamically assigned. In this case, it is a good practice to assign a specific port to SQL Server and to use this port in the ACL. See <Appendix> for examples.

6.4.6 The Secure Gateway Dashboard

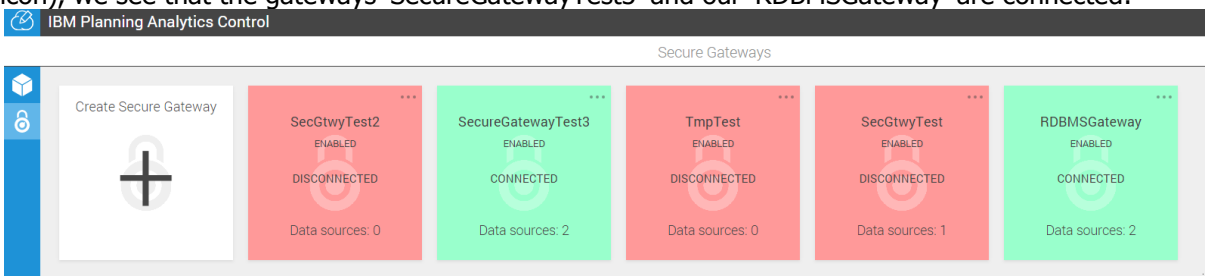
6.4.6.1 The Dashboard with DSNs

After we have configured a DSN, our Secure Gateway Dashboard looks like this:



(because the gateway tile is green we know that it is connected, i.e. that a client has established connection)

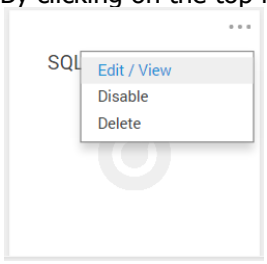
If we go to the Secure Gateway overview panel (the main secure gateway menu accessed via the lock icon), we see that the gateways 'SecureGatewayTest3' and our 'RDBMSGateway' are connected:



They are now both connected because in the example, the secure gateway client was started in multi-gateway mode as described in section <Multi-Gateway Mode>.

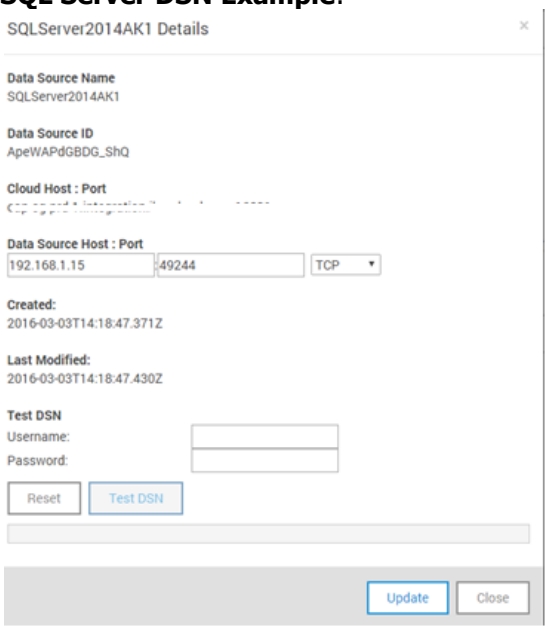
6.4.6.2 Changing DSNs

By clicking on the top right corner of a Data Source tile,



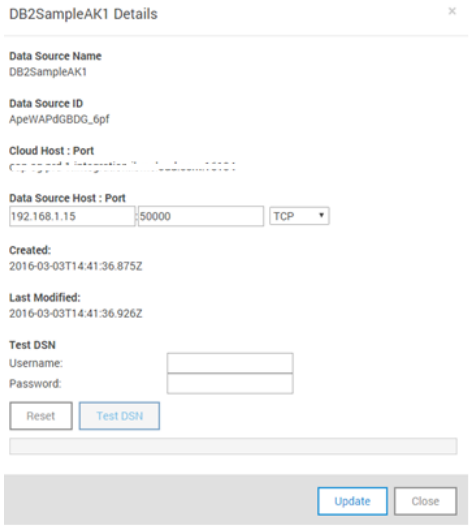
you can view and edit the DSN configuration:

SQL Server DSN Example:



Note the Cloud destination address
<FQDN>:<Port> (<machine>.integration.ibmcloud.com:<port>):
This is the destination for the DSN and ODBC driver. It is the Cloud Gateway Service destination that on the Gateway client side will get (re-)mapped or 'forwarded' to the on-premises address *192.168.1.15:42944*

DB2 DSN Example:

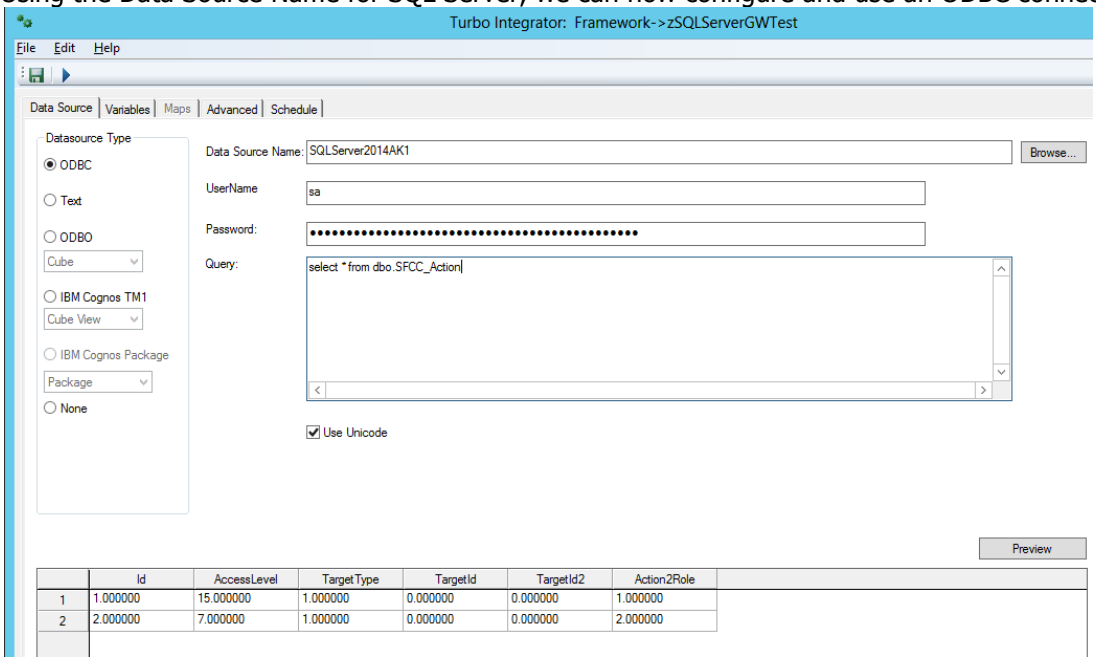


Note the Cloud destination address
<FQDN>:<Port> (<machine>.integration.ibmcloud.com:<port>)
This is the destination for the DSN and ODBC driver. It is the Cloud Gateway Service destination that on the Gateway client side will get (re-)mapped or 'forwarded' to the on-premises address *192.168.1.15:50000*

6.4.6.3 Using the DSN

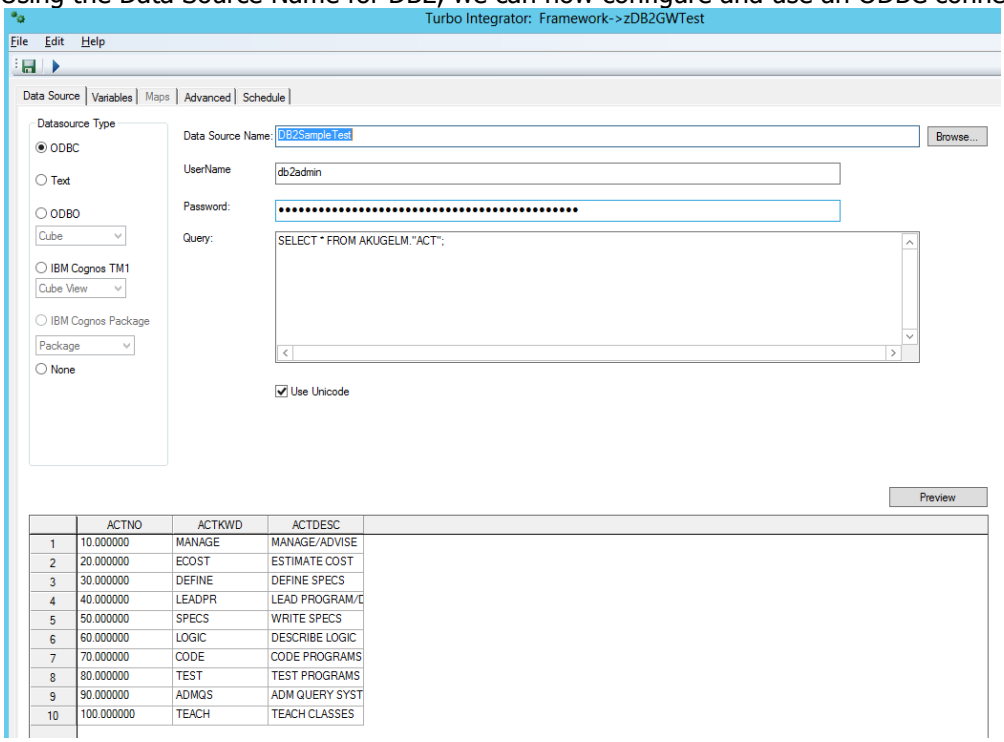
6.4.6.3.1 SQL Server Example

Using the Data Source Name for SQL Server, we can now configure and use an ODBC connection in TI:



6.4.6.3.2 DB2 Example

Using the Data Source Name for DB2, we can now configure and use an ODBC connection in TI:



7. Appendix

7.1 ODBC Configuration

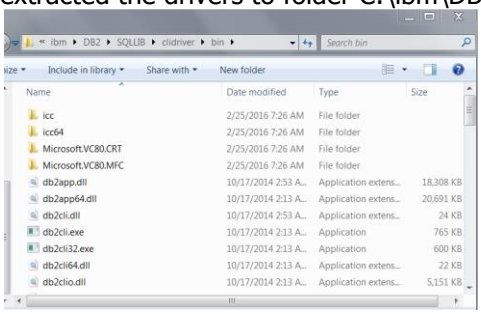
7.1.1 DB2

7.1.1.1 DB2 ODBC Driver Download

IBM Tech Note: https://www-304.ibm.com/support/docview.wss?uid=swg21418043
32bit DB2 ODBC CLI Drivers: https://www-933.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm%2FInformation%2BManagement&product=ibm/Information+Management/IBM+Data+Server+Client+Packages&release=10.5.*&platform=Windows+32-bit,+x86&function=fixId&fixids=*odbc_cli*&includeSupersedes=0
64bit DB2 ODBC CLI Drivers: https://www-933.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm%2FInformation%2BManagement&product=ibm/Information+Management/IBM+Data+Server+Client+Packages&release=10.5.*&platform=Windows+64-bit,+x86&function=fixId&fixids=*odbc_cli*&includeSupersedes=0

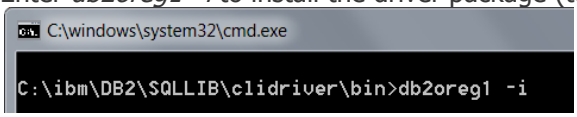
7.1.1.2 DB2 ODBC Driver Installation & Registration

1. Download and extract the DB2 ODBC Driver package to a folder. In the below example, we extracted the drivers to folder C:\ibm\DB2\SQLLIB\clidriver\bin

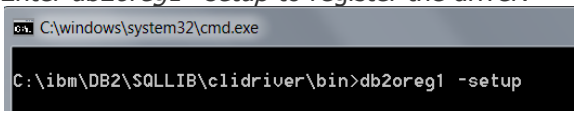


2. Open a CMD prompt with Administrator privileges ('run as Administrator') and go to the folder where you extracted the driver package (in our case, C:\ibm\DB2\SQLLIB\clidriver\bin)

3. Enter *db2oreg1 –i* to install the driver package (to uninstall it, you would enter *db2org1 –u*).

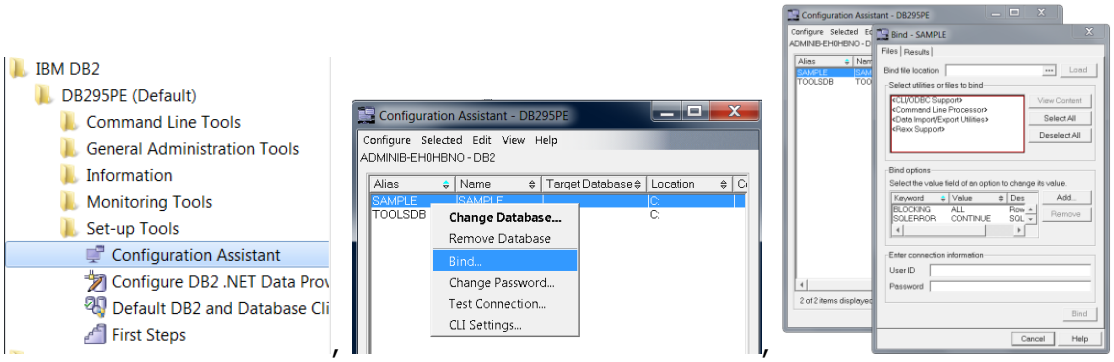


4. Enter *db2oreg1 –setup* to register the driver:



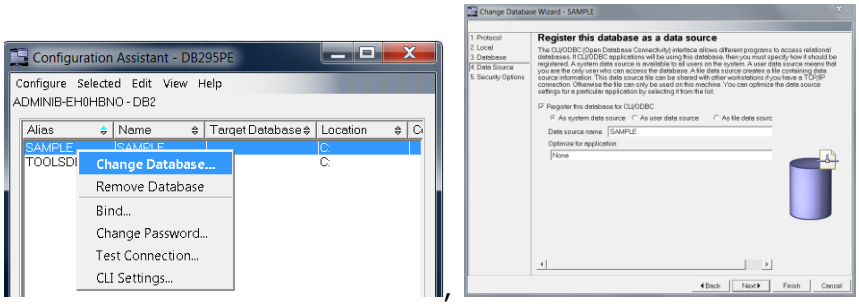
7.1.1.3 Bind CLI/ODBC

Bind using the Configuration Assistant (or consult your DB2 documentation on how to bind a DB to ODBC):

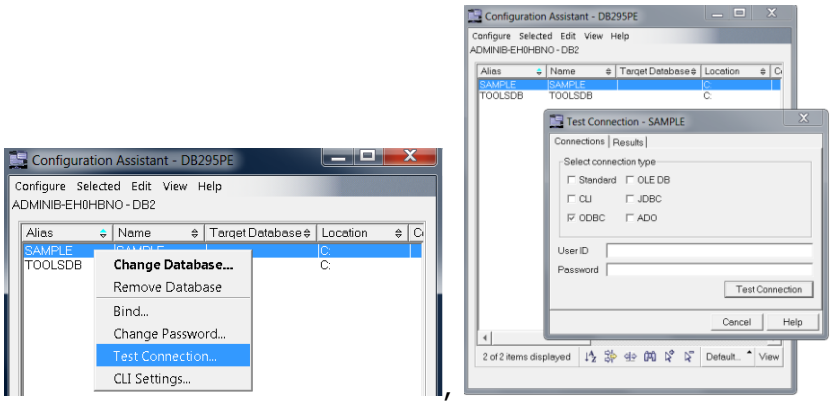


Also see https://www.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.apdv.cli.doc/doc/t0006343.html

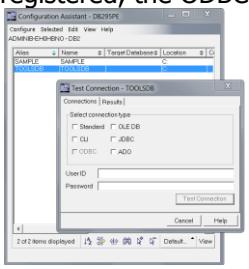
7.1.1.4 Register the DB for ODBC



7.1.1.5 Test the connection



If the DB is not properly registered for ODBC access and/or the ODBC driver is not installed and registered, the ODBC box may be grayed out:

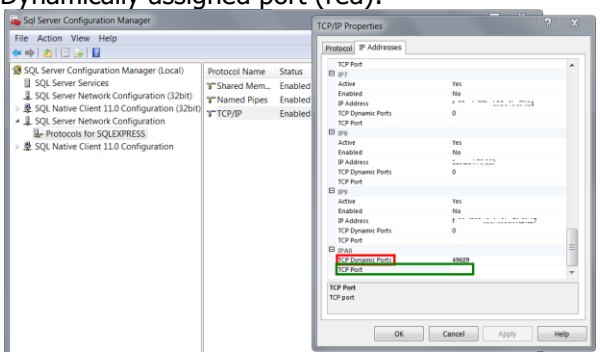


7.1.2 SQL Server

7.1.2.1 SQL Server Database Port

SQL Server for allows ports to be dynamically assigned. In this case, it is a good practice to assign a specific port to SQL Server and to use this port in the ACL.

Dynamically assigned port (red):



Static Port (green):