MS SQL on GCP - Cloud IAP

Connect to and manage MS SQL on GCP compute using SQL management software (SQL Management Studio) via Cloud IAP.

In a cloud world where zero-trust is a must and context is key, using Cloud IAP to secure your cloud resources is vital. There are several reasons to use Cloud IAP as opposed to traditional VPN, first and foremost, it's secure, it works and it's never been compromised.

Some other reasons include, but are not limited to:
reducing the attack surface of your environment.
building context/intelligence around access, and providing a light weight & secure way for your internal staff to manage infrastructure from anywhere.

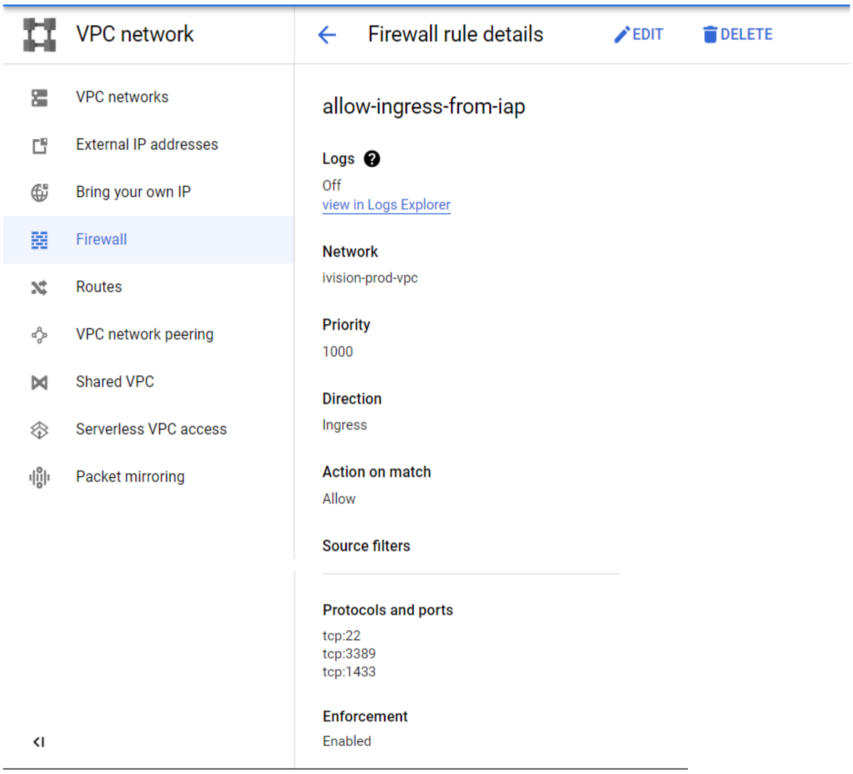
Cloud IAP was initially developed as part of Google's BeyondCorp model, as a way to allow their own employees to connect to internal resources from anywhere, permitting the context made sense.

CloudIAP essentially tunnels/forwards TCP traffic via HTTPS. As a result, it allows you to connect securely from anywhere.

Firewall Rules

To keep things simple, in this use case I leveraged network tags. In higher security environments though, you would ideally use firewall rules based on service accounts.

Create a new firewall rule for inbound traffic and specify the following:



allow-ingress-from-iap

Logs
Off
view in Logs Explorer

Network
ivision-prod-vpc

Priority
1000

Direction
Ingress

Action on match
Allow

Source filters

Protocols and ports
tcp:22
tcp:3389
tcp:1433

Enforcement
Enabled

Name: Name for the Rule - to be able to identify it (allow-ingress-from-iap)

Network: select the network your SQL instances live in, if they live in multiple networks, you'll have to create separate firewall rules.

Direction of Traffic: Ingress

Action on Match: Allow

Targets: select '*Specified Target Tags*' from drop down menu

Target tags: enter the tag used for your MS SQL instances (in my case, 'allow-ingress-from-iap') … if you haven't entered tags on your instances already, you'll have to edit your compute instances running MS SQL and add the network tag you specify here.

Source Filter: Select '*IP Ranges*' from the drop down menu

Source IP Ranges: 35.235.240.0/20 (this is the Cloud IAP CIDR block)

Protocols and Ports: select '*Specified protocols and ports*', then check the box next to '*TCP*' and enter 1433 (or 1433,3389 to enable RDP access also via tunnel) next to it

Grant permissions to access the IAP secured tunnel

- Next, we will need to grant access to users to access the secured IAP resources to tunnels. One of the great things about IAP, is that you have the option to granularly grant access to users on a per-instance basis. Ideally, you will grant users access via a group they have membership in. This would be best practice and make permission management easier and more scalable.
- Click the menu in the top left /Select Security
- Select Identity-Aware Proxy
- Click the 'SSH and TCP Resources' tab
- Select the checkboxes next to the resources you would like to grant access to. (you can choose to do this by selecting all resources, individual resources, or zonal/regional resources… whatever makes sense in your environment. Simply check the boxes next to the resources you want to apply the permissions to)
- In the frame on the right hand side, click 'Add Member'

New Members: Enter the Google group email address for users (or individual users email, but using groups would be best practice)

Select a Role: Click the drop down under '*Select a Role*', then select '*Cloud IAP*' and finally, select '*IAP-secured Tunnel User*'.

Condition (Optional, but recommended): You can choose to begin building context around access by selecting ‘Add Condition’ under Condition. The possibilities are pretty extensive as far as the context you can select, to keep things simple for now, simple restrict access to port 1433 (and optionally also 3389 to enable RDP access). After clicking add condition, enter a title to identify the condition, then select the ‘*Condition Editor*’ tab. In the text box, enter the following text, without the quotes, then click save: ‘destination.port == 1433 || destination.port == 3389’

Building the tunnel on local machine

This is actually incredibly easy, given that you have the Cloud SDK installed and configured on your computer.

Note: if you manage several projects via your account, set the project that contains the resources you want to connect to first, by running: gcloud config set project <project-id>

There is only one command to build the Cloud IAP tunnel from the local machine and start the TCP listener locally.

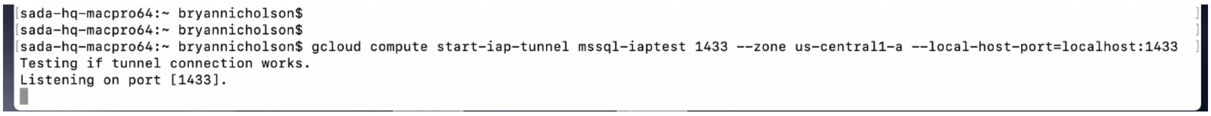
1. Open the cloud shell on Windows, or Terminal on MacOS (or whichever OS you’re using) and type the following:

gcloud compute start-iap-tunnel <instancename> 1433 — zone <zoneinstanceisin> local-host-port=localhost:1433

gcloud compute start-iap-tunnel ubuntu-console 8080 --zone europe-west1-b --local-host-port=localhost:8080

replace <instancename> with the instance name you’d like to tunnel to, replace <zone> with the zone that the instance resides in. If you would like to build a tunnel to a different port, simply replace 1433 with the desired port.

— local-host-port=localhost:1433 is the command that sets the local listener which will receive traffic, then push it through the tunnel to the set endpoint.

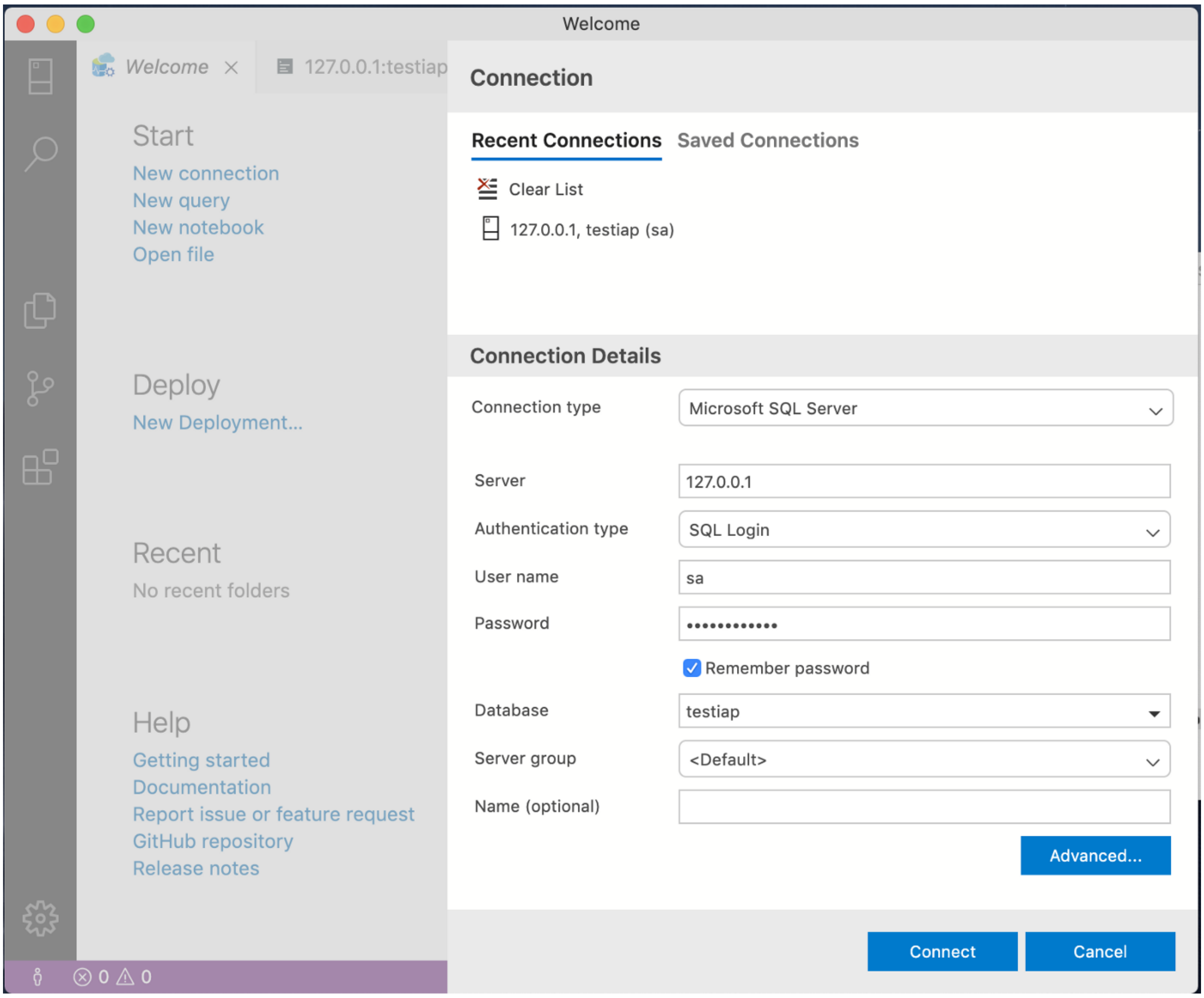


```
sada-hq-macpro64:~ bryannicholson$
sada-hq-macpro64:~ bryannicholson$
sada-hq-macpro64:~ bryannicholson$ gcloud compute start-iap-tunnel mssql-iaptest 1433 --zone us-central1-a --local-host-port=localhost:1433
Testing if tunnel connection works.
Listening on port [1433].
```

View of active tunnel/TCP listener

Note: The tunnel does not tear itself down. When you are done with your session, simple stop the listener (ctr + c on Windows or MacOS) or close the instance of cloud shell with the listener. This will effectively terminate your Cloud IAP tunnel.

That’s all, to connect to your remote instance, point your management software at the localhost (127.0.0.1) on port 1433.



Connection

Recent Connections | Saved Connections

Clear List

127.0.0.1, testiap (sa)

Connection Details

Connection type: Microsoft SQL Server

Server: 127.0.0.1

Authentication type: SQL Login

User name: sa

Password: ••••••••••••

☑ Remember password

Database: testiap

Server group: <Default>

Name (optional):

Connecting to MS SQL on GCP Compute instance via local listener/Cloud IAP tunnel

Now all connections to your MS SQL instances on GCP compute are secured via Cloud IAP, a zero-trust, context-aware method!

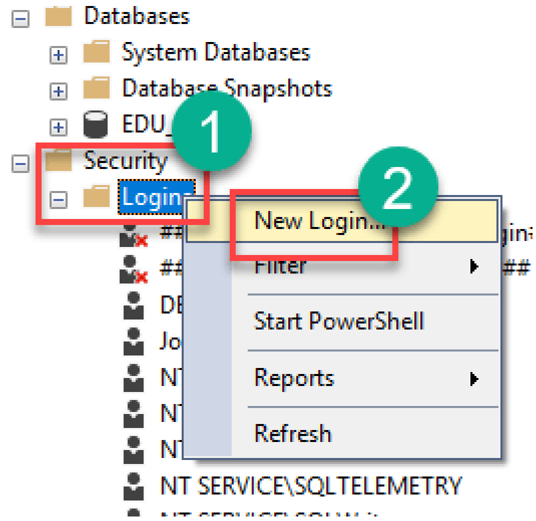
SQL Management Studio :

Change authentication mode with SSMS

- In SQL Server Management Studio Object Explorer, right-click the server, and then click Properties.
- On the Security page, under Server authentication, select the new server authentication mode, and then click OK.
- In the SQL Server Management Studio dialog box, click OK to acknowledge the requirement to restart SQL Server.
- In Object Explorer, right-click your server, and then click Restart. If SQL Server Agent is running, it must also be restarted.

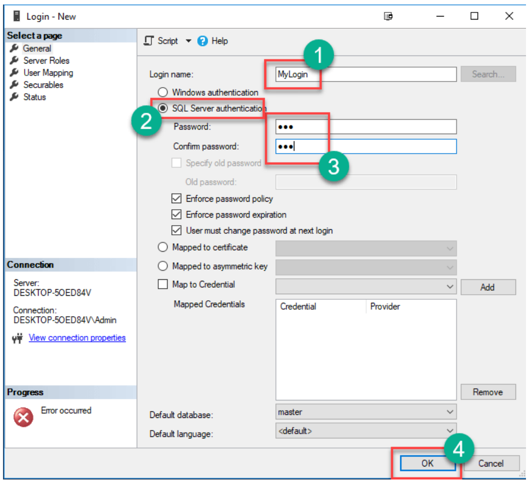
Create a SQL User :

Step 1) To create login SQL server, Navigate to Security > Logins



Step 2) In the next screen, Enter

1. Login Name
2. Select SQL Server authentication
3. Enter Password for MySQL create user with password
4. Click Ok


Login - New

Login name: MyLogin

○ Windows authentication
● SQL Server authentication

Password: •••
Confirm password: •••

□ Specify old password
Old password:

☑ Enforce password policy
☑ Enforce password expiration
☑ User must change password at next login

○ Mapped to certificate
○ Mapped to asymmetric key
□ Map to Credential

Mapped Credentials

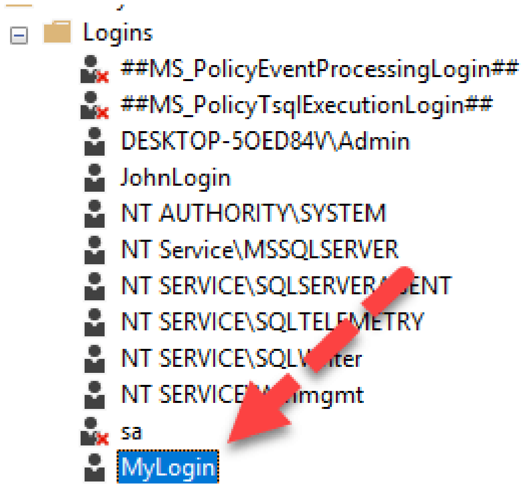
Credential	Provider

Default database: master
Default language: <default>

Connection
Server: DESKTOP-5OED84V
Connection: DESKTOP-5OED84V\Admin

Progress
Error occurred

Step 3) Login is created



Logins
##MS_PolicyEventProcessingLogin##
##MS_PolicyTsqlExecutionLogin##
DESKTOP-5OED84V\Admin
JohnLogin
NT AUTHORITY\SYSTEM
NT Service\MSSQLSERVER
NT SERVICE\SQLSERVERAGENT
NT SERVICE\SQLTELEMETRY
NT SERVICE\SQLWriter
NT SERVICE\Winmgmt
sa
MyLogin

You can also create a login using the T-SQL command for SQL server create login and user.

```
CREATE LOGIN MyLogin WITH PASSWORD = '123';
```

Note : Make sure to check the hostname if it is being resolved or not on your PC.

Reference to
https://medium.com/@bryan.nicholson
https://www.guru99.com/sql-server-create-user.html
https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/create-a-database-user?view=sql-server-ver15
https://github.com/GoogleCloudPlatform/iap-desktop
https://cloud.google.com/sdk/docs/install-sdk
https://www.liquidweb.com/kb/edit-host-file-windows-10/