

Group Membership Report Setup Guide

updated: 2020-02-24

version: 1.0

author: [hnacino](#)

G Suite Group / User Report

The purpose of this document is to describe the setup and configuration requirements to run a job that lists all the users in a given group. For nested groups, the job will recursively list all members in any groups found within groups. Setup / configuration is required in both Google Cloud Platform using the UI console at console.cloud.google.com, and in G Suite using the admin console at admin.google.com.

Disclaimer

This code is not an official Google product. The code is in an open-source project, which is available under the [Apache License, Version 2.0](#). You can use the code as a starting point and configure it to fit your requirements. You are responsible for ensuring that the environment and applications that you build on top of Google Cloud are properly configured and secured.

G Suite Group / User Report	1
Disclaimer	1
Overview	2
GCP Configuration	3
Step 1: Enable the admin SDK API	3
Step 2: Create GCP Service Account	4
Step 3: Create API Credentials	8
G Suite Configuration	9
Step 1: Create a custom admin role	9
Step 2: Create an account	10
Step 2: Update API Access settings	12
Step 3: Create a destination Drive Folder for the reports	13
Step 4: Grant 'Edit' access to the G Suite account created in step 1	14
Step 5: Grant 'Viewer' access to the destination folder for the appropriate user(s)	15
Gather configuration artifacts	16
Code Setup	16
Execution	16

Overview

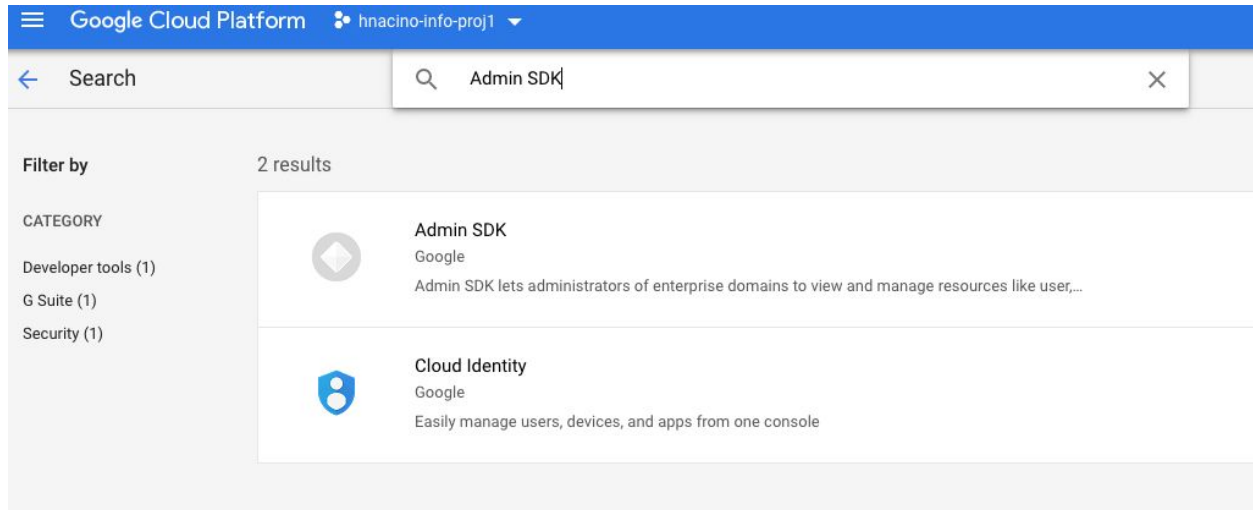
This document outlines the required steps needed to successfully run the G Suite Group / User Report. The high level steps are:

1. GCP configuration
 - a. Enable the admin SDK API
 - b. Create GCP Service Account
 - c. Create GCP API Credentials to create a client ID
2. G Suite configuration
 - a. Create a custom Admin role
 - b. Create an account and assign it to the custom Admin role
 - c. Update API Access settings with correct client ID and scopes
 - d. Create a Drive Folder as the destination for the reports
 - e. Grant 'Edit' access to the G Suite account created in 2(a)
 - f. Grant the appropriate end users 'View' access to the folder
3. Code Setup
4. Execution

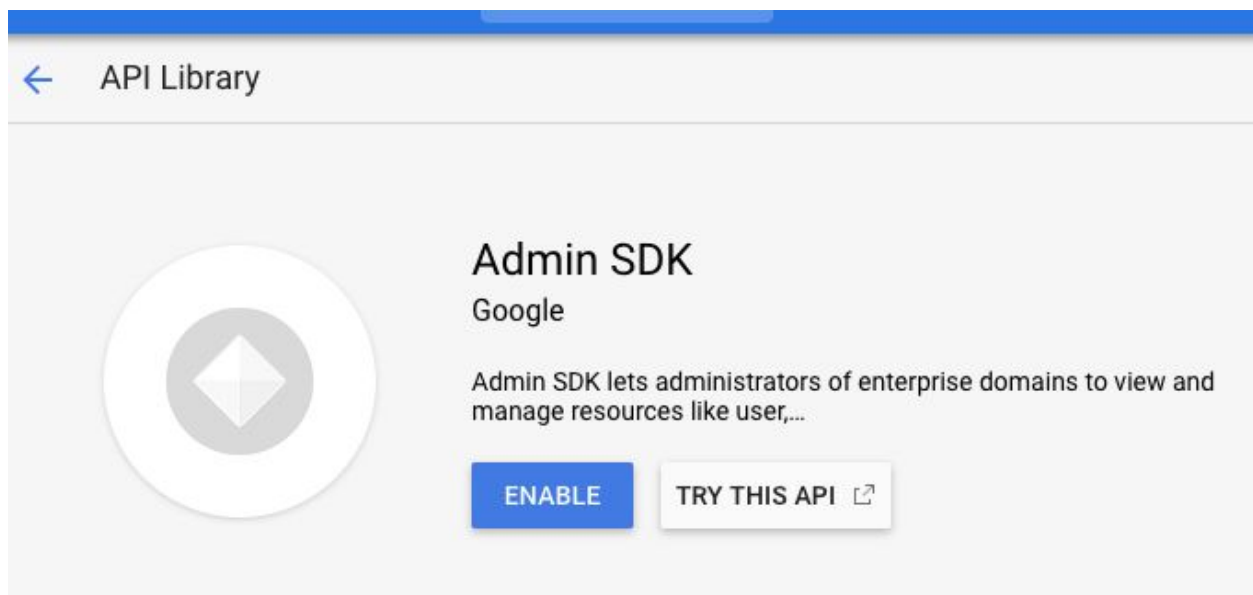
GCP Configuration

Step 1: Enable the admin SDK API

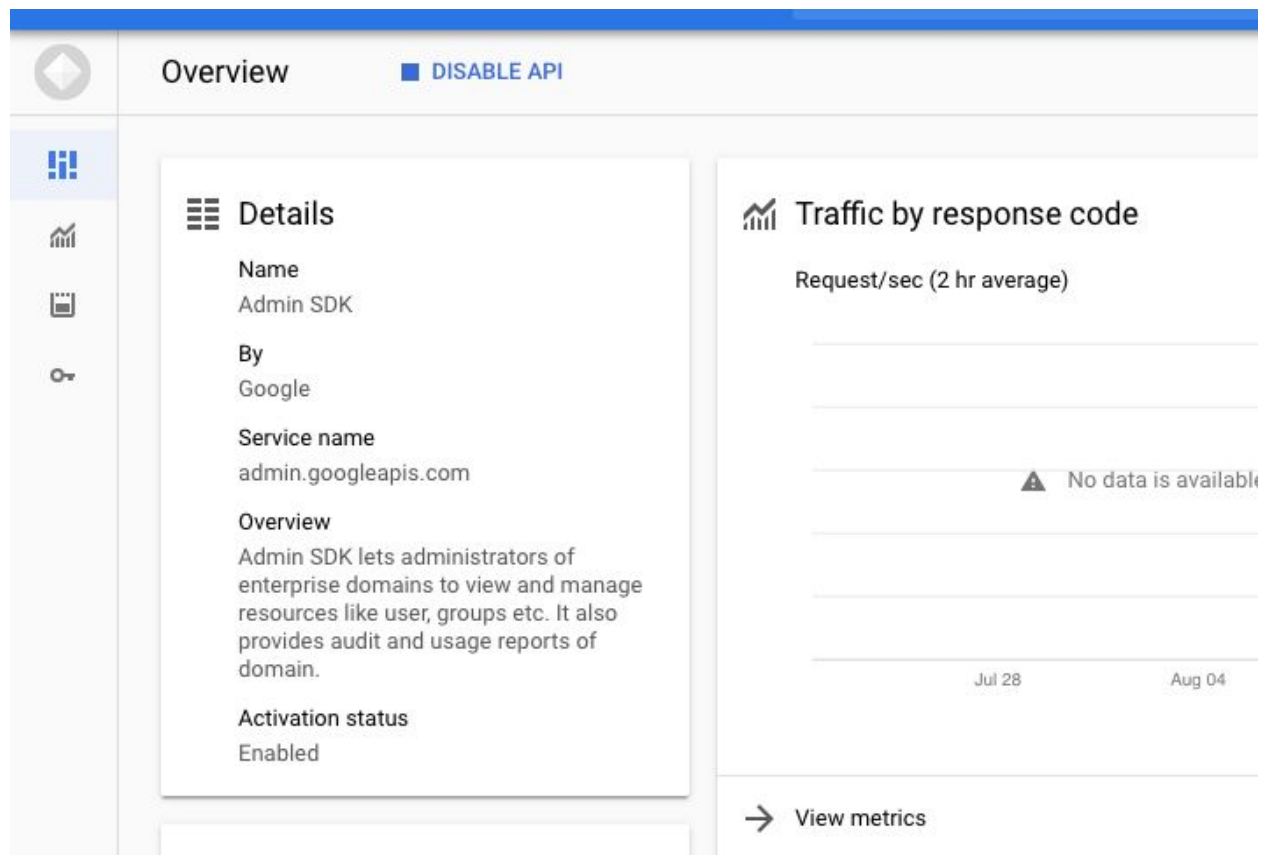
Login to console.cloud.google.com and select the appropriate GCP project. click on the top-left hamburger menu, scroll down to "**APIs & Services**", then "**Library**". In the library search box enter '**Admin SDK**', and you should see the following.



Click on the first result 'Admin SDK', and you will see the following.



Click '**Enable**', and you will see the following which confirms that API is enabled.



Step 2: Create GCP Service Account

Login to console.cloud.google.com and select the appropriate GCP project. click on the top-left hamburger menu, scroll down to "**IAM & Admin**", then "**Service Accounts**". In the displayed form click '**+CREATE SERVICE ACCOUNT**' in the top portion of the form.

In the following form, enter a service account name and click the '**CREATE**' button.

<div><div>IAM & admin</div><div><div>IAM</div><div>Identity & Organization</div><div>Organization policies</div><div>Quotas</div><div>Service accounts</div><div>Labels</div><div>Settings</div><div>Privacy & Security</div><div>Cryptographic keys</div><div>Identity-Aware Proxy</div></div></div>	<div>Create service account</div> <div><div>1 Service account details</div><div>2 Grant this service account access to project (optional)</div></div> <div><div>Service account details</div><div>Service account name<div>my-service-account</div><div>Display name for this service account</div></div><div>Service account ID<div>my-service-account</div><div>@james-nacino-sandbox.iam.gserviceaccount.</div><div>X</div><div>↺</div></div><div>Service account description<div></div><div>Describe what this service account will do</div></div><div><div>CREATE</div><div>CANCEL</div></div></div>
---	---

In the next form, do **NOT** add any roles and click the '**Continue**' button.

<div><div>IAM & admin</div><div><div>IAM</div><div>Identity & Organization</div><div>Organization policies</div><div>Quotas</div><div>Service accounts</div><div>Labels</div><div>Settings</div><div>Privacy & Security</div><div>Cryptographic keys</div></div></div>	<div>Create service account</div> <div><div>✓ Service account details</div><div>2 Grant this service account access to project</div></div> <div><div>Service account permissions (optional)</div><div>Grant this service account access to james-nacino-sandbox so that it has permission to complete specific actions on the resources in your project. Learn more</div><div><div>Select a role</div><div>+</div><div>ADD ANOTHER ROLE</div></div><div><div>CONTINUE</div><div>CANCEL</div></div></div>
--	--

In the next form, do **NOT** grant any users access to the service account. Click **Create key**, and choose the default key type, JSON and click **Create**

IAM & admin

IAM

Identity & Organization

Organization policies

Quotas

Service accounts

Labels

Settings

Privacy & Security

Cryptographic keys

Identity-Aware Proxy

Roles

Audit Logs

Create service account

Service account details

Grant this service account access to project (optional)

Grant users access to this service account (optional)

Grant access to users or groups that need to perform actions as this service account.
[Learn more](#)

Service account users role

?

Grant users the permissions to deploy jobs and VMs with this service account

Service account admins role

?

Grant users the permission to administer this service account

Create key (optional)

Download a file that contains the private key. Store the file securely because this key can't be recovered if lost. However, if you are unsure why you need a key, skip this step for now.

+ CREATE KEY

DONE

CANCEL

Create key (optional)

Download a file that contains the private key. Store the file securely because this key can't be recovered if lost. However, if you are unsure why you need a key, skip this step for now.

Key type

☒ JSON

Recommended

☐ P12

For backward compatibility with code using the P12 format

CREATE

CANCEL

After creating the key, the private key will be downloaded to your computer. Store it securely, so you can retrieve it later. We will **need** the key file later. Click **CLOSE** and then click **DONE**

Private key saved to your computer



2d.json allows access to your cloud resources, so store it securely. [Learn more](#)

CLOSE

The service account has been successfully created. Open the newly created service account in 'Edit' mode to '**Enable G Suite Domain wide delegation**' as shown below. Click '**Save**' to save the change.

← gsuite-svc-account [EDIT](#) [DELETE](#)

Service account details

Name

gsuite-svc-account

Description

Email

gsuite-svc-account@james-nacino-sandbox.iam.gserviceaccount.com

Unique ID

104874638144634736571

Disable service account

Disabling your account allows you to preserve your policies without having to delete it.

✓ Account currently active

☒ Enabled

☒ Enable G Suite Domain-wide Delegation

Allows this service account to be authorized to access all users' data on a G Suite domain without manual authorization on their parts. [Learn more](#)

Client ID

104874638144634736571



[^ HIDE DOMAIN-WIDE DELEGATION](#)

Keys

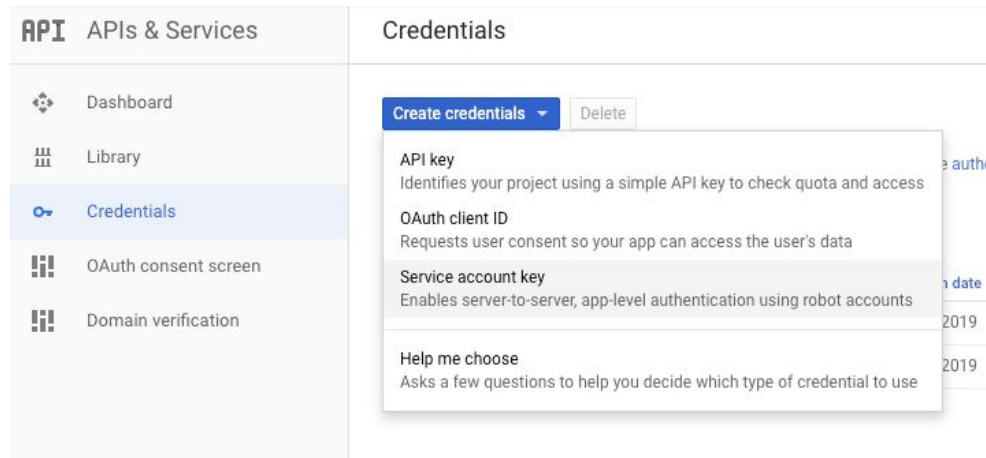
Key ID

db5646ba29936440abdef55439bbefbbca454182

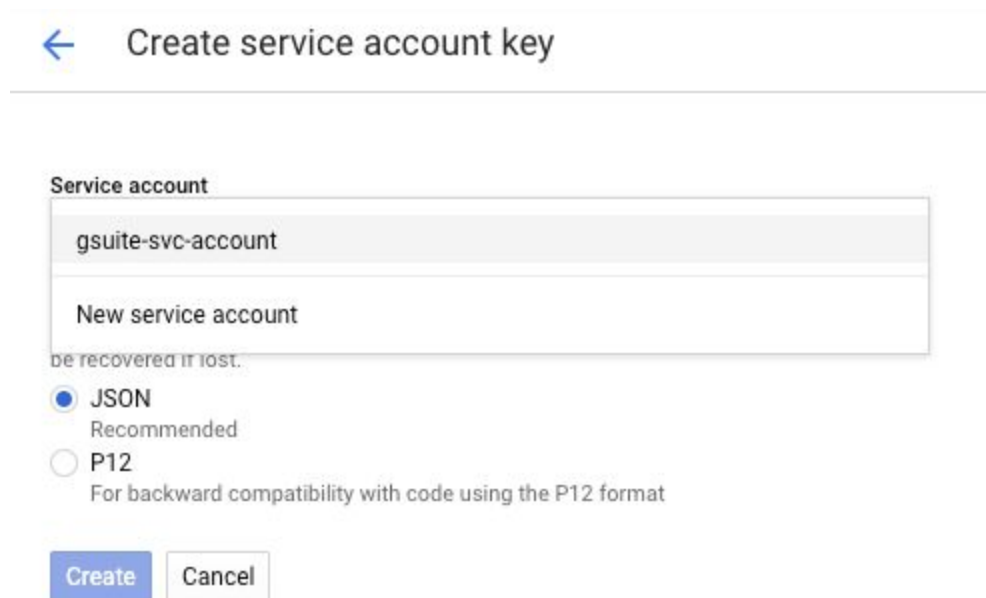


Step 3: Create API Credentials

Login to console.cloud.google.com and select the appropriate GCP project. click on the top-left hamburger menu, scroll down to "**APIs & Services**", then "**Credentials**". In the form click '**Create credentials**' and select '**Service account key**' as shown below.



Select the service account that you created in the previous step and click **Create**



In the main 'Credentials' page, you should see the newly created Client ID listed as shown below. Please note the 'Client ID', you will need it later to configure G Suite API access.

API

Credentials

Credentials

OAuth consent screen



Domain verification

Create credentials

Delete

Create credentials to access your enabled APIs. For more information, see the [authentication documentation](#).

OAuth 2.0 client IDs

<input type="checkbox"/>	Name	Creation date	Type	Client ID	
<input type="checkbox"/>	Client for gsuite-svc-account	Jun 7, 2017	Service account client	115991248861495489380	 

G Suite Configuration

Step 1: Create a custom admin role

You will create a custom Admin role that has API Admin privileges to read Users and Groups. Login to [admin.google.com](#). Click on the top-left hamburger menu, scroll down to "**Account**", then "**Admin roles**". On the resulting page, click '**CREATE NEW ROLE**'. In the form shown below, enter a name for the new role and click '**Create**'.

Super Admin

Role

Ac

ASS

Create New Role

Name

Directory Reader API

Description

CANCEL

CREATE

In the resulting page, there are 2 sets of privileges, admin console and admin api, that can be enabled. We will only set the admin api privileges to enable **Users:Read** and **Groups:Read** as shown below. Then click **'Save'**

Admin roles

CREATE A NEW ROLE

Directory Reader API

System Roles ?

Super Admin

Groups Admin

User Management Admin

Help Desk Admin

Services Admin

User Created Roles ?

Admins

Privileges

Admin API Privileges ?

▸

☐

Organization Units

▼

☐

Users

☐ Create

☒ Read

▸

☐ Update

☐ Delete

☐

Groups

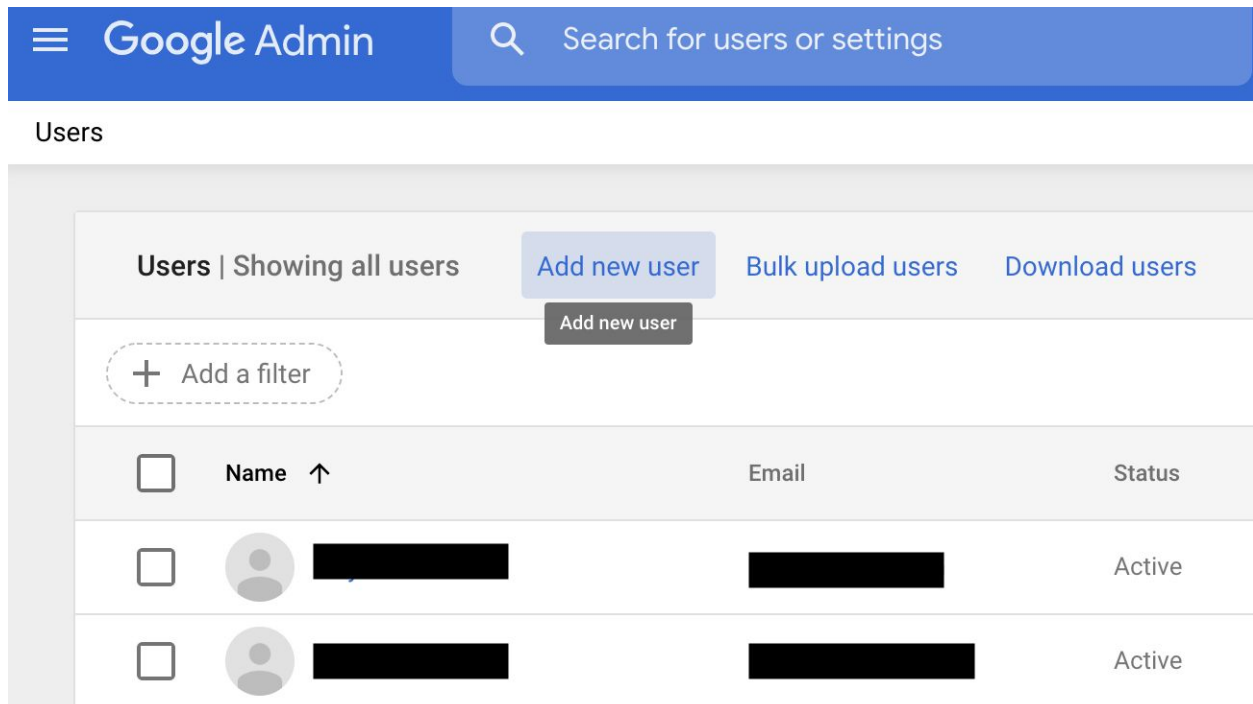
☐ Create

☒ Read

Step 2: Create an account

You will create a user and assign the user to the custom Admin role created in the previous step. We need a G Suite account for delegation that has read access on users and groups. The application generating the reports will use this account for delegated access to the users and groups.

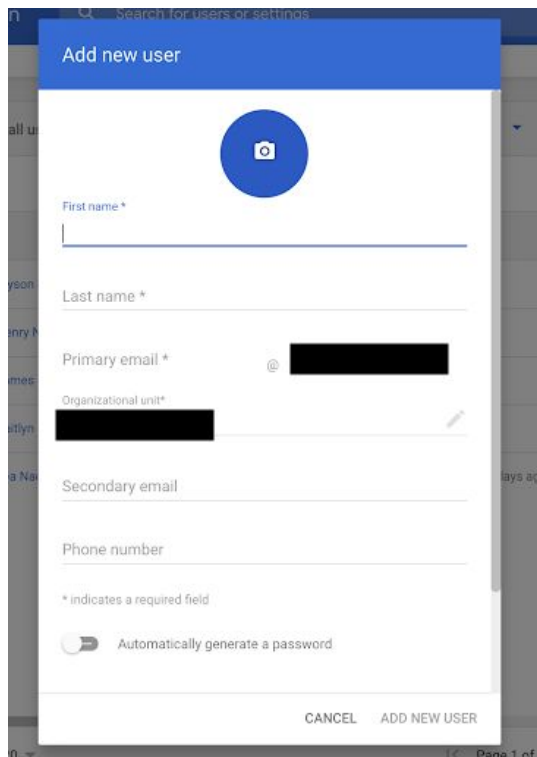
Login to admin.google.com. Click on the top-left hamburger menu, scroll down and click on "Home". Then click on 'Users'. You should see the following page, then click 'Add new user'.



The screenshot shows the Google Admin console's 'Users' page. At the top, there's a blue header with the 'Google Admin' logo and a search bar. Below the header, the 'Users' section is active, displaying 'Showing all users'. There are three buttons: 'Add new user' (highlighted), 'Bulk upload users', and 'Download users'. A dashed box highlights a '+ Add a filter' button. Below this is a table with columns for selection (checkbox), Name (with an upward arrow), Email, and Status. Two user entries are visible, both with redacted names and emails, and a status of 'Active'.

<input type="checkbox"/>	Name ↑	Email	Status
<input type="checkbox"/>	[Redacted]	[Redacted]	Active
<input type="checkbox"/>	[Redacted]	[Redacted]	Active

Fill out the Add new user form with the appropriate details, and click 'ADD NEW USER'



The screenshot shows the 'Add new user' form. It has a blue header with a camera icon for profile picture. The form fields are: 'First name *', 'Last name *', 'Primary email *' (with a redacted email address), 'Organizational unit*' (with a redacted unit name), 'Secondary email', and 'Phone number'. A note states '* indicates a required field'. There is a toggle switch for 'Automatically generate a password'. At the bottom, there are 'CANCEL' and 'ADD NEW USER' buttons.

First name *

Last name *

Primary email * @ [Redacted]

Organizational unit* [Redacted]

Secondary email

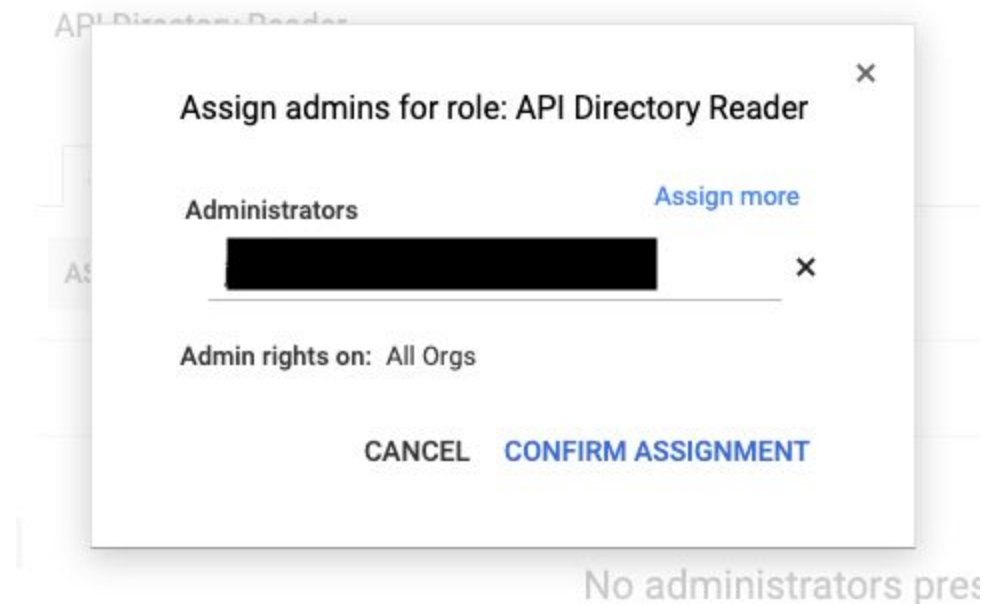
Phone number

* indicates a required field

☐ Automatically generate a password

CANCEL ADD NEW USER

Next, assign the newly created account to the custom admin role created in Step 1. Click on the top-left hamburger menu, scroll down to "**Account**", then "**Admin roles**". Then select the custom admin role that was created in Step 1 and click '**ASSIGN ADMINS**'. In the next form, type in the email address of the user account created in Step 2, and click '**CONFIRM ASSIGNMENT**'



Step 2: Update API Access settings

In this step, we will configure API security settings that enable specific OAuth scopes for the ClientID created in Step 3 of the GCP Configuration section above.

In admin.google.com, click on the top-left hamburger menu, scroll down to "**Security**", then "**Settings**". In the resulting page, scroll down and click on '**Advanced settings**'. Then click on '**Manage API client access**'. Leave this browser tab open.

In a different browser tab, login to console.cloud.google.com and select the appropriate GCP project. click on the top-left hamburger menu, scroll down to "**APIs & Services**", then "**Credentials**". You should see the OAuth clientIDs for your project as shown below. **Copy** the Client ID for the service account you created earlier.

API APIs & Services

Dashboard
Library
Credentials
OAuth consent screen
Domain verification

Create credentials
Delete

Create credentials to access your enabled APIs. For more information, see the [authentication documentation](#).

OAuth 2.0 client IDs

<input type="checkbox"/>	Name	Creation date	Type	Client ID
<input type="checkbox"/>	Apps Script	Aug 7, 2019	Web application	463104292293-lqv1rtkvea063mscpfvbojlve0l
<input type="checkbox"/>	Client for gsuite-svc-account	Aug 7, 2019	Service account client	104874638144634736571

Navigate back to admin.google.com browser tab, and **paste** the clientID in the Client Name field, and enter the following scopes in the One or More API Scopes field as shown below:

<https://www.googleapis.com/auth/drive>, <https://www.googleapis.com/auth/admin.directory.user>, <https://www.googleapis.com/auth/admin.directory.group>

Security

Manage API client access

Developers can register their web applications and other API clients with Google to enable access to these registered clients to access your user data without your users having to individually give conser

Authorized API clients

The following API client domains are registered with Google and author

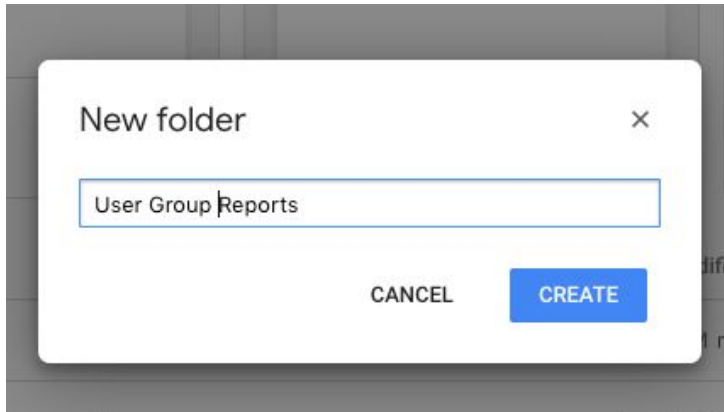
Client Name
10487463814463473l
Example:
www.example.com

One or More API Scopes
<https://www.googleapis.com/auth/drive, http>
Authorize

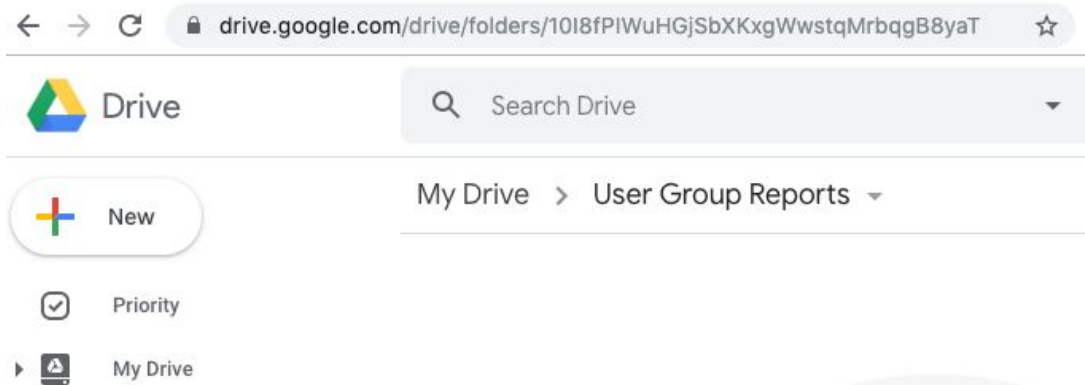
Example: <http://www.google.com/calendar/feeds/> (comma-delimited)

Step 3: Create a destination Drive Folder for the reports

Navigate to your Google Drive at drive.google.com. Create a new folder by clicking **'New'** then **'Folder'**. Enter an appropriate name as shown below and click the **'Create'** button.

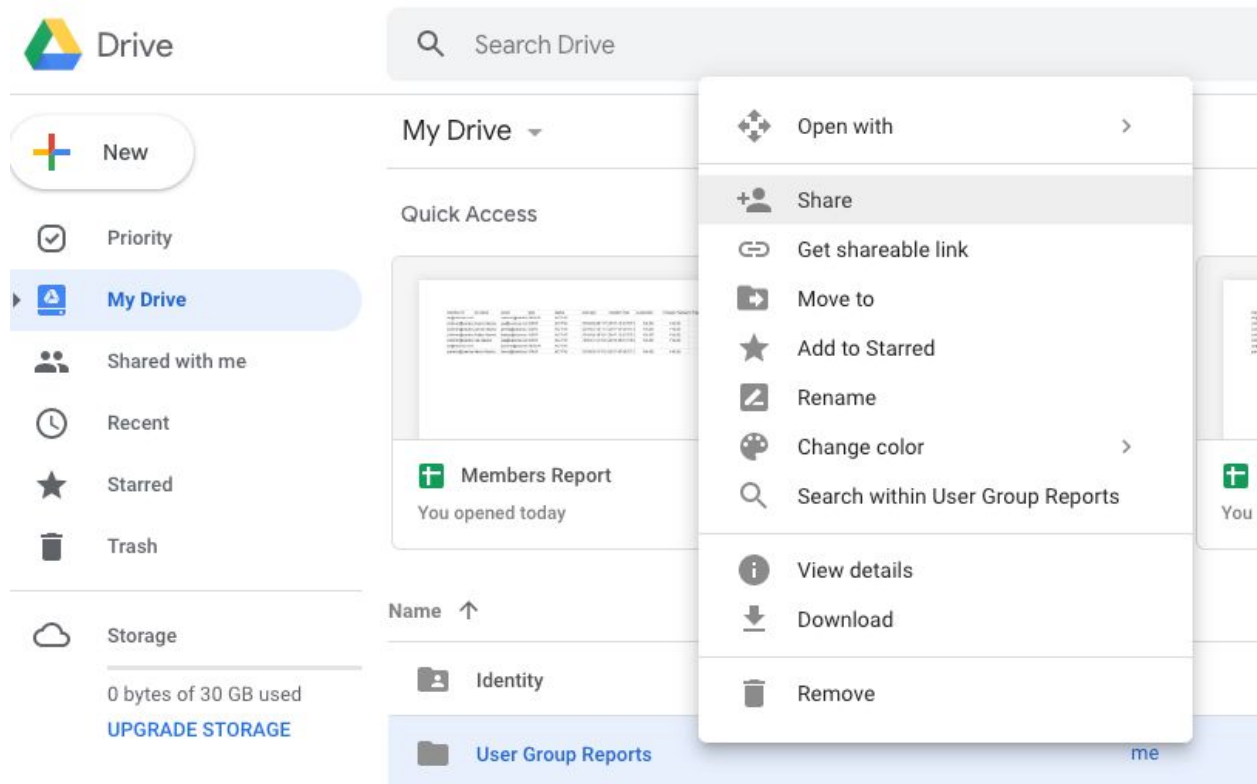


After the folder has been successfully created. Navigate to the newly created folder, and make a note of the folder id in the url. The folder id is the string of characters following the ../folders/. In the example below the folder id is: 1018fPIWuHGjSbXKxgWwstqMrbqgB8yaT

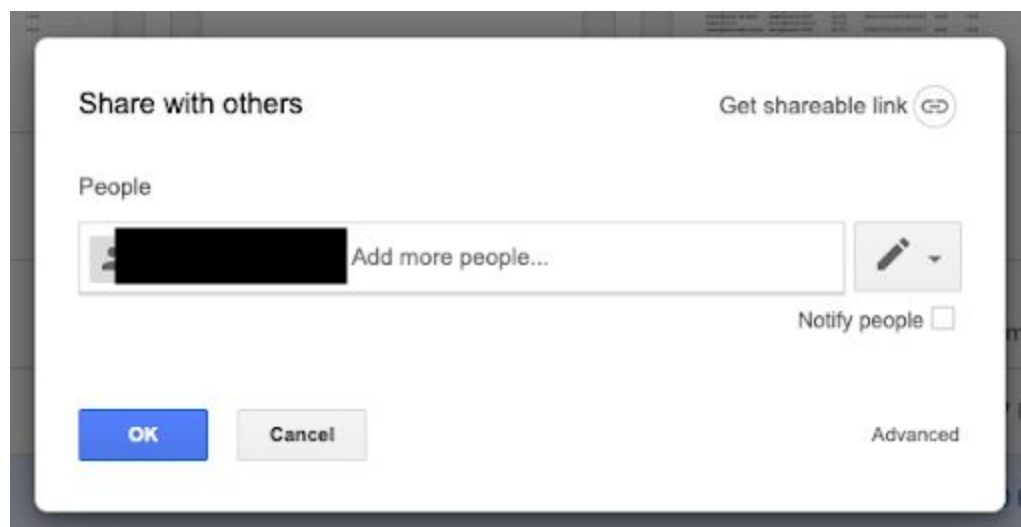


Step 4: Grant 'Edit' access to the G Suite account created in step 1

In Google Drive, locate and **right-click** on the folder, and then click **Share** as shown below.



In the resulting form, enter the user account created in step 1, uncheck '**Notify people**' and click '**OK**'



Step 5: Grant 'Viewer' access to the destination folder for the appropriate user(s)

By default, the person who created the Google Drive folder in Step 3 above is the only user who has rights to view the files that are stored in that folder. Users who require access to the Group Membership report should follow their defined approval process to get approval to access this content. The approved users should then be granted 'Viewer' access to the destination Google Drive folder.

Gather configuration artifacts

You will need to share the following configuration artifacts with the person deploying / running the report job.

1. The GCP Service account private key file created and downloaded in GCP Configuration:Step 2
2. The email of the user account created in G Suite Configuration:Step 2
3. The folder id of the Drive folder created in G Suite Configuration: Step 3

Code Setup

The Group Membership Report script can be run from an on-premises computer or a Google Compute Engine instance running in Google Cloud Platform by completing the following:

1. Clone the github repository

```
git clone https://github.com/hanknac/admin-sdk.git
```

2. Navigate to the admin-sdk directory and install python dependencies

```
cd admin-sdk  
pip3 install -r requirements.txt
```

3. Copy the service account private key file to the admin-sdk directory
4. Update the config.json file using your favorite text editor (ie vim) with your:
 - a. GCP service account key file
 - b. GSuite user account that had API access
 - c. Destination folderID for the report

Execution

```
cd admin-sdk  
python3 members.py --group group@abc.com
```