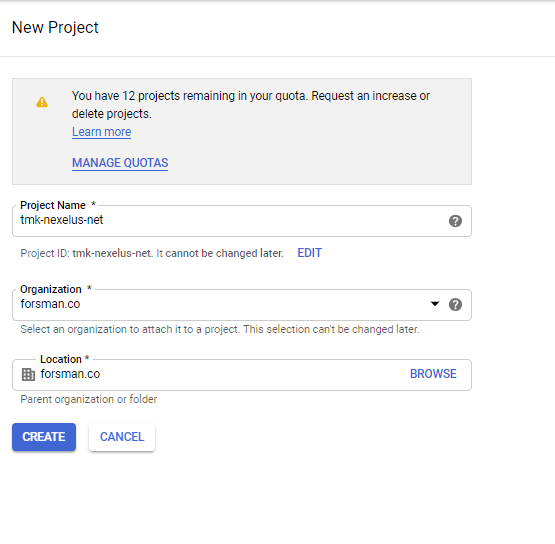
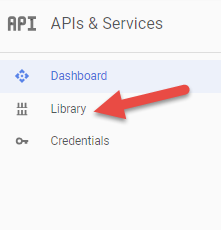
Configure New DCM Client:

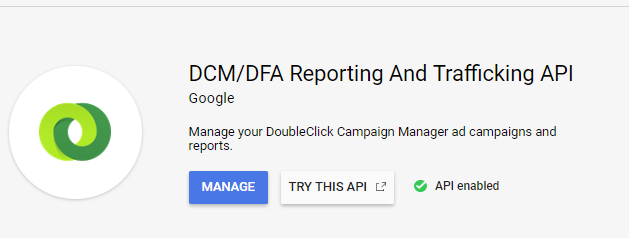
* Goto <https://console.developers.google.com> and sign in using the clients credentials
* If no project is created, then create a new project e.g. “tmk-nexelus-net”



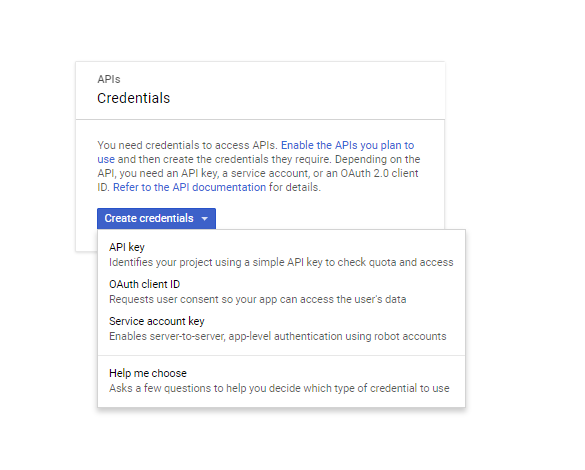
* Select any organization/location from the dropdown, normally it will be auto selected.
* Goto Libraries



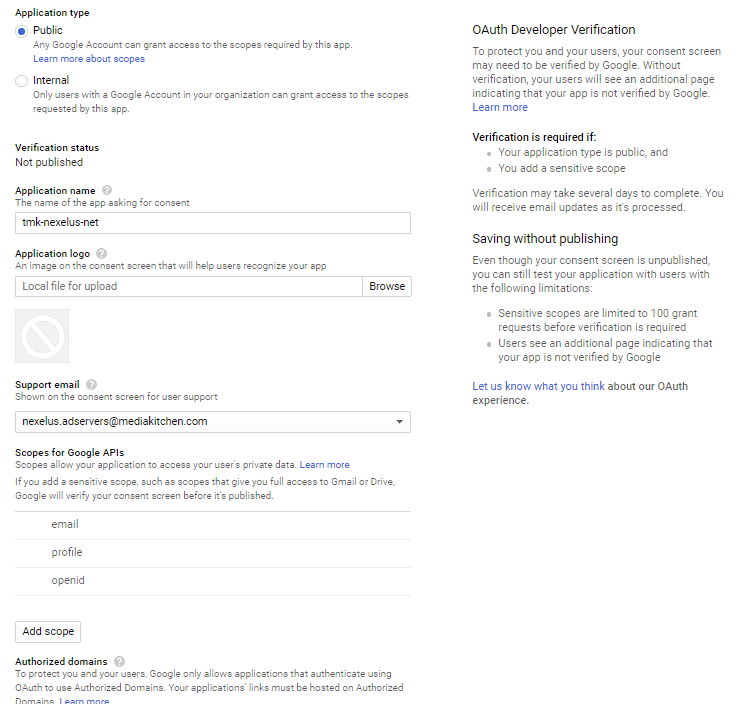
* Search for “dfa” and activate the API. It should look like below once enabled.



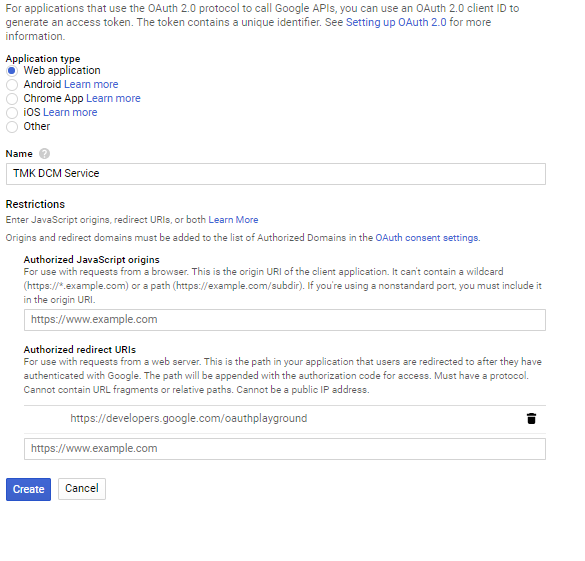
* After enabling the API, we need to create credentials for IPC to use the API
* Goto credentials and create new OAuth Client ID



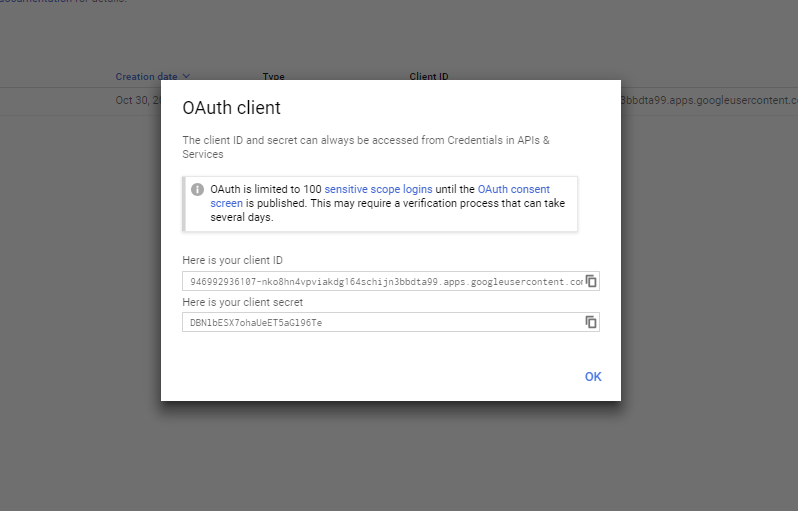
* When creating the credentials first time it will ask to create a product first, go ahead and create that.



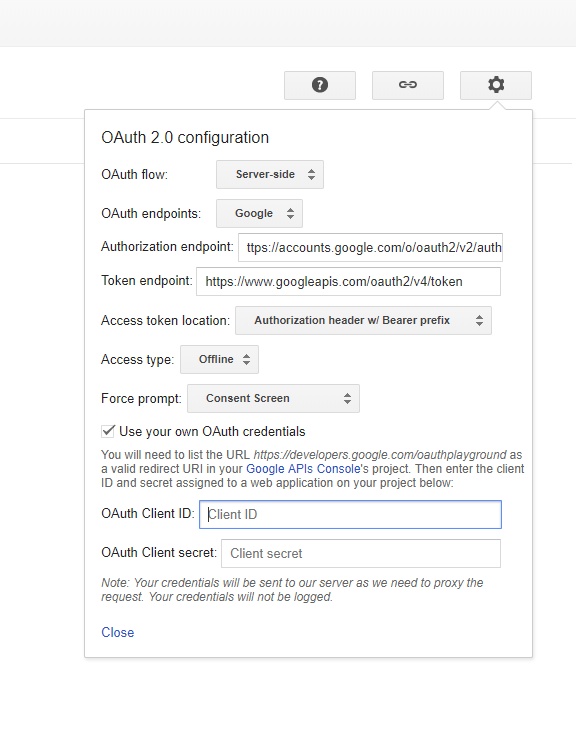
* You can give any application name but preferably the same name as project name.
* After creating product, go ahead and create the service key



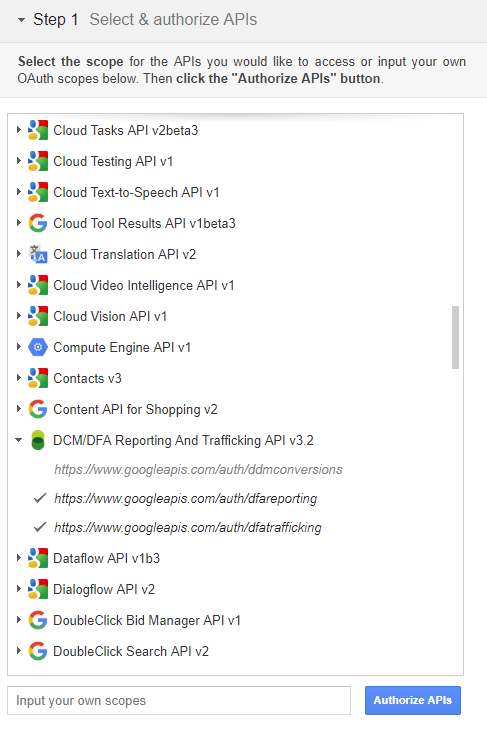
* Make sure to create as “Web Application”.
* Give name
* Make sure to add “https://developers.google.com/oauthplayground” as an Authorized redirect URIs
* Once created console will provide you with the client id and client secret



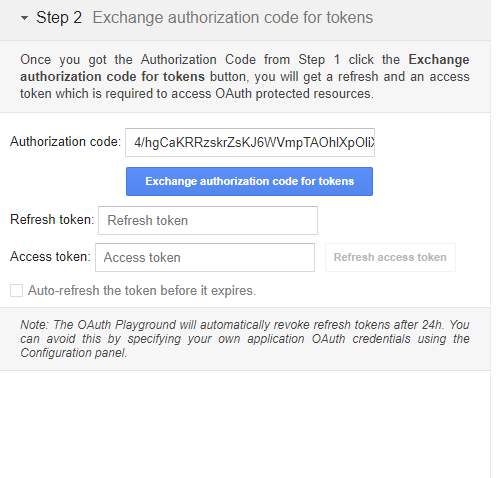
* Make note of these two things as we would need this in configuring IPC and fetching OAuth 2.0 refresh token and access key.
* Now goto <https://developers.google.com/oauthplayground/>
* In the settings menu on the right hand side, make sure the following are selected as:



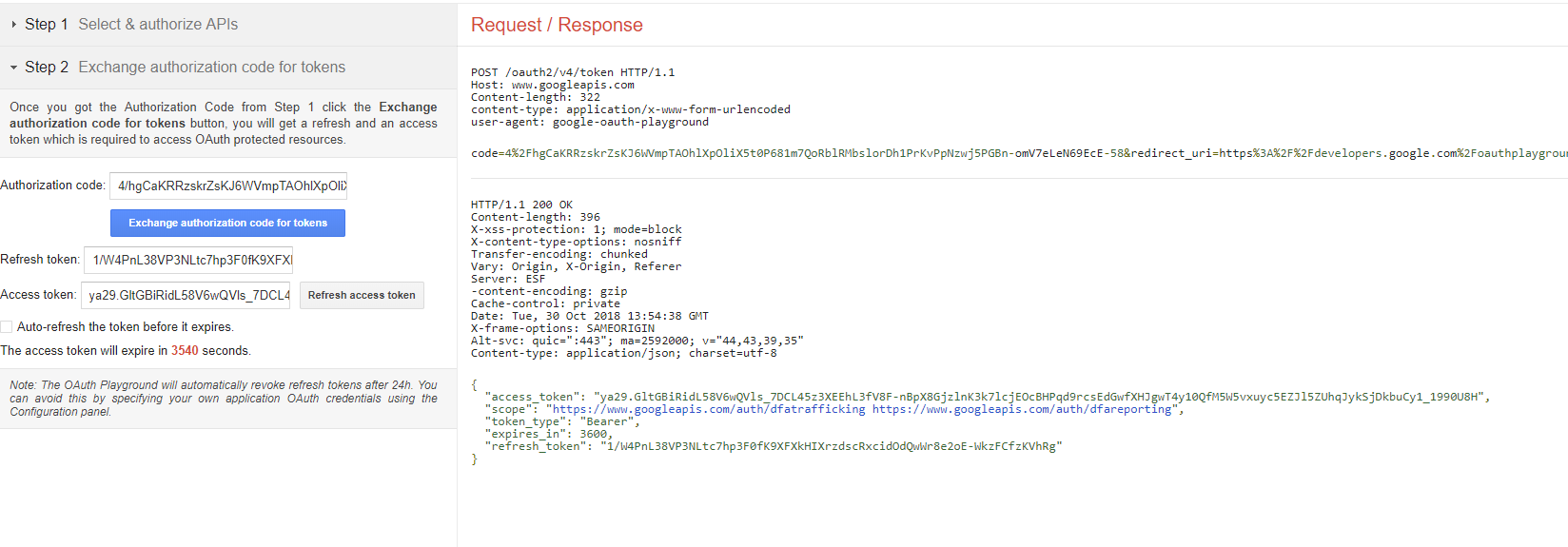
* Provide the client id and client secret from the earlier step.
* In right hand side select 2 APIs to authorize and press “Authorize APIs”



* Google will redirect to re-login for permissions, go ahead and do that using the google credentials.
* Once done you will be redirected back to oath playground and authorization code will be provided



* Go ahead and press “Exchange authorization code for tokens”.
* Once done, you will be provided with refresh token and access token.



* Make sure you note these or the JSON on the right side. We will need these for configuring IPC as well.
* Now go ahead and connect to the server where IPC is deployed and make your way to the IPC root directory. (Please note that there might be different versions of IPC deployed, make sure you are in the correct directory by confirming where client’s Nexelus IPC is pointed to.)
* If the Nexelus’s IPC pointing is changed in future, then the next step would need to be repeated as well.
* Inside the IPC root folder create a folder with client name
* Inside this folder create a file with name “Google.Apis.Auth.OAuth2.Responses.TokenResponse-{project-name-from-above-steps}”. The extensions will be “TokenResponse-{project-name-from-above-steps}”
* Open the file in notepad.
* From the refresh token step. Copy the JSON created and paste it in the file.
* Please note the path of this file. We would need to use it in the scripts.
* After this run the scripts provided. Make sure to add the information required in the declared variables.