

# Sending Mail via GMail API

📄 Goto [console.cloud.google.com](https://console.cloud.google.com) and create an account

Try Google Cloud for free

### Step 1 of 2 Account Information

**Anubhav Patrick**  
anubhav.patrick@glindia.com [SWITCH ACCOUNT](#)

**Good news!** You're eligible for an additional \$100.00 in Free Trial credits for a total of \$400.00. You'll receive these credits within 24 hours of completing signup.

**Country**  
India

**What best describes your organization or needs?**  
Please select  
Startup

**Terms of Service**  
☒ I have read and agree to the [Google Cloud Platform Terms of Service](#), [Supplemental Free Trial Terms of Service](#), and the terms of service of [any applicable services and APIs](#).  
Required to continue

[CONTINUE](#)

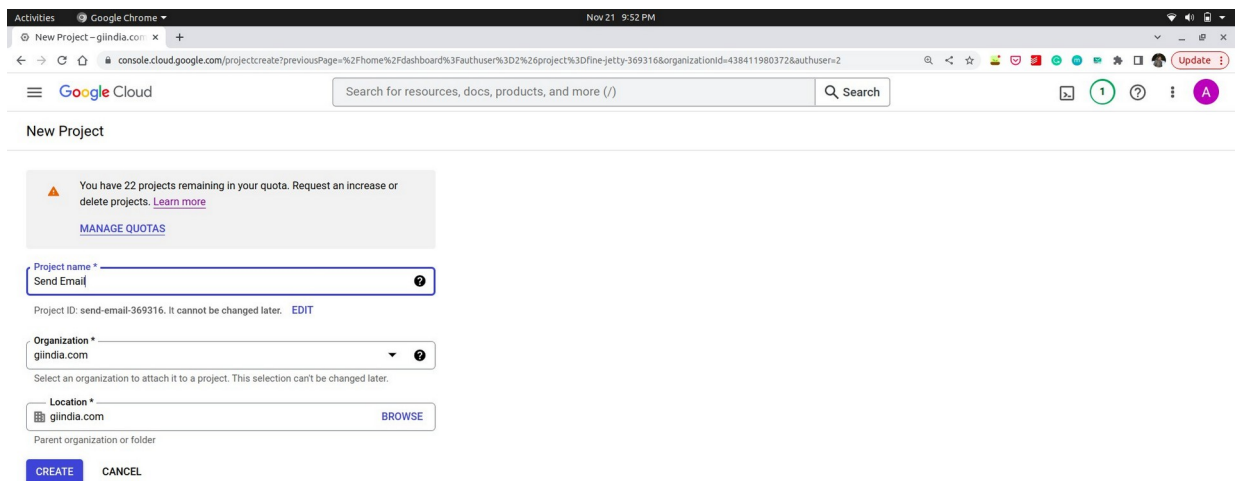
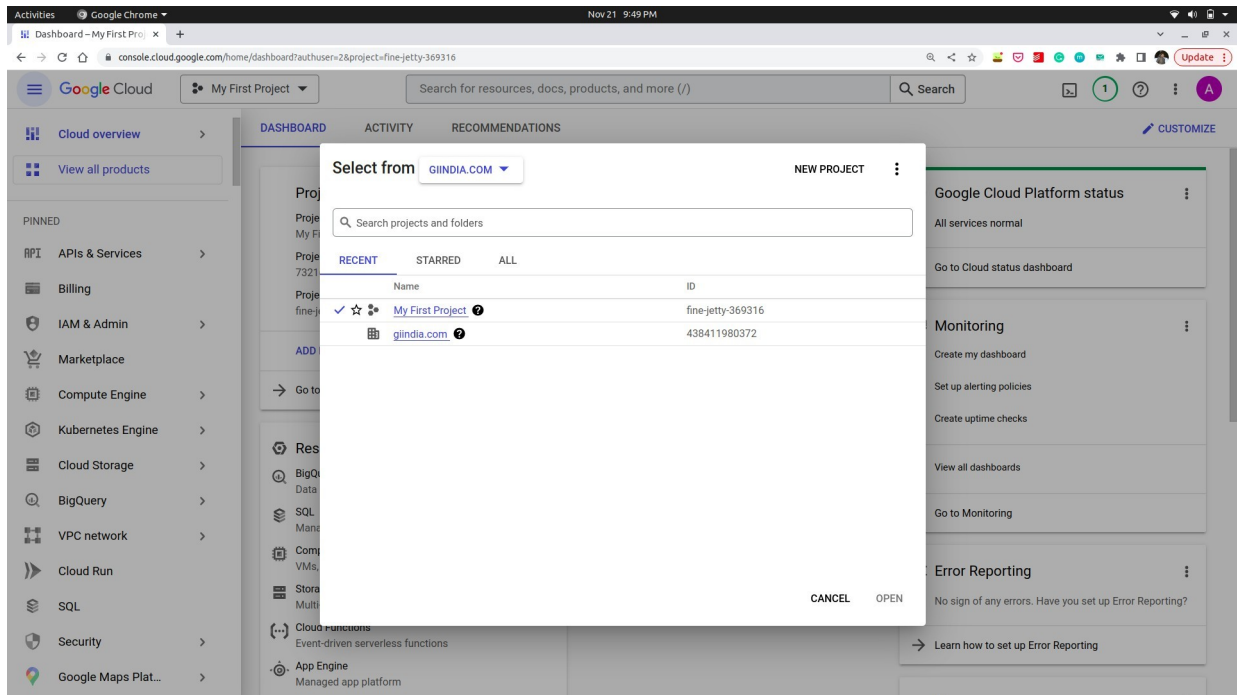
**Verify your card to get started**  
Your card is used to verify you're not a robot. Don't worry, it won't be charged until you manually upgrade to a paid account.

**Access all Google Cloud products**  
Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.

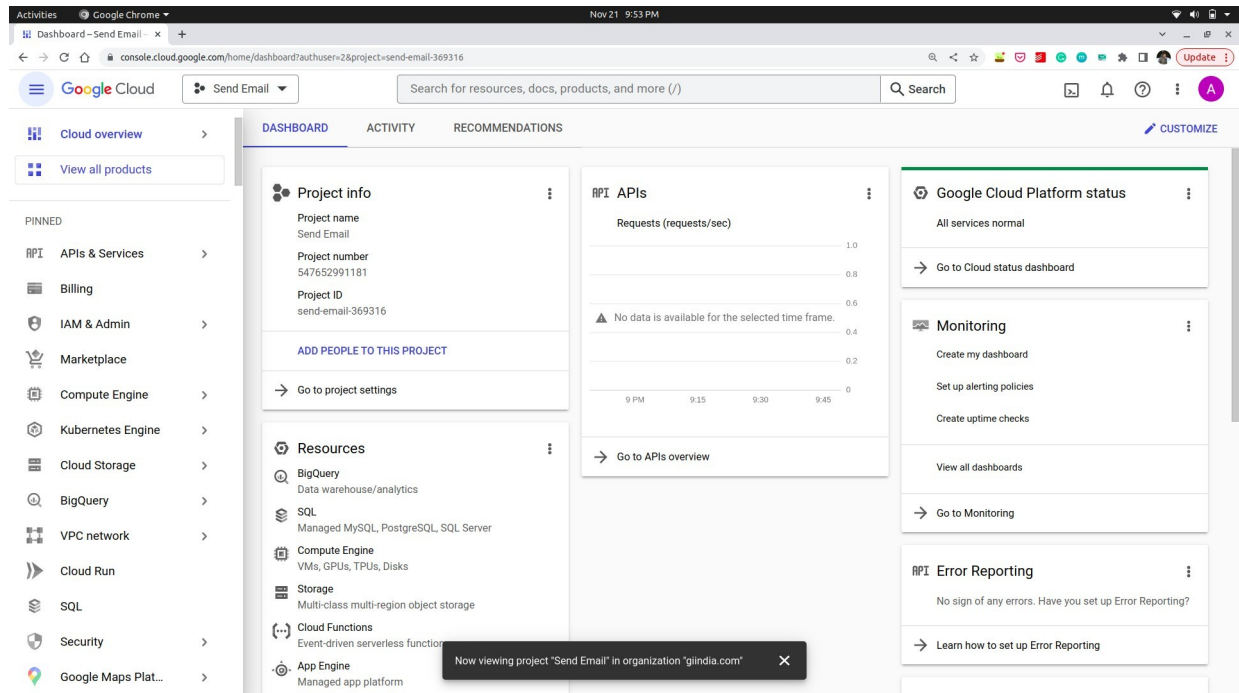
**\$300 credit for free**  
Put Google Cloud to work with \$300 in credit to spend over the next 90 days.

📄 Complete the setup of account

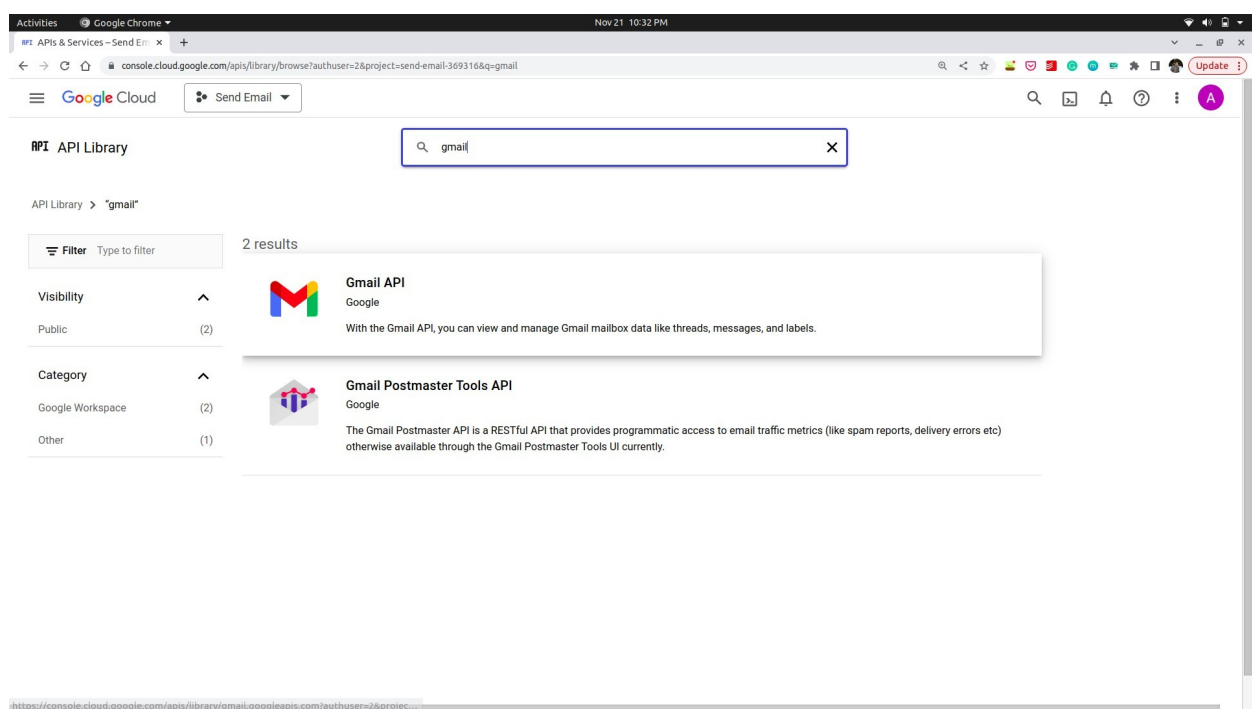
## Create a New Project by navigating the top pane



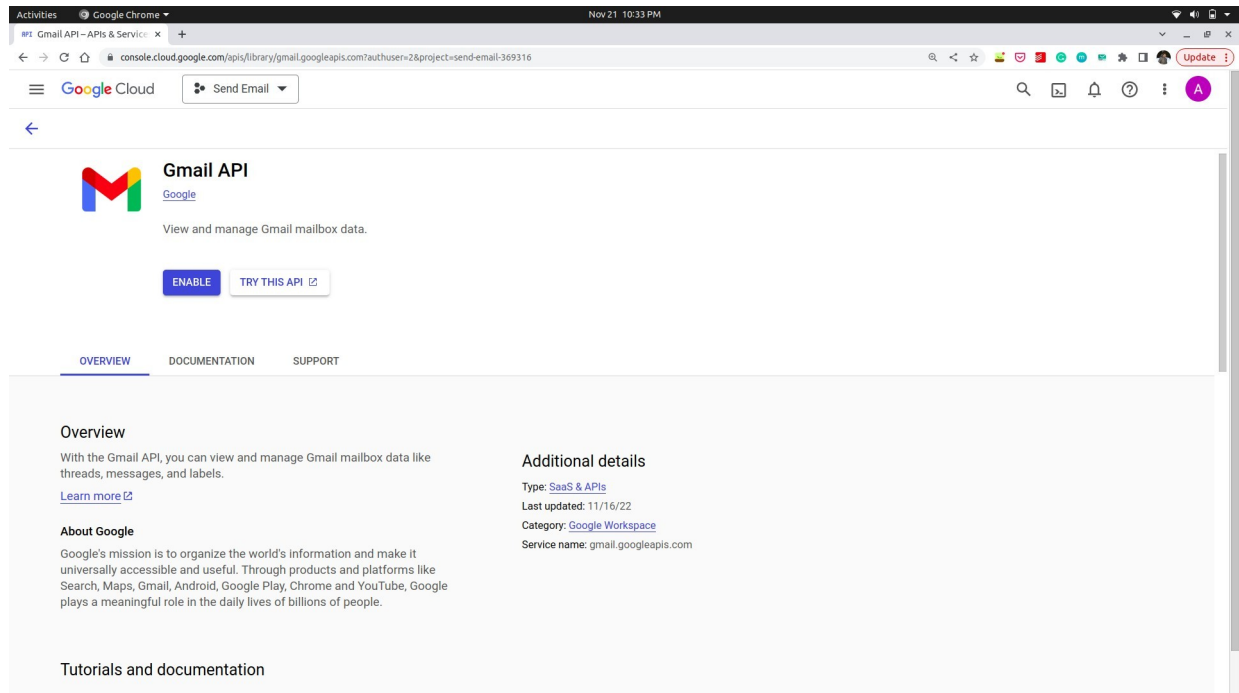
## Change the project to new project from top pane



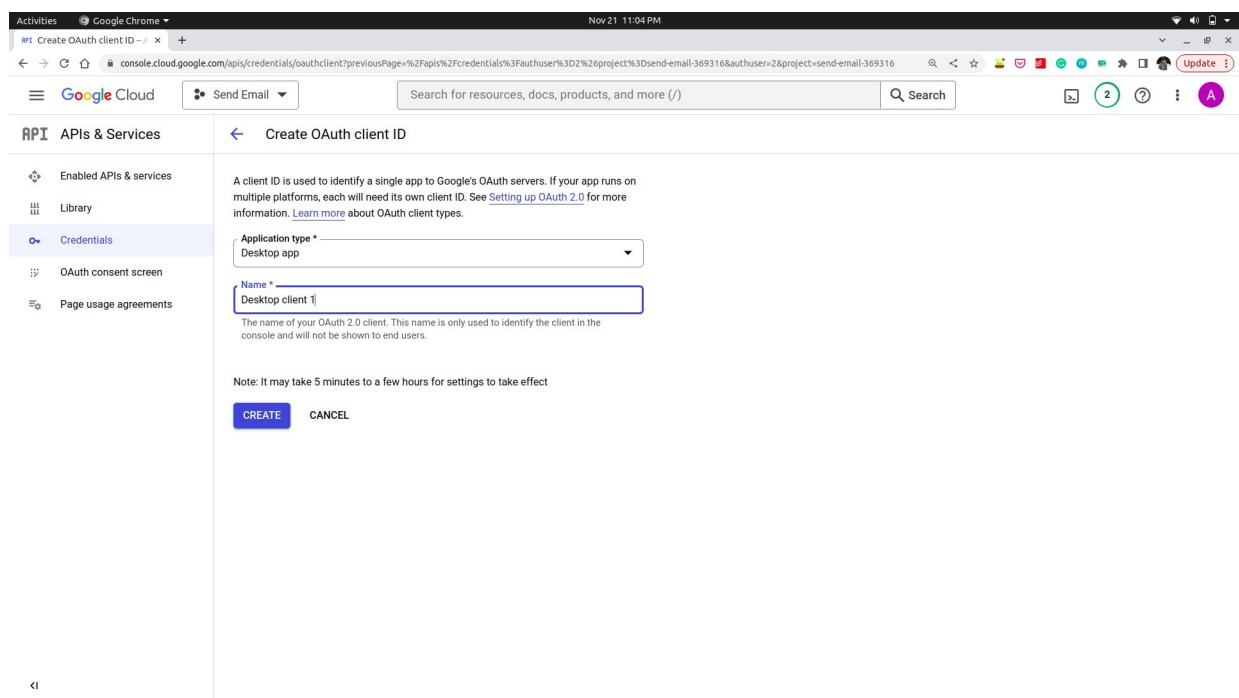
## Goto *APIs & Services* from the left pane then select *Enabled API & Services*. Then select *Enable API & Services*



Search for *Gmail* \* Then select *Enable*



Click *Credentials* → then click *Create Credentials*



Select Application type → *Desktop app*

Give the app a relevant name or choose the default name. Click on *Create*

Download the client secret in *.json* format and rename it to *client\_secrets.json* . **The client\_secret.json file contains critical account information and should not be shared with anyone.**

Create a project directory and change directory

```
mkdir gmail_api cd gmail_api
```

❏ Create a python virtual environment and activate it

```
python -m venv venv/ source venv/bin/activate
```

❏ Install the gmail library inside the virtual environment

```
pip install --upgrade google-api-python-client google-auth-http2 google-auth-oauthlib
```

❏ Move the client secret (*client\_secrets.json*) downloaded in step 8 to *gmail\_api/* directory and **do not share this file anywhere else** since your gmail account can be accessed by anyone using these credentials.

❏ Create a new python file inside *gmail\_api/* titled *send\_mail.py*:

```
'''
This module sends emails with attachments to the participants
Reference - https://developers.google.com/gmail/api/quickstart/python
'''

import os

from google.auth.transport.requests import Request
from google.oauth2.credentials import Credentials
from google_auth_oauthlib.flow import InstalledAppFlow
from googleapiclient.discovery import build
from googleapiclient.errors import HttpError
from email.mime.text import MIMEText
import base64

# If modifying these scopes, delete the file token.json.
SCOPES = ['https://www.googleapis.com/auth/gmail.send']

def authentication():
    creds = None
    # The file token.json stores the user's access and refresh tokens,
    and is
    # created automatically when the authorization flow completes for the
    first
    # time.
```

```

if os.path.exists('token.json'):
    creds = Credentials.from_authorized_user_file('token.json',
SCOPES)
# If there are no (valid) credentials available, let the user log in.
if not creds or not creds.valid:
    if creds and creds.expired and creds.refresh_token:
        creds.refresh(Request())
    else:
        flow = InstalledAppFlow.from_client_secrets_file(
            'client_secrets.json', SCOPES)
        creds = flow.run_local_server(port=0)
    # Save the credentials for the next run
    with open('token.json', 'w') as token:
        token.write(creds.to_json())
return creds

def prepare_and_send_email(recipient, subject, message_text):
    """Prepares and send email with attachment to the participants
    """
    creds = authentication()

    try:
        # Call the Gmail API
        service = build('gmail', 'v1', credentials=creds)

        #create message
        msg = create_message('anubhav.patrick@giindia.com', recipient,
subject, message_text)
        send_message(service, 'me', msg)

    except HttpError as error:
        # TODO(developer) - Handle errors from gmail API.
        print(f'An error occurred: {error}')

def create_message(sender, to, subject, message_text):
    """Create a message for an email.

    Args:
        sender: Email address of the sender.
        to: Email address of the receiver.
        subject: The subject of the email message.
        message_text: The text of the email message.

    Returns:
        An object containing a base64url encoded email object.
    """

```

```

message = MIMEText(message_text)
message['from'] = sender
message['to'] = to
message['subject'] = subject
return {'raw':
base64.urlsafe_b64encode(message.as_string().encode()).decode()}

def send_message(service, user_id, message):
    """Send an email message.

    Args:
        service: Authorized Gmail API service instance.
        user_id: User's email address. The special value "me"
        can be used to indicate the authenticated user.
        message: Message to be sent.

    Returns:
        Sent Message.
    """
    try:
        message = (service.users().messages().send(userId=user_id,
body=message)
                    .execute())
        print('Message Id: %s' % message['id'])
        return message
    except HttpError as error:
        print('An error occurred: %s' % error)

if __name__ == '__main__':
    prepare_and_send_email('anubhavpatrick@gmail.com', 'Greeting from
Global Infoventures', 'This is a test email for our upcoming app')

```

- ☐ Run the python file. At the first time you will be prompted to enter you Google account credentials. **Use the same credentials** that you used to sign up in Google Cloud.

chooseaccount?response\_type=code&client\_id=547652991181-5j54ef3tkfo1ik0n4icnbt-pofa8pqtcp.apps.googleusercontent.com&redirect\_uri=http%3A%2F...

 Sign in with Google

Choose an account

to continue to [Safety Equipment Detector](#)



Anubhav Patrick  
anubhavpatrick@gmail.com