Requirements

Before installation, your server must have following requirements to run the script properly

- PHP v8.1
- PHP Mbstring Extension
- PHP PDO Extension
- PHP FileInfo Extension
- PHP JSON Extension
- PHP CURL Extension
- PHP symlink() function
- PHP file_get_contents() function
- Support for MySQL v5.7
- Apache Server Recommended (nginx configuration and installation up to you, we don't provide support on that)

Installation Instructions

Follow the steps as explained, and you should be all set. If you face any challenges with installtion steps, reach out to us for a support.

Setup overview:

- **Step 1:** Unzip and upload only the contents of **source.zip** to the **root directory** of your hosting server. Ex: **/var/www/html/** or **/home/username/public_html** or whatever is the root folder of your domain/subdomain. Which will make it reachable as follows: **http://yourwebsitename/**
- **Step 2a:** **Warning!** Some hosting platform hide files starting with **'.' (hidden files)**, so make sure that in your settings of your cPanel you enable this option.
- **Step 2b:** **Warning!** If your local Windows/Linux/MaxOS environment does not show hidden files, and you are planning to upload the script via FTP, make sure you enable viewing hidden files locally first even before unzipping, otherwise you will not upload all hidden files such as .env/.htaccess to your FTS server simply because your local environment will not see them.
- **Step 3:** Everything inside of the **public** folder is accessible by anyone, so make sure you don't put anything sensitive there.
- **Step 4:** After uploading all files and making sure that domain name has proper path set, create Mysql database and proper user in case if you don't already have one, to access this database. You can either create manually via your phpMyAdmin panel or use phpMyAdmin Wizard in your cPanel to create one.
- **Step 5:** Now you are ready to run the install script. Type in your browser **https://yourwebsitename/install** and press Enter.
- **Step 6:** This will start the installation process, click **Start Intallation** button.
- **Step 7:** Make sure required php packages are installed and enabled on your server. Click **Next** to continue.
- **Step 8:** Make sure listed folder are publicly accessbile. Click **Next** to continue.
- **Step 9:** For database configuration, enter your newly created database credentials. Server address: **localhost** and Port: **3306**
- **Step 10:** If all required php libraries are turned on in your hosting, installation will finish quickly and you are now fully ready to start. The default login name is: **admin@example.com** and default password is: **admin12345**. You can change it after loggin in in your profile settings.
- **Step 11:** Next step is to include your **OpenAI API key** in Davinci Configuration page under Admin Panel.
- **Step 12:** Final step is to configure CRON job correctly, refer to **CRON JOB (last page)**

Paypal Setup

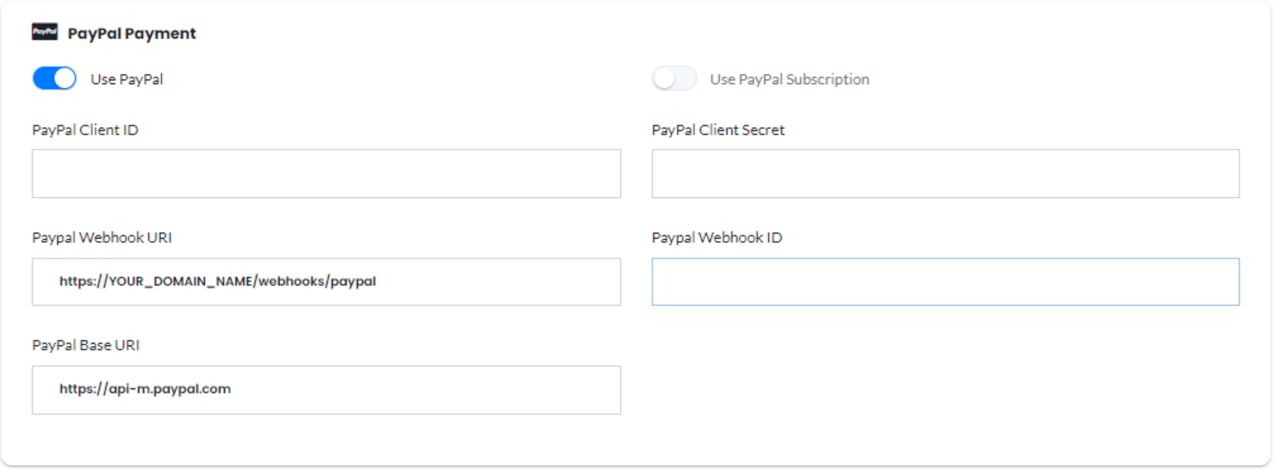
For accepting payments both for Monthly **Subscriptions** and for **Prepaid** options you can use Paypal.

You will need to include your stripe credentials under Admin -> Finance Management -> Payment Settings -> Paypal. Include all your Paypal Credentials, and don't forget to active the Paypal checkbox(es):

- PayPal Client ID

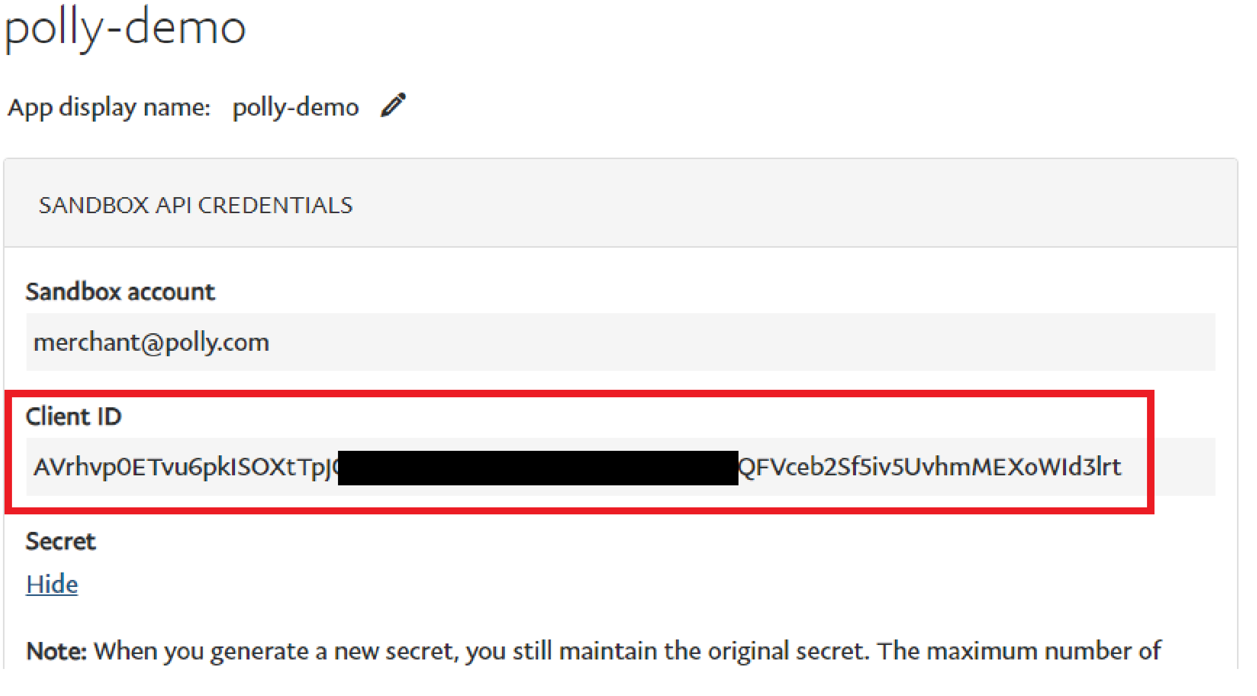
- PayPal Client Secret
- Domain Name for Paypal Webhook URI
- Paypal Webhook ID

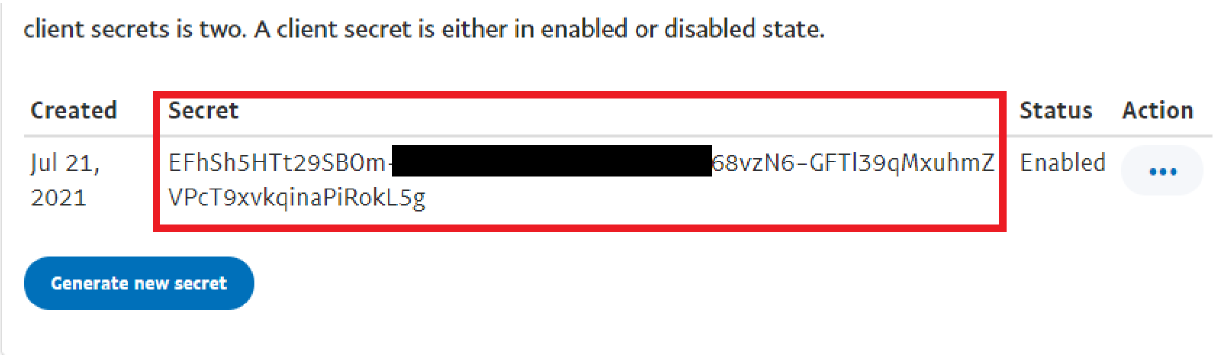
Include required data and click Save button:



How to get Paypal Client ID and Paypal Client Secret:

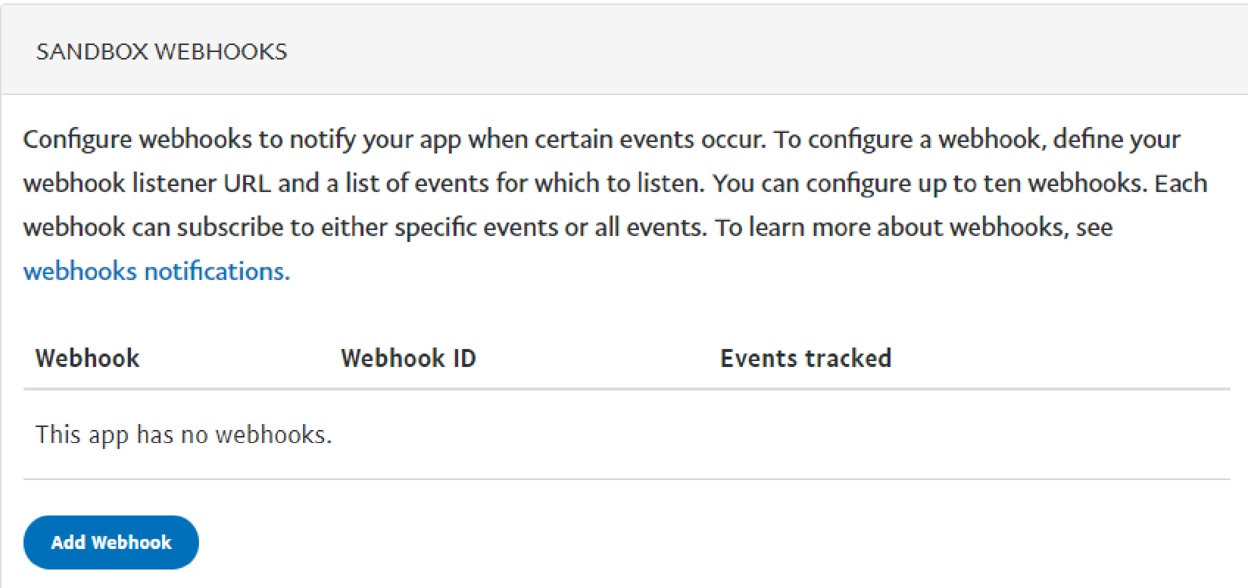
1. Login into your dashboard at developer.paypal.com
2. On the left menu bar, select Dashboard -> My Apps & Credentials -> Create New App
3. Provide desired app name and click create
4. Paypal will provide you Client ID and Client Secret for your newly created app as shown in the demo image below, in your Polly script store them under Management -> Payment Settings -> Paypal Client ID and Client Secret fields.



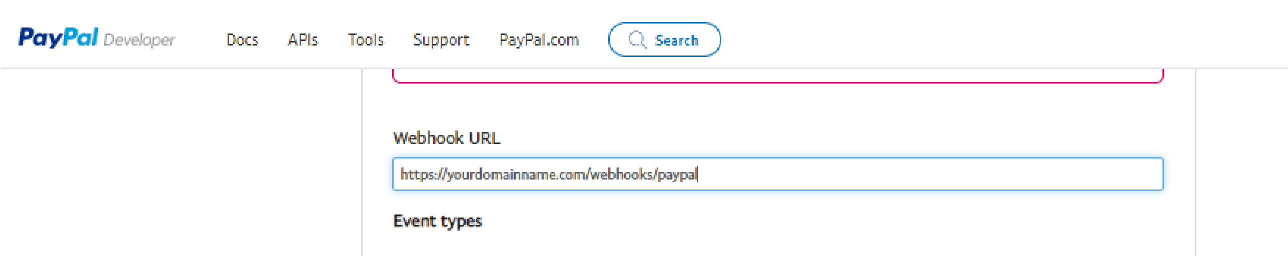


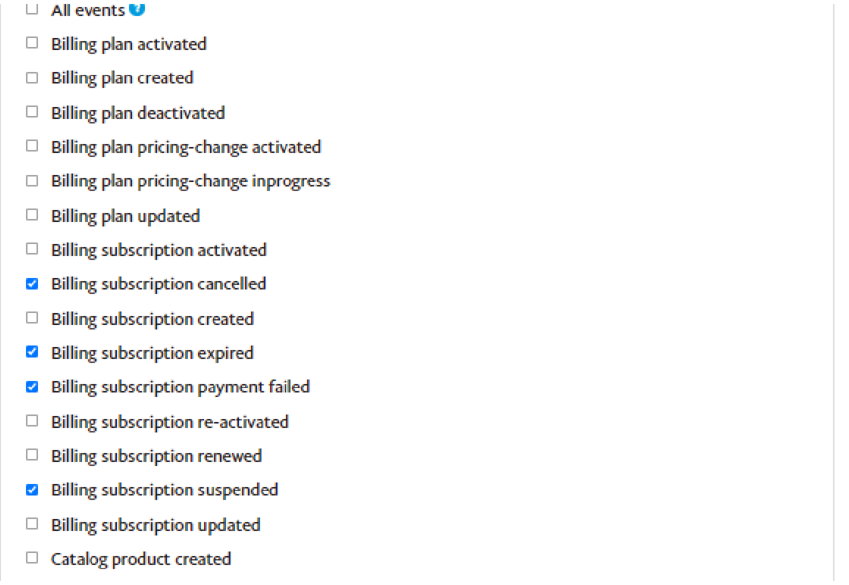
Create Paypal Webhook with select events:

- On the same page as expained above, under Dashboard -> My Apps & Credentials, select your newly created app, and scroll to the bottom of the page and you reach "Sandbox Webhooks", click "Add Webhook"

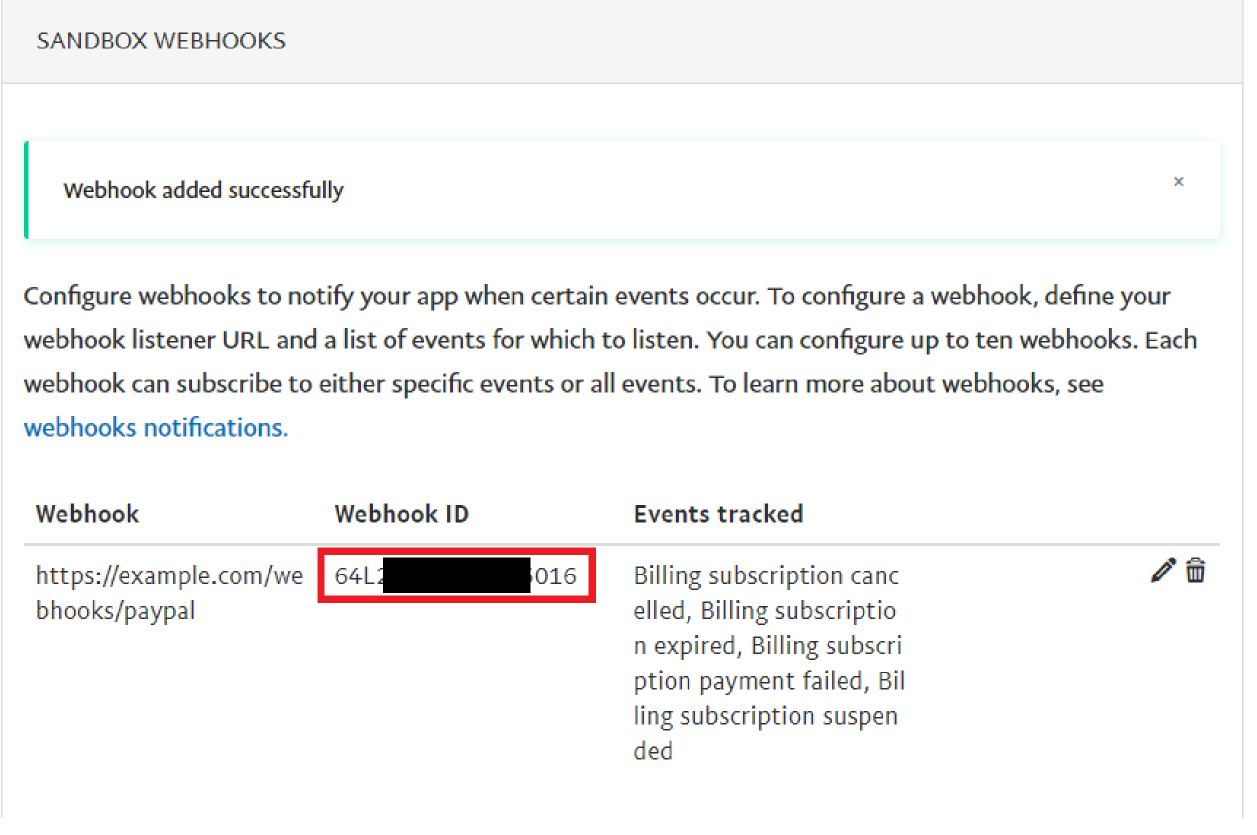


- Include your domain name followed by **/webhooks/paypal**
- Ex: https://www.domainname.com/webhooks/paypal
- Important! Make sure you have the same path included for Paypal Webhook URI under Finance Management -> Payment Settings -> Paypal
- Select following webhook events for subscriptions and click save:
  - BILLING.SUBSCRIPTION.CANCELLED
  - BILLING.SUBSCRIPTION.SUSPENDED
  - BILLING.SUBSCRIPTION.PAYMENT.FAILED
  - BILLING.SUBSCRIPTION.EXPIRED





- Save the Webhook ID under Finance Management -> Payment Settings -> Paypal -> Paypal Webhook ID



Stripe Setup

For accepting payments both for Monthly **Subscriptions** and for **Prepaid** options you can also use Stripe.

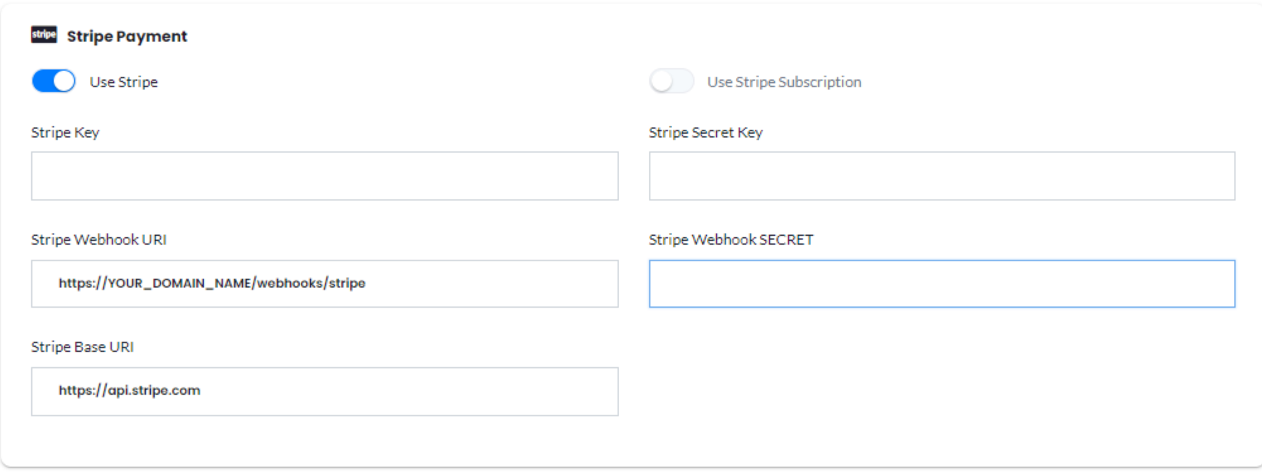
You will need to include your stripe credentials under Admin -> Finance Management -> Payment Settings -> Stripe. Include all your Stripe Credentials, and don't forget to active the Stripe checkbox(es):

- Stripe Key

- Stripe Secret Key
- Domain Name for Stripe Webhook URI
- Stripe Webhook Secret

Include required data and click Save button:

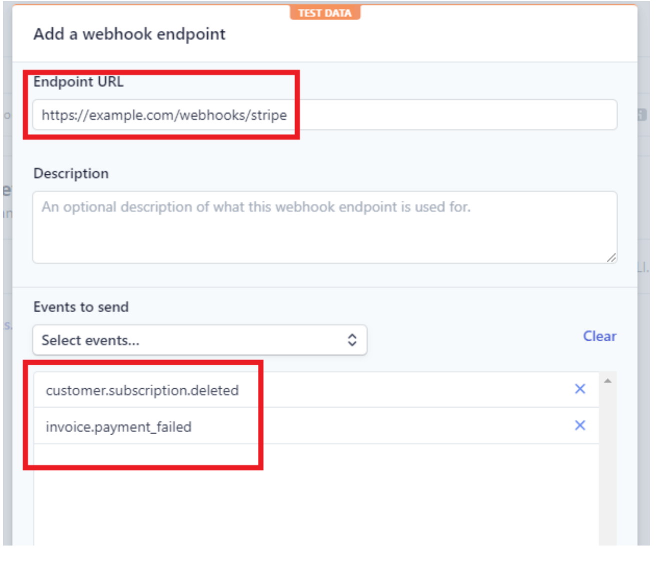
How to get Stripe API Key and Stripe Secret Key:

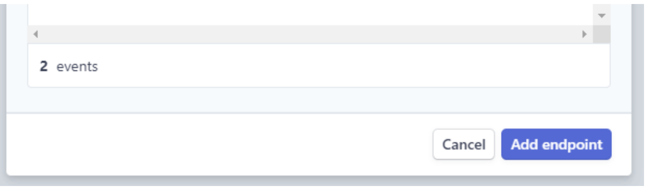


1. Login into your dashboard at dashboard.stripe.com
2. On the left menu bar, select Developers -> API Keys -> You will see your Publishable Key (API Key) and Secret Key
3. Copy and paste them to Cloud Polly, under Finance Management -> Payment Settings -> Stripe

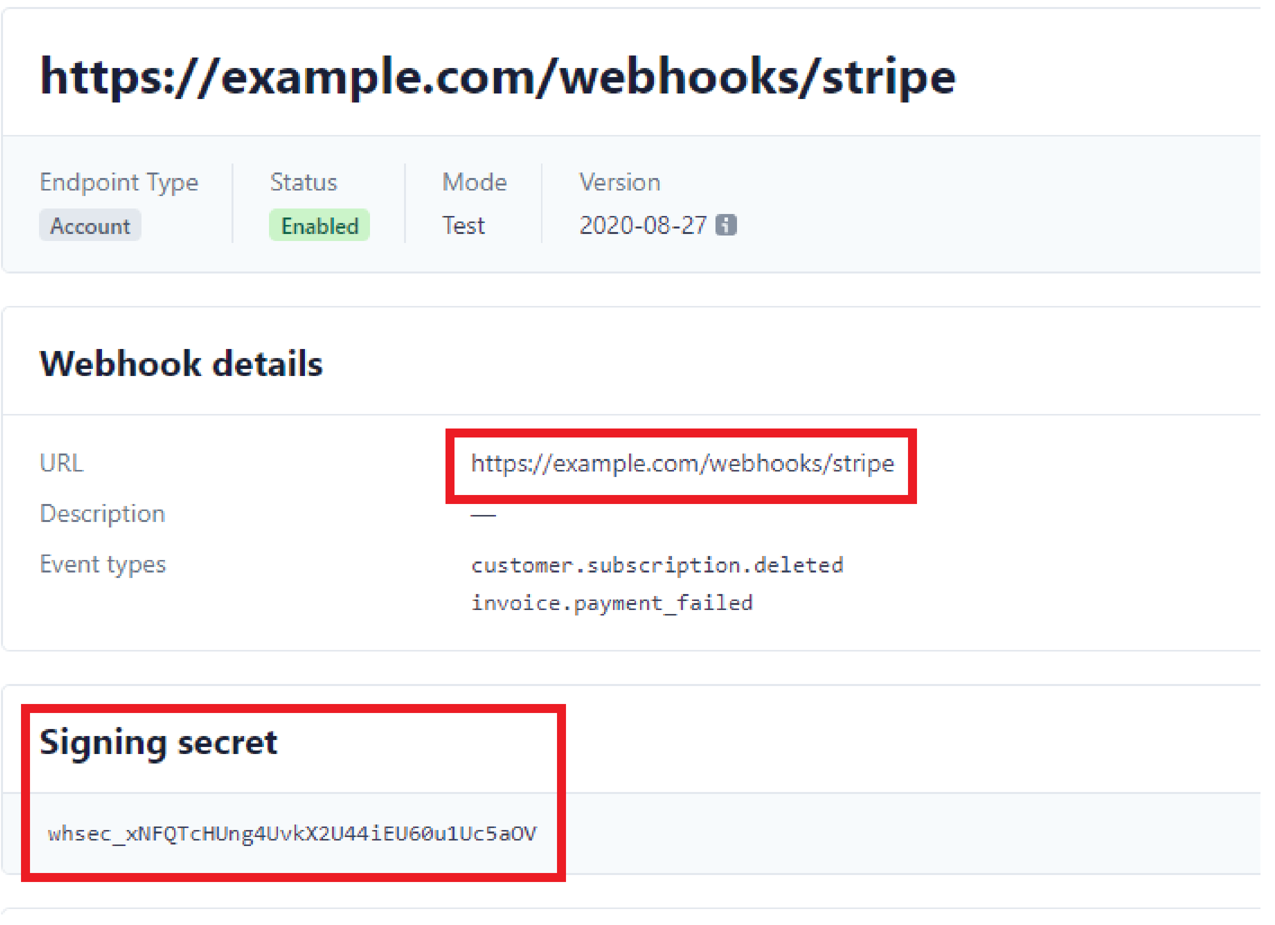
Create Stripe Webhook with select events:

1. Login into your dashboard at dashboard.stripe.com
2. On the left menu bar, select Developers -> Webhooks -> Add Endpoint
3. Create a Webhook in your Stripe Developer Account
4. Include your domain name followed by **/webhooks/stripe**
5. Ex: https://www.domainname.com/webhooks/stripe
6. And add following two events:
   1. invoice.payment_failed
   2. customer.subscription.deleted





7- After your webhook is created, copy Webhook URL and Webhook Secret to Admin -> Finance Management -> Payment Settings -> Stripe to the respective fields.



Subscription Plans Creating

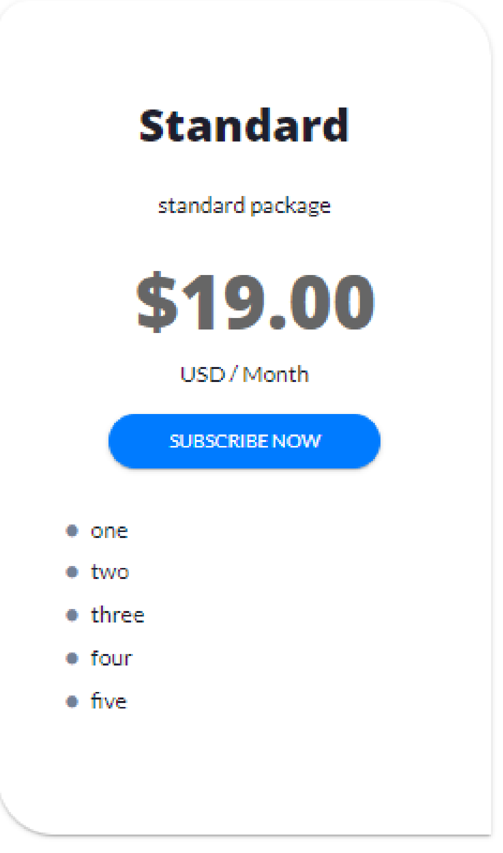
You can create as many subscription plans as you want, both for Monthly and Yearly subscription plans.

All subscription plans must have respective payment gateway plan ids, and these plan ids can be generated **only** via your payment gateway's account interface.

How to create Subscription Plan:

- **Step 1:** Go to your Paypal/Stripe/Razorpay/Paystack accounts and Create New Subscriptions/Plans.
- **Step 1a:** Creating **Paypal** Plan IDs:
- **Step 1b:** Creating **Stripe** Product IDs:
- **Step 1c:** Creating **Paystack** Plan Codes:
- **Step 1d:** Creating **Razorpay** Plan IDs:

- **Step 2:** Copy those Subscription/Plan IDs.
- **Step 3:** Go to "Admin Panel" -> "Finance Management" -> "Subscription Plans" -> click on "Create New Subscription" button.
- **Step 4:** Fill up the all required fields and inlcude Paypal/Stripe/Paystack/Razorpay - Subscription/Plan IDs.
- **Step 5:** Note: Plan Name must be a unique string value
- **Step 6:** Plan descriptions must be a comma separated string values
- **Step 7:** After filling up the fields, click "Create" button. Example: your newly created subscription will looks like this:



SMTP Configuration

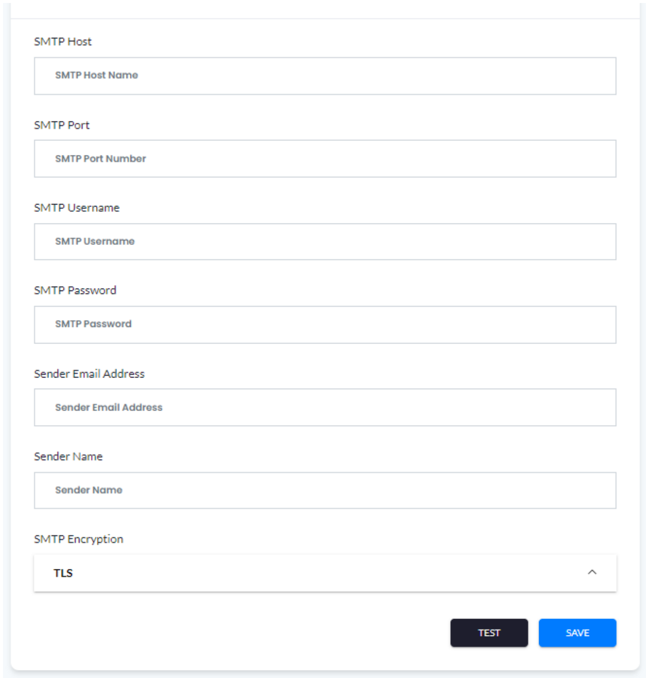
For sending all emails, such as email verification, support ticket creation etc, SMTP needs to be configured.

All the **SMTP settings** can be configured and set in **Admin Panel** under **General Settings -> SMTP Settings**.

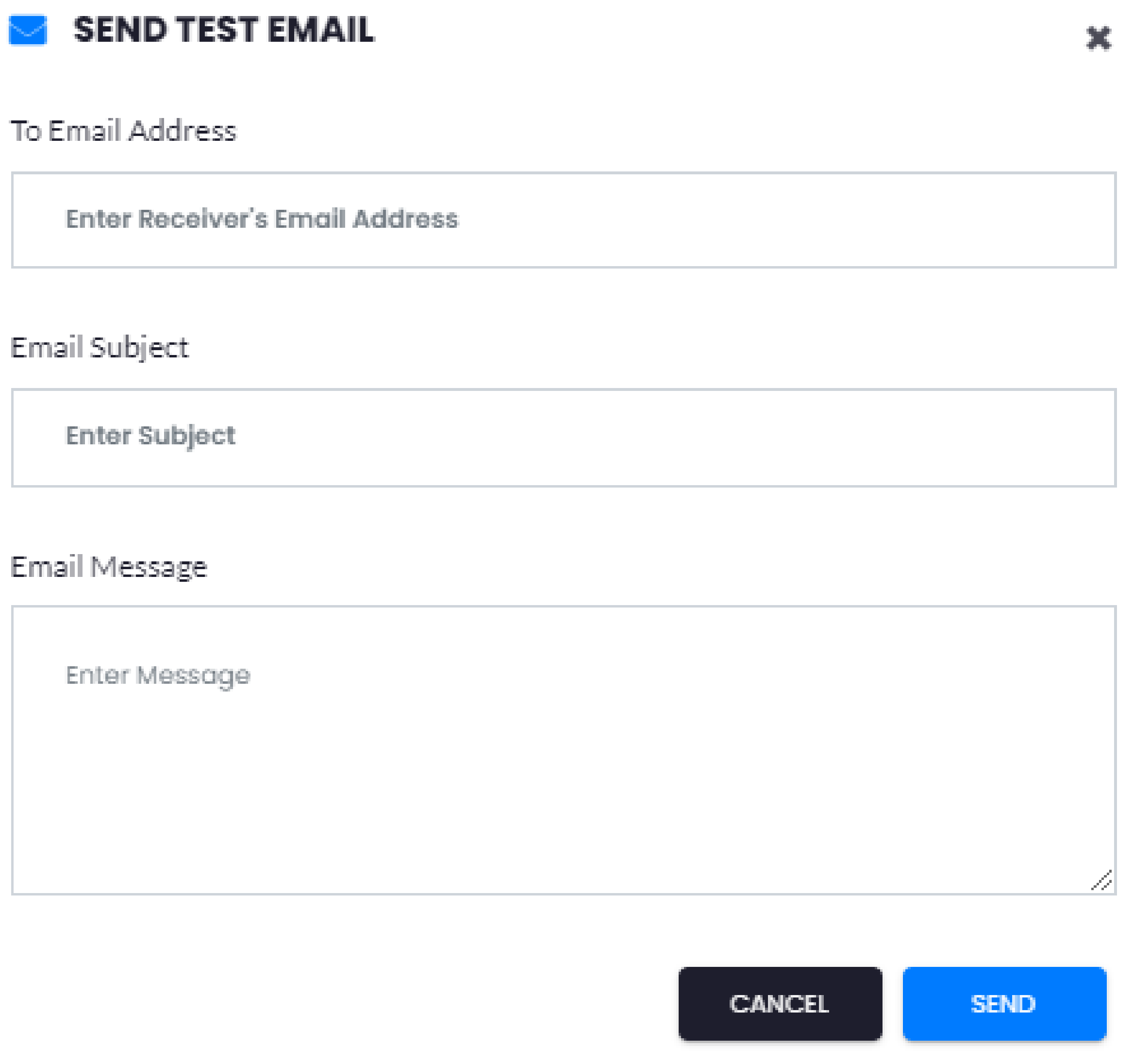
Required SMTP Settings:

- SMTP Host
- SMTP Port
- SMTP Encryption
- SMTP EMail Login (Username)
- SMTP EMail Password
- SMTP Sender Email Address
- SMTP Sender Name

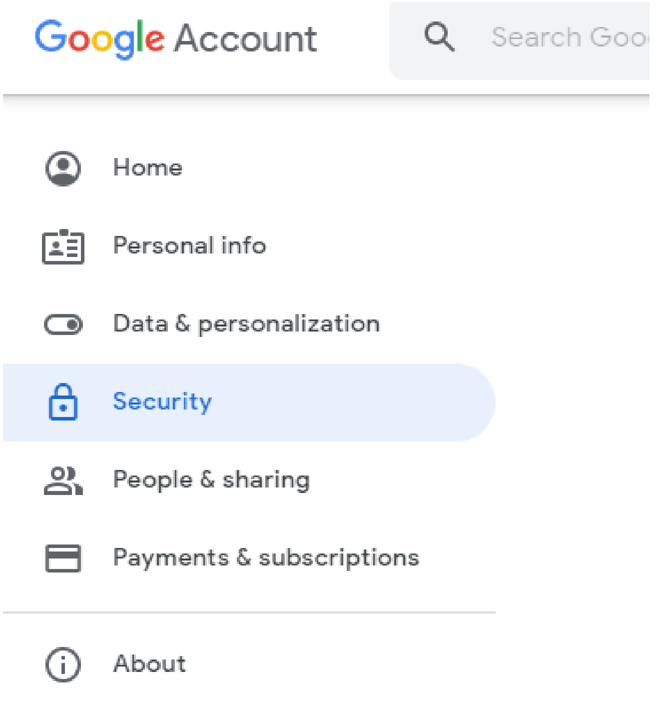
Include Required SMTP Settings and click Save Changes button:



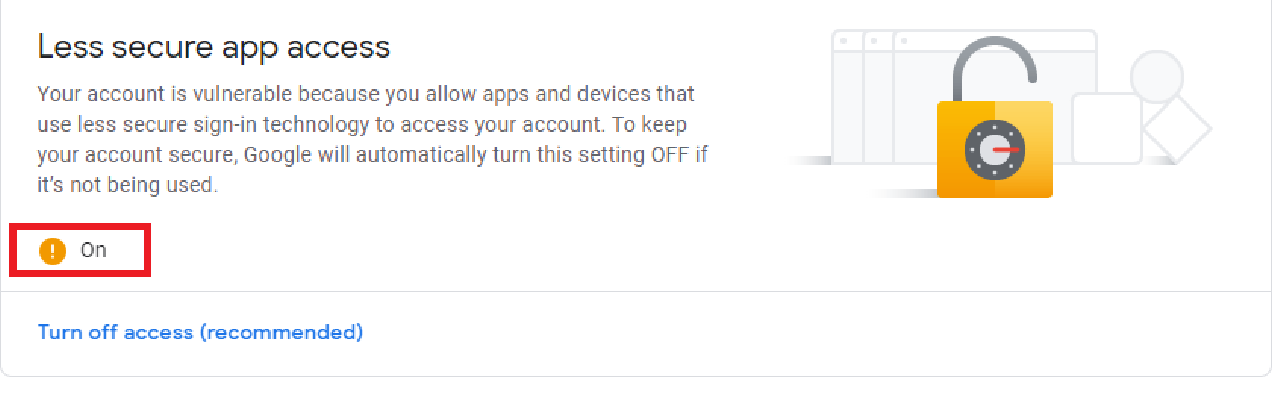
You can also test your SMTP Settings immediately via Test button:



If you are planning to use Gmail for sending emails:



In your **Gmail Account** settings, under **Security** tab, make sure your **Less secure app access** feature is **turned on**.



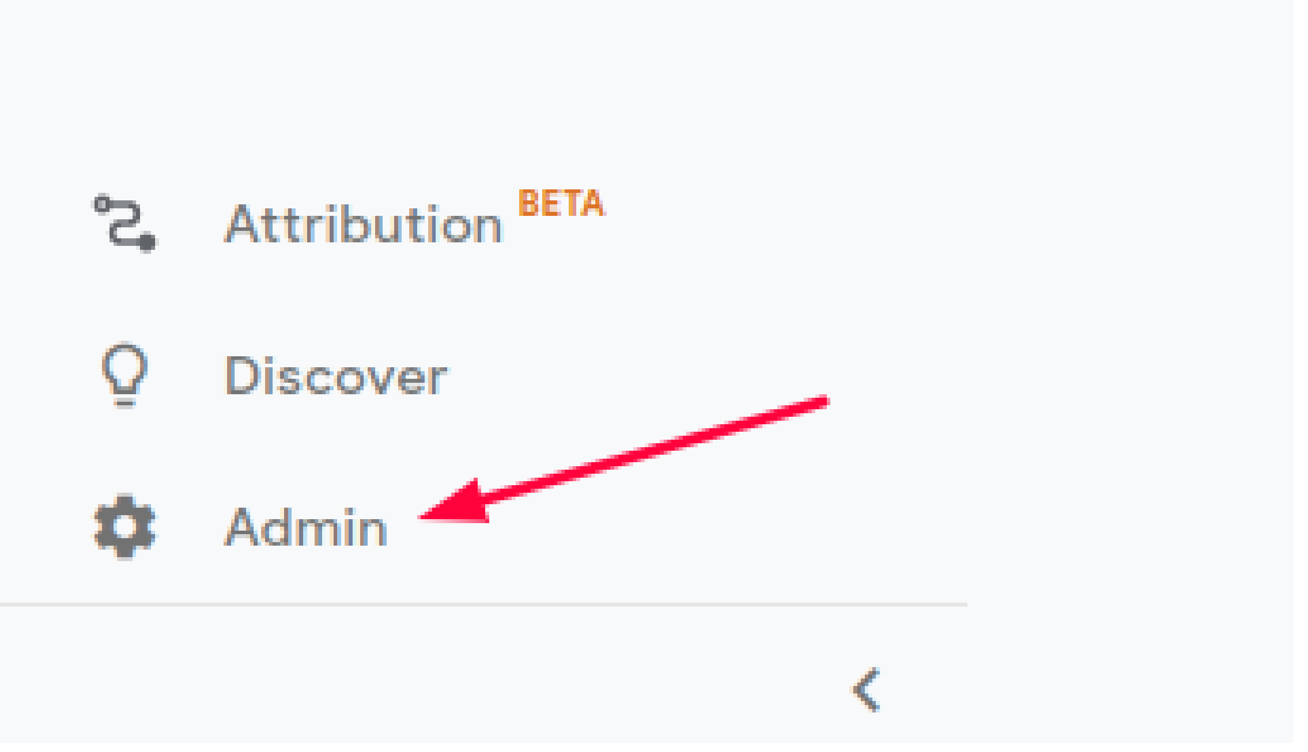
Google Analytics Configuration

Analyze in-depth detail about the visitors on your main frontend page.

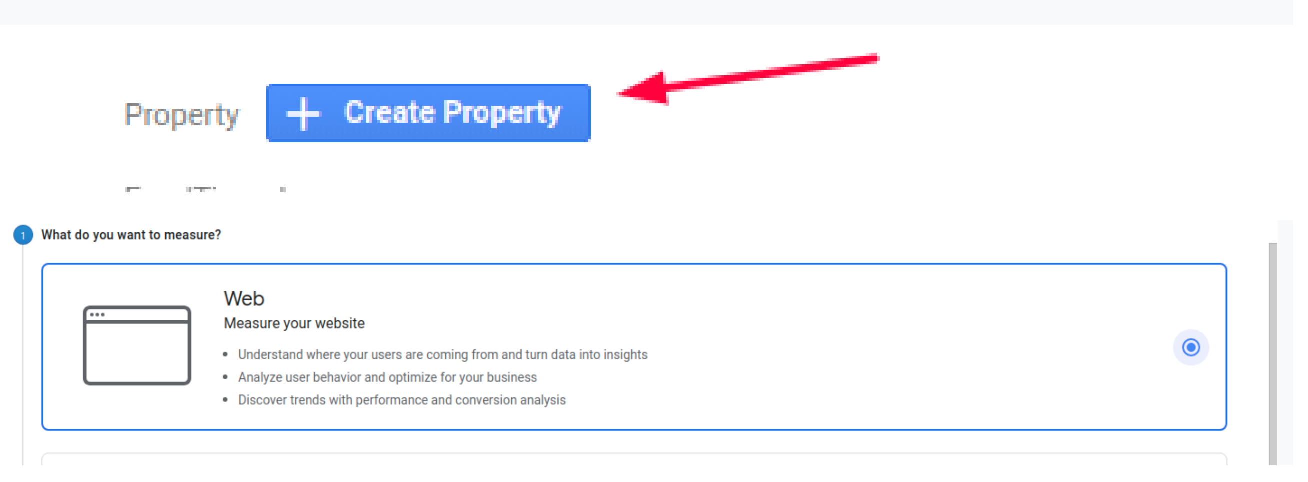
Setup Google Analytics:

- **Step 1:** Visit Google Analytics page.

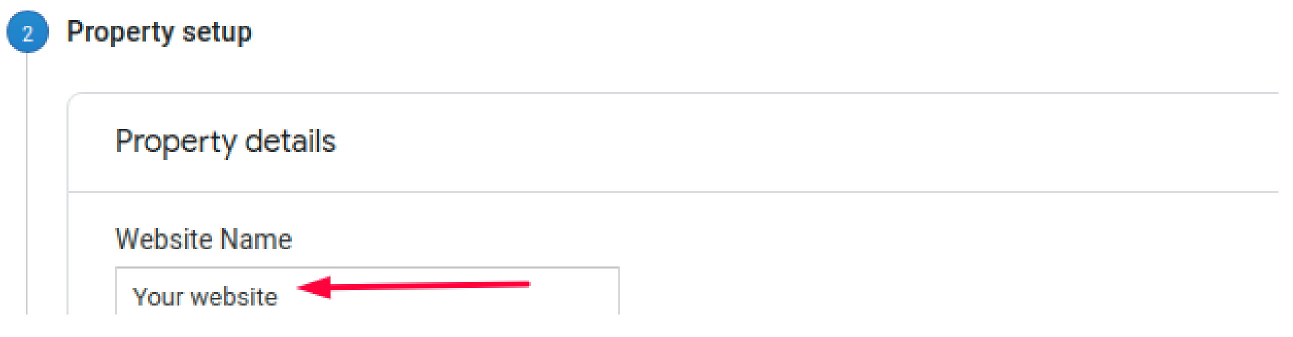
- **Step 2:** On the left menu select **Admin**.

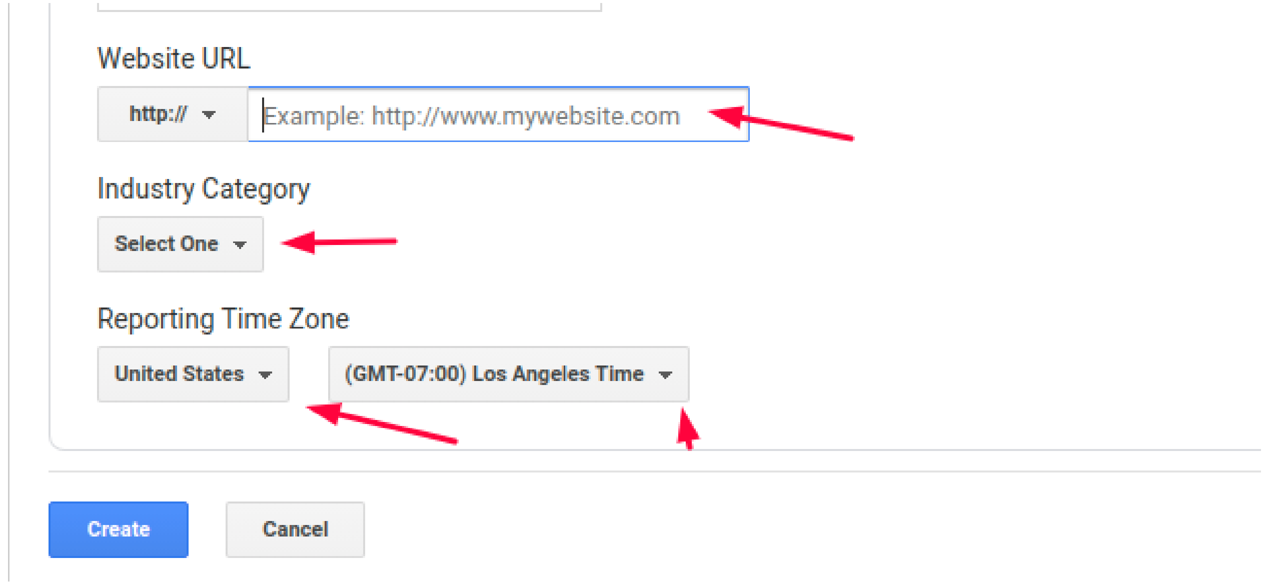


- **Step 3:** Click **Create Property**, **select **Web** and then **Continue**.**.

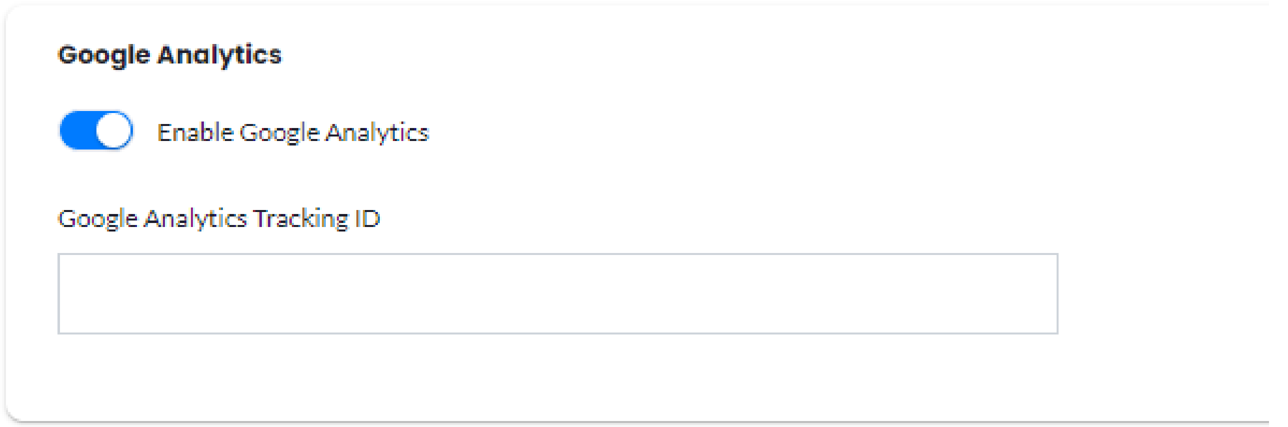


- **Step 4:** After that it will appear screen with information for your website.
- **Step 5:** Fill it and then click **Create**.





- **Step 6:** Include **Tracking ID** into **Google Analytics Tracking ID** field in **Admin Panel** under **General Settings -> Global Settings -> Google Analytics**, and don't forget to enable the Google Analytics checkbox and click **Save** button.

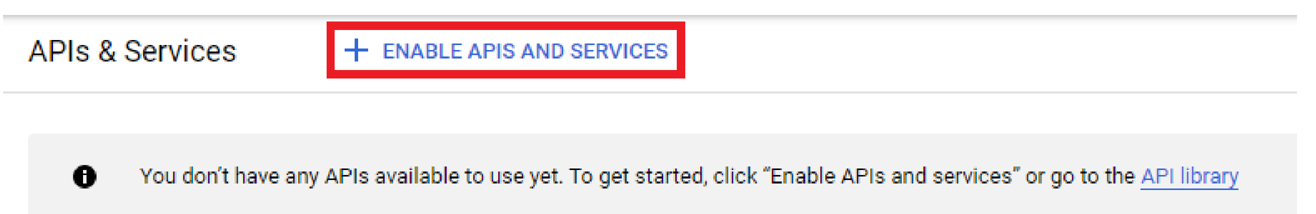


Google Maps Configuration

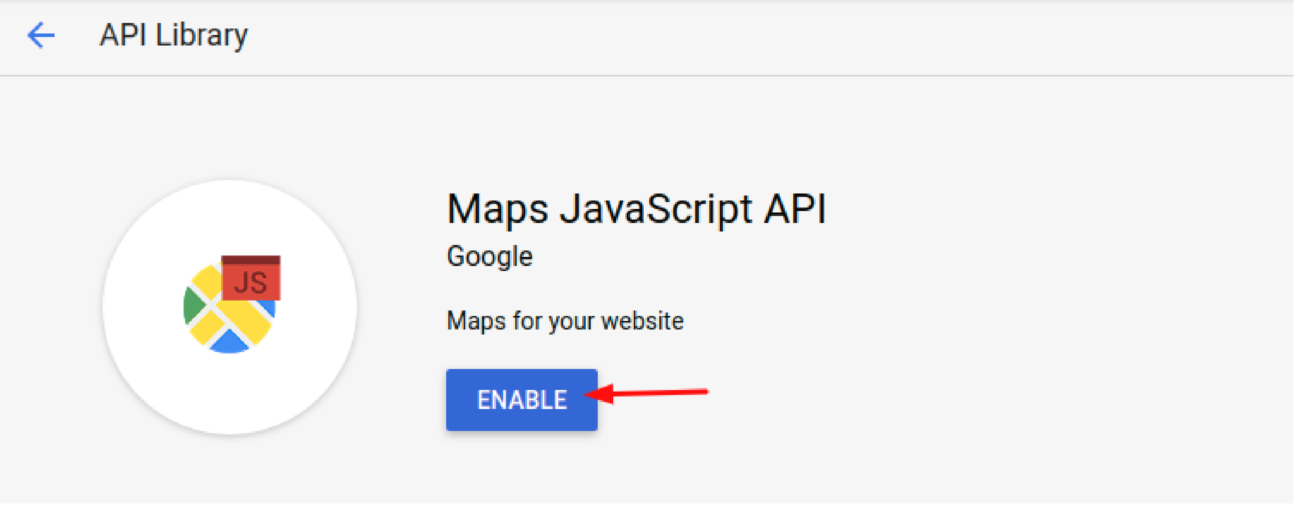
To use the Maps JavaScript you must have an API key. The API key is a unique identifier that is used to authenticate requests associated with your project for usage and billing purposes.

To Get a Maps API Key:

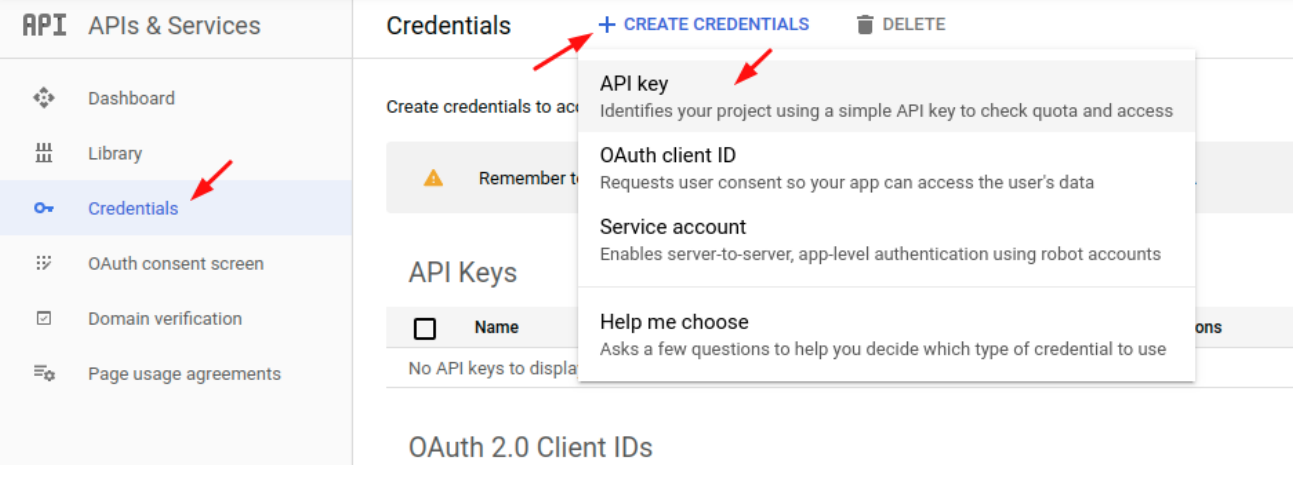
- **Step 1:** Go to the Google API Console
- **Step 2:** Click the project drop-down and select or **create project** for which you want to add an API key and give it a **project name.**
- **Step 3:** If you are using Google API for the first time you will see this screen after creating the project:



- **Step 4:** Click **Enable Apis and Services** button or select **Library** button on the left.
- **Step 5:** Find **Maps JavaScript API** and enable it as shown below.

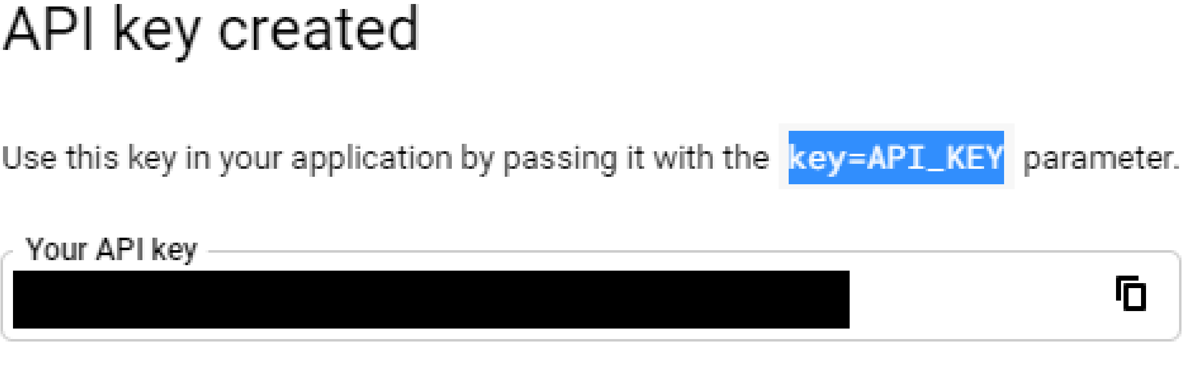


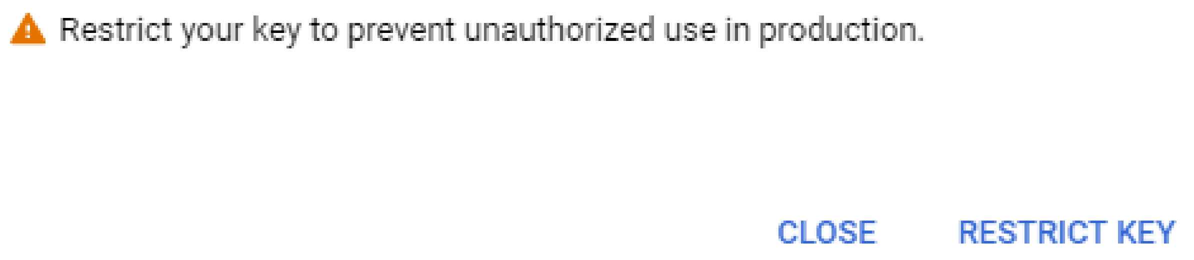
- **Step 6:** Click the menu button and select **Credentials** and click **Create credentials** > **API key**.



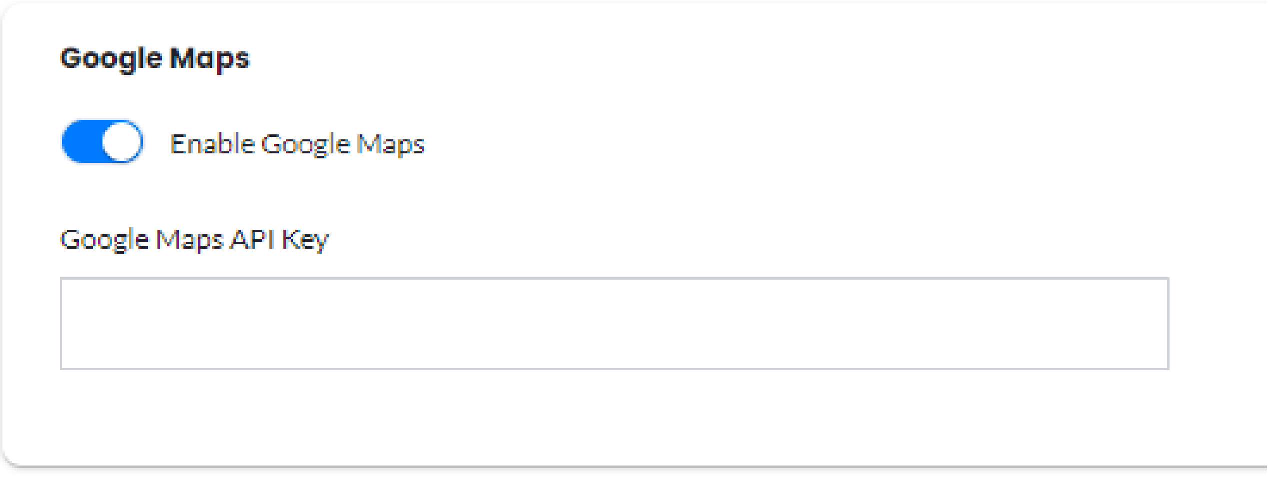
- **Step 7:** The **API key created** dialog displays your newly created API key. Copy and paste it as shown in Step 8. The new API key is listed on the **Credentials** page under **API keys**.

Important: Remember to restrict the API key before using it in production.





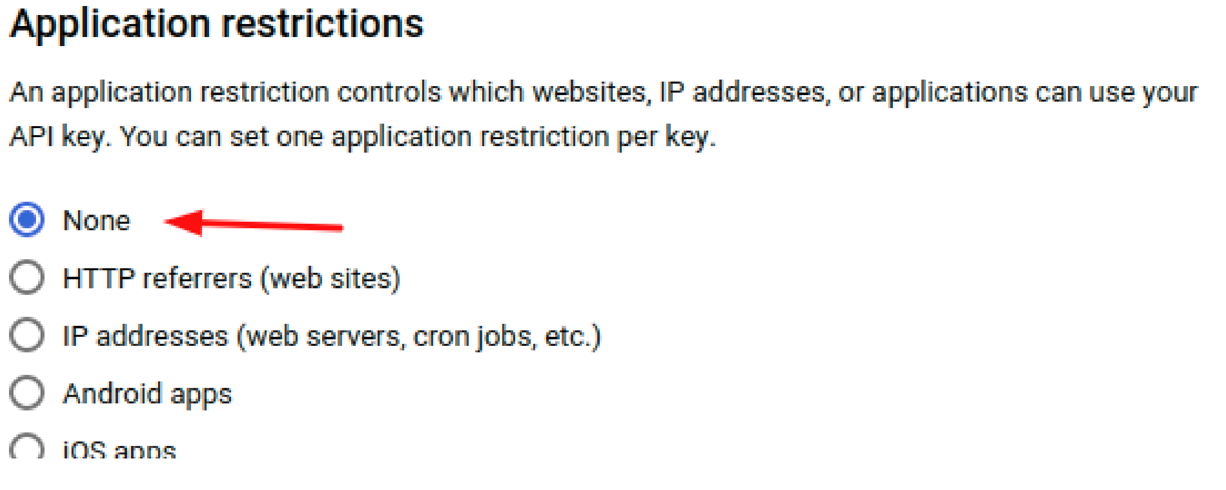
- **Step 8:** Include the API key in **Admin Panel** under **General Settings -> Global Settings -> Google Maps**, and don't forget to enable the Google Maps checkbox and click **Save** button.



Change application restrictions:

Depending on your server/domain configuration the script maybe won't accept the **HTTP referrers** restrictions so try with changing the application restrictions back to **None**

Enable Billing on your project:



Sometimes after adding the key these features maybe won't work again. Then depending on the google account associated with the project created and maybe you will need to enable Billing.

Learn more here about it: Billing or here Getting started with Google Maps Platform

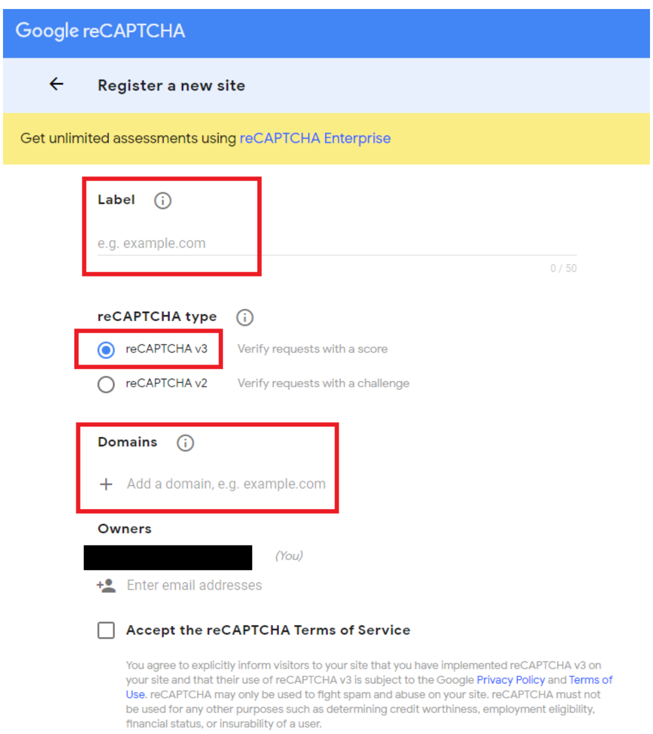
Google reCaptcha v3 Configuration

To avoid spams, all contact and registration forms have Google reCaptcha v3 enabled. It is optional feature, but strongly recommended.

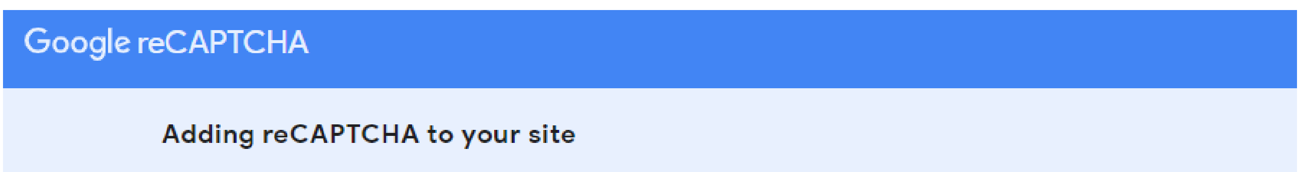
To use this feature, you will need to have **reCaptcha Site Key** and **reCaptcha Secret Key**.

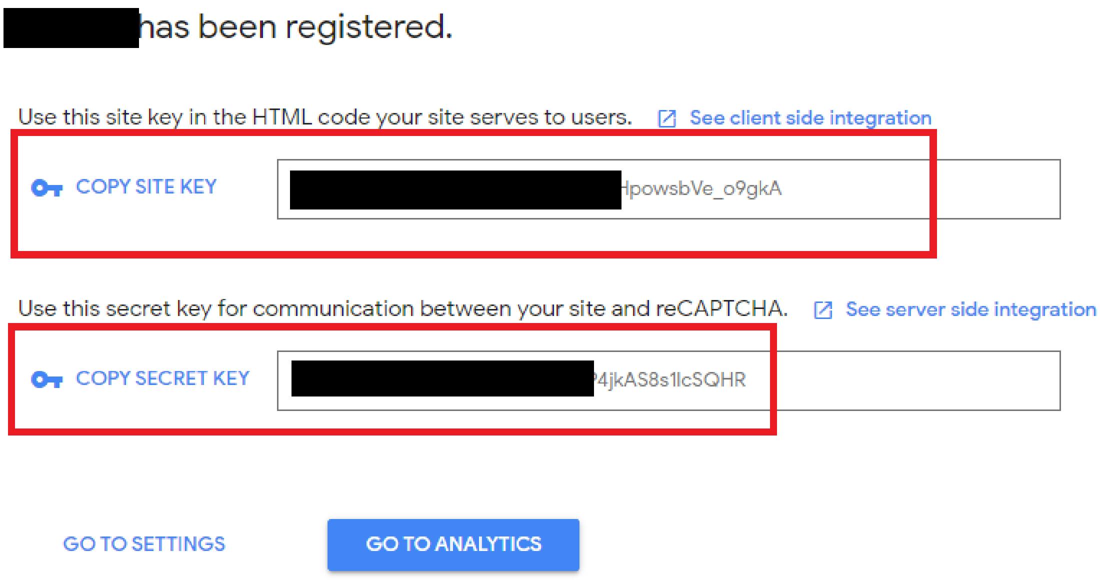
To Get a Google reCaptcha v3 Keys:

- **Step 1:** Go to the Google reCaptcha Console
- **Step 2:** Enter **Label** to identify your reCaptcha keys.
- **Step 3:** For reCaptcha Type, select "reCAPTCHA v3":



- **Step 4:** Enter your domain name.
- **Step 5:** Accept the reCAPTCHA Terms of Service
- **Step 6:** Click the "Submit" button.
- **Step 7:** Copy both "Site Key" and "Secret Key".





- **Step 8:** Include the API keys in **Admin Panel** under **General Settings -> Global Settings -> Google reCaptcha**, and don't forget to enable the Google reCaptcha checkbox and click **Save** button.

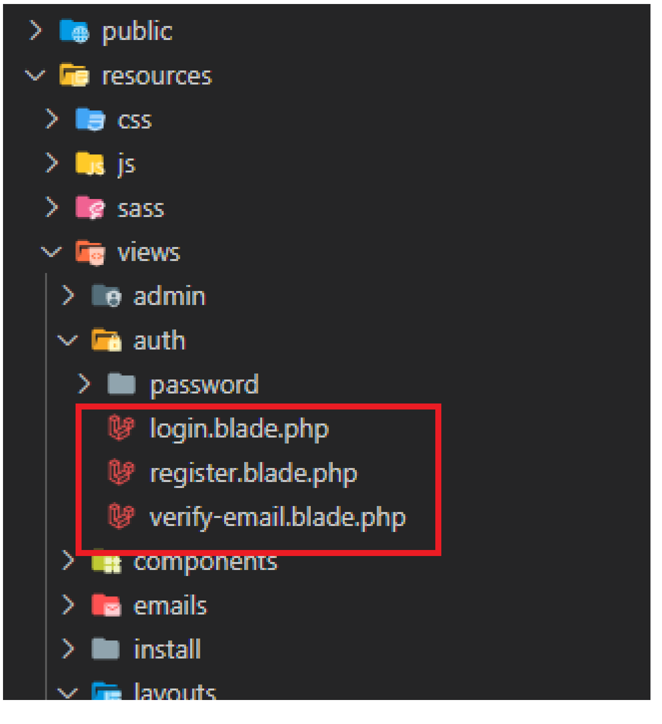
How to Customize Frontend

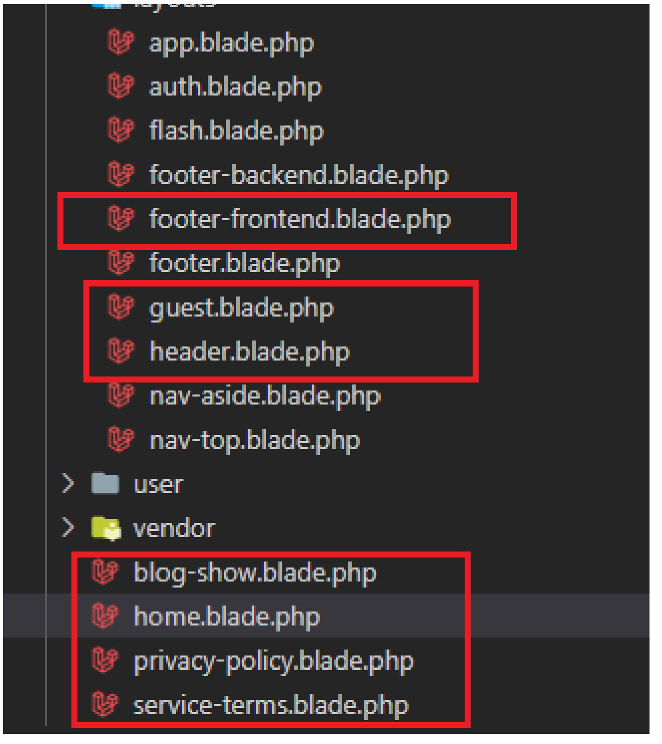
Frontend is created as a landing page and consist of single **home.blade.php** blade file that was created with standard HTML5 + CSS3 Bootstrap structure. SaSS files for frontend are separated from backend Sass files so it is easy to edit.

Warning: If you are planning to add heavy css customization, it would be better to create a separate css file and keep intact upon each update, otherwise next update might rewrite all previously made changes in case if we add any improvements to frontend.

How to customize frontend:

- **Step 1:** All files that are used as frontend pages or for frontend can be found under **resources/views** and couple files under **resources/views/layouts** directories as shown in the picture.





- **Step 2:** For frontend styling we used SaSS, they can be found under **resources/sass/frontend** directory. If you plan to use sass files for customization don't forget to compile your sass files first to apply the effects. Otherwise, you can add custom css files into **guest.blade.php** below line 23.
- **Step 3:** All pages are stored direcly under **resources/views** directory and named accordingly. You can edit the text in any of these files as you prefer.
- **Step 4:** If you want to add new page, make sure it has the same **@extends** and **@section** parts as in other pages.

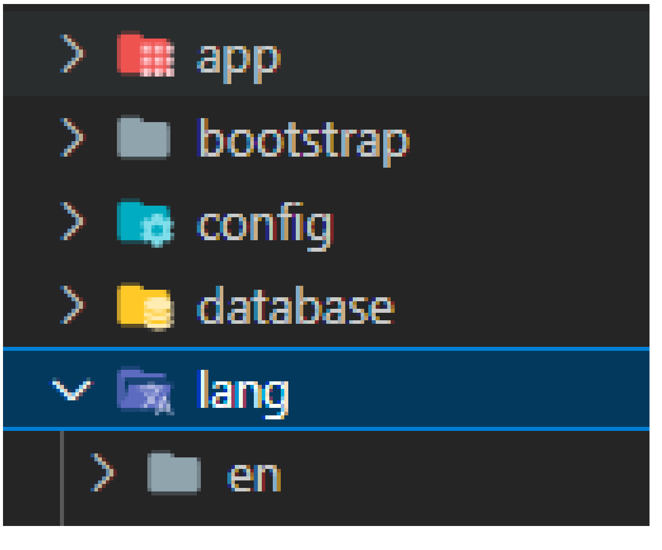
Translate to other Languages

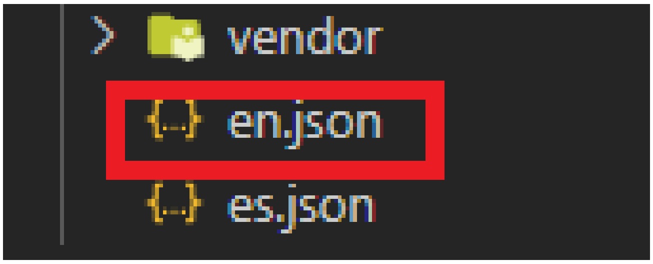
Cloud Archive by default is written in English Language, we also included as an example support for Spanish language, translated by Google Translate (Warning! Quality of these translations might not be the best).

You can translate or add any language as you prefer.

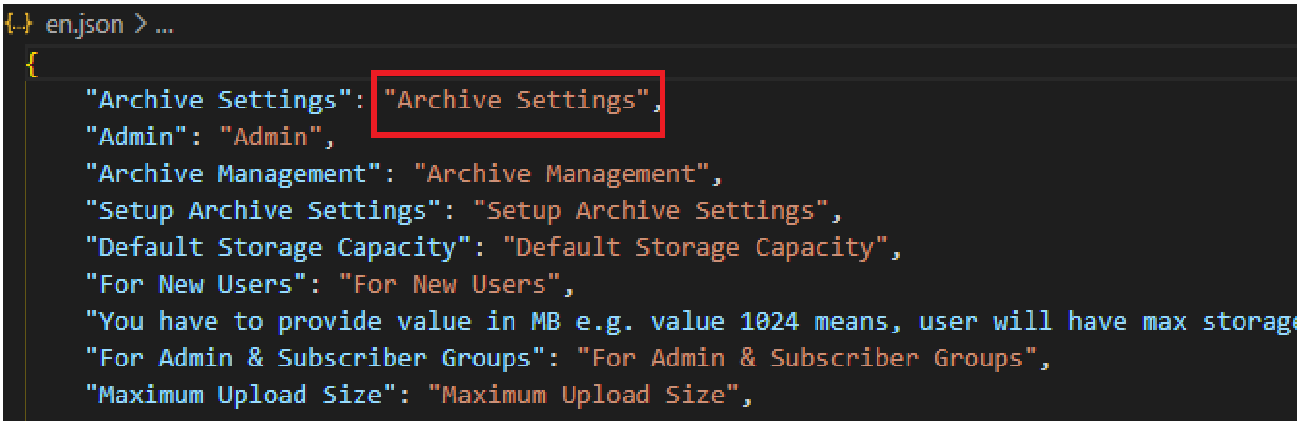
How to Translate:

- **Step 1:** Go to **"Lang"** folder in the root directory of the script.
- **Step 2:** Copy "en.json" to your desired language code. Ex: "ru.json"





- **Step 3:** Translate only the value part, and make sure they stay within **double quotes** with **comma** at the end except the last value in the json file.



- **Step 4:** Save your file in the same "Lang" folder.
- **Step 5:** Go to **"Config" -> "locale.php"** file.
- **Step 6:** Add your new language code with proper fields: 'code', 'display', 'flag' as shown below, here is an example

```
'ru' => [
    'code' => 'RU',
    'display' => 'Русский',
    'flag' => 'ru'
],
```

Set as Default Language

How to set as Default Language:

- **Step 1:** After completing step above and including your language. Go to config/app.php file
- **Step 2:** On line 89: 'locale' => 'en', - change 'en' to the country code of your language
- **Step 3:** Save the config file.

CRON Job Configuration

The software requires cron jobs to be set up in order to automate some of its functions, such as checking the status of the accounts, as well as additional storage clean-ups.

How to create CRON job:

- **Step 1:** Open **terminal** in your hosting platform or if you are using cPanel, copy and paste the **Step 3** in your **Cron Jobs** tab.

- **Step 2:** Run **crontab -e** command

```
crontab -e
```

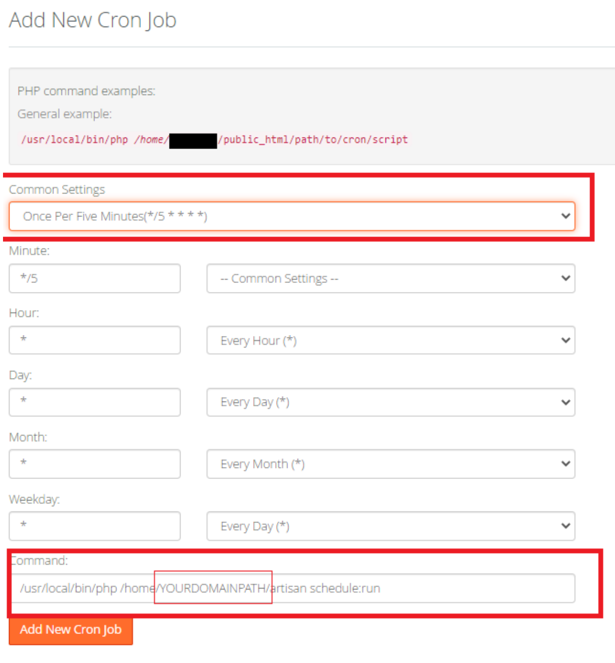
Step 3: Enter following cron job, it is scheduled to run twice a day, include **YOURWEBSITE DOMAIN PATH**, you can customize frequency as you prefer:>

```
0 0,12 * * * /usr/local/bin/php /home/YOURDOMAINPATH/artisan schedule:run
```

Step 4: (If using terminal): You can check all local cron jobs that are running via:

```
crontab -l
```

Step 5: If you are setting up via cpanel here how it looks like. Select the minimum time that is available in your hosting, ideally **1 Minute** interval if your hosting allows. Change **YOURDOMAINPATH** accoringly, make sure it includes your username stated in your hosting directory.



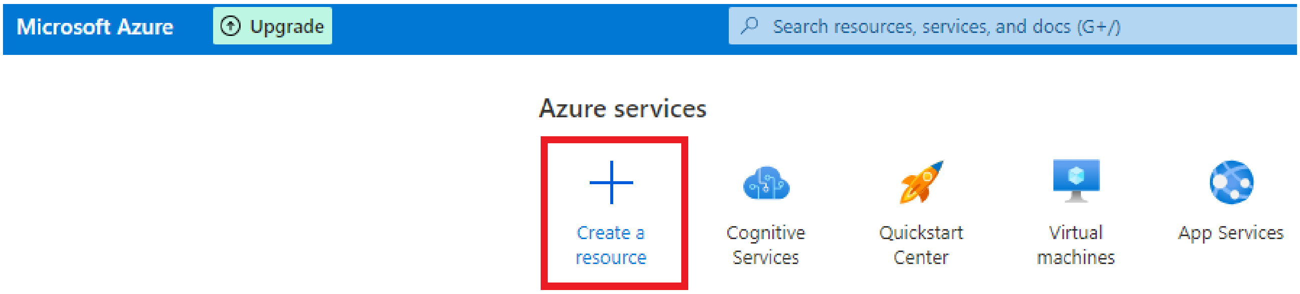
Azure Account Creation

Getting started with Microsoft Azure.

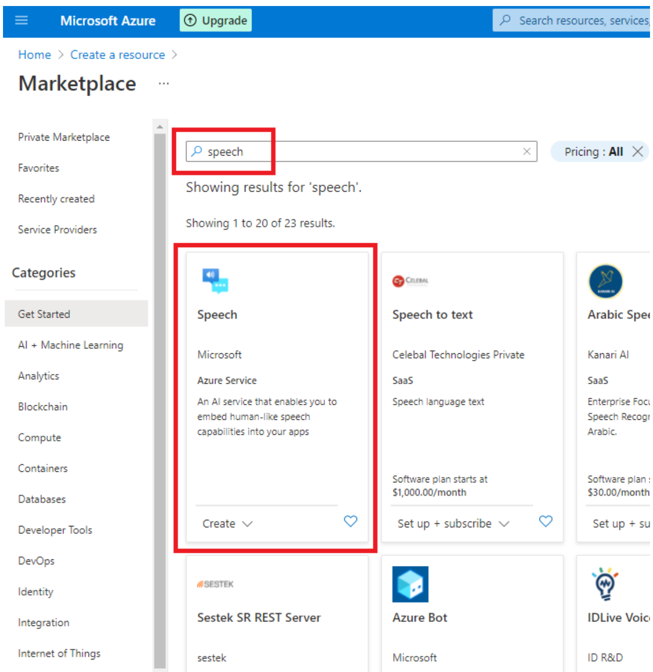
How to Create Microsoft Azure Account:

- **Step 1:** Go to the Azure home page.
- **Step 2:** Create new account and afterwards sigin into your **Azure Portal** under your Profile on the top right corner.

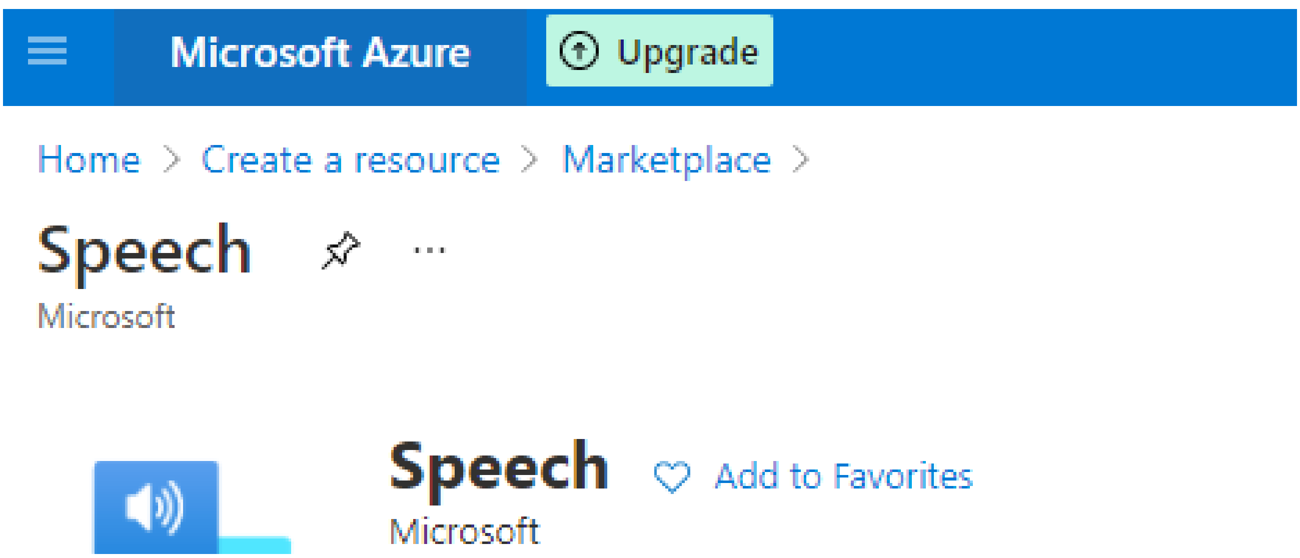
- **Step 3:** Click on **Create a resource**.

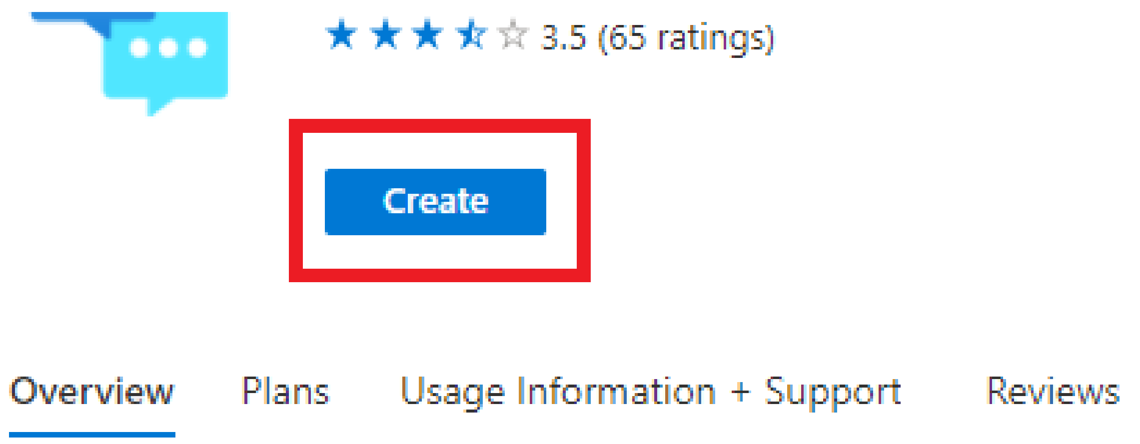


- **Step 4:** Search for keyword: **speech** and select the first Speech service from Azure.

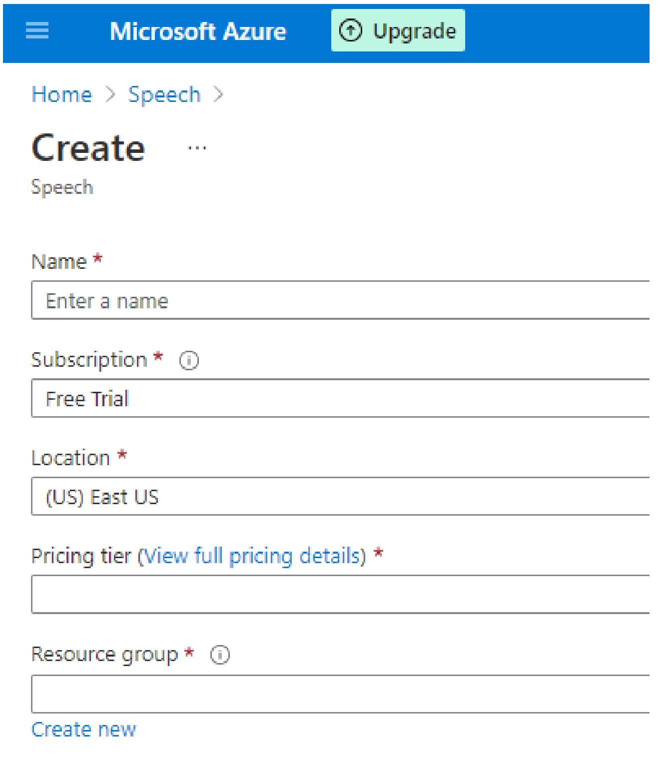


- **Step 5:** Afterwards, click **Create** button.

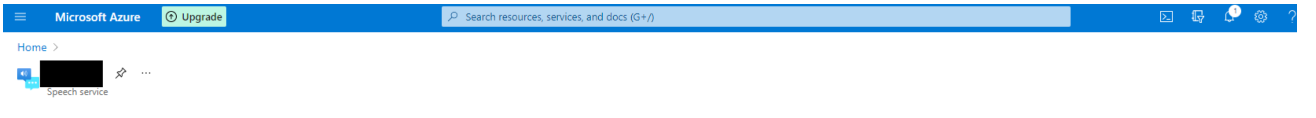


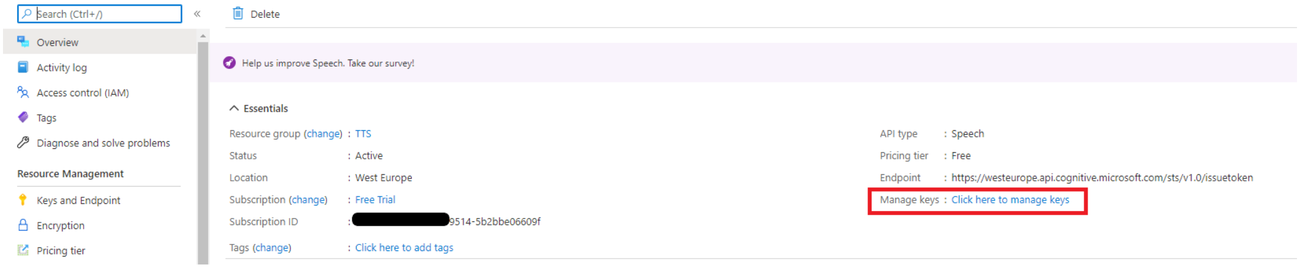


- **Step 6:** Fill in the required information.

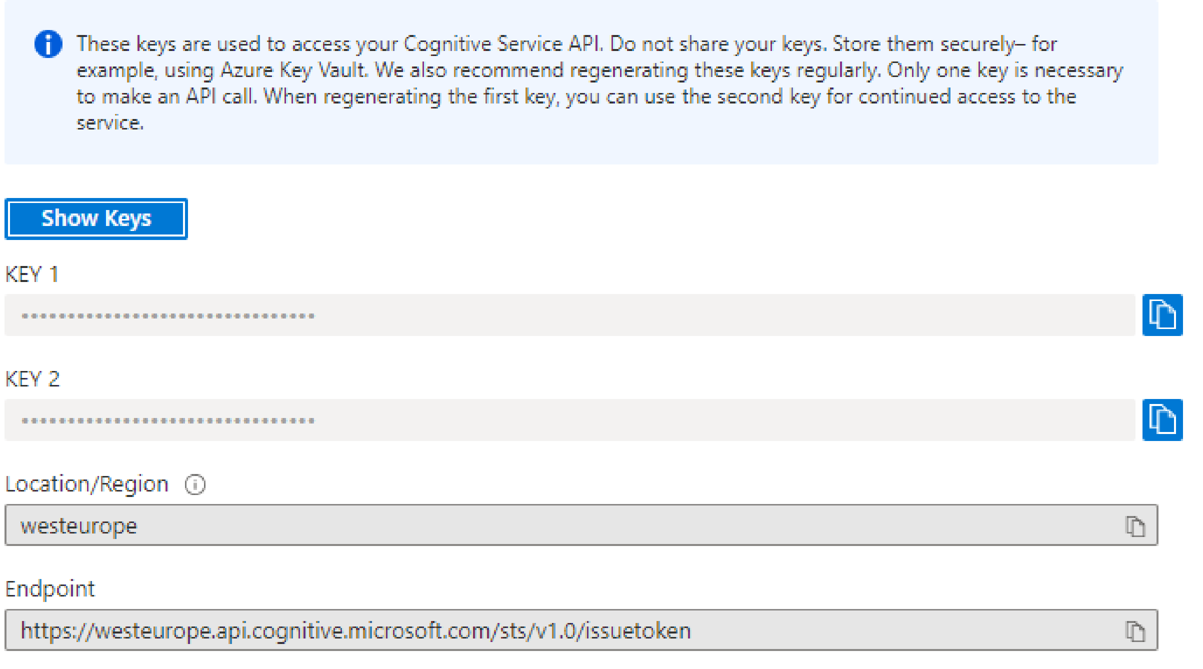


  - **Name**: Any string to identify this resource
  - **Subscription**: Free Trial (default)
  - **Location**: select Azure region as you prefer. **Warning!** Not every language/voice is available in all Azure regions, we strongly recommend to consider either **US East** or **West Europe** regions.
  - **Pricing Tier**: Free F0 is good for starting only, it has limitation in concurrency, refer to Azure Documentations to get exact numbers. Standard S0 will start billing.
  - **Resrouce Group**: Any string to group your resources.
- **Step 7:** Almost there, on this the next page click on **Click here to manage keys**.

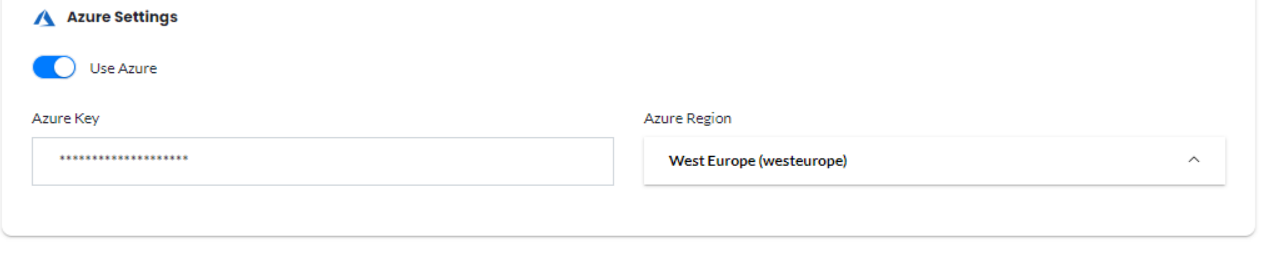




- **Step 8:** Copy one of the keys.



- **Step 9:** These keys are used to access your Azure TTS Service API. Do not share your keys. Store them securely– for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.
- **Step 10:** Go to **Admin Panel**, Davinci Management -> Davinci Settings, and enter API Key for Azure, make sure you select proper **Azure Region** as shown in the previous step with your key.



GCP Account Creation

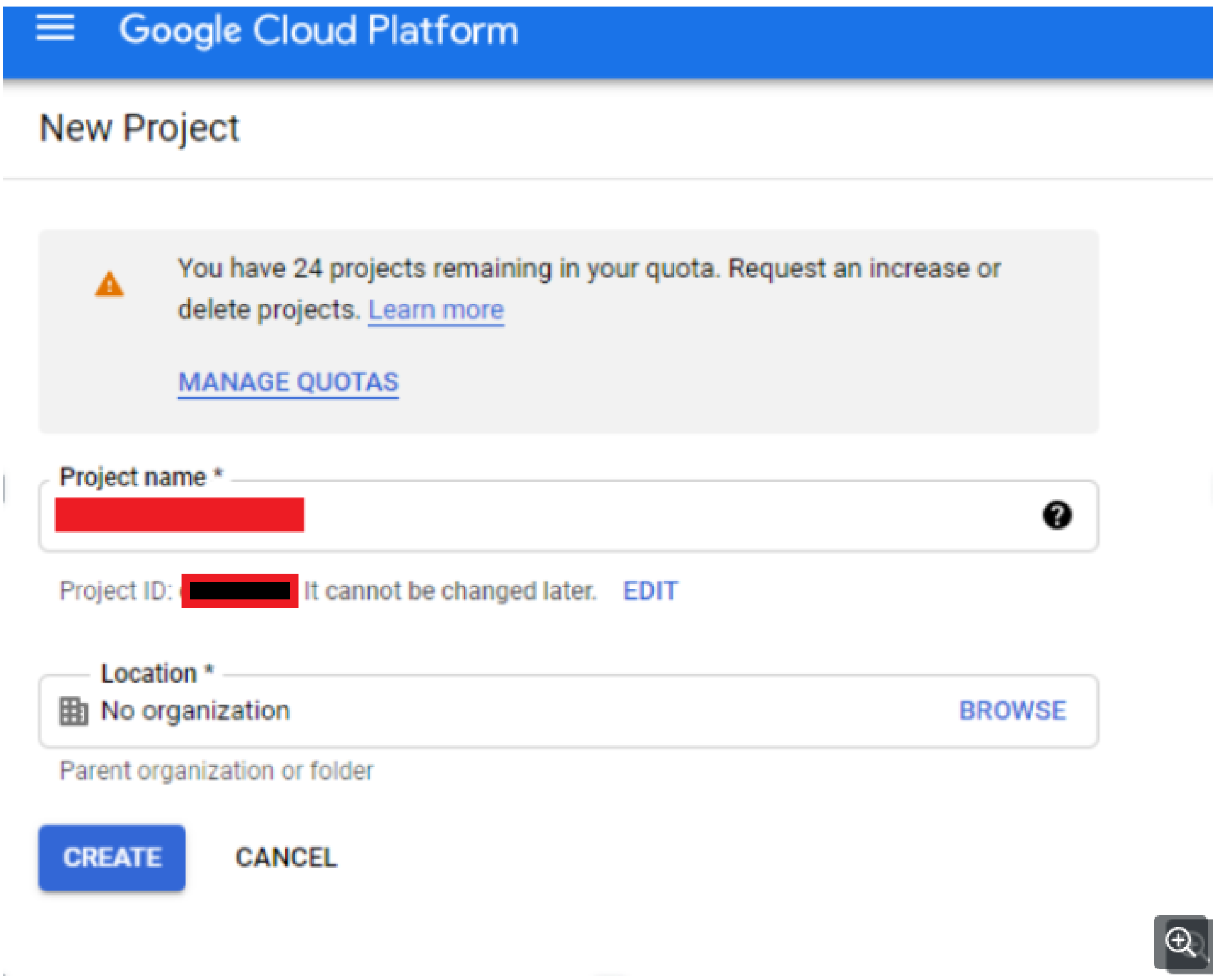
Getting started with Google Cloud.

Note: When you open a Google Cloud account, they provide you 300$ of credit that is valid for 2 months.

ONLY GCP Text to Speech Setup

How to Configure GCP for Text to Speech:

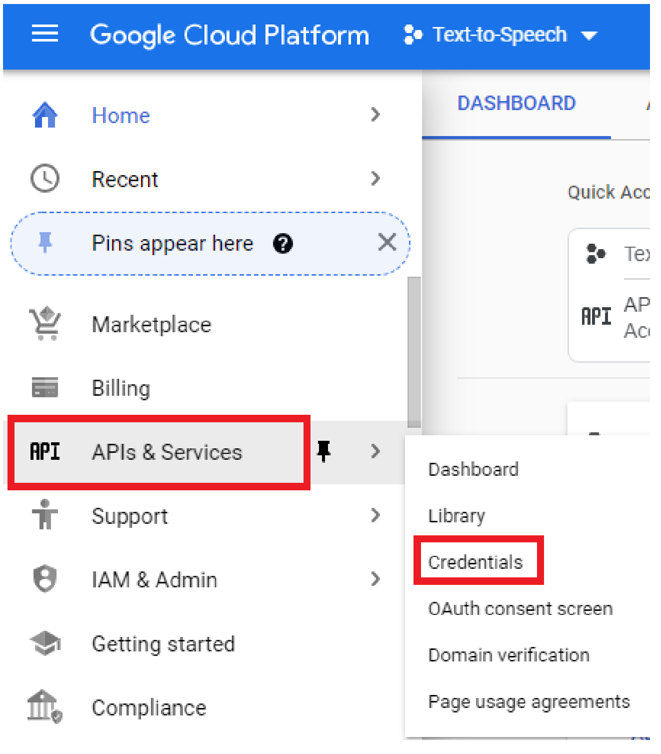
- **Step 1:** Go to the Google Cloud home page.
- **Step 2:** Click on the **New Project** button on the top left corner, and provide project name.



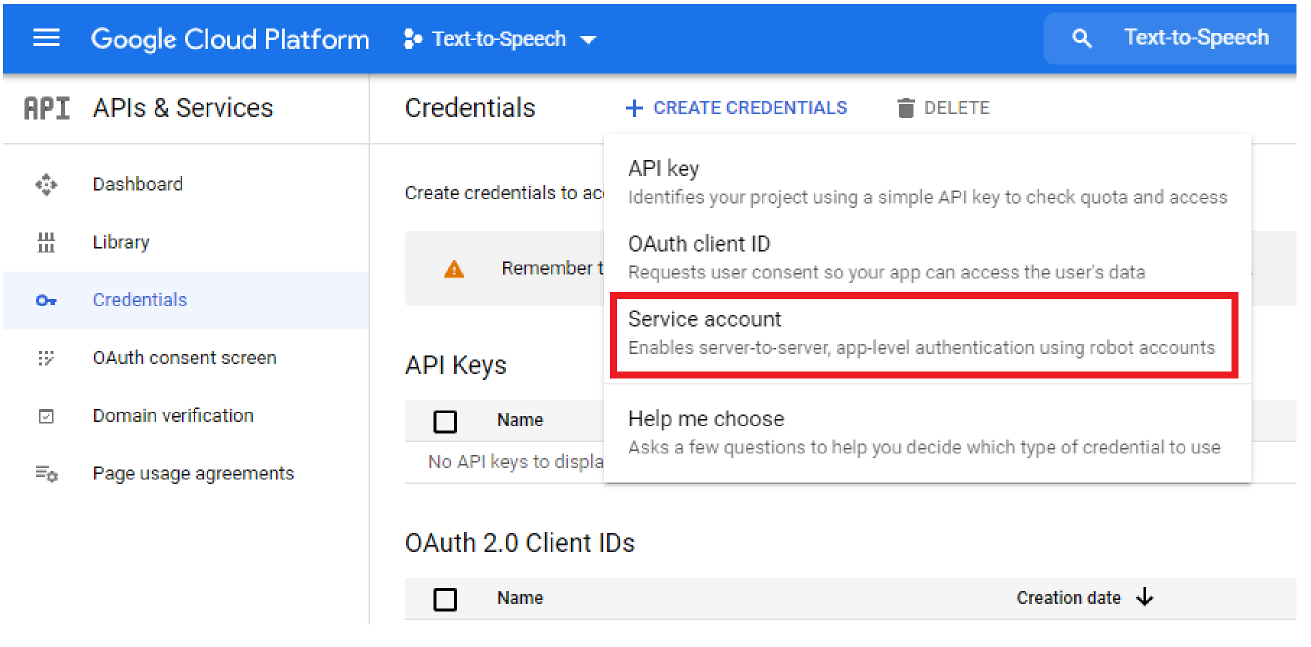
- **Step 3:** Afterwards search for **Text to Speech** keyword, and select **Cloud Text to Speech API** from the list.

- **Step 4:** Enable **Cloud Text to Speech API**.

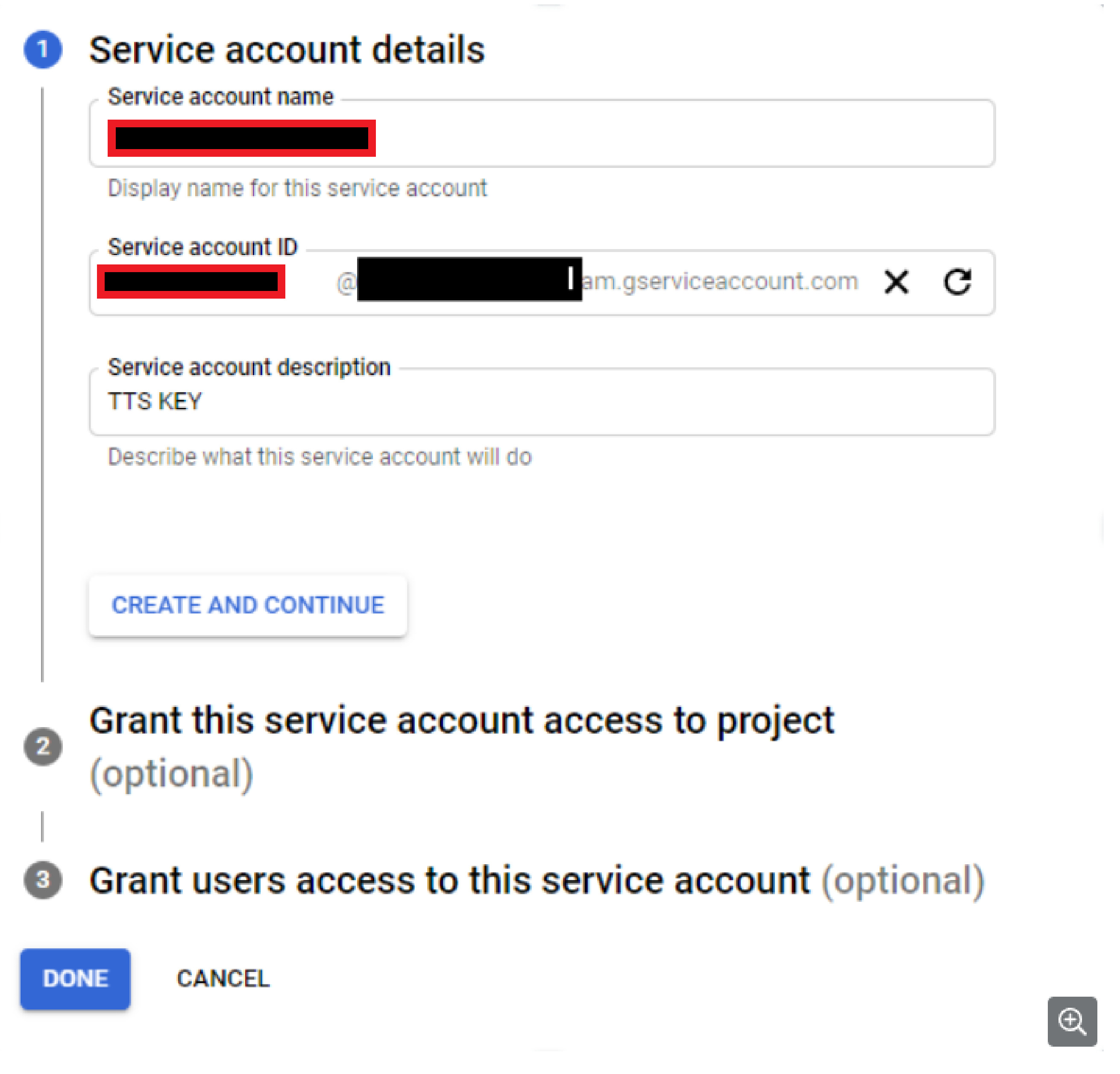
- **Step 5:** Go to API & Services -> Credentials.



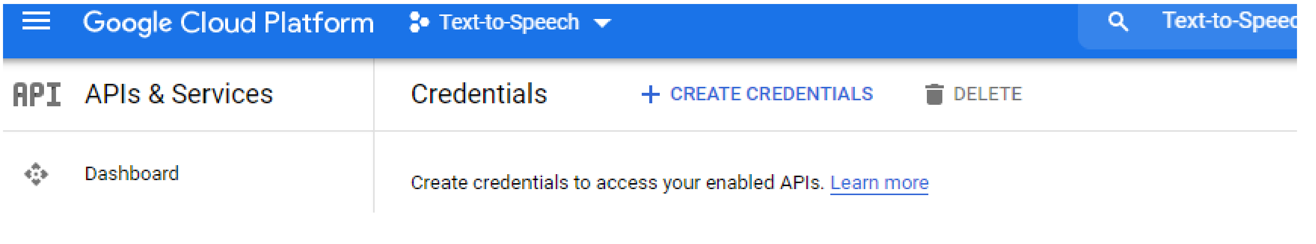
- **Step 6:** Click on **+ Create Credentials** and select **Service account** from the list.

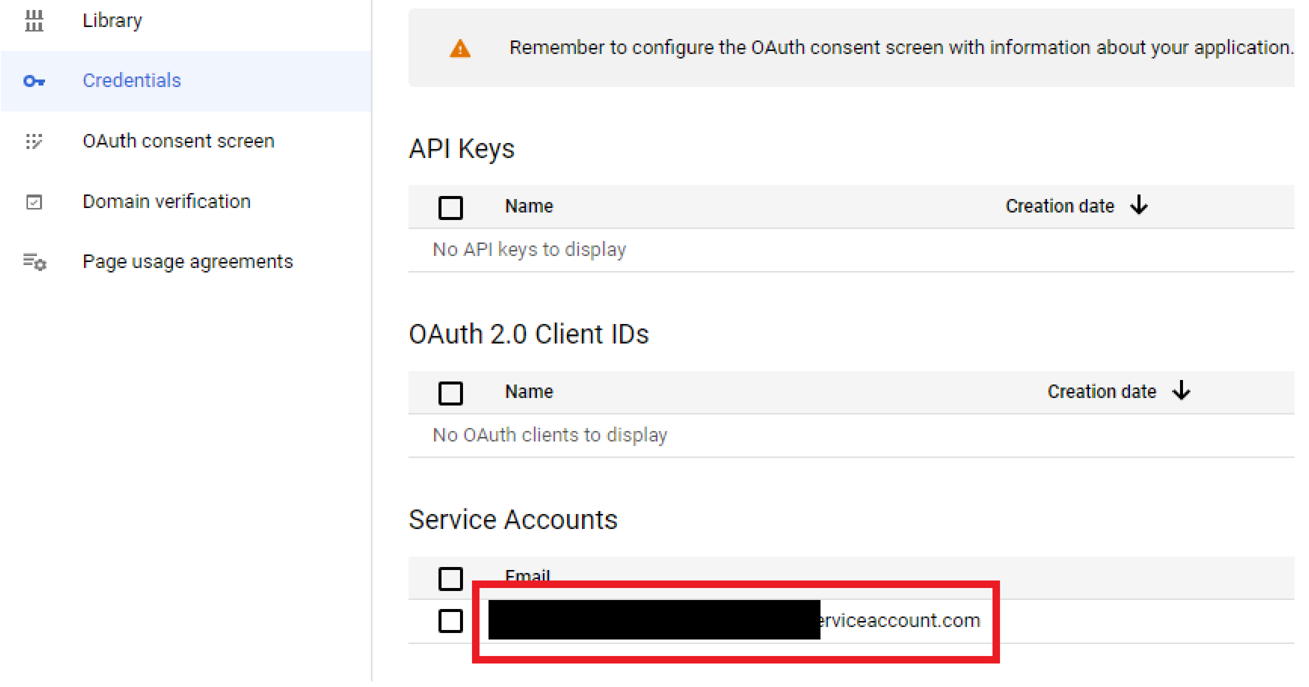


- **Step 7:** Provide **Service account name** and **Service account description**.

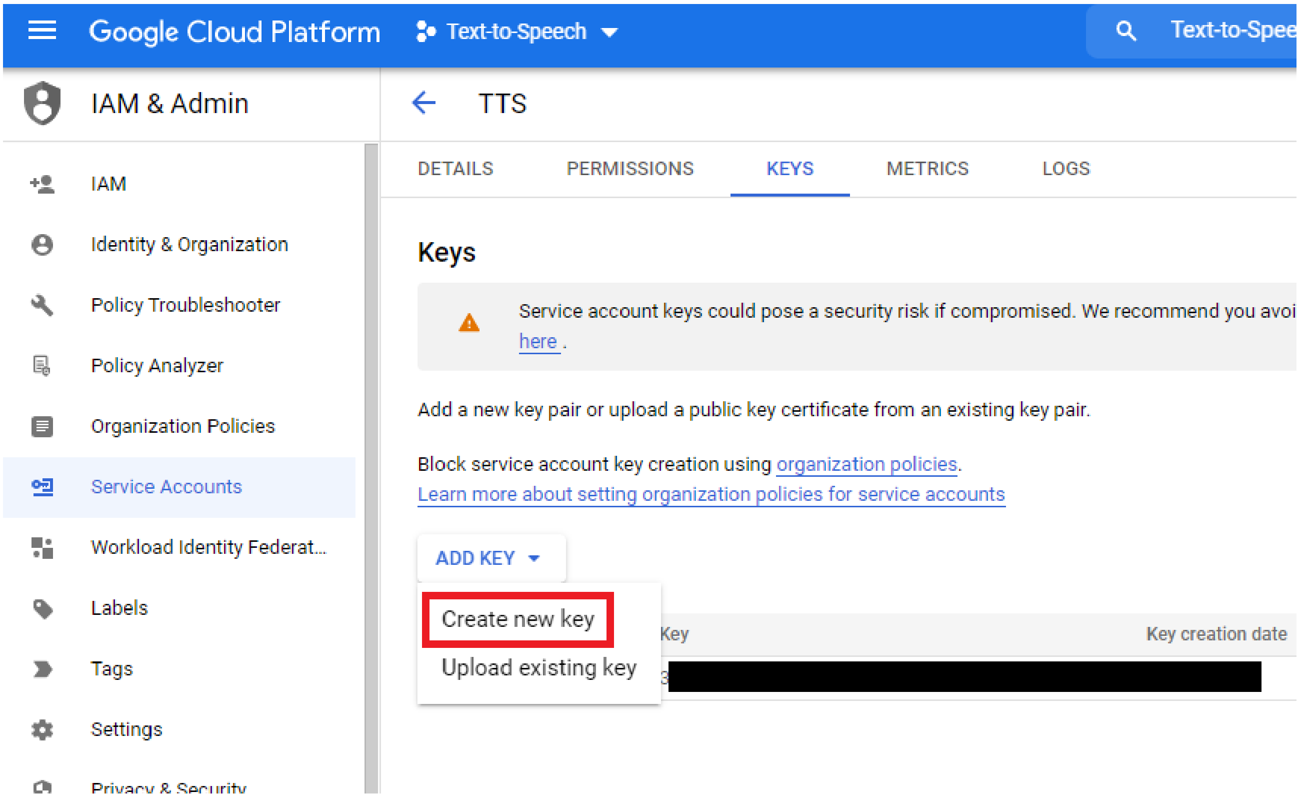


- **Step 8:** Click on your newly created **Service account name**.

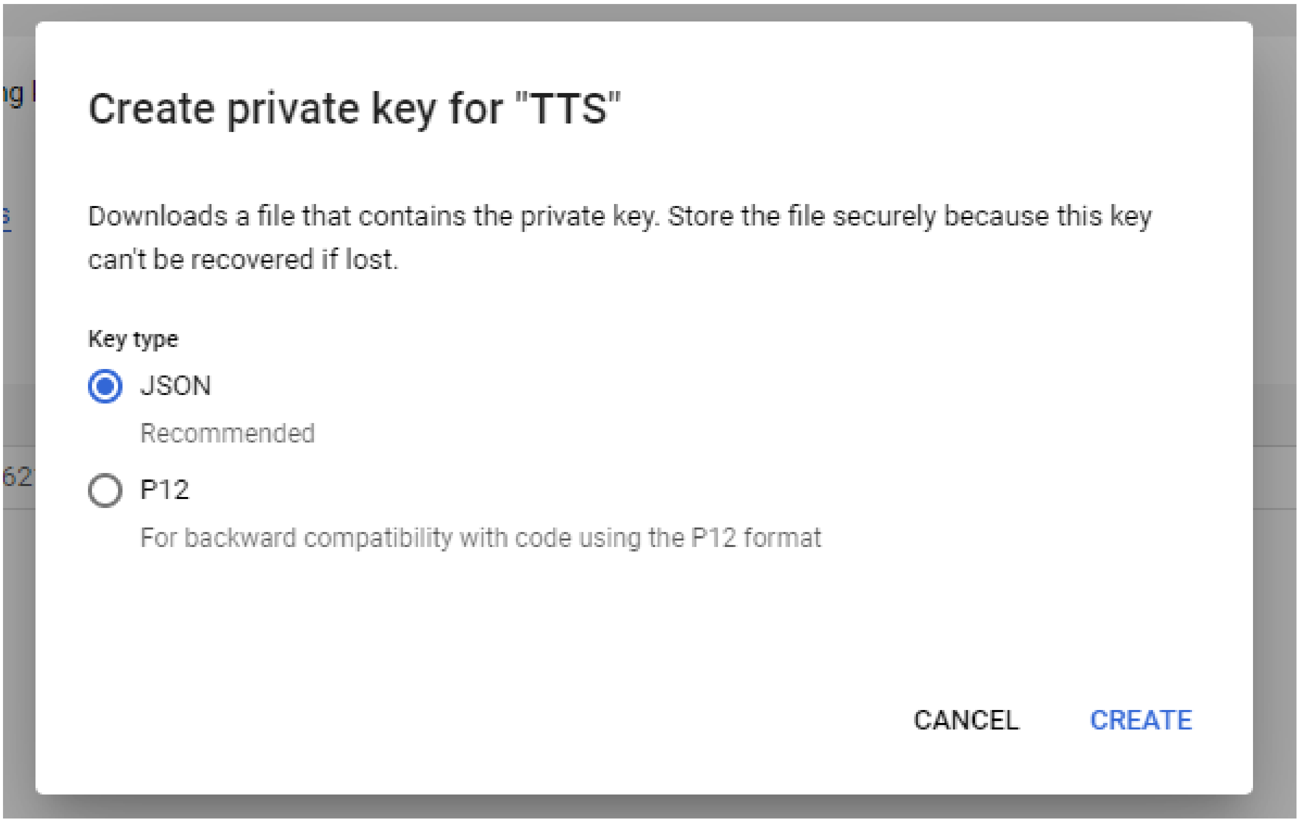




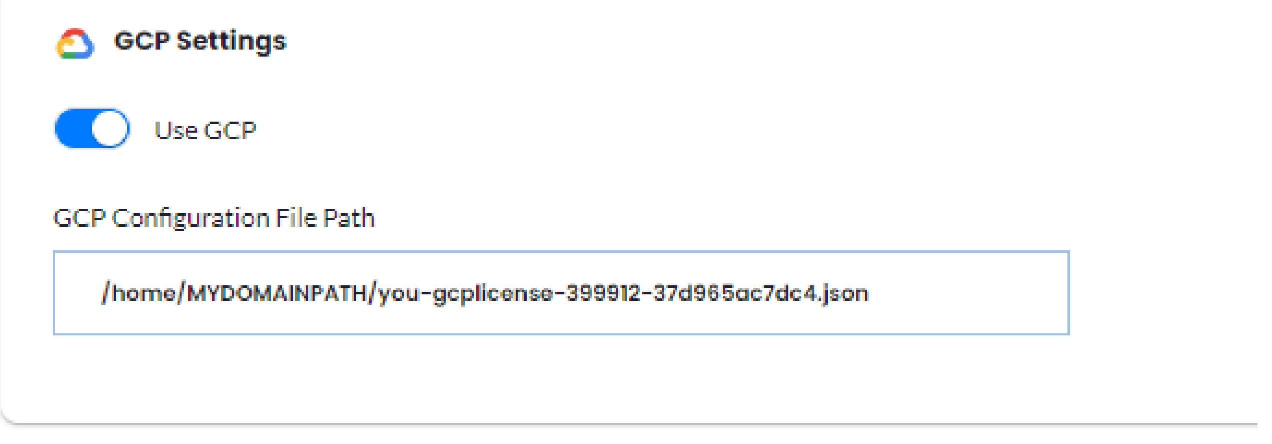
- **Step 9:** Under **Keys** tab, select **Add Key -> Create new key**.



- **Step 10:** Select **JSON** as key type. Click **Create**, it will store in on your desktop.



- **Step 11:** Warning! Make sure you store this json file outside of your public folder, it must be stored in a secure place accessible only to you on your hosting server.
- **Step 12:** Include the path for the json in **Admin Panel**, under **Davinci Settings**. JSON file must be stored outside of your **Public** folder. Show full path for the JSON file including JSON filename. Here is an example for cPanel:



AWS Account Creation

Getting started with AWS. If you already have AWS Account, skip to next section.

How to Create AWS Account:

- **Step 1:** Open the Amazon Web Services home page.
- **Step 2:** Choose **Create an AWS Account.**
  Note: If you signed in to AWS recently, choose **Sign in to the Console**. If **Create a new AWS account** isn't visible, first choose **Sign in to a different account**, and then choose **Create a new AWS account**.
- **Step 3:** Enter your account information, and then choose **Continue**. Be sure that you enter your account information correctly, especially your email address. If you enter your email address incorrectly, you can't access your account.
- **Step 4:** Choose **Personal** or **Professional**.
  Note: Personal accounts and professional accounts have the same features and functions.
- **Step 5:** Enter your company or personal information.
  Important: For professional AWS accounts, it's a best practice to enter the company phone number rather than a personal cell phone. Configuring a root account with an individual email address or a personal phone number can make your account insecure.
- **Step 6:** Read and accept the AWS Customer Agreement.
  Note: Be sure that you read and understand the terms of the AWS Customer Agreement.
- **Step 7:** Choose **Create Account and Continue**.
- **Step 8:** On the **Payment Information** page, enter the information about your payment method, and then choose **Verify and Add**.
  Note: If you want to use a different billing address for your AWS billing information, choose **Use a new address** before you choose **Verify and Add**
  Important: You can't proceed with the sign-up process until you add a valid payment method.
- **Step 9:** Verify your phone number. Enter a phone number where you can be reached in the next few minutes.
- **Step 10:** In a few moments, an automated system contacts you.
- **Step 11:** Enter the PIN you receive, and then choose **Continue**.
- **Step 12:** On the **Select a Support Plan** page, choose one of the available Support plans. For a description of the available Support plans and their benefits, see Compare AWS Support plans.
- **Step 13:** After you choose a Support plan, a confirmation page indicates that your account is being activated. Accounts are usually activated within a few minutes, but the process might take up to 24 hours.
- **Step 14:** You can sign in to your AWS account during this time. The AWS home page might display a **Complete Sign Up** button during this time, even if you've completed all the steps in the sign-up process.
- **Step 14:** When your account is fully activated, you receive a confirmation email. Check your email and spam folder for the confirmation email. After you receive this email, you have full access to all AWS services.

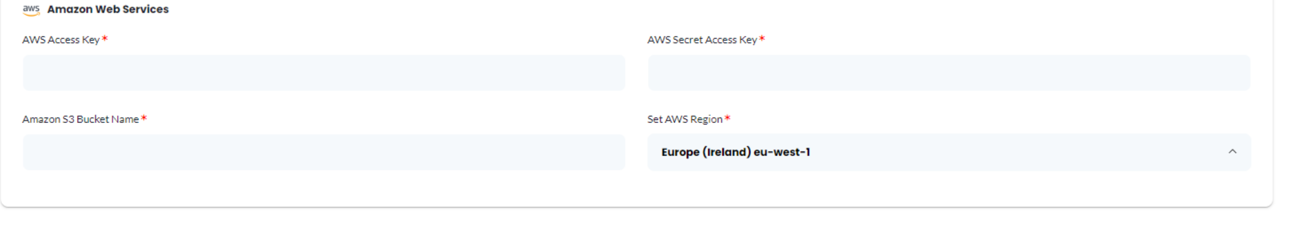
AWS IAM User Creation

You will need IAM user with programmatic access and with S3 Read & Write Policies attached.

Creating AWS IAM Users (via AWS Console):

- **Step 1:** Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
- **Step 2:** In the navigation pane, choose **Users** and then choose **Add user**.
- **Step 3:** Type the user name for the new user. This is the sign-in name for AWS.
  Note: User names can be a combination of up to 64 letters, digits, and these characters: plus (+), equal (=), comma (,), period (.), at sign (@), and hyphen (-). Names must be unique within an account.
- **Step 4:** Select the type of access this set of users will have. Select **programmatic access.**
- **Step 5:** Choose **Next: Permissions**.
- **Step 6:** On the **Set permissions** page, specify how you want to assign permissions to this set of new users. Choose **Attach existing policies to user directly**
- **Step 7:** Seach for AmazonS3FullAccess, choose it and afterwards select click **Next: Review**
  Note: If you are familar with IAM Policies, you can create custom policy that will match your needs and security requirements.
- **Step 8:** To view the users' access keys (access key IDs and secret access keys), choose **Show** next to each password and access key that you want to see. To save the access keys, choose **Download .csv** and then save the file to a safe location.
  Important: This is your only opportunity to view or download the secret access keys, and you must provide this information to your users before they can use the AWS API. Save the user's new access key ID and secret access key in a safe and secure place. **You will not have access to the secret keys again after this step.**
- **Step 9:** Include your **Access Key** and **Secret Access Key** to the **AWS Credentials Settings** tab in the **admin panel** as shown below.

Under Admin -> Studio Management -> Voiceover Studio Settings and Transcribe Studio Settings (in case if you are using Speech to Text feature of AWS) -> AWS, include your Access and Secret Access Keys:



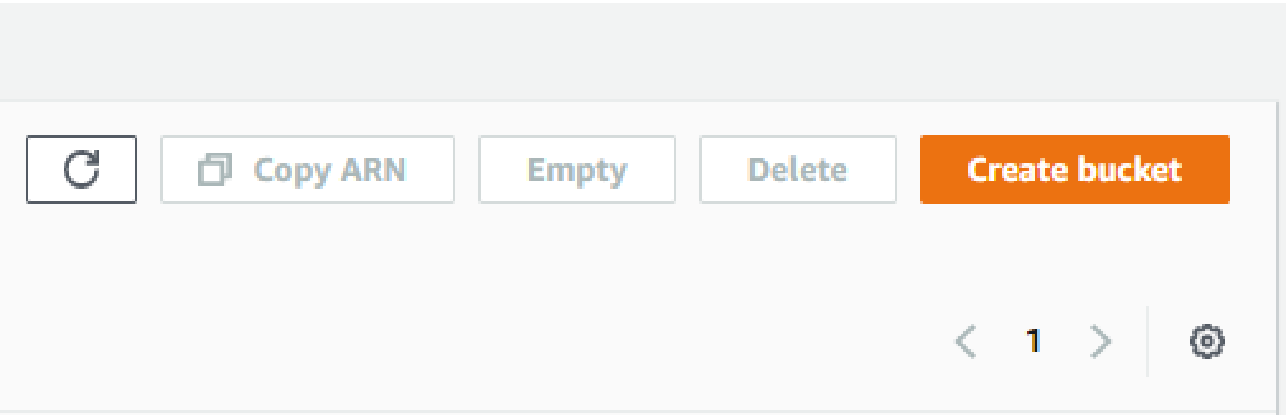
Amazon S3 Bucket Creation

You will need an S3 Bucket with public read & write access in the region that you prefer.

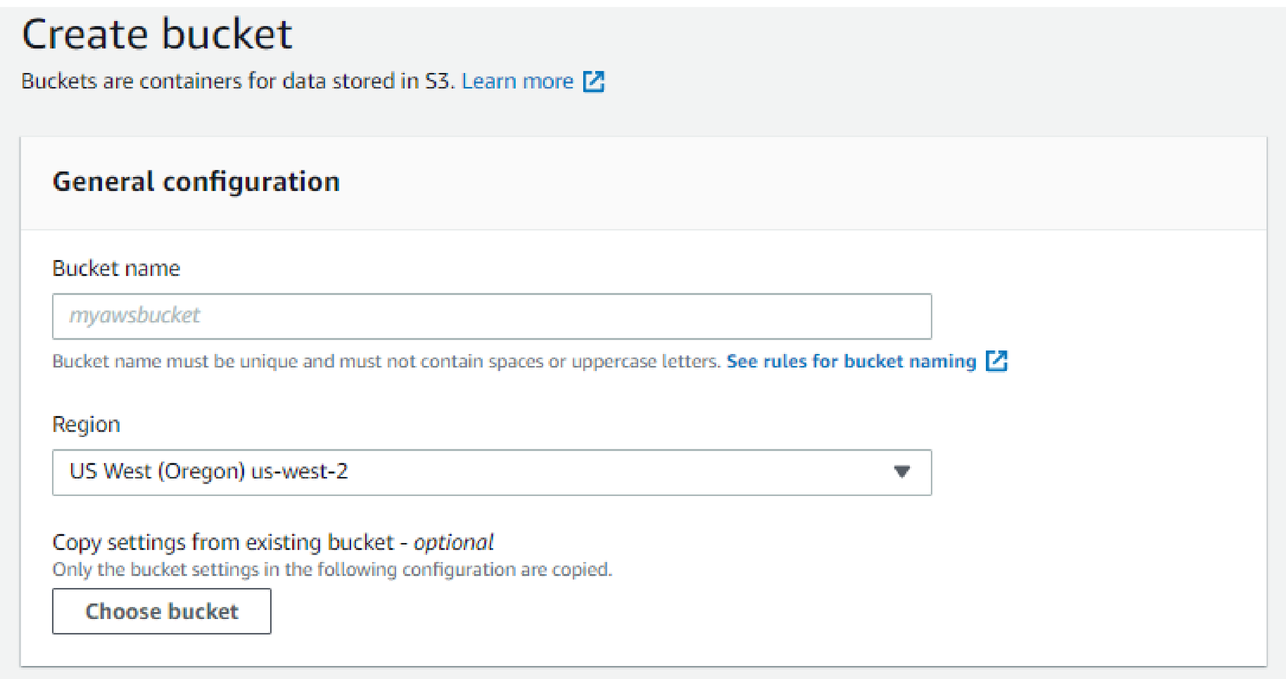
A bucket is owned by the AWS account that created it. By default, you can create up to 100 buckets in each of your AWS accounts. If you need additional buckets, you can increase your account bucket limit to a maximum of 1,000 buckets by submitting a service limit increase. For information about how to increase your bucket limit, see AWS Service Limits

Create an S3 bucket:

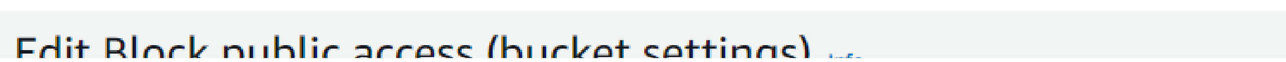
- **Step 1:** Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/.
- **Step 2:** Choose **Create bucket**.

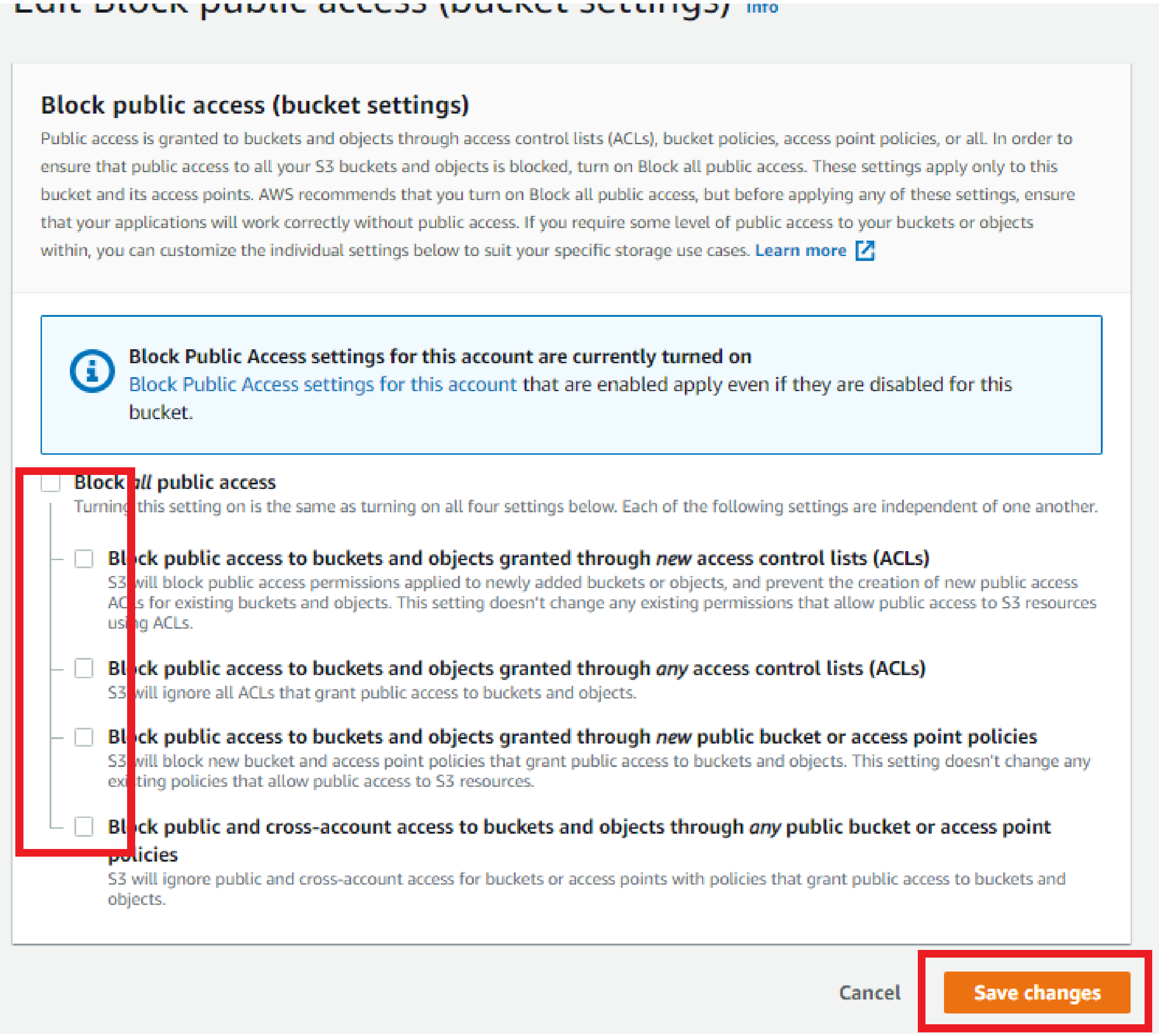


- **Step 3:** Enter Bucket Name and Select AWS Region where you would like to store your files and click **Next**. **Note:** Bucket name must be unique at a global level.



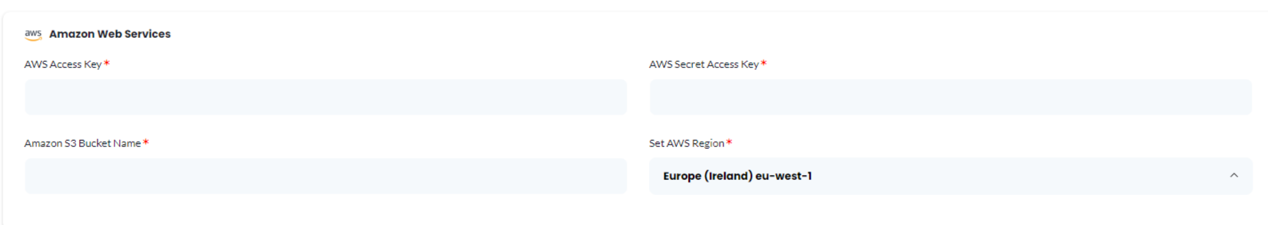
- **Step 4:** In the **Set Permissions** step, you need to turn off all 4 options if you want to make your S3 bucket completely public.





- **Step 5:** Review rest of the settings, or you can leave them as they are and click **Create Bucket.**
- **Step 6:** Include your **Bucket Name** and **Select your AWS Region** in the **AWS Credentials Settings** tab in the **admin panel** as shown below.

Voiceover Studio Settings and Transcribe Studio Settings (in case if you are using Speech to Text feature of AWS) -> AWS, include your S3 Bucket Name and select your AWS Region where bucket is hosted:

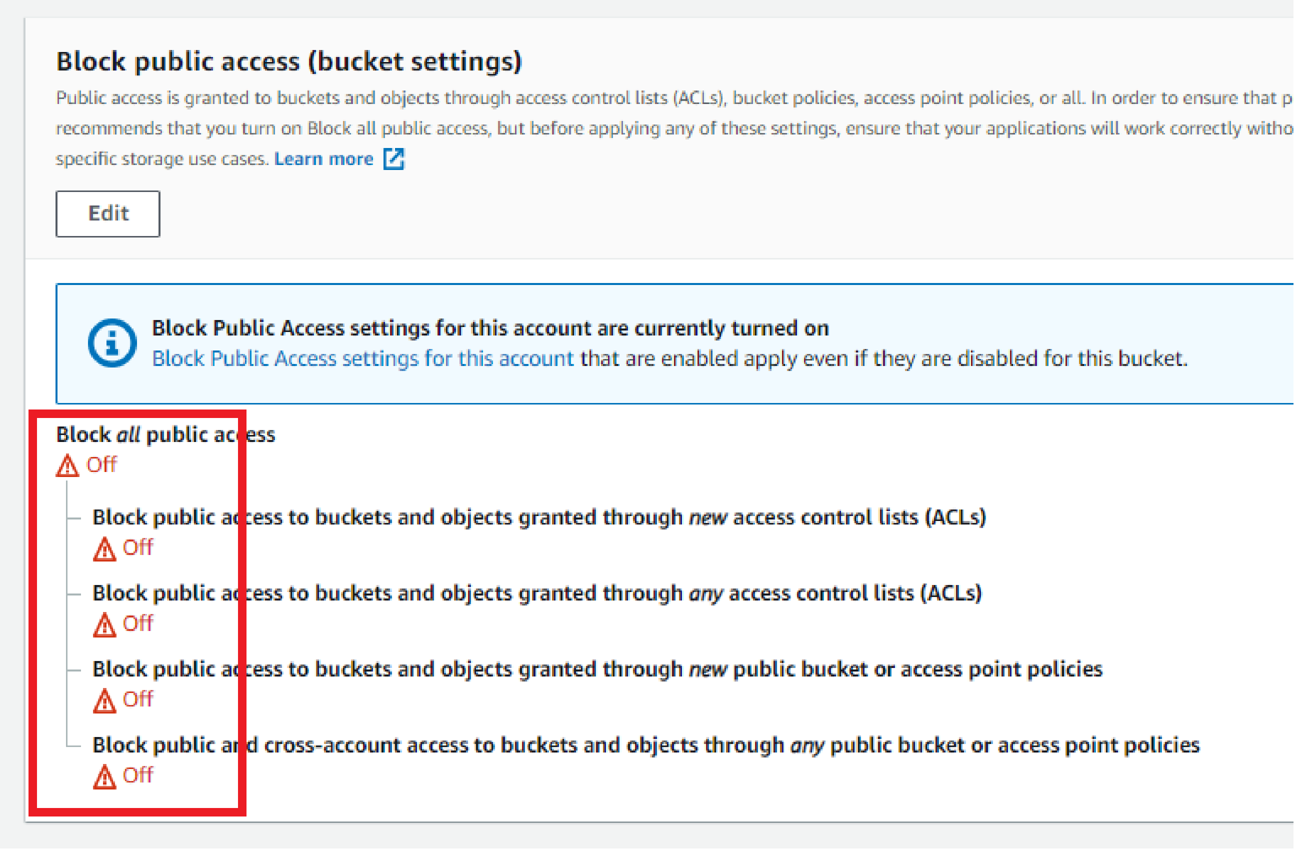


Amazon S3 Bucket Configuration

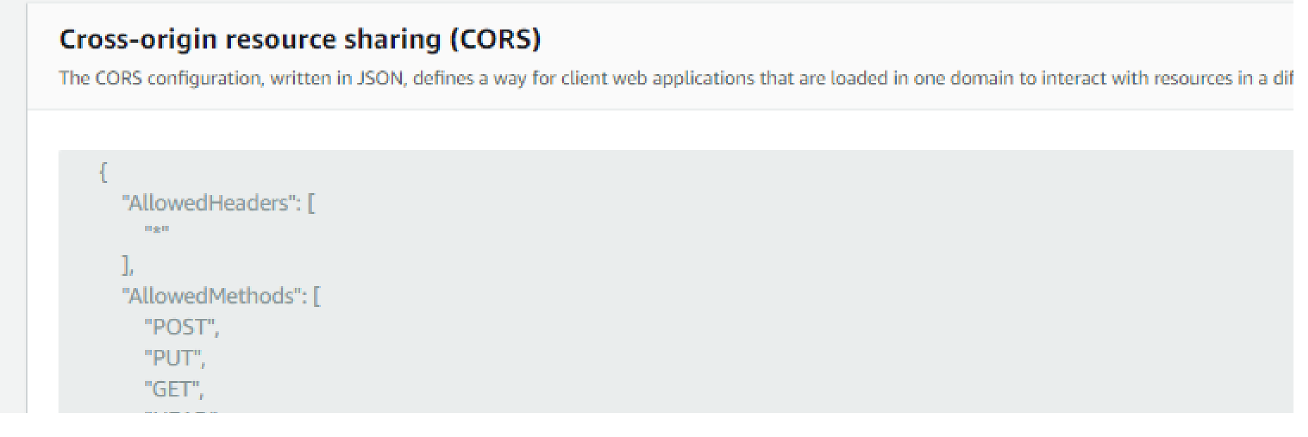
After successfully creating a bucket, add these addtional features for your S3 bucket.

Check these 4 steps:

- **Step 1:** Make sure your **Permissions -> Block Public Access** looks at least like this and has all 4 options **turned off**.



- **Step 2:** Add CORS settings as shown below in the **Permissions -> Cross-origin resource sharing (CORS) Configuration** section and click **Save**.



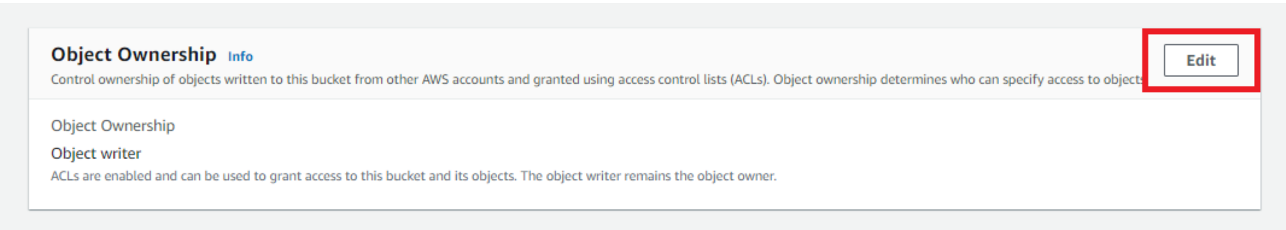
```
            "HEAD",
            "DELETE"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": [
            "ETag"
        ],
        "MaxAgeSeconds": 30000
    }
]
```

Copy paste these CORS settings to your bucket:

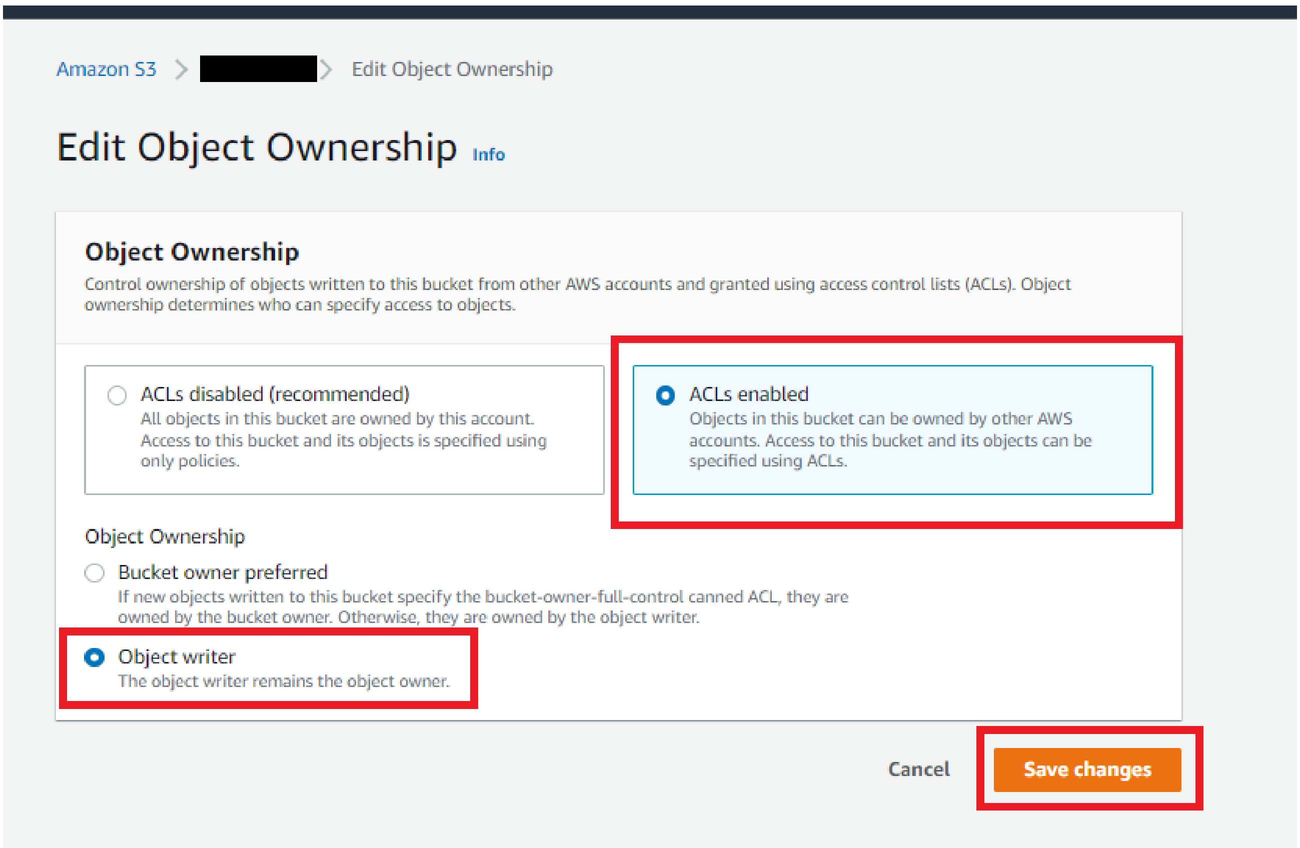
```
[
  {
    "AllowedHeaders": [
      "*"
    ],
    "AllowedMethods": [
      "POST",
      "PUT",
      "GET",
      "HEAD",
      "DELETE"
    ],
    "AllowedOrigins": [
      "*"
    ],
    "ExposeHeaders": [
      "ETag"
    ],
```

```
    "MaxAgeSeconds": 30000
  }
]
```

- **Step 3:** Go to your newly created S3 Bucket click on **Permissions -> Object Ownership** and click **Edit**.



- **Step 4:** Select **ACLs enabled** and **Object writer** and click **Save Changes**.



Configure Wasabi Storage

Getting started with Wasabi Storage.

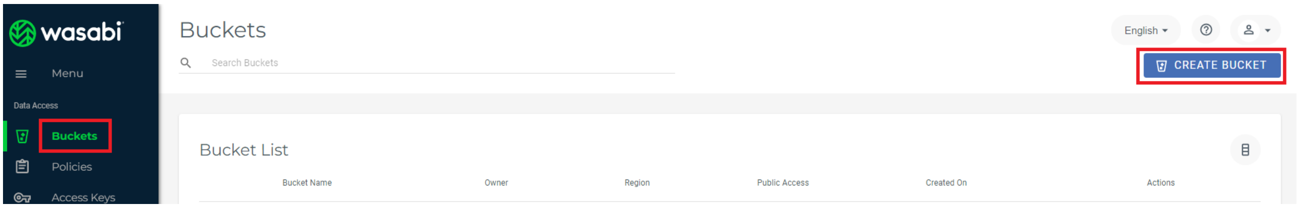
Apart from Local Server Storage, and Amazon S3 bucket, you can also use Wasabi Storage as your default storage for results.

Warning! Make sure you have a paid Wasabi account. Trial version of Wasabi account will not let you download file nor make your bucket public.

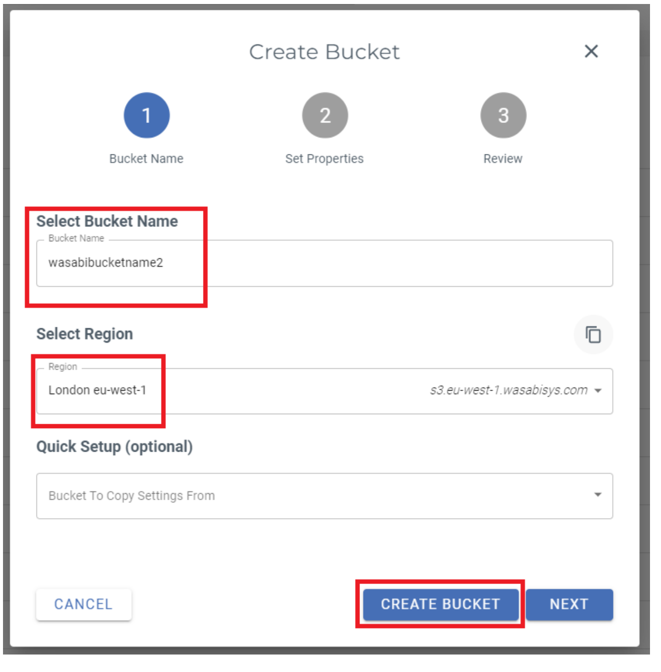
How to Configure Wasabi Storage:

- **Step 1:** Login to your Wasabi Account, select **Buckets** under Data Access menu panel and click **Create Bukcet**

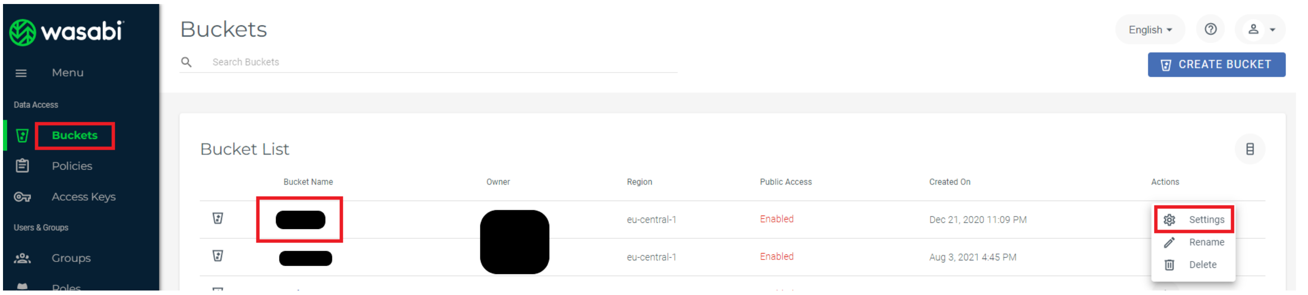
.



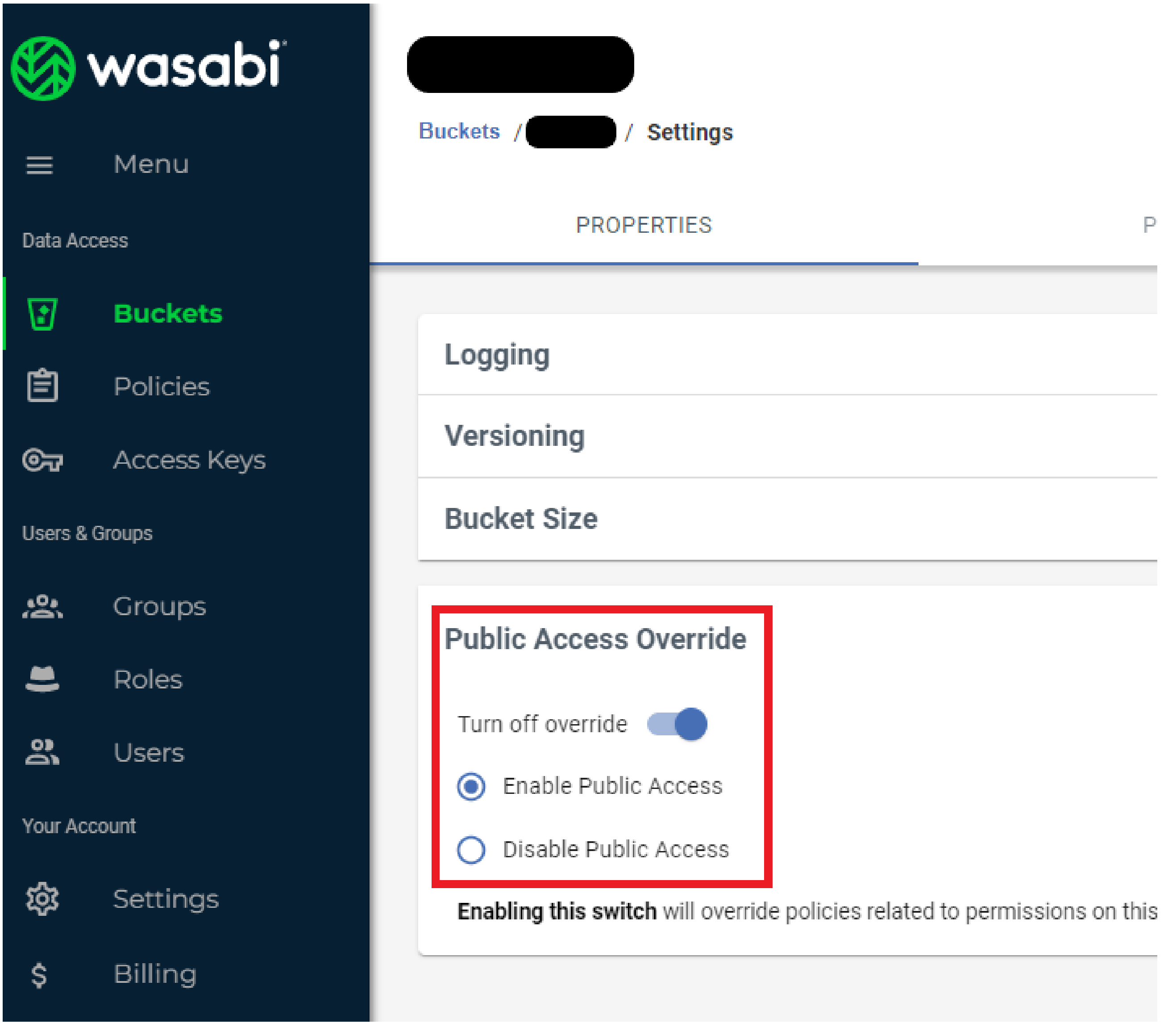
- **Step 2:** Prorive a unique **Bucket Name** and select your **Region** and click **Create Bucket**. Make a note of both them, you will be including them into the script at the last step.



- **Step 3:** After the bucket is created, select your bucket from the list and under Actions tab, click on **Settings**.

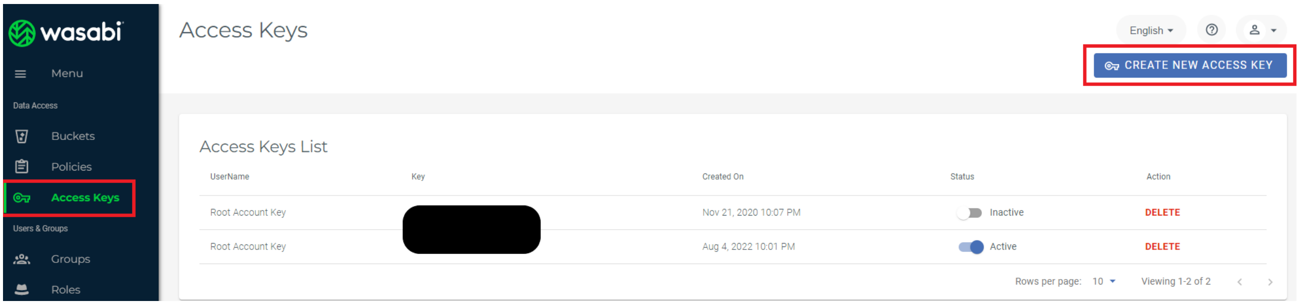


- **Step 4:** In the Settings page, make sure you **Enable Public Access** for this bucket.

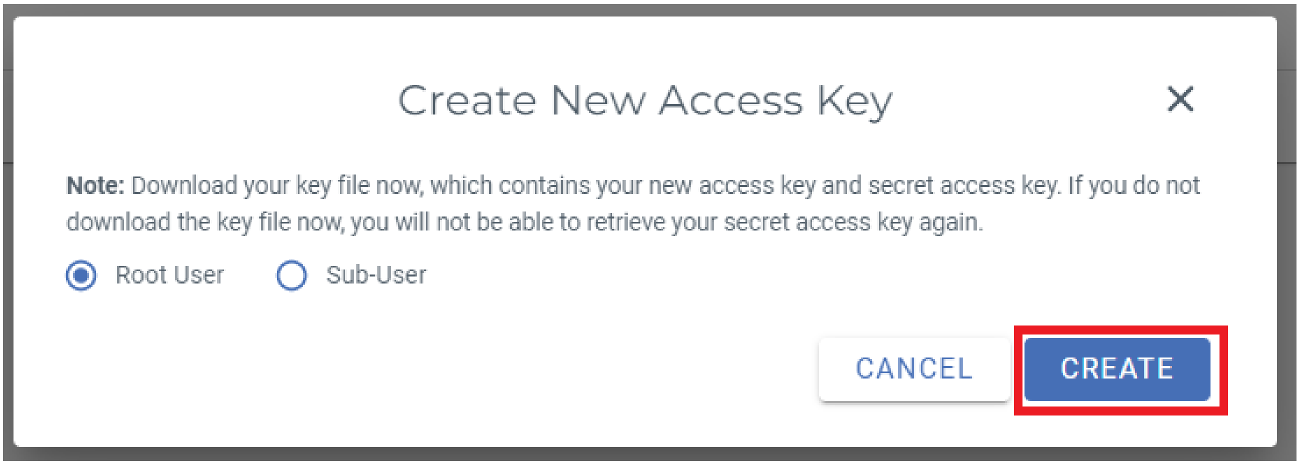


- **Step 5:** Now when you are done creating and configuring your Wasabi Bucket, you will need to create **Access Keys** to be able to use your Wasabi Account.

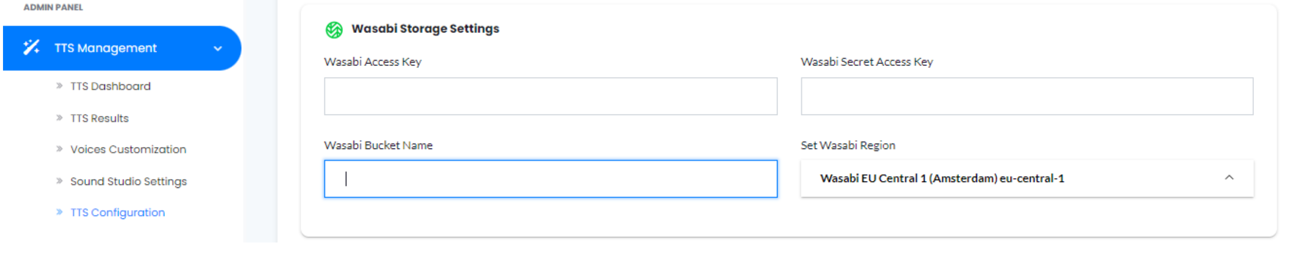
- **Step 6:** Go to **Access Keys** under Data Access menu panel and click on **Create New Access Key**.



- **Step 7:** Click **Create** button and download or copy your keys immediate as you get to see your Secret Access key only once.



- **Step 6:** Enter your Wasabi Access Keys, Wasabi Bucket and Wasabi Bucket Region in the Admin Panel of the script.



Room Designer Setup

Create a VPS account - Get one from AWS (One year of full credit) or Digital Ocean. Make an ubuntu instance (Preferably 20.04)

In order to setup Room Designer, it need to have external domain or subdomain. You need to point your VPS to that subdomain or domain. It's already included how to do it in the next steps.

SSH into the instance and write the following commands

sudo apt update

sudo apt install apache2

sudo apt install mysql-server

sudo mysql

then write

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'mynewpassword';

exit mysql

sudo mysql_secure_installation

then follow the steps

sudo apt install ca-certificates apt-transport-https software-properties-common

sudo add-apt-repository ppa:ondrej/php

sudo apt install php8.0-cli php8.0-common php8.0-imap php8.0-redis php8.0-xml php8.0-zip php8.0-mbstring php8.0-bcmath

sudo nano /etc/apache2/sites-available/server.conf

then paste the following text: (replace example.com with your new domain and point the IP of your VPS to the domain dashboard)

```
<VirtualHost *:80>

  ServerAdmin webmaster@example.com

  ServerName example.com



  DocumentRoot /var/www/designer

  <Directory />

    Options FollowSymLinks

    AllowOverride all
```

```
   </Directory>

   <Directory /var/www/designer/>

      Options Indexes FollowSymLinks MultiViews

      AllowOverride all

      Order allow,deny

      allow from all

   </Directory>

   ErrorLog ${APACHE_LOG_DIR}/error-sub.example.com.log

   # Possible values include: debug, info, notice, warn, error, crit,

   # alert, emerg.

   LogLevel warn

   CustomLog ${APACHE_LOG_DIR}/access-sub.example.com.log combined

   # sub:

 ProxyPreserveHost On

   ProxyRequests Off

   ProxyPass / http://localhost:3000/

   ProxyPassReverse / http://localhost:3000/

RewriteEngine on

RewriteCond %{SERVER_NAME} =example.com

RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]

   </VirtualHost>
```

sudo a2ensite server

sudo a2dissite 000-default.conf

sudo a2enmod rewrite

sudo a2enmod proxy

sudo a2enmod proxy_http

sudo a2enmod proxy_balancer

sudo a2enmod lbmethod_byrequests

sudo mkdir /var/www/designer

sudo chmod -R a+rwx /var/www/

sudo systemctl restart apache2

sudo apt install certbot python3-certbot-apache

sudo certbot --apache (to install SSL - leave if you don't want to)

sudo /bin/dd if=/dev/zero of=/var/swap.1 bs=1M count=1024

sudo /sbin/mkswap /var/swap.1

sudo /sbin/swapon /var/swap.1

curl -sL https://deb.nodesource.com/setup_19.x | sudo -E bash -

sudo apt install nodejs

sudo npm install pm2 -g

Decompress the source code "**Designer.zip**" and

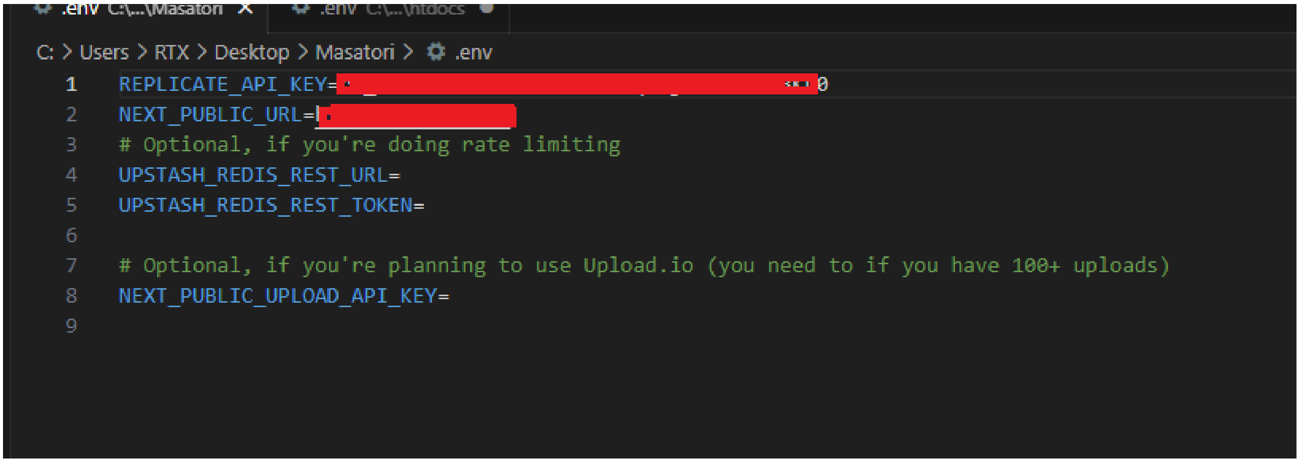
Creating a account on Replicate to get an API key.

1. Go to Replicate to make an account.
2. Click on your profile picture in the top right corner, and click on "Dashboard".
3. Click on "Account" in the navbar. And, here you can find your API token, copy it. (Remember you need to activate billing in order for API key to work always without crashing)

Go to .env file and fill the following:

REPLICATE_API_KEY=YourReplicateAPIKEY

NEXT_PUBLIC_URL=https://PromptPressURL.com (Warning: NOT the URL of Designer, but put the URL of PromptPress) (Warning 2: If this URL has SSL installed then include https, if it doesn't include http only)



```
.env C:\...\Masatori ×   .env C:\...\htdocs ●
C: > Users > RTX > Desktop > Masatori > .env
REPLICATE_API_KEY=[redacted]0
NEXT_PUBLIC_URL=[redacted]
# Optional, if you're doing rate limiting
UPSTASH_REDIS_REST_URL=
UPSTASH_REDIS_REST_TOKEN=

# Optional, if you're planning to use Upload.io (you need to if you have 100+ uploads)
NEXT_PUBLIC_UPLOAD_API_KEY=
```

Upload all files in **/var/www/designer**

move your line to the **designer** folder using:

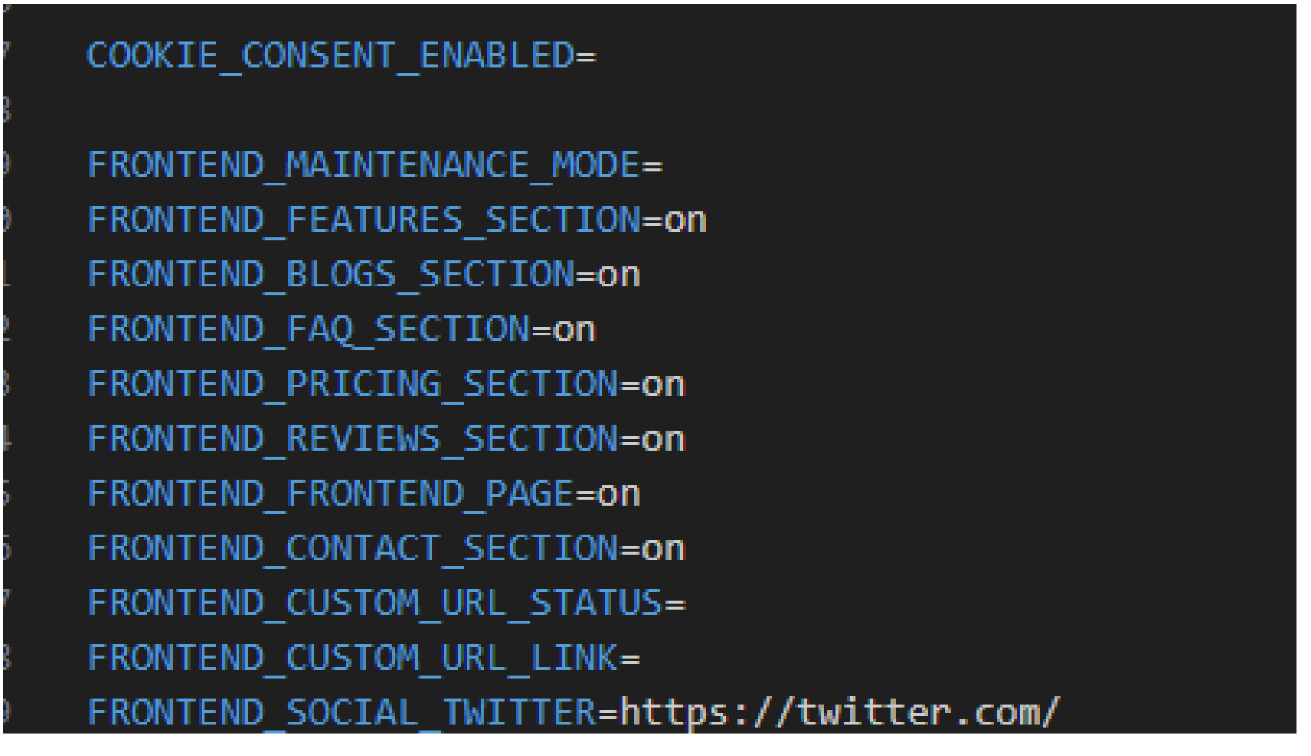
cd /var/www/designer

run the following commands:

sudo npm install

sudo pm2 start npm --name "designer" -- run dev -- --port 3000

After finishing setting up Designer, go to PromptPress Laravel Project, go to **.env** and paste DesignerURL here (make sure you include http://yoursdesignerurl.com)



```
COOKIE_CONSENT_ENABLED=

FRONTEND_MAINTENANCE_MODE=
FRONTEND_FEATURES_SECTION=on
FRONTEND_BLOGS_SECTION=on
FRONTEND_FAQ_SECTION=on
FRONTEND_PRICING_SECTION=on
FRONTEND_REVIEWS_SECTION=on
FRONTEND_FRONTEND_PAGE=on
FRONTEND_CONTACT_SECTION=on
FRONTEND_CUSTOM_URL_STATUS=
FRONTEND_CUSTOM_URL_LINK=
FRONTEND_SOCIAL_TWITTER=https://twitter.com/
```

```
FRONTEND_SOCIAL_FACEBOOK=https://facebook.com/
FRONTEND_SOCIAL_LINKEDIN=https://linkedin.com/
FRONTEND_SOCIAL_INSTAGRAM=https://instagram.com/


DESIGNER_SRC=YourDesignerURL.com
```