

# Message Recall: Deployment Guide

Last Updated: Wed May 25, 2016

## CONTENTS

### [Overview](#)

#### [1. Create a Google Apps Domain Super-Admin 'Role' account](#)

#### [2. Create the App Engine hosting environment](#)

##### [2.1 Create the application in App Engine](#)

##### [2.2 Enable billing for your application](#)

#### [3. Configure the Google APIs Console Project](#)

##### [3.1 Enable the Admin SDK API](#)

##### [3.2 Create Credentials \(a Project Service Account\) File](#)

##### [3.3 Enable the Google Cloud Billing API](#)

#### [4. Configure your Google Apps Domain for the App Engine application](#)

##### [4.1 Add the App Engine service-account to the domain security model](#)

#### [5. Prepare the source code](#)

##### [5.1 Get the source code](#)

##### [5.2 Make local updates](#)

##### [5.3 Create a service account certificate .pem file](#)

#### [6. Download the Google App Engine SDK](#)

#### [7. Upload the source code into the AppEngine host](#)

#### [8. Test the running application](#)

### [Common Problems](#)

### [Appendices](#)

#### [Getting Help](#)

## Overview

Message Recall is a Google-AppEngine-hosted application designed to run in a **single Google Apps domain**. The following instructions describe a detailed process for deploying the application:

1. Create a Google Apps Domain Super-Admin 'Role' account
2. Create the App Engine hosting environment
3. Configure the Google APIs Console Project
4. Configure your Google Apps Domain for the App Engine application
5. Prepare the source code
6. Download the Google App Engine SDK
7. Upload the source code into the App Engine host
8. Test the running application

The end to end process should take about 30 minutes (not including study time to learn about App Engine or applications).

NOTE: We will show examples using a Google Apps Domain named gappslabs.com.

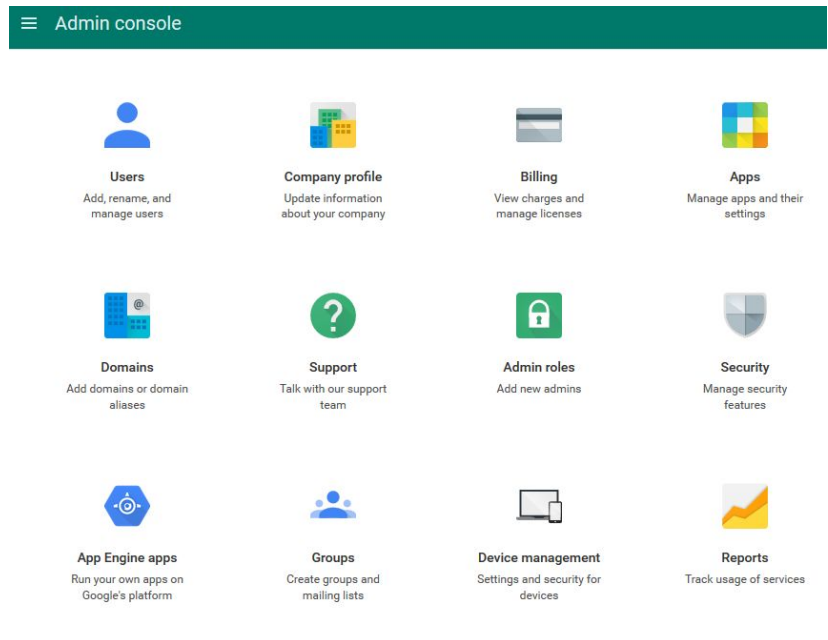
---

# 1. Create a Google Apps Domain Super-Admin 'Role' account

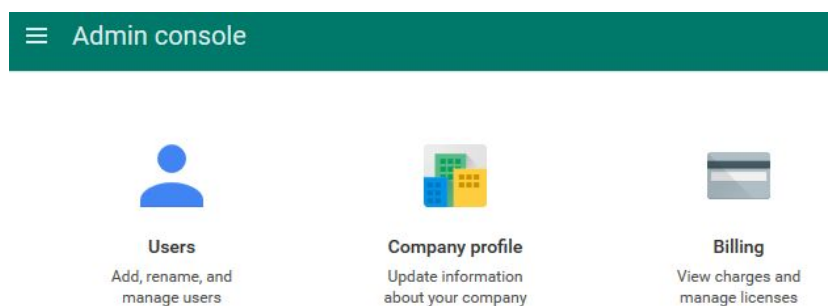
For additional reference, the following articles from the Google Support center discuss accounts and Admin roles:

- [Adding users individually](#)
- [Pre-built administrator roles](#)
- [Assigning Administrator roles to users](#)

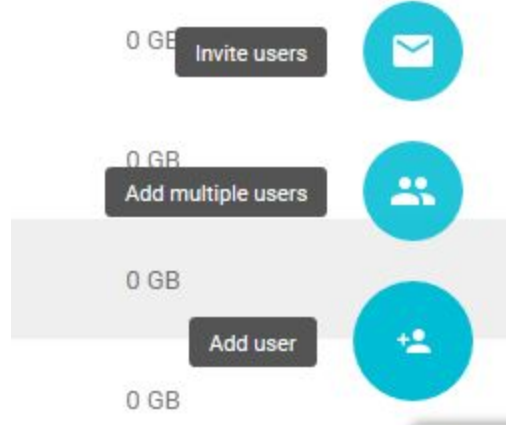
In your Google Apps Domain, while logged in as a Domain Admin, navigate to the Admin console at: <https://admin.google.com>.



Select 'Users' to create the Role account.



Add a new user by clicking 'Add user' that appears after clicking the + graphic.



Enter 'First name', 'Last name' and 'Primary email address' (we used message\_recall\_role for this example) then choose 'Create'.

On the next confirmation page, note the Password generated for the new account. You may choose to send email or print details.

Choose Done.

### Create a new user ×

RecallFirst RecallLast

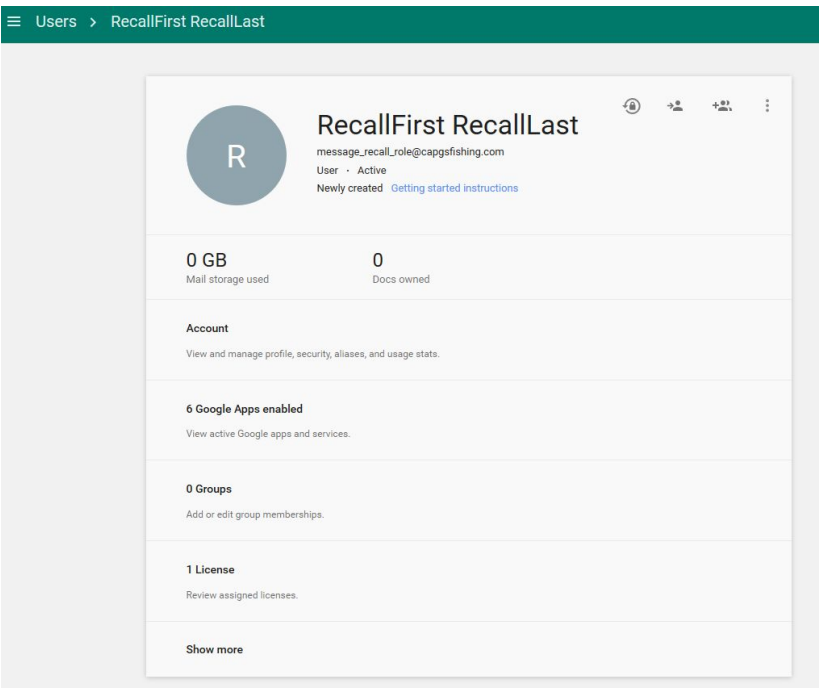
message\_recall\_role@gappslabs.com

Temporary password will be assigned - [Set Password](#)

ADDITIONAL INFO

CANCEL CREATE

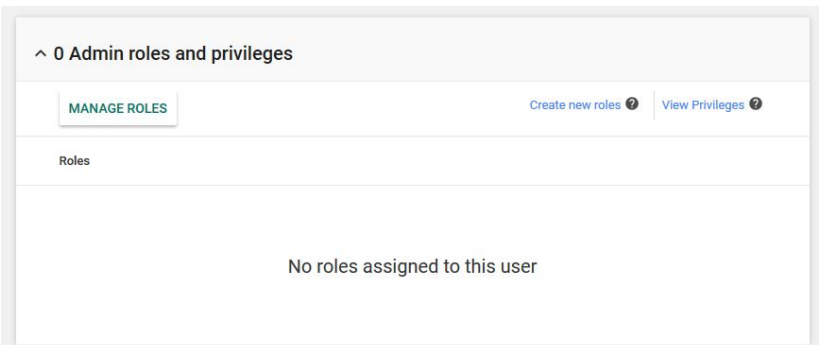
Select the new user,  
choose 'Show more' details.



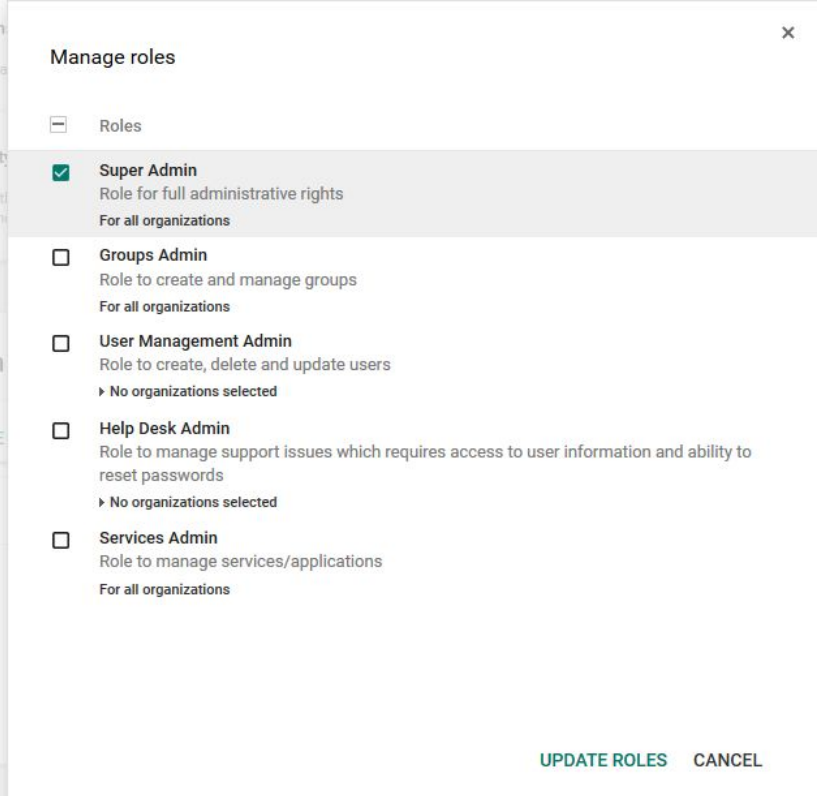
Choose:  
'Admin roles and its  
privileges'.



Choose: 'Manage roles'.



Select: 'Super Admin' and choose 'Update Roles'.



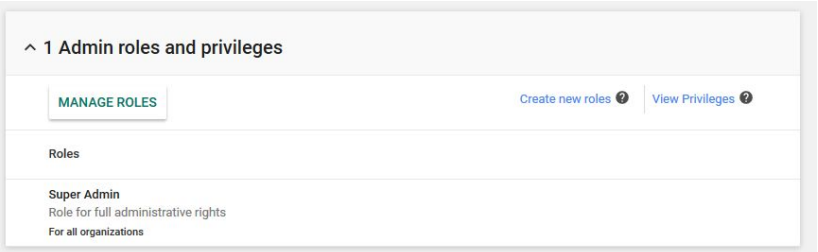
Manage roles

Roles

- ☒ **Super Admin**  
Role for full administrative rights  
For all organizations
- ☐ **Groups Admin**  
Role to create and manage groups  
For all organizations
- ☐ **User Management Admin**  
Role to create, delete and update users  
▶ No organizations selected
- ☐ **Help Desk Admin**  
Role to manage support issues which requires access to user information and ability to reset passwords  
▶ No organizations selected
- ☐ **Services Admin**  
Role to manage services/applications  
For all organizations

UPDATE ROLES CANCEL

You should notice your user now has '1' next to 'Admin roles and its privileges'.



^ 1 Admin roles and privileges

MANAGE ROLES Create new roles View Privileges


Roles

- Super Admin**  
Role for full administrative rights  
For all organizations

Sign out.

Sign back in as message\_recall\_role@yourdomain.com.

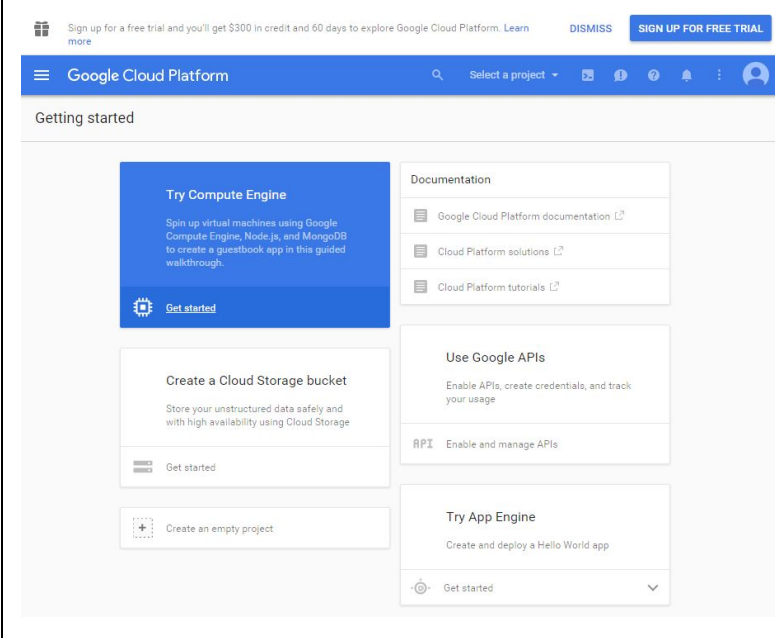
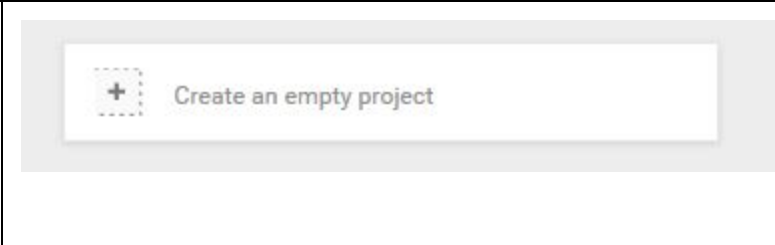
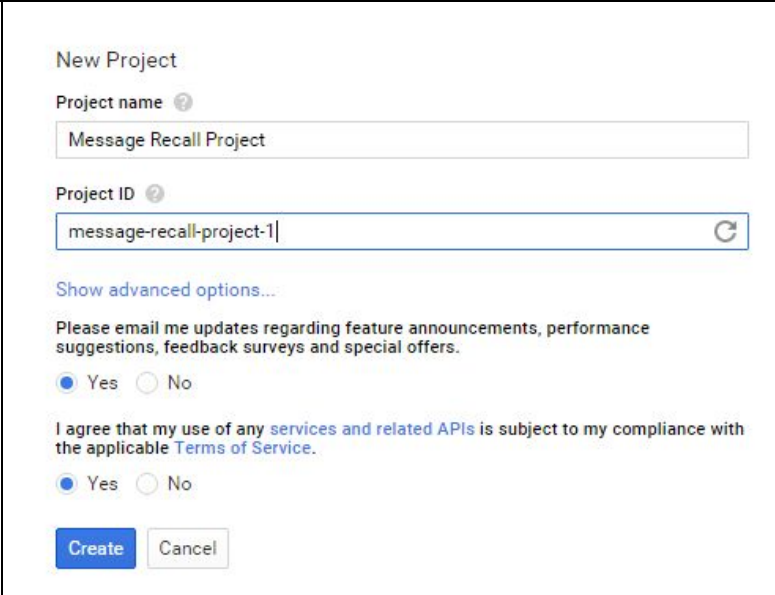
[You will need to Accept the Google Apps (Online) Agreement].

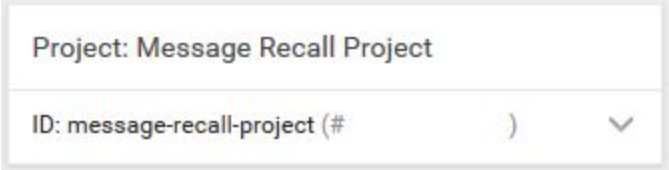

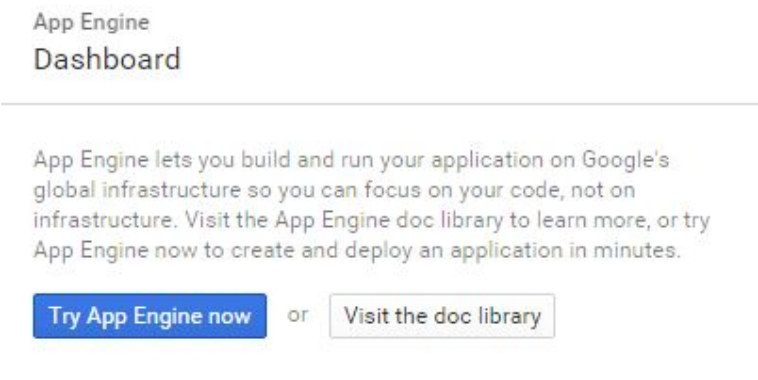


message\_recall\_role@gappslabs.com

## 2. Create the App Engine hosting environment

### 2.1 Create the application in App Engine

<p>While logged in as message_recall_role, navigate to the 'Google Cloud Platform' console at <a href="https://console.cloud.google.com">https://console.cloud.google.com</a></p> <p>.</p>	
<p>Choose 'Create an empty project'.</p> <p>(You may be requested to verify your account via text message or voice call.)</p>	
<p>Fill out the new project form.</p> <p>A Project name like 'Message Recall &lt;yourdomainname&gt;' is recommended. This maps to an application 'Project ID' of message-recall-&lt;yourdomain&gt;. You'll need this 'Project ID' later.</p> <p>Select 'I agree...'</p> <p>Choose 'Create'.</p>	

<p>You should see the new Project.</p>	 A screenshot of a Google Cloud interface showing a project selection dropdown. The top part of the dropdown is labeled "Project: Message Recall Project". Below it, the selected project is shown as "ID: message-recall-project (# )" with a downward arrow icon on the right.
<p>Now navigate to App Engine.</p> <p>Open the left-sidebar menu and choose: 'COMPUTE' &gt; 'App Engine'.</p>	 App Engine
<p>Choose 'Try App Engine now'</p>	 A screenshot of the "App Engine Dashboard" page. At the top, it says "App Engine Dashboard". Below a horizontal line, there is a paragraph: "App Engine lets you build and run your application on Google's global infrastructure so you can focus on your code, not on infrastructure. Visit the App Engine doc library to learn more, or try App Engine now to create and deploy an application in minutes." At the bottom, there are two buttons: a blue button labeled "Try App Engine now" and a white button labeled "Visit the doc library", separated by the word "or".



Choose 'View your project dashboard'.

Make sure 'Python' is selected.

## Try Google App Engine now

Creating an App Engine app is easy, and it's free to start. Upload your app and share it with users right away, at no charge and with no commitment required.

### 1. SELECT YOUR LANGUAGE



### 2. INSTALL GOOGLE APP ENGINE SDK

Download and install the [Google App Engine SDK for Python](#). Note that the instructions vary by operating system.

The App Engine SDK includes command-line tools and libraries for developing apps on your computer, testing them with a local server, and deploying them to App Engine.

### 3. DOWNLOAD THE SAMPLE APP

Select a sample app:



Download the [Flask starter code](#) and unpack it to create the project directory.

### 4. DEPLOY THE APP

In your local development environment, change into the directory in which you unpacked the sample app and deploy your app using the following command:

```
$ appcfg.py -A message-recall-project-1 update app.yaml
```

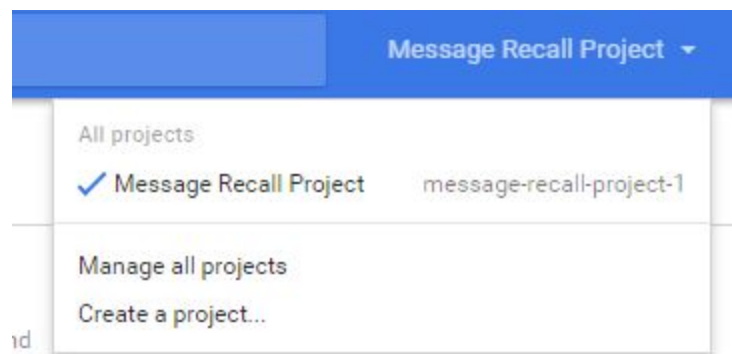
After deploying your app, you can visit it with your browser at this URL:

[message-recall-project-1.appspot.com](#)

That's it! You're running on Google App Engine. Go to your project dashboard to see how your app is performing.

[View your project dashboard](#)

At this point the App Engine project is created. This can be observed by clicking the 'Message Recall Project' dropdown.

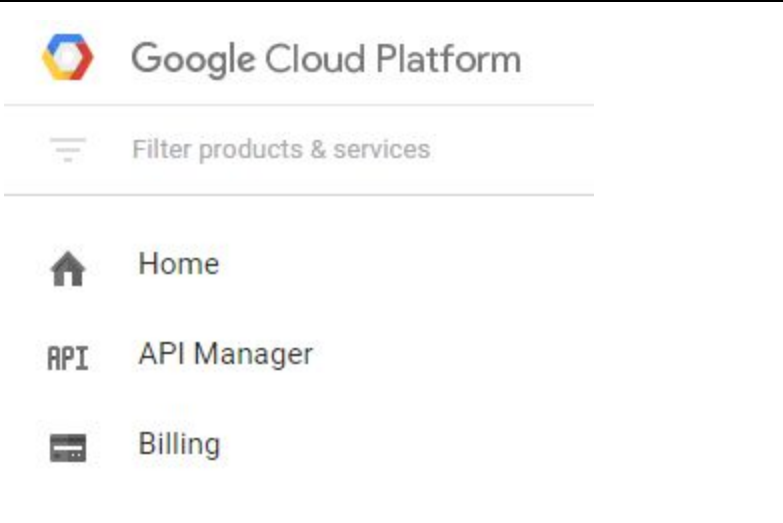
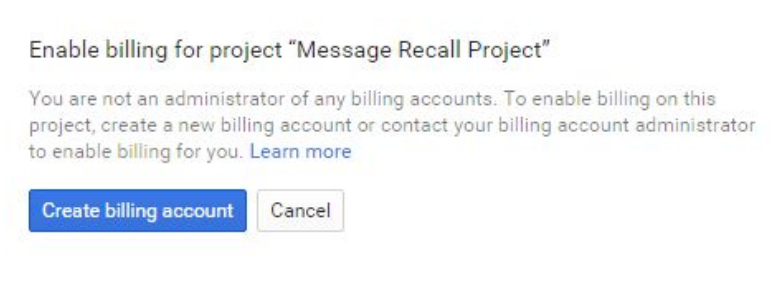
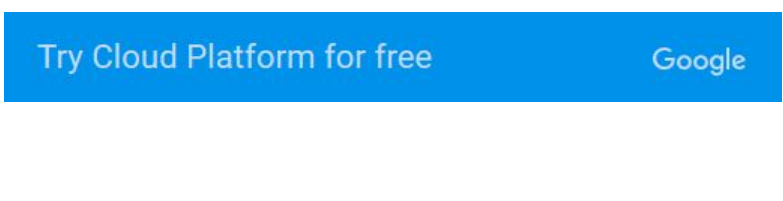



id  
id.

## 2.2 Enable billing for your application

This application uses the IMAP mail API which requires us to 'Enable Billing' in the application. Our use of this API is very small so any actual billing is very small.

However, Message Recall's use of backends is not free. It is **highly recommended** that you track your Billing Status on the application dashboard frequently. Running out of daily billing quota will cause your application to show a **OverQuotaError**.

Open the sidebar menu and choose 'Billing'.	 The screenshot shows the Google Cloud Platform interface. At the top is the Google Cloud Platform logo. Below it is a search bar labeled 'Filter products & services'. A sidebar menu is visible with three items: 'Home' with a house icon, 'API Manager' with an 'API' icon, and 'Billing' with a credit card icon. The 'Billing' item is highlighted with a blue background.
Choose 'Create billing account'.	 The screenshot shows a dialog box titled 'Enable billing for project "Message Recall Project"'. The text inside says: 'You are not an administrator of any billing accounts. To enable billing on this project, create a new billing account or contact your billing account administrator to enable billing for you. <a href="#">Learn more</a> '. At the bottom are two buttons: 'Create billing account' (blue) and 'Cancel' (white with a grey border).
Enter billing details (even though you won't be charged during your free trial).	 The screenshot shows a blue banner with the text 'Try Cloud Platform for free' on the left and the Google logo on the right.
Choose 'Accept and Start Free Trial'.	 The screenshot shows a blue button with the text 'Accept and start free trial'.

You can 'Tour the Console' or choose 'Got it'.



## Welcome RecallFirst!

Thanks for signing up for the 60-day free trial.

We've given you \$300 in free trial credit to spend. If you run out of credit, don't worry, you won't be billed until you give your permission.

[TOUR THE CONSOLE](#) [GOT IT](#)

You should see billing details.

Billing account ID: 00

### Credits

Promotion ID	Expires ^	Promotion value	Amount remaining
Free Trial	May 16, 2016	\$300.00	\$300.00

### Administrators

Billing administrators can add or delete other billing administrators, manage payment methods, view transactions, see credits, and set billing alerts.

#### Billing administrators

message\_recall\_role@



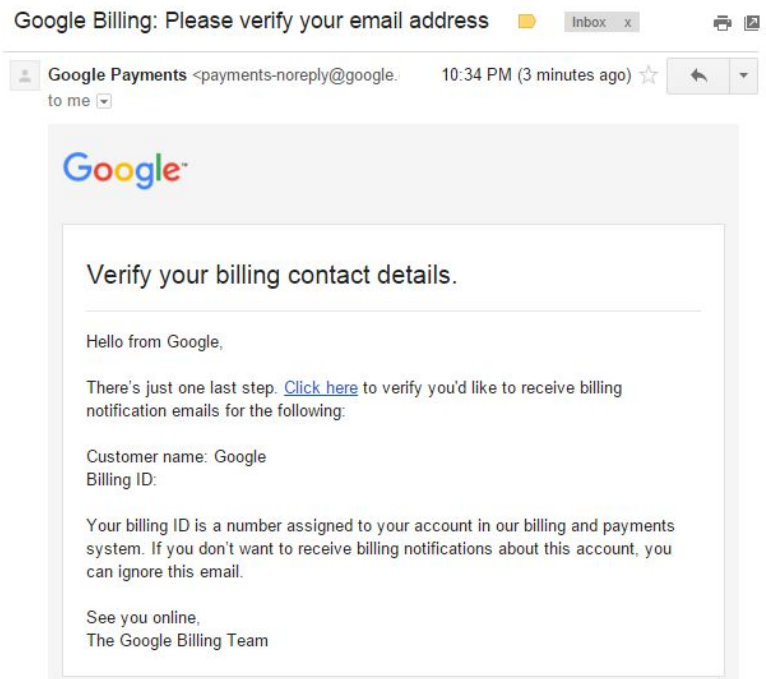
Enter an email address

### Projects linked to this billing account

Project name	Project ID
Message Recall Project	message-recall-project-1

You should also receive an email to confirm your billing contact details.

Make sure you choose 'Click here' in the email to confirm your billing details.



You should see a confirmation that your billing details are confirmed.

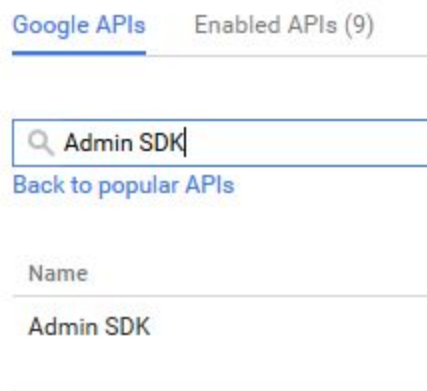
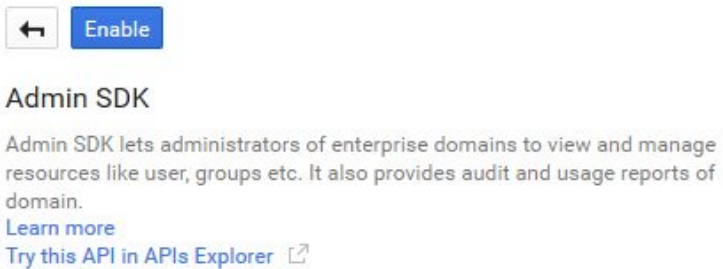
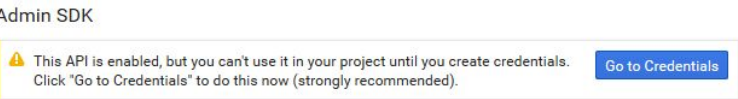
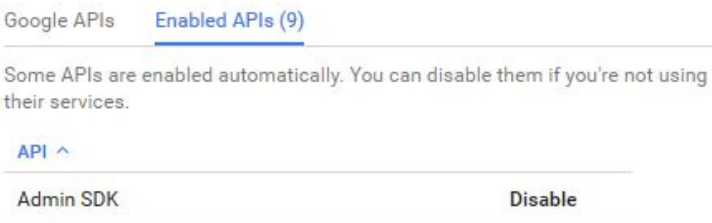


**Thanks! Your email address has been verified**

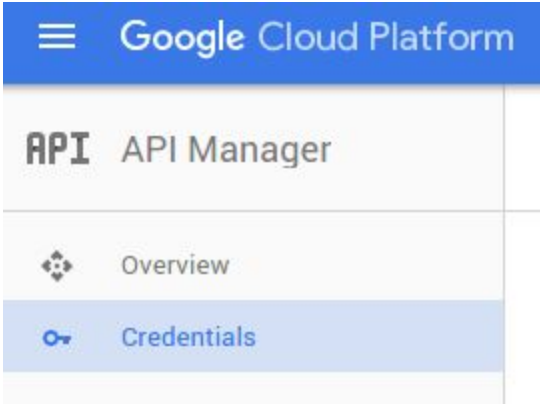
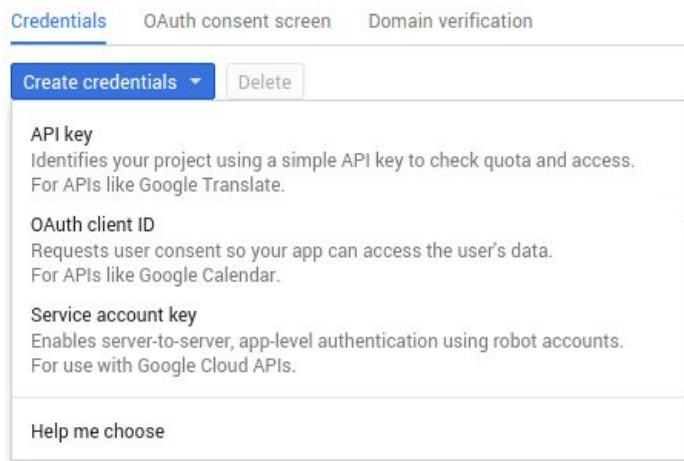
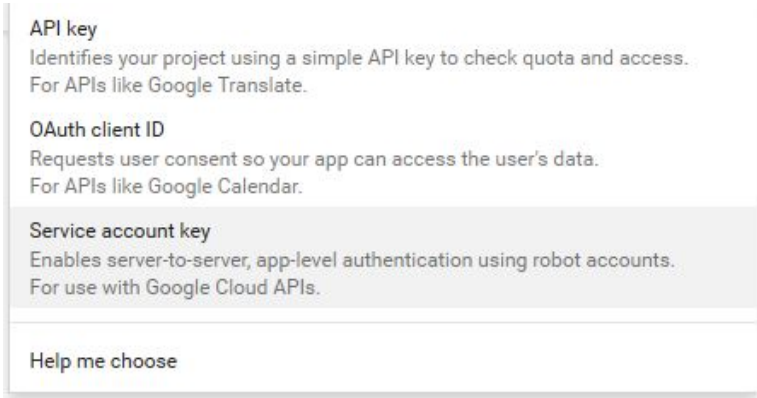
You may now start receiving billing emails for this customer.  
You can [unsubscribe](#) from these notifications at any time.

## 3. Configure the Google APIs Console Project

### 3.1 Enable the Admin SDK API





<p>In the Cloud Console open the sidebar menu and choose 'API Manager'.</p>	 <p>The screenshot shows the 'Google APIs' tab with 'Enabled APIs (9)' listed. A search bar contains 'Admin SDK'. Below the search bar is a link 'Back to popular APIs'. Under the 'Name' filter, 'Admin SDK' is listed.</p>
<p>Type 'Admin SDK' in the Search box.</p> <p>Choose 'Admin SDK' in the results.</p> <p>Choose 'Enable' to enable this API.</p>	 <p>The screenshot shows the 'Admin SDK' page with an 'Enable' button. Below the button, it says 'Admin SDK' and provides a description: 'Admin SDK lets administrators of enterprise domains to view and manage resources like user, groups etc. It also provides audit and usage reports of domain.' There are links for 'Learn more' and 'Try this API in APIs Explorer'.</p>
<p>The API is enabled, but we must create credentials now for our server-to-server interactions with APIs.</p> <p>We will do that next.</p>	 <p>The screenshot shows the 'Admin SDK' page with a warning message: 'This API is enabled, but you can't use it in your project until you create credentials. Click "Go to Credentials" to do this now (strongly recommended).' There is a 'Go to Credentials' button.</p>
<p>Navigate back to 'Overview' in the 'API Manager'.</p> <p>The 'Admin SDK' should now be listed in the 'Enabled APIs' list.</p>	 <p>The screenshot shows the 'Enabled APIs (9)' tab. It states: 'Some APIs are enabled automatically. You can disable them if you're not using their services.' Below this, there is a table with one entry: 'Admin SDK' with a 'Disable' button.</p>

### 3.2 Create Credentials (a Project Service Account) File

<p>Choose 'Credentials' under the 'API Manager' to create your new OAuth 2.0 API application client.</p>	
<p>Then, click the 'Create credentials' drop-down.</p>	
<p>Choose 'Service account key'.</p>	

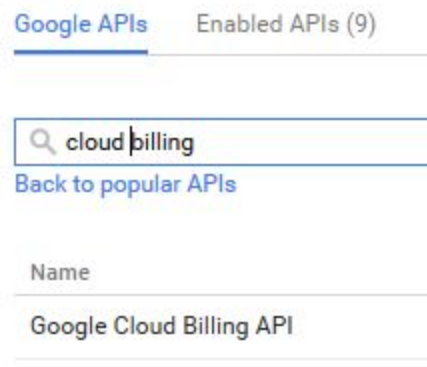
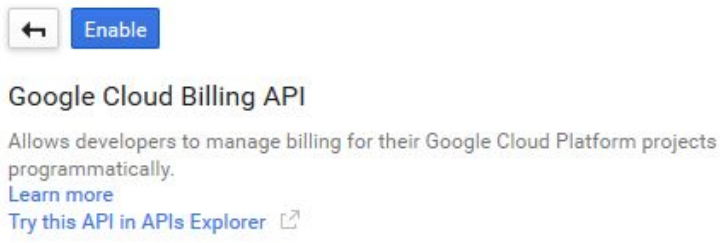
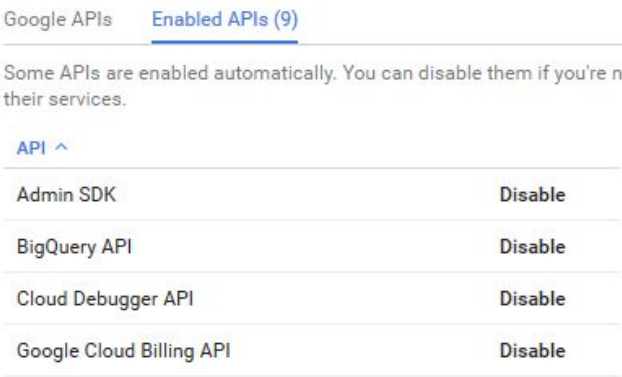
<p>Choose 'App Engine default service account'.</p>	<p>Create service account key</p> <p>Service account</p> <div><div>App Engine default service account</div><div>Compute Engine default service account</div><div>New service account</div></div> <p><input type="radio"/> JSON Recommended</p> <p><input checked="" type="radio"/> P12 For backward compatibility with code using the P12 format</p>						
<p>Choose 'P12' for backward compatibility.</p> <p>Click 'Create'.</p>	<p>Create service account</p> <p>Key type</p> <p>Downloads a file that contains the public/private key pair. It is the only copy of the key, so store it securely.</p> <p><input type="radio"/> JSON Recommended</p> <p><input checked="" type="radio"/> P12 For backward compatibility with code using the P12 format</p> <div><div>Create</div><div>Cancel</div></div>						
<p>A file is generated and downloaded to your local system with your private key material. The file has a name of the form 'Message Recall Project-X.p12'.</p> <p>You may need to click a 'Save' button.</p> <p>Choose 'Close' to proceed.</p>	<p>New public/private key pair</p> <p>Message Recall Project-965d8d5d8195.p12 has been saved on your computer. This is the only copy of the key, so store it securely.</p> <p>This is the private key's password. It will not be shown again. You must present this password to use the private key. <a href="#">Learn more</a></p> <div>notasecret</div> <div><div>Close</div></div>						
<p>Your OAuth 2.0 client information is now displayed in the Cloud Console.</p> <p>Click the 'Manage service accounts' to see these details.</p>	<p>Service account keys <a href="#">Manage service accounts</a></p> <table><thead><tr><th><input type="checkbox"/> ID</th><th>Creation date</th><th>Service account</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td>Mar 25, 2016</td><td>App Engine default service account</td></tr></tbody></table>	<input type="checkbox"/> ID	Creation date	Service account	<input type="checkbox"/>	Mar 25, 2016	App Engine default service account
<input type="checkbox"/> ID	Creation date	Service account					
<input type="checkbox"/>	Mar 25, 2016	App Engine default service account					
<p>Take special note of the 'Service account ID' (e.g. message-recall-xxxxx@appspot</p>	<div><div>Service account name</div><div><input checked="" type="checkbox"/> App Engine default service account</div></div> <div><div>Service account ID</div><div>message-recall-@appspot.gserviceaccount.com</div></div>						



<p>.gserviceaccount.com) it will be needed later.</p>							
<p>We need to allow this service account domain-wide access in the Google Apps domain.</p> <p>Under Options click the  and select 'Edit'.</p> <p>Check 'Enable Google Apps Domain-wide Delegation'.</p> <p>This creates a 'service account client' that will be used for API access.</p> <p>Click 'Save'.</p>	<div data-bbox="672 344 1403 911"> <h3>Edit service account</h3> <p>Service account name </p> <p>App Engine default service account</p> <p><input checked="" type="checkbox"/> Enable Google Apps Domain-wide Delegation  <small>Grants a client access to all users' data on a Google Apps domain without manual authorization on their part. <a href="#">Learn more</a></small></p> <div>  To change settings for Google Apps domain, product name for the OAuth consent screen must be configured. Assign the product name below or configure the OAuth consent screen. </div> <p>Product name for the consent screen</p> <p>Message Recall YourDomain</p> <p> <a href="#">Save</a> <a href="#">Configure consent screen</a> <a href="#">Cancel</a> </p> </div>						
<p>Take special note of the generated (numeric) 'Client ID' for new 'service account client'. It is used to register the service account in the Google Apps domain later.</p>	<div data-bbox="672 974 1403 1310"> <h3>Client ID for Service account client</h3> <div>  Service account clients are created when <a href="#">domain-wide delegation</a> is enabled on a service account. <a href="#">Manage service accounts</a> </div> <table> <tr> <td>Client ID</td> <td>#####</td> </tr> <tr> <td>Service account</td> <td>App Engine default service account message-recall-@appspot.gserviceaccount.com</td> </tr> <tr> <td>Creation date</td> <td>Apr 8, 2016, 10:20:52 AM</td> </tr> </table> <p>Name</p> <p>Client for message-recall-gapslabs</p> </div>	Client ID	#####	Service account	App Engine default service account message-recall-@appspot.gserviceaccount.com	Creation date	Apr 8, 2016, 10:20:52 AM
Client ID	#####						
Service account	App Engine default service account message-recall-@appspot.gserviceaccount.com						
Creation date	Apr 8, 2016, 10:20:52 AM						



### 3.3 Enable the Google Cloud Billing API

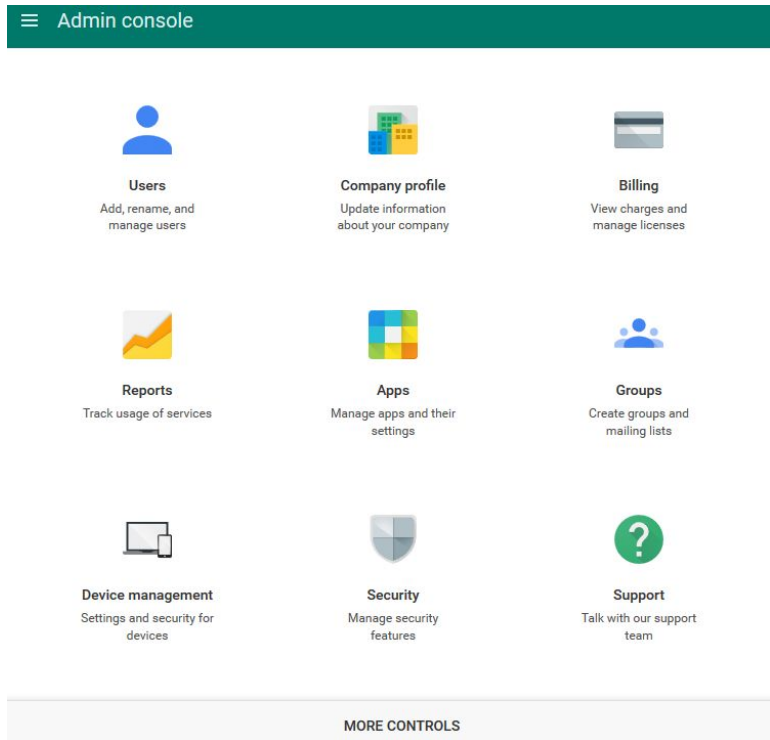
<p>Return to the Google APIs view within the 'API Manager' of the Cloud Console.</p>	
<p>Type 'Cloud Billing' in the Search box.</p> <p>Choose 'Google Cloud Billing API' in the results.</p> <p>Choose 'Enable' to enable this API.</p>	
<p>Navigate back to 'Overview' in the 'API Manager'.</p> <p>The 'Google Cloud Billing API' should now be listed in the 'Enabled APIs' list.</p>	

## 4. Configure your Google Apps Domain for the App Engine application

### 4.1 Add the App Engine service-account to the domain security model

Navigate back to the 'Admin console' for your domain at <https://admin.google.com>.

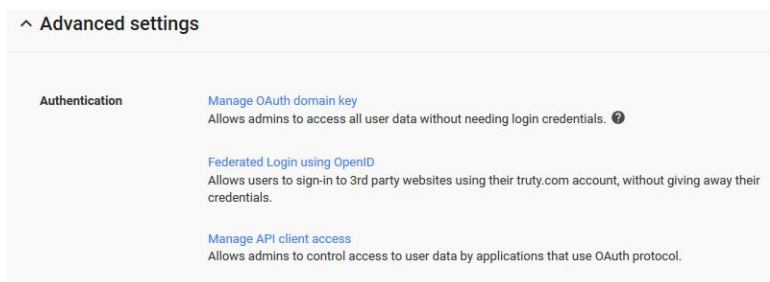
Select 'Security'.



In Security, select 'Show more'.

Select 'Advanced settings'.




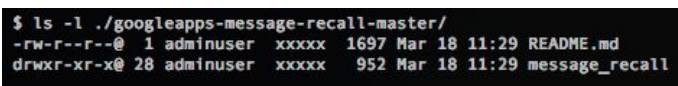
Select 'Manage API client access'.



<p>In the 'Client Name' field, enter the Service Account client 'Client ID' from the very end of step 3.2 earlier.</p> <p>It should be of the form: ##### (only numbers).</p>	<p><b>Authorized API clients</b></p> <div data-bbox="678 285 1062 453"> <p>Client Name</p> <p>#####</p> <p>Example: www.example.com</p> </div>
<p>In the 'One or More API Scopes' box, enter the following scopes string:</p> <p><a href="https://www.googleapis.com/auth/cloud-billing.readonly">https://www.googleapis.com/auth/cloud-billing.readonly</a>,<a href="https://mail.google.com/">https://mail.google.com/</a>,<a href="https://www.googleapis.com/auth/admin.directory.user.readonly">https://www.googleapis.com/auth/admin.directory.user.readonly</a></p> <p>Choose 'Authorize'</p>	<div data-bbox="678 604 1419 743"> <p>One or More API Scopes</p> <p><a href="https://www.googleapis.com/auth/cloud-billing.readonly">https://www.googleapis.com/auth/cloud-billing.readonly</a> <input type="button" value="Authorize"/></p> <p>Example: <a href="http://www.google.com/calendar/feeds/">http://www.google.com/calendar/feeds/</a> (comma-delimited)</p> </div>
<p>You should notice an entry for your service account client now listed with enabled scopes.</p>	<p>Email (Read/Write/Send) <a href="https://mail.google.com/">https://mail.google.com/</a>  View users on your domain <a href="https://www.googleapis.com/auth/admin.directory.user.readonly">https://www.googleapis.com/auth/admin.directory.user.readonly</a>  <a href="https://www.googleapis.com/auth/cloud-billing.readonly">https://www.googleapis.com/auth/cloud-billing.readonly</a></p>

## 5. Prepare the source code

### 5.1 Get the source code

Browse to the github.com page for <a href="#">googleapps-message-recall</a> .	
Click the 'Download ZIP' button.	
Create an empty folder for the project source code.  Copy the downloaded googleapps-message-recall-master.zip file to the empty folder.  Unzip the project source code.	
Notice the source files under a new folder: googleapps-message-recall-master/ message_recall.	

### 5.2 Make local updates

In the extracted source, edit the app.yaml file.  Change the 'application' from 'message_recall' to your app engine 'Application Identifier'.  Save your new app.yaml file.	<b>From:</b> application: message-recall  <b>To:</b> application: message-recall-gapplabs
In the extracted source, edit the service_account.py file.  Change the SERVICE_ACCOUNT_NAME to reflect the 'Service account ID' discovered in step 3.3.	<b>From:</b> SERVICE_ACCOUNT_NAME = ( '000000000000-xx' '@developer.gserviceaccount.com' )  <b>To:</b> <your Service account ID (Email) address>

### 5.3 Create a service account certificate .pem file

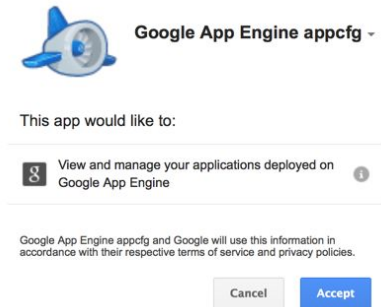
Find the file downloaded 'XXXX-privatekey.p12' file and copy it to your application folder.	googleapps-message-recall-master/message_recall/
Use the openssl tool to convert the PKCS12 certificate file to PEM format.  Note: openssl is available on Linux and <a href="#">Windows</a>  Note the output filename of 'messagerecall_privatekey.pem' is required.	<pre>\$ openssl pkcs12 -in xxxx-privatekey.p12 -out messagerecall_privatekey.pem -nodes -nocerts</pre>
When prompted for a password, supply the one presented earlier: <i>notasecret</i> .	Enter the password: notasecret
You should see the following output response and a new file created: messagerecall_privatekey.pem.	MAC verified OK
You can now remove the privatekey.p12 file.	<pre>\$ rm xxxx-privatekey.p12</pre>
Edit the new messagerecall_privatekey.pem file.  Delete the Bag Attributes lines.  Delete the Key Attributes line.  The .pem file should begin with the 'BEGIN PRIVATE KEY' line and end with the 'END PRIVATE KEY' line.	<pre>Bag Attributes     friendlyName: privatekey     localKeyID: 54 69 6D 65 20 31 33 38 31 37 37 36 30 34 33 31 37 34 Key Attributes: &lt;No Attributes&gt;  -----BEGIN PRIVATE KEY----- MIICdwIB..... -----END PRIVATE KEY-----</pre>
The edited file must be placed in the root source folder next to the app.yaml file.	googleapps-message-recall-master/message_recall/

## 6. Download the Google App Engine SDK

Find your platform, download and install the 'Google App Engine SDK for Python'	<a href="https://developers.google.com/appengine/downloads#Google_App_Engine_SDK_for_Python">https://developers.google.com/appengine/downloads#Google_App_Engine_SDK_for_Python</a>
Your installation process may be a single unzip or an executable installation program.	

---

## 7. Upload the source code into the AppEngine host

<p>Deploy the front-end code.</p> <p>From your application source root folder (the folder that contains app.yaml), run the appcfg.py command that was installed with the 'Google App Engine SDK for Python'.</p> <p>Authenticate as your <a href="mailto:message_recall_role@yourdomain.com">message_recall_role@yourdomain.com</a> user.</p>	<pre>\$ appcfg.py --noauth_local_webserver update .</pre>
<p>When prompted, choose 'Accept' to allow the appcfg program to upload the source code to your app engine hosting environment.</p>	 <p>The dialog box shows the Google App Engine logo and the title 'Google App Engine appcfg'. It asks 'This app would like to:' and lists 'View and manage your applications deployed on Google App Engine'. At the bottom, there are 'Cancel' and 'Accept' buttons. A small disclaimer at the bottom states: 'Google App Engine appcfg and Google will use this information in accordance with their respective terms of service and privacy policies.'</p>
<p>You should notice the following progress on your console confirming the initial application upload (deployment).</p>	<pre>Compilation starting. Compilation completed. Starting deployment. Checking if deployment succeeded. Deployment successful. Checking if updated app version is serving. Completed update of app:     message-recall-gappslabs, version: 1 Uploading index definitions. Uploading task queue entries.</pre>
<p>Now deploy the back-end code.</p>	<pre>\$ appcfg.py backends . update</pre>
<p>You should notice the following progress on your console confirming the deployment.</p>	<pre>Starting update of app: message-recall-gappslabs, backend: recall-backend Getting current resource limits. Scanning files on local disk. ... Compilation starting. Compilation completed. Starting deployment. Checking if deployment succeeded. Deployment successful.</pre>

	Checking if updated app version is serving. Completed update of app: message-recall-gapplabs, backend: recall-backend
--	--

---



## 8. Test the running application

### A. Navigate to the application and view the landing page:

<https://message-recall-gapplabs.appspot.com>



### B. Navigate to the History page:

NOTE: The initial application data store indices need to be constructed. If you see the following message on the History page, initial index construction has not yet completed.

You can also check progress on initial index creation from the AppEngine administrator page under 'Datastore Indexes'.



### C. Try recalling a message:



## Message Recall - Recall Message

[Home](#)[Recall Message](#)[History](#)[About](#)[Sign out](#)

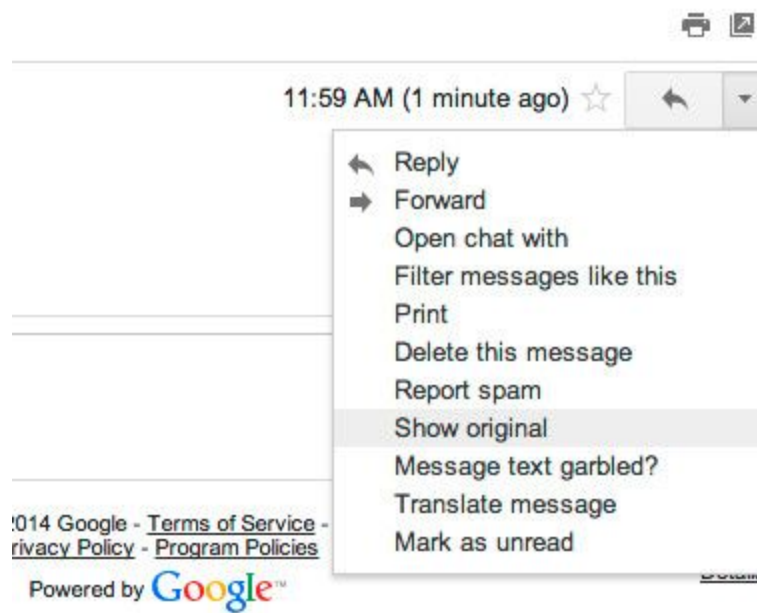
Use this tool to remove an Email message from your domain-users mailboxes. This is useful in critical situations when an email message has been inadvertently sent to your domain users.

Supply the Message-ID and all Google Apps domain users will be checked.

The email will be deleted from any active user that has received it.

Enter a Message-ID for the email to purge.

1. From Gmail, send a message to any domain user.
2. From your Gmail 'Sent Mail' folder, open the sent message.
3. You can see detailed information about your message by clicking the 'Show original'.



4. Note the **Message-ID** of your sent message (it is the text inside of the <> brackets including '@mail.gmail.com').
  5. Navigate to the 'Recall Messages' page of your application.
  6. Enter the Message-ID of your sent message in the textbox.
  7. Click 'Submit'.
  8. You are dropped into the Message Recall 'Task' page. Refresh it frequently to see status of your recall task.
  9. You may choose the 'View Report' page to see a different view of progress.
  10. The Task state starts at '*Getting Users*' and progresses through '*Recalling*' to '*Done*'.
- 
11. When the application completes, you should notice the following on the 'Task' page if the message was only sent to 1 user. This indicates the user was identified, is not suspended, and the message was able to be deleted.



## Message Recall - Task

[Home](#)[Recall Message](#)[History](#)[About](#)[Sign out](#)

Tip: Refresh frequently to see updates.

State	
Owner	
Message-ID	
Start (UTC)	20151024 10:20:49
Stop (UTC)	20151024 10:21:06
Elapsed (m:s)	
Total Users	
Users Processed	
Users with Messages Recalled	1

[View Users](#)[View Report](#)[Debug Task](#)

## Common Problems

The following problems arise if one of the setup steps is missed or incomplete.

Problem	Correction
"unauthorized_client" error.	Add the App Engine service account to your Google Apps domain security with scopes.
"You are not an authorized user of this application."	This occurs if you have not enabled the 'Admin SDK' API properly.
"This AppEngine application requires billing status: "Billing Enabled".	Walk through the Billing pages in the Cloud Console to ensure your project has a Billing Account ID that is enabled.

# Appendices

## Getting Help

If you need additional assistance setting up and/or configuring your application, please try the following resources:

<a href="#">Help with Google App Engine</a>
<a href="#">The Google App Engine SDK for Python</a>
<a href="#">Google Apps Documentation &amp; Support: Sign in to your Admin console</a>
<a href="#">Google APIs Console Help</a>
<a href="#">The Google Apps Admin SDK</a>
<a href="#">Service Account Information</a>
<a href="#">The googleapps-message-recall Google Group</a>