

[Solutions](https://cloud.google.com/solutions/) (https://cloud.google.com/solutions/) [Solutions](#)

Invoking Legacy Code

You want to migrate your app to App Engine, but there's a problem: the server-side code is tied to a specific operating system and can't easily be ported. Perhaps you're using an aging COM object written in C++ or Visual Basic; or, worse, your source code no longer exists, and only the executable binary remains.

This tutorial describes an architectural pattern for invoking legacy code using App Engine, Compute Engine, and Pub/Sub, and provides an example implementation of the pattern. Though the example implementation focuses on Windows-specific code, you can reuse the pattern outlined here for any legacy code.

Objectives

- Learn about a simple architectural pattern for invoking legacy code on Google Cloud.
- Download and deploy an example implementation of the architectural pattern.

An architectural pattern for invoking legacy code

The following diagram illustrates an architectural pattern you can use to invoke legacy code from your App Engine app.

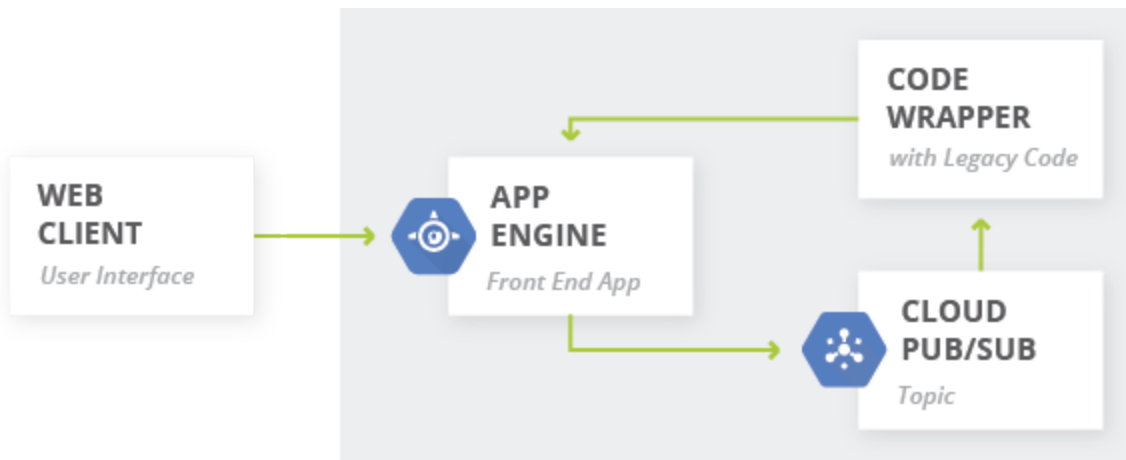


Figure 1: Architectural pattern for invoking legacy code

This architecture works as follows:

1. The web client sends a request to the App Engine app, which writes the request to a Pub/Sub topic.
2. A code wrapper pulls the request from the topic.
3. The code wrapper invokes the legacy code.
4. The legacy code returns the result to the wrapper. The wrapper then returns the result to the App Engine app, which returns it to the web client.

An example implementation of the pattern

You can find an [implementation](https://github.com/GoogleCloudPlatform/pubsub-shout-csharp) (<https://github.com/GoogleCloudPlatform/pubsub-shout-csharp>) of the above pattern at GitHub. This example app takes a string as input, processes the string using the components outlined in the pattern, and returns an all-uppercase version. The app code comprises two main components:

- The *App Engine app*, which publishes incoming requests to a Pub/Sub topic.
- The *code wrapper*, which subscribes to the Pub/Sub topic and, when it receives a request, sends the request to some legacy Windows-based code.

In this implementation, the wrapper is a Windows service, written in C#. This wrapper service is configured to run when the instance starts up, and can invoke any script or binary that can be invoked from C#. Both the wrapper service and the invoked binary reside on a Compute Engine instance running Windows Server 2012 R2.

The app's user interface looks similar to this:

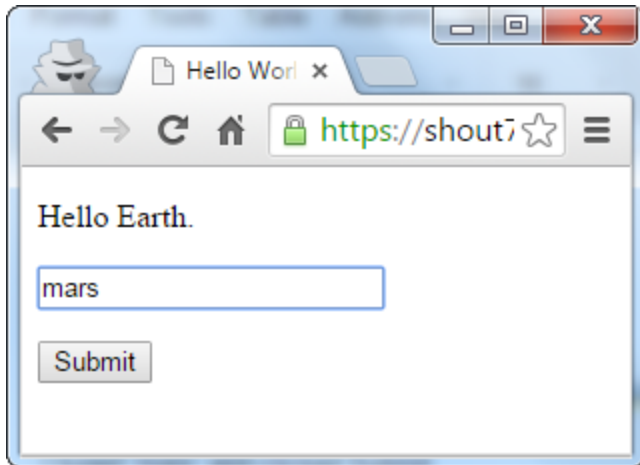


Figure 2: Shout app user interface

If you type the word "mars" and click **Submit**, the result is as follows:

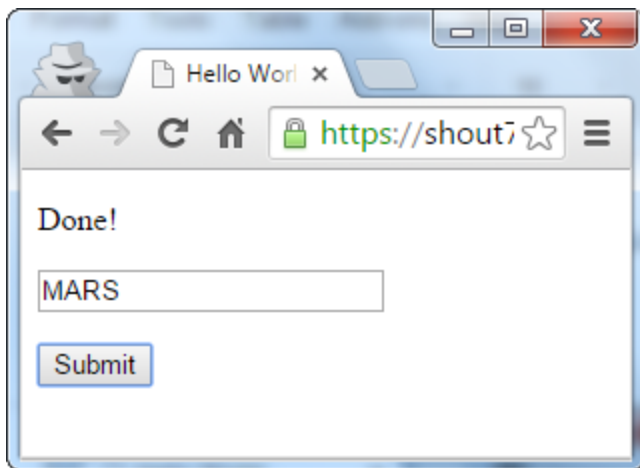


Figure 3: Shout app result

This tutorial uses the following billable components of Google Cloud:

- A Compute Engine instance with a Windows Server 2012 license
- Pub/Sub services

To generate a cost estimate based on your projected usage, use the [pricing calculator](https://cloud.google.com/products/calculator) (<https://cloud.google.com/products/calculator>). New Google Cloud users might be eligible for a [free trial](https://cloud.google.com/free-trial) (<https://cloud.google.com/free-trial>).

Prerequisites

This example app assumes the following:

- You have a Windows development machine with Virtual Studio 2012 (or later) installed.
- You have an RDP client installed on your development machine.
- You've created a Google Cloud Console project (<https://console.cloud.google.com/project>).
- You've installed the App Engine Python SDK (<https://cloud.google.com/appengine/docs/standard/python/download>).
- You've installed Python 2.7 (<https://www.python.org/downloads/>) and added the folder containing the python binary to your PATH environment variable.
- You've installed pip 1.4 or later (<https://pypi.python.org/pypi/pip>), and added the folder containing the pip binary to your PATH environment variable.

Note: Python 2.7.9 and subsequent Python releases include **pip** by default.

Get the source code

Download the example code here

(<https://github.com/GoogleCloudPlatform/pubsub-shout-csharp/archive/master.zip>) and unzip it. If you prefer, you can clone the GitHub project repository instead by running the following command:

```
C:\> git clone https://github.com/GoogleCloudPlatform/pubsub-shout-csharp.git
```

Install the dependencies

In your command-line terminal, navigate to the `appengine-python-flask` folder in your local copy of the `pubsub-shout-csharp` repository, and then run the following:

```
C:\> pip install -r requirements.txt -t lib
```

Create a Compute Engine instance

Create the Compute Engine instance that the code wrapper and legacy code will run on:

1. In the Cloud Console, go to the Create a new instance page (<https://console.cloud.google.com/compute/instancesAdd>).

2. In the **Boot disk** section, click **Change**.
3. From the list of preconfigured images, select **Windows Server 2012 R2 Datacenter Edition**, and then click **Select**.
4. In the **Firewalls** section, allow HTTP and HTTPS traffic.
5. In the **Project access** section, allow full API access to all Google Cloud services in your project.
6. To reveal advanced options, click **Management, security, disks, networking, sole tenancy**, and then click the **Networking** tab.
7. In the **Network tags** section, add the tag `rdp-tag`. Later in the tutorial, you'll create a firewall rule that allows external RDP access to all instances tagged with this tag.
8. To begin creating your new instance, click **Create**.

Note: Windows instances can take up to eight minutes to deploy fully.

After you create the instance, add a default Windows user to the instance:

1. In the Cloud Console, go to the [VM instances page](https://console.cloud.google.com/compute/instances) (<https://console.cloud.google.com/compute/instances>).
2. Click the name of your newly deployed instance.
3. On the instance page, click the **Set Windows password** button.
4. To create the user account on your instance, in the **Set new Windows password** dialog, add your username and click **Set**.
5. Make a note of the provided password and close the dialog.

Next, create a firewall rule to enable RDP access to your instance:

1. In the Cloud Console, go to the [Create a firewall rule page](https://console.cloud.google.com/networking/firewalls/add) (<https://console.cloud.google.com/networking/firewalls/add>).
2. Fill out the form as follows:
 - **Name:** `rdp-rule`
 - **Source IP ranges:** `0.0.0.0/0`
 - **Allowed protocols and ports:** `tcp:3389`
 - **Target tags:** `rdp-tag`

3. To create your firewall rule, click **Create** .

Build and deploy the code wrapper

To build the code wrapper service:

1. In the Cloud Console, go to the [Credentials page](https://console.cloud.google.com/apiui/credential) (<https://console.cloud.google.com/apiui/credential>).
2. Click **New credentials** and select **Service account key**.
3. On the **Create service account key** page, fill out the form as follows:
 - *Service account*: New service account
 - *Name*: A name of your choosing.
 - *Service account ID*: An ID of your choosing.
 - *Key type*: JSON
4. Click **Create**. After a few seconds, a JSON key file downloads to your local computer.
5. On your local machine, [create a new environment variable](https://technet.microsoft.com/en-us/library/cc772047.aspx) (<https://technet.microsoft.com/en-us/library/cc772047.aspx>) called `GOOGLE_APPLICATION_CREDENTIALS` and set it to the absolute path of the JSON key file.
6. Go to the `windows-csharp` folder in your local copy of the `pubsub-shout-csharp` repository.
7. Open `ShoutService.sln` in Visual Studio 2012 or later.
8. Edit `ShoutLib/Constants.cs`, set the variable `ProjectId` to your <https://console.cloud.google.com/> project ID, and save.
9. In the **Build** menu, click **Build Solution** to build the app.

After building the code wrapper service, copy it to your Compute Engine instance:

1. Establish a remote desktop connection to the instance you created earlier in the tutorial.
2. On your local machine, in Windows Explorer, go to the `ShoutService\bin` folder under your `windows-csharp` folder.
3. Copy the `ReLease` folder from Windows Explorer, and then paste it inside your remote connection window to copy the folder to your instance. The `ReLease` folder contains the wrapper and the Shout service.

Next, allow the Windows Service Control Manager to start, stop, and otherwise interact with your service by granting appropriate permissions to **LOCAL SERVICE**, a predefined Windows user account used by the Service Control Manager:

1. In Windows Explorer, go to the `Release` folder.
2. Right-click the folder to open the context menu, and then click **Properties**.
3. In the **Release Properties** dialog, click the **Security** tab.
4. Click **Edit**.
5. In the **Permissions for Release** dialog, click **Add**.
6. In the **Select Users or Groups** dialog, under **Enter the object names to select**, type `LOCAL SERVICE` and then click **OK**.
7. In the **Permissions for Release** dialog, click **LOCAL SERVICE**.
8. Under **Permissions for LOCAL SERVICE**, make sure that the following permissions are enabled:
 - Read & execute
 - List folder contents
 - Read
9. Click **OK**.

Finally, install the code wrapper service:

1. On your remote machine, open the command-line terminal as Administrator and navigate to the `Release` folder.
2. Install the code wrapper service:

```
C:\> C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe ShoutServi
```

Deploy the App Engine app

To deploy the App Engine app:

1. On your local machine, in your command-line terminal, navigate to the `appengine-python-flask` folder in your local copy of the `pubsub-shout-csharp` repository.
2. Deploy the app.

```
C:\> appcfg.py -A <your_project_id> -V 1 --oauth2 update .
```

3. Authenticate the app when prompted.

4. Enable the Pub/Sub API.

ENABLE THE API ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/FLOWS/ENABLEAPI?APIID=PUBSUB](https://console.cloud.google.com/flows/enableapi?apiid=pubsub))

5. To initialize the app, visit the following URL in your web browser:

https://<your_project_id>.appspot.com/init

When you visit this URL, the App Engine app creates a Pub/Sub topic and then subscribes the code wrapper service to the topic.

Note: Visit this page only once. If all is well, you will see an "ok" message. If you refresh the page or return to it later, you'll see errors, as the resources it creates will already exist.

Congratulations! Your app is now live. Visit [.<your_project_id>.appspot.com](https://<your_project_id>.appspot.com) to see it in action.

Cleaning up

After you've finished the Invoking Legacy Code tutorial, you can clean up the resources that you created on GCP so they won't take up quota and you won't be billed for them in the future. The following sections describe how to delete or turn off these resources.

Delete your Google Cloud project

The easiest way to eliminate billing is to delete the project that you created for the tutorial.

To delete the project:

Caution: Deleting a project has the following effects:


- **Everything in the project is deleted.** If you used an existing project for this tutorial, when you delete it, you also delete any other work you've done in the project.
- **Custom project IDs are lost.** When you created this project, you might have created a custom project ID that you want to use in the future. To preserve the URLs that use the project ID, such

as an **appspot.com** URL, delete selected resources inside the project instead of deleting the whole project.

If you plan to explore multiple tutorials and quickstarts, reusing projects can help you avoid exceeding project quota limits.

1. In the Cloud Console, go to the **Manage resources** page.

[GO TO THE MANAGE RESOURCES PAGE](https://console.cloud.google.com/iam-admin/projects) (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/IAM-ADMIN/PRO


2. In the project list, select the project you want to delete and click **Delete** .
3. In the dialog, type the project ID, and then click **Shut down** to delete the project.

Delete your Compute Engine instance

To delete a Compute Engine instance:

1. In the Cloud Console, go to the **VM Instances** page.

[GO TO THE VM INSTANCES PAGE](https://console.cloud.google.com/compute/instances) (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/COMPUTE/INSTANCES

2. Click the checkbox for the instance you want to delete.
3. Click **Delete**  to delete the instance.

What's next

- Try out other Google Cloud Platform features for yourself. Have a look at our [tutorials](https://cloud.google.com/docs/tutorials) (<https://cloud.google.com/docs/tutorials>).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 19, 2019.