

[Python](https://cloud.google.com/python/) (<https://cloud.google.com/python/>) [Guides](#)

Deploying the Python Bookshelf app using Cloud Deployment Manager

This tutorial shows how to deploy the Python Bookshelf sample app using [Cloud Deployment Manager](https://cloud.google.com/deployment-manager/docs/) (<https://cloud.google.com/deployment-manager/docs/>).

Cloud Deployment Manager lets you create the necessary Google Cloud [resources](https://cloud.google.com/deployment-manager/latest/resources) (<https://cloud.google.com/deployment-manager/latest/resources>) for a deployment in a single step, through a declarative, repeatable process. With Cloud Deployment Manager, you can update your deployments, track your modifications over time, create templates using [Jinja](http://jinja.pocoo.org/) (<http://jinja.pocoo.org/>) or Python, and parameterize your templates, so that similar deployments share a template.

Objectives

- Clone and configure the Bookshelf sample app.
- Create Cloud Deployment Manager configurations and templates.
- Create Cloud Deployment Manager deployments.
- Deploy the Bookshelf sample app using Cloud Deployment Manager.

Costs

This tutorial uses the following billable components of Google Cloud:

- [Compute Engine](https://cloud.google.com/compute/pricing) (<https://cloud.google.com/compute/pricing>)

- [Cloud Storage](https://cloud.google.com/storage/pricing) (https://cloud.google.com/storage/pricing)
- [Datastore](https://cloud.google.com/datastore/pricing) (https://cloud.google.com/datastore/pricing)
- [Stackdriver Logging](https://cloud.google.com/stackdriver/pricing) (https://cloud.google.com/stackdriver/pricing)
- [Pub/Sub](https://cloud.google.com/pubsub/pricing) (https://cloud.google.com/pubsub/pricing)

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).

When you finish this tutorial, you can avoid continued billing by deleting the resources you created. For more information, see [Cleaning up](#) (#clean-up).

Before you begin

1. [Sign in](https://accounts.google.com/Login) (https://accounts.google.com/Login) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp) (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Google Cloud project.

Note: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

[GO TO THE PROJECT SELECTOR PAGE](https://console.cloud.google.com/projectselector) (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECTOR)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](https://cloud.google.com/billing/docs/how-to/modify-project) (https://cloud.google.com/billing/docs/how-to/modify-project).
4. Enable the Datastore API, Cloud Storage API, Pub/Sub API APIs.

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

5. [Install and initialize the Cloud SDK](https://cloud.google.com/sdk/docs/) (https://cloud.google.com/sdk/docs/).
6. Install Python, `pip`, and `virtualenv` on your system. For instructions, see [Setting up a Python development environment](https://cloud.google.com/python/setup) (https://cloud.google.com/python/setup) for Google Cloud.

Creating a Cloud Storage bucket

The following instructions detail how to create a Cloud Storage bucket. Buckets are the basic containers that hold your data in Cloud Storage.

Note: You can choose [any name](https://cloud.google.com/storage/docs/naming) (<https://cloud.google.com/storage/docs/naming>) for your Cloud Storage bucket. It's a good practice to give your bucket the same name as your project ID. Bucket names must be unique across all of Google Cloud, so it's possible that you can't use your project ID as the bucket name.

1. In your terminal window, create a Cloud Storage bucket, where ***YOUR_BUCKET_NAME*** represents the name of your bucket:

```
gsutil mb gs://YOUR_BUCKET_NAME
```

2. To view uploaded images in the Bookshelf app, set the bucket's default access control list (ACL) to `public-read`:

```
gsutil defacl set public-read gs://YOUR_BUCKET_NAME
```

Cloning the sample app

The sample app is available on GitHub at [GoogleCloudPlatform/getting-started-python](https://github.com/GoogleCloudPlatform/getting-started-python) (<https://github.com/GoogleCloudPlatform/getting-started-python>).

1. Clone the repository:

```
git clone https://github.com/GoogleCloudPlatform/getting-started-python.git -b
```

2. Go to the sample directory:

```
cd getting-started-python/7-gce
```

Configuring the app

1. Open `config.py` for editing.

- Set the value of `PROJECT_ID` to your project ID.
- Set the value `CLOUD_STORAGE_BUCKET` to the name of your Cloud Storage bucket.

2. Save and close `config.py`.

Running the app on your local computer

1. Create an isolated Python environment, and install dependencies:

```
LINUX/MACOS  WINDOWS
virtualenv -p python3 env
source env/bin/activate
pip install -r requirements.txt
```

2. Run the application:

```
python main.py
```

3. In your browser, enter the following address:

<http://localhost:8080> (`http://localhost:8080`)

To stop the local web server, press `Control+C`. If you want to exit the virtual environment, enter `deactivate`.

Deploying the sample app

Push your code to a repository

You can get your code onto a running Compute Engine instance in several ways. One way is to use [Cloud Source Repositories](https://cloud.google.com/source-repositories/) (`https://cloud.google.com/source-repositories/`). Every project includes a Git repository that is available to Compute Engine instances. Your instances then pull the latest version of your app's code during startup. Using a Git repository is convenient because updating your app doesn't require configuring new images or instances; just restart an existing instance or create one.

1. For your first time using Git, use `git config --global` to set up your identity.
(https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup#_your_identity).
2. In your Cloud Console, create a repository:

CREATE REPOSITORY ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/CODE/DEVELOP/REPO](https://console.cloud.google.com/code/develop/repo))

3. Then push your app code to your project's repository where `[YOUR_PROJECT_ID]` is your project ID and `[YOUR_REPO]` is the name of your repository:

```
git commit -am "Updating configuration"  
git config credential.helper gcloud.sh  
git remote add cloud https://source.developers.google.com/p/[YOUR_PROJECT_ID]/r  
git push cloud master
```

Create the deployment

After your configuration is committed and your code is uploaded to Cloud Source Repositories, you can use the Cloud Deployment Manager to create the deployment:

1. Go to the `deployment_manager` directory:

```
cd getting-started-python/7-gce/gce/deployment_manager
```

2. Create the deployment:

```
gcloud deployment-manager deployments create my-deployment --config config.yaml
```

3. View a list of all the deployments:

```
gcloud deployment-manager deployments list
```

4. Get a description of the deployment and the resources it created:

```
gcloud deployment-manager deployments describe my-deployment
```

View your app

After you create the forwarding rule, it can take several minutes for your configuration to propagate and for traffic to be routed to your instances.

1. To check the progress:

```
gcloud compute backend-services get-health bookshelf-my-deployment-frontend
```

2. When at least one of your instances reports **HEALTHY**, get the forwarding IP address for the load balancer:

```
gcloud compute forwarding-rules list --global
```

Your forwarding-rules IP address is in the **IP_ADDRESS** column.

3. In your browser, enter the IP address from the list.

Your app is now running on Google Cloud.

Understanding the code

Configure the deployment

You can see how the deployment is configured in **config.yaml**:

[7-gce/gce/deployment_manager/config.yaml](https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/config.yaml)

(https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/config.yaml)

TFORM/GETTING-STARTED-PYTHON/BLOB/STEPS/7-GCE/GCE/DEPLOYMENT_MANAGER/CONFIG.YAML)

```
imports:
- name: bookshelf.jinja
  path: ./bookshelf.jinja

resources:
- name: bookshelf
  type: bookshelf.jinja
  properties:
    zone: us-central1-f
    machine-type: n1-standard-1
    machine-image: https://www.googleapis.com/compute/v1/projects/debian-cloud/global
    min-instances: 1
```

```
max-instances: 10
target-utilization: 0.6
scopes:
- https://www.googleapis.com/auth/cloud-platform
```

The configuration file imports the template file as a resource and uses it to define a resource named `bookshelf`. The template takes several properties as parameters. For more information about creating Cloud Deployment Manager configuration files, see [Creating a configuration](https://cloud.google.com/deployment-manager/docs/step-by-step-guide/create-a-configuration) (<https://cloud.google.com/deployment-manager/docs/step-by-step-guide/create-a-configuration>).

Configure the deployment template's schema

Look at the schema file `bookshelf.jinja.schema`, which defines and documents the parameters that the deployment is expected to provide in its configuration file. For more information about configuration schemas, see [Using schemas](https://cloud.google.com/deployment-manager/docs/configuration/templates/using-schemas) (<https://cloud.google.com/deployment-manager/docs/configuration/templates/using-schemas>).

[7-gce/gce/deployment_manager/bookshelf.jinja.schema](https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja.schema)

(https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja.schema)

NG-STARTED-PYTHON/BLOB/STEPS/7-GCE/GCE/DEPLOYMENT_MANAGER/BOOKSHELF.JINJA.SCHEMA)

```
info:
  title: Bookshelf GCE Deploy
  author: Google Inc.
  description: Creates a GCE Deployment

imports:
- name: startup-script
  path: ../startup-script.sh

required:
- zone
- machine-type
- min-instances
- max-instances
- scopes

properties:
  zone:
```

```
description: Zone to create the resources in.
type: string
machine-type:
description: Type of machine to use
type: string
machine-image:
description: The OS image to use on the machines
type: string
min-instances:
description: The minimum number of VMs the autoscaler will create
type: integer
max-instances:
description: The maximum number of VMs the autoscaler will create
type: integer
target-utilization:
description: The target CPU usage for the autoscaler to base its scaling on
type: number
scopes:
description: A list of scopes to create the VM with
type: array
minItems: 1
items:
type: string
```

Configure the deployment template

The template defines several resources needed to create an autoscaled, load-balanced, managed instance group. For a full description of the resources created, see the comments in the template and review the [Bookshelf on Compute Engine](#) (<https://cloud.google.com/python/getting-started/run-on-compute-engine>) tutorial.

Keep in mind that Cloud Deployment Manager provides a way to declaratively define resources, but most of the configuration for a given resource is defined by its API. For example, most of the configuration options for the instance template resource are found in the [instance template resource definition](#)

(<https://cloud.google.com/compute/docs/reference/latest/instanceTemplates#resource>) in the reference documentation for the [Compute Engine API](#) (<https://cloud.google.com/compute/docs/reference/latest/>).

The template is written using the [Jinja](http://jinja.pocoo.org/) (<http://jinja.pocoo.org/>) templating language. When writing Cloud Deployment Manager templates, you can either use Jinja or Python. Jinja has the

advantage of being more declarative, which can be more readable and easier to understand than Python. For some complex deployments, the full expressiveness of Python might make things simpler. In this case, Jinja is sufficient to create the necessary resources.

In a template, certain environment variables are automatically set, and you can access these variables by using the `env` dictionary. In this case, the name of the deployment is referenced as a name to be reused when assigning names to the resources that are created. For more information about available environment variables, see [Using environment variables](https://cloud.google.com/deployment-manager/docs/configuration/templates/use-environment-variables) (<https://cloud.google.com/deployment-manager/docs/configuration/templates/use-environment-variables>)

in the Cloud Deployment Manager documentation:

[7-gce/gce/deployment_manager/bookshelf.jinja](https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja)

(https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja)

IM/GETTING-STARTED-PYTHON/BLOB/STEPS/7-GCE/GCE/DEPLOYMENT_MANAGER/BOOKSHELF.JINJA)

```
{% set NAME = "bookshelf-" + env["deployment"] %}
{% set SERVICE = "bookshelf-" + env["deployment"] + "-frontend" %}
```

The template uses the same startup script that is used in the [Compute Engine tutorial](https://cloud.google.com/python/getting-started/run-on-compute-engine) (<https://cloud.google.com/python/getting-started/run-on-compute-engine>). The script's content is inserted into the template, and the [Jinja indent directive](http://jinja.pocoo.org/docs/dev/templates/)

(<http://jinja.pocoo.org/docs/dev/templates/>) indents the content correctly:

[7-gce/gce/deployment_manager/bookshelf.jinja](https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja)

(https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja)

IM/GETTING-STARTED-PYTHON/BLOB/STEPS/7-GCE/GCE/DEPLOYMENT_MANAGER/BOOKSHELF.JINJA)

```
value: |
{{imports['startup-script']|indent(14, true)}}
```

In a Cloud Deployment Manager Jinja template, you can also refer to resources created elsewhere in the template. In the following example, the backend service uses the reference of the managed instance group to obtain the instance group it should point to:

[7-gce/gce/deployment_manager/bookshelf.jinja](https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja)

(https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja)

```
IM/GETTING-STARTED-PYTHON/BLOB/STEPS/7-GCE/GCE/DEPLOYMENT_MANAGER/BOOKSHELF.JINJA)
```

```
- group: $(ref.{{ NAME }}-frontend-group.instanceGroup)
  zone: {{ properties['zone'] }}
```

Properties specified in `config.yaml` can be used in the template:

[7-gce/gce/deployment_manager/bookshelf.jinja](https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja)

(https://github.com/GoogleCloudPlatform/getting-started-python/blob/7-gce/gce/deployment_manager/bookshelf.jinja)

```
IM/GETTING-STARTED-PYTHON/BLOB/STEPS/7-GCE/GCE/DEPLOYMENT_MANAGER/BOOKSHELF.JINJA)
```

```
minNumReplicas: {{ properties['min-instances'] }}
maxNumReplicas: {{ properties['max-instances'] }}
loadBalancingUtilization:
  utilizationTarget: {{ properties['target-utilization'] }}
```

Cleaning up

To avoid incurring charges to your Google Cloud Platform account for the resources used in this tutorial:

Delete the 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/PRO](https://console.cloud.google.com/iam-admin/projects)

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 deployment

To delete your deployment, enter the following command. This command deletes the load balancer and any Compute Engine instances that are associated with your deployment:

```
gcloud deployment-manager deployments delete my-deployment
```



What's next

- Learn more about [defining Cloud Deployment Manager templates](https://cloud.google.com/deployment-manager/docs/configuration/adding-templates) (<https://cloud.google.com/deployment-manager/docs/configuration/adding-templates>).
- Learn how to [run the Python Bookshelf sample app on Google Kubernetes Engine](https://cloud.google.com/python/tutorials/bookshelf-on-kubernetes-engine) (<https://cloud.google.com/python/tutorials/bookshelf-on-kubernetes-engine>).
- Learn how to [run the Python Bookshelf sample app in the App Engine flexible environment](https://cloud.google.com/python/getting-started/tutorial-app) (<https://cloud.google.com/python/getting-started/tutorial-app>).
- 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>).
- Explore other [Google Cloud services](https://cloud.google.com/) (<https://cloud.google.com/>).

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>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 26, 2019.