Getting started tutorial

Last Updated: 2017-10-26[Edit in GitHub](https://github.com/IBM-Bluemix-Docs/python/blob/master/getting-started.md)

* Congratulations, you deployed a Hello World sample application on IBM® Cloud! To get started, follow this step-by-step guide. Or, [download the sample code](https://console-regional.au-syd.bluemix.net/rest/apps/302cfd45-54d2-4f6c-b41b-f0970aec0afb/starter-download) and explore on your own.

By following the Python getting started tutorial, you'll set up a development environment, deploy an app locally and on IBM® Cloud, and integrate a database service in your app.

**Before you begin**

You'll need the following:

* [IBM Cloud account](https://console.ng.bluemix.net/registration/)
* [Cloud Foundry CLI](https://github.com/cloudfoundry/cli#downloads)
* [Git](https://git-scm.com/downloads)
* [Python](https://www.python.org/downloads/)

**Step 1: Clone the sample app**

Now you're ready to start working with the app. Clone the repo and change to the directory where the sample app is located.

git clone https://github.com/IBM-Bluemix/get-started-python

cd get-started-python

Peruse the files in the *get-started-python* directory to familiarize yourself with the contents.

**Step 2: Run the app locally**

See [The Hitchhiker’s Guide to Python!](http://docs.python-guide.org/en/latest/) for help setting up Python on your system.

Install the dependencies listed in the [requirements.txt](https://pip.readthedocs.io/en/stable/user_guide/#requirements-files) file to be able to run the app locally.

You can optionally use a [virtual environment](https://packaging.python.org/installing/#creating-and-using-virtual-environments) to avoid having these dependencies clash with those of other Python projects or your operating system.

pip install -r requirements.txt

Alternatively with Python3 you can issue

python3 -m pip install -r requirements.txt

Run the app.

python hello.py

View your app at: [http://localhost:8000](http://localhost:8000/)

**Step 3: Prepare the app for deployment**

To deploy to IBM Cloud, it can be helpful to set up a manifest.yml file. The manifest.yml includes basic information about your app, such as the name, how much memory to allocate for each instance and the route. We've provided a sample manifest.yml file in the get-started-python directory.

Open the manifest.yml file, and change the name from GetStartedPython to your app name, *MyTestPythonApp*.

applications:

- name: GetStartedPython

random-route: true

memory: 128M

In this manifest.yml file, **random-route: true** generates a random route for your app to prevent your route from colliding with others. If you choose to, you can replace **random-route: true** with **host: myChosenHostName**, supplying a host name of your choice. [Learn more...](https://console-regional.au-syd.bluemix.net/docs/manageapps/depapps.html#appmanifest)

**Step 4: Deploy the app**

You can use the Cloud Foundry CLI to deploy apps.

Choose your API endpoint

cf api <API-endpoint>

Replace the *API-endpoint* in the command with an API endpoint from the following list.

| **URL** | **Region** |
| --- | --- |
| [https://api.ng.bluemix.net](https://api.ng.bluemix.net/) | US South |
| [https://api.eu-gb.bluemix.net](https://api.eu-gb.bluemix.net/) | United Kingdom |
| [https://api.au-syd.bluemix.net](https://api.au-syd.bluemix.net/) | Sydney |
| [https://api.eu-de.bluemix.net](https://api.eu-de.bluemix.net/) | Frankfurt |

Log in to your IBM Cloud account

cf login

If you cannot log in using the cf login or bx login commands because you have a federated user ID, use either the cf login --sso or bx login --sso commands to log in with your single sign on ID. See [Logging in with a federated ID](https://console.bluemix.net/docs/cli/login_federated_id.html#federated_id) to learn more.

From within the *get-started-python* directory push your app to IBM Cloud

cf push

This can take a minute. If there is an error in the deployment process you can use the command cf logs <Your-App-Name> --recent to troubleshoot.

When deployment completes you should see a message indicating that your app is running. View your app at the URL listed in the output of the push command. You can also issue the

cf apps

command to view your apps status and see the URL.

**Step 5: Add a database**

Next, we'll add a NoSQL database to this application and set up the application so that it can run locally and on IBM Cloud.

1. Log in to IBM Cloud in your Browser. Browse to the Dashboard. Select your application by clicking on its name in the Name column.
2. Click on Connections then Connect new.
3. In the Data & Analytics section, select Cloudant NoSQL DB and Create the service.
4. Select Restage when prompted. IBM Cloud will restart your application and provide the database credentials to your application using the VCAP\_SERVICES environment variable. This environment variable is only available to the application when it is running on IBM Cloud.

Environment variables enable you to separate deployment settings from your source code. For example, instead of hardcoding a database password, you can store this in an environment variable which you reference in your source code. [Learn more...](https://console-regional.au-syd.bluemix.net/docs/manageapps/depapps.html#app_env)

**Step 6: Use the database**

We're now going to update your local code to point to this database. We'll create a json file that will store the credentials for the services the application will use. This file will get used ONLY when the application is running locally. When running in IBM Cloud, the credentials will be read from the VCAP\_SERVICES environment variable.

1. Create a file called vcap-local.json in the get-started-python directory with the following content:
2. {
3. "services": {
4. "cloudantNoSQLDB": [
5. {
6. "credentials": {
7. "username":"CLOUDANT\_DATABASE\_USERNAME",
8. "password":"CLOUDANT\_DATABASE\_PASSWORD",
9. "host":"CLOUDANT\_DATABASE\_HOST"
10. },
11. "label": "cloudantNoSQLDB"
12. }
13. ]
14. }
15. }
16. Back in the IBM Cloud UI, select your App -> Connections -> Cloudant -> View Credentials
17. Copy and paste the username, password, and host from the credentials to the same fields of the vcap-local.json file replacing **CLOUDANT\_DATABASE\_USERNAME**, **CLOUDANT\_DATABASE\_PASSWORD**, and **CLOUDANT\_DATABASE\_URL**.
18. Run your application locally.
19. python hello.py

View your app at: [http://localhost:8000](http://localhost:8000/). Any names you enter into the app will now get added to the database.

Your local app and the IBM Cloud app are sharing the database. View your IBM Cloud app at the URL listed in the output of the push command from above. Names you add from either app should appear in both when you refresh the browsers.