



INTERNAL

## **Demo of the 12 Factor methodology in BTP Cloud Foundry**

This document will guide you step by step to deploy an app that demonstrates the 12 Factor methodology in BTP Cloud Foundry

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.

**THE BEST RUN**



## Table of Contents

<b>DISCLAIMER.....</b>	<b>4</b>
<b>OBJECTIVE.....</b>	<b>4</b>
<b>SCENARIO .....</b>	<b>4</b>
<b>PREREQUISITES.....</b>	<b>4</b>
<b>SUGGESTED PRE-READS.....</b>	<b>4</b>
<b>SOURCE CODE REPOSITORY .....</b>	<b>4</b>
<b>DEPLOYING THE APPLICATION – FROM THE TERMINAL / COMMAND PROMPT .....</b>	<b>5</b>
<b>CONFIGURE POSTGRESQL IN CLOUD FOUNDRY FOR EXTERNAL ACCESS .....</b>	<b>13</b>
<b>CONNECT TO POSTGRESQL AND RUN A SCRIPT IN THE CLOUD.....</b>	<b>13</b>
<b>FINAL STEPS.....</b>	<b>14</b>
<b>SCALING THE APP .....</b>	<b>15</b>
<b>Scaling Horizontally .....</b>	<b>15</b>

## DISCLAIMER

The information shared in this document is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. All functionality presented here is subject to change and may be changed by SAP at any time for any reason without notice.

## OBJECTIVE

This document is aimed for beginners.

The objective of this exercise is to demonstrate the [12 Factor methodology](#) in [BTP Cloud Foundry](#).

Several concepts like Cloud Foundry Spaces, Cloud Foundry Marketplace and many Cloud Foundry commands will be explained, and a sample source code will be provided.

## SCENARIO

This exercise demonstrates the [12 Factor methodology](#) in [BTP Cloud Foundry](#) using a Node.js app.

## PREREQUISITES

- SAP BTP Trial Account - [Get a Free Account on SAP BTP Trial](#)
- Enable **Cloud Foundry Environment** and create an Organization and a Space – This is done automatically for you when you get a Free Account on SAP BTP trial - [Creating a Cloud Foundry Organization and Space](#)

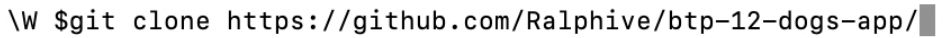
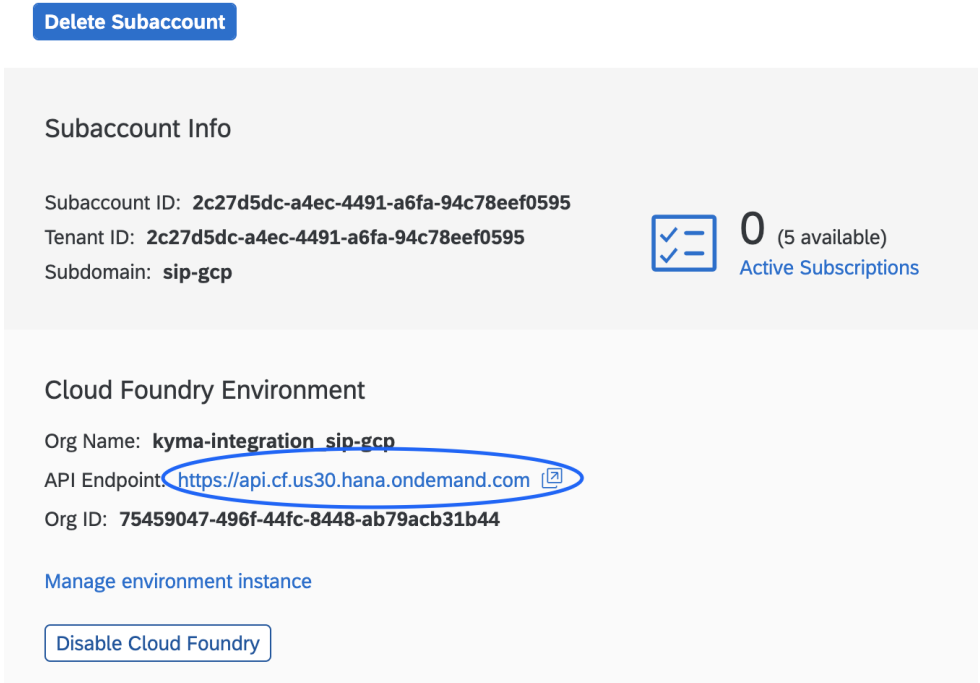
## SUGGESTED PRE-READS

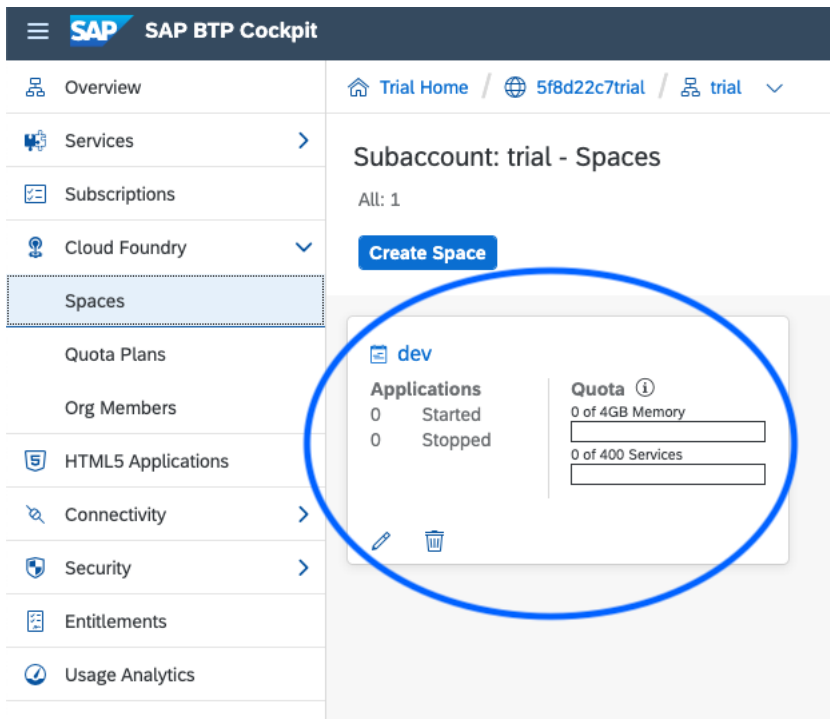
- [Get Ready to Develop on SAP BTP](#)
- [Getting Started in the Cloud Foundry Environment](#)
- [Development in the Cloud Foundry Environment](#)
- [Developing Node.js in the Cloud Foundry Environment](#)
- [Install the Cloud Foundry Command Line Interface \(CLI\)](#)
- [Creating a Cloud Foundry Organization and Space](#)

## SOURCE CODE REPOSITORY


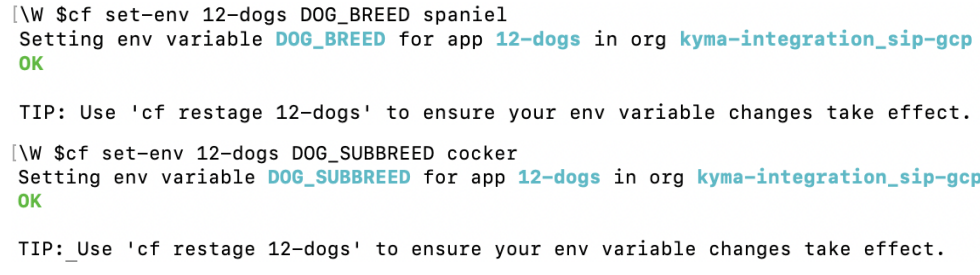
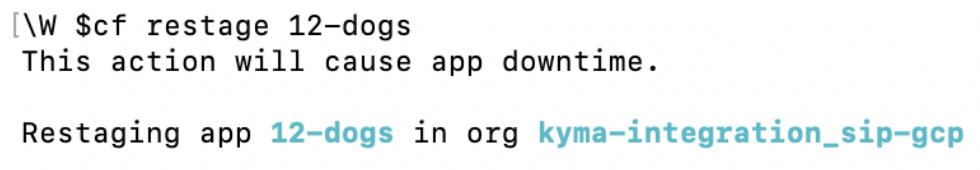
- <https://github.com/Ralphive/btp-12-dogs-app/>

## DEPLOYING THE APPLICATION – FROM THE TERMINAL / COMMAND PROMPT

Explanation	Screenshot
<p><b>Clone the GitHub repository</b></p> <p>Clone the GitHub repository using the following command:</p> <pre>git clone https://github.com/Ralphive/btp-12-dogs-app/</pre>	
<p><b>Login to Cloud Foundry</b></p> <p>The first step is to login to Cloud Foundry.</p> <p>When using the <code>cf login</code> command, the cf CLI prompts for credentials as needed. If you are a member of multiple orgs or spaces, <code>cf login</code> prompts you for which ones to log in to. Otherwise, it targets your org and space automatically.</p> <p>Run the following command:</p> <pre>cf login -a &lt;API-endpoint&gt;</pre> <p>The API endpoint can be retrieved from the BTP subaccount. It is listed under the Cloud Foundry Environment section as shown in the screenshot on the right.</p> <p><b>Example:</b></p> <pre>cf login -a https://api.cf.us30.hana.ondemand.com</pre>	 <p><b>Subaccount Info</b></p> <p>Subaccount ID: 2c27d5dc-a4ec-4491-a6fa-94c78eef0595    Tenant ID: 2c27d5dc-a4ec-4491-a6fa-94c78eef0595    Subdomain: sip-gcp</p> <p><b>Cloud Foundry Environment</b></p> <p>Org Name: kyma-integration sip-gcp    API Endpoint: <a href="https://api.cf.us30.hana.ondemand.com">https://api.cf.us30.hana.ondemand.com</a>    Org ID: 75459047-496f-44fc-8448-ab79acb31b44</p> <p><a href="#">Manage environment instance</a></p> <p><a href="#">Disable Cloud Foundry</a></p> <p><b>Active Subscriptions</b></p> <p>0 (5 available)</p> <p><b>Terminal Command:</b></p> <pre>\W \$cf login -a https://api.cf.us30.hana.ondemand.com</pre> <p>API endpoint: <a href="https://api.cf.us30.hana.ondemand.com">https://api.cf.us30.hana.ondemand.com</a></p> <p>Email: </p>

Explanation	Screenshot
<p>When you get a free Account on SAP BTP Trial, an <b>org</b> and a <b>space</b> with the name “dev” are created for you automatically.</p>	 <p>The screenshot shows the SAP BTP Cockpit interface. On the left is a navigation menu with options: Overview, Services, Subscriptions, Cloud Foundry, Spaces (highlighted), Quota Plans, Org Members, HTML5 Applications, Connectivity, Security, Entitlements, and Usage Analytics. The main area displays 'Subaccount: trial - Spaces' with 'All: 1' and a 'Create Space' button. A modal window for the 'dev' space is open, showing 'Applications' (0 Started, 0 Stopped) and 'Quota' (0 of 4GB Memory, 0 of 400 Services). This modal is circled in blue.</p>
<p><b>Create a Cloud Foundry Space</b></p> <p><b>Note:</b> Please follow this step only if you don't have a space, or if you want to use a new space.</p> <p>Use <code>cf create-space {space_name}</code> to create a new space within your Org.</p> <p>Create a Cloud Foundry Space by using the following command.</p> <p><code>cf create-space &lt;Space-Name&gt;</code></p> <p><b>Example:</b></p> <pre>cf create-space dev</pre>	<pre>[\W \$cf create-space dev</pre>
<p><b>Target the new space</b></p> <p><code>cf target -o ORG -s SPACE</code></p> <p><b>Example:</b></p>	<pre>[\W \$cf target -o "kyma-integration_sip-gcp" -s "dev"</pre>

Explanation	Screenshot
<pre>cf target -o "kyma- integration_sip-gcp" -s "dev"</pre>	
<p><b>Push the app to Cloud Foundry</b></p> <p>The <code>cf</code> CLI command <code>cf push</code> pushes apps to CF. There are two main ways to run the <code>cf push</code> command:</p> <ul style="list-style-type: none"> <li>• Run <code>cf push APP-NAME</code> to push an app the easiest way, using default settings.</li> <li>• Run the <code>cf push</code> command with flags and helper files to customize: <ul style="list-style-type: none"> <li>a. How the pushed app runs, including its route (URL), instance count, and memory limits.</li> <li>b. How the push process works: whether it's configured with a manifest, runs a startup script, or limits files uploaded to the Cloud Controller.</li> </ul> </li> </ul> <p>For more information, please refer to <a href="#">this document</a>.</p> <p>Navigate to the directory you cloned and push the app to Cloud Foundry.</p> <pre>cf push --random-route</pre>	<pre>\W \$cf push --random-route</pre>

Explanation	Screenshot
<p>The app will be deployed and started. You can now access it via the route displayed in the terminal.</p> <p>Example:</p> <p><a href="https://12-dogs-bogus-raven-el.cfapps.us30.hana.ondemand.com">https://12-dogs-bogus-raven-el.cfapps.us30.hana.ondemand.com</a></p>	 <pre> Waiting for app 12-dogs to start...  Instances starting... Instances starting... Instances starting... Instances starting...  name: 12-dogs requested state: started routes: 12-dogs-bogus-raven-el.cfapps.us30.hana.ondemand.com last uploaded: Tue 08 Jun 15:55:38 EDT 2021 stack: cflinuxfs3 buildpacks:   name          version  detect output  buildpack name   https://github.com/cloudfoundry/nodejs-buildpack.git  1.7.53  nodejs        nodejs  type: web sidecars: instances: 1/1 memory usage: 128M start command: npm start  state since      cpu  memory  disk  details #0 _running 2021-06-08T19:55:51Z 0.0% 0 of 0 0 of 0 </pre>
<p><b>Set environment variables</b></p> <p>The user-provided variables set using the <code>cf set-env</code> command</p> <p>Set the environment variables using the following commands.</p> <pre>cf set-env 12-dogs DOG_BREED spaniel</pre> <pre>cf set-env 12-dogs DOG_SUBBREED cocker</pre>	 <pre> \$ cf set-env 12-dogs DOG_BREED spaniel Setting env variable DOG_BREED for app 12-dogs in org kyma-integration_sip-gcp OK  TIP: Use 'cf restage 12-dogs' to ensure your env variable changes take effect.  \$ cf set-env 12-dogs DOG_SUBBREED cocker Setting env variable DOG_SUBBREED for app 12-dogs in org kyma-integration_sip-gcp OK  TIP: Use 'cf restage 12-dogs' to ensure your env variable changes take effect. </pre>
<p><b>Restage Your App</b></p> <p>To restage your app, run the following command:</p> <pre>cf restage YOUR-APP</pre> <p>Restaging your app stops your app and restages it, by compiling a new droplet and starting it.</p> <p>Restage your app if you have changed the environment in a way that affects your staging process, such as setting an environment variable that</p>	 <pre> \$ cf restage 12-dogs This action will cause app downtime.  Restaging app 12-dogs in org kyma-integration_sip-gcp </pre>



Explanation	Screenshot
<p>the buildpack consumes. The staging process has access to environment variables, so the environment can affect the contents of the droplet. You should also restage your app whenever you edit any configuration settings, such as when you rename it, add metadata, or configure health checks. The new settings often do not take effect until you restage the app.</p> <p>Restaging your app compiles a new droplet from your app without updating your app source.</p> <p>Use 'cf restage 12-dogs' to ensure your env variable changes take effect.</p> <p>Example:</p> <pre>cf restage 12-dogs</pre>	
<p>All instances of the app are restarted when restaging the app.</p>	<pre>Stopping app... Waiting for app to start...  Instances starting... Instances starting... Instances starting...  name: 12-dogs requested state: started routes: 12-dogs-bogus-raven-el.cfapps.us30.hana.ondemand.com last uploaded: Tue 08 Jun 15:59:54 EDT 2021 stack: cflinuxfs3 buildpacks:   name version detect output buildpack name   https://github.com/cloudfoundry/nodejs-buildpack.git 1.7.53 nodejs nodejs  type: web sidecars: instances: 1/1 memory usage: 128M state since cpu memory disk details #0 running 2021-06-08T20:00:06Z 1.2% 30.6M of 128M 119.7M of 1G</pre>
<p><b>List Marketplace Services</b></p> <p>After targeting and logging into Cloud Foundry, run the <code>cf marketplace</code> cf CLI command to view the services available to your</p>	<pre>\W \$cf marketplace  \W \$cf marketplace -e postgresql-db</pre>

Explanation	Screenshot										
<p>targeted organization. Available services may differ between organizations and between Cloud Foundry marketplaces.</p> <p>Get the list of service offerings that are available in the service marketplace by using the following command.</p> <div>cf marketplace</div> <p>You'll notice the following PostgreSQL service offering in the list that is returned.</p> <pre>postgresql-db trial PostgreSQL service on SAP BTP</pre> <p>Use the following command to view descriptions of individual plans of a given service offering.</p> <pre>cf marketplace -e SERVICE_OFFERING</pre> <p>Example:</p> <div>cf marketplace -e postgresql-db</div> <p>In the next step, create an instance of this service using one of its plans.</p>	<table><tr><th>plan</th><th>description</th><th>free or paid</th><th>costs</th><th>available</th></tr><tr><td>trial</td><td>Trial PostgreSQL service offering</td><td>free</td><td></td><td>yes</td></tr></table>	plan	description	free or paid	costs	available	trial	Trial PostgreSQL service offering	free		yes
plan	description	free or paid	costs	available							
trial	Trial PostgreSQL service offering	free		yes							
<p><b>Creating a Service Instance</b></p> <p>You can create a service instance with the following command: cf create-service SERVICE PLAN SERVICE_INSTANCE</p>	<pre>\$cf create-service postgresql-db trial dog-db</pre>										

Explanation	Screenshot
<p>Use the information in the list below to replace SERVICE, PLAN, and SERVICE_INSTANCE with appropriate values.</p> <ul style="list-style-type: none"> <li>• SERVICE: The name of the service you want to create an instance of.</li> <li>• PLAN: The name of a plan that meets your needs. Service providers use plans to offer varying levels of resources or features for the same service.</li> <li>• SERVICE_INSTANCE: The name you provide for your service instance. You use this name to refer to your service instance with other commands. Service instance names can include alpha-numeric characters, hyphens, and underscores, and you can rename the service instance at any time.</li> </ul> <p>Create the DB service Instance using the following command.</p> <pre>cf create-service &lt;Service Name&gt; &lt;Service Plan&gt; &lt;Instance Name&gt;</pre> <p><b>Example:</b></p> <pre>cf create-service postgresql-db trial dog-db</pre>	
<p><b>Bind the Service Instance to the App</b></p> <p>Depending on the service, <u>binding a service instance</u></p>	<pre>[W] \$cf bind-service 12-dogs dog-db</pre> <p>Binding service <b>dog-db</b> to app <b>12-dogs</b> in org <b>kyma-integration_sip-gcp</b></p>

Explanation	Screenshot
<p>to your app may deliver credentials for the service instance to the app. See the <a href="#">Delivering Service Credentials to an App</a> topic for more information.</p> <p>Binding a service instance to an app may also trigger app logs to be streamed to the service instance. For more information, see <a href="#">Streaming App Logs to Log Management Services</a>.</p> <p>Bind the service instance with the application using the following command.</p> <p><i>cf bind-service &lt;Application-Name&gt; &lt;Service-Instance-Name&gt;</i></p> <p><b>Example:</b></p> <pre>cf bind-service 12-dogs dog-db</pre>	

## CONFIGURE POSTGRES SQL IN CLOUD FOUNDRY FOR EXTERNAL ACCESS

When deploying the app locally, PostgreSQL and be configured in Cloud Foundry for external access via the steps listed below.

First, create a service key to access the PostgreSQL service from outside Cloud Foundry.

```
cf create-service-key <SERVICE_INSTANCE> <SERVICE_KEY>
```

```
cf create-service-key dog-db db-key
```

Then, check the PostgreSQL credentials. User, Password, IP and DB Name.

```
cf service-key <SERVICE_INSTANCE> <SERVICE_KEY>
```

Example:

```
cf service-key dog-db db-key
```

The above command will return a response along with the **dbname**, **hostname**, **password**, **port** and other details.

Then, create a secure SSH tunnel from your local system to the remote PostgreSQL.

```
cf ssh -L <local-port>:<service-IP>:<service-port> <your-app>
```

Example:

```
cf ssh -L 63306:10.11.241.35:60262 12-dogs
```

## CONNECT TO POSTGRES SQL AND RUN A SCRIPT IN THE CLOUD

Now connect to PostgreSQL from another terminal.

```
psql -d <DatabaseName> -U <UserName> -p <LocalPort> -h localhost
```

```
psql -d zTumYIKsJnai -U e30c278d60b4 -p 63306 -h localhost
```

or

```
\c <DatabaseName> <UserName> localhost <LocalPort>  
\c zTumYIKsJnai e30c278d60b4 localhost 63306
```

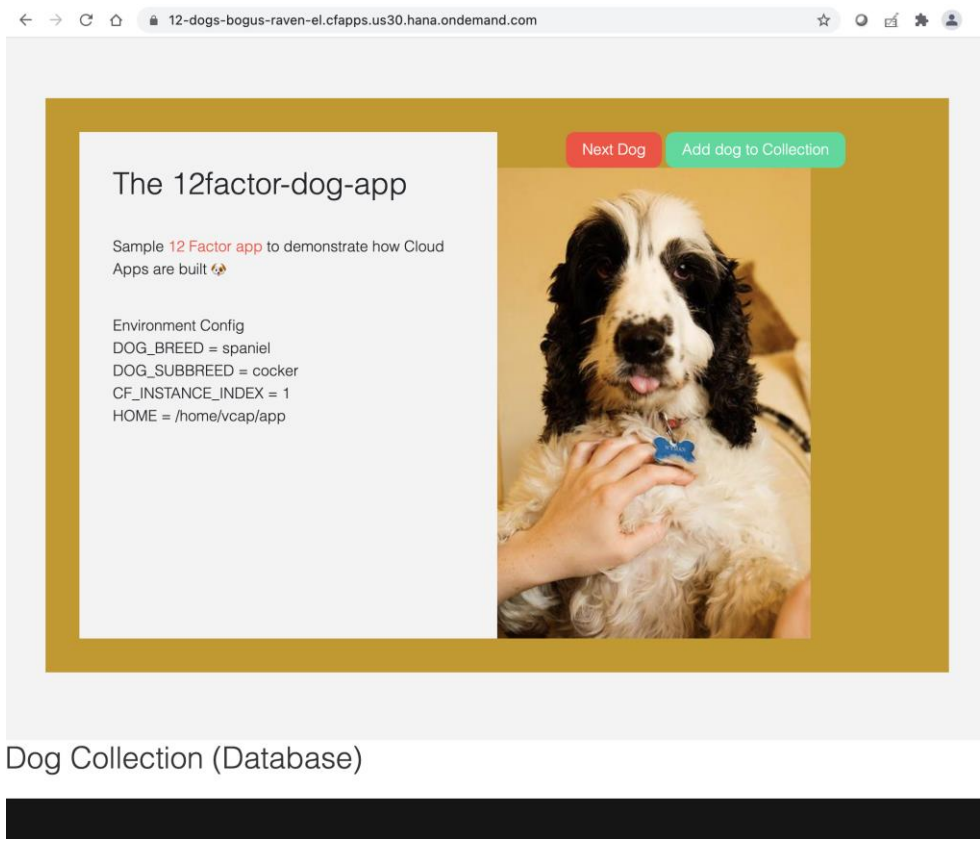
After inserting the password, you can list all the DBs.

```
\l
```

And, create the table that the app needs by running the code that is within the following file: /db/initialize.sql

```
SELECT * FROM dog_collection
```

## FINAL STEPS

Explanation	Screenshot
<p><b>Restart Your App</b></p> <p>To restart your app, run the following command:</p> <pre>cf restart YOUR-APP</pre> <p><b>Example:</b></p> <pre>cf restart 12-dogs</pre>	<pre>[W \$cf restart 12-dogs Restarting app 12-dogs in org kyma-integration_sip-gcp</pre>
<p><b>Access you App</b></p> <p>Run the app on the route displayed in the terminal.</p> <p><b>Example:</b></p> <p><a href="https://12-dogs-bogus-raven-el.cfapps.us30.hana.ondemand.com">https://12-dogs-bogus-raven-el.cfapps.us30.hana.ondemand.com</a></p>	

## SCALING THE APP

### Scaling Horizontally

Horizontally scaling an app creates or destroys instances of your app.

Incoming requests to your app are automatically load balanced across all instances of your app, and each instance handles tasks in parallel with every other instance. Adding more instances allows your app to handle increased traffic and demand.

Use the following command to horizontally scale your app. Cloud Foundry will increase or decrease the number of instances of your app to match INSTANCES.

*cf scale <your app> -i <# instances>*

Example:

```
cf scale 12-dogs -i 3
```

```
cf app 12-dogs
```

Reference: [Scaling an App Using cf scale](#)