

Cloud Foundry 12-Factor App Hands-On Instructions





TABLE OF CONTENTS

PREREQUISITES.....	3
1. Local Resources.....	3
2. Cloud Resources	3
2.3. Create Redis Cache instance	4
2.4. Create a Microsoft Azure PostgreSQL DB on a learn sandbox	8
STEP 1: PREPARE THE APP ON YOUR COMPUTER.....	11
STEP 2: RUN THE APP LOCALLY	12
STEP 3: PREPARE AZURE POSTGRESQL DATABASE AND TABLES	17
STEP 4: RUN ON SCP CF ENVIRONMENT	19



PREREQUISITES

1. Local Resources

1.1. Install Git client

<https://git-scm.com/downloads>

1.2. Install NodeJS runtime

<https://nodejs.org/en/download/>

1.3. Install PostgreSQL

<https://www.postgresql.org/download/>

1.4. Install Redis Cache

<https://redis.io/download>

For Windows: <https://github.com/microsoftarchive/redis/releases>

1.5. Install Visual Studio Code IDE

<https://code.visualstudio.com/>

1.6. Install Cloud Foundry CLI

<https://github.com/cloudfoundry/cli#downloads>

2. Cloud Resources

2.1. Configure your SAP Cloud Platform CF Environment

<https://developers.sap.com/tutorials/cp-trial-quick-onboarding.html>

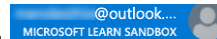
2.2. Create a Microsoft Azure free account

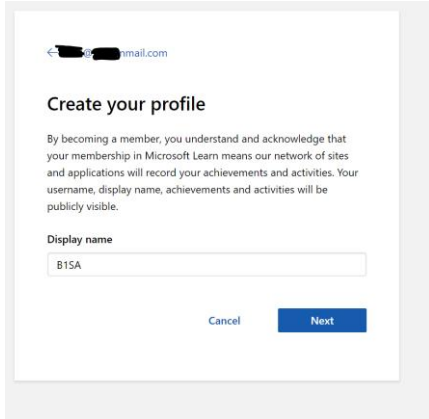
There are at least two options to create a Microsoft Azure Account free of charge for learning purposes and a limited period of time:

- You can use a Sandbox to test and learn Microsoft Azure services for a very short period of time (4 hours): <https://docs.microsoft.com/en-us/learn/azure/?tab=tab-learn-for-free> (see the step-by-step instructions in the next page); OR
- Using your credit card - follow this link <http://azure.microsoft.com/free>, create your account and use it free of charge for 30 days. Nevertheless, a credit card is required to validate identity and avoid dummy accounts.

2.2.1. Create an Azure sandbox account for learning purposes

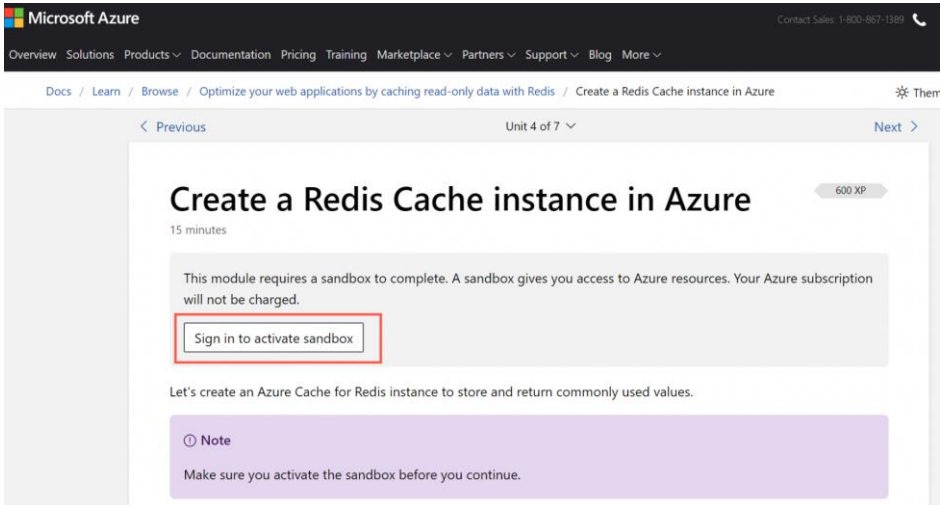
Please follow the steps below to create your *learn sandbox*.


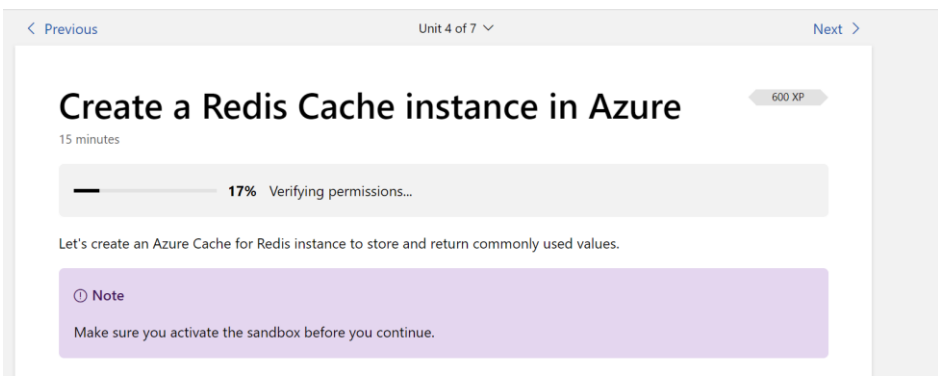
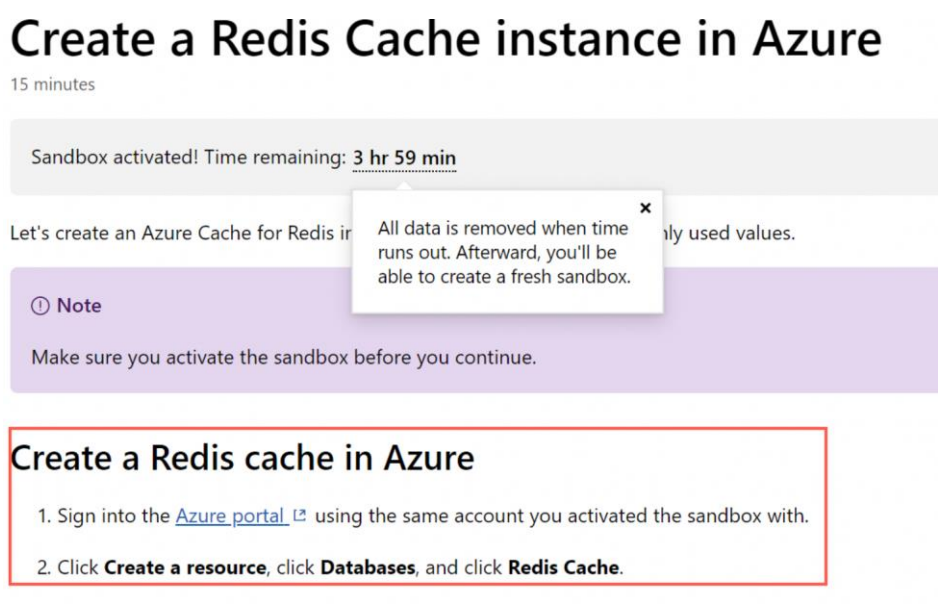


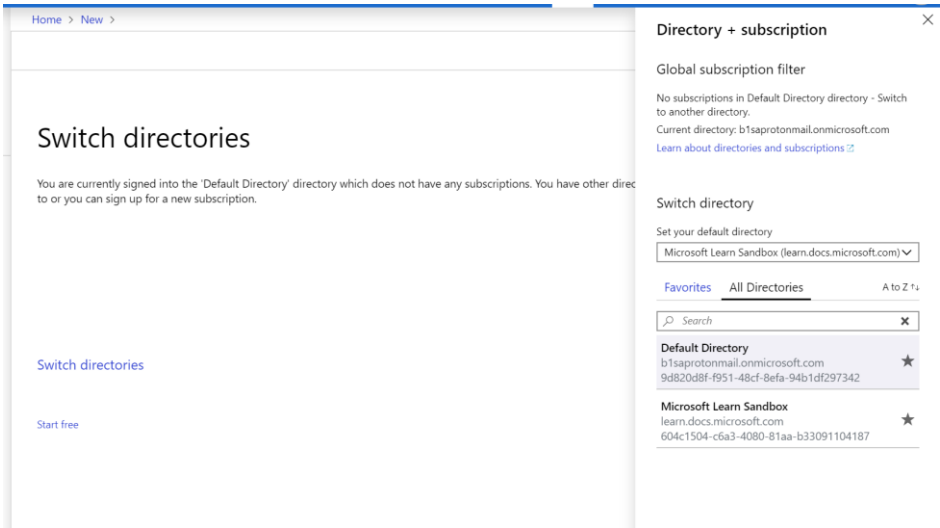
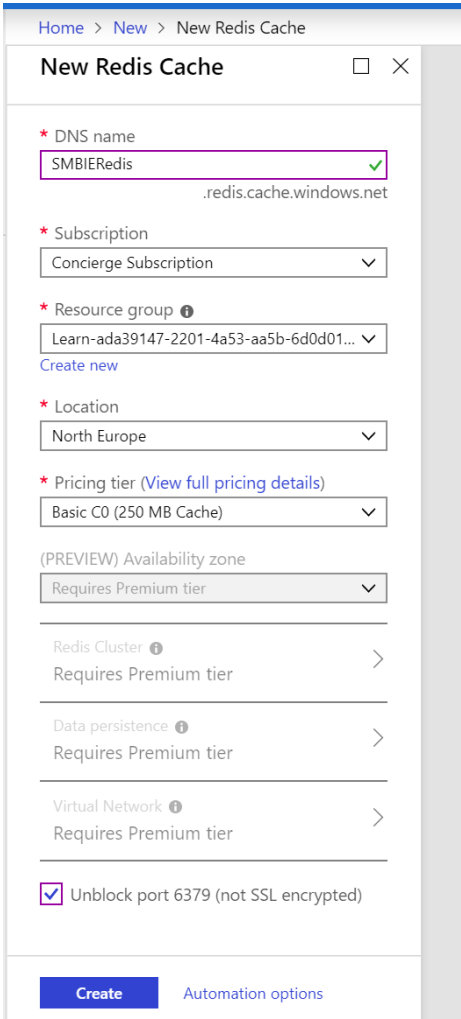
Explanation	Screenshot
<p>Create an account:</p> <p>https://docs.microsoft.com/en-us/learn/azure/?tab=tab-learn-for-free</p> <p>You can use the same email as of your Microsoft account.</p>	 A screenshot of the 'Create your profile' form in the Microsoft Learn Sandbox. The form has a title 'Create your profile' and a paragraph explaining that by becoming a member, the user understands and acknowledges that their membership in Microsoft Learn means their network of sites and applications will record their achievements and activities. Below this, there is a 'Display name' field with the text 'B1SA' entered. At the bottom of the form are two buttons: 'Cancel' and 'Next'.

2.3. Create Redis Cache instance

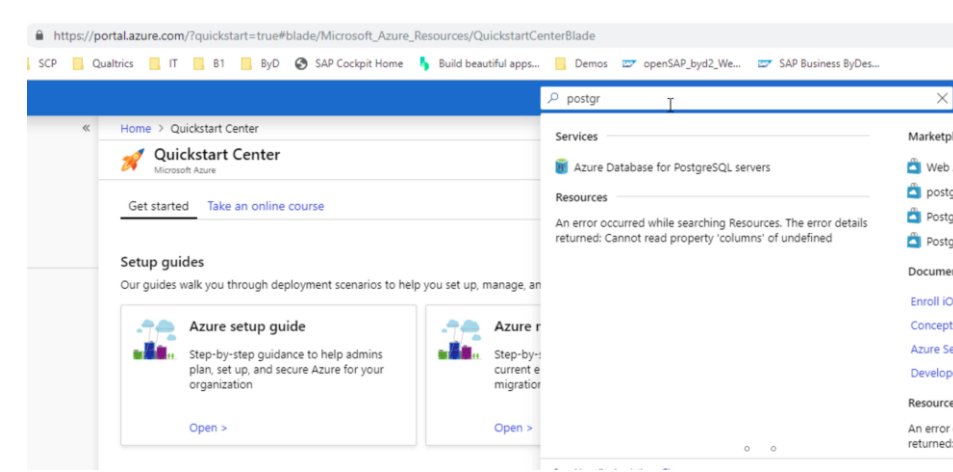
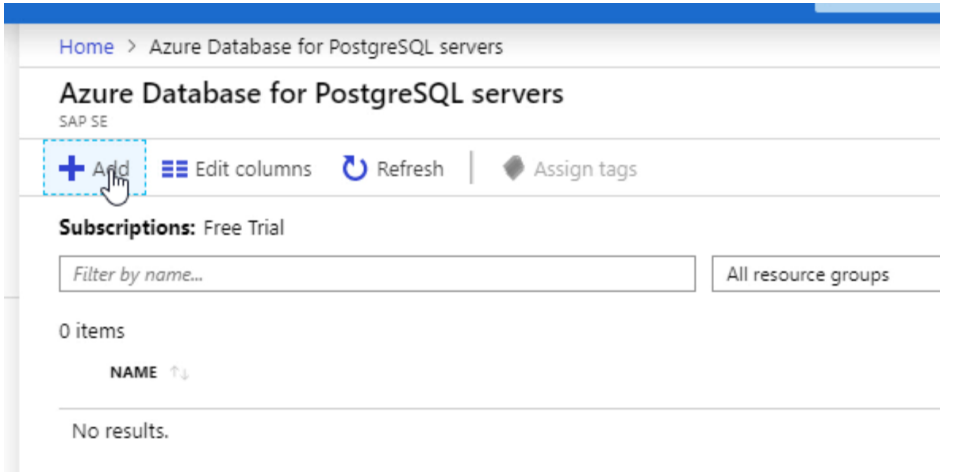
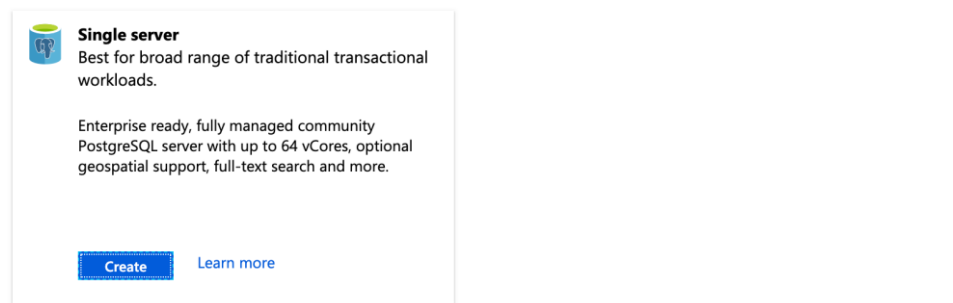
Please follow the steps below to get the learn sandbox running

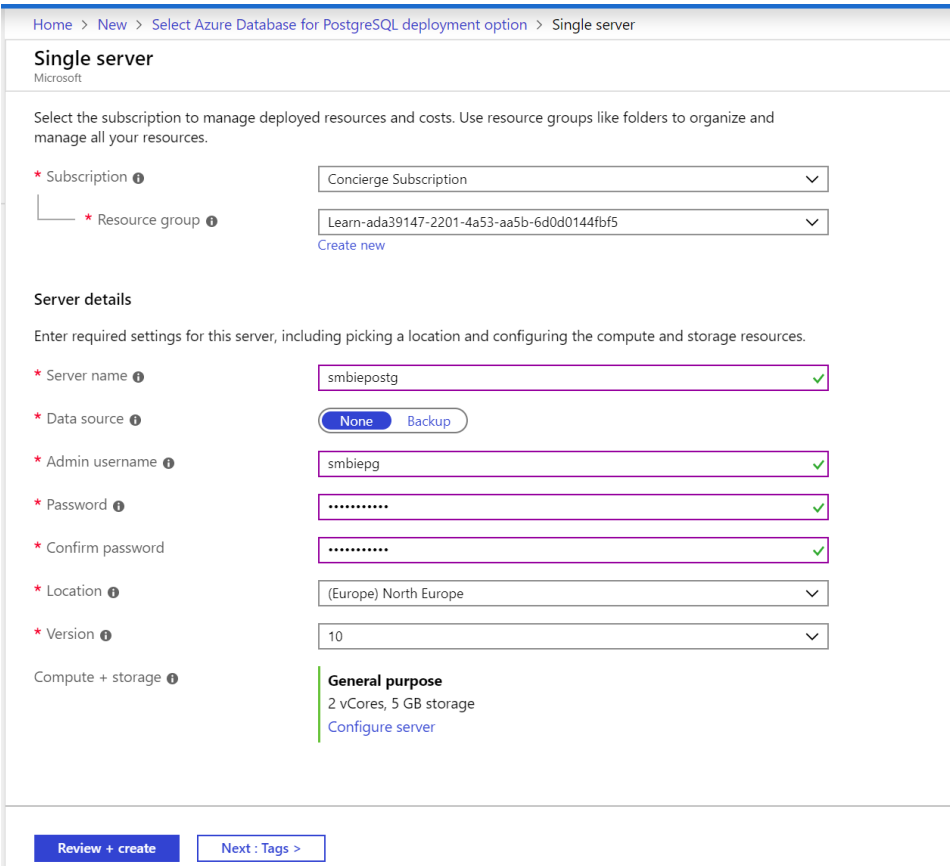
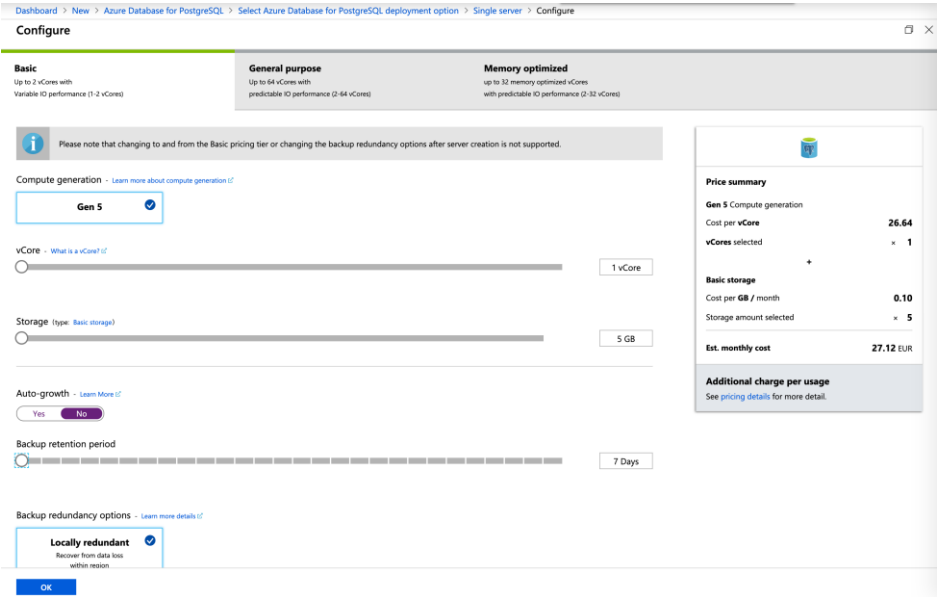
Explanation	Screenshot
<p>Open this exercise tutorial:</p> <p>https://docs.microsoft.com/en-us/learn/modules/optimize-your-web-apps-with-redis/4-exercise-create-redis-cache</p> <p>Press the button "Sign to activate sandbox"</p>	 A screenshot of the 'Create a Redis Cache instance in Azure' tutorial page on the Microsoft Azure website. The page has a dark header with the Microsoft Azure logo and navigation links. The main content area has a title 'Create a Redis Cache instance in Azure' and a sub-header '15 minutes'. Below the title, there is a paragraph explaining that this module requires a sandbox to complete. A button labeled 'Sign in to activate sandbox' is highlighted with a red rectangle. Below the button, there is a 'Note' box with the text 'Make sure you activate the sandbox before you continue.'.

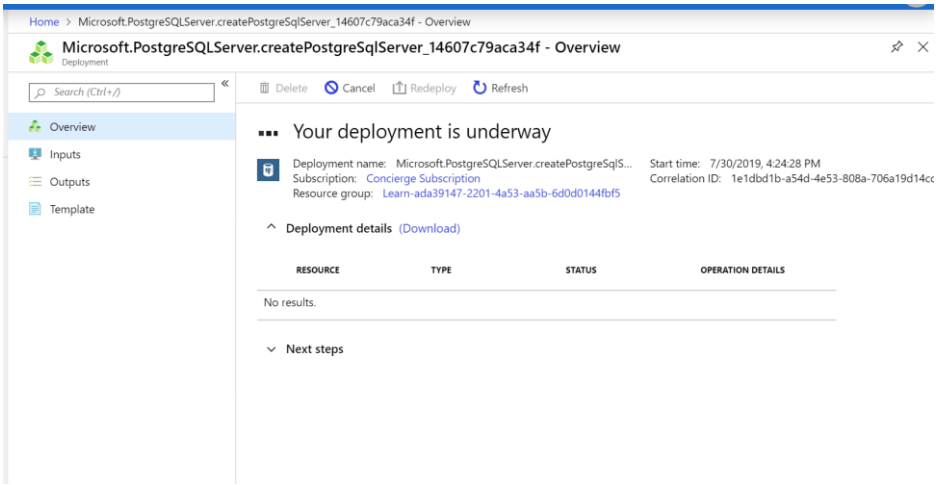
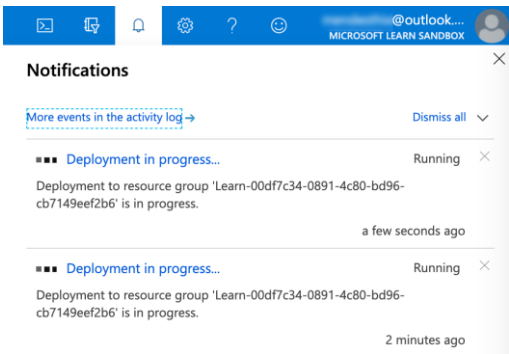
Explanation	Screenshot
Press Activate sandbox.	
The page will show the progress:	
<p>Wait until the sandbox is created.</p> <p>The page will show the sandbox remaining time (4 hours).</p> <p>Follow the steps shown in that page to Create a Redis cache in Azure.</p>	

Explanation	Screenshot
<p>While trying to create the Redis cache IF you get an error asking you to switch directories just follow the screens and select the "Microsoft Learn Sandbox" directory.</p>	
<p>Fill in the creation form as per this screen shot.</p> <p>Change the Location to match the nearest to you.</p> <p>Do not forget to unblock the port 6379.</p>	

2.4. Create a Microsoft Azure PostgreSQL DB on a learn sandbox

Explanation	Screenshot
Search for postgresql services in your account (directory sandbox).	 A screenshot of the Azure Quickstart Center search results for 'postgresql'. The search bar at the top contains 'postgr'. The results are divided into 'Services' and 'Resources'. Under 'Services', 'Azure Database for PostgreSQL servers' is listed. Under 'Resources', there is an error message: 'An error occurred while searching Resources. The error details returned: Cannot read property 'columns' of undefined'. The left sidebar shows 'Home > Quickstart Center' and 'Setup guides'.
Select “Azure Database for PostgreSQL servers” and hit Add	 A screenshot of the 'Azure Database for PostgreSQL servers' page. The breadcrumb is 'Home > Azure Database for PostgreSQL servers'. The title is 'Azure Database for PostgreSQL servers'. Below the title, there is a '+ Add' button, 'Edit columns', 'Refresh', and 'Assign tags'. The 'Subscriptions: Free Trial' section is visible. Below it, there is a 'Filter by name...' input field and a 'All resource groups' button. The main content area shows '0 items' and a table with a 'NAME' header. The text 'No results.' is displayed at the bottom.
Go for the “ Single Server ” setup.	 A screenshot of the 'Single server' setup option. It features a PostgreSQL icon, the title 'Single server', and the description 'Best for broad range of traditional transactional workloads.' Below this, it says 'Enterprise ready, fully managed community PostgreSQL server with up to 64 vCores, optional geospatial support, full-text search and more.' At the bottom, there are 'Create' and 'Learn more' buttons.

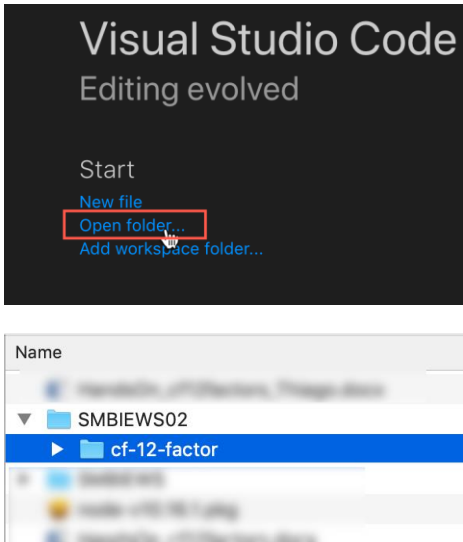
Explanation	Screenshot
<p>Enter all required details.</p> <p>Choose your own Server name.</p> <p>Select Configure server link.</p>	
<p>Reduce the Compute and Storage to the minimum basic level.</p>	

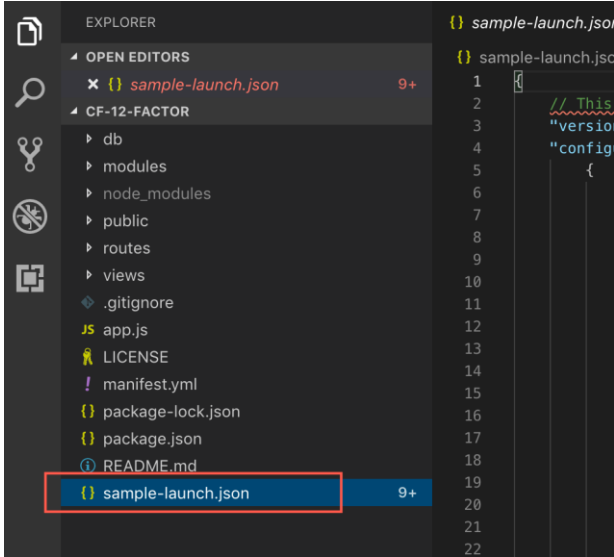
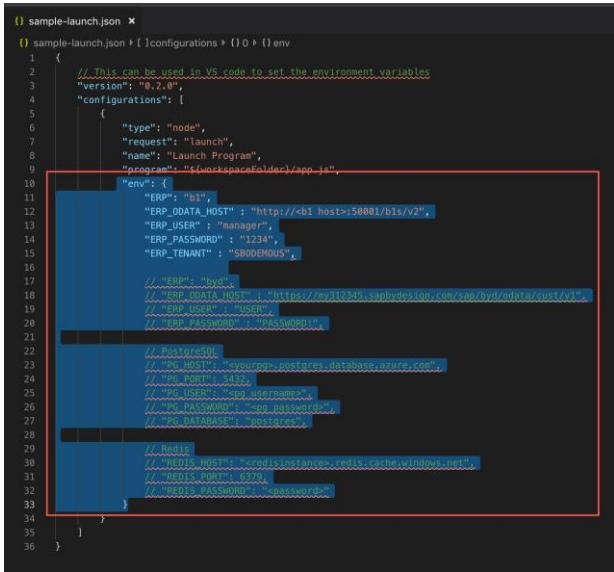
Explanation	Screenshot
<p>Click Review + Create.</p> <p>Click Create.</p> <p>The PostgreSQLServer deployment starts.</p>	
<p>Wait until the deployment is finished to start using PostgreSQL and Redis.</p>	

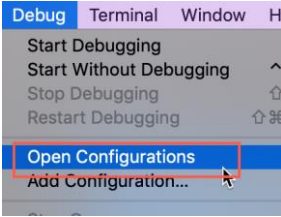
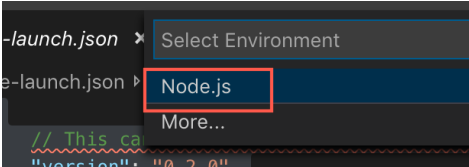
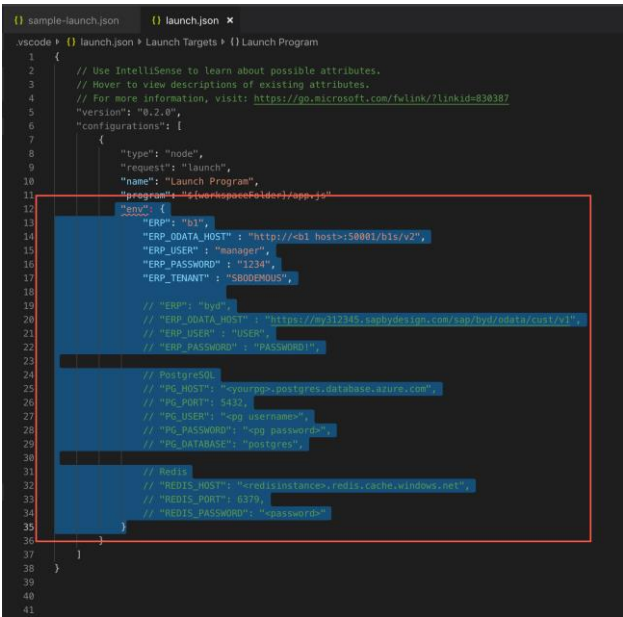
STEP 1: PREPARE THE APP ON YOUR COMPUTER

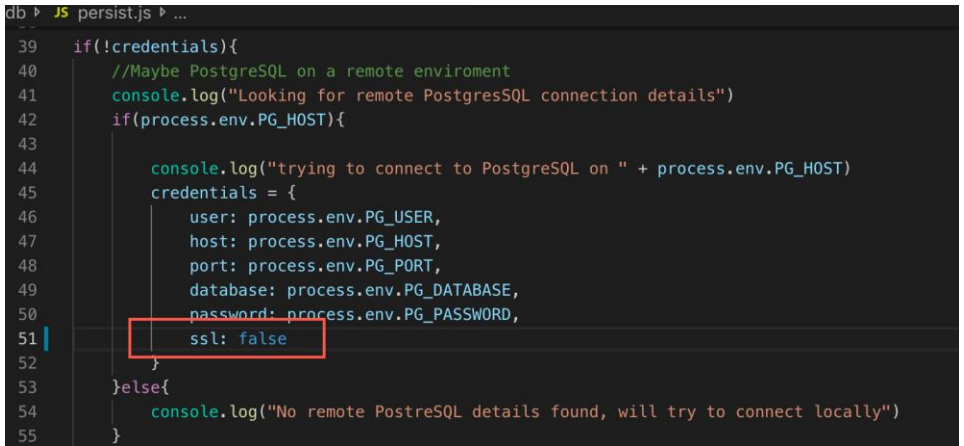
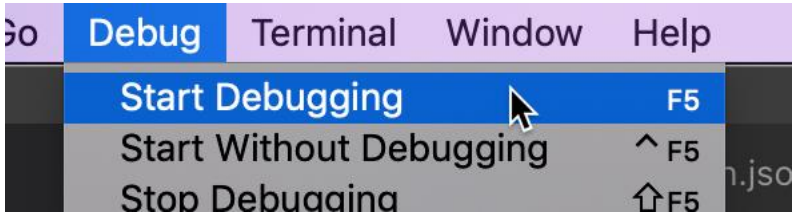
Explanation	Screenshot
<p>Clone the repository</p> <p>\$ git clone https://github.com/Ralphive/cf-12-factor.git</p>	<pre> :Desktop I844173\$ mkdir SMBIEWS02 :Desktop I844173\$ cd SMBIEWS SMBIEWS/ SMBIEWS02/ :Desktop I844173\$ cd SMBIEWS SMBIEWS/ SMBIEWS02/ :Desktop I844173\$ cd SMBIEWS02/ :SMBIEWS02 I844173\$ git clone https://github.com/Ralphive/cf-12-factor.git Cloning into 'cf-12-factor'... remote: Enumerating objects: 34, done. remote: Counting objects: 100% (34/34), done. remote: Compressing objects: 100% (22/22), done. remote: Total 349 (delta 14), reused 26 (delta 11), pack-reused 315 Receiving objects: 100% (349/349), 92.03 KiB 532.00 KiB/s, done. Resolving deltas: 100% (159/159), done. </pre>
<p>Install the dependencies</p> <p>\$ cd cf-12-factor/ \$ npm install</p>	<pre> :SMBIEWS02 I844173\$ ls cf-12-factor :SMBIEWS02 I844173\$ cd cf-12-factor/ :cf-12-factor I844173\$ ls LICENSE app.js manifest.yml package-lock.json public sample-launch.json README.md db modules package.json routes views :cf-12-factor I844173\$ npm install added 127 packages from 105 contributors and audited 238 packages in 1.4 62s found 0 vulnerabilities :cf-12-factor I844173\$ █ </pre>

STEP 2: RUN THE APP LOCALLY

Explanation	Screenshot
<p>Go to the terminal and run the psql command.</p> <p>Then create the table to store items in Postgres:</p> <pre>CREATE TABLE IF NOT EXISTS fact12_bps (code varchar(256) NOT NULL primary key, name varchar(256) NOT NULL, type varchar(1) NOT NULL default 'C', integrated boolean NOT NULL default false);</pre>	<pre>:~ I844173\$ /Applications/Postgres.app/Contents/Versions/11/bin/psql -p5432 "postgres" psql (11.4) Type "help" for help. postgres=# CREATE TABLE IF NOT EXISTS fact12_bps postgres=# (code varchar(256) NOT NULL primary key, postgres=# name varchar(256) NOT NULL, postgres=# type varchar(1) NOT NULL default 'C', postgres=# integrated boolean NOT NULL default false); CREATE TABLE postgres=#</pre>
<p>Open Visual Studio Code;</p> <p>Click "Open folder...";</p> <p>And choose the folder with the repository you cloned;</p>	

Explanation	Screenshot
<p>Open sample-launch.json file;</p> <p>Copy the environment lines;</p>	 

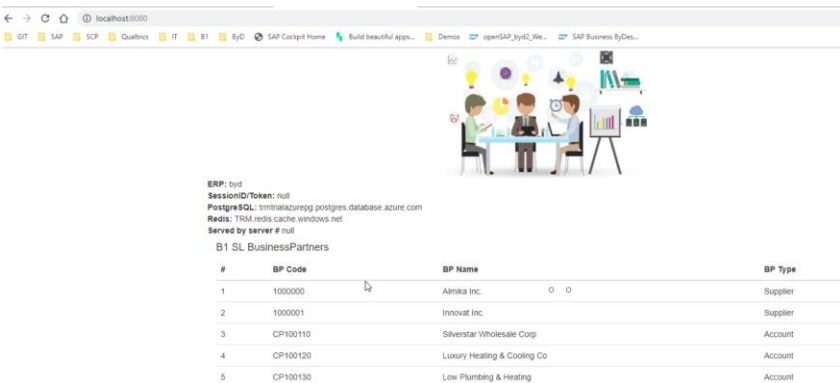
Explanation	Screenshot
<p>Go to the main menu and select Debug > Open Configurations;</p> <p>Then select Node.js;</p> <p>And paste the environment content after the program value.</p> <p>Setup the environment variable with actual values and save the file.</p> <p>Note: this app runs either with B1 OR ByD, not both at same time.</p>	  

Explanation	Screenshot
Customize the variables depending on your environment.	<pre> "version": "0.2.0", "configurations": [{ "type": "node", "request": "launch", "name": "Launch Program", "program": "\${workspaceFolder}\\app.js", "env": { "ERP": "byd", "ERP_ODATA_HOST" : "https://XYZ.sapbydesign.com/sap/byd/odata/cust/v1", "ERP_USER" : "XYZ", "ERP_PASSWORD" : "XYZ", // PostgreSQL "PG_HOST": "localhost", "PG_PORT": 5432, "PG_USER": "postgres", "PG_PASSWORD": "XYZ", "PG_DATABASE": "postgres", // Redis "REDIS_HOST": "localhost", "REDIS_PORT": 6379, "REDIS_PASSWORD": "" } }] </pre>
<p>Open file db/persist.js and set the ssl to false.</p> <p>Save it.</p>	 <pre> 39 if(!credentials){ 40 //Maybe PostgreSQL on a remote enviroment 41 console.log("Looking for remote PostgreSQL connection details") 42 if(process.env.PG_HOST){ 43 44 console.log("trying to connect to PostgreSQL on " + process.env.PG_HOST) 45 credentials = { 46 user: process.env.PG_USER, 47 host: process.env.PG_HOST, 48 port: process.env.PG_PORT, 49 database: process.env.PG_DATABASE, 50 password: process.env.PG_PASSWORD, 51 ssl: false 52 } 53 }else{ 54 console.log("No remote PostreSQL details found, will try to connect locally") 55 } </pre>
Go to the main menu and select Debug - Start Debugging	

Explanation

Screenshot

Open the page <http://localhost:8080> on your browser.

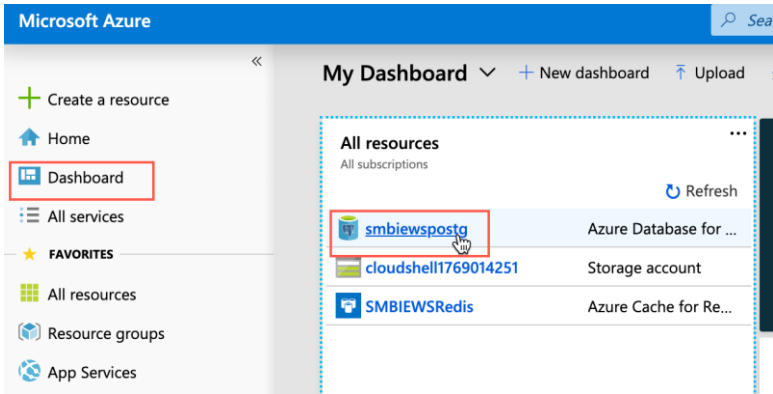
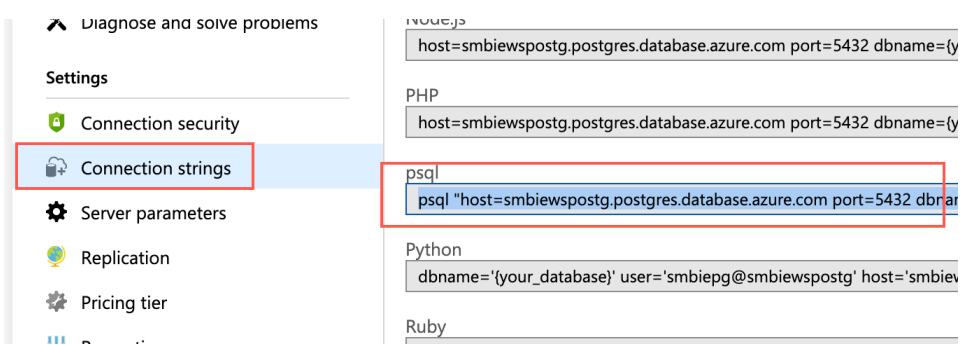
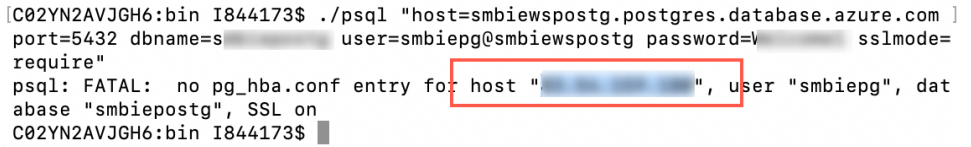
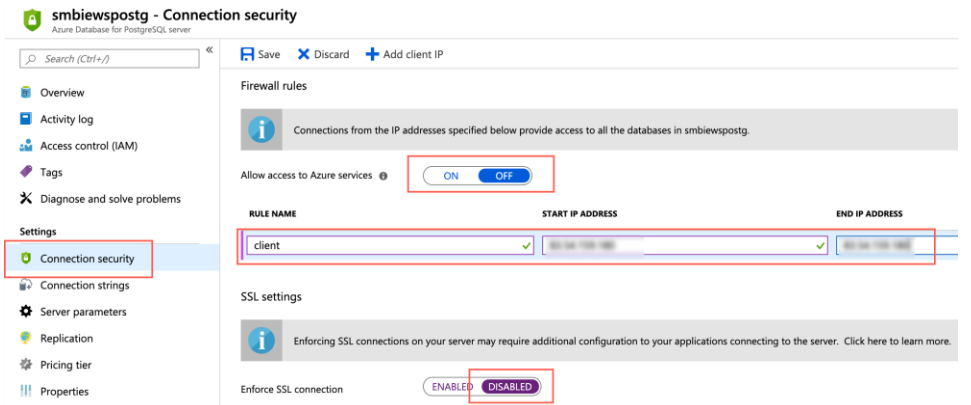


ERP: byd
SessionID/Token: null
PostgreSQL: 1m1n1a2u2e2p2 postgres.database.azure.com
Redis: 1104.redis.cache.windows.net
Served by server: null

B1 SL BusinessPartners

#	BP Code	BP Name	BP Type
1	1000000	Almika Inc.	Supplier
2	1000001	Innovat Inc.	Supplier
3	CP100110	Silverstar Wholesale Corp	Account
4	CP100120	Luxury Heating & Cooling Co	Account
5	CP100130	Low Plumbing & Heating	Account

STEP 3: PREPARE AZURE POSTGRESQL DATABASE AND TABLES

Explanation	Screenshot
Go to the Azure dashboard and select the postgres database	
<p>Under settings, select Connection strings.</p> <p>Copy the string.</p>	
<p>Paste the command into the terminal and run it.</p> <p>The request will fail with an error indicating that your local IP is not authorized.</p> <p>Copy the IP from the error message.</p>	
<p>Go to the Microsoft Azure PostgreSQL Connection security menu on the Settings.</p> <p>Select ON to Allow access to Azure services.</p> <p>Add your local IP to the Firewall Rules.</p> <p>And disable SSL connections.</p> <p>Click SAVE</p>	

Explanation	Screenshot
Run the psql command to connect again.	<pre>[C02YN2AVJGH6:bin I844173\$./psql "host=... .postgres.database.azure.com] port=5432 dbname=postgres user=smbiepg@smbiewspostg password=... sslmode=require" psql (11.4, server 10.9) SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off) Type "help" for help. postgres=> █</pre>
<p>Create a DB</p> <p>create database smbsadb;</p> <p>Exit and connect again, but now to the new DB.</p>	<pre>[postgres=> create database smbsadb; CREATE DATABASE postgres=> exit [C02YN2AVJGH6:bin I844173\$./psql "host=smbiewspostg.postgres.database.azure.com] port=5432 dbname=smbsadb user=smbiepg@smbiewspostg password=... sslmode=require" psql (11.4, server 10.9) SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off) Type "help" for help. smbsadb=> █</pre>
<p>Create a Table</p> <p>Check Initialize.sql file inside the nodejs app code.</p> <p>CREATE TABLE IF NOT EXISTS fact12_bps (code varchar(256) NOT NULL primary key, name varchar(256) NOT NULL, type varchar(1) NOT NULL default 'C', integrated boolean NOT NULL default false);</p>	<pre>smbsadb=> CREATE TABLE IF NOT EXISTS fact12_bps smbsadb-> (code varchar(256) NOT NULL primary key, smbsadb(> name varchar(256) NOT NULL, smbsadb(> type varchar(1) NOT NULL default 'C', smbsadb(> integrated boolean NOT NULL default false); CREATE TABLE smbsadb=> █</pre>
You can run other commands to get details of your database.	<pre>\dt (show tables list) SELECT * FROM smbsadb.fact12_bps;</pre>
Exit	<pre>smbsadb=> exit</pre>

STEP 4: RUN ON SCP CF ENVIRONMENT

Explanation	Screenshot
<p>Go to the terminal and push the app to Cloud Foundry:</p> <p>cf push --random-route</p>	  
<p>Set the environment variables for the core ERP (B1 OR ByD) and for PostgreSQL and Redis.</p>	
<p>You can also set them into the Manifest.yml file but attention to mixing up environment variables defined directly via Cloud Foundry CLI and in the Manifest.yml. In a productive environment, variables must be defined via Cloud Foundry CLI.</p>	
<p>Restart the app after setting the environment variables.</p>	
<p>Look at the logs to find out the error message providing the IP of the app running on CF.</p> <p>After checking the logs, you might need</p>	

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 <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.