Cloud Foundry 12-Factor App
Hands-On Instructions

SAP Run Simple

TABLE OF CONTENTS

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

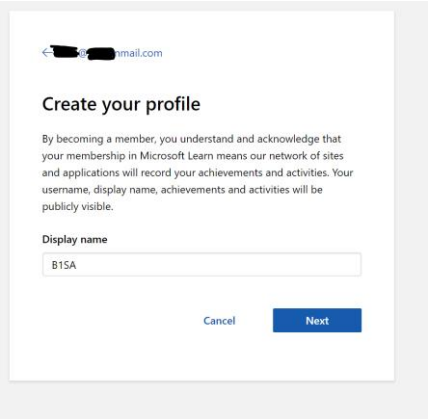
2.2. Create a Microsoft Azure free account
There are at least two options to create a Microsoft Azure Account free of charge for learning purposes and a limited period of time:

- 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.

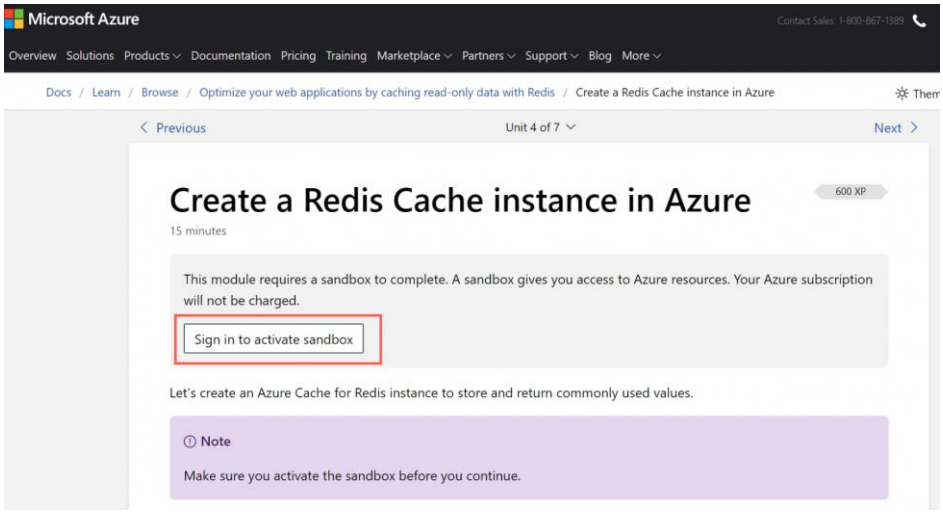
2.2.1.Create an Azure sandbox account for learning purposes

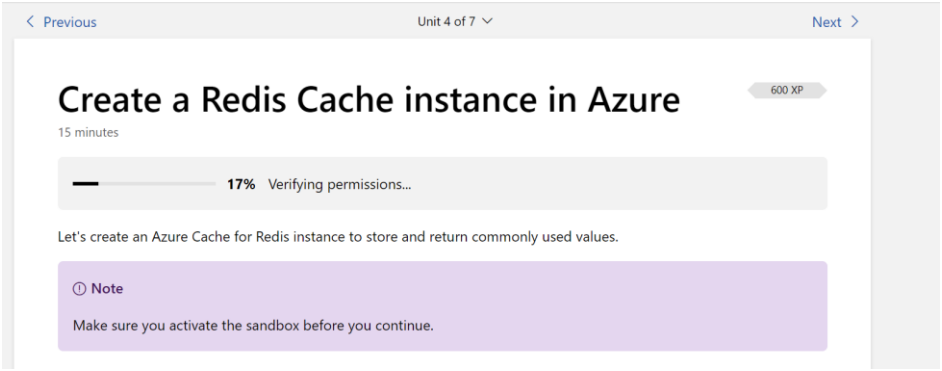
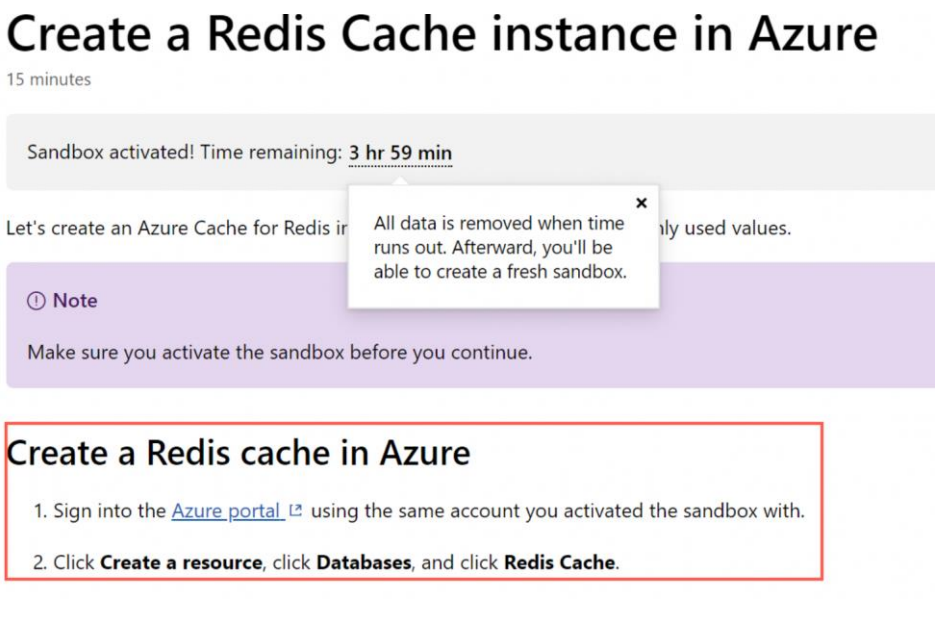
Please follow the steps below to create your ***learn sandbox***. [redacted]@outlook.... MICROSOFT LEARN SANDBOX

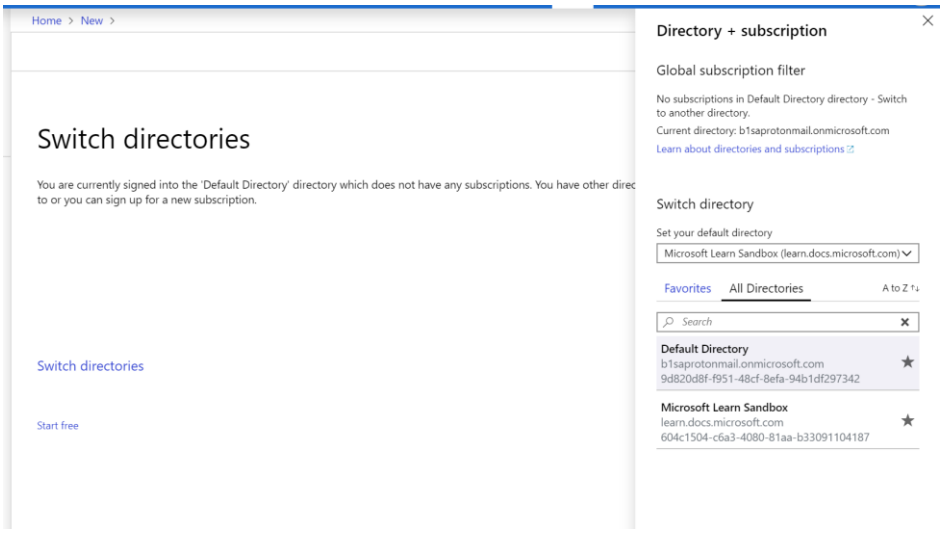
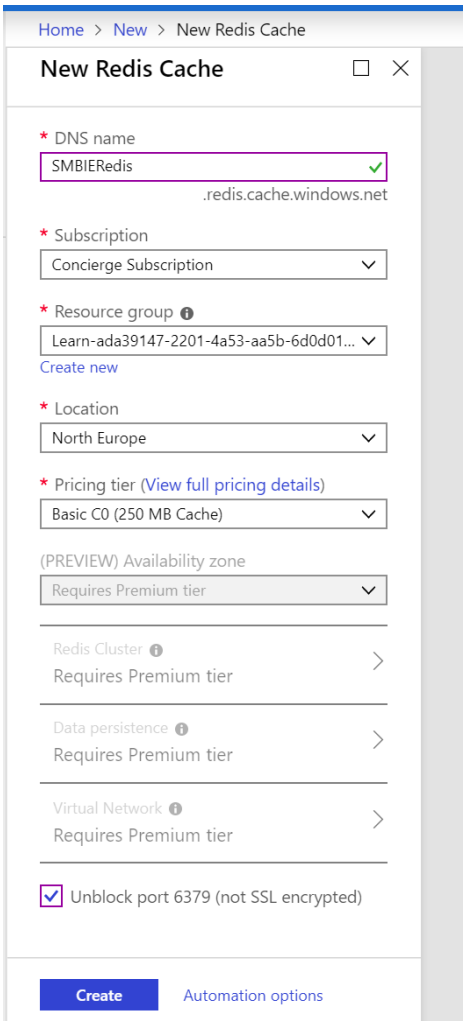
Explanation	Screenshot
Create an account: https://docs.microsoft.com/en-us/learn/azure/?tab=tab-learn-for-free You can use the same email as of your Microsoft account.	

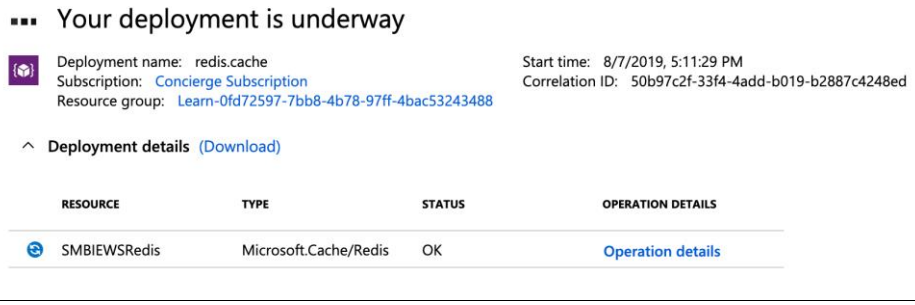
2.3. Create Redis Cache instance

Please follow the steps below to get the learn sandbox running

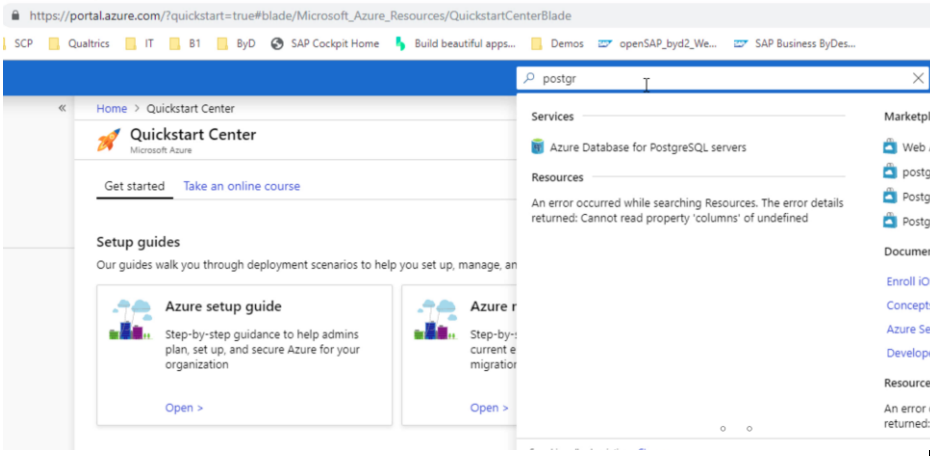
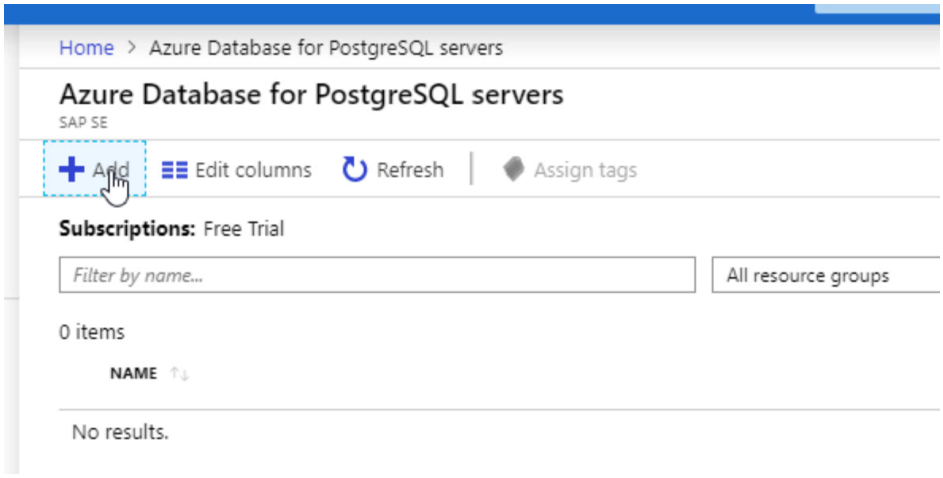
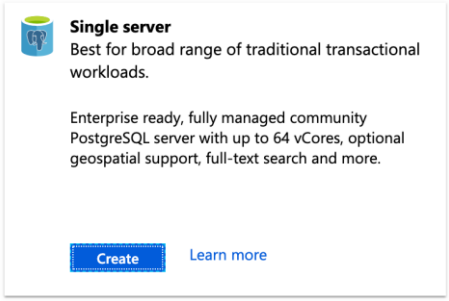
Explanation	Screenshot
Open this exercise tutorial: https://docs.microsoft.com/en-us/learn/modules/optimize-your-web-apps-with-redis/4-exercise-create-redis-cache Press the button "**Sign to activate sandbox**"	

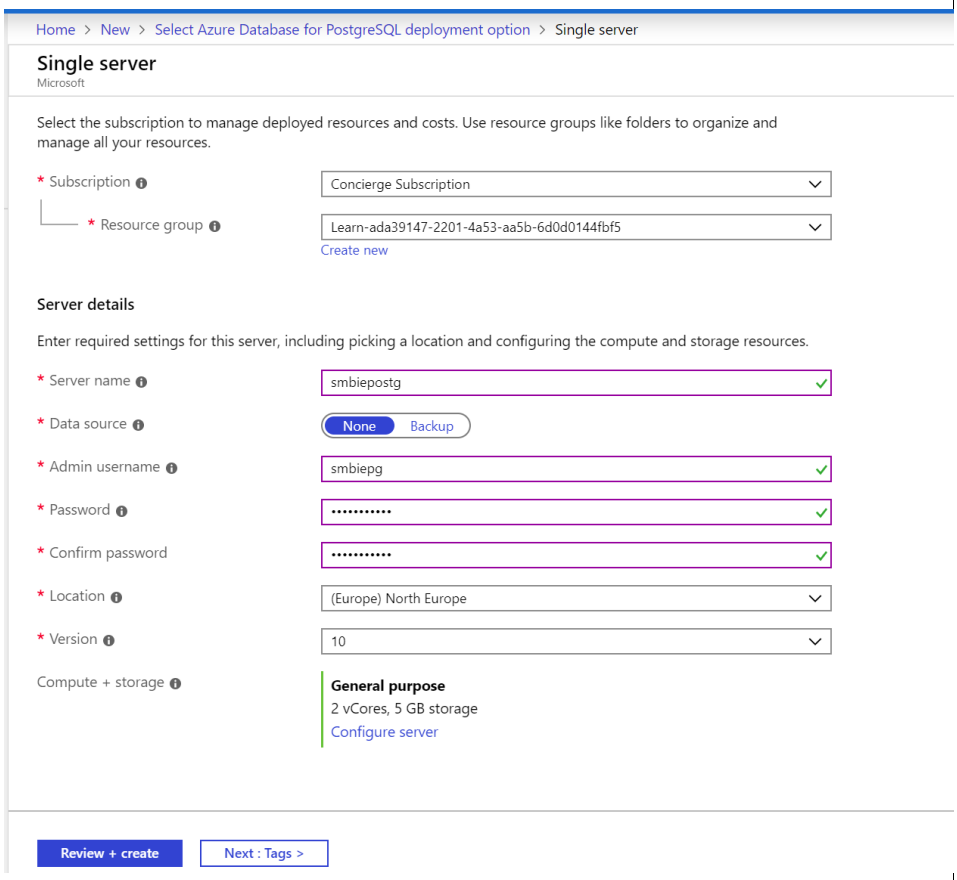
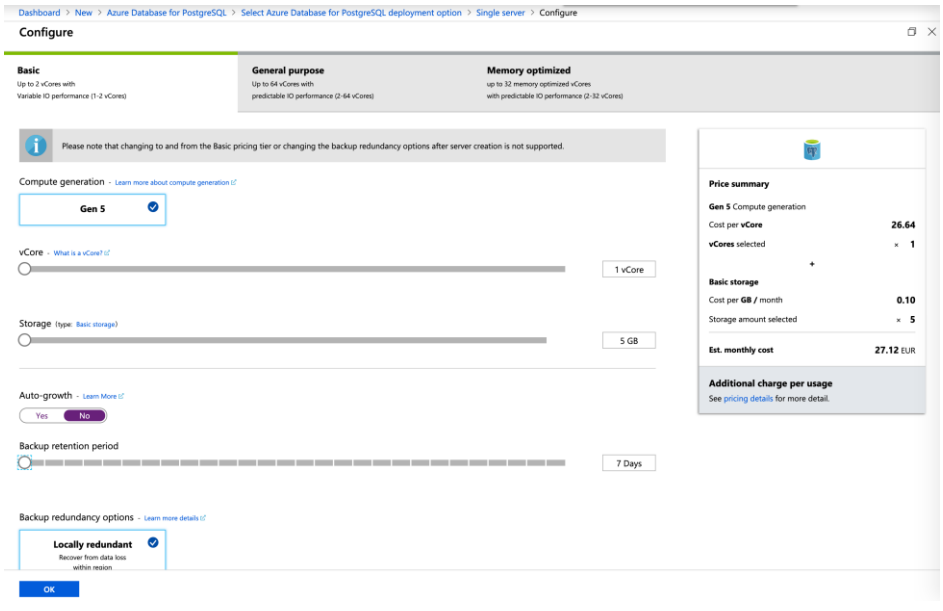
Explanation	Screenshot
Press Activate sandbox.	
The page will show the progress:	
Wait until the sandbox is created. The page will show the sandbox remaining time (4 hours). Follow the steps shown in that page to **Create a Redis cache in Azure**.	

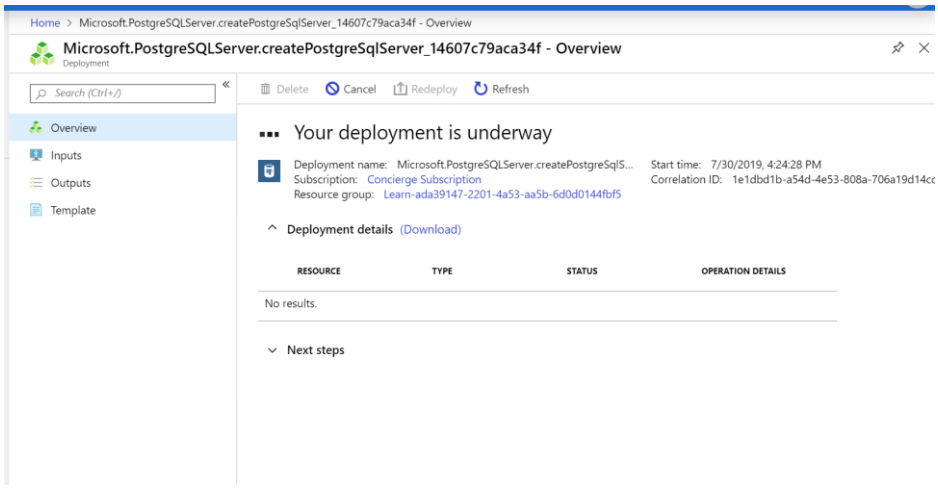
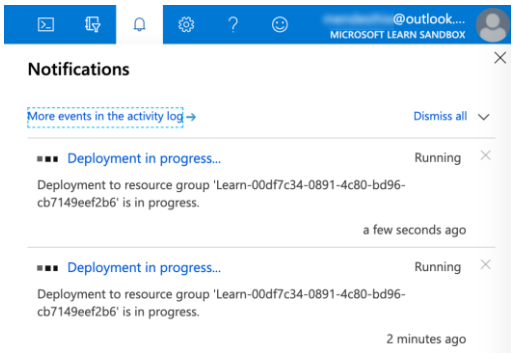
Explanation	Screenshot
While trying to create the Redis cache **IF** you get an error asking you to switch directories just follow the screens and select the "**Microsoft Learn Sandbox**" directory.	
Fill in the creation form as per this screen shot. Change the **Location** to match the nearest to you.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

Explanation	Screenshot
Go to the notifications icon on the top right and check the progress	

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

Explanation	Screenshot
Search for **postgresql** services in your account (directory sandbox).	
Select "**Azure Database for PostgreSQL servers**" and hit **Add**	
Go for the "**Single Server**" setup.	

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

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

STEP 1: PREPARE THE APP ON YOUR COMPUTER

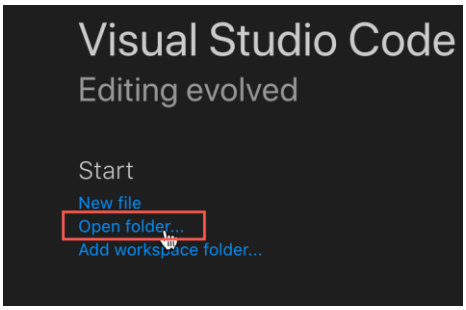
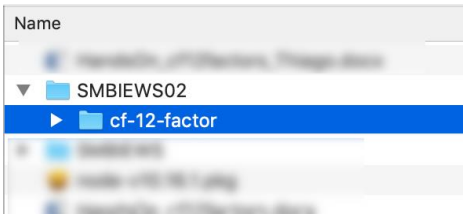
Explanation	Screenshot
Clone the repository $ git clone https://github.com/Ralphive/cf-12-factor.git	[redacted]:Desktop I844173$ mkdir SMBIEWS02 [redacted]:Desktop I844173$ cd SMBIEWS SMBIEWS/ SMBIEWS02/ [redacted]:Desktop I844173$ cd SMBIEWS SMBIEWS/ SMBIEWS02/ [redacted]:Desktop I844173$ cd SMBIEWS02/ [redacted]: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.

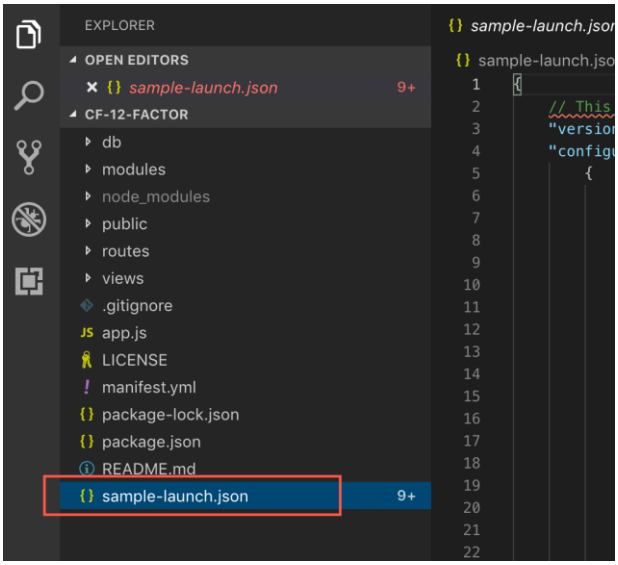
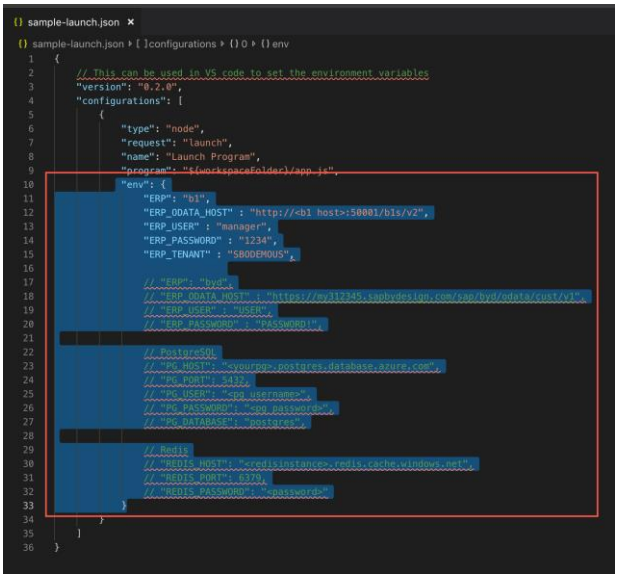
Install the dependencies
 	
$ cd cf-12-factor/
 	
$ npm install
	[redacted]:SMBIEWS02 I844173$ ls
	cf-12-factor
	[redacted]:SMBIEWS02 I844173$ cd cf-12-factor/
	[redacted]:cf-12-factor I844173$ ls
	LICENSE app.js manifest.yml
	package-lock.json public sample-launch.json
	README.md db modules
	package.json routes views
	[redacted]:cf-12-factor I844173$ npm install
	added 127 packages from 105 contributors and audited 238 packages in 1.462s
	found 0 vulnerabilities
	
	[redacted]:cf-12-factor I844173$

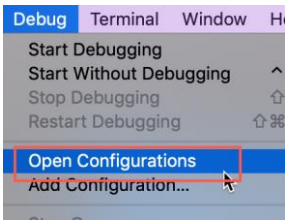
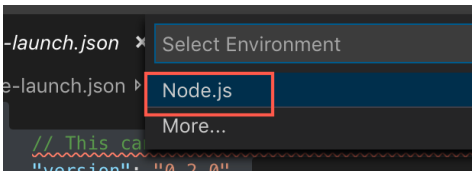
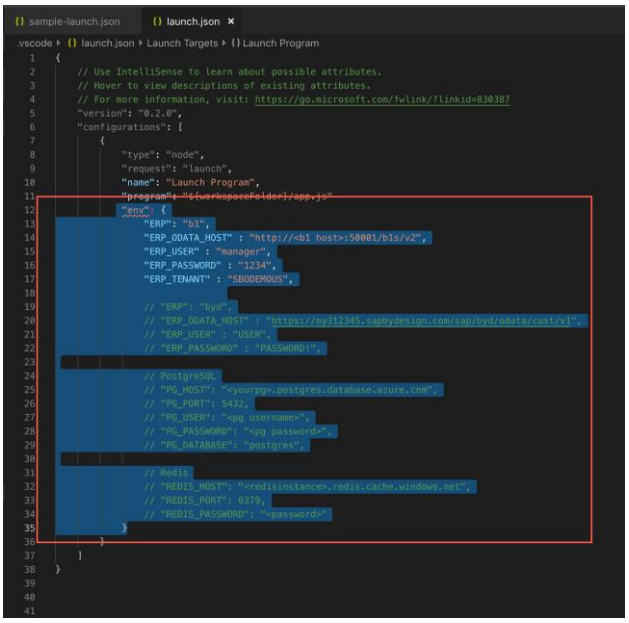
STEP 2: RUN THE APP LOCALLY

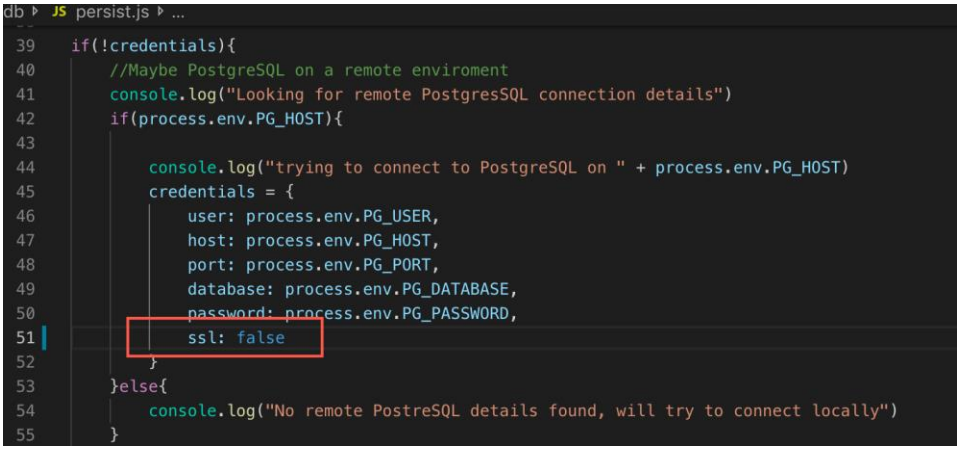
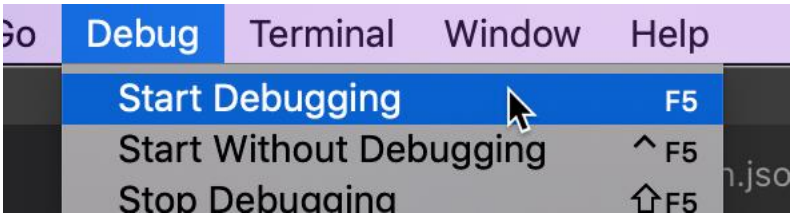
Explanation	Screenshot

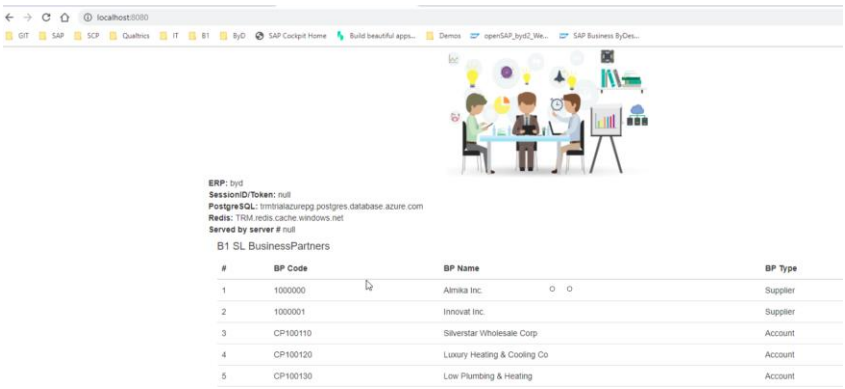
Go to the terminal and run the **psql** command.
Then **create the table** to store items in Postgres:
 	
`CREATE TABLE IF NOT EXISTS fact12_bps`
`(code varchar(256) NOT NULL primary key,`
`name varchar(256) NOT NULL,`
`type varchar(1) NOT NULL default 'C',`
`integrated boolean NOT NULL default false);`
	`[redacted]:~ 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=#`

Open **Visual Studio Code;** Click "**Open folder…**"; And choose the folder with the repository you cloned;	 

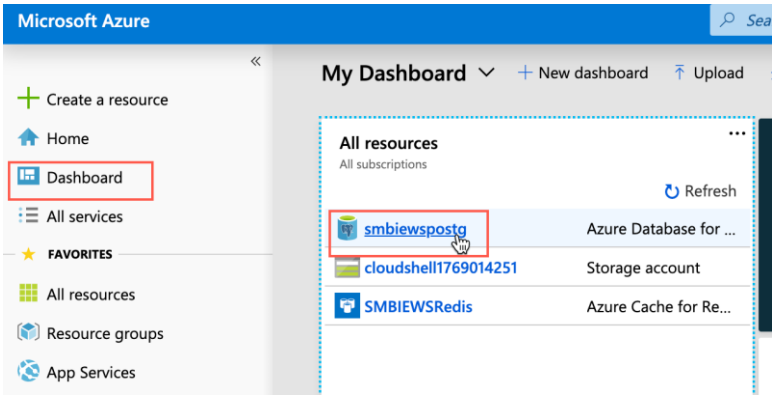
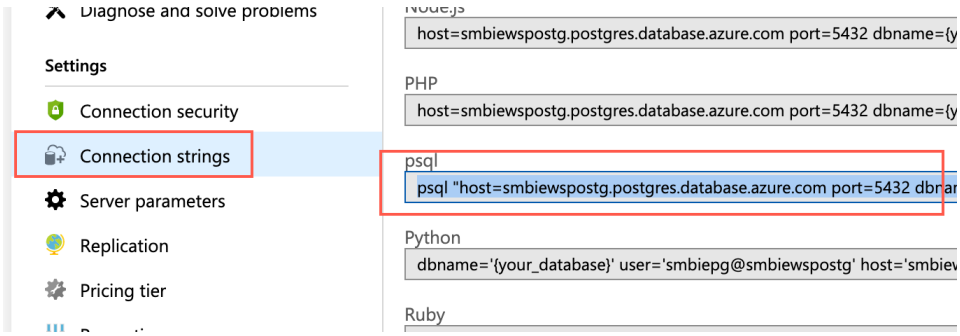
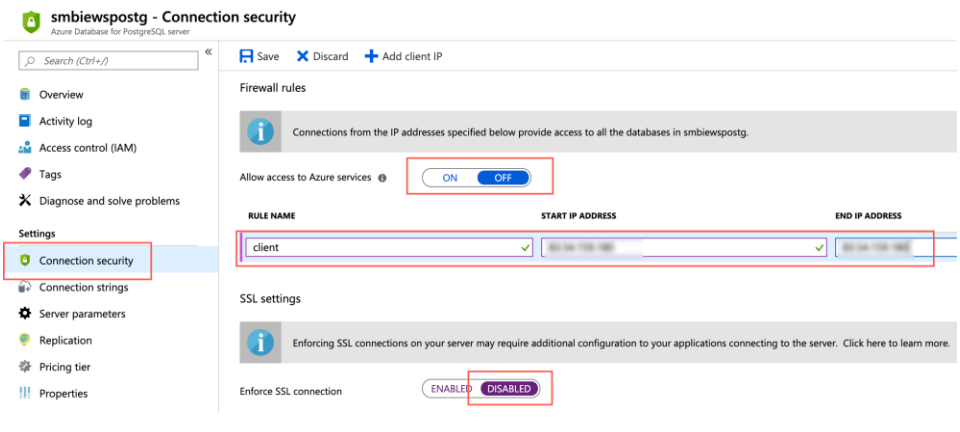
Explanation	Screenshot
Open **sample-launch.json** file; Copy the environment lines;	 

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

Explanation	Screenshot
Customize the variables depending on your environment.	
Open file **db/persist.js** and set the **ssl** to **false**. **Save** it.	
Go to the main menu and select **Debug - Start Debugging**	

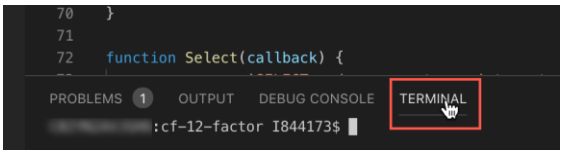
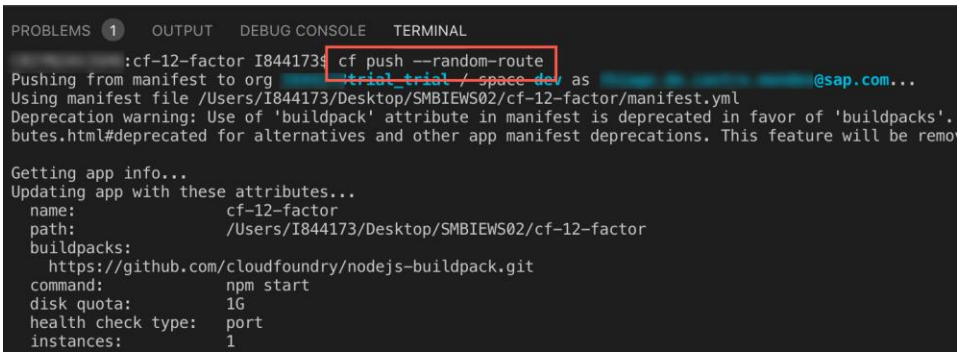
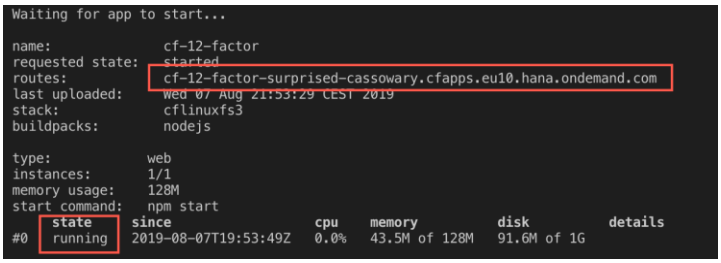
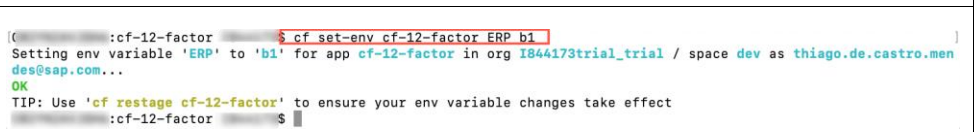
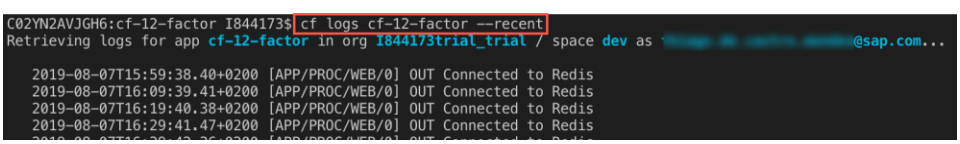
Explanation	Screenshot
Open the page http://localhost:8080 on your browser.	

STEP 3: PREPARE AZURE POSTGRESQL DATABASE AND TABLES

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

Explanation	Screenshot
Run the psql command to connect again.	C02YN2AVJGH6:bin I844173$ ./psql "host=[redacted].postgres.database.azure.com port=5432 dbname=postgres user=smbiepg@smbiewspostg password=[redacted] 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=>
Create a DB create database smbsadb; Exit and connect again, but now to the new DB.	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=[redacted] 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=>
Create a Table Check Initialize.sql file inside the nodejs app code. 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);	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=>
You can run other commands to get details of your database.	\dt (show tables list) SELECT * FROM smbsadb.fact12_bps;
Exit	smbsadb=> exit

STEP 4: RUN ON SCP CF ENVIRONMENT

Explanation	Screenshot
Go to the terminal and **push** the app to Cloud Foundry: **cf push --random-route**	  
Set the environment variables for the core **ERP** (B1 **OR** ByD) and for **PostgreSQL** and **Redis**.	
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.	
Restart the app after setting the environment variables.	
Look at the logs to find out the error message providing **the IP of the app running on CF**. After checking the logs, you might need	

Explanation	Screenshot
to **add the CF IP to the PostgreSQL server Connection Security** as you did with your local IP.	
Restart the app	`$ cf restart <appName>`
Check the app is running on CF	C:\Users\i029162\GitHub\cf-12-factor>cf apps Getting apps in org I029162trial_i029162 / space dev as maria.trinidad.martinez.gea@sap.com... OK name requested state instances memory disk urls leoimg stopped 0/1 256M 1G leoimg-courteous-possum.cfapps.eu10.hana.ondemand.com freightcalctrm stopped 0/1 64M 1G freightcalctrm.cfapps.eu10.hana.ondemand.com smbiotdemocfapptrm stopped 0/1 64M 1G smbiotdemocfapptrm.cfapps.eu10.hana.ondemand.com cf-12-factor started 1/1 128M 1G cf-12-factor-excellent-okapi.cfapps.eu10.hana.ondemand.com C:\Users\i029162\GitHub\cf-12-factor>
Open the app page by using the corresponding generated url.	https://cf-12-factor-excellent-okapi.cfapps.eu10.hana.ondemand.com GIT SAP SCP Qualtrics IT B1 ByD SAP Cockpit Home Build beautiful apps... Demos openSAP_byd2_We...  **ERP:** byd **SessionID/Token:** null **PostgreSQL:** trmtrialazurepg.postgres.database.azure.com **Redis:** TRM.redis.cache.windows.net **Served by server #** 1 B1 SL BusinessPartners | # | BP Code | BP Name | | --- | --- | --- | | 1 | 1000000 | Almika Inc. | | 2 | 1000001 | Innovat Inc. | | 3 | CP100110 | Silverstar Wholesale Corp |

www.sap.com/contactsap