# Learning Containers

## Overview

This section will introduce you to containers, and result in your being able to view geographic data on an interactive map by deploying solution components into “containers” – virtual machines running on your workstation. You will:

1. Create a PostgreSQL database tier server
2. Create an administrative console connected to your database server to view and manage your database
3. Import geo spatial data into your database from Geo-json files
4. Setup a middle tier server that can read your database geo data and provide that data to calling applications via a WMS API
5. Setup an application tier that consumes the WMS API service to enable users to view the geospatial data on a map

These steps will illustrate how four virtual machines running on your desktop will emulate a cloud environment with the same containers hosted on cloud infrastructure.

Application

Interoperability (APIs) 

Figure 1 - Basic 3-Tier Architecture

Administration

Database

To run this exercise, you will need sufficient memory and CPU for reasonable performance. A desktop platform a multi-core processor, 16GB RAM and Windows 10 or higher (iOS may also be used) is recommended.

## Creating your fist container

First create a directory on your workstation where you will hold all the files related to this exercise, and where you will create file systems for use by your containers. An appropriate option would be:

C:\Users\*username*\Container-Demo

A container is a virtual machine running on a real machine or within another container/virtual machine. The software that will be used in this demo is docker desktop. You can download that here:

[Docker Desktop - Docker](https://www.docker.com/products/docker-desktop/)

Install the software and reboot. It is likely you will need to install additional updates to your Windows machine. Complete such updates as prompted. This will include the need to install a unix operating system shell to support docker. This is native for Apple iOS and Windows 11, but not Windows 10. If prompted to do so, install a suitable Unix shell form the list of options. Alternatively, download and install Ubuntu LTS here:

[Ubuntu 18.04 LTS](Ubuntu%2018.04%20LTS)

It is also recommended that you install MS Terminal from the Microsoft store to help manage the command line terminals in the following steps. That can be found here:

[Get Windows Terminal - Microsoft Store en-CA](https://www.microsoft.com/en-ca/p/windows-terminal/9n0dx20hk701?msclkid=c6654367bff311ecb1654ac7acba983f&activetab=pivot:overviewtab)

After installing these software and updates, you are ready to create your first virtual machine:

Open windows PowerShell command window. Change directory to your working director:

PS C:\Users\*username*> cd C:\Users\*username*\Container-Demo

Then start your database container:

PS C:\Users\*username*\Container-Demo> docker-compose -f docker-compose.db.yml up

Docker will automatically download all system components it needs to construct this virtual machine. Then it will start the virtual machine. To verify this process is running, first look at the docker desktop to confirm that your container is up and running.

If that is working, access your database administration product here:

[localhost:5555](localhost:5555/)

Set the master password to be postgres

Add new server

Go to connect and set host to be postgres

Save to connect