

Deploying a Project to Azure

Ashesi Web Development2020

Version 1.0

Overview

This document provides instructions on how to deploy your project to Azure. It includes provisioning the virtual machine and ip address, configuring Apache and PHP, configuring the mysql database. It assumes the following pre-requisites from the team:

- The team have a working project locally on one or more team members machines. It is strongly suggested that all team members can run and host the current project.
- The code that runs the working project is pushed to github.com in the team repository.
- You can create an export file (.sql) of the database that matches your current repository.
- You have accepted the invitations sent over the mid-semester break for your azure subscriptions at your ashesi.edu.gh email address.

Pre-Work

Ensure that the current project can be pulled from GitHub, and the database can be created on a clean local machine or XAMPP installation. If you have not already:

- All team member should **push** their working code for the sprint to GitHub.
- Team members will **pull** the code from those changes and resolve any conflicts.
- The database will be exported to a .sql file. This can be using the mysql utility mysqldump (for example see [this article](#) on a XAMPP installation on windows; by default this is in c:\xampp\mysql\bin).

If you can recreate your app on a local machine; you are ready to deploy to Azure

Creating the VM

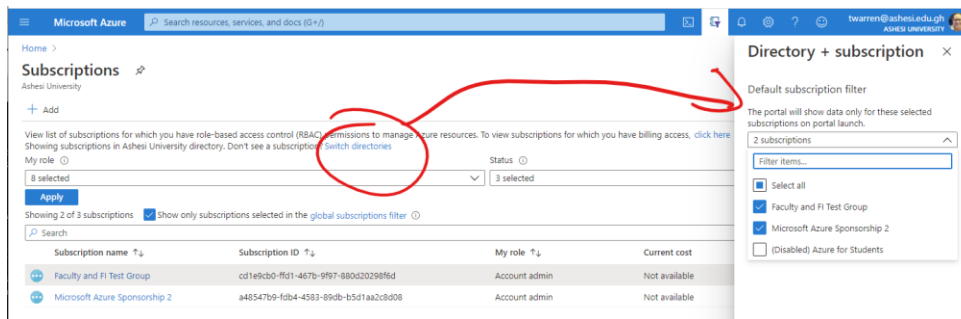
Azure Subscription Groups

Each person on a team should have both an individual allocation of azure credit and a team allocation of credit. You should be able to see these by going to:

https://portal.azure.com/#blade/Microsoft_Azure_Billing/SubscriptionsBlade

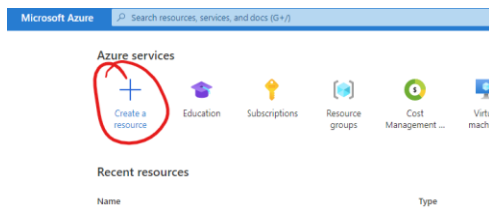
You should see a subscription group that contains your team group letter (will appear as **Group Web Dev Project <team letter>**) and you should also see some credit individually. allocated for the “**Group web dev project <student name>**”. When you create azure resources, you can use either the team letter subscription for your shared team deployment; and the individual subscription for any resource you may want to use to create a personal test environment. These subscription groups are specified when you create resources.

If you don't see the group under subscriptions click “Switch Directories” and you should see the team letter and student letter accounts. Select them so you can allocate new machines to that portal.

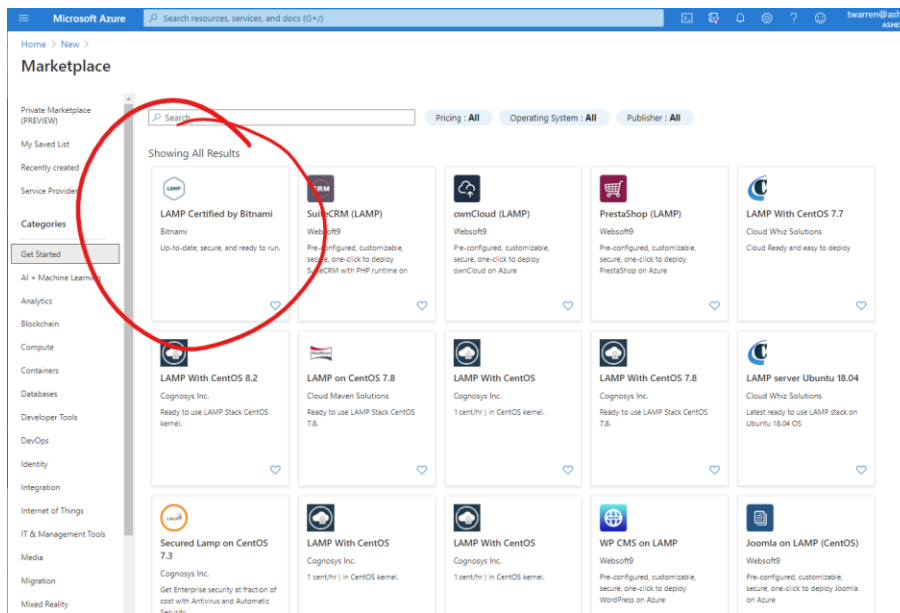


Create a Bitnami LAMP VM from the azure portal

Only one person on the team needs to do this. Create a new VM by clicking “create a resource” from the home screen of <https://port.azure.com>.



We are going to use a pre-configured LAMP stack to save some time---think of it as kind of a XAMPP for azure; so unlike Lab 2 where we created each service individually; it will just be created for us. When you click create resource, type LAMP in the search box. You should see the bitnami LAMP stack:



Click it; and say **create**. You should see a screen like this:

Configure the Virtual Machine

The screenshot shows the 'Create a virtual machine' page in the Microsoft Azure portal. The breadcrumb trail is 'Home > New > Marketplace > LAMP Certified by Bitnami >'. The page title is 'Create a virtual machine'. Below the title is a link 'for full customization. Learn more'. The page is divided into sections: 'Project details', 'Instance details', and 'Administrator account'. Each section contains various configuration options with dropdown menus, text inputs, and radio buttons. Green checkmarks are visible next to several fields, indicating they are valid.

Home > New > Marketplace > LAMP Certified by Bitnami >

Create a virtual machine

for full customization. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Faculty and FI Test Group ✓

Resource group * ⓘ (New) ash-test-11-9 ✓
[Create new](#)

Instance details

Virtual machine name * ⓘ ash-11-9 ✓

Region * ⓘ (Europe) UK South ✓

Availability options ⓘ No infrastructure redundancy required ✓

Image * ⓘ LAMP Certified by Bitnami - Gen1 ✓
[Browse all public and private images](#)

Azure Spot instance ⓘ ☐ Yes ☒ No

Size * ⓘ Standard_B1ls - 1 vcpu, 0.5 GiB memory (\$4.31/month) ✓
[Select size](#)

Administrator account

Authentication type ⓘ ☐ SSH public key ☒ Password

Username * ⓘ ash-test ✓

Password * ⓘ ✓

Confirm password * ⓘ ✓

Please read carefully for each field:

Subscription and Resource Group

Select your group name subscription. (for example, **Group Web Project X**). This ensures that the usage will be billed to your team account; and also that all your other team members will have access to the resource to start and top it. Name the resource group; for example, "sprint-3-X". This is a way of organizing the VM, IP and other Azure resources together for your project.

Instance Details – Virtual Machine Name

Pick a name for your instance. This won't be shown to the user, so something you and your team members can remember.

Instance Details-Spot Instance

We will say **no** to this. This is something that maybe useful for calculation or non-public machines; but since it's a web server we want people to access; no is the answer for now

Instance Details-Region

Internet Routing in West Africa connects to the broader world generally through Europe. The thing that is important is that you can connect to it from where you are; and for this exercise; that it is inexpensive since we don't have a very intensive workload I picked **North Europe** and the Standard_B1ls instance; but UK South was also about the same price.

Authentication Type

As in the lab, pick **password**. This is the simplest for accessing the vm; especially because you will likely share your account with your other team.

Please record the username and password you pick. You will need to share this with your teammates so that you can access the VM once it is started with SSH

Management Tab

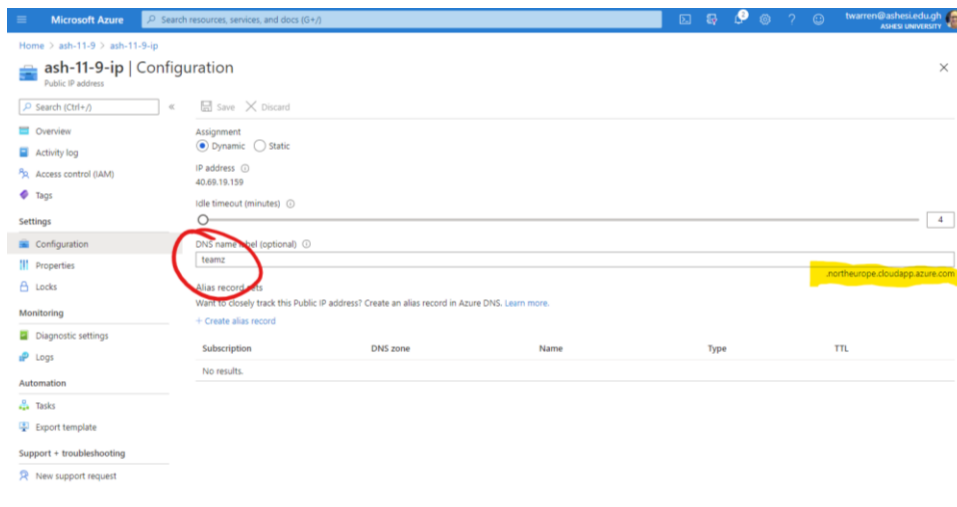
On the management tab you can select auto shutdown

You do not have to set auto-shutdown; but it is a good way to insure you do not accidentally deplete all the funds for your team. On the management tab you can enable automatic shutdown so it will turn off your VM at a particular time.

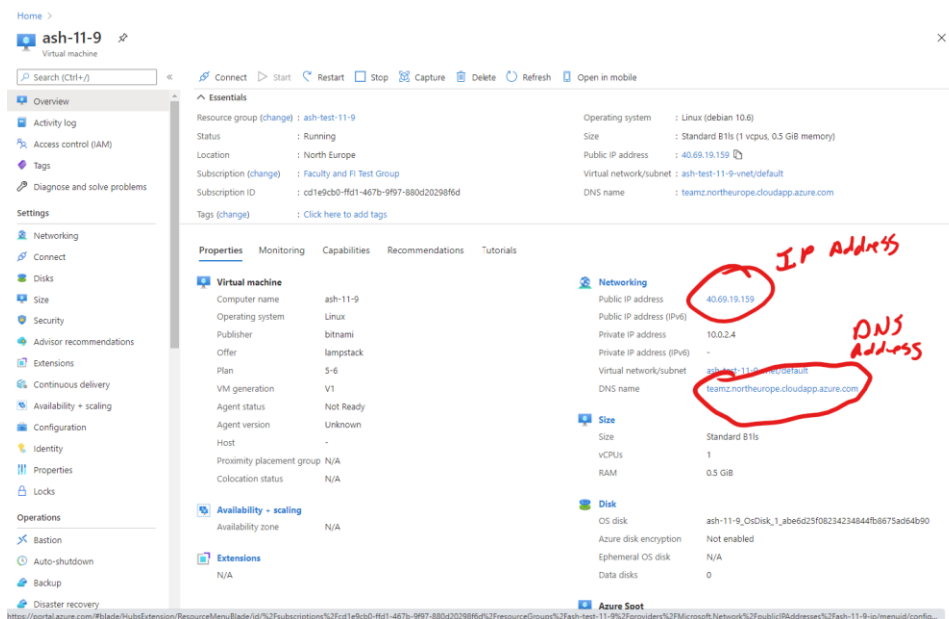
Finally click create!

Configure you IP address/DNS name

By default, the VM is set with a dynamic IP address. Unlike in Lab 2; we will show you how this will work fine. You may recall from early on; the domain name service (DNS) lets you map a name (like ashesi.edu.gh) to an IP address. This will work with the dynamic IP addresses. To configure your DNS name, go to the resource you created; and click the ip address. You will see a screen like this:



Enter a name for your server in the “DNS Name Label” field (circled above). This will then make it so that the dns address **teamz.northeurope.cloudapp.azure.com** maps to the VM’s ip address (in this example 40.69.19.159). When you have saved this; you will see it on the overview of the VM. You may also choose to have a static IP address so that it is always available; but the dns address will accomplish the same thing. On the machine overview you can quickly find both things:



Connecting to the VM with SSH

You can now connect to the VM using the ssh tool. In windows or the Linux bash shell; use ssh to connect, using the username you created in the “Authentication Type” section above. For example, if the user name is **ash-test** for the **teamz** domain name.

In this case the command would look like this:

ssh -l ash-test teamz.northeurope.cloudapp.azure.com

Because we are using a password; and not a secure key; you will need to proceed without having “authenticity”. Say yes at the prompt. Then you will be prompted for the password; and it should connect. You should see a screen like this:

```
toddwseattle@DESKTOP-JRT2HC0:/mnt/c/Users/ToddWarren$ ssh -l ash-test teamz.northeurope.cloudapp.azure.com
The authenticity of host 'teamz.northeurope.cloudapp.azure.com (40.69.19.159)' can't be established.
ECDSA key fingerprint is SHA256:MNx8zFK+DE9ZSRr5brUHjSty6Fc5szXahBvnGYbTyNA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'teamz.northeurope.cloudapp.azure.com,40.69.19.159' (ECDSA) to the list of known hosts.
Password:
Linux ash-11-9 4.19.0-12-cloud-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

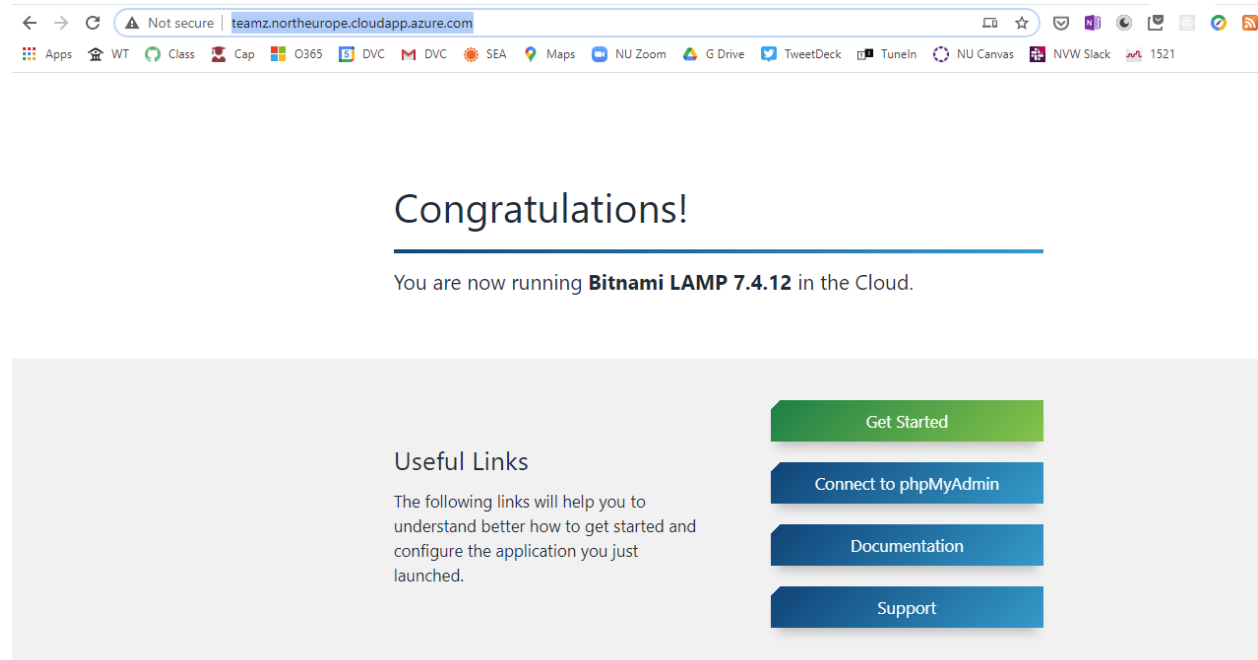
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

  _ _ _ _ _
 | _ | _ | _ | _ | _ | _ |
 | _ | _ | _ | _ | _ | _ |
 | _ | _ | _ | _ | _ | _ |

*** Welcome to the Bitnami LAMP 7.4.12-0 ***
*** Documentation: https://docs.bitnami.com/azure/infrastructure/lamp/ ***
*** https://docs.bitnami.com/azure/ ***
*** Bitnami Forums: https://community.bitnami.com/ ***
bitnami@ash-11-9:~$
```

You are in! now it's time to configure your server.

You should also be able to navigate to the DNS address, in this case <http://teamz.northeurope.cloudapp.azure.com/> and see a screen like this:



← → ↻ Not secure teamz.northeurope.cloudapp.azure.com

Apps WT Class Cap O365 DVC DVC SEA Maps NU Zoom G Drive TweetDeck Tunnl NU Canvas NVW Slack 1521

Congratulations!

You are now running **Bitnami LAMP 7.4.12** in the Cloud.

Useful Links

The following links will help you to understand better how to get started and configure the application you just launched.

- Get Started
- Connect to phpMyAdmin
- Documentation
- Support

If you get here; you are ready to configure your server. Please try to get to this point before Friday Lab. If you get stuck; bring your questions.

Configuring the Server

Understanding the bitnami LAMP server configuration.

The bitnami server is configured a little bit differently than XAMPP or the server we set up in Lab-2. The benefit is like XAMPP; Apache, PHP, and MySQL are already configured. That said, it sets a few things up differently. Specifically:

MySQL Password

When we set up in lab-2 and with XAMPP; we have a blank password for the “root” user in SQL. This isn’t secure for a server open on the internet; so the bitnami configuration creates a password. You can find this password in a file in the home directory. The file is `~/bitnami_credentials`. You can view it with the following command after you have SSH’ed to the server:

```
cat ~/bitnami_credentials.
```

You will see something like this:

```
Welcome to the Bitnami LAMP Stack

*****
The default password is 'P@ssw0rd!'.
*****

You can also use this password to access the databases and any other component the stack includes.

Please refer to https://docs.bitnami.com/ for more details.
```

With your actual password between the single quotes.

Htdocs – where the website goes

By default, the bitnami services are under `/opt/bitnami/apache2/htdocs`.

Other services and configuration are also under `/opt/bitnami`; like `php.ini` (`/opt/bitnami/php/php.ini`) and the apache web server configuration file (`/opt/bitnami/apache2/conf/httpd.conf`).

Other documentation and information

The default webpage links to other documentation about the bitnami configuration. We will delete the default webpage in the next step; so for your reference; this is the link with more information about how this VM is configured: <https://docs.bitnami.com/azure/infrastructure/lamp/get-started/get-started/>

Cloning your GitHub repository to htdocs.

GitHub will only clone not an empty directory. By default; the Bitnami installation has an `index.html`.

- Navigate in the ssh shell to `/opt/bitnami/apache2/htdocs`:
`cd /opt/bitnami/apache2/htdocs`
- Remove `index.html`
`sudo rm index.html`
- Now clone your project to the `htdocs` directory; for example:
`git clone https://github.com/ashweb2020/ashweb200-team-z`

Please note the “.” On the end to clone into the current directory (`/opt/bitnami/apache2/htdocs`).

You should now be able to browse to your website; but it is not hooked up to SQL yet

Configure your .sql file

Copy the .sql file from a local machine to the Azure VM

To move the .sql file from a local machine to your Azure VM; the easiest way is to use a utility built into Windows PowerShell and Unix called SCP. SCP is “Secure Copy” and uses SSH to move files from your machine to a remote machine (or vice versa). Since we have SSH setup on our VM; this is quite easy to do. Open a PowerShell and go to the directory where you have it on your local machine. The format of the SCP command is:

scp <source-file> <destination>

Where **<destination>** or source can specify a file on a remote host (like our Azure VM); with the format **<host-name-and-account>:<path-or-file>**. **<host-name-and-account>** has the format user@host; where the host is an IP address or a DNS address. So, to copy the sql file exported for from mysql (my-database.sql in the example) from our local directory to our azure machine in the root of the home directory; we use our ssh username and the azure dns address:

scp my-database.sql [ash-test@teamz.northeurope.cloudapp.azure.com:~](https://teamz.northeurope.cloudapp.azure.com/~)

This should take you through a series of prompts similar to ssh; where you enter your password; and the file is copied.

Import the .sql file into mysql

Use ssh to connect to your azure machine (see [Connecting to the VM with SSH](#)).

If you copied the file to the home directory like the example scp command; you should be able to see it there:

cd ~

ls *.sql

Then, you need to import it into mysql. You will need the password for your bitnami installation (See [MySQL Password](#)). To import your database; use the following command:

mysql -u root -p *bitnami-password* < *database-file*.

Replace the italicized words with the password and file name you are working with.

You should now be able to see your database. You can verify it is there by launching mysql from the command line with the root user and bitnami-password and executing sql commands.

Connecting PHP to MYSQL: Password Tradeoffs

Likely, your local password to MYSQL and the bitnami-password are not the same. There are several options:

- You can change the root password (this is harder than you think), for example to no password as XAMPP installs by default; however, it's a big security flaw to have a blank password.
- You could modify the source code on each installation to match the current password of the server. This is ok; but it turns out, it's a bad idea to store passwords in source code; if it's stored on git, it

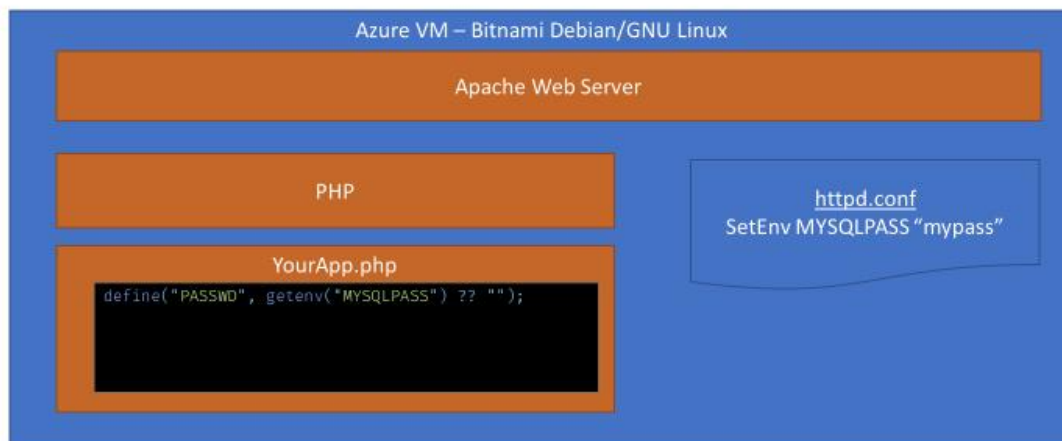
could get cloned and a hacker can break your database. Indeed, hackers often crawl sites like GitHub looking for passwords; and while your repository is private, source code often gets shared...

It turns out there are many ways to secure passwords and credentials for a PHP application; all with lots of different tradeoffs. You can [read this article for a current discussion of the broader set of options](#).

I'm going to suggest a method that is a variant of what the article calls the "Apache Pull" method; where the credentials are stored in an environment variable in the web server; then in our source code, we look for whether that environment variable is set; and if it isn't set; we assume the SQL user is *root* with a blank password. This way the code will work fine on a default XAMPP installation; and on an Azure server; or a developer can have a different root password and change their configuration, so the application runs:



Passing an Environment Variable to PHP



So on the server, Apache is configured in a file named **httpd.conf**. This is the file in `/opt/bitnami/conf/httpd.conf`. An environment variable can be set and passed to PHP by using the Apache Configuration command `SetEnv`.

Our PHP program can read environment variables, using the PHP function **getenv()**.

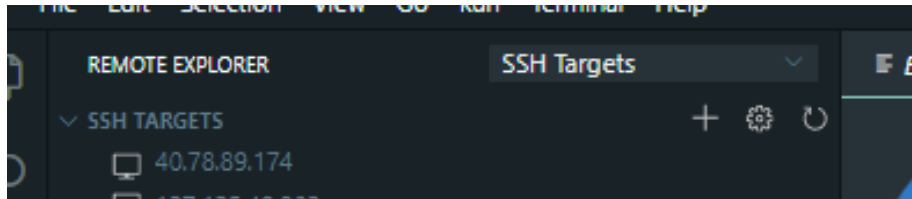
We want to set it up so that we can pass the SQL password in the environment variable; and if it's not there, we can have our code just try a blank password for root. The code to set a constant `PASSWD` like

this is by defining a constant included in our database connection. We can do that by using define; as in the diagram.

How to access and edit files on the VM server with VS code

VSCode will let you ssh to your machine and open and edit files. This article goes through the steps to add the ssh remote extensions: <https://code.visualstudio.com/docs/remote/ssh-tutorial>

After the extension is installed in vscode; you can use it to connect to your vm. Click the “+” by ssh targets:



Then type the same ssh command used to get to your vm; for example in our case **ssh -l ash-test teamz.northeurope.cloudapp.azure.com** Save it in your local user settings; and you can connect to this server.

You can then open the path to `/opt/bitnami/conf/httpd.conf` and edit the file to add the environment variable.