VMware Horizon on Nutanix

Reference Architecture

v1.0 – February 2013

Graphical API Demo

Requirements & How-To Instructions

### Requirements

To run the SE Team’s Graphical API Demo on a local workstation you’ll need an OS X or Windows workstation capable of operating as a basic development environment. The easiest way to do this without manual configuration is to download and install MAMP (Mac/Apache/MySQL/PHP) from <https://www.mamp.info/en/downloads/>.

Notes for Windows users:

* .NET Framework 4.0 and a number of other Windows updates are required to install MAMP, although the MAMP installer can do them all for you
* The ‘MAMP PRO’ component requires a license and is optional – it can be safely deselected during installation

To get started:

* Using the ‘Download ZIP’ button, download the source repository from <https://github.com/digitalformula/public-scripts/>
* Extract the ZIP file to a location that makes sense e.g. ~/Documents/Demos on OS X or ‘<Home>\Documents\Demos’ on Windows

Once installed, refer to the section below for further configuration. These steps should only take around 2 minutes.

### Option 1 – Use MAMP Application’s Web Server (Easiest)

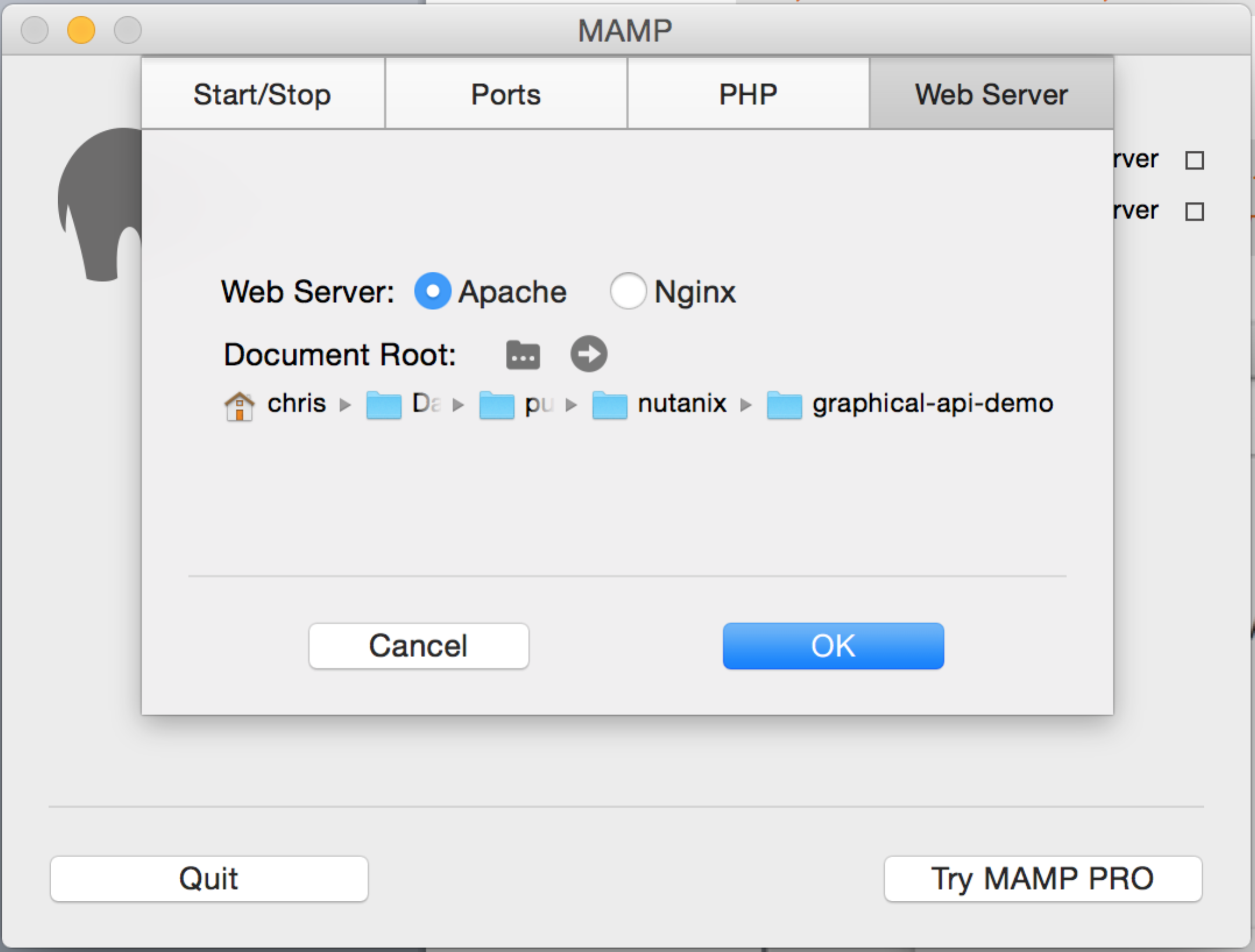
* Open MAMP from the Applications folder (OS X) or Start button > All Programs (Windows)
* Select Preferences, open the Web Server tab and set the home directory to the ‘graphical-api-demo’ folder. Example paths are shown below - change the path to match your system, if necessary.

OS X:

|  |
| --- |
| *~/Documents/Demos/public-scripts-master/nutanix/graphical-api-demo* |

Windows 7:

|  |
| --- |
| *C:\Users\<Username>\Documents\Demos\public-scripts-master\nutanix\graphical-api-demo* |



**Fig 1. MAMP screenshot during Web Server root configuration (OS X)**

* Click ‘OK’
* Click ‘Start Servers’
* Browse <http://localhost:8888> (OS X) or <http://localhost:80> (Windows). If required, opening the ‘Ports’ tab after selecting ‘Preferences’ will allow modification of the Apache web server port.

At this point you should have an environment that can successfully run the API demo.

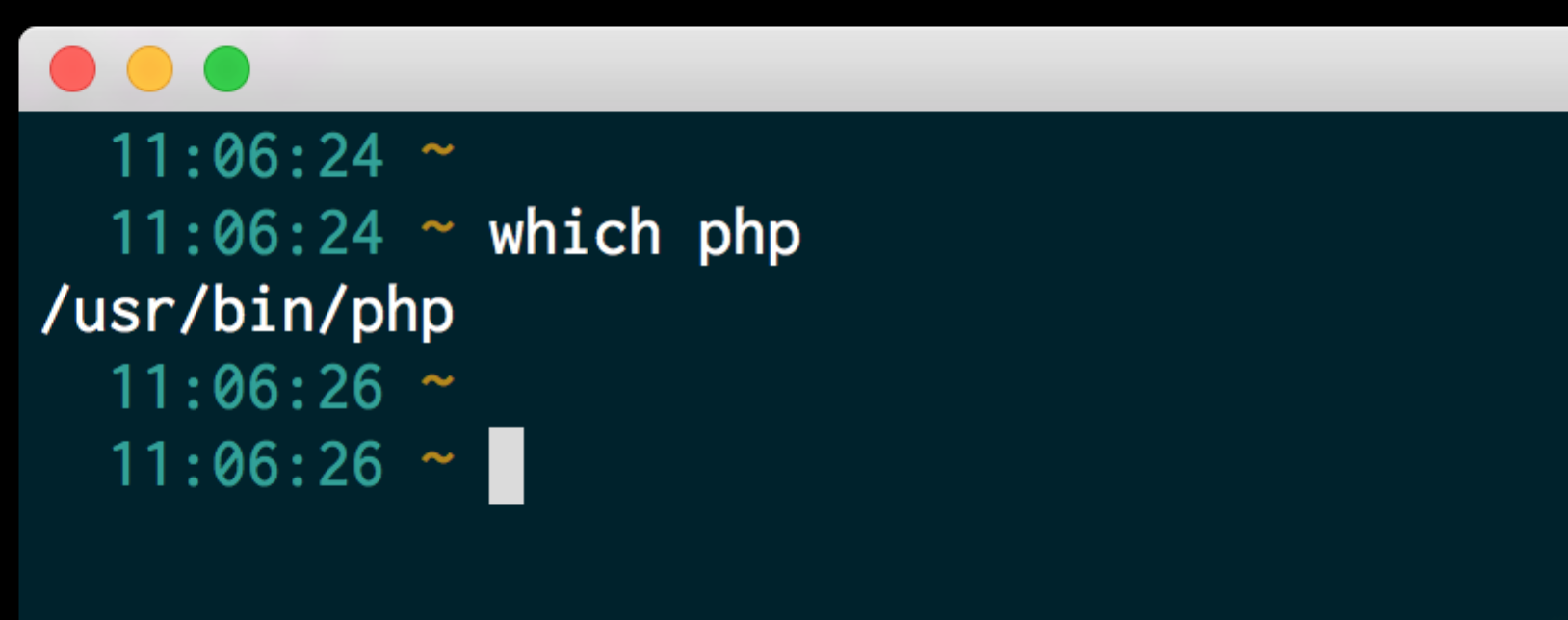
### Option 2 – MAMP binaries + OS X Command Line (for CamO)

Why have option 2? I’ve found MAMP can use more resources than I want it to, partly due to the default install also starting a MySQL server. Yes, that can be disabled, although the process is outside the scope of this document. Running the demo from command line is slightly lighter and can be useful if you are running a Windows or Foundation VM at the same time.

Open a terminal session or command prompt and enter the following command.

|  |
| --- |
| *> which php* |

This will return the path to the PHP binary file, probably similar to the following:



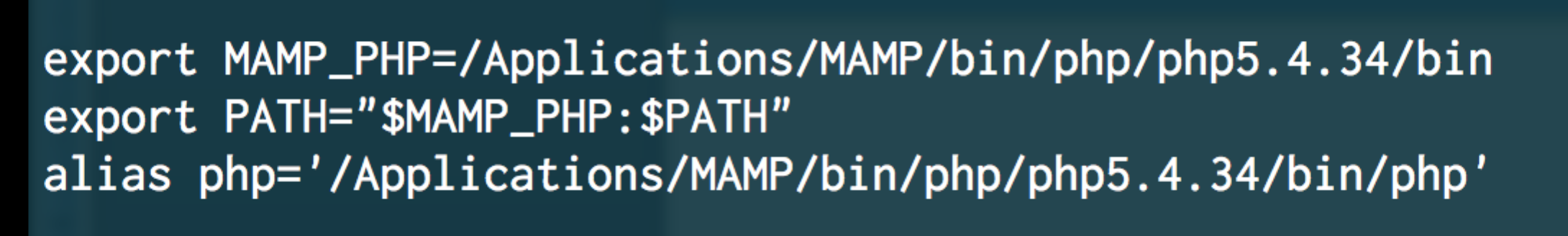
**Fig 2. Default PHP binary path for OS X**

If the default path is /usr/bin/php, you’ll need to change the PHP binary path so that it uses the PHP binary provided by MAMP (installed earlier).

On unmodified OS X systems you’ll likely be using the system default shell (Bash). Using your favourite editor e.g. nano, Emacs if you are Carlo or vi if you are Cam Stockwell ☺, create/edit the following file:

* ~/.bashrc for unmodified systems, as above
* ~/.zshrc if you are using Z shell (e.g. if you have installed Oh My Zsh)

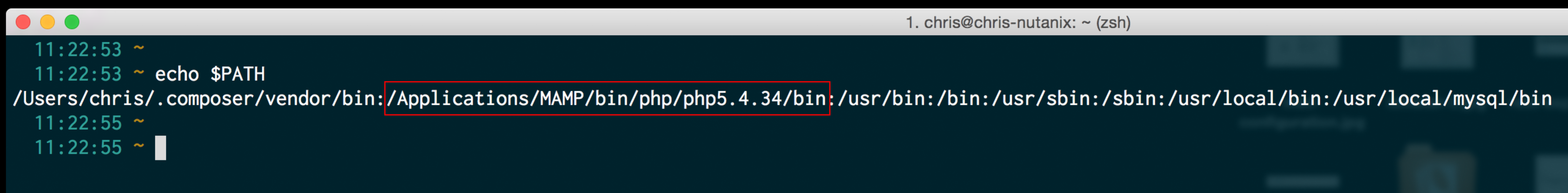
In addition, an alias has been specified so that running ‘php’ uses the correct binary. These lines must be placed below all other lines that set/modify the $PATH variable.



**Fig 3. Lines to be added to ~/.bashrc or ~/.zshrc**

Once this is done, close the Terminal session, open a new Terminal session (so the changes take effect) and check the $PATH variable, making sure the MAMP path specified above is present.

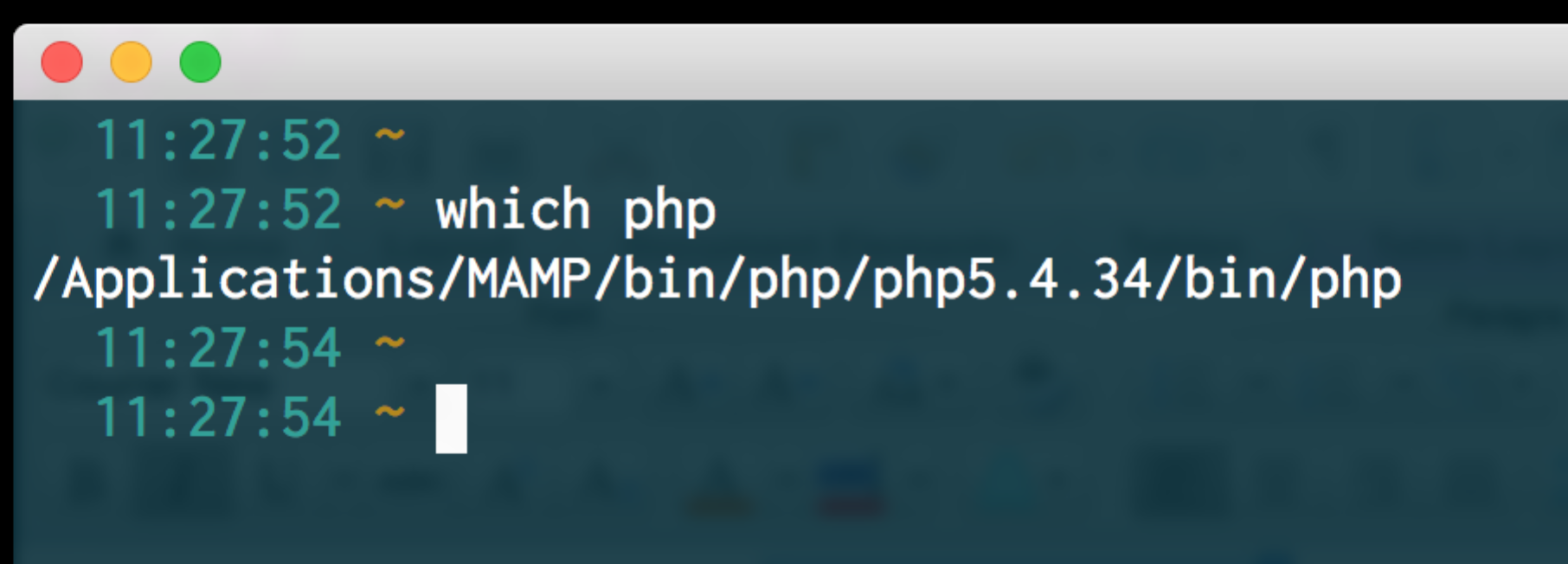
|  |
| --- |
| *> echo $PATH* |



**Fig 4. Updated $PATH shown in Terminal window (OS X)**

Finally, verify the PHP binary path has been set.

|  |
| --- |
| *> which php* |



**Fig 5. Updated PHP binary showing in Terminal window (OS X)**

To run the demo, you can now run the following commands – this assumes the demo files are in ~/Documents/Demos

|  |
| --- |
| *> cd ~/Documents/Demos/public-scripts-master/nutanix/graphical-api-demo*  *> php –S localhost:8000* |

### Additional Notes

If anyone viewing the demo wants to see the specific source code that carries out the API request, please open the following file. All other files in the demo are for rendering the demo’s main page and processing the results returned by the API request. As with previous examples, this assumes the demo files are in ~/Documents/Demos on OS X.

|  |
| --- |
| *~/Documents/Demos/public-scripts-master/nutanix/graphical-api-demo/model/apiRequest.php* |

At this point there is an assumption that anyone viewing the *doAPIRequest()* method understands the usage of *$this* …

*/\*\*  
 \* Process the API request based on the current apiRequest instance  
 \*  
 \** ***@return*** *mixed  
 \*/***public function** doAPIRequest()  
{  
  
 $client = **new** GuzzleHttp\Client();  
  
 $response = $client->get(  
 sprintf( "https://%s:%s%s",  
 $this->cvmAddress,  
 $this->cvmPort,  
 $this->requestPath  
 ),  
 [  
 'config' => [  
 'curl' => [  
 *CURLOPT\_HTTPAUTH* => *CURLAUTH\_BASIC*,  
 *CURLOPT\_USERPWD* => $this->username . ':' . $this->password,  
 *CURLOPT\_SSL\_VERIFYHOST* => **false**,  
 *CURLOPT\_SSL\_VERIFYPEER* => **false** ],  
 'headers' => [  
 'Accept' => 'application/json'  
 ],  
 'verify' => **false**,  
 'timeout' => $this->connectionTimeout,  
 'connect\_timeout' => $this->connectionTimeout,  
 ]  
 ]  
 );  
  
 /\* return the response data in JSON format \*/  
 **return**( $response->json() );  
  
}