WGA2

Static Site using Bootstrap 4, Gulp, Serverpilot and Digital Ocean

[Create a new droplet and Generate SSH Keys 3](#_Toc512960480)

[Setup an SSH Session in Putty 3](#_Toc512960481)

[ServerPilot 3](#_Toc512960482)

[Login as System User 4](#_Toc512960483)

[SSH Commands 4](#_Toc512960484)

[**Install Bootstrap 4 Sass, Gulp, DevDependencies and Create Folder Structure** 4](#_Toc512960485)

[Setup Git with SSH Keys on Local 7](#_Toc512960486)

[**Setup Git on Local** 7](#_Toc512960487)

[Setup Git on Server 8](#_Toc512960488)

[Setup Git Webhook 9](#_Toc512960489)

[Secure the Hook 10](#_Toc512960490)

# Create a new droplet and Generate SSH Keys

Select Ubuntu 16.04.4 \*64, 1GB - $5mth, Singapore.

New SSH Key – In PuttyGen, change the number of bits to 4096, click on Generate and wave the mouse around in the window area.

Copy the public key and paste it into DigitalOcean.

Name the ssh key – Desktop.

Name the droplet host name – wga2.

Save the public and private keys.

# Setup an SSH Session in Putty

In Putty, enter the IP address of the droplet, Port 22, Connection type: SSH.

Under Connection select Data, Auto-login username: root.

Under Connection select SSH, then Auth and Browse for the private key and select Open.

Under Session > Saved Sessions enter wga2 and select Save.

Now when connecting with the server, under Session > Saved Sessions select the session, select Load then select Open.

# ServerPilot

Enter the ip address of wga2, then click on ‘I don’t have a root password or…’

Hostname – wga2.

SFTP Password – 7#b3WedT. (Changed to 83#B!Qr9)

Select ‘Enable SSH password auth…’

Connect to ServerPilot.

Copy the script and paste it into the server.

Create App.

Name – wga2.

Domain – 167.99.73.134

PHP 7.2

Server – wga2.

System User – serverpilot.

Create App.

## Login as System User

Exit out of Putty, in Putty enter the ip address of wga2 (167.99.73.134), in Putty cmd enter serverpilot then the password – 83#B!Qr9

# SSH Commands

List the files: $ll.

Change directories: $cd apps.

Delete directories: $rm –rf foldername/

Copy files: $cp filename newfilename.

New file: $vi filename.

Save file and exit: esc :wq

Open a file and edit: vim filename.

**Install Bootstrap 4 Sass, Gulp, DevDependencies and Create Folder Structure**

In Desktop create a folder named wga2

$cd into wga2 and run $npm init

Change the main entry js to “app.js”

In wga2 run $npm install bootstrap - -save

In wga2 run $npm install gulp gulp-sass gulp-clean-css gulp-autoprefixer gulp-sourcemaps gulp-html-minifier gulp-rename --save-dev (run $npm init again if the saves don’t work) (browser-sync isn’t accepting styles, so forget it)

In wga2 root create a folder named src, inside src create a folder named scss

In node\_modules > bootstrap > scss copy bootstrap.scss and paste it into src > scss and rename it main.scss

Add the new path to all the scss files, comment them all out and uncomment the ones required

In src add a file named index.html with doctype and an h1, just so we can see if gulp works we’ll have content in the browser. Add the link to the main.min.css

In main.scss add a style for the h1

In wga2 root create a folder named docs and keep the word doc in there.

Once the public folder is created using Gulp, add an index.php file with this script:



**Setup the gulpfile**



# Setup Git with SSH Keys on Local

In a new Git Bash, enter:

**$ssh-keygen -t rsa -b 4096 -C "pikej307@gmail.com"**

When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.

Add the private key to the agent with:

**$eval $(ssh-agent -s)**

**$ssh-add ~/.ssh/id\_rsa**

Copy the public key to clipboard with:

**clip < ~/.ssh/id\_rsa.pub**

In the upper-right corner of any page, click your profile photo, then click Settings.

In the user settings sidebar, click SSH and GPG keys.

Click New SSH key or Add SSH key.

In the "Title" field, add a descriptive label for the new key. For example, if you're using a personal Mac, you might call this key "Personal MacBook Air".

Paste your key into the "Key" field.

Click Add SSH key.

**Setup Git on Local**

Add a .gitignore file with cancelling out node\_modules and the package-lock.json file.

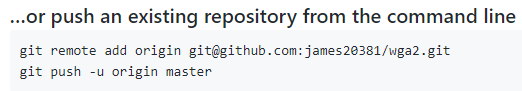
In Git Bash **$cd** into wga2 project folder and run **$git init**.

**$git add** .

**$git commit –m “Initial commit of wga2”**

Sign in to Github, click on New Repository, name it wga2.

Making sure the SSH switch is on, not the HTTPS, copy:



Enter it into the bash and press enter.

**$git push origin master** (don’t use the -u)?

Are you sure you want to continue connecting? **Type in yes**, then enter.

# Setup Git on Server

In Putty, inside wga2 delete the public folder and run $**ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"**

Press enter to save the key where prompted (/srv/users/serverpilot/.ssh/id\_rsa)

Run **$eval $(ssh-agent -s)**

Run **$ssh-add ~/.ssh/id\_rsa**

Run **$cat ~/.ssh/id\_rsa.pub**

Copy the key.

In the upper-right corner of any page, click your profile photo, then click Settings.

In the user settings sidebar, click SSH and GPG keys.

Click New SSH key or Add SSH key.

In the "Title" field, add a descriptive label for the new key. For example, if you're using a personal Mac, you might call this key "Personal MacBook Air".

Paste your key into the "Key" field.

Click Add SSH key.

Make sure to be in the apps/wga2 folder and run **$git init**.

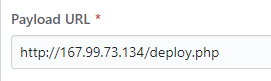
Making sure SSH is set, clone the url into Putty, inside the wga2/public folder. **$git remote add origin git@github.com:james20381/wga2.git**

Run **$git pull origin master**

# Setup Git Webhook

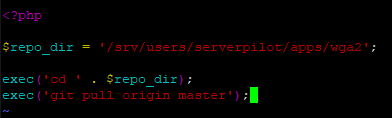
From the wga2 repo at Github, go to Settings > Webhooks > Add Webhook

In Payload URL enter the wga2 ip address/deploy.php:

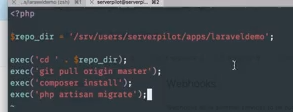


Click on Add webhook.

In Putty > apps > wga2 > public run $vi deploy.php and enter the following script:



Laravel:



Add deploy.php to the .gitignore file using /public/deploy.php

Run $git add –A

Run $git commit –m “Update .gitignore”

Run $git push origin master.

In Git Bash run $git pull origin master.

Change something in the project and run gulp to pass it through to the public folder.

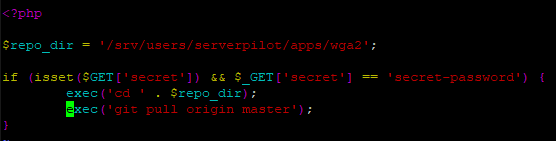
In Git Bash run $git add –A or ., git commit and git push origin master.

Reload the browser to test if the change went through.

# Secure the Hook

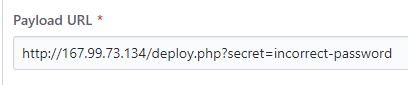
In Putty as serverpilot, go to apps/wga2/public and run $vim deploy.php

Add this if statement to the hook script:



Run $esc :wq to save and exit.

In Github add the hook password script to the hook, with an incorrect password so we can test it:



Make a change locally, push it to test it.

Now enter the proper password into the Github hook and reload the browser, the change should go through.