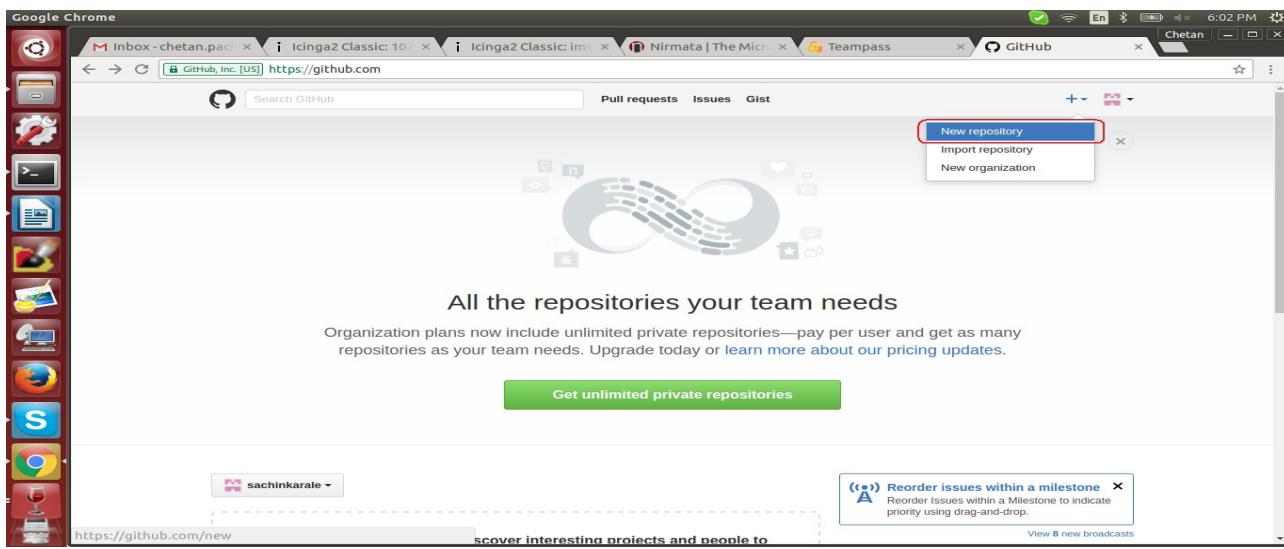


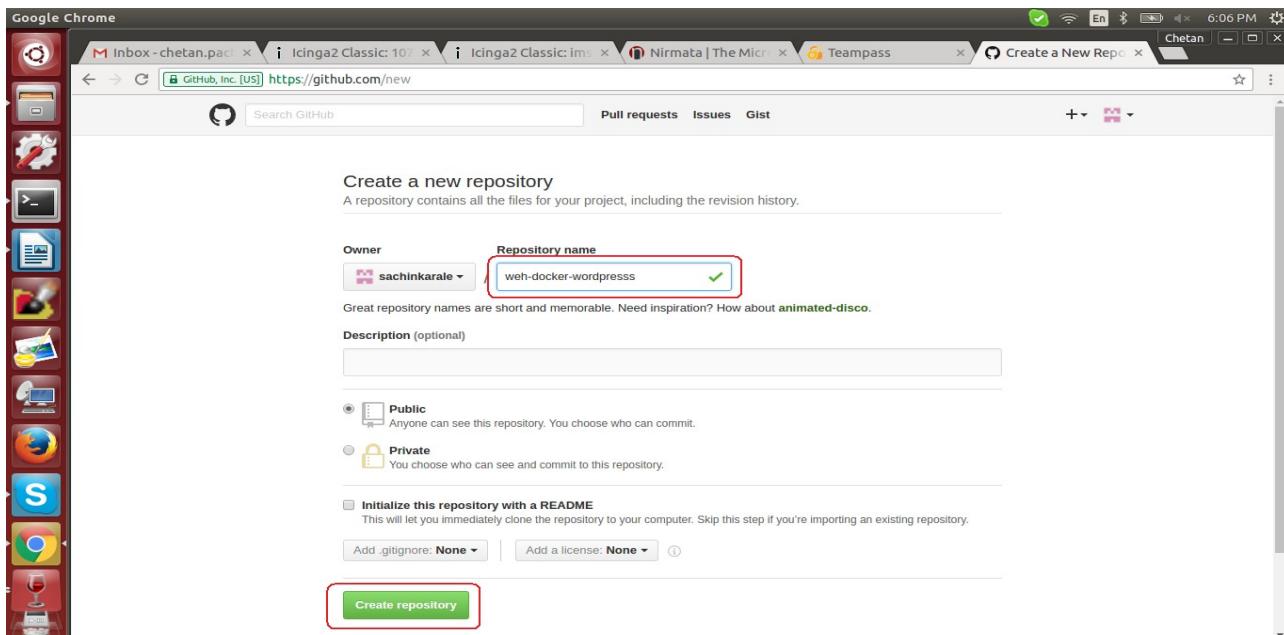
WordPress Documentation : -

1) Creating Repository & Branch in Github Account.

- First step is ,Login into your github account.
- Next ,Create New Repository as shown below fig.



- Give proper name to your newly created repository. In Our case we are using name "**"weh-docker-wordpress"** repository for File System.



C . Next step is that, we have to push all wordpress directory contents to our newly created repository excluding wp-config.php file. This repository will work as FileSystem for us. Meaning is that whatever plugins we will add that will be stored here.

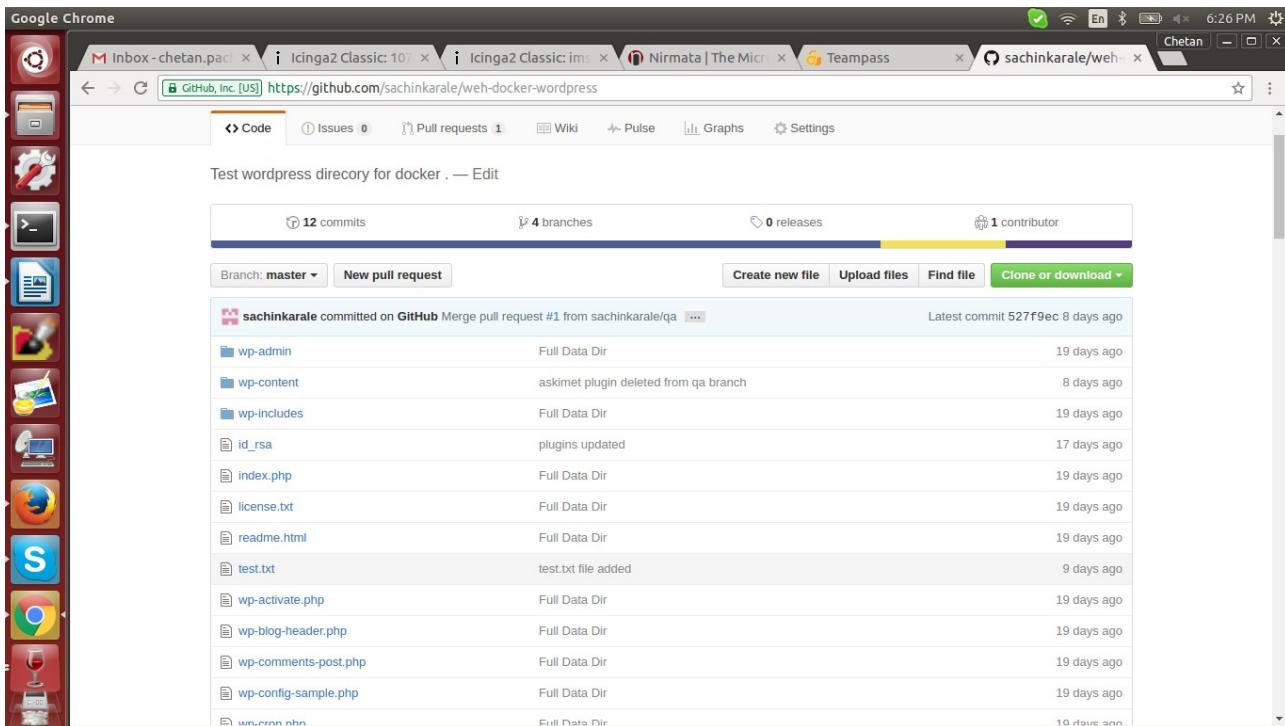
We will push wordpress directories contents to repository using following command.

```
# git clone https://github.com/sachinkarale/weh-docker-wordpress.git
# cd weh-docker-wordpress
```

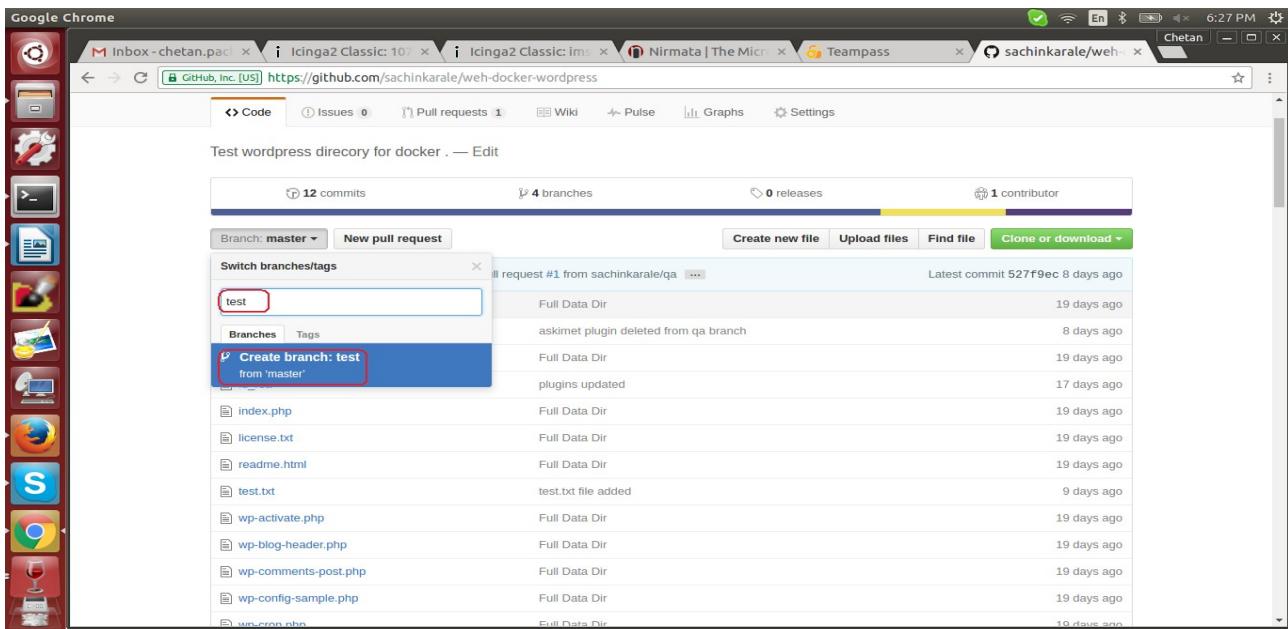
Now, Copy wordpress directories contents to this directories.

```
# cp /opt/wordpress/ .
# git add *
# git commit -a -m "some message"
# git push -u origin master
```

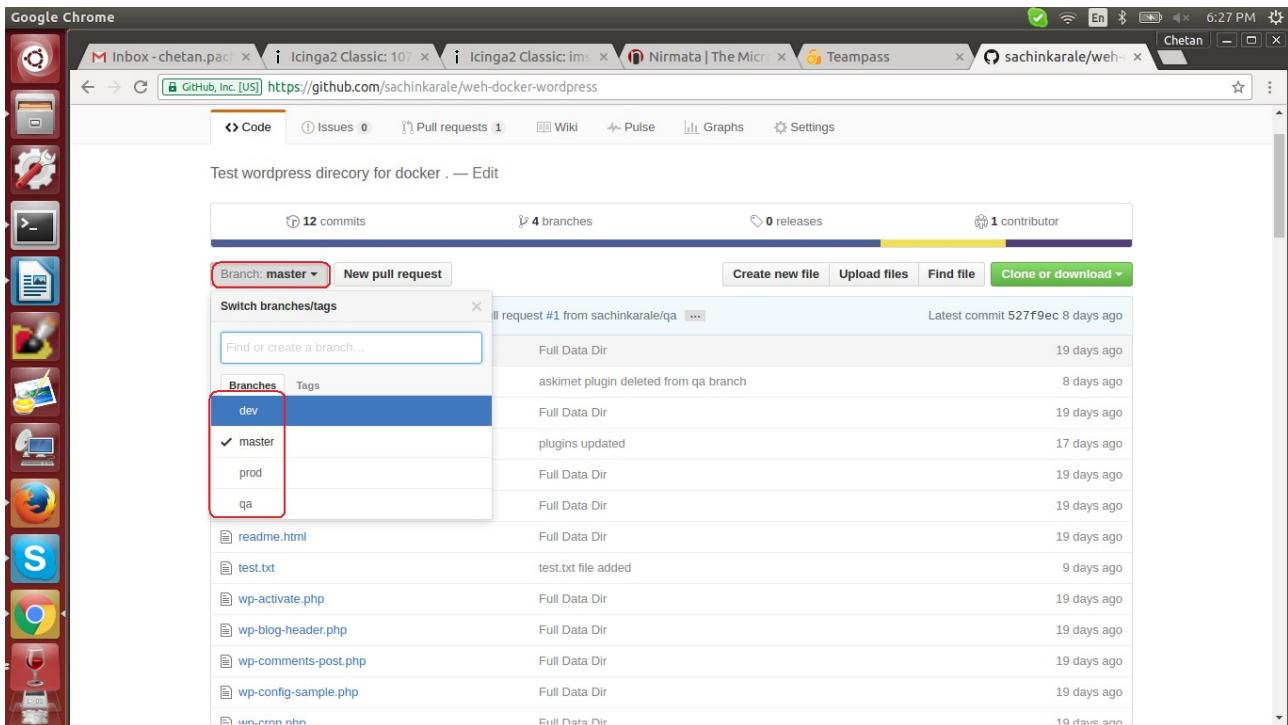
d. After push command you will see all contents to your repository as shown in the following fig.



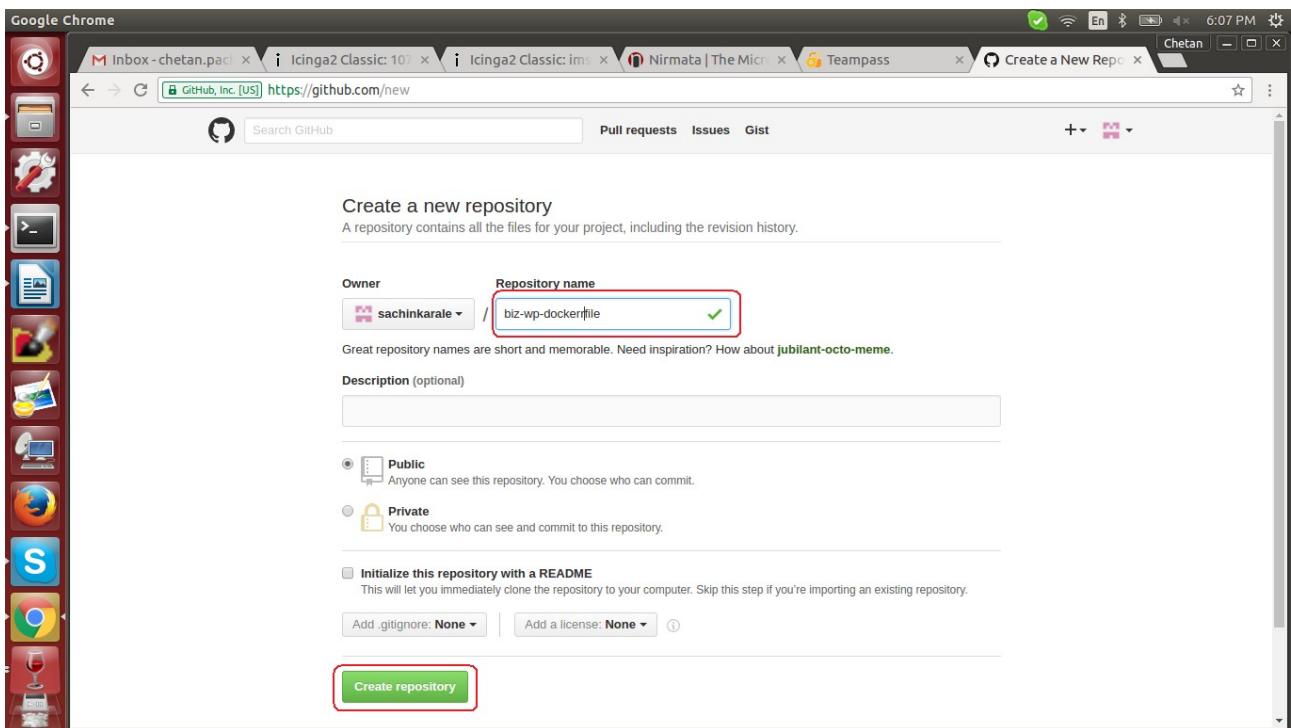
d. Next , We need to create three branches with name dev, qa , prod.
As shown in the following fig create three different branches.



e. When you create new branches then initially data from master branch will automatically sync to all branches. No need to push data to all branches.



- f. Again , we need to create another repository. In our case we created with name “**biz-wp-dockerfile**” .



- g. We created this repository to store Dockerfile, id_rsa key , and wp-config.php files.

Again push all of this contents to this repository using following commands.

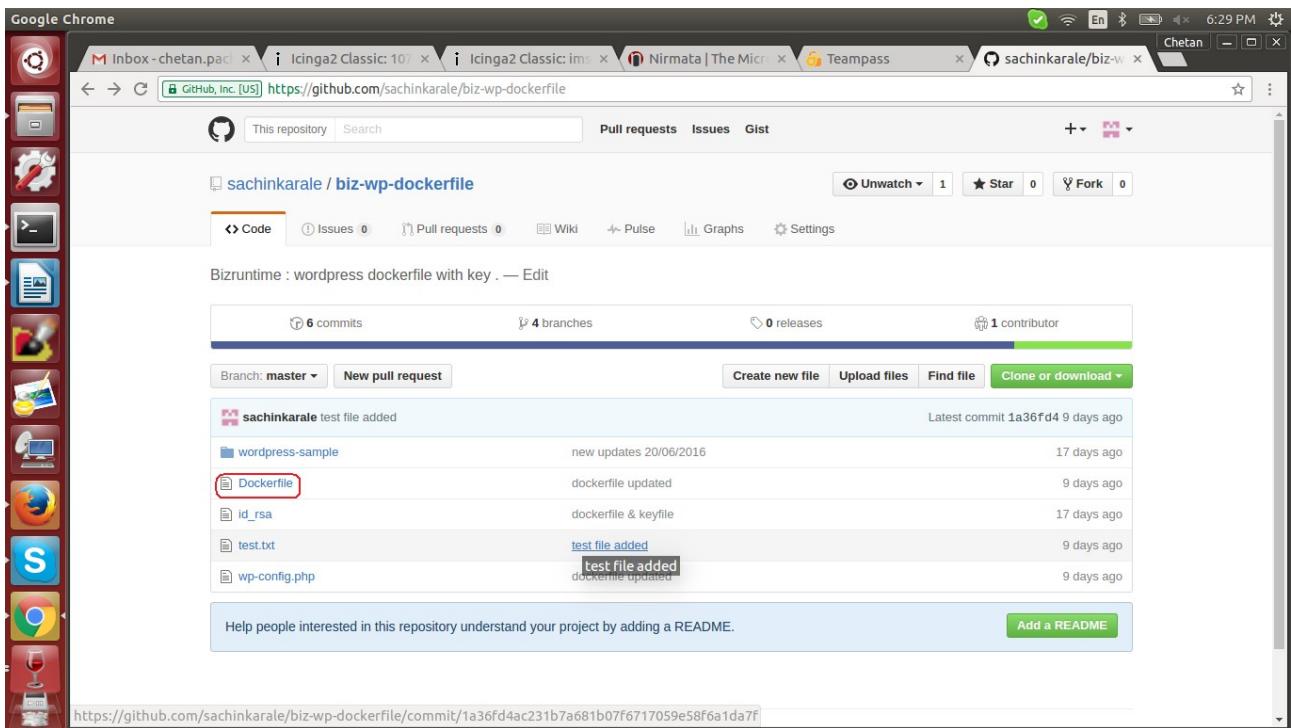
```
# git clone https://github.com/sachinkarale/biz-wp-dockerfile.git
# cd biz-wp-dockerfile
```

Copy Dockerfile , id_rsa kye and wp-config files here.

Then,

```
# git add *
# git commit -a -m "some message"
# git push -u origin master
```

h. After push successfully, you will see all the contents to this repository as shown in the following fig.

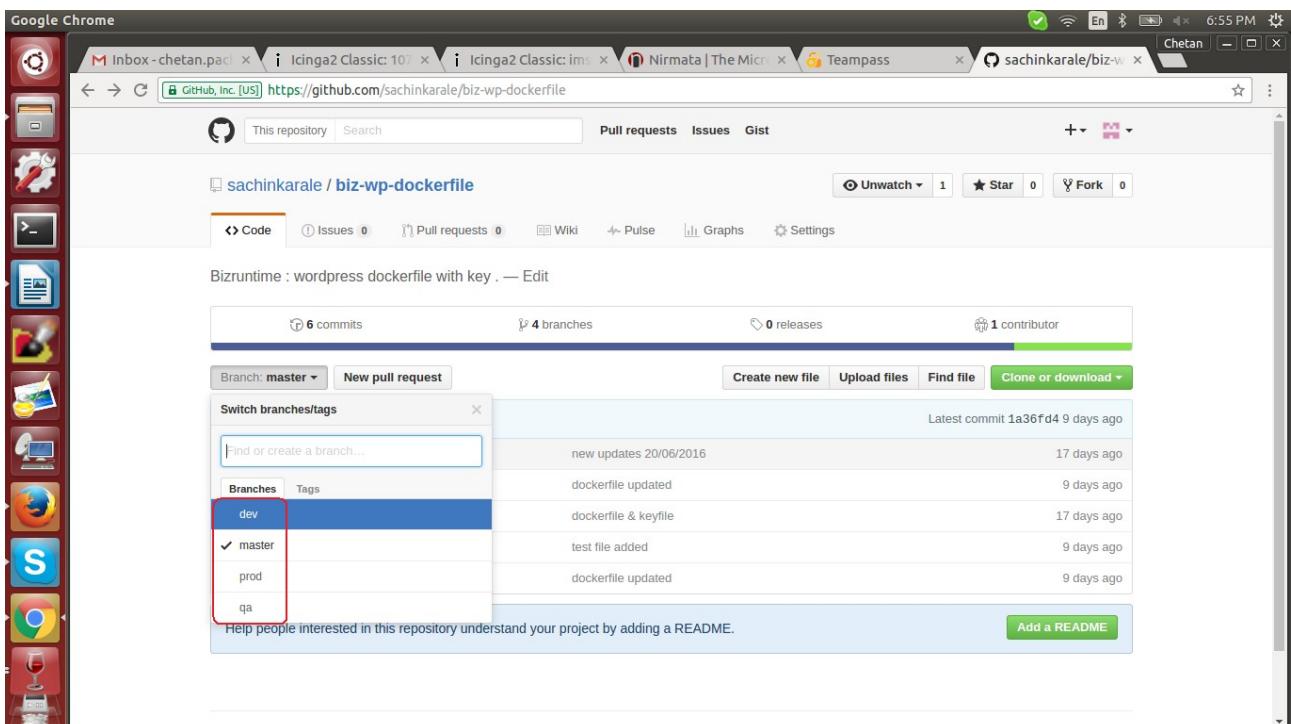


Bizruntime : wordpress dockerfile with key . — Edit

Author	Message	Time
sachinkarale	test file added	Latest commit 1a36fd4 9 days ago
	wordpress-sample	new updates 20/06/2016
	Dockerfile	dockerfile updated
	id_rsa	dockerfile & keyfile
	test.txt	test file added
	wp-config.php	test file added dockerfile updated

Help people interested in this repository understand your project by adding a README.

h. As we have created three branches previously. Same things we have to do here.



Bizruntime : wordpress dockerfile with key . — Edit

Switch branches/tags

Branch	Tags
dev	
✓ master	
prod	
qa	

Help people interested in this repository understand your project by adding a README.

2 . Installation of jenkins and configuration of jenkins jobs.

Installation :

- a. First step is, install java package using following command.

```
# apt-get install openjdk-7-jdk
```

- b. Next, we need to set java home path.

Open environment file & add following line in that.

```
“JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64/”
```

```
# nano /etc/environment
```

- c. Next step is that, we have to download tomcat and jenkins.

- Following commad will download tar file for tomcat.

```
# cd /opt/  
# wget http://mirror.fibergrid.in/apache/tomcat/tomcat-8/v8.0.36/bin/apache  
-tomcat-8.0.36.tar.gz
```

- Following commad will download war file for jenkins.

```
# wget http://mirrors.jenkins-ci.org/war-stable/latest/jenkins.war
```

- d. Next, copy jenkins war as follows.

```
# cp jenkins.war apache-tomcat-8.0.36/webapps/
```

- e. start tomcat service as follows.

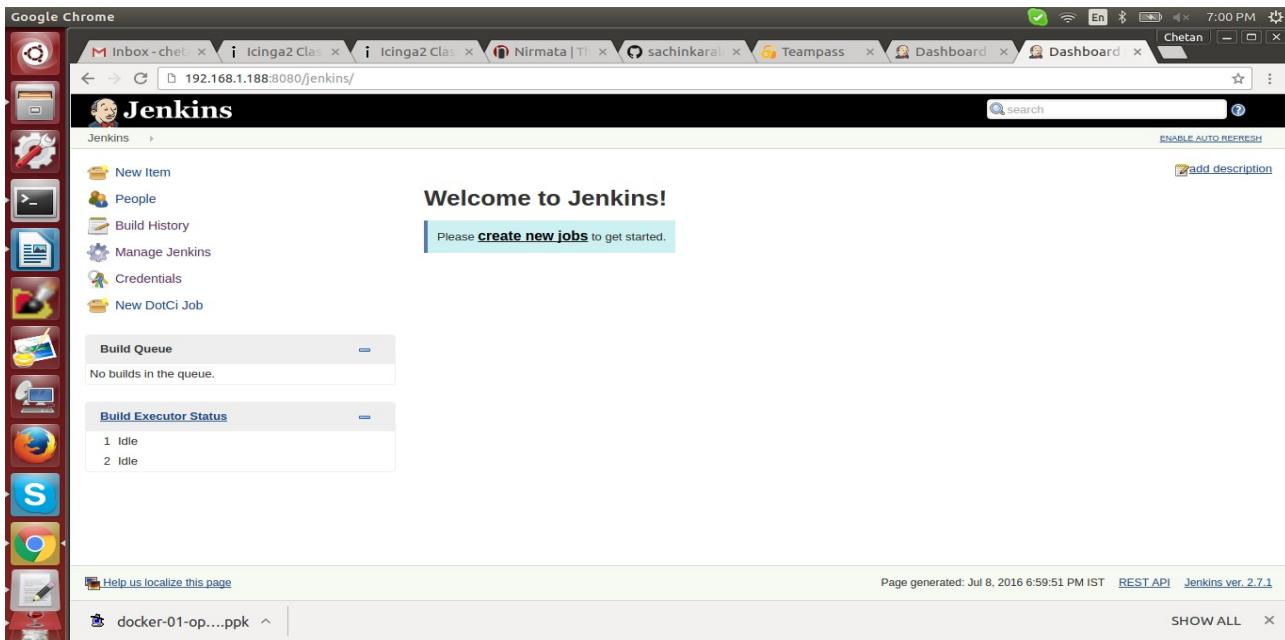
```
# cd /opt/apache-tomcat-8.0.36/bin
```

```
# ./catalina.sh start
```

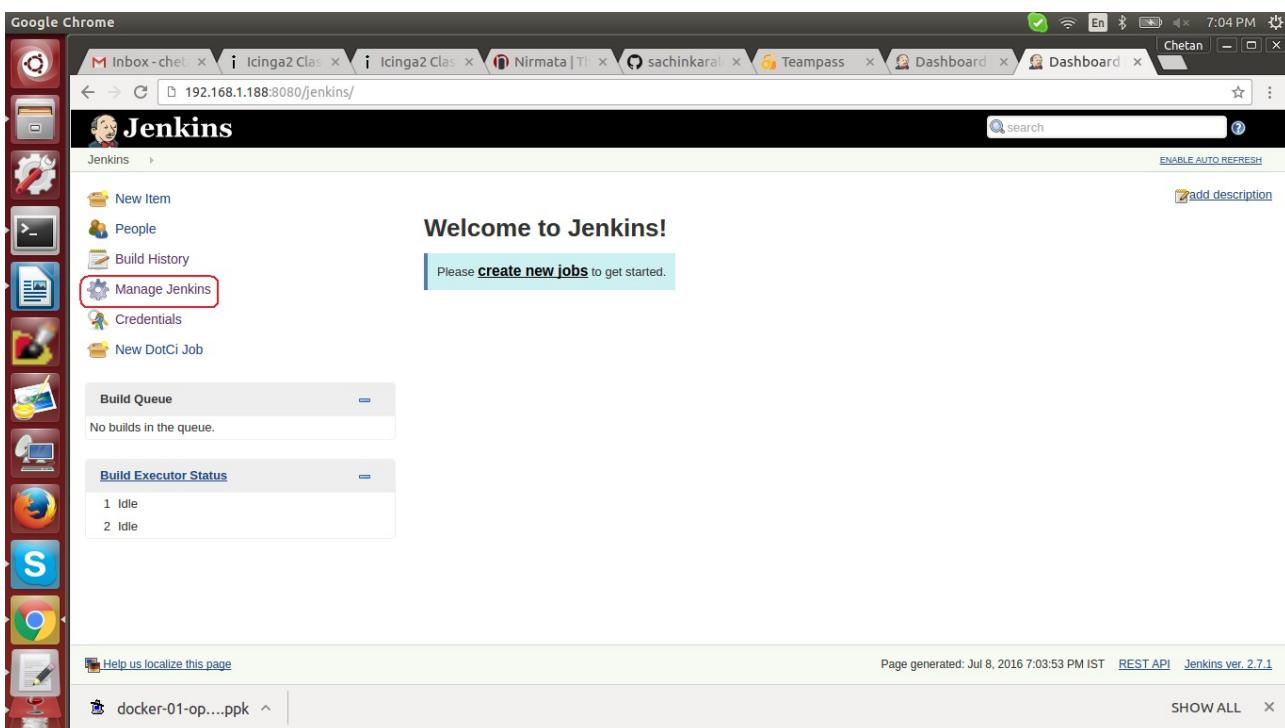
f. Now, open your browser & enter url as follow.

<http://ip-add-of-server:8080/jenkins>

you will get page as follows.



g. At this point we have to install Plugins for **Docker & Github**. So, go through the following pictures.



Google Chrome

192.168.1.188:8080/jenkins/manage

Jenkins

Manage Jenkins

Unsecured Jenkins allows anyone on the network to launch processes on your behalf. Consider at least enabling authentication to discourage misuse.

Setup Security **Dismiss**

New Item
People
Build History
Manage Jenkins
Credentials
New DotCi Job

Build Queue
No builds in the queue.

Build Executor Status
1 Idle
2 Idle

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. (updates available)

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

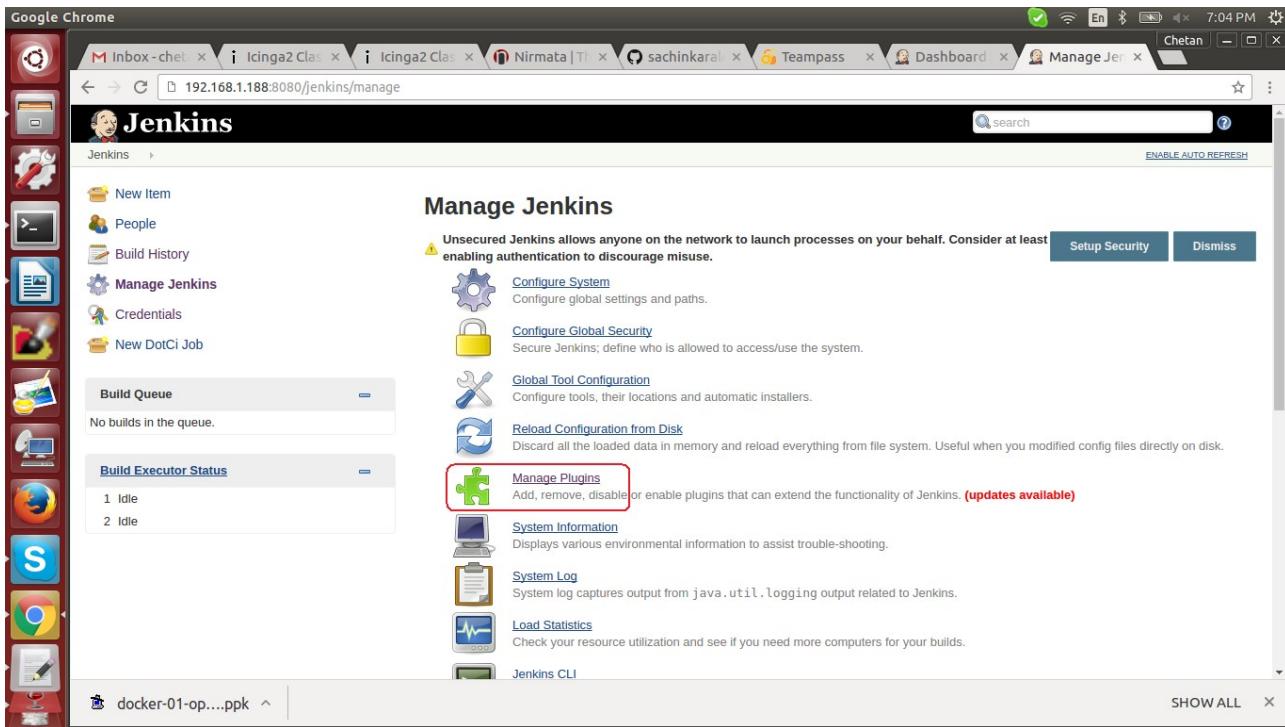
System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from java.util.logging output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

Jenkins CLI

SHOW ALL



Google Chrome

192.168.1.188:8080/jenkins/pluginManager/available

Jenkins

Plugin Manager

Back to Dashboard
Manage Jenkins
Update Center

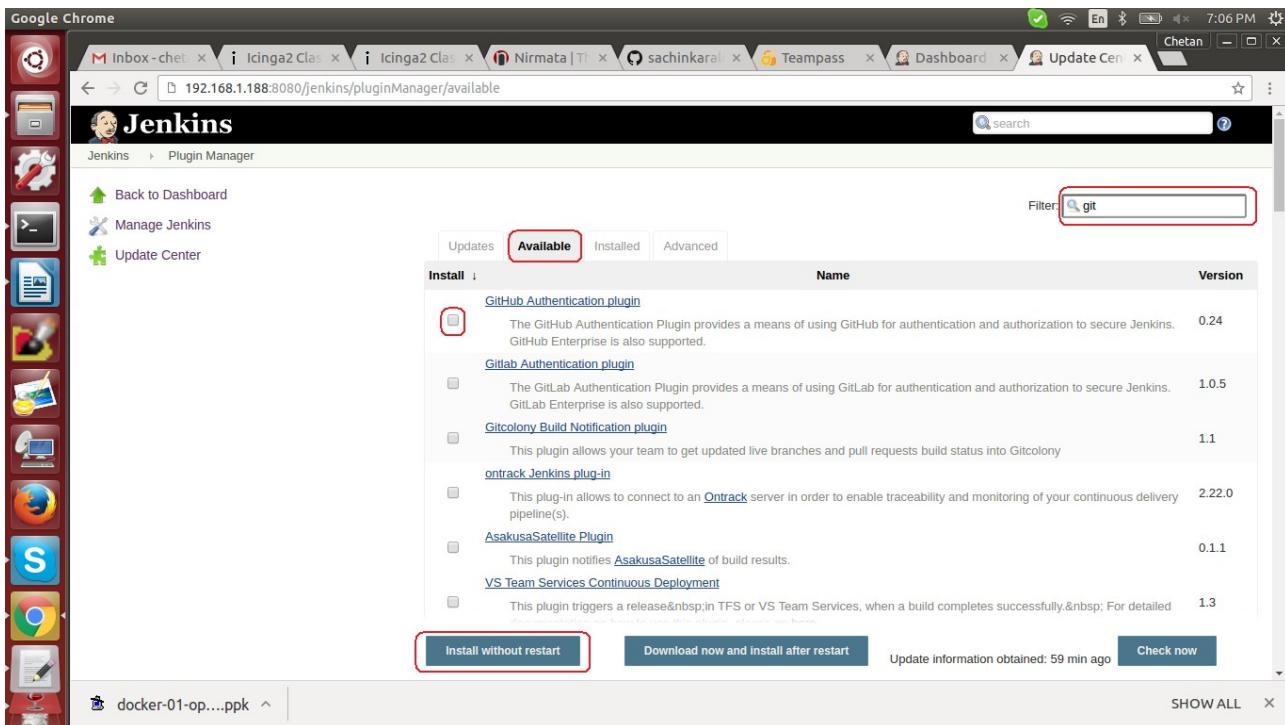
Updates Available Installed Advanced

Filter: git

Install	Name	Version
<input checked="" type="checkbox"/>	GitHub Authentication plugin The GitHub Authentication Plugin provides a means of using GitHub for authentication and authorization to secure Jenkins. GitHub Enterprise is also supported.	0.24
<input checked="" type="checkbox"/>	Gitlab Authentication plugin The GitLab Authentication Plugin provides a means of using GitLab for authentication and authorization to secure Jenkins. GitLab Enterprise is also supported.	1.0.5
<input checked="" type="checkbox"/>	Gitcolony Build Notification plugin This plugin allows your team to get updated live branches and pull requests build status into Gitcolony	1.1
<input checked="" type="checkbox"/>	Ontrack Jenkins plug-in This plug-in allows to connect to an Ontrack server in order to enable traceability and monitoring of your continuous delivery pipeline(s).	2.22.0
<input checked="" type="checkbox"/>	AsakusaSatellite Plugin This plugin notifies AsakusaSatellite of build results.	0.1.1
<input checked="" type="checkbox"/>	VS Team Services Continuous Deployment This plugin triggers a release in TFS or VS Team Services, when a build completes successfully. For detailed information about this plugin, please refer to the VS Team Services Continuous Deployment documentation .	1.3

Install without restart **Download now and install after restart** Update information obtained: 59 min ago **Check now**

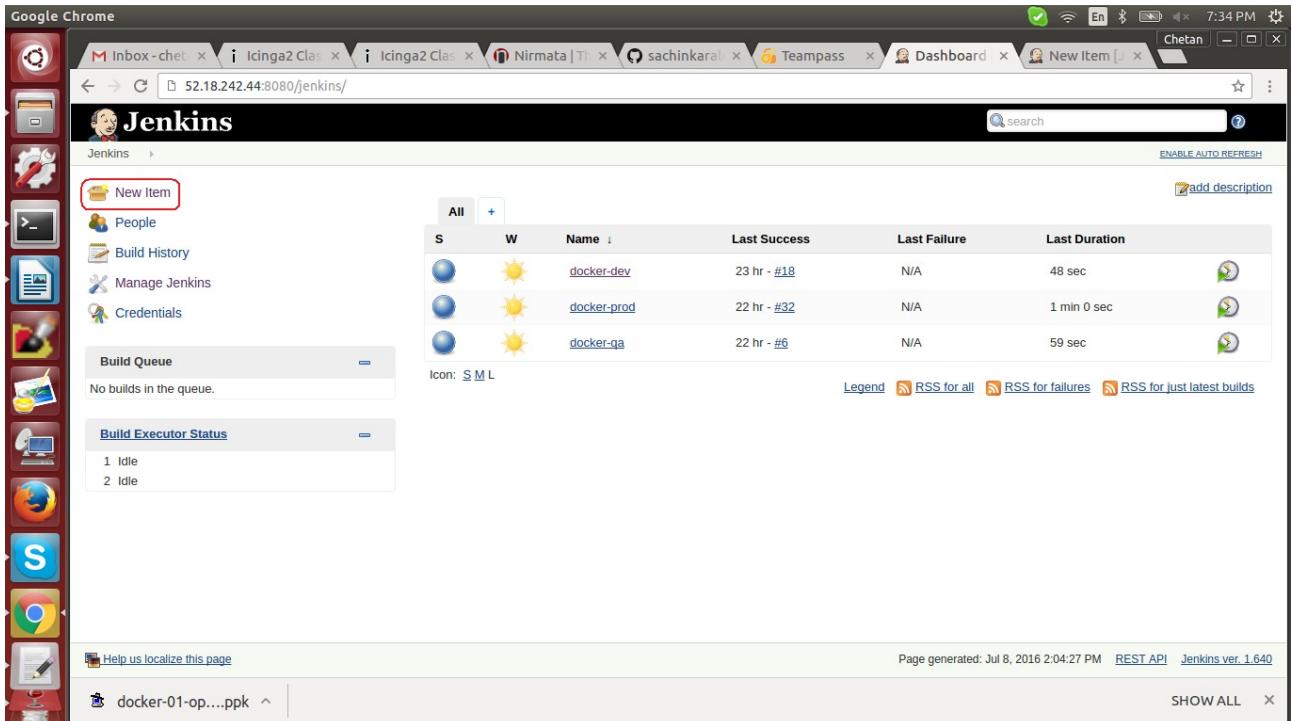
SHOW ALL



Creating New jobs in jenkins :

a. First of all, login to your Jenkins server, we have to create jobs for our three branches ie. dev, qa, prod. Here we are explaining for only dev. Other two can be created based on dev.

Following are steps to create new job dev.

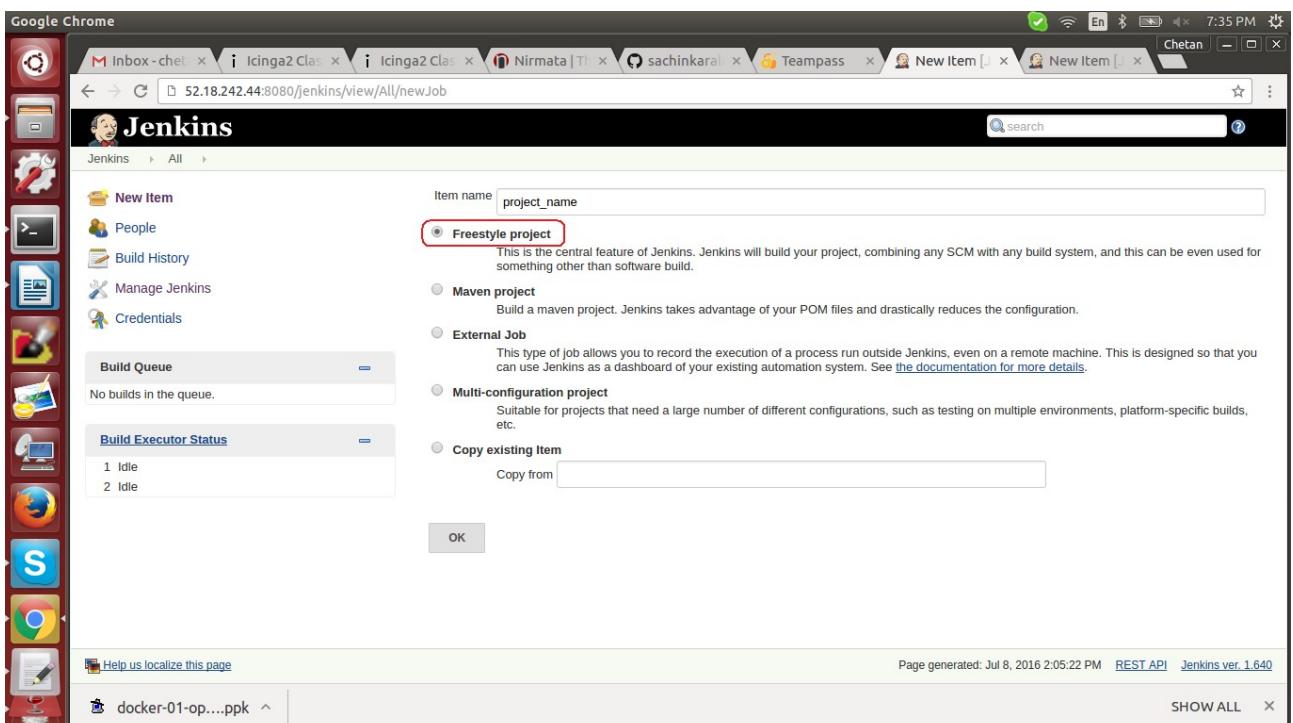


The screenshot shows the Jenkins dashboard with the URL <http://52.18.242.44:8080/jenkins/>. The left sidebar includes links for New Item, People, Build History, Manage Jenkins, and Credentials. The main area displays a table of existing jobs:

S	W	Name	Last Success	Last Failure	Last Duration
●	☀	docker-dev	23 hr - #18	N/A	48 sec
●	☀	docker-prod	22 hr - #32	N/A	1 min 0 sec
●	☀	docker-qa	22 hr - #6	N/A	59 sec

Below the table, there are links for RSS feeds: RSS for all, RSS for failures, and RSS for just latest builds. The bottom right corner shows the page was generated on Jul 8, 2016 at 2:04:27 PM, with REST API and Jenkins ver. 1.640 links.

Fig. 1



The screenshot shows the 'New Item' creation dialog in Jenkins. The 'Item name' field is set to 'project_name'. The 'Freestyle project' radio button is selected, with a tooltip explaining it's the central feature of Jenkins, combining any SCM with any build system. Other options shown are Maven project, External Job, Multi-configuration project, and Copy existing Item. The 'OK' button is visible at the bottom.

Fig. 2

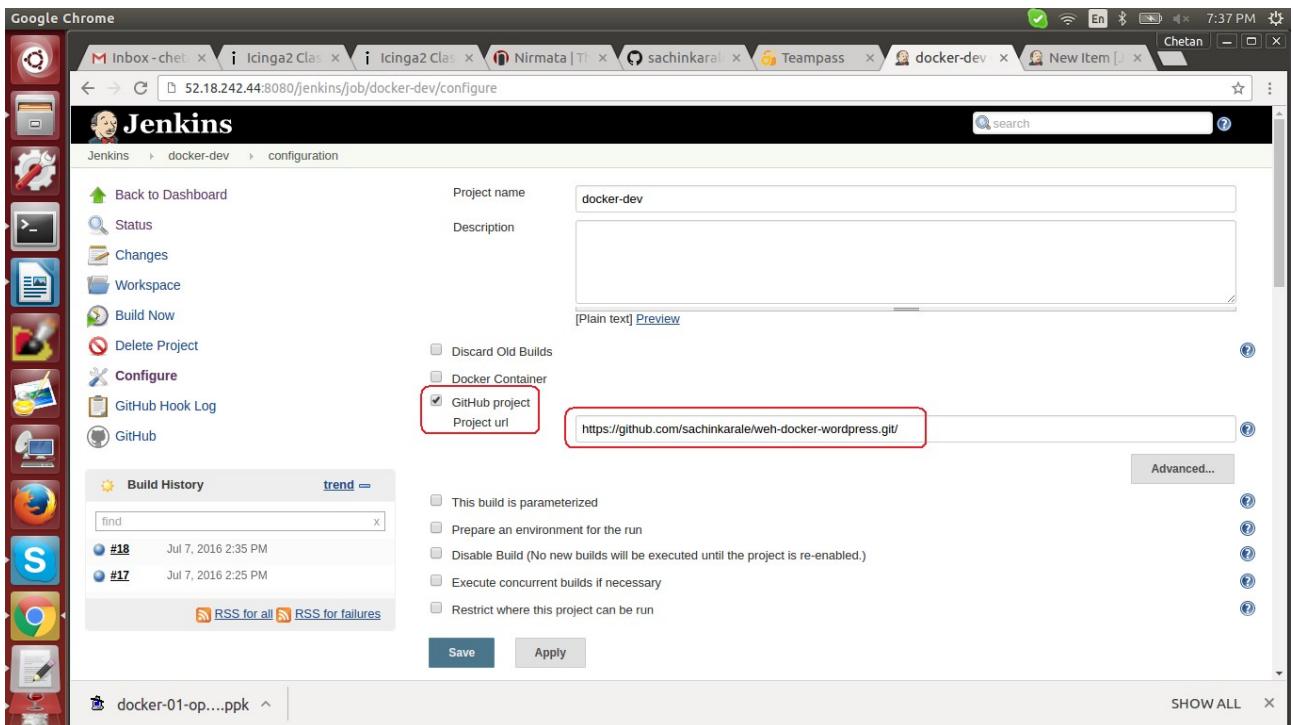


Fig. 3

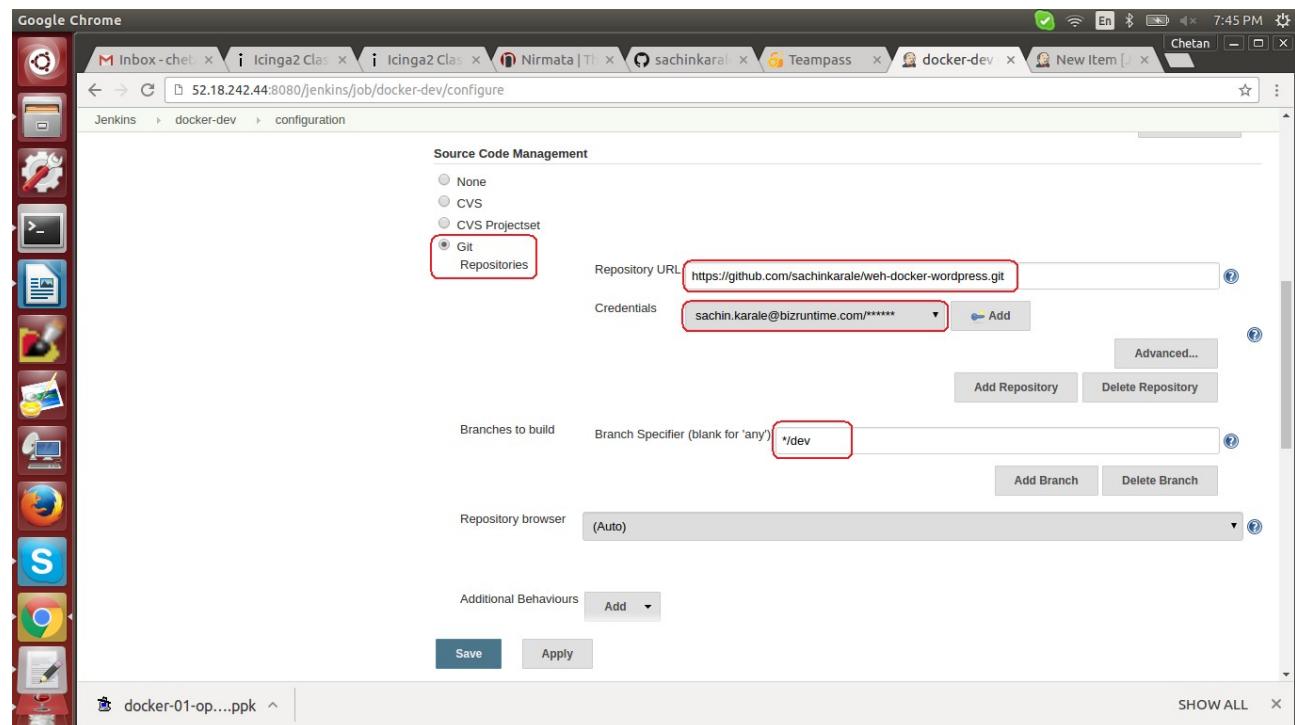


Fig. 4

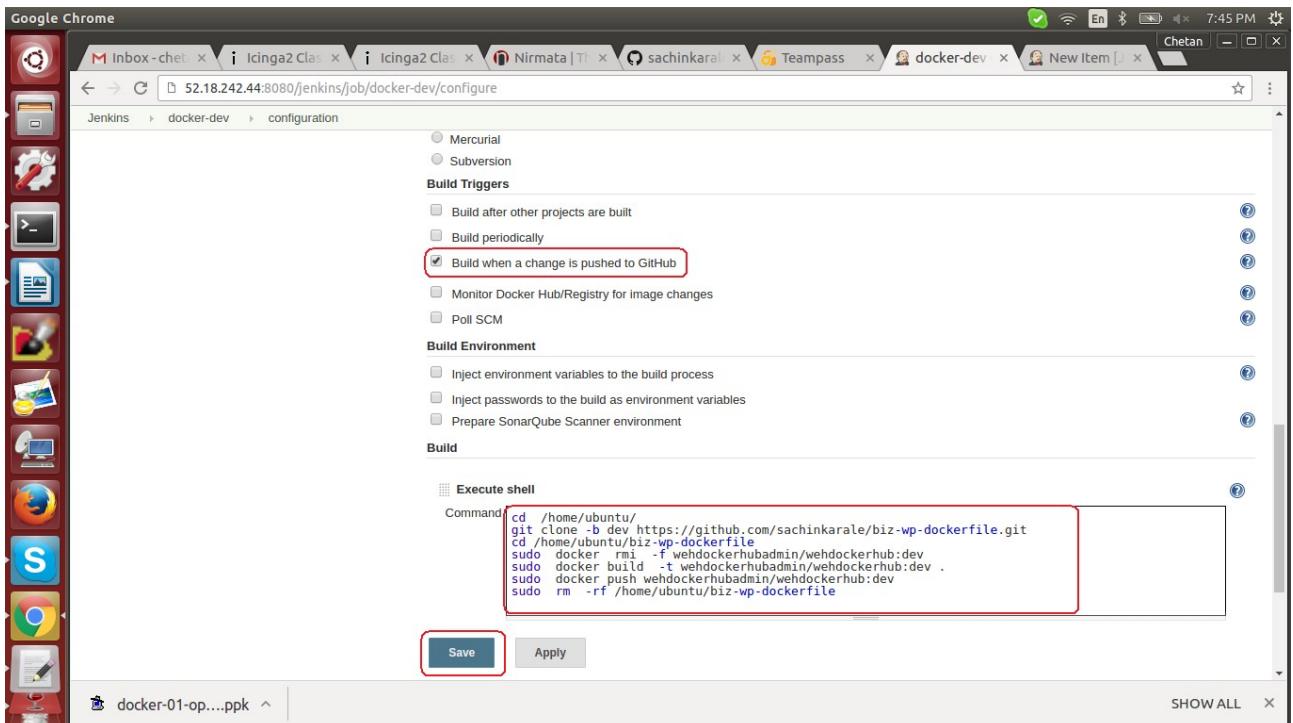


Fig. 5

b. Next step is to create another two jobs for **prod & qa** with above reference.

c. Followings are shell command for all three jobs.

dev :

```

cd /home/ubuntu/
git clone -b dev https://github.com/sachinkarale/biz-wp-dockerfile.git
cd /home/ubuntu/biz-wp-dockerfile
sudo docker rmi -f wehdockerhubadmin/wehdockerhub:dev
sudo docker build -t wehdockerhubadmin/wehdockerhub:dev .
sudo docker push wehdockerhubadmin/wehdockerhub:dev
sudo rm -rf /home/ubuntu/biz-wp-dockerfile

```

prod :

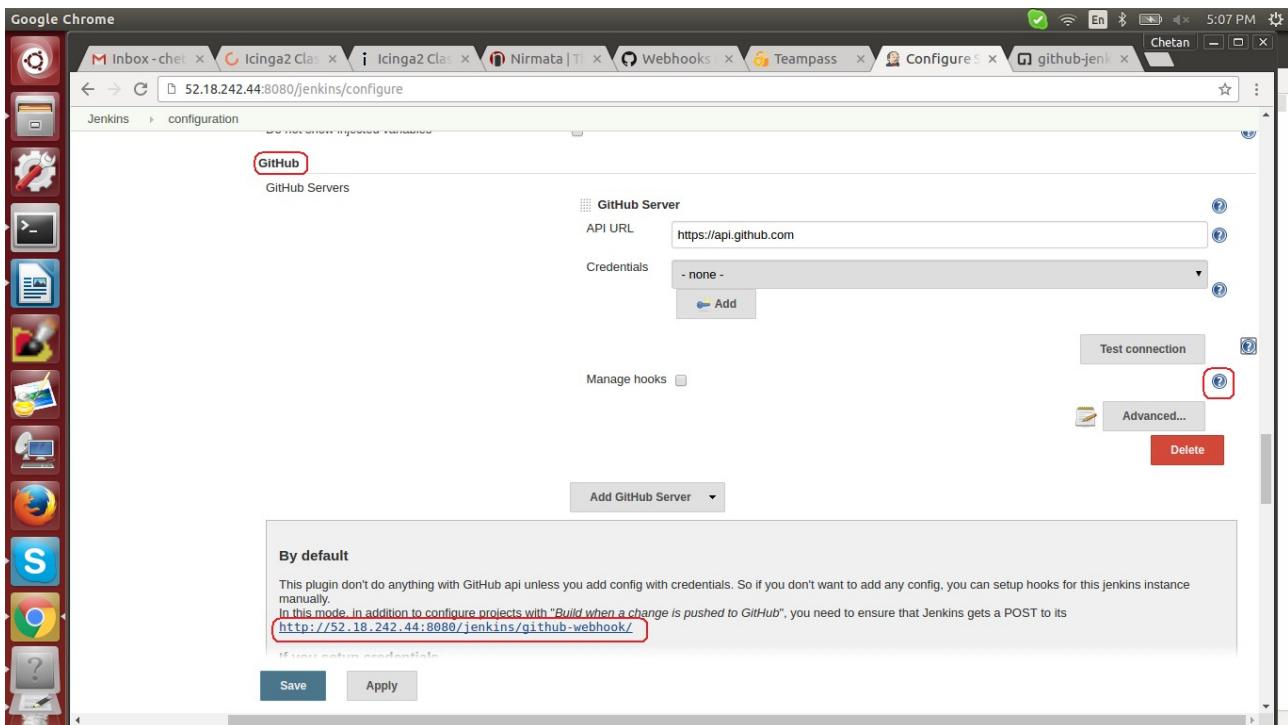
```
cd /home/ubuntu/  
git clone -b prod https://github.com/sachinkarale/biz-wp-dockerfile.git  
cd /home/ubuntu/biz-wp-dockerfile  
sudo docker rmi -f wehdockerhubadmin/wehdockerhub:prod  
sudo docker build -t wehdockerhubadmin/wehdockerhub:prod .  
sudo docker push wehdockerhubadmin/wehdockerhub:prod  
sudo rm -rf /home/ubuntu/biz-wp-dockerfile
```

qa :

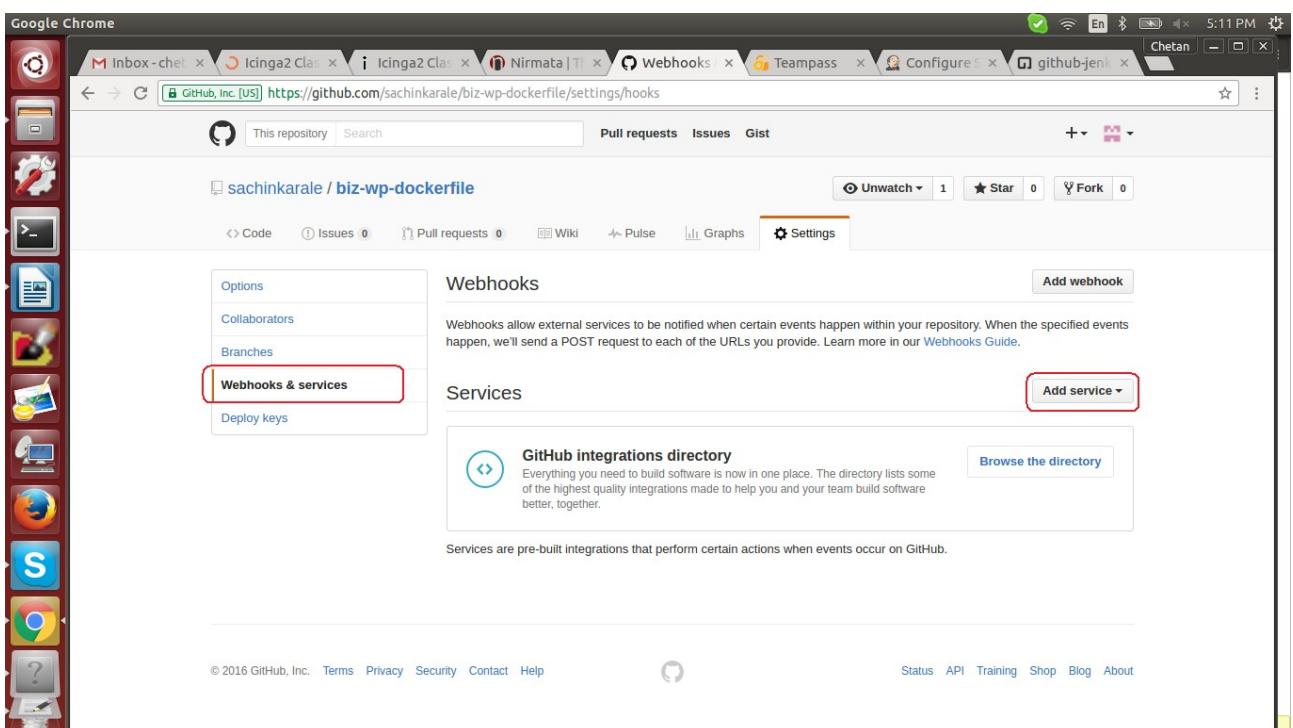
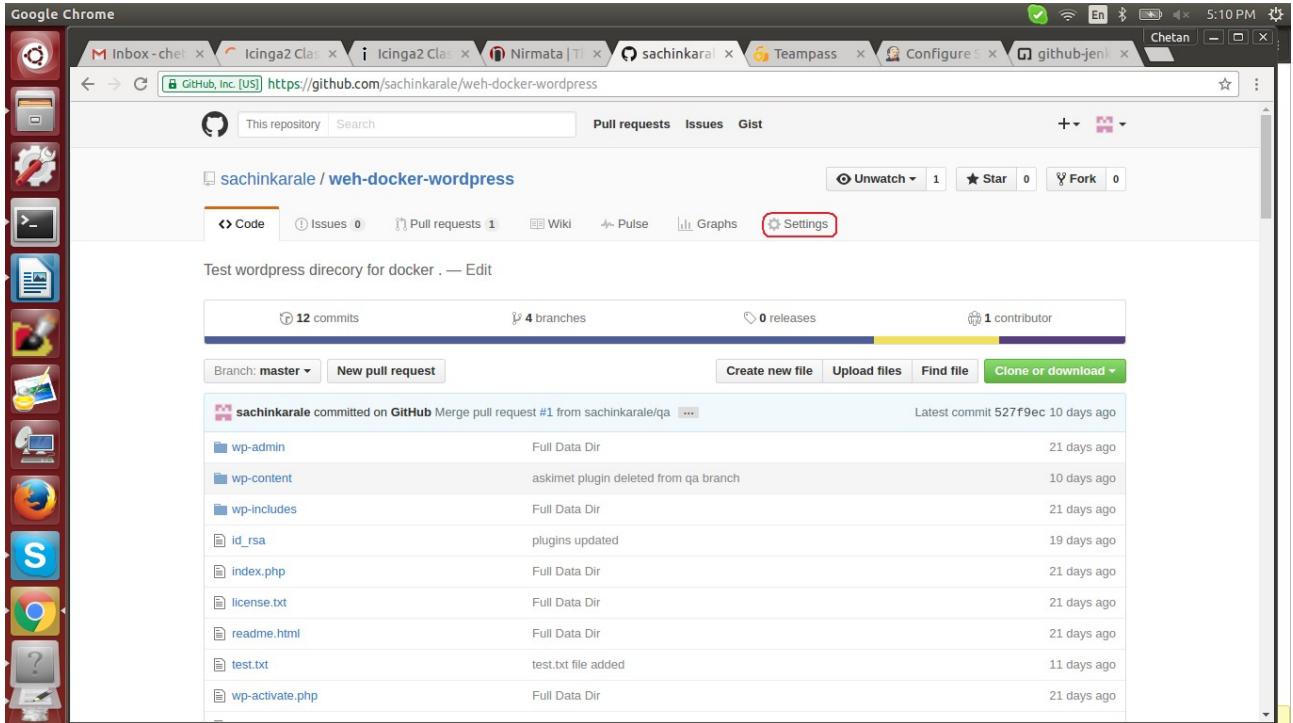
```
cd /home/ubuntu/  
git clone -b qa https://github.com/sachinkarale/biz-wp-dockerfile.git  
cd /home/ubuntu/biz-wp-dockerfile  
sudo docker rmi -f wehdockerhubadmin/wehdockerhub:qa  
sudo docker build -t wehdockerhubadmin/wehdockerhub:qa .  
sudo docker push wehdockerhubadmin/wehdockerhub:qa  
sudo rm -rf /home/ubuntu/biz-wp-dockerfile
```

3. How to Configure a Jenkins build process by a GitHub push

- a. This time we will cover how the build process is automatically started when a contributor pushes to the respective GitHub repository.
- b. To be able to trigger the build process by GitHub, you have to configure the Jenkins instance which should be triggered after the push. For this purpose the Jenkins web hook URL is required and must be submitted in the GitHub project.
- c. To fetch the Jenkins hook URL go to the Jenkins system settings (Manage Jenkins -> Configure System). Scroll down to the very bottom of the page and look in the “GitHub” section.



d. In the next step, enter your hook URL in the GitHub project. Thereto, go to your GitHub project settings and click on “Webhooks & Services”. Press the “Add service” button and choose “Jenkins (GitHub plugin)” from the service list.



This screenshot shows the GitHub repository settings for 'sachinkarale / biz-wp-dockerfile'. The 'Webhooks' tab is active. A modal window titled 'Services' is open, showing a list of available services. The 'Jenkins (GitHub plugin)' option is highlighted with a red box.

e. Enter your webhook URL in the “Jenkins hook url” field and press “Add service”.

This screenshot shows the 'Services / Add Jenkins (GitHub plugin)' configuration page. The 'Jenkins hook url' field contains 'http://52.18.242.44:8080/jenkins/github-webhook/'. The 'Active' checkbox is checked. The 'Add service' button is highlighted with a red box.

f. To verify that the hook is working, click on the created service and press the “Test service” button in the upper right corner. If everything is correctly configured, the Jenkins service has got a green checkmark icon in the service overview and the build process has started.

The screenshot shows a Google Chrome window with multiple tabs open. The active tab is 'GitHub, Inc. [US] https://github.com/sachinkarale/biz-wp-dockerfile/settings/hooks/9056085'. The page displays the 'Services / Manage Jenkins (GitHub plugin)' section. On the left, there's a sidebar with options like 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The 'Webhooks & services' option is highlighted with a red box. The main content area shows an 'Install Notes' section with instructions about Jenkins hooks and URLs. Below it is a 'Jenkins hook url' input field containing 'http://52.18.242.44:8080/jenkins/github-webhook/'. A checked 'Active' checkbox is followed by a note: 'We will run this service when an event is triggered.' At the bottom are 'Update service' and 'Delete service' buttons. A 'Test service' button is located at the top right of the main content area, also highlighted with a red box.

The screenshot shows a Google Chrome window with multiple tabs open. The active tab is 'GitHub, Inc. [US] https://github.com/sachinkarale/biz-wp-dockerfile/settings/hooks'. The page displays the 'Webhooks' and 'Services' sections. The 'Webhooks & services' option in the sidebar is highlighted with a red box. The 'Webhooks' section contains a 'GitHub integrations directory' link and a 'Browse the directory' button. The 'Services' section lists 'Jenkins (GitHub plugin)' with a green checkmark icon, which is also highlighted with a red box. Other service entries like 'Travis CI', 'CircleCI', and 'AppVeyor' are listed below it. At the bottom of the page, there are links for 'Status', 'API', 'Training', 'Shop', 'Blog', and 'About'.

4. Automated Deploys With Docker Hub And Nirmata.

- a. Docker image builds are automated, the next step is to automate the deployment of these builds. This can be done using Docker Hub webhooks and Nirmata.
- b. Log in to Nirmata. You can find the tenant-id on the Settings->Account page in Nirmata web interface as follows.

The screenshot shows the Nirmata web client interface. The left sidebar has a vertical list of icons and menu items: Activity, Dashboard, Applications, Environments, Policies, Host Groups, Cloud Providers, Image Registries, Settings (selected), Getting Started, Private Cloud, Users, and Account (selected). The main content area is titled 'Account' and shows a summary for the user 'weh-nirmata-admin' (Administrator). It displays the user's email (redacted), Tenant ID (highlighted with a red box), API key (Generate an API key...), and Last Login (May 15th, 2016 14:19:36).

- c. Now, go to Docker Hub and add webhooks for your image repositories that should trigger automated deploys. The URL is displayed below:

<https://www.nirmata.io/imageregistry/api/dockerhub/notify/<tenant-id>>

The screenshot shows the Docker Hub repository settings for 'wehdockerhubadmin/wehdockerhub'. The top navigation bar includes links for Dashboard, Explore, Organizations, and a search bar. The repository name 'wehdockerhubadmin/wehdockerhub' is displayed with a star icon. Below the repository name, it says 'Last pushed: 2 days ago'. The 'Webhooks' tab is selected in the navigation bar. The page contains fields for 'Short Description' (with placeholder 'Short description is empty for this repo.'), 'Full Description' (with placeholder 'Full description is empty for this repo.'), 'Docker Pull Command' (set to 'docker pull wehdockerhubadmin/wehddo'), and 'Owner' (with a blurred profile picture). At the bottom, there is a 'Comments (0)' section and a link to the repository URL.

Google Chrome

Inbox - Icinga2 - Icinga2 - Nirmata - Webhooks - Teamp... - Dashboard - https:// - Google webhooks - Automati...

https://hub.docker.com/r/wehdockerhubadmin/wehdockerhub/-/settings/webhooks/

Dashboard Explore Organizations Search Create

PRIVATE REPOSITORY

wehdockerhubadmin/wehdockerhub ☆

Last pushed: 2 days ago

Repo Info Tags Collaborators Webhooks Settings

Workflows

TRIGGER EVENT

Image Pushed

When an image is pushed to this repo, your workflows will kick off based on your specified webhooks. [Learn More](#)

WEB HOOKS X

Webhook name: test

Webhook URL: <https://www.nirmata.io/imageresistry/api/dockerhub/notify/<ten-id>>

Cancel Save

nirmata-docker

https://www.nirmata.io/imageresistry/api/dockerhub/notify/ [REDACTED]

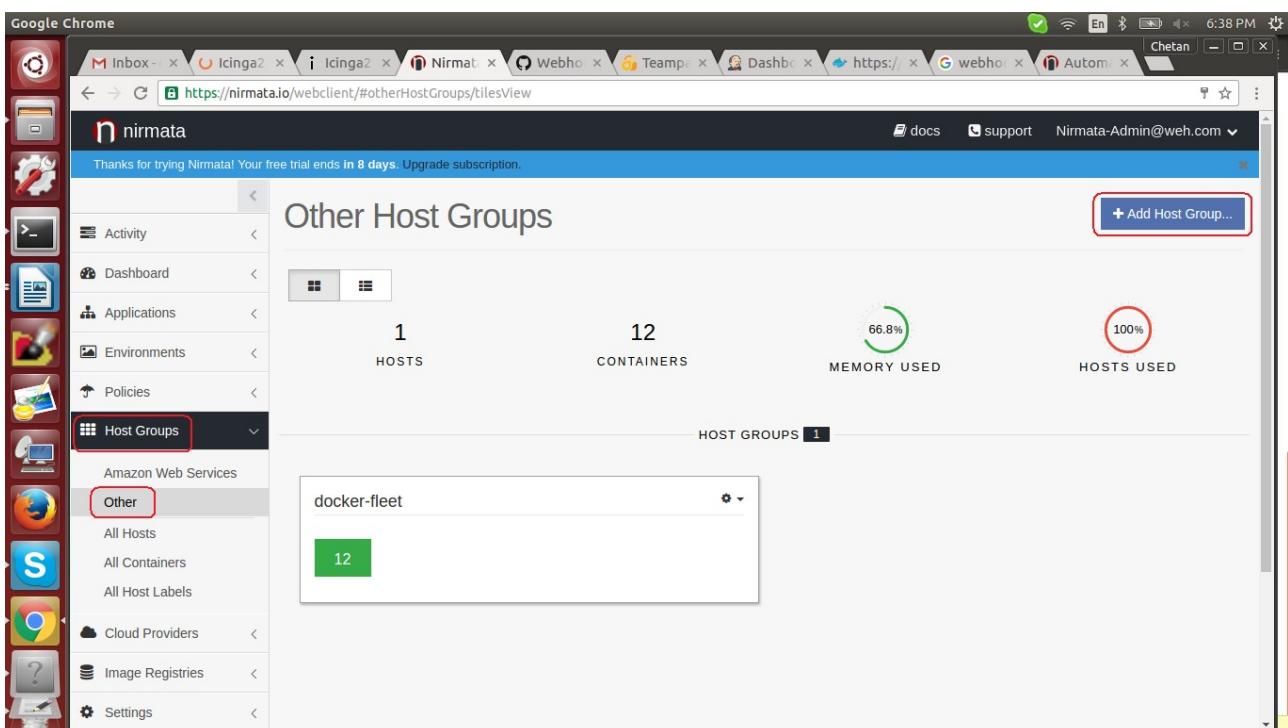
This screenshot shows the Docker Hub settings page for a private repository named 'wehdockerhubadmin/wehdockerhub'. The 'Webhooks' tab is selected. A 'Trigger Event' section shows 'Image Pushed' is selected. The 'Webhooks' section contains one entry named 'test' with the URL 'https://www.nirmata.io/imageresistry/api/dockerhub/notify/<ten-id>'. There are 'Cancel' and 'Save' buttons at the bottom of the webhook card. Below the card, there is another entry for 'nirmata-docker' with a similar URL.

5. Nirmata Host Groups Configuration

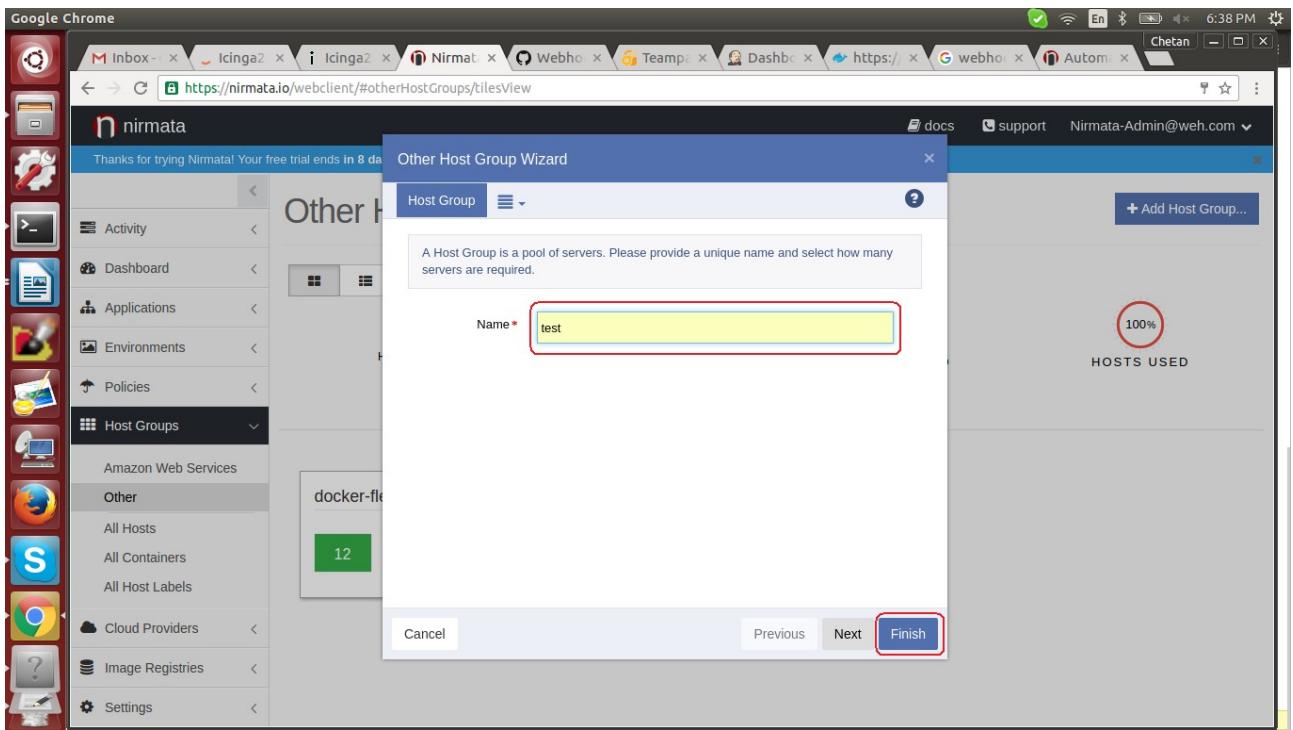
Host Setup :

Here are the general requirements for a Host that is managed by Nirmata.

1. Can be any Linux flavor that can run Docker.
 2. Requires Docker 1.10+
 3. The Nirmata agent must be installed.
- a. Login into your Nirmata account. Click on Host Groups -> Other -> +Add Host Group in Nirmata web interface.

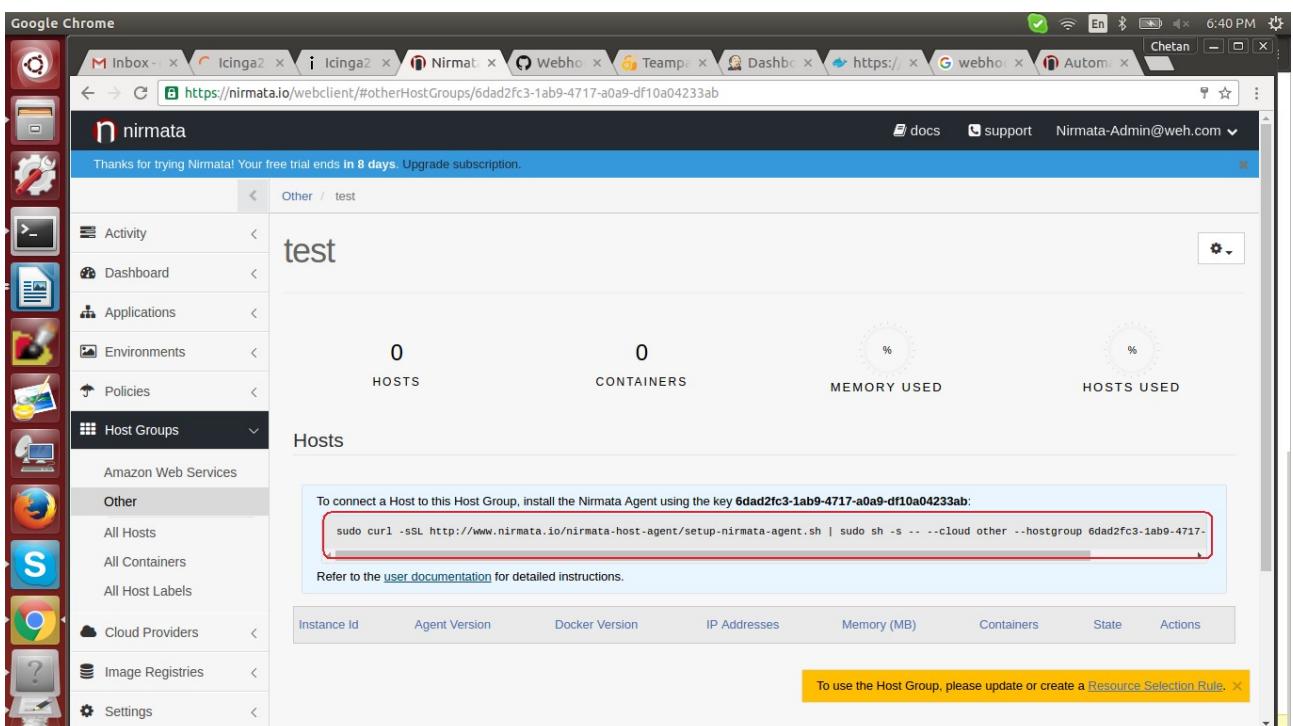


- b. Assign proper name to your Host Group. Next, Click on Finish button.



c. Next step is that, you have to copy the url as shown in the following fig.

And run this on your host where you want to run nirmata agent.



d. Once you run above url on host machin, nirmata agent will be running.

You can verify that your host group is properly configured with your host machin as shown in the following fig.

The screenshot shows a Google Chrome browser window with the URL <https://nirmata.io/webclient/#/otherHostGroups/83db8b7d-9906-4738-abcf-18abd6af029>. The page title is "docker-fleet". The left sidebar has a "Host Groups" dropdown open, showing "Other" selected. The main content area displays statistics: 1 HOSTS, 12 CONTAINERS, 66.8% MEMORY USED, and 100% HOSTS USED. Below this is a "Hosts" table with one row:

Instance Id	Agent Version	Docker Version	IP Addresses	Memory (MB)	Containers	State	Actions
6a25fed2-d0a9-4d86-92f0-b9fd6999477e	1.0	1.11.2	172.16.102.35	7480	12	Connected	

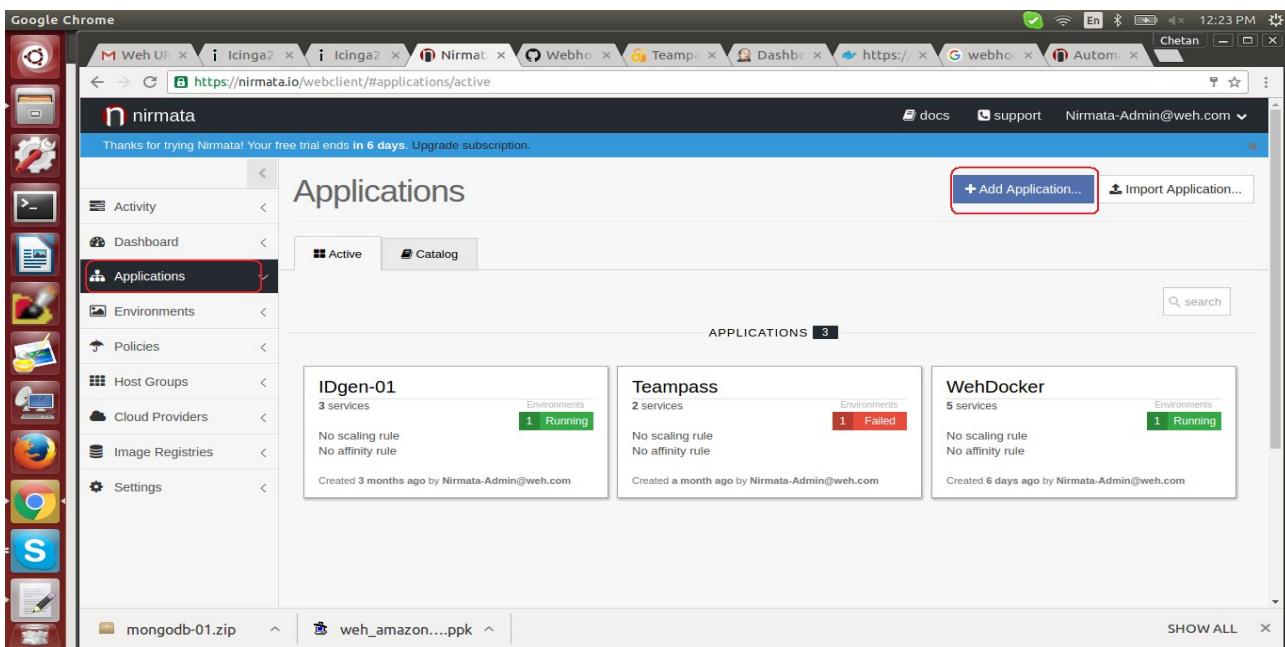
A red box highlights the "Connected" status in the Actions column of the table.

6. Nirmata Application And Environments Configuration

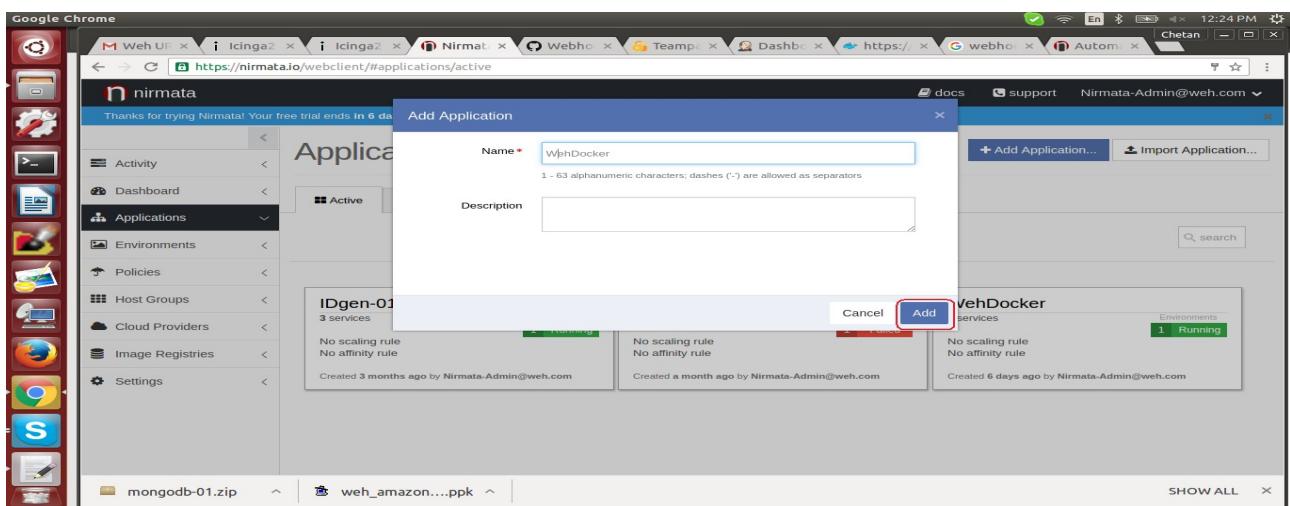
Application :

1. Adding New Application & Service.

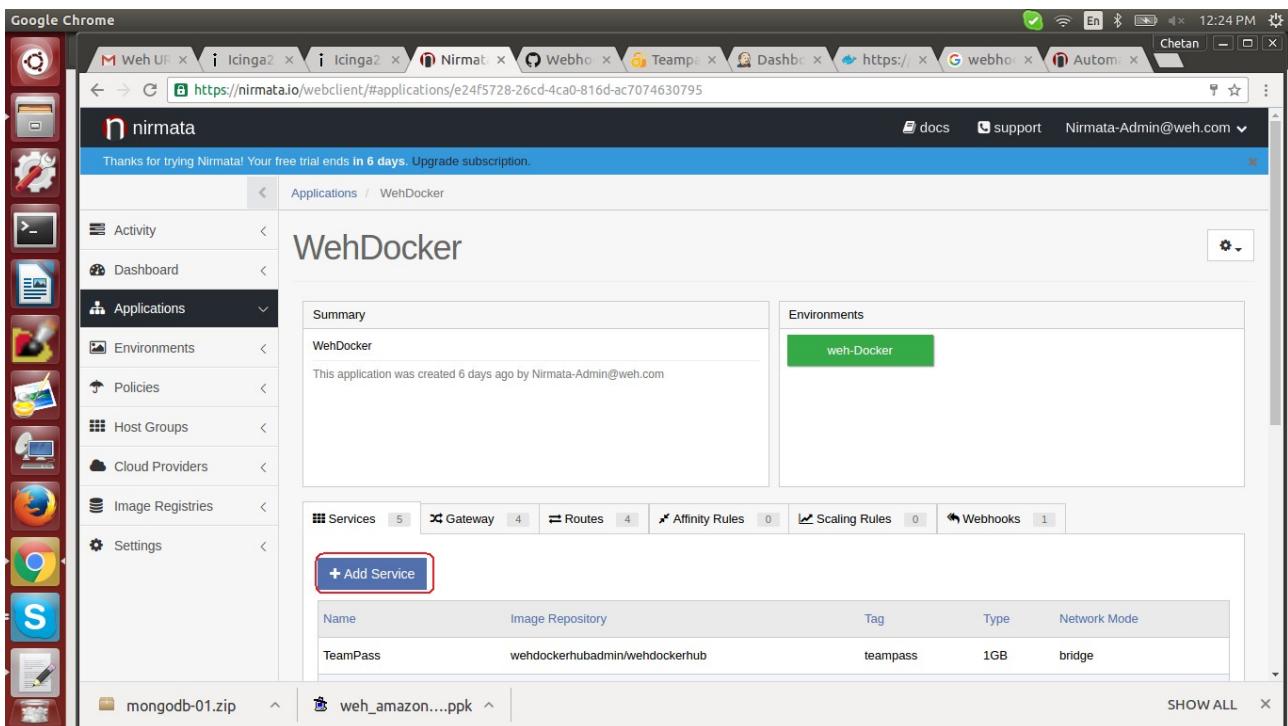
- First step is that, We have to add application in nirmata web interface. As shown in the following click on **Application --> +Add Application** .



- Provide proper name to your application & then click on **Add** button.

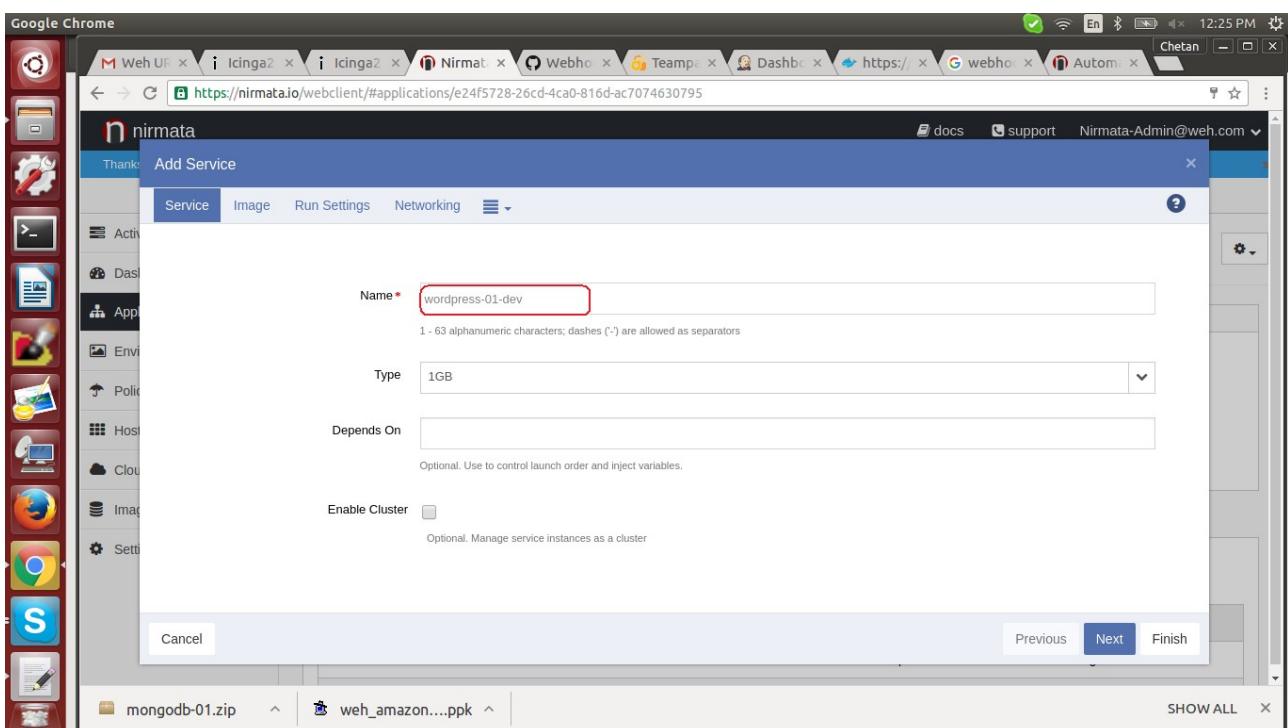


c. Next step is that, we have to add services in newly created application. Click on **+Add Service** button as below.



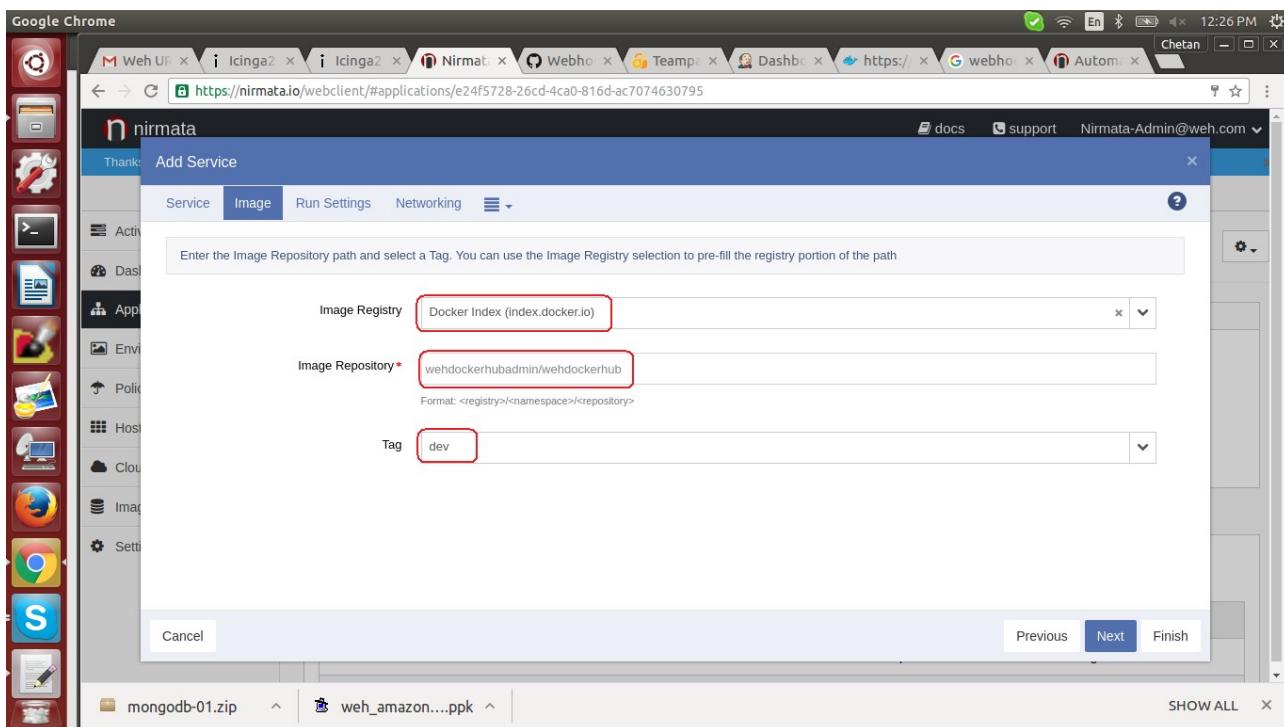
The screenshot shows the Nirmata WehDocker application interface. On the left, there is a sidebar with various icons for Activity, Dashboard, Applications (which is selected), Environments, Policies, Host Groups, Cloud Providers, Image Registries, and Settings. The main area is titled 'WehDocker' and contains two tabs: 'Summary' and 'Environments'. The 'Summary' tab shows details about the application, including its name 'WehDocker', creation date (6 days ago), and creator (Nirmata-Admin@weh.com). The 'Environments' tab lists one environment named 'weh-Docker'. At the bottom of the main area, there is a table with columns for Name, Image Repository, Tag, Type, and Network Mode. One row is visible: 'TeamPass' with 'wehdockerhubadmin/wehdockerhub' as the image repository, 'teampass' as the tag, '1GB' as the type, and 'bridge' as the network mode. Below the table is a red box highlighting the '+ Add Service' button. The browser's address bar shows the URL <https://nirmata.io/webclient/#applications/e24f5728-26cd-4ca0-816d-ac7074630795>.

d. Here, we are going to add service for dev, as we have to add another two service (ie. For prod & qa) based on this configuration.

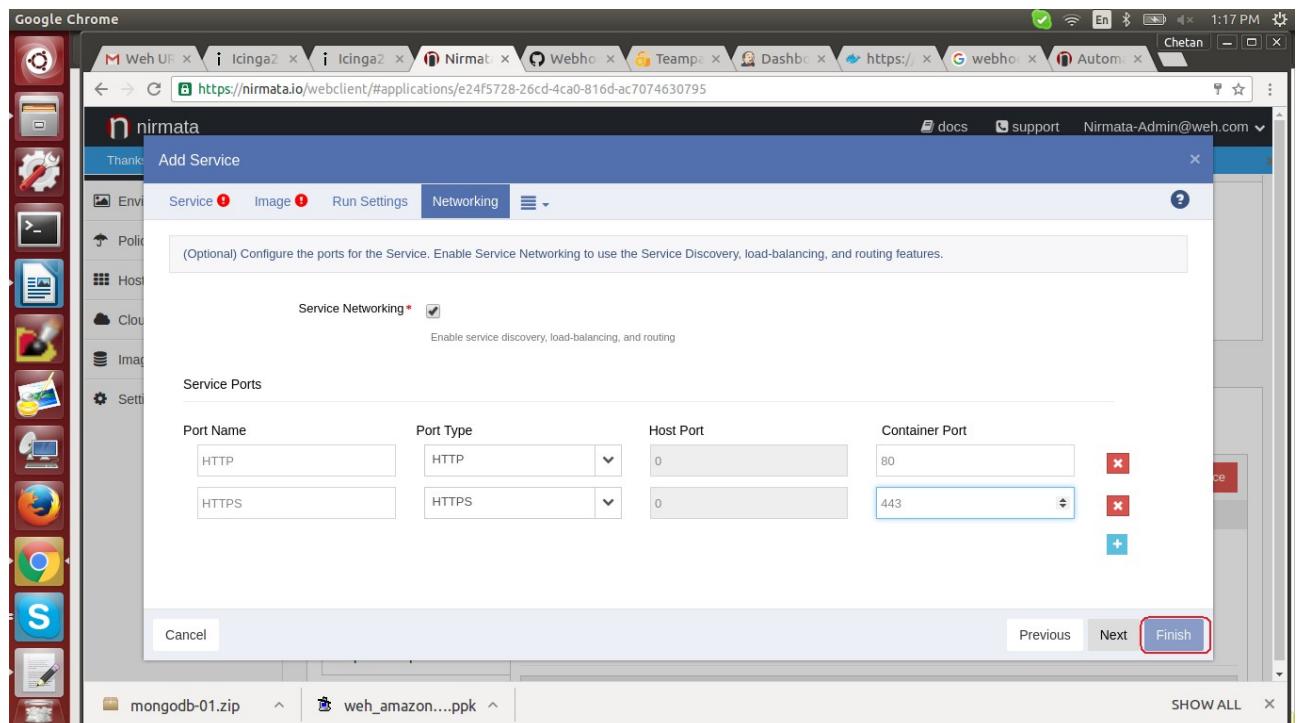


The screenshot shows the 'Add Service' dialog box. The 'Service' tab is active, displaying fields for 'Name' (set to 'wordpress-01-dev'), 'Type' (set to '1GB'), and 'Depends On' (an empty field). There are tabs for 'Service', 'Image', 'Run Settings', and 'Networking'. At the bottom right of the dialog are 'Cancel', 'Previous', 'Next', and 'Finish' buttons. The browser's address bar shows the URL <https://nirmata.io/webclient/#applications/e24f5728-26cd-4ca0-816d-ac7074630795>.

e. In the next step, select your image repository & tag as below fig.



f. Next step is, Configured service port.



2. Adding Gateway For Application.

a. Here we have to add gateway service, all our requests will go through gateway port. So, to add gateway service Click on Application --> Add Gateway. As shown in the below fig.

The screenshot shows the Nirmata WehDocker application interface. On the left, there is a vertical sidebar with various icons. The 'Applications' icon is highlighted with a red box. The main area shows a summary for an application named 'WehDocker'. On the right, there is a sidebar with several options: 'Edit Application', 'Add Service' (which is also highlighted with a red box), 'Add Gateway' (also highlighted with a red box), 'Deploy Application', 'Export Application', and 'Delete Application'. Below these options, there is a table listing services, with one row for 'weh-docker-gateway'.

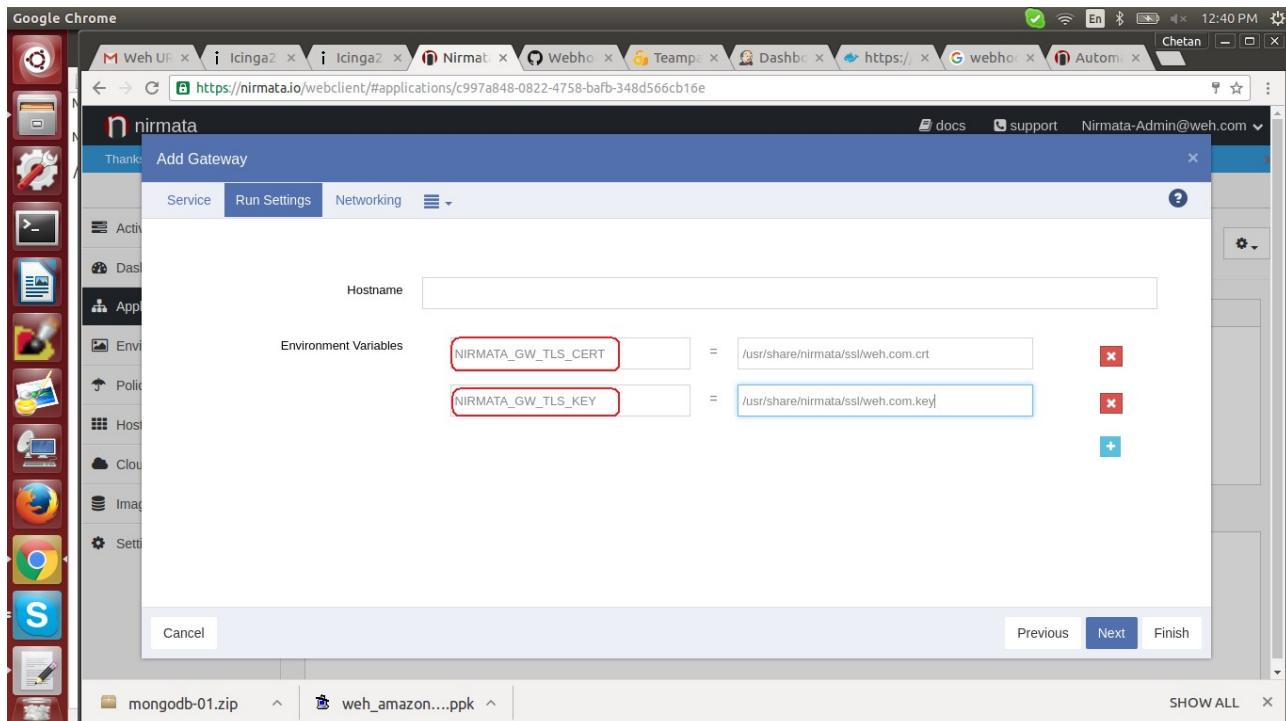
b. Next step is, give name to your gateway sevice.

The screenshot shows the 'Add Gateway' configuration dialog. The 'Service' tab is selected. The 'Name' field contains 'weh-gateway'. The 'Type' dropdown is set to '1GB'. The 'Image Repository' field contains 'nirmata/nirmata-gateway'. There is a 'Tag' field which is currently empty. At the bottom of the dialog, there are 'Cancel', 'Previous', 'Next', and 'Finish' buttons. The 'Next' button is highlighted with a blue box.

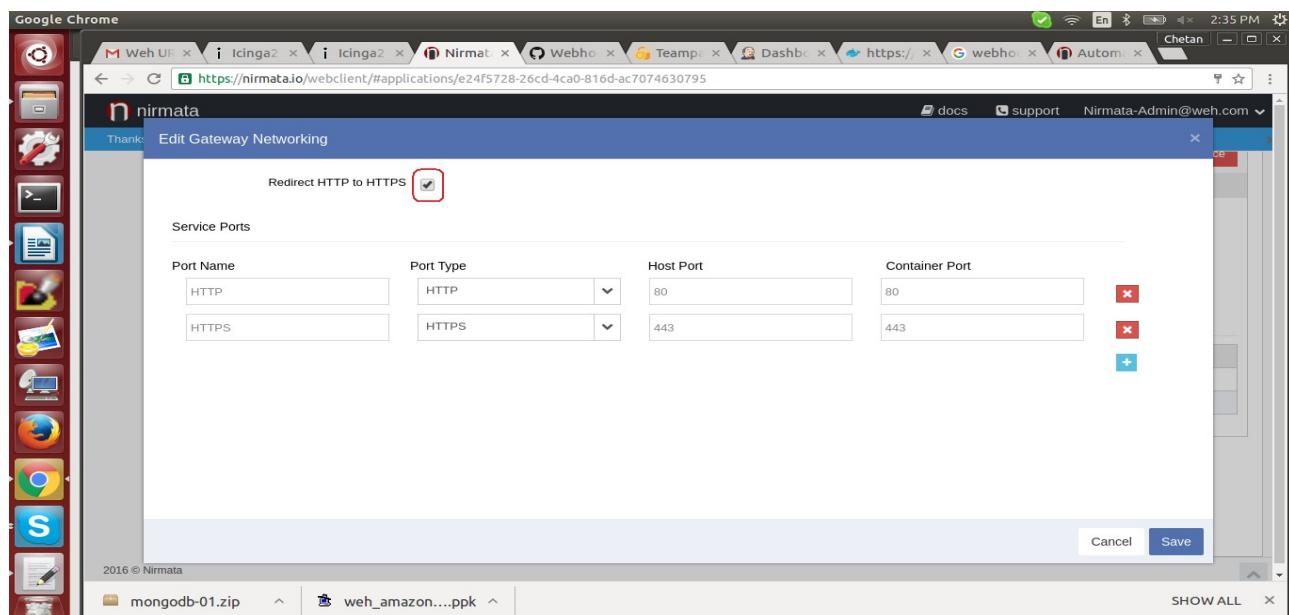
c. Next step is that, we have configure our gateway service for SSL. So, we need to provide environment variables & certificate path as shown in the following fig.

NIRMATA_GW_TLS_CERT = /usr/share/nirmata/ssl/weh.com.crt

NIRMATA_GW_TLS_KEY = /usr/share/nirmata/ssl/weh.com.key



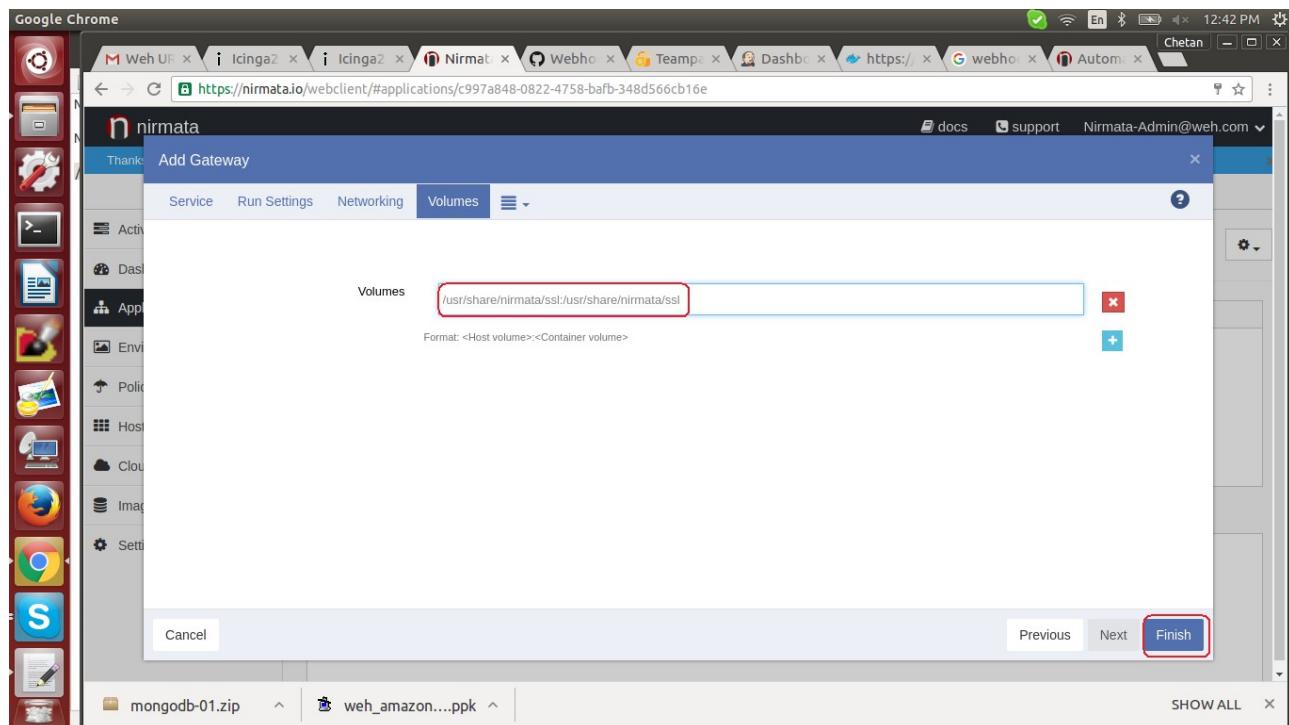
d. Next step is, we have to configured our gateway in such way that whenever request will come from HTTP it will automatically redirect to HTTPS.



e. Next step is, we have to mount ssl certificate for gateway service.

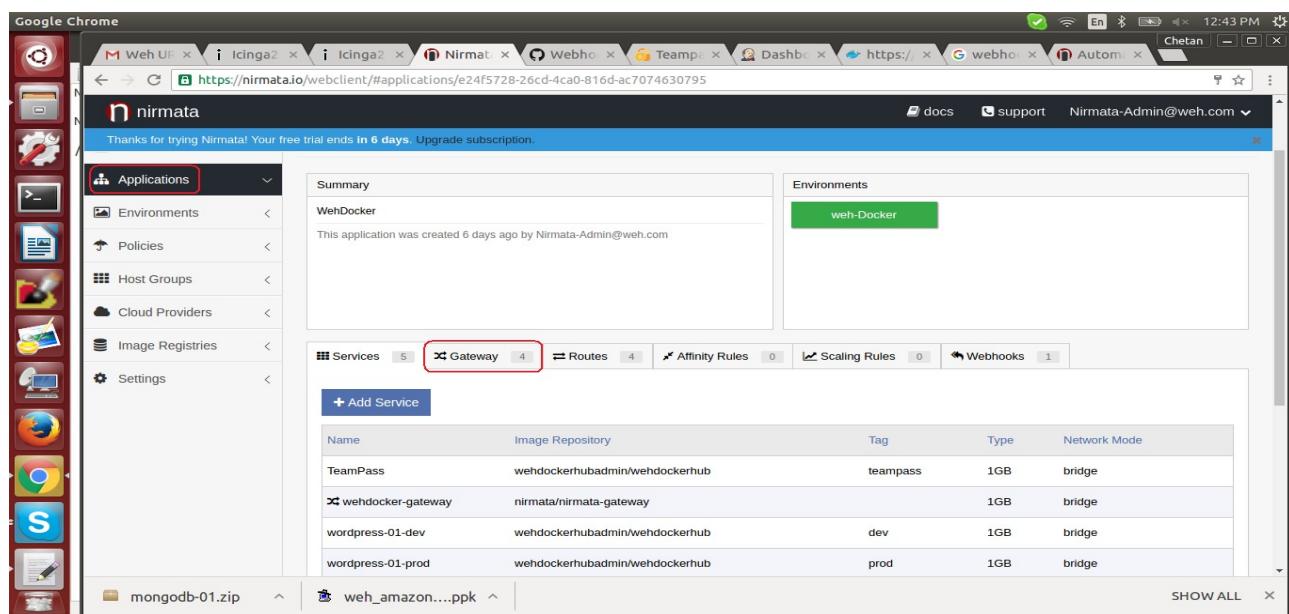
Format: <Host volume>:<Container volume>

ex . /usr/share/nirmata/ssl:/usr/share/nirmata/ssl



3. Adding gateway Route

a. Next step is, adding gateway route for our services. Click on Application --> Gateway --> Add Gateway route . As shown in below fig.



Google Chrome

https://nirmata.io/webclient/#applications/e24f5728-26cd-4ca0-816d-ac7074630795

nirmata

Thanks for trying Nirmata! Your free trial ends in 6 days. Upgrade subscription.

Policies Host Groups Cloud Providers Image Registries Settings

Services 5 Gateway 4 Routes 4 Affinity Rules 0 Scaling Rules 0 Webhooks 1

Add Gateway Route

Type	Route	Service	Port	Target URL	Sticky	Actions
URL Route	/	TeamPass	HTTP:80	/	-	[Edit] [Delete]
DNS Route	prod.weh.local	wordpress-01-prod	HTTPS:443	/	✓	[Edit] [Delete]
DNS Route	dev.weh.local	wordpress-01-dev	HTTPS:443	/	✓	[Edit] [Delete]
DNS Route	qa.weh.local	wordpress-01-qa	HTTPS:443	/	✓	[Edit] [Delete]

SHOW ALL

mongodb-01.zip weh_amazon....ppk

Google Chrome

https://nirmata.io/webclient/#applications/e24f5728-26cd-4ca0-816d-ac7074630795

nirmata

Thanks for trying Nirmata! Your free trial ends in 6 days.

Policies Host Groups Cloud Providers Image Registries Settings

Services 5 Gateway 4 Routes 4 Affinity Rules 0 Scaling Rules 0 Webhooks 1

Edit Gateway Route

Type*: dnsRoute

Route*: dev.weh.local

Service: wordpress-01-dev

Port: HTTP:443

Target URL: /

Sticky:

Cancel Save

Target URL Sticky Actions

/	-	[Edit] [Delete]
/	✓	[Edit] [Delete]
/	✓	[Edit] [Delete]
/	✓	[Edit] [Delete]

SHOW ALL

mongodb-01.zip weh_amazon....ppk

Environments :

- a. Next step is , Creating Environments for our Application. So, that Click On Enviorment --> + Add Environment as below.

The screenshot shows the Nirmata web client interface. The left sidebar has a navigation menu with 'Environments' selected. In the main area, there's a list of environments: 'Idgen' (Running, 0 instances), 'TeamPass' (Failed, 1 instance), and 'weh-Docker' (Running, 5 instances). A red box highlights the '+ Add Environment' button in the top right of the main content area. The bottom of the screen shows a file manager with 'mongodb-01.zip' and 'weh_amazon....ppk' files.

- b. Next, Give name to your Environment & Select your Application From Drop Down list as shown in the following fig.

The screenshot shows the 'Add Environment' dialog box. The 'Environment' tab is selected. The 'Name' field is filled with 'WehDocker'. The 'Type' dropdown is set to 'Dev-Test'. The 'Application' dropdown is set to 'WehDocker'. The 'Version' dropdown is set to 'latest'. At the bottom right, there are 'Previous', 'Next', and 'Finish' buttons, with 'Finish' highlighted by a red box. The bottom of the screen shows a file manager with 'mongodb-01.zip' and 'weh_amazon....ppk' files.

c. As you created your Environment, You will see your services are running.

The screenshot shows the Nirmata web client interface. On the left is a vertical sidebar with various icons for configuration, monitoring, and management. The main area displays a table titled "Service Instances" under the "Analytics" tab. The table lists five service instances:

Container	Service	Tag	Host	IP Address	State	Actions
TeamPass_46104	TeamPass	teampass	6a25fed2-d0	172.16.102.35.64534	Running	[Edit]
wordpress-01-dev.87788	wordpress-01-dev	dev	6a25fed2-d0	172.16.102.35.64644	Running	[Edit]
wehdocker-gateway.79608	wehdocker-gateway	latest	6a25fed2-d0	172.16.102.35.443	Running	[Edit]
wordpress-01-qa.56381	wordpress-01-qa	qa	6a25fed2-d0	172.16.102.35.52856	Running	[Edit]
wordpress-01-prod.28031	wordpress-01-prod	prod	6a25fed2-d0	172.16.102.35.57524	Running	[Edit]

At the bottom of the interface, there are download links for "mongodb-01.zip" and "weh_amazon....ppk".

d. Next step is , We have to add Service gate in environment. As shown in the following fig click on Edit gateway service button.

The screenshot shows the Nirmata web client interface, specifically the "Environments" section for "weh-Docker". The sidebar on the left shows the "Environments" option is selected. The main area displays the environment details for "weh-Docker" (Dev-Test, WehDocker [latest]). It includes performance metrics for Avg CPU (%) and Avg Memory (MB), and a table of service instances.

Performance Metrics:

- Avg CPU (%): 0 to 100 scale
- Avg Memory (MB): 0 to 5,000 scale

Service Instances (Table):

Container	Service	Tag	Host	IP Address	State	Actions
TeamPass_46104	TeamPass	teampass	6a25fed2-d0	172.16.102.35.64534	Running	[Edit]

At the bottom of the interface, there are download links for "mongodb-01.zip" and "weh_amazon....ppk".

e. As shown in following fig. Add gateway routes in your Environments & make sure to enable Redirect HTTP to HTTPS.

The screenshot shows the Nirmata web client interface. The main window displays the 'Edit Gateway Routes' dialog. At the top left of this dialog, there is a checkbox labeled 'Redirect HTTP' which is checked. Below this, it says 'Redirect HTTP requests to HTTPS'. The main area contains a table with four rows of gateway routes:

Type	Route	To Service	Tag	Port	Target URL	Sticky
urlRoute	/	TeamPass	teampass	HTTP:80	/	<input checked="" type="checkbox"/>
dnsRoute	dev.weh.local	wordpress-01-dev	dev	HTTPS:443	/	<input checked="" type="checkbox"/>
dnsRoute	prod.weh.local	wordpress-01-prod	prod	HTTPS:443	/	<input checked="" type="checkbox"/>

At the bottom right of the dialog, there are 'Cancel' and 'Save' buttons. The 'Save' button is highlighted with a red box. The background of the interface shows a sidebar with various icons and a terminal-like bottom panel with some log entries.

7. Testing of Complete PipeLine.

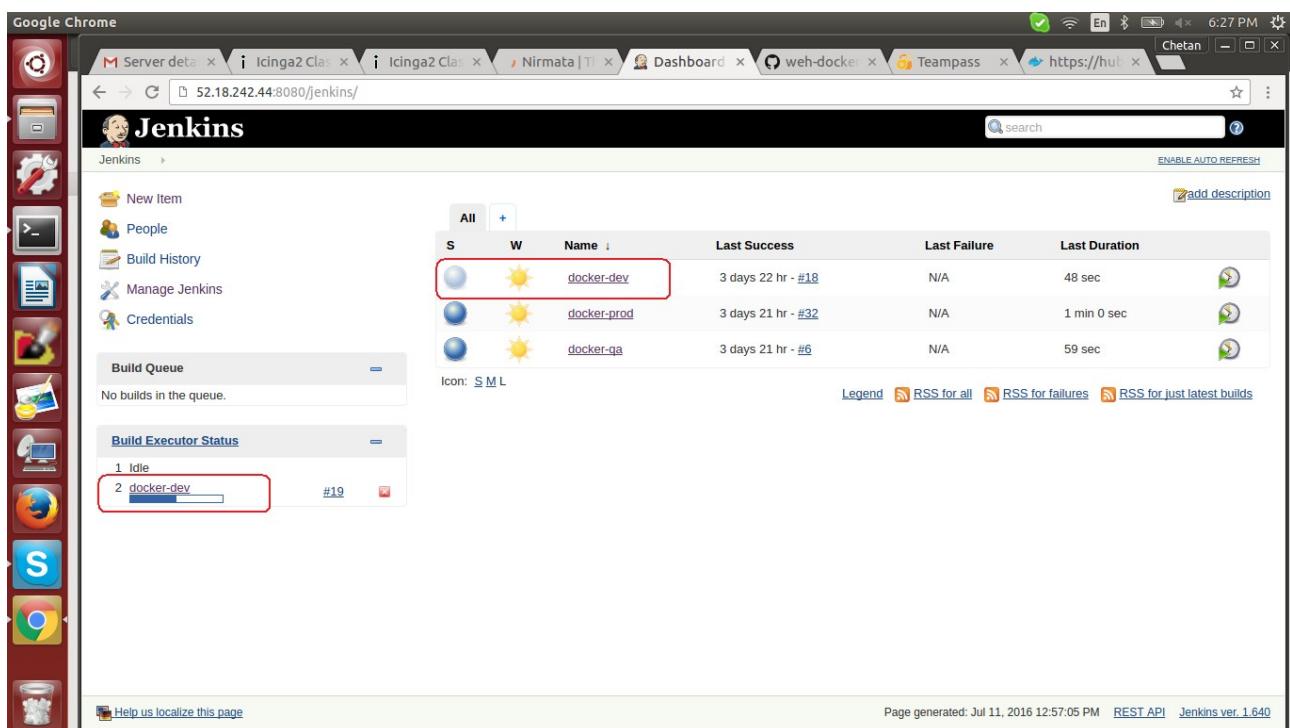
1. First step is, Clone wordpress filesystem from github i.e. dev

```
# git clone -b dev https://github.com/sachinkarale/weh-docker-wordpress.git
# cd weh-docker-wordpress
```

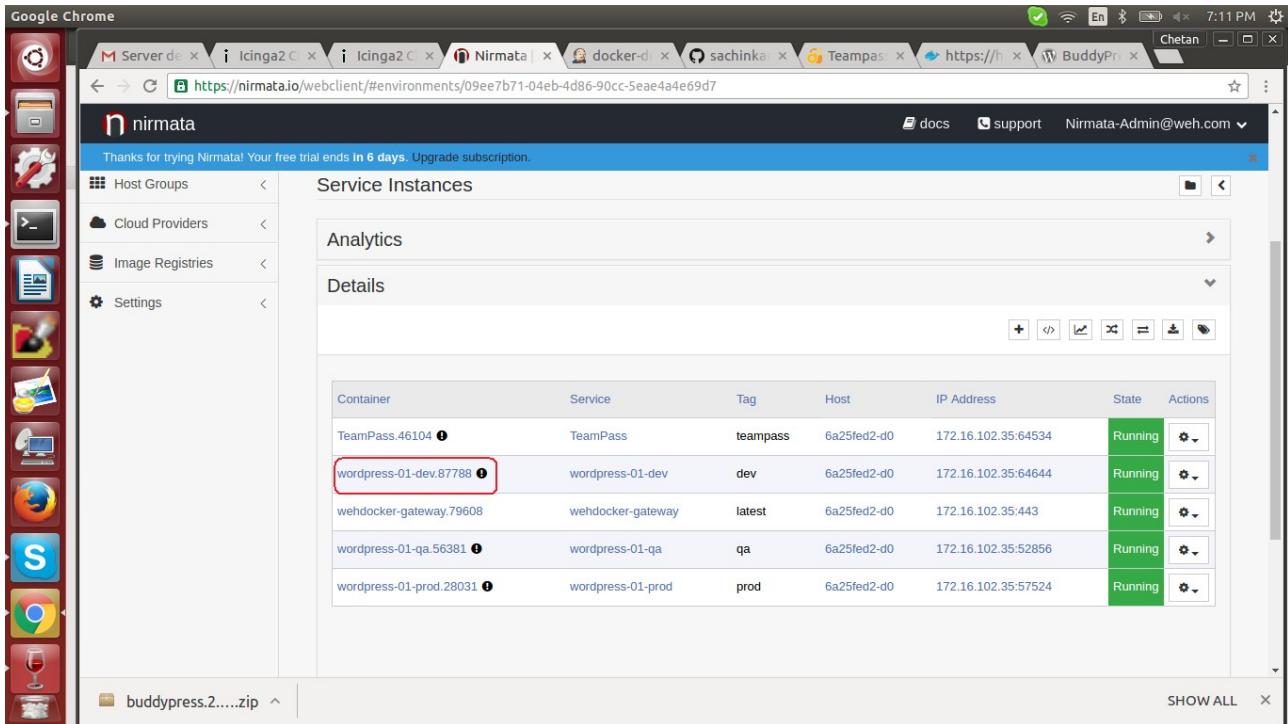
Copy plugins zip file into plugins folder using below command.

```
# cp /opt/woocommerce.2.6.2.zip .
# git add *
# git commit -a -m "some message"
# git push -u origin dev
```

2. As you entered git push command , jenkins will build jobs automatically as shown in the following fig.



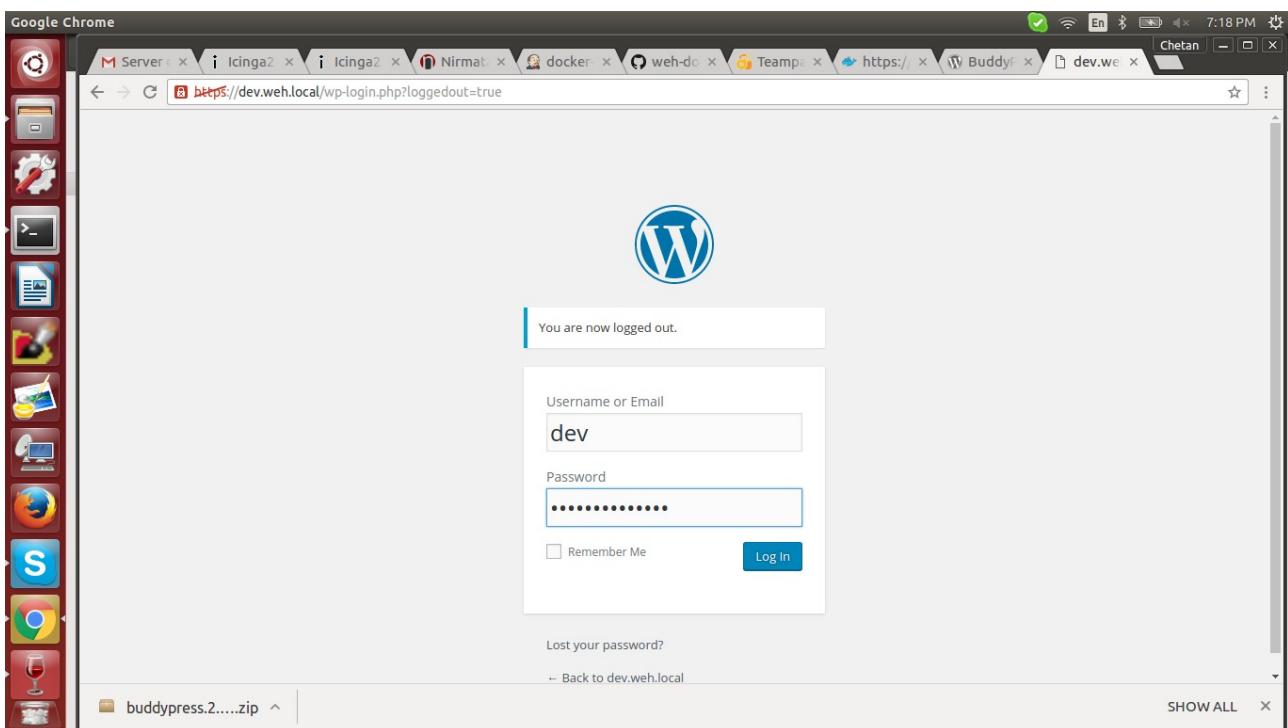
3. Once jenkins build job successfully, Nirmata will redeploy our docker container with newly added changes.



The screenshot shows the Nirmata web client interface. On the left is a sidebar with icons for Host Groups, Cloud Providers, Image Registries, and Settings. The main area has tabs for Analytics and Details. Under Details, there is a table titled 'Service Instances' showing the following data:

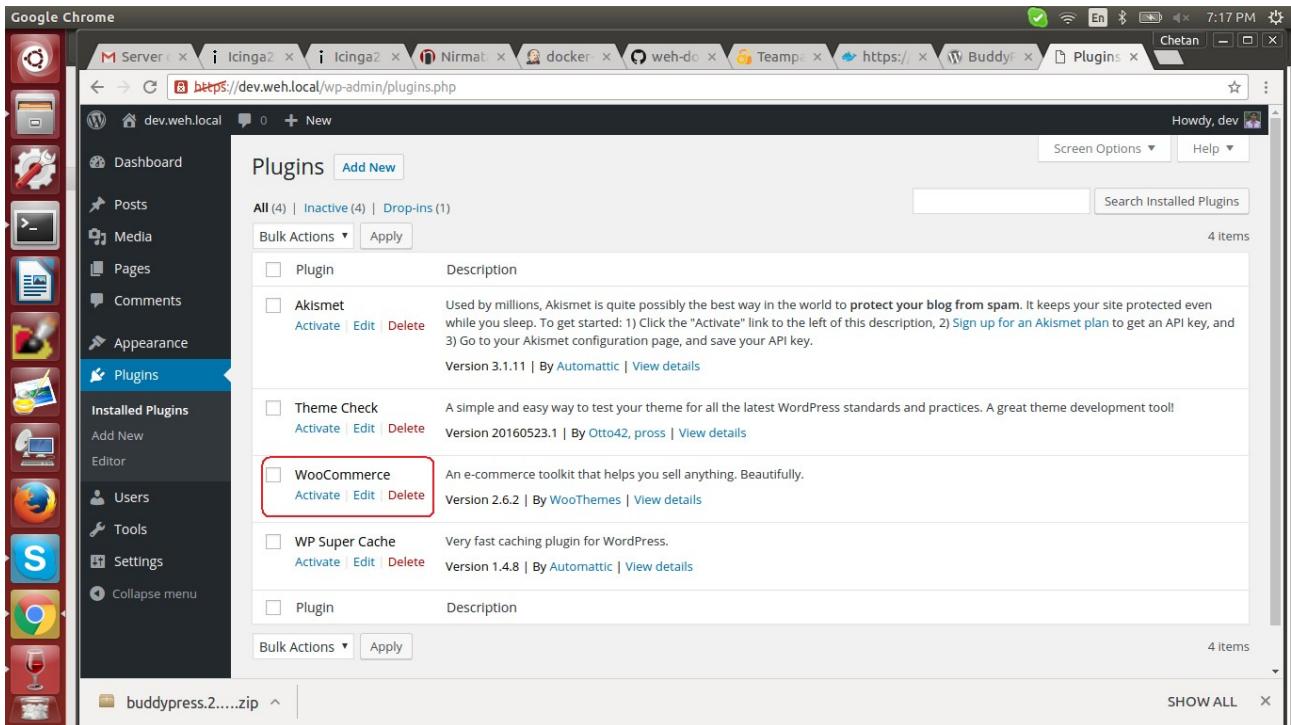
Container	Service	Tag	Host	IP Address	State	Actions
TeamPass.46104	TeamPass	teampass	6a25fed2-d0	172.16.102.35:64534	Running	[Action]
wordpress-01-dev.87788	wordpress-01-dev	dev	6a25fed2-d0	172.16.102.35:64644	Running	[Action]
wehdocker-gateway.79608	wehdocker-gateway	latest	6a25fed2-d0	172.16.102.35:443	Running	[Action]
wordpress-01-qa.56381	wordpress-01-qa	qa	6a25fed2-d0	172.16.102.35:52856	Running	[Action]
wordpress-01-prod.28031	wordpress-01-prod	prod	6a25fed2-d0	172.16.102.35:57524	Running	[Action]

4. verify this by login into wordpress application as shown in the following fig.



The screenshot shows a browser window with the URL <https://dev.weh.local/wp-login.php?loggedout=true>. The page displays the WordPress logo at the top. Below it, a message says "You are now logged out." A login form is present with fields for "Username or Email" containing "dev" and "Password" (redacted). There is a "Remember Me" checkbox and a "Log In" button. At the bottom of the form, there are links for "Lost your password?" and "← Back to dev.weh.local".

You can see, we have added **wooCommerce** plugins.



The screenshot shows a Google Chrome browser window with the address bar pointing to <https://dev.weh.local/wp-admin/plugins.php>. The WordPress admin sidebar is visible on the left, with 'Plugins' selected. The main content area displays a list of installed plugins:

Plugin	Description
Akismet	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam. It keeps your site protected even while you sleep. To get started: 1) Click the "Activate" link to the left of this description, 2) Sign up for an Akismet plan to get an API key, and 3) Go to your Akismet configuration page, and save your API key. Version 3.1.11 By Automattic View details
Theme Check	A simple and easy way to test your theme for all the latest WordPress standards and practices. A great theme development tool! Version 20160523.1 By Otto42, pross View details
WooCommerce	An e-commerce toolkit that helps you sell anything. Beautifully. Version 2.6.2 By WooThemes View details
WP Super Cache	Very fast caching plugin for WordPress. Version 1.4.8 By Automattic View details
Plugin	Description

A red box highlights the 'WooCommerce' plugin entry. At the bottom of the page, there is a file download indicator for 'buddypress.2.....zip'.