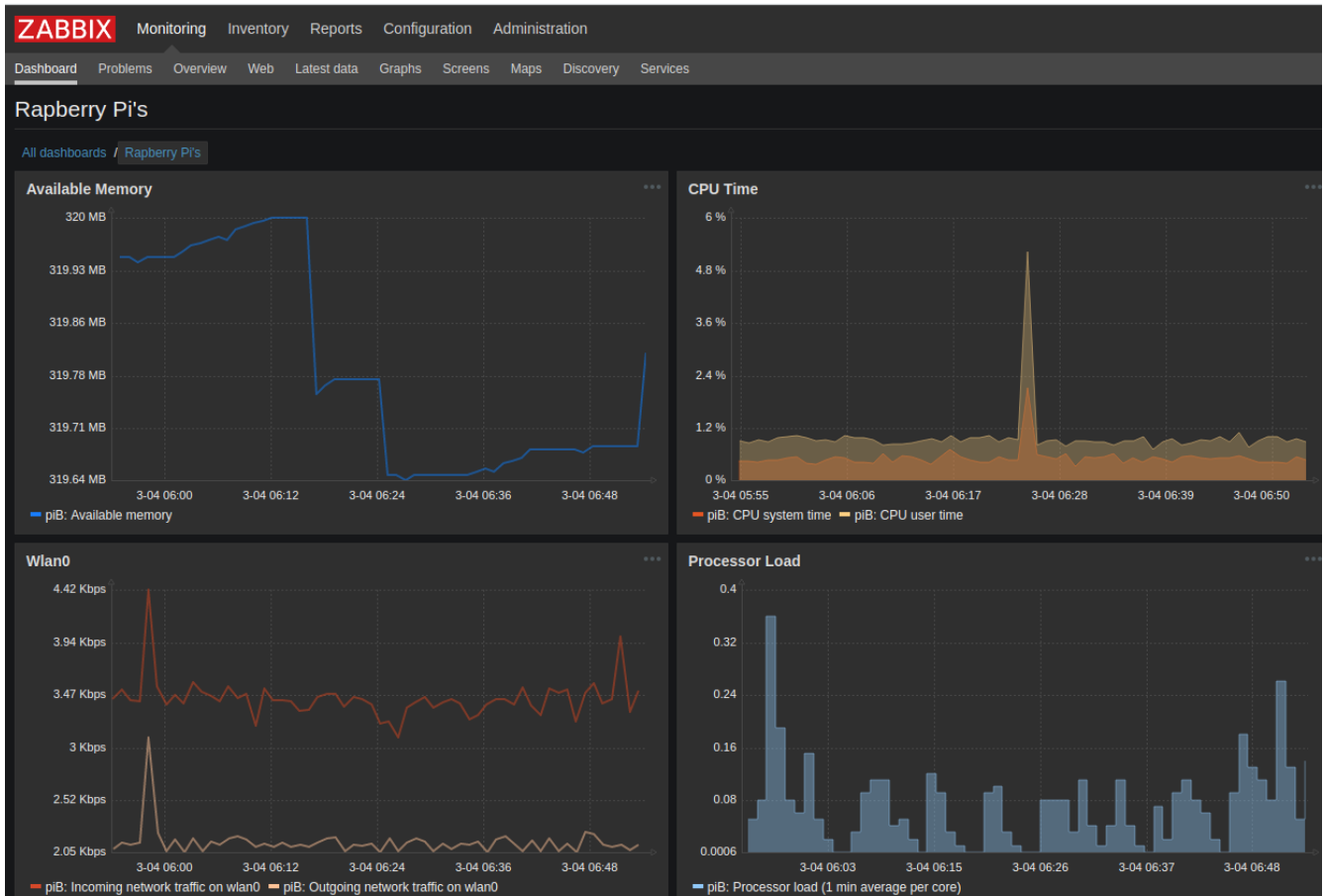


Raspberry Pi 'How-To' Series

Zabbix Agent Installation Guide



Written by: Sopwith
Revision 1.0
March 4, 2019
sopwith@ismellsmoke.net

"If it works out of the box – what fun is that?"

Introduction

[Zabbix](#) is a popular open-source platform used by IT professionals all over the world to monitor their security infrastructure. The [list](#) of companies that use the platform is impressive.



Figure 1: Zabbix dashboard

In a recent 'How-To', I provided detailed information to get a Zabbix server up and running on a Raspberry Pi. In this document, I will show how to install a Zabbix agent on a Raspberry Pi. Once installed, you can monitor the performance of your Pi.

Step-by-Step

In this guide, I will install the Zabbix agent on an original Raspberry Pi B.

There are 6 steps to installing Zabbix and an agent.

1. Get your Pi up and running with Raspian Stretch
2. Install the Zabbix server platform
3. Install the Zabbix agent on a Pi
4. Configure the Zabbix agent
5. Create and configure a Zabbix server dashboard
6. Add widgets to the dashboard

Step-1 - Get your Pi up and running with Raspian Stretch.

The first thing you need to do is get your Pi running. Head out to raspberrypi.org and download *Raspian Stretch* or *Stretch Lite*. The former has a Window GUI while the latter is a command-line only OS. Either one works. I will be installing *Stretch Lite*.

Once the download completes, burn the image to an SD card and fire up your Pi. If you need help with this step, there is a ton of resources on the web to help you. When you are up and running and connected to the Internet, be sure to update your Pi with the latest patches.

```
$ sudo apt-get update  
$ sudo apt-get upgrade  
$ sudo reboot
```

Step-2 – Install the Zabbix server

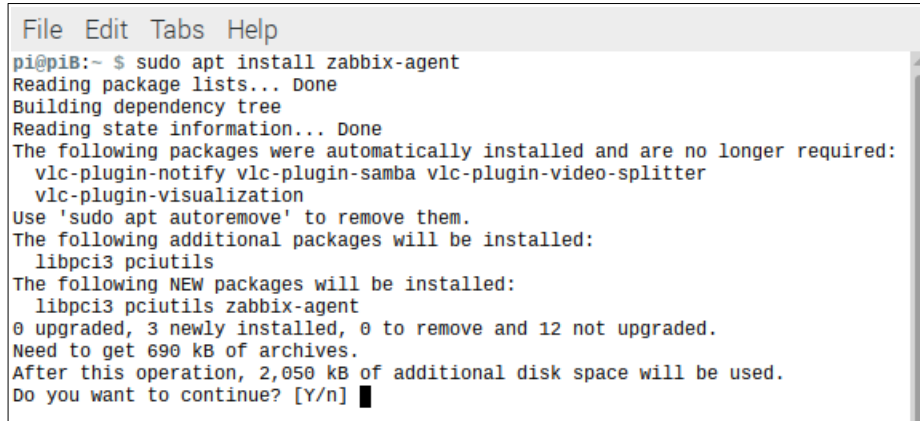
Install the Zabbix server platform. All the major OS platforms are supported so you can install the server anywhere you like. I installed it on a Raspberry Pi 3. You can download the Zabbix Server [‘How-To’ here](#).

Step-3 – Install the Zabbix agent

Installing the Zabbix agent on a Pi is a snap. Type in the below command and all the magic happens.

```
$ sudo apt install zabbix-agent
```

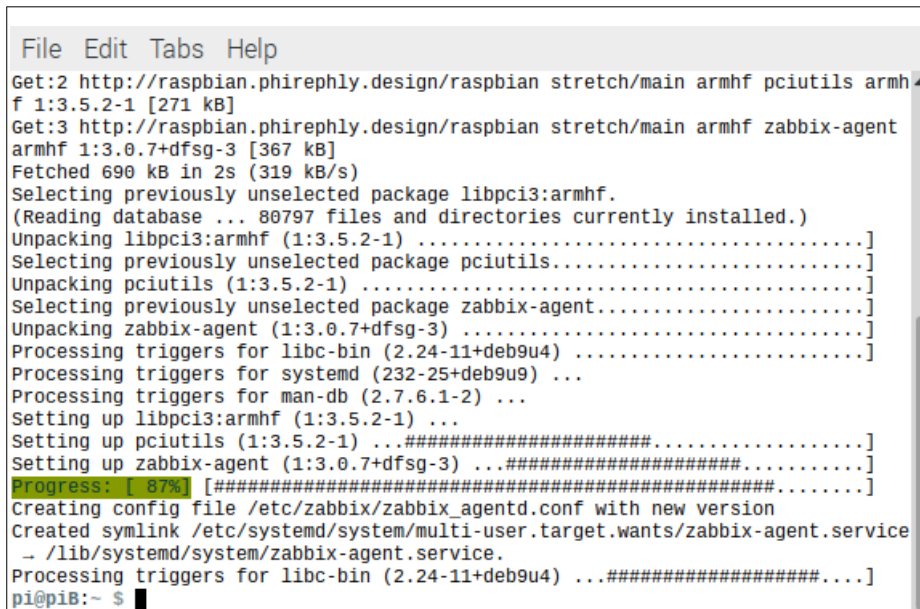
This is shown below in Figure 2.



```
File Edit Tabs Help
pi@piB:~ $ sudo apt install zabbix-agent
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  vlc-plugin-notify vlc-plugin-samba vlc-plugin-video-splitter
  vlc-plugin-visualization
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libpci3 pciutils
The following NEW packages will be installed:
  libpci3 pciutils zabbix-agent
0 upgraded, 3 newly installed, 0 to remove and 12 not upgraded.
Need to get 690 kB of archives.
After this operation, 2,050 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Figure 2: Install Zabbix agent

The required packages will be downloaded and installed as shown in Figure 3.




```
File Edit Tabs Help
Get:2 http://raspbian.phirephly.design/raspbian stretch/main armhf pciutils armh
f 1:3.5.2-1 [271 kB]
Get:3 http://raspbian.phirephly.design/raspbian stretch/main armhf zabbix-agent
armhf 1:3.0.7+dfsg-3 [367 kB]
Fetched 690 kB in 2s (319 kB/s)
Selecting previously unselected package libpci3:armhf.
(Reading database ... 80797 files and directories currently installed.)
Unpacking libpci3:armhf (1:3.5.2-1) .....
Selecting previously unselected package pciutils.....
Unpacking pciutils (1:3.5.2-1) .....
Selecting previously unselected package zabbix-agent.....
Unpacking zabbix-agent (1:3.0.7+dfsg-3) .....
Processing triggers for libc-bin (2.24-11+deb9u4) .....
Processing triggers for systemd (232-25+deb9u9) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up libpci3:armhf (1:3.5.2-1) ...
Setting up pciutils (1:3.5.2-1) .....
Setting up zabbix-agent (1:3.0.7+dfsg-3) .....
Progress: [ 87%] .....
Creating config file /etc/zabbix/zabbix_agentd.conf with new version
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service
→ /lib/systemd/system/zabbix-agent.service.
Processing triggers for libc-bin (2.24-11+deb9u4) .....
pi@piB:~ $
```

Figure 3: Zabbix agent installed

Step-4 – Configure the Zabbix agent

Once the agent is installed we need to configure it. Change directories to /etc/zabbix as shown in Figure 4.



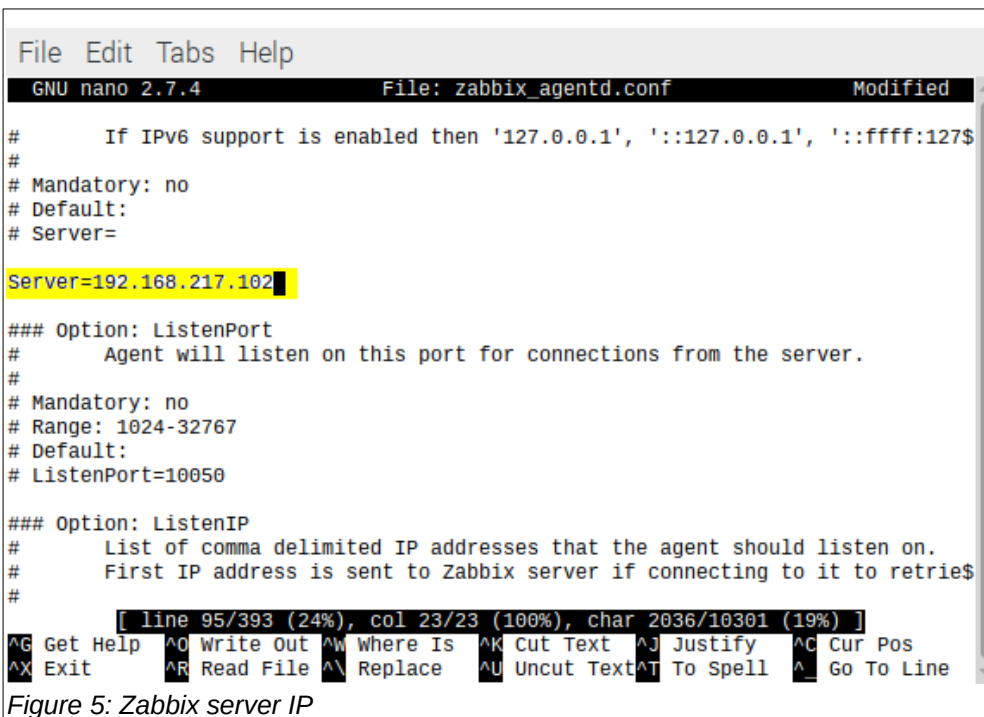
```
File Edit Tabs Help
pi@piB:~ $ cd /etc/zabbix/
pi@piB:/etc/zabbix $ ls -l
total 16
-rw-r--r-- 1 root root 10295 Jun  4 2017 zabbix_agentd.conf
drwxr-xr-x 2 root root 4096 Jun  4 2017 zabbix_agentd.conf.d
pi@piB:/etc/zabbix $
```

Figure 4: Zabbix agent configuration file location

Open the agent configuration file for editing:

```
$ sudo nano zabbix-agent.conf
```

Scroll down to the section labeled '### Option: Server'. Enter the IP address of your Zabbix server. This is shown below in Figure 5.



```
File Edit Tabs Help
GNU nano 2.7.4 File: zabbix_agentd.conf Modified

# If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1'
#
# Mandatory: no
# Default:
# Server=
Server=192.168.217.102

### Option: ListenPort
# Agent will listen on this port for connections from the server.
#
# Mandatory: no
# Range: 1024-32767
# Default:
# ListenPort=10050

### Option: ListenIP
# List of comma delimited IP addresses that the agent should listen on.
# First IP address is sent to Zabbix server if connecting to it to retrieve
#
[ line 95/393 (24%), col 23/23 (100%), char 2036/10301 (19%) ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Figure 5: Zabbix server IP

Next, scroll down to the section labelled '### Option: Hostname'.

Here you want to enter a hostname the server will use to identify this particular agent. The hostname does *not* have to be the same as the computer hostname, although in my case I did use the same name. My PiB hostname is piB and that is the name I configured the agent to use.

NOTE: This item is very important. You must use this name when you configure the server or it will never find this agent.

The entry in my agent configuration file is shown below in Figure 6.

```

File Edit Tabs Help
GNU nano 2.7.4 File: zabbix_agentd.conf Modified

# Default:
# ServerActive=

ServerActive=127.0.0.1

### Option: Hostname
# Unique, case sensitive hostname.
# Required for active checks and must match hostname as configured on the$
# Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=piB

### Option: HostnameItem
# Item used for generating Hostname if it is undefined. Ignored if Hostna$
# Does not support UserParameters or aliases.
#
[ line 147/393 (37%), col 13/13 (100%), char 3513/10291 (34%) ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Figure 6: Zabbix agent hostname

Finally, make sure the Agent will allow remote commands from the server. In the section labelled '### Option: EnableRemoteCommands', add an entry with the value of 1 as shown below in Figure 7.

```

File Edit View Search Terminal Help
GNU nano 2.7.4 File: zabbix_agentd.conf Modified

#
# Mandatory: no
# Default:
# SourceIP=

### Option: EnableRemoteCommands
# Whether remote commands from Zabbix server are allowed.
# 0 - not allowed
# 1 - allowed
#
# Mandatory: no
# Default:
# EnableRemoteCommands=0

EnableRemoteCommands=1

### Option: LogRemoteCommands
# Enable logging of executed shell commands as warnings.
# 0 - disabled

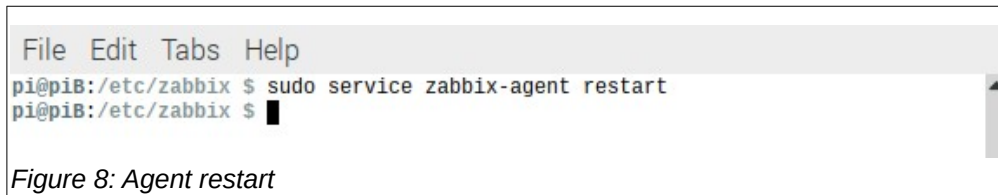
```

Figure 7: Enable remote commands

Save the configuration file <Ctrl> o <Enter> <Ctrl> x.

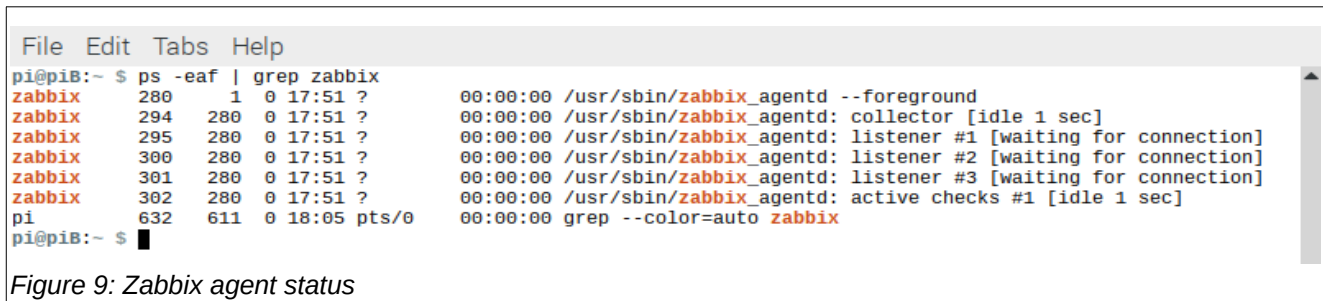
Since we made changes to the configuration file, we need to restart the agent so it is aware of the changes.
\$ sudo service zabbix-agent restart

This is shown below in Figure 8.



To be sure the Zabbix agent is running, enter the below command:
\$ ps -eaf | grep zabbix

The output of this command should look similar to Figure 9.

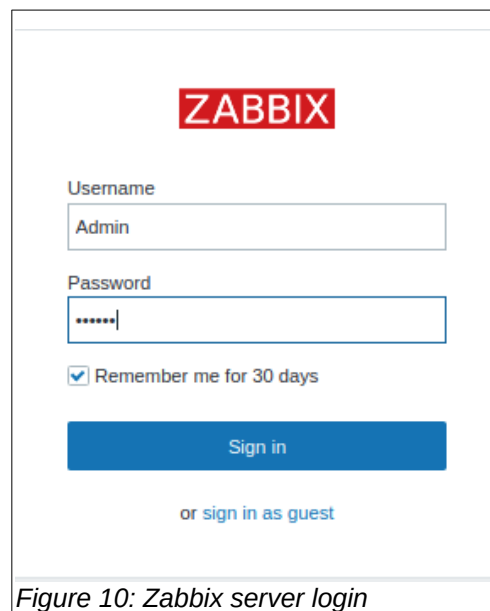


Notice the agent has three listeners running. They are awaiting instructions from a server.

Now that the agent is configured, let's make some charts on the server!

Step-5 – Create and configure a Zabbix server dashboard

Open a web browser and login to your Zabbix server. Figure 10. Remember, the default credentials are:
Username: Admin
Password: zabbix



If you installed an agent on your Zabbix server, you will see a dashboard named 'Zabbix server health' in the dashboard Global view. Figure 11.

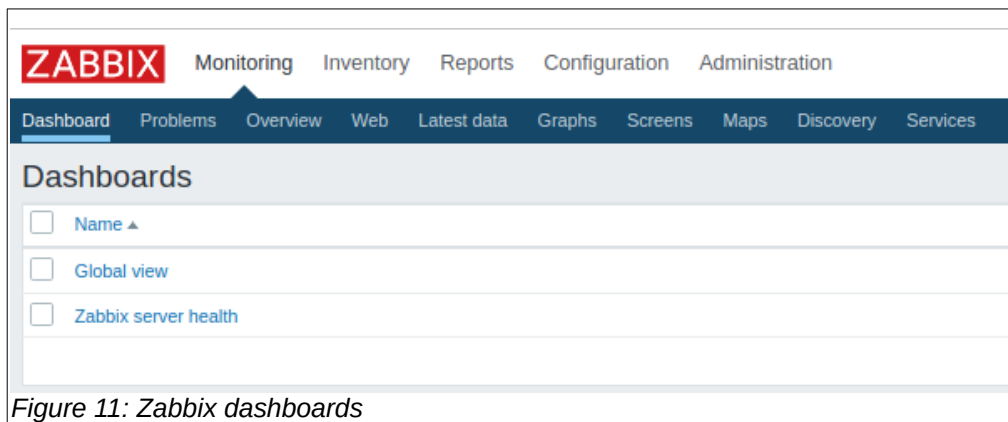


Figure 11: Zabbix dashboards

If you select this dashboard, you are presented with a series of charts that show you the health of your Zabbix server. This is shown in Figure 12.



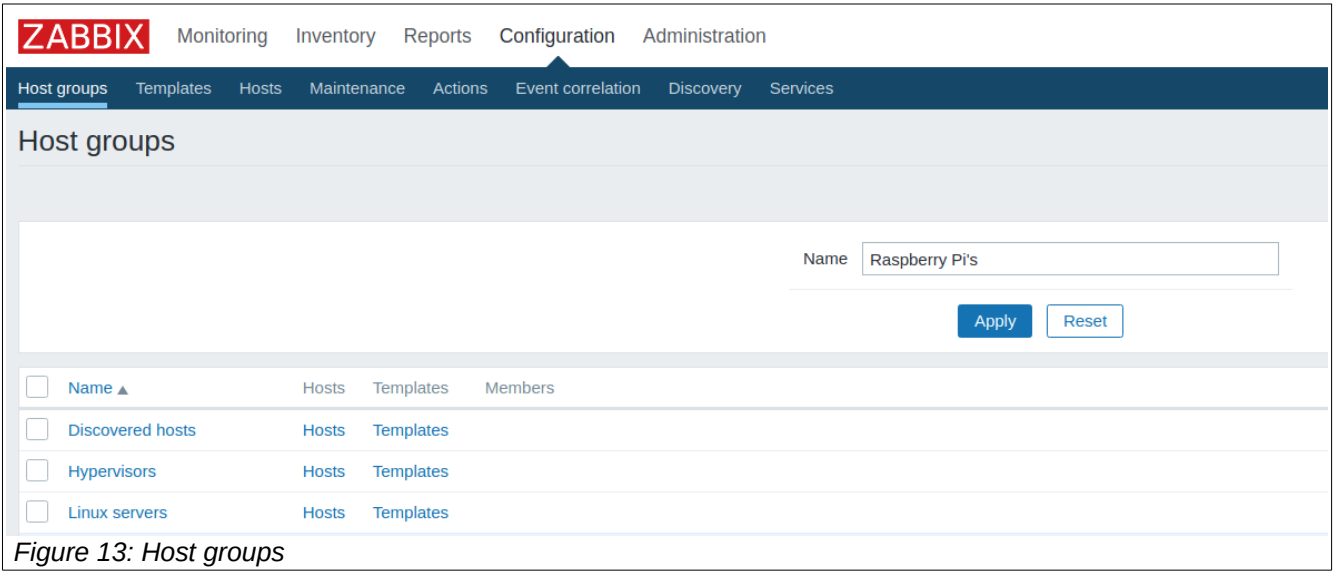
Figure 12: Zabbix server health dashboard

The top panel allows you to select the time-frame you want to view. You can hide this panel by clicking on the small clock in the upper right tab. The second panel will show you any alerts. To the right of this panel is a clock that shows the local time.

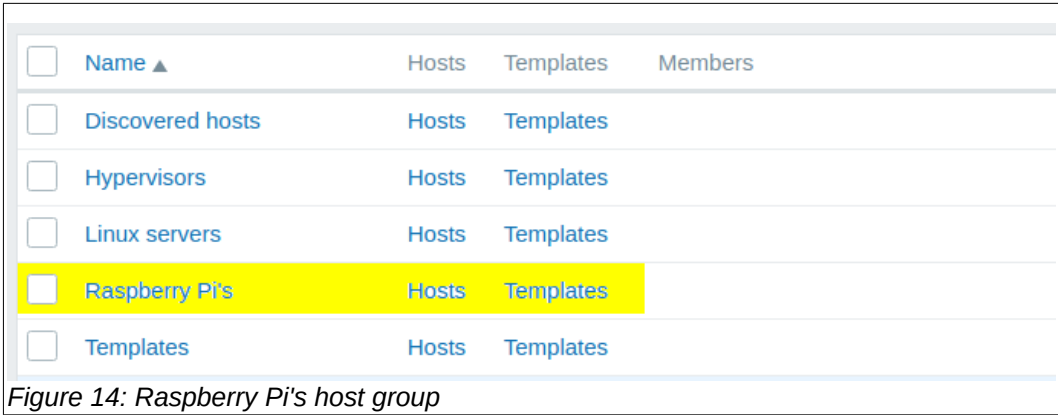
The lower part of the dashboard contains a series of charts that shows all kinds of interesting metrics about the server. What we are going to do is build a dashboard that shows some wicked cool information about our Raspberry Pi.

The Zabbix platform is very structured which makes it easy to use and very flexible. It uses a concept of host groups and hosts. All hosts must belong to a host group. So, let's create a host group. Click on 'Configuration | Host groups'. Add a host group named Raspberry Pi's. This is where we will manage our Pi

fleet. Figure 13.



Once added, you will see the new group in the list, Figure 14.



Next, we need to create a host entry for our Raspberry Pi. Click on 'Configuration | Hosts'. In the upper right corner click on the 'Create Host' button. You will see a screen shown in Figure 15.

ZABBIX Monitoring Inventory Reports **Configuration** Administration

Host groups Templates **Hosts** Maintenance Actions Event correlation Discovery Services

Hosts

Host Templates IPMI Macros Host inventory Encryption

* Host name

Visible name

* Groups

* At least one interface must exist.

Agent interfaces	IP address	DNS name	Connect to	Port	Default
<input type="text" value="127.0.0.1"/>	<input type="text"/>	<input type="button" value="IP"/> <input type="button" value="DNS"/>	<input type="text" value="10050"/>	<input checked="" type="radio"/> Remove	

[Add](#)

SNMP interfaces [Add](#)

JMX interfaces [Add](#)

IPMI interfaces [Add](#)

Description

Monitored by proxy

Enabled ☒

Figure 15: Add a host

It is critically important that the Host name in this form is the same as the Host name you entered in the agent configuration file on your Pi. In my case, I named in 'piB' so I will use that name. My completed form is shown below in Figure 16.

* Host name

Visible name

* Groups

* At least one interface must exist.

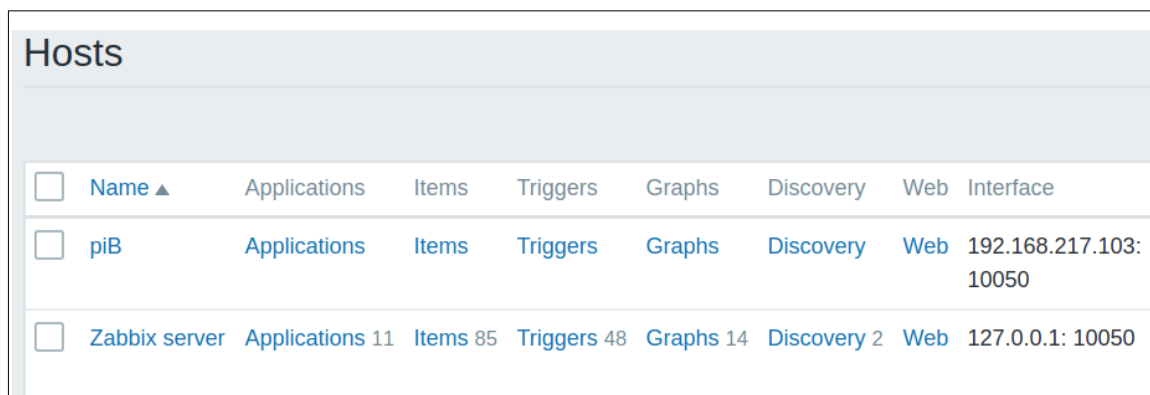
Agent interfaces	IP address	DNS name	Connect to	Port	Default
<input type="text" value="192.168.217.103"/>	<input type="text"/>	<input type="button" value="IP"/> <input type="button" value="DNS"/>	<input type="text" value="10050"/>	<input checked="" type="radio"/> Remove	

[Add](#)

Figure 16: Host configuration

My configuration is quite minimal. The 'Host name' matches the agent Host name. The 'Visible name' is the same. It will belong to the Raspberry Pi's Group, and it can be found at the IP address listed. I did not fill out any of the other fields on the form.

Once added, the piB host now shows in the Hosts list, Figure 17.



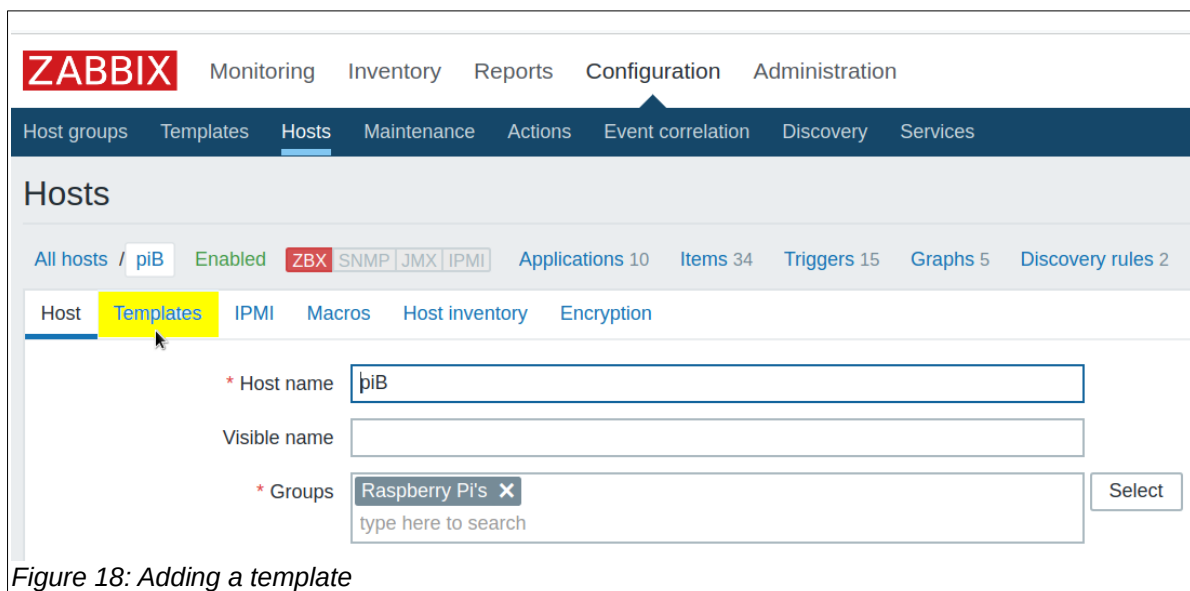
The screenshot shows the Zabbix 'Hosts' configuration page. At the top, there's a header 'Hosts'. Below it is a table with columns: Name, Applications, Items, Triggers, Graphs, Discovery, Web, and Interface. The first row is for 'piB', which is enabled (checkbox checked) and has a status of 'Enabled'. It shows 11 Applications, 85 Items, 48 Triggers, 14 Graphs, 2 Discovery rules, and a Web interface at 192.168.217.103:10050. The second row is for 'Zabbix server', which is also enabled and shows 11 Applications, 85 Items, 48 Triggers, 14 Graphs, 2 Discovery rules, and a Web interface at 127.0.0.1:10050.

<input type="checkbox"/>	Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface
<input checked="" type="checkbox"/>	piB	Applications	Items	Triggers	Graphs	Discovery	Web	192.168.217.103:10050
<input checked="" type="checkbox"/>	Zabbix server	Applications 11	Items 85	Triggers 48	Graphs 14	Discovery 2	Web	127.0.0.1: 10050

Figure 17: piB host

Now that the Host has been added, the next thing we need to do is assign the host some Items. Items in the Zabbix platform are the things we would like to monitor on a host. One of the really cool features of Zabbix is that it comes with a comprehensive set of pre-built templates so we do not have to build an item list.

For our Pi, lets use the pre-built 'OS Linux' template. To do so, we need to add this template to our host configuration. From the 'Hosts' configuration panel, click on your Pi, Figure 18.



The screenshot shows the Zabbix 'Hosts' configuration page for the 'piB' host. The 'Hosts' tab is selected in the top navigation bar. Below the 'Hosts' header, there's a sub-header 'All hosts / piB' with a status 'Enabled' and a list of templates: ZBX, SNMP, JMX, IPMI. Below this, there's a list of statistics: Applications 10, Items 34, Triggers 15, Graphs 5, and Discovery rules 2. The 'Templates' tab is selected in the sub-header. The main form area shows the 'Host name' field with 'piB', the 'Visible name' field, and the 'Groups' field with 'Raspberry Pi's' selected. There's a 'Select' button next to the 'Groups' field.

ZABBIX Monitoring Inventory Reports Configuration Administration

Host groups Templates **Hosts** Maintenance Actions Event correlation Discovery Services

Hosts

All hosts / piB Enabled ZBX SNMP JMX IPMI Applications 10 Items 34 Triggers 15 Graphs 5 Discovery rules 2

Host Templates IPMI Macros Host inventory Encryption

* Host name piB

Visible name

* Groups Raspberry Pi's X type here to search Select

Figure 18: Adding a template

In the 'Templates' panel, add the 'OS Linux' template by following the below steps:

1. In the Search field, select the 'TemplateOS Linux' template.
2. Click the 'Add' link.
3. Click the 'Update' button

Now when you go back to the 'Hosts' panel, you will see that the template has been added to your Pi Host. This is shown in Figure 19.

You may also notice the 'Status' field in this panel shows 'Enabled' in green text. This indicates that the Zabbix server has established communication with the piB Host.

<input type="checkbox"/> Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Templates	Status	Availability
<input type="checkbox"/> piB	Applications 10	Items 43	Triggers 17	Graphs 8	Discovery 2	Web 192.168.217.105:10050		Template OS Linux (Template App Zabbix Agent)	Enabled	ZBX SNMP

Figure 19: Host template

Now that the template has been added to the Host, if you click on the piB Host entry you will discover there are now 10 Applications, 43 Items, 17 Triggers, and 8 Graphs available for use. This makes monitoring your Pi a piece of cake. (Ok – sorry for that dumb analogy). See Figure 20.

Host groups

Templates

Hosts

Maintenance

Actions

Event correlation

Discovery

Services

Graphs

Groupall

HostpiB

All hosts / piB

Enabled

ZBX

SNMP

JMX

IPMI

Applications 10

Items 43

Triggers 17

Graphs 8

Discovery rules 2

Web scenarios

<input type="checkbox"/>	Name ▲	Width
<input type="checkbox"/>	Template OS Linux: CPU jumps	900
<input type="checkbox"/>	Template OS Linux: CPU load	900
<input type="checkbox"/>	Template OS Linux: CPU utilization	900
<input type="checkbox"/>	Mounted filesystem discovery: Disk space usage /	600
<input type="checkbox"/>	Template OS Linux: Memory usage	900
<input type="checkbox"/>	Network interface discovery: Network traffic on eth0	900
<input type="checkbox"/>	Network interface discovery: Network traffic on wlan0	900
<input type="checkbox"/>	Template OS Linux: Swap usage	600

Figure 20: Template Items

Next, let's create a dashboard and add some graphs. Click on 'Monitoring | Dashboard' and select 'All dashboards.' Click on the 'Create dashboard' button shown in Figure 21.

ZABBIX

Monitoring

Inventory

Reports

Configuration

Administration

Support

Share

?

Dashboard

Problems

Overview

Web

Latest data

Graphs

Screens

Maps

Discovery

Services

Dashboards

Create dashboard

☐ Name ▲

☐ Global view

☐ Zabbix server health

Figure 21: Create dashboard

Give the dashboard a name and click on the 'Apply' button. This will add a new dashboard. Figure 22.

Dashboard properties

* Owner: Admin (Zabbix Administrator) [Select]

* Name: Raspberry Pi's

[Apply] [Cancel]

Figure 22: Dashboard properties

Step-6 – Add widgets to the dashboard

Zabbix dashboards contain 'Widgets.' You add a Widget by clicking on the '+ Add widget' button or clicking on the 'Add a new widget' box. Figure 23.

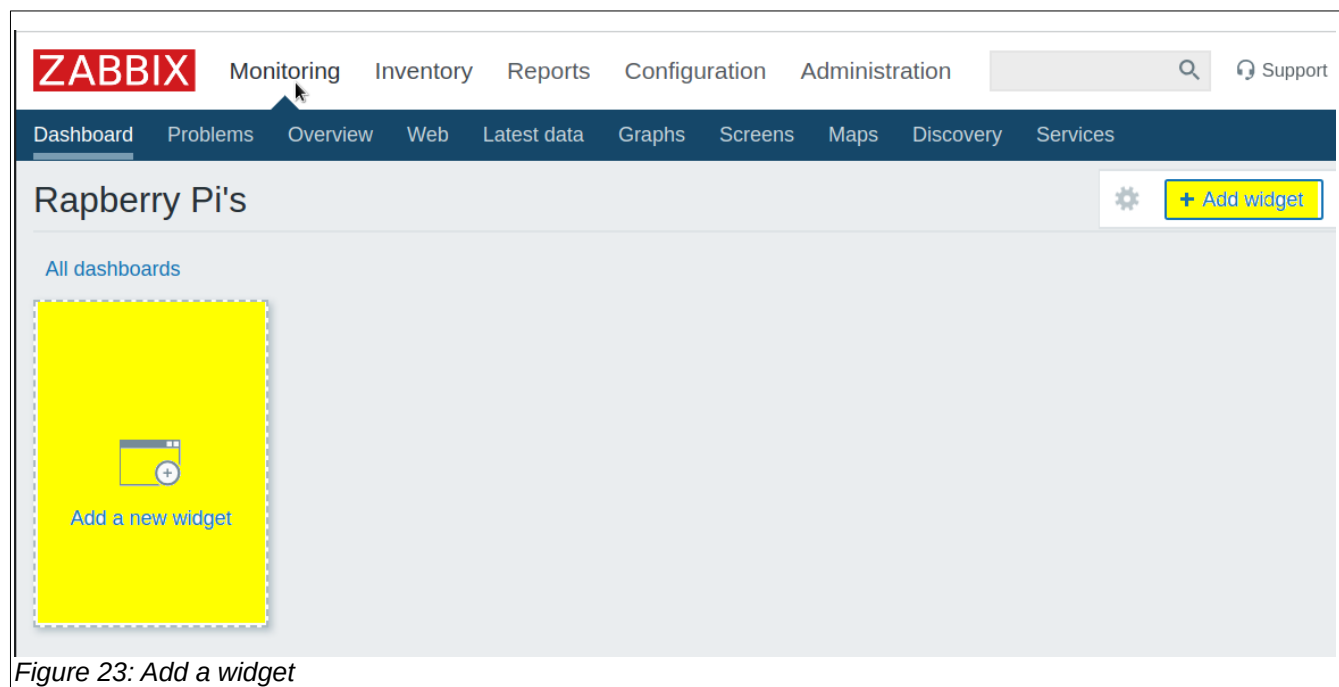


Figure 23: Add a widget

The first widget we will add to our dashboard is a local time clock. Click on '+ Add widget' and select the Type as Clock. Click the 'Add' button. Figure 24.

Add widget

Type: Clock

Name: Local Time

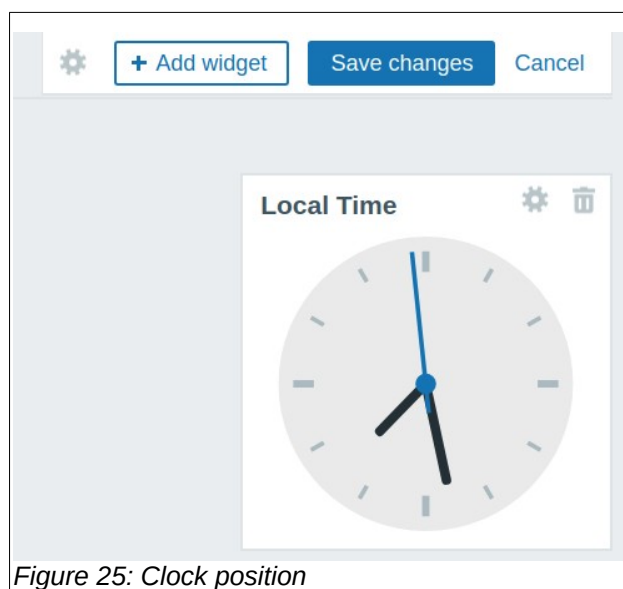
Refresh interval: Default (15 minutes)

Time type: Local time

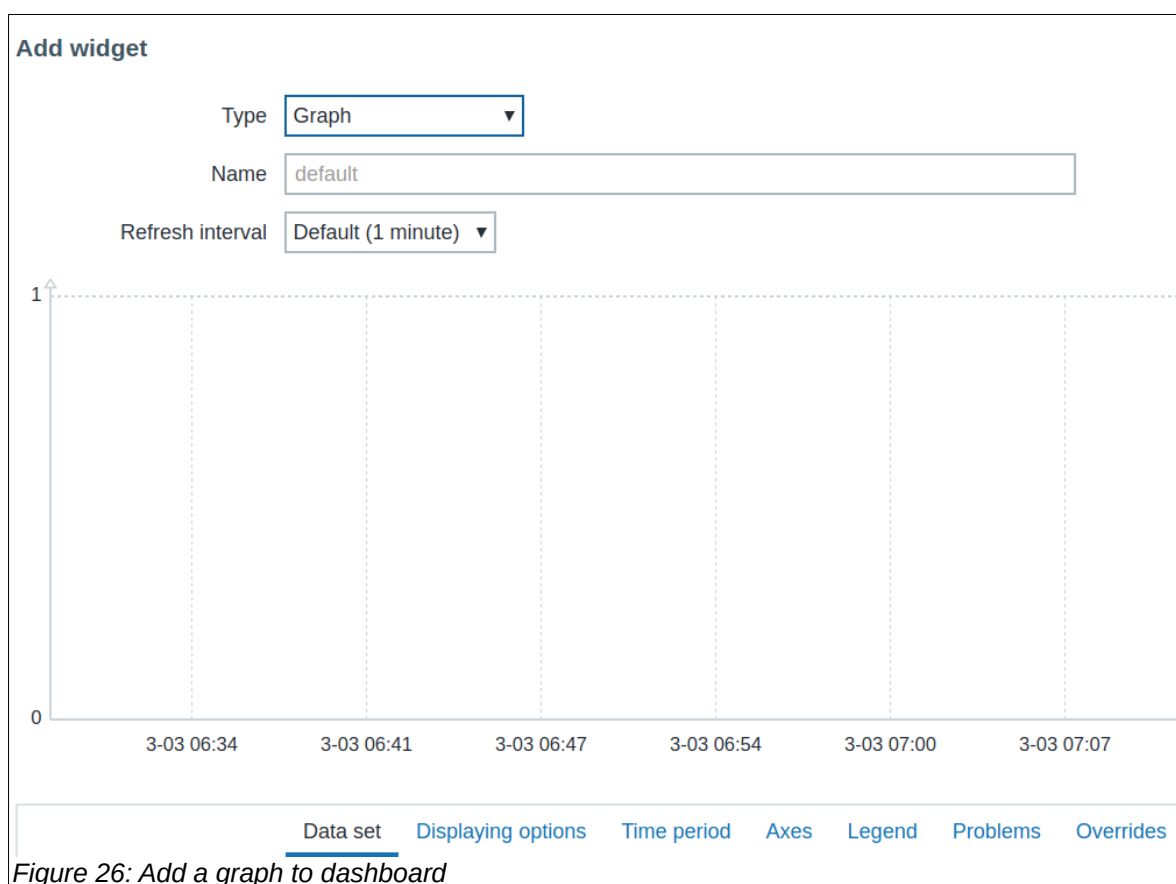
[Add] [Cancel]

Figure 24: Add click widget

Once created, drag the clock over to the top right corner of the panel as shown in Figure 25.



Next, add another widget. This time select the widget type as a Graph as shown in Figure 26.



I selected 'Available memory' from the list as shown in Figure 27.

Items					
		Group Raspberry Pi's ▾		Host piB ▾	
<input type="checkbox"/>	Name	Key	Type	Type of information	Status
<input type="checkbox"/>	Agent ping	agent.ping	Zabbix agent	Numeric (unsigned)	Enabled
<input checked="" type="checkbox"/>	Available memory	vm.memory.size[available]	Zabbix agent	Numeric (unsigned)	Enabled
<input type="checkbox"/>	Checksum of /etc/passwd	vfs.file.cksum[/etc/passwd]	Zabbix agent	Numeric (unsigned)	Enabled
<input type="checkbox"/>	Context switches per second	system.cpu.switches	Zabbix agent	Numeric (unsigned)	Enabled

Figure 27: Available memory widget

You must configure the 'Data set' for the widget. In the 'Data set' section of the panel, select the Hostname on the left, and the graph type on the right. This is highlighted below in Figure 28.

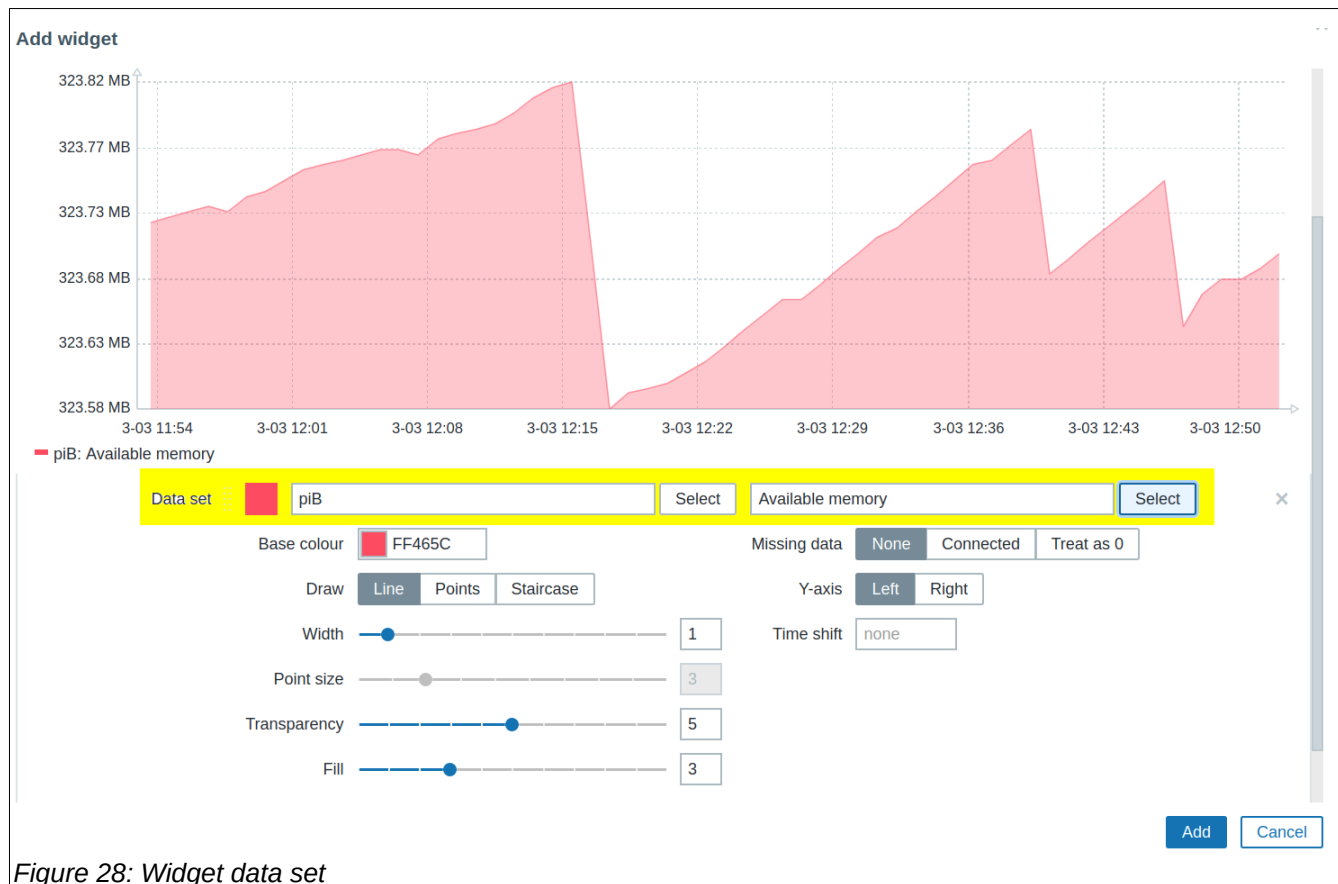


Figure 28: Widget data set

Once the data set is configured, click on the 'Add' button at the bottom right of the panel. This will place the new widget on the dashboard. There are numerous controls that allow you to customize the appearance of a widget. This includes the color, trend line width, transparency, fill characteristics, etc.

For example, If you prefer to display a line instead of a 'mountain' for the trend, simply move the Fill control all the way to the left. If you do this, you may want to increase the line width to 2 or 3.

I added several more widgets and arranged them on the dashboard (Figure 29). Be sure to experiment with the different display options. It is quite amazing how much flexibility the Zabbix platform provides.

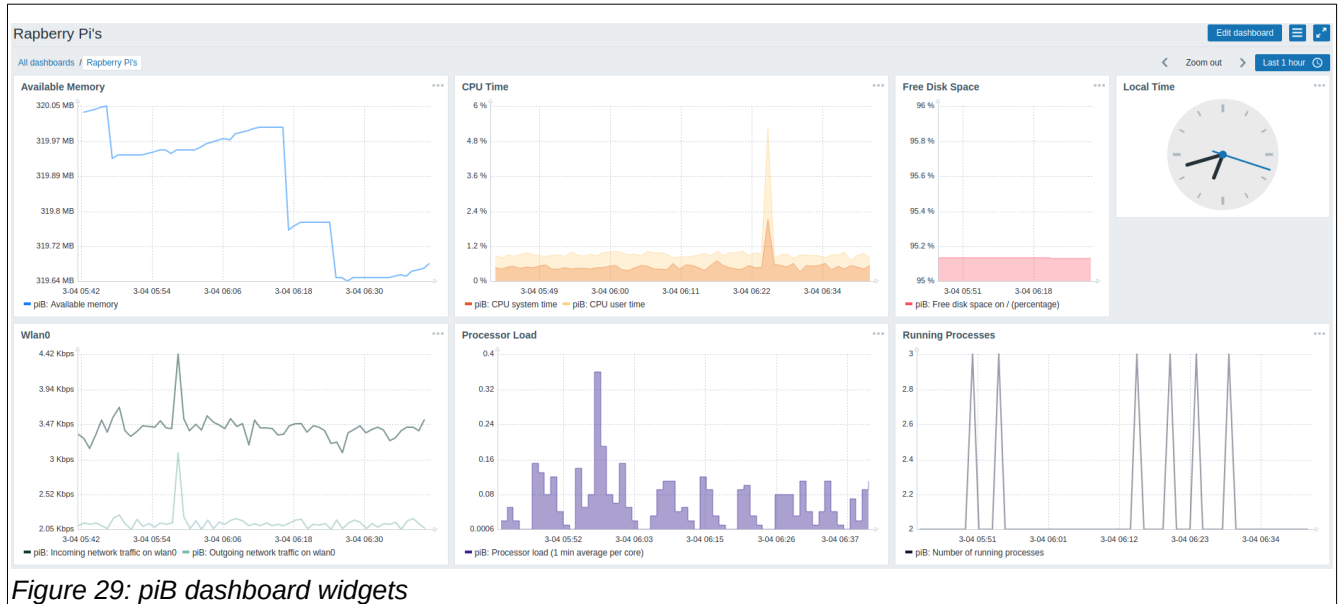


Figure 29: piB dashboard widgets

For those of you that live on the darker side, it is easy to change the default theme from light to dark. To change the theme, click on 'Administration | General' and change the 'Default theme' from Blue to Dark. Then click Update (Figure 30).

Figure 30: Default Theme

If you do change from a light to dark theme you will probably need to use lighter colors in your charts to make them look nicer (Figure 31).

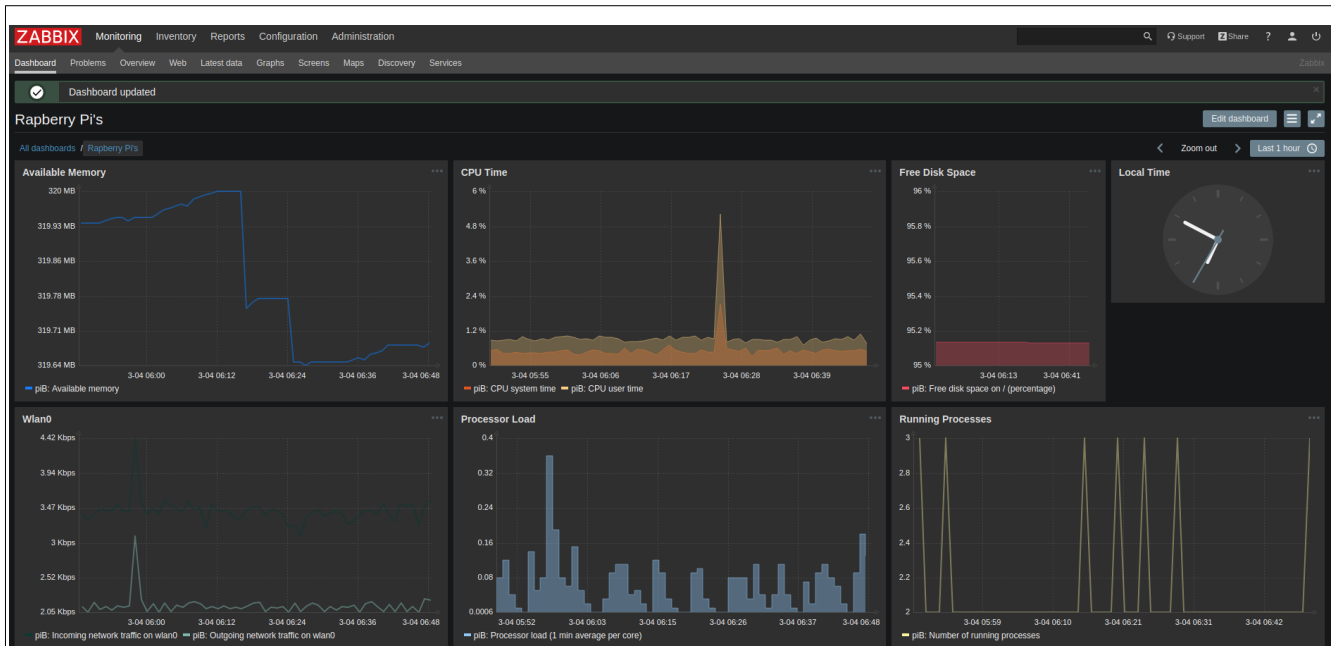


Figure 31: Dark Theme

Congratulations! You now can monitor your entire fleet of Raspberry Pi's using Zabbix.

Summary

The Zabbix platform is a terrific open-source project that can be very useful to the Raspberry Pi community. In this 'How-To' guide you learned how to monitor a Raspberry Pi on a custom dashboard. I encourage you to read the excellent Zabbix documentation to learn all of the cool things you can do with it.

In an upcoming 'How-To' document, I will show how to create custom parameters for a Zabbix agent. This feature allows you to run custom programs or scripts and display the output on a dashboard. With this capability, you can monitor any sensor that you have connected to a Pi. Stay tuned.

Send corrections, comments, complaints, ideas, or any other feedback to: sopwith@ismellsmoke.net.