



Monitoring Your FileMaker Server

Zabbix: Full Installation from Scratch

By Wim Decorte, Senior Technical Solution Architect
and Mislav Kos, Senior Technical Project Lead
Soliant Consulting, Inc.

July 29, 2019

Table of Contents

Choose Your Linux	3
Choose Your Location	4
Setting Up CentOS On AWS	6
SSH To Server	8
Install nano.....	8
SELinux Configuration Change	9
Install Zabbix.....	10
Install MySQL.....	11
Create the Zabbix Database	13
Configure Zabbix Server	15
Configure Zabbix Frontend	19
Next Steps.....	24
Disable MySQL Binary Logging	24

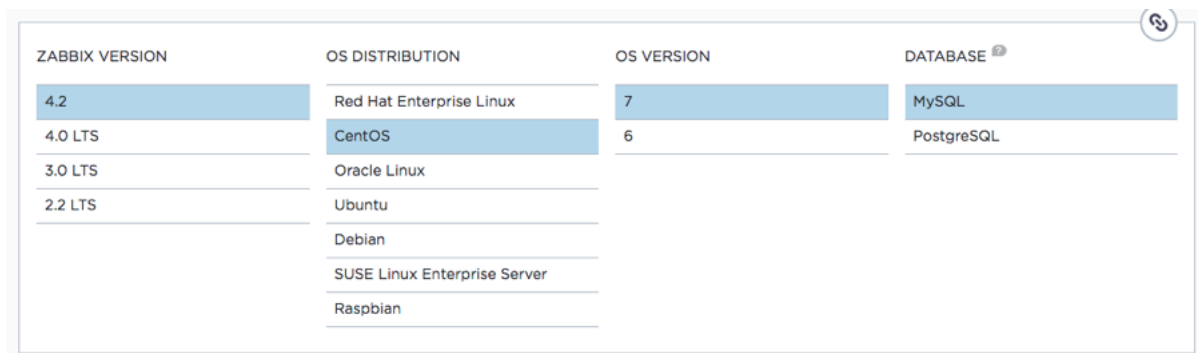
This document is one in a series of guides that walk you through installing, configuring, and using Zabbix to monitor your FileMaker servers. The full set of guides is available at <https://www.soliantconsulting.com/filemaker-zabbix>.

This guide will walk you through installing Zabbix Server. These steps assume you are somewhat proficient with Linux and the use of the command line or that you are willing to expand your skill set in this area. If this does not describe you, see the guide (2a – Zabbix as an Appliance) about using the Zabbix Server Appliance.

Choose Your Linux

Zabbix Server only runs on Linux, so we will have to pick a flavor of Linux we are comfortable with that is supported by Zabbix.

The [Zabbix download page](#) guides you neatly through the available choices of operating systems, versions of the chosen operating system, and backend-databases you want Zabbix Server to use to store its data:



ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	DATABASE [?]
4.2	Red Hat Enterprise Linux	7	MySQL
4.0 LTS	CentOS	6	PostgreSQL
3.0 LTS	Oracle Linux		
2.2 LTS	Ubuntu		
	Debian		
	SUSE Linux Enterprise Server		
	Raspbian		

Figure 1. Zabbix download

We chose CentOS, because it is also the operating system used for FileMaker Cloud, so whatever skills we pick up working with CentOS will serve us well both in working with the underpinnings of Zabbix Server and FileMaker Cloud. CentOS is also the operating system used in FileMaker Inc's [installation guide](#).

Note that the white paper that FileMaker Inc. published with the release of FileMaker 18 back in May of 2019 uses Zabbix Server version 3.x. The current version of as July

2019 is Zabbix 4.2 and its installation instructions are just slightly different. But the FMI guide is still a good reference¹.

Selecting the OS, version, and database on the Zabbix download page, update the command line information further down on the page that you need for the installation of Zabbix Server and its components:



The screenshot shows a webpage titled "Install and configure Zabbix server for your platform". It contains two sections: "a. Install Zabbix repository" and "b. Install Zabbix server, frontend, agent". Section a includes a terminal command to add the Zabbix repository and clean the yum cache. Section b includes a terminal command to install the Zabbix server, web frontend, and agent. A "documentation" link is visible in the top right of section a.

```
Install and configure Zabbix server for your platform
```

a. Install Zabbix repository [documentation](#)

```
# rpm -Uvh https://repo.zabbix.com/zabbix/4.2/rhel/7/x86_64/zabbix-release-4.2-1.el7.noarch.rpm
# yum clean all
```

b. Install Zabbix server, frontend, agent

```
# yum -y install zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

Figure 2. Updated command line information

Those steps assume that you already have a running Linux server of your chosen version, so that you can copy and paste these commands into the command line terminal. Let's take a step back and get one up and running.

There are a couple of different ways to get a running CentOS depending on whether you want it on-premise on physical hardware, on-premise as a virtual machine, or in the cloud.

Choose Your Location

For an on-premise install on physical hardware, you can grab the ISO installer directly from <https://centos.org/>:

¹ For complete reference, the official installation guide is here:

https://www.zabbix.com/documentation/4.2/manual/installation/install_from_packages/rhel_centos

and the one that FileMaker Inc has put together:

<https://community.filemaker.com/en/s/article/Using-Zabbix-for-Monitoring-FileMaker-Server>

Between those two and this guide you are reading now, you should have all the information you need to get to a successful Zabbix Server installation.



Figure 3. Download CentOS

You can also use that ISO image to install it on-premise as a virtual machine. Or, if you want to cut out a few steps, you can download a pre-built virtual machine image from <https://www.osboxes.org/centos>.



Figure 4. Pre-built virtual image from CentOS

Setting Up CentOS On AWS

If you prefer a cloud server, pre-built images are available as well in the cloud provider's market place. For AWS, centos.org provides an official instance for CentOS 7:

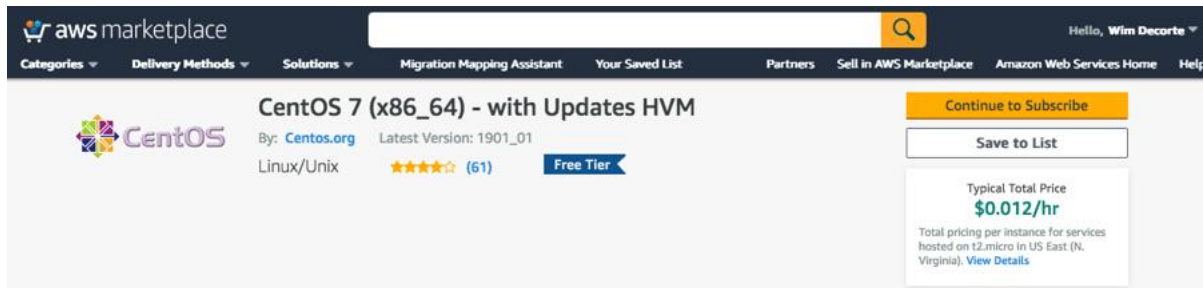
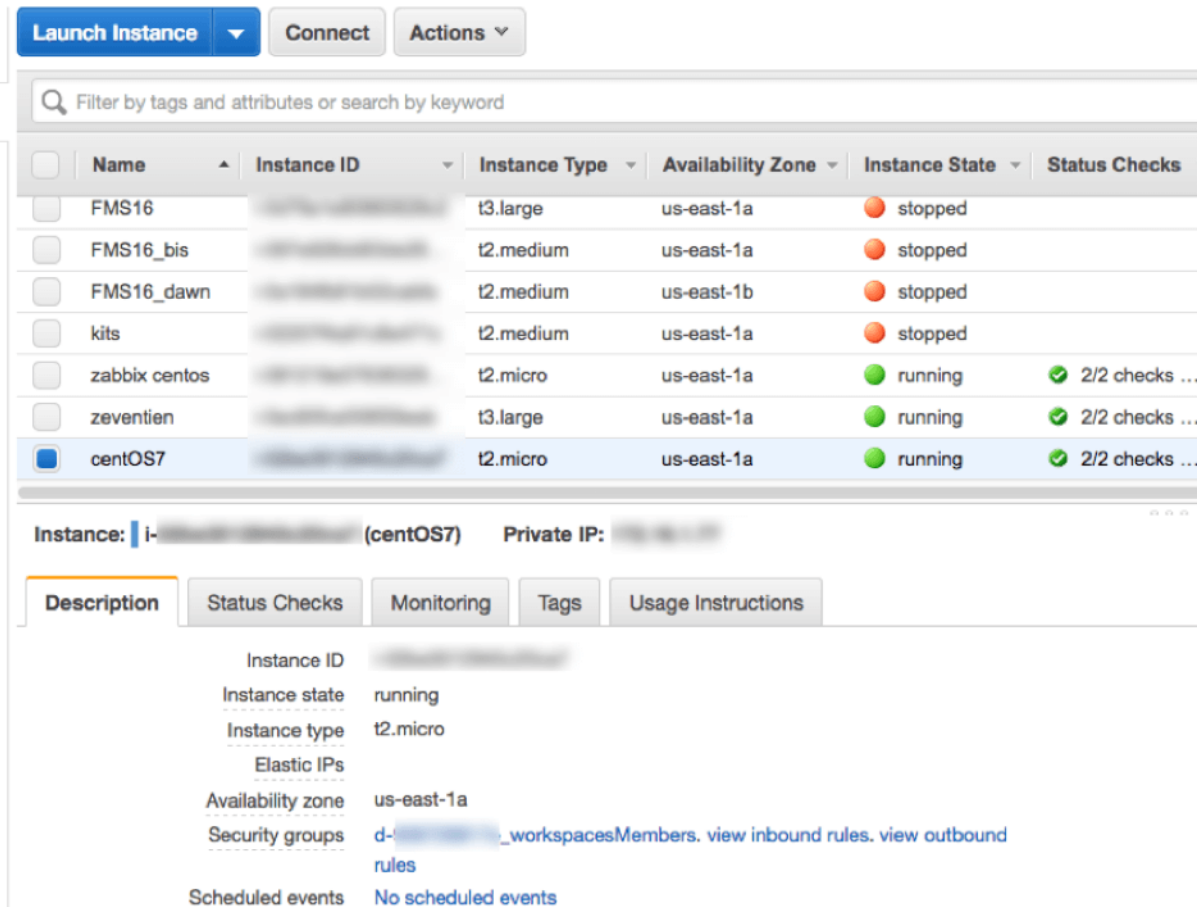


Figure 5. Pre-built image for AWS

The default instance type is a t2.micro (one virtual CPU and 1GB of RAM with 8GB of disk space), which is very cheap and more than capable of handling a Zabbix server monitoring multiple FileMaker Server hosts.

For this guide we will use AWS, since it is easy to spin up a new instance (and abandon it) without having to make changes to our in-house infrastructure.

Once you go through the few steps required in AWS to have the machine launched, you will see it in your EC2 console. The “Name” column will be empty initially. I named it centOS7:



Launch Instance ▾ Connect Actions ▾

Filter by tags and attributes or search by keyword

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
<input type="checkbox"/>	FMS16	...	t3.large	us-east-1a	stopped	
<input type="checkbox"/>	FMS16_bis	...	t2.medium	us-east-1a	stopped	
<input type="checkbox"/>	FMS16_dawn	...	t2.medium	us-east-1b	stopped	
<input type="checkbox"/>	kits	...	t2.medium	us-east-1a	stopped	
<input type="checkbox"/>	zabbix centos	...	t2.micro	us-east-1a	running	2/2 checks ...
<input type="checkbox"/>	zeventien	...	t3.large	us-east-1a	running	2/2 checks ...
<input checked="" type="checkbox"/>	centOS7	...	t2.micro	us-east-1a	running	2/2 checks ...

Instance: j-... (centOS7) Private IP: ...

Description | Status Checks | Monitoring | Tags | Usage Instructions

Instance ID: ...

Instance state: running

Instance type: t2.micro

Elastic IPs: ...

Availability zone: us-east-1a

Security groups: d-..._workspacesMembers. [view inbound rules](#). [view outbound rules](#)

Scheduled events: [No scheduled events](#)

Figure 6. EC2 console

The first thing we will do is click on the security group at the bottom of that screenshot to view and update the AWS ‘firewall’ so that we can use SSH to remote into the instance.

Specifically, we want to work with the “inbound rules.” By default, there will be none, so we will add the ones that we need:

- Port 22 for SSH
- Port 10051 for incoming data from the Zabbix agents that we will deploy later
- Ports 80 and 443 for access to the browser-based Zabbix admin console

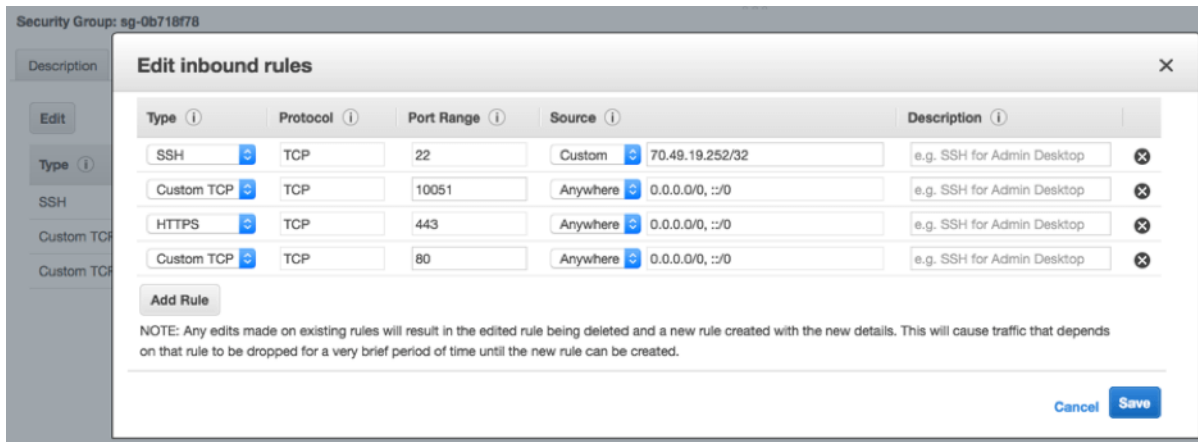


Figure 7. Set up the inbound rules

SSH To Server

At this point, we have a working Linux server. To connect to it, open your favorite SSH client (on macOS I am just using Terminal) and issue the proper SSH command:

```
ssh -i /Users/wimdecorte/Documents/projects/ETS18/zabbix_resources/wim_ets_15.pem centos@xxx.xxx.xxx.xxx
```

The path to the .pem file is required by AWS to allow SSH connections. “centos” is the default username for CentOS, and what comes after the @ is the public IP address or DNS name of your Linux server.

The first time you log in you will be asked for confirmation to connect and then will see the command prompt of an SSH session on your Linux server.

```
Wims-MBP:~ wimdecorte$ ssh -i /Users/wimdecorte/Documents/projects/ETS18/zabbix_resources/wim_ets_15.pem centos@
The authenticity of host 'xxx.xxx.xxx.xxx' can't be established.
ECDSA key fingerprint is SHA256:n0b+Has1bRRgXph0eGua13W1bQigd/BtwhvwHx6rxDs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'xxx.xxx.xxx.xxx' (ECDSA) to the list of known hosts.
[centos@ip-xxx.xxx.xxx.xxx ~]$
```

Figure 8. Command prompt of an SSH session

Install nano

The very first thing we will do is install “nano,” our favorite Linux text editor.

Type in:

```
sudo yum install nano
```


(yum stands for YellowDog Updater, Modified and is the CentOS default software manager. It is used to install new software and system updates).

As with all installations and updates, you will see a bit of an overview of what will happen, and you will be asked to confirm with “Y” that you want to proceed:

```

[centos@ip-172-31-36-233 ~]$ sudo yum install nano
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: mirrors.advancedhosters.com
 * extras: mirrors.advancedhosters.com
 * updates: mirrors.advancedhosters.com
base                                                    | 3.6 kB  00:00:00
extras                                                  | 3.4 kB  00:00:00
updates                                                 | 3.4 kB  00:00:00
(1/4): base/7/x86_64/group_gz                          | 166 kB  00:00:00
(2/4): extras/7/x86_64/primary_db                     | 205 kB  00:00:00
(3/4): updates/7/x86_64/primary_db                   | 6.5 MB  00:00:00
(4/4): base/7/x86_64/primary_db                      | 6.0 MB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package nano.x86_64 0:2.3.1-10.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch             Version          Repository      Size
=====
Installing:
nano                   x86_64           2.3.1-10.el7    base            440 k
=====
Transaction Summary
=====
Install 1 Package

Total download size: 440 k
Installed size: 1.6 M
Is this ok [y/d/N]:

```

Figure 9. Type “Y” to proceed

A few seconds later, we will be finished:

```

Installed:
 nano.x86_64 0:2.3.1-10.el7

Complete!
[centos@ip-172-31-36-233 ~]$

```

Figure 10. Nano install is complete

SELinux Configuration Change

Second on our to-do list is to change a security setting in CentOS’s default configuration. By default, CentOS has [SELinux](#) enabled, which will get in the way of Zabbix Server functioning properly, so we need to adjust it:

Type in:

```
sudo nano /etc/selinux/config
```

Change the line that starts with “SELINUX” to read “SELINUX=permissive”

```
GNU nano 2.3.1 File: /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Figure 11. Line changed to “SELINUX=permissive”

Press control-o and then enter to save the changes and control-x to quit the text editor.

Install Zabbix

Now we can go back to the instructions on the [Zabbix download page](#) that tell us how to install Zabbix Server:

```
Install and configure Zabbix server for your platform

a. Install Zabbix repository documentation

# rpm -Uvh https://repo.zabbix.com/zabbix/4.2/rhel/7/x86_64/zabbix-release-4.2-1.el7.noarch.rpm
# yum clean all

b. Install Zabbix server, frontend, agent

# yum -y install zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

Figure 12. Install Zabbix Server

The first set of commands is basically telling ‘yum’ where the installers are located and to clean its internal database of available software locations.

Remember to run all of these commands as ‘super user’ by prefixing them with ‘sudo’. The first command will look like this:

```
[centos@ip-10.10.10.10 ~]$ sudo rpm -Uvh https://repo.zabbix.com/zabbix/4.2/rhel/7/x86_64/zabbix-release-4.2-1.el7.noarch.rpm
```

Figure 13. Prefix all commands with “sudo”

The result of running the first two commands will look like this:

```
[centos@ip-10.10.10.10 ~]$ sudo rpm -Uvh https://repo.zabbix.com/zabbix/4.2/rhel/7/x86_64/zabbix-release-4.2-1.el7.noarch.rpm
Retrieving https://repo.zabbix.com/zabbix/4.2/rhel/7/x86_64/zabbix-release-4.2-1.el7.noarch.rpm
warning: /var/tmp/rpm-tmp.TFap95: Header V4 RSA/SHA512 Signature, key ID a14fe591: NOKEY
Preparing... ##### [100%]
Updating / installing...
 1:zabbix-release-4.2-1.el7 ##### [100%]
[centos@ip-10.10.10.10 ~]$ sudo yum clean all
Loaded plugins: fastestmirror
Cleaning repos: base extras updates zabbix zabbix-non-supported
Cleaning up list of fastest mirrors
[centos@ip-10.10.10.10 ~]$
```

Figure 14. After running commands under “a. Install Zabbix repository”

The third command on the Zabbix downloads page (under b) is where the actual installation happens:

```
sudo yum -y install zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

That one will run for a little while, install everything needed, and report back what it has done:

```
Installed:
  zabbix-agent.x86_64 0:4.2.4-1.el7  zabbix-server-mysql.x86_64 0:4.2.4-1.el7  zabbix-web-mysql.noarch 0:4.2.4-1.el7

Dependency Installed:
  OpenIPMI-libs.x86_64 0:2.0.23-2.el7          OpenIPMI-modalias.x86_64 0:2.0.23-2.el7
  apr.x86_64 0:1.4.8-3.el7_4.1                apr-util.x86_64 0:1.5.2-6.el7
  centos-logos.noarch 0:70.0.6-3.el7.centos    dejavu-fonts-common.noarch 0:2.33-6.el7
  dejavu-sans-fonts.noarch 0:2.33-6.el7        fontpackages-filesystem.noarch 0:1.44-8.el7
  fping.x86_64 0:3.10-1.el7                  httpd.x86_64 0:2.4.6-89.el7.centos
  httpd-tools.x86_64 0:2.4.6-89.el7.centos    libX11.x86_64 0:1.6.5-2.el7
  libX11-common.noarch 0:1.6.5-2.el7          libXau.x86_64 0:1.0.8-2.1.el7
  libXpm.x86_64 0:3.5.12-1.el7                libjpeg-turbo.x86_64 0:1.2.90-6.el7
  libtool-ltdl.x86_64 0:2.4.2-22.el7_3       libxcb.x86_64 0:1.13-1.el7
  libxslt.x86_64 0:1.1.28-5.el7              libzip.x86_64 0:0.10.1-8.el7
  mailcap.noarch 0:2.1.41-2.el7              net-snmp-libs.x86_64 1:5.7.2-37.el7
  php.x86_64 0:5.4.16-46.el7                 php-bcmath.x86_64 0:5.4.16-46.el7
  php-cli.x86_64 0:5.4.16-46.el7            php-common.x86_64 0:5.4.16-46.el7
  php-gd.x86_64 0:5.4.16-46.el7             php-ldap.x86_64 0:5.4.16-46.el7
  php-mbstring.x86_64 0:5.4.16-46.el7       php-mysql.x86_64 0:5.4.16-46.el7
  php-pdo.x86_64 0:5.4.16-46.el7            php-xml.x86_64 0:5.4.16-46.el7
  t1lib.x86_64 0:5.1.2-14.el7               unixODBC.x86_64 0:2.3.1-11.el7
  zabbix-web.noarch 0:4.2.4-1.el7

Complete!
[centos@ip-10.10.10.10 ~]$
```

Figure 15. Showing what has been done

Install MySQL

The next step is to configure the underlying MySQL (c and d on the Zabbix downloads page). But this is where the instructions may fail; MySQL is likely not installed at this point yet. You can try the command, but if an error comes back, follow the steps below to install MySQL:

Add the MySQL 8.0 repository to your server:

```
sudo yum install https://dev.mysql.com/get/mysql80-community-release-el7-2.noarch.rpm
```

Then install MySQL itself:

```
sudo yum -y install mysql-community-server
```

This one is a fairly hefty download and will take a while. After it is installed, start the MySQL service:

```
sudo systemctl start mysqld
```

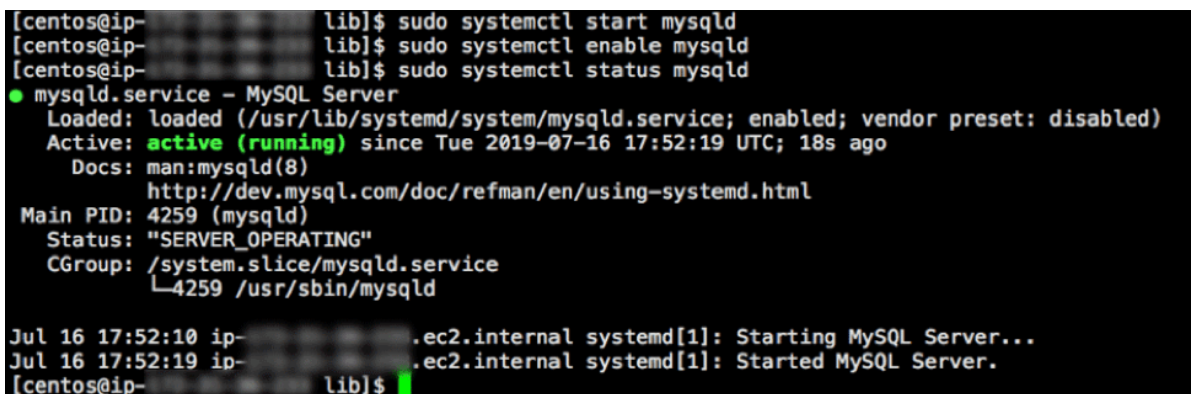
Enable it to auto-start when the machine starts:

```
sudo systemctl enable mysqld
```

At any time, you can check if MySQL is running by using this command:

```
sudo systemctl status mysqld
```

When all is well, you will see an output from that 'status' command, confirming MySQL is up and running

A terminal window showing the execution of three systemctl commands: 'start', 'enable', and 'status' for the 'mysqld' service. The 'status' command output shows the service is 'active (running)' since Tue 2019-07-16 17:52:19 UTC, with a main PID of 4259. Log messages at the bottom confirm the service started successfully.

```
[centos@ip-... lib]$ sudo systemctl start mysqld
[centos@ip-... lib]$ sudo systemctl enable mysqld
[centos@ip-... lib]$ sudo systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-07-16 17:52:19 UTC; 18s ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
   Main PID: 4259 (mysqld)
   Status: "SERVER_OPERATING"
   CGroup: /system.slice/mysqld.service
           └─4259 /usr/sbin/mysqld

Jul 16 17:52:10 ip-... ec2.internal systemd[1]: Starting MySQL Server...
Jul 16 17:52:19 ip-... ec2.internal systemd[1]: Started MySQL Server.
[centos@ip-... lib]$
```

Figure 16. MySQL up and running

MySQL installs with a temporary password that we need to grab before we do anything else. To do this, type in:

```
sudo nano /var/log/mysqld.log
```

and make note that the master user is 'root' with the password listed there.

```

GNU nano 2.3.1                               File: /var/log/mysql.log
2019-07-16T17:52:12.581281Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.16) initializing of server in progress as process 425
2019-07-16T17:52:15.444675Z 5 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: qgn4YUamQh*s
2019-07-16T17:52:16.666931Z 0 [System] [MY-013170] [Server] /usr/sbin/mysqld (mysqld 8.0.16) initializing of server has completed
2019-07-16T17:52:18.585446Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.16) starting as process 4259
2019-07-16T17:52:19.056126Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2019-07-16T17:52:19.083955Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.16' socket: '/var/lib/mysq
2019-07-16T17:52:19.156679Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Socket: '/var/run/mysqld/mysqlx.sock' bind-address:

```

Figure 17. Make note of the temporary password generated for master user

Hit control-x to quit the text editor.

Create the Zabbix Database

Now we are ready to resume our Zabbix installation following the step c instructions on the Zabbix downloads page to create the required database:

The first command is to enter 'mysql mode':

```
sudo mysql -uroot -p
```

At the password prompt, use the password that you retrieved earlier. After doing so, you will find yourself at a MySQL prompt.

```

[centos@ip-10.10.10.10 ~]$ sudo mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.16

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █

```

Figure 18. Use the temporary password

Before we execute the commands listed on the Zabbix page, we need to change the password for the root user, since that temporary password cannot be used beyond this point. (The commands listed in step c will fail if you try.)

Type in:

```
alter user root@localhost identified by 'someNewPassword';
```

The new password is in-between the single quotes. Don't forget to add the ";" at the end. This is how MySQL knows you are done with the command.

```
mysql> alter user root@localhost identified by 'MyN3wP@ssword';
Query OK, 0 rows affected (0.00 sec)

mysql> █
```

Figure 19. Temporary password is replaced

Now type in (or copy/paste) the first of the three separate MySQL commands of step c to create the database named "zabbix.."

```
c. Create initial database documentation

# mysql -uroot -p
password
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> grant all privileges on zabbix.* to zabbix@localhost identified by 'password';
mysql> quit;
```

Figure 20. Create the database named "zabbix"

Before we do the next command, we actually have to create the Zabbix user in MySQL. Older versions of MySQL allow you to use the GRANT command to both create the user and assign rights to that user, but that is no longer supported in newer version of MySQL (versions 8+).

Type in the following to create a MySQL "zabbix" user with a password you specify:

```
CREATE USER zabbix@localhost IDENTIFIED WITH mysql_native_password BY
'MyOtherN3wP@ssword';
```

Then instead of the command shown in the Zabbix instructions, use this slightly modified one to grant rights to the "zabbix" database for the "zabbix" user:

```
GRANT ALL PRIVILEGES ON zabbix.* TO zabbix@localhost WITH GRANT OPTION;
```

```
mysql> GRANT ALL PRIVILEGES ON zabbix.* TO zabbix@localhost WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> █
```

Figure 21. Grant rights to the "zabbix" database for the "zabbix" user

The last command is easy and just drops out of the MySQL mode and back into Linux:


```
mysql> quit;  
Bye  
[centos@ip- lib]$ █
```

Figure 22. Quit MySQL and return back into Linux

The last item in step c is to import the schema for the "zabbix" database:

```
sudo zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p Zabbix
```

```
[centos@ip- lib]$ sudo zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix  
Enter password:  
[centos@ip- lib]$ █
```

Figure 23. Import schema for the "zabbix" database

Note that the password requested here is for the newly created "zabbix" user.

Configure Zabbix Server

The next step in the Zabbix instructions calls for an edit to the Zabbix config file to make sure that Zabbix knows the MySQL password for the "zabbix" user:

d. Configure the database for Zabbix server

Edit file /etc/zabbix/zabbix_server.conf

```
DBPassword=password
```

Figure 24. Edit the Zabbix config file

Type in:

```
sudo nano /etc/zabbix/zabbix_server.conf
```

to open the Zabbix configuration file and scroll down to the entry for the database password:

```
GNU nano 2.3.1 File: /etc/zabbix/zabbix_server.conf
# Default:
# DBSchema=

### Option: DBUser
# Database user.
#
# Mandatory: no
# Default:
# DBUser=

DBUser=zabbix

### Option: DBPassword
# Database password.
# Comment this line if no password is used.
#
# Mandatory: no
# Default:
# DBPassword=
### Option: DBSocket
# Path to MySQL socket.
#
```

Figure 25. Scroll to entry for the database password

Remove the “#” at the start of the line and add the password for the Zabbix MySQL user.

```
### Option: DBPassword
# Database password.
# Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=My0therN3wP@ssword
```

Figure 26. # symbol removed from start of the line

Hit control-o and then enter to save the change and then control-x to quit nano.

At this point, much of the heavy lifting is done; the next step listed on the Zabbix download page is to update the time zone that will be used by Zabbix:

e. Configure PHP for Zabbix frontend

Edit file /etc/httpd/conf.d/zabbix.conf, uncomment and set the right timezone for you.

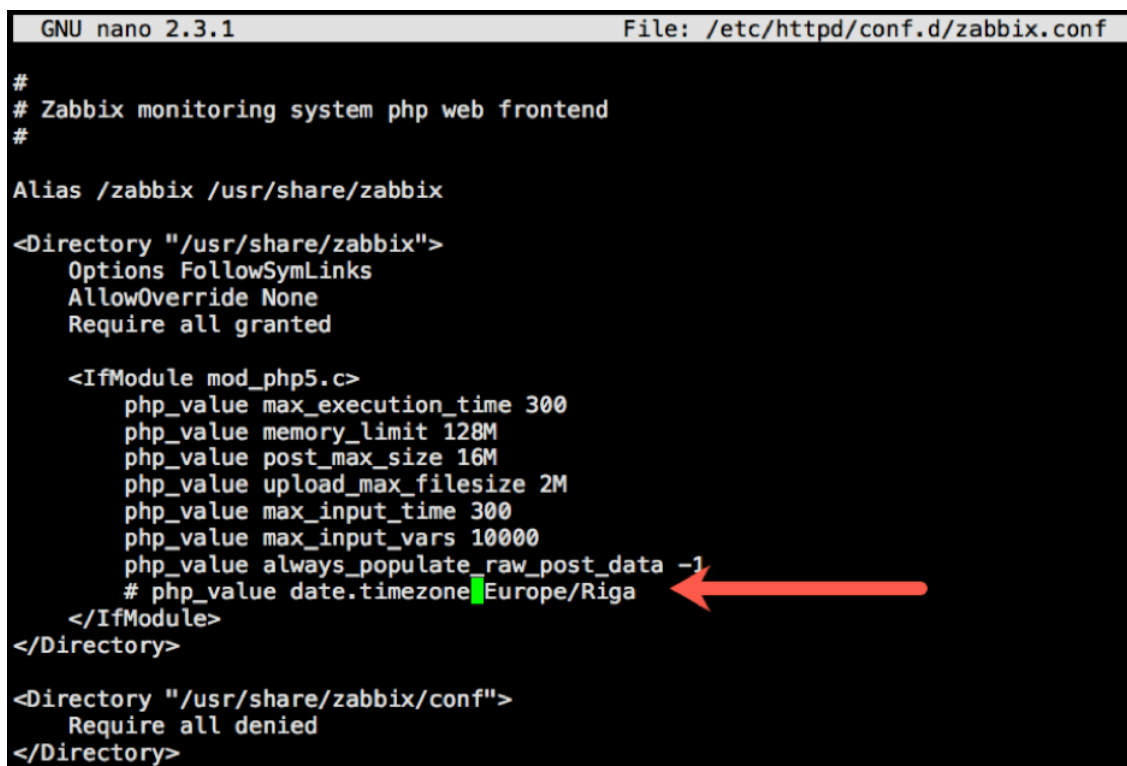
```
# php_value date.timezone Europe/Riga
```

Figure 27. Update the time zone

Type in

```
sudo nano /etc/httpd/conf.d/zabbix.conf
```

and scroll down to the time zone setting:



```
GNU nano 2.3.1 File: /etc/httpd/conf.d/zabbix.conf
#
# Zabbix monitoring system php web frontend
#
Alias /zabbix /usr/share/zabbix

<Directory "/usr/share/zabbix">
  Options FollowSymLinks
  AllowOverride None
  Require all granted

  <IfModule mod_php5.c>
    php_value max_execution_time 300
    php_value memory_limit 128M
    php_value post_max_size 16M
    php_value upload_max_filesize 2M
    php_value max_input_time 300
    php_value max_input_vars 10000
    php_value always_populate_raw_post_data -1
    # php_value date.timezone Europe/Riga
  </IfModule>
</Directory>

<Directory "/usr/share/zabbix/conf">
  Require all denied
</Directory>
```

Figure 28. Time zone setting

And change it to your time zone. All supported time zones are listed here:

<https://www.php.net/manual/en/timezones.php> Since we are on the East Coast, we changed it to America/New_York and removed the “#” at the start of the line:

```

GNU nano 2.3.1                               File: /etc/httpd/conf.d/zabbix.conf
#
# Zabbix monitoring system php web frontend
#
Alias /zabbix /usr/share/zabbix

<Directory "/usr/share/zabbix">
  Options FollowSymLinks
  AllowOverride None
  Require all granted

  <IfModule mod_php5.c>
    php_value max_execution_time 300
    php_value memory_limit 128M
    php_value post_max_size 16M
    php_value upload_max_filesize 2M
    php_value max_input_time 300
    php_value max_input_vars 10000
    php_value always_populate_raw_post_data -1
    php_value date.timezone America/New_York
  </IfModule>
</Directory>

```

Figure 29. # symbol removed from start of the time zone line

Hit control-o and then enter to save the change and control-x to exit the text editor.

At this point we can start the Zabbix server so that all the changes we have made take effect.

f. Start Zabbix server and agent processes

Start Zabbix server and agent processes and make it start at system boot:

```

# systemctl restart zabbix-server zabbix-agent httpd
# systemctl enable zabbix-server zabbix-agent httpd

```

Figure 30. Start the Zabbix server

sudo systemctl restart zabbix-server zabbix-agent httpd

This command is actually restarting three services:

1. The Zabbix server
2. The Zabbix agent (each Zabbix server also monitors itself)
3. The web server (https)

The 2nd command under step F ensures that all three of these services will auto-start when the machine restarts:

sudo systemctl enable zabbix-server zabbix-agent httpd

At this stage you will want to confirm that all three of these services are properly working by running these three checks:

```
sudo systemctl status zabbix-server  
sudo systemctl status zabbix-agent  
sudo systemctl status httpd
```

After each command you should see the service “in the green” next to ‘Active’:

```
[centos@ip-172-31-36-233 ~]$ sudo systemctl status httpd  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)  
   Active: active (running) since Tue 2019-07-16 20:37:49 UTC; 6min ago  
     Docs: man:httpd(8)  
           man:apachectl(8)  
 Main PID: 3795 (httpd)  
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/sec"  
    CGroup: /system.slice/httpd.service  
            └─3795 /usr/sbin/httpd -DFOREGROUND  
              └─3857 /usr/sbin/httpd -DFOREGROUND  
                └─3858 /usr/sbin/httpd -DFOREGROUND  
                  └─3859 /usr/sbin/httpd -DFOREGROUND  
                    └─3861 /usr/sbin/httpd -DFOREGROUND  
                      └─3862 /usr/sbin/httpd -DFOREGROUND  
  
Jul 16 20:37:49 ip-172-31-36-233.ec2.internal systemd[1]: Starting The Apache HTTP Server...  
Jul 16 20:37:49 ip-172-31-36-233.ec2.internal systemd[1]: Started The Apache HTTP Server.  
[centos@ip-172-31-36-233 ~]$
```

Figure 31. Service shown in green

Configure Zabbix Frontend

Going back to following the instructions shown on the Zabbix download page, we can now turn our attention to the Zabbix frontend by opening it in the browser:

3 Configure Zabbix frontend

Connect to your newly installed Zabbix frontend: http://server_ip_or_name/zabbix
Follow steps described in Zabbix documentation: [Installing frontend](#)

Figure 32. Open Zabbix frontend

Use the URL as indicated in your favorite browser, and you should see this:

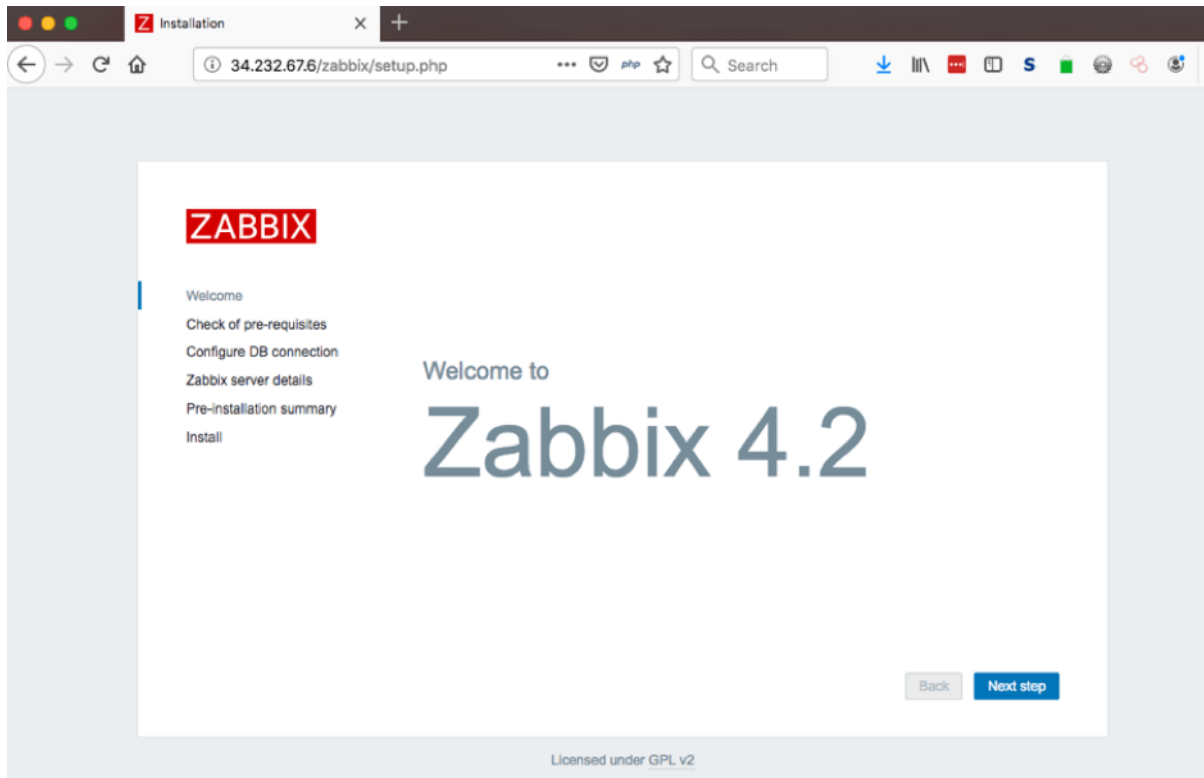


Figure 33. Zabbix frontend in browser

“Next Step” brings us to an overview of the PHP pre-requisites, and if we did the installation correctly, everything here should be in the green:

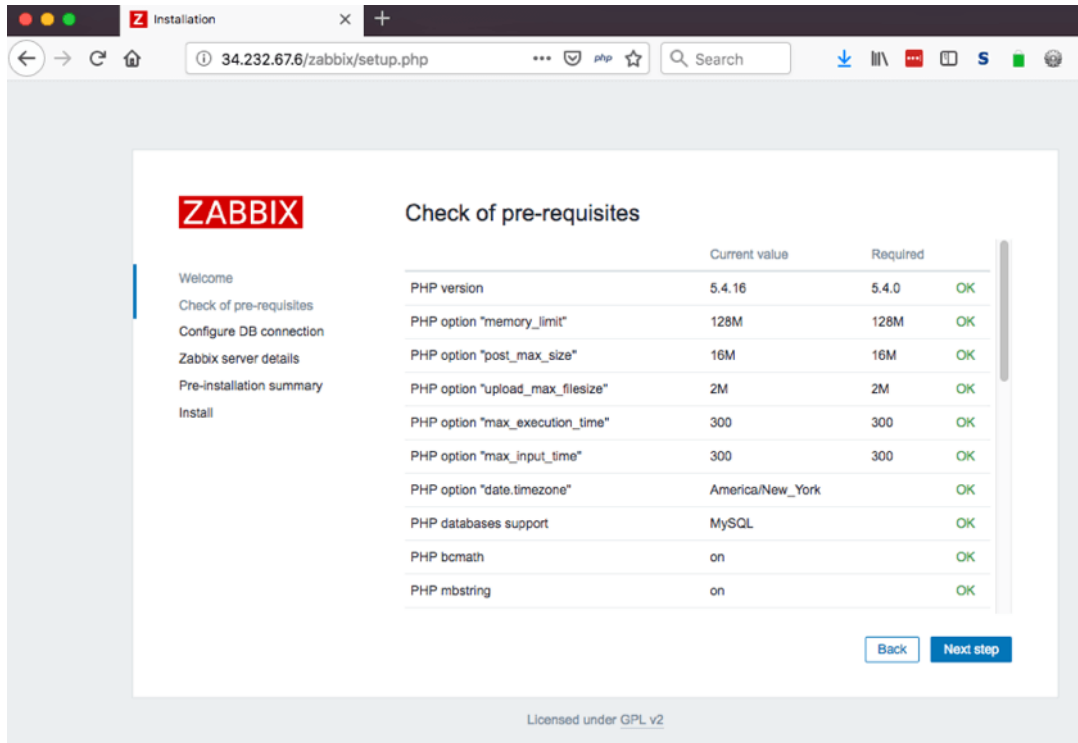


Figure 34. PHP prerequisites

The following step is a confirmation of the Zabbix MySQL database. Enter the password here for the "zabbix" user that was added earlier on.

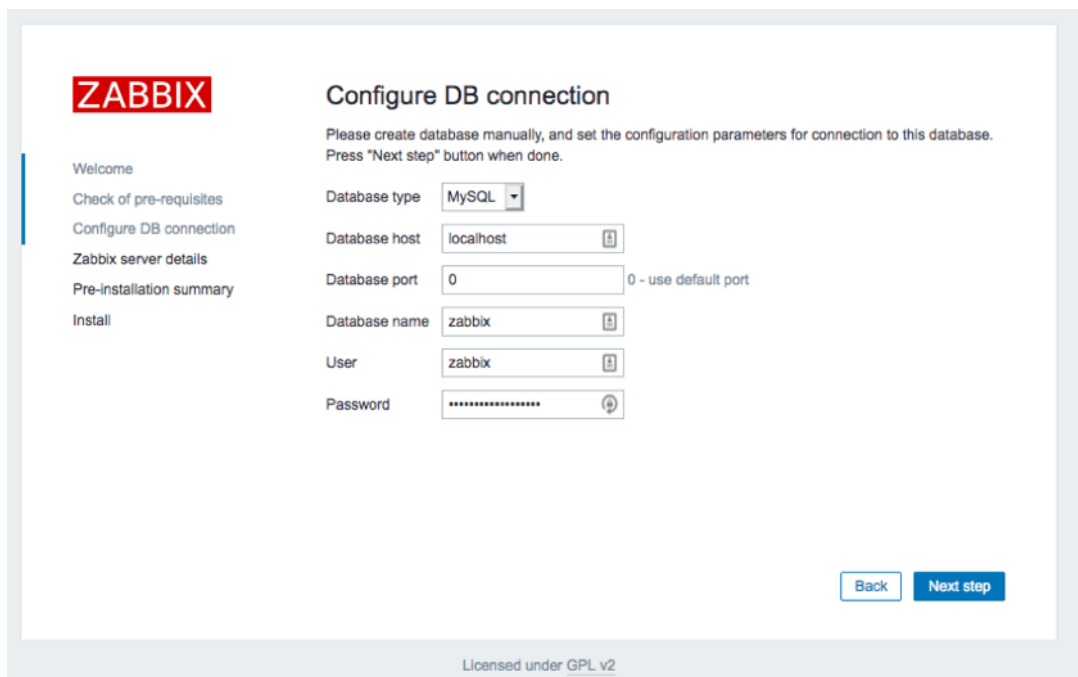
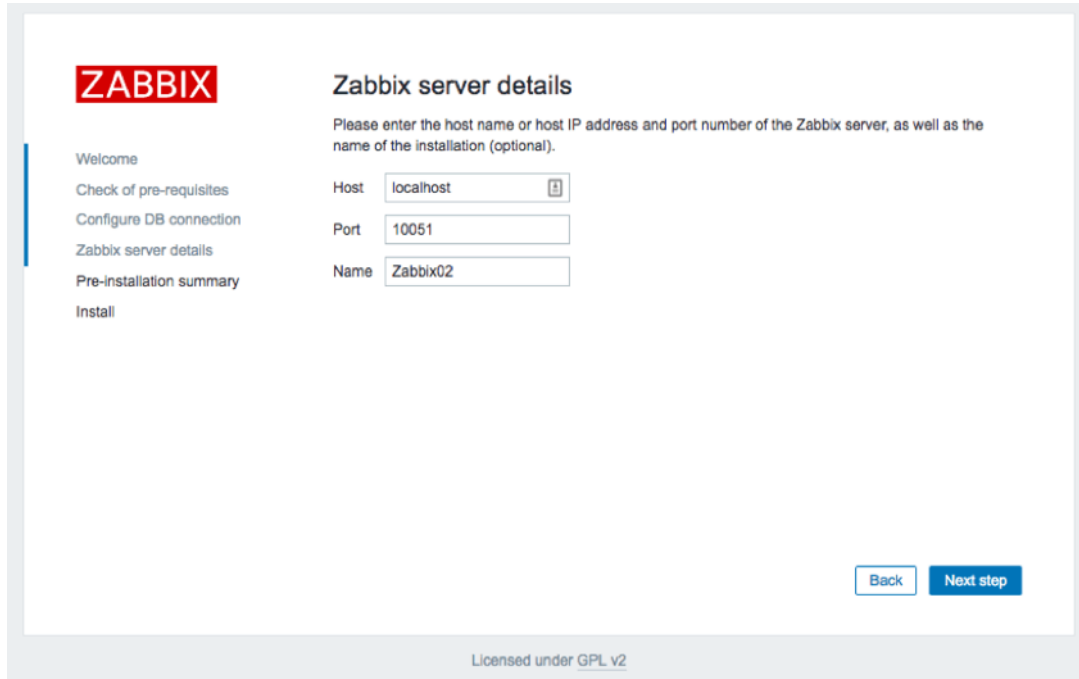


Figure 35. Enter password for the "zabbix" user

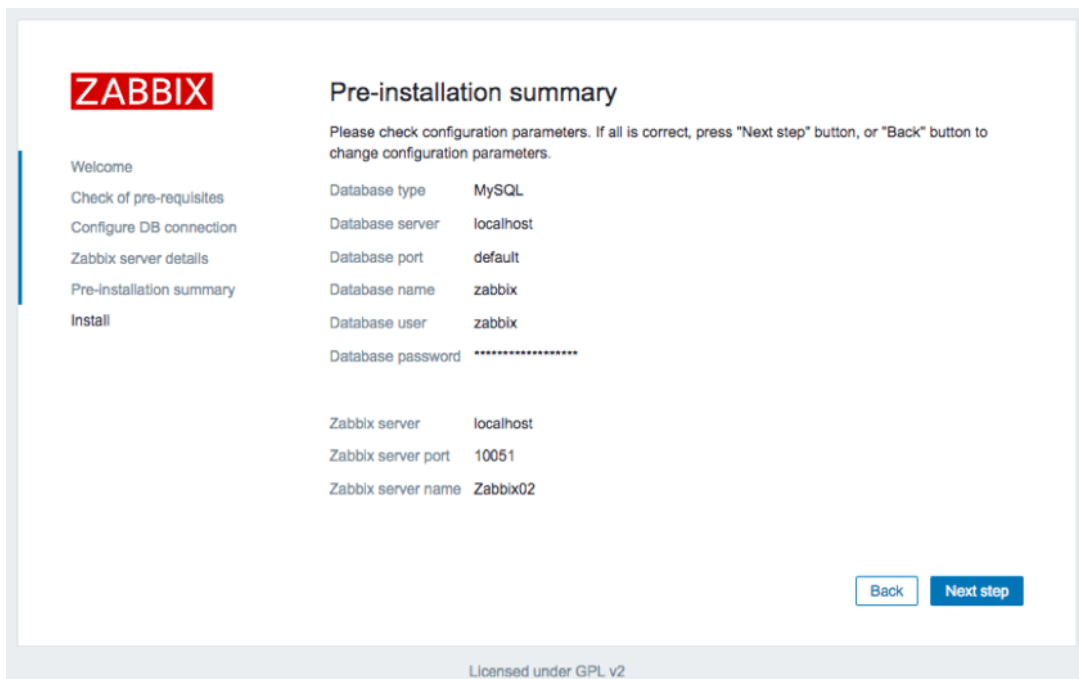
Provide a name for your Zabbix installation:



The screenshot shows the 'Zabbix server details' configuration screen. On the left is a navigation menu with the following items: Welcome, Check of pre-requisites, Configure DB connection, Zabbix server details (highlighted), Pre-installation summary, and Install. The main content area is titled 'Zabbix server details' and includes the instruction: 'Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional)'. Below this are three input fields: 'Host' with the value 'localhost', 'Port' with the value '10051', and 'Name' with the value 'Zabbix02'. At the bottom right are 'Back' and 'Next step' buttons. At the bottom center, it says 'Licensed under GPL v2'.

Figure 36. Enter name for the Zabbix installation

And you get one last chance to confirm all the settings:



The screenshot shows the 'Pre-installation summary' screen. The navigation menu on the left is: Welcome, Check of pre-requisites, Configure DB connection, Zabbix server details, Pre-installation summary (highlighted), and Install. The main content area is titled 'Pre-installation summary' and includes the instruction: 'Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.' Below this is a list of configuration parameters: Database type (MySQL), Database server (localhost), Database port (default), Database name (zabbix), Database user (zabbix), Database password (masked with asterisks), Zabbix server (localhost), Zabbix server port (10051), and Zabbix server name (Zabbix02). At the bottom right are 'Back' and 'Next step' buttons. At the bottom center, it says 'Licensed under GPL v2'.

Figure 37. Pre-installation summary

And we are done:

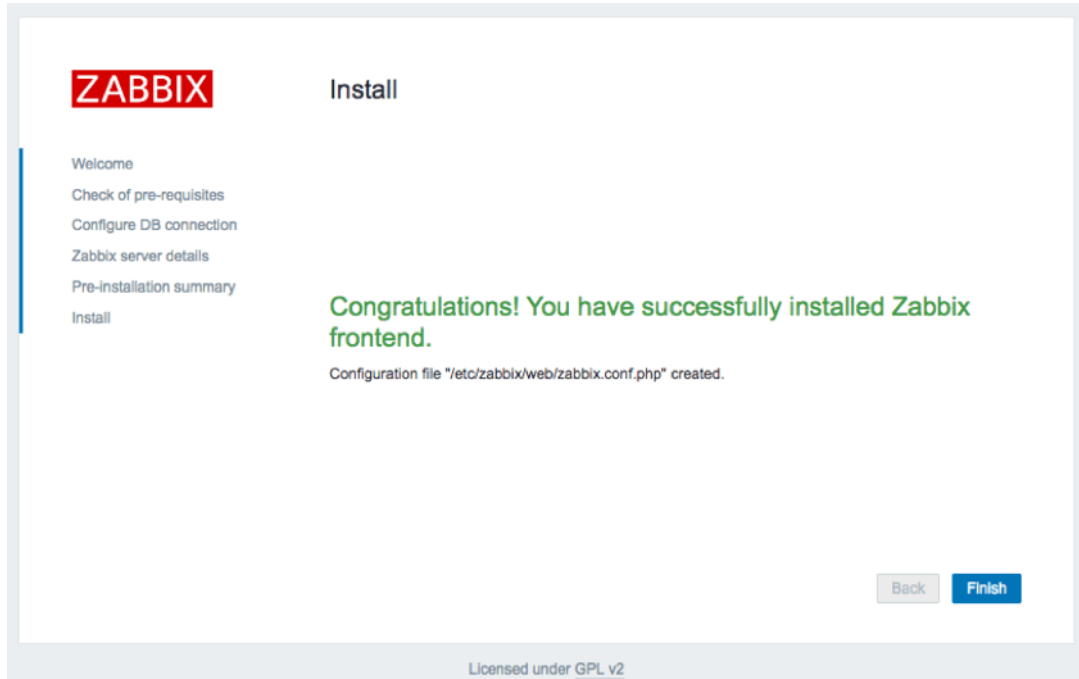


Figure 38. Zabbix frontend installation completed

Now you can log in to the Zabbix admin console. The default credentials are username Admin (with a capital!) and password zabbix.

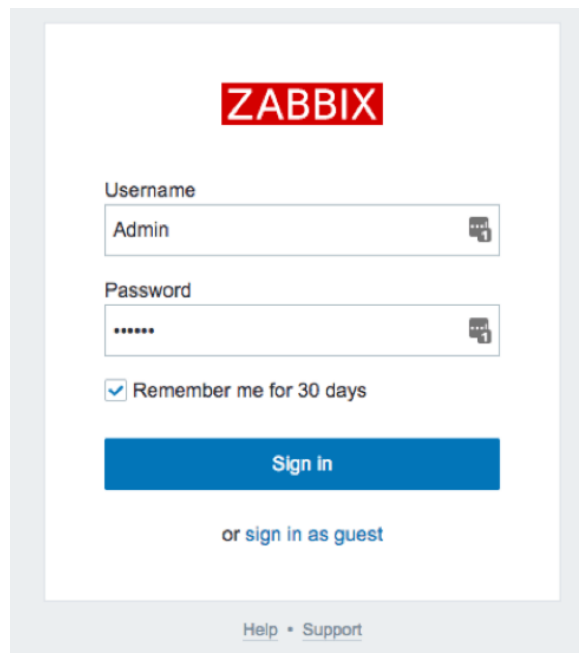


Figure 39. Log into the Zabbix admin console

Congratulations, you have a fully functional Zabbix Server:

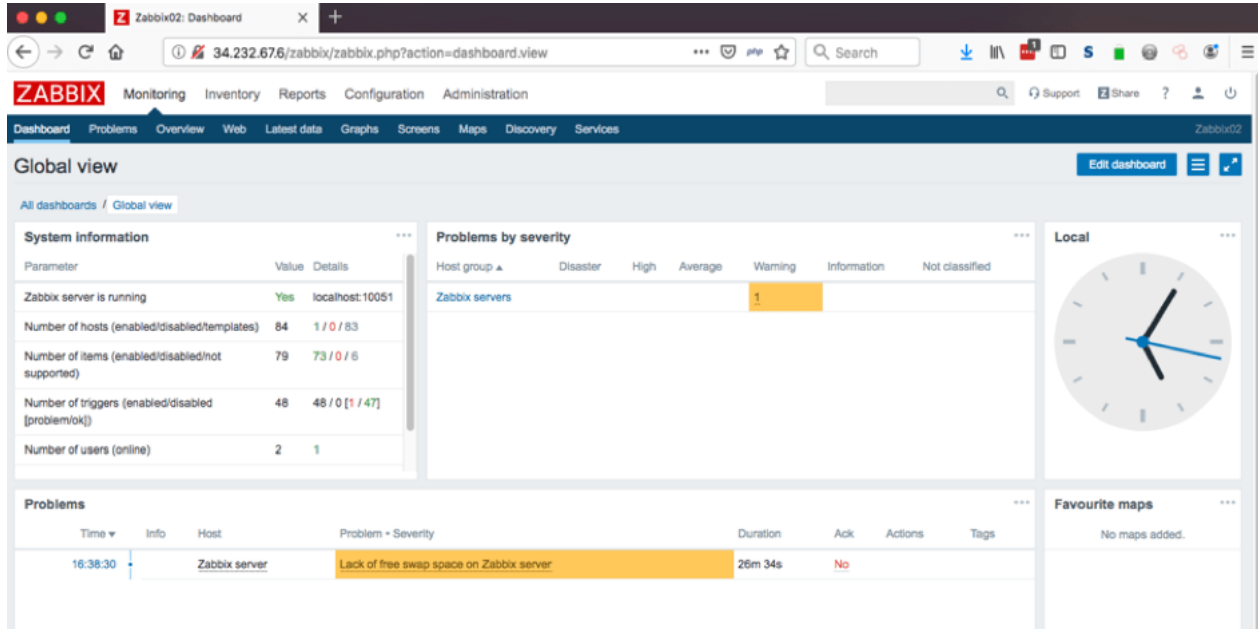


Figure 40. Zabbix Server

Next Steps

The next guides in this series explain how to install Zabbix agents on your FileMaker Servers (3 – Zabbix Agents) and how to add those FileMaker Servers as hosts to monitor here in the Zabbix Server (4 – Zabbix Configuration).

Disable MySQL Binary Logging

But there is one more important change that we want to make to MySQL before Zabbix Server starts to collect data.

We are running on an AWS t2.micro with 8GB of disk space, and at the end of the installation, we have just more than half of that disk space still available:

To check, type the following command and look at the Use % of the root directory (/):

```
df
```



```
[centos@ip-~]$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/xvda1      8377344 3410852  4966492  41% /
devtmpfs        483740      0    483740   0% /dev
tmpfs           506596      0    506596   0% /dev/shm
tmpfs           506596    13140    493456   3% /run
tmpfs           506596      0    506596   0% /sys/fs/cgroup
tmpfs           101320      0    101320   0% /run/user/1000
```

Figure 41. Disk space used

MySQL will collect binary logs (in folder /var/lib/mysql/) that will very quickly fill up that disk space. Those binary logs are only required if you intend to replicate this particular MySQL instance with others, and for our purpose, we do not. If you do want that default MySQL behavior, you will need to increase the disk size for this server.

In our deployment we want to disable those binary logs.

Type in:

```
sudo nano /etc/my.cnf
```

and scroll down to the section indicated in Figure 42:

```
GNU nano 2.3.1 File: /etc/my.cnf
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/8.0/en/server-configuration-defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove the leading "# " to disable binary logging
# Binary logging captures changes between backups and is enabled by
# default. It's default setting is log_bin=binlog
# disable_log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
#
```

Figure 42. Scroll down to “# disable_log_bin”

Remove the “#” at the start of the line so that “disable_log_bin” becomes active:

```
GNU nano 2.3.1 File: /etc/my.cnf
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/8.0/en/server-configuration-defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove the leading "# " to disable binary logging
# Binary logging captures changes between backups and is enabled by
# default. It's default setting is log_bin=binlog
# disable_log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
```

Figure 43. Remove the “#” from the line

Hit control-o and then enter to save the changes and then control-x to quit the text editor.

Restart MySQL for the change to take effect:

```
sudo systemctl restart mysqld
```

On to the next guide and installing Zabbix Agents (3 – Zabbix Agents).