Monitoring Your FileMaker Server

Installing Zabbix Agent

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

July 29, 2019

Table of Contents

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.

Do We Need an Agent?

Zabbix agents are responsible for collecting data from the host (FileMaker Server) being monitored. While Zabbix server can monitor servers and devices without the presence of an agent on the host, the amount of data you can collect, and its relevance, would be much less.

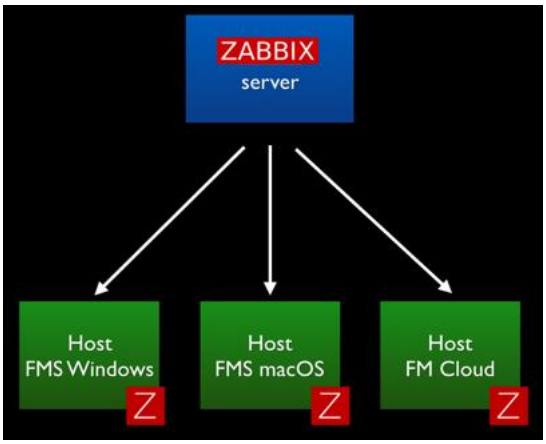

Figure 1. Zabbix Server

The agent is a small piece of software that runs completely in the background as a service/daemon. It is designed to be lightweight so that its monitoring activity does not affect the host that it is monitoring. These agents exist for all three of the platforms that matter for us: Windows, macOS and CentOS (FileMaker Cloud).

The Zabbix agent footprint is small. As an example, the screenshots below are from one of our Zabbix servers that monitors four development FileMaker Servers. Over the course of three months, the processor time for the Zabbix agent did not exceed 1% and used about 20MB of memory.

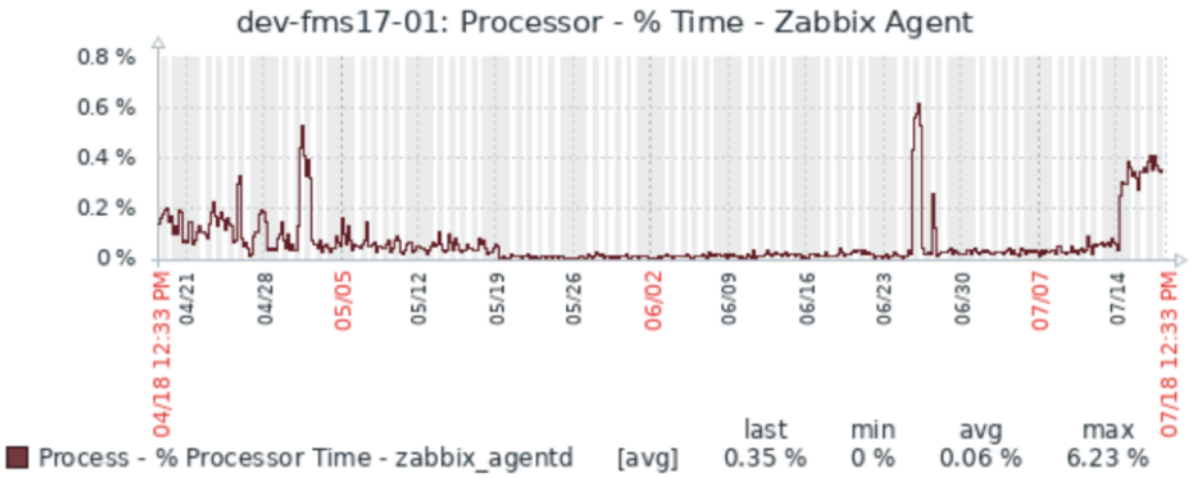

Figure 2. Zabbix Agent - processor time

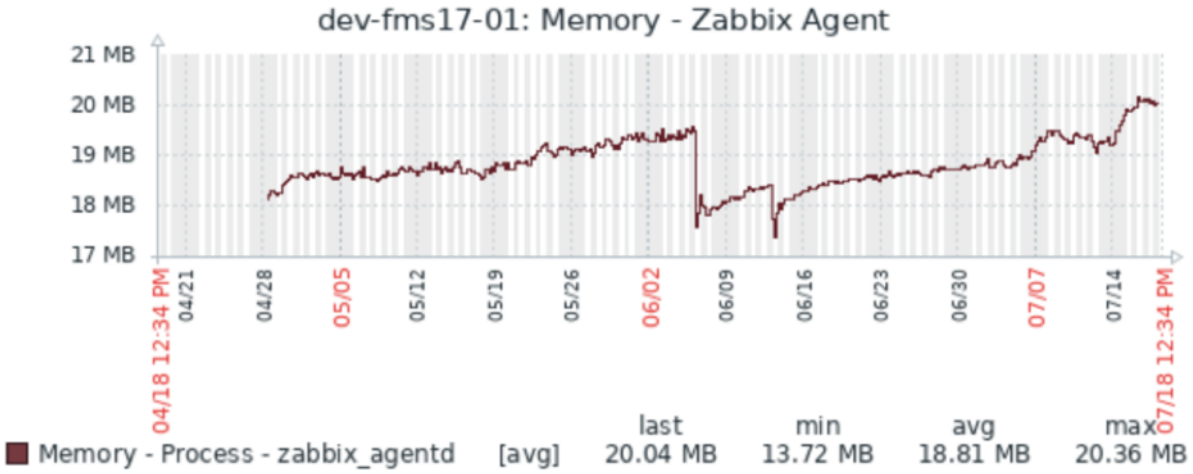

Figure 3. Zabbix Agent – memory

Active or Passive Agent and Firewall ports

Agents can operate in two modes – active or passive – and the difference can matter to you in terms of whether you are comfortable with opening an extra port on the FileMaker Server.

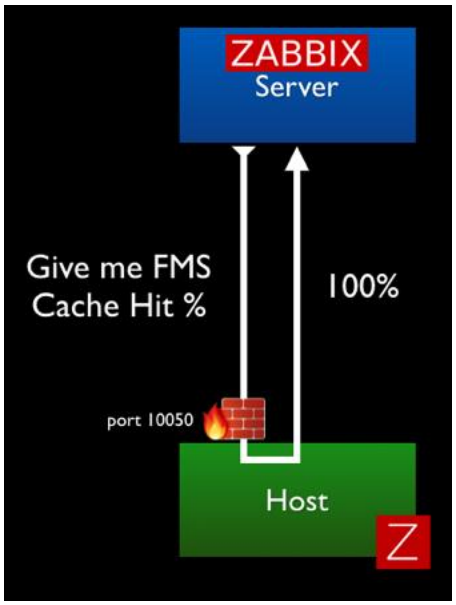

Figure 4. Zabbix Server – passive mode

In Passive mode, the agent does not do anything at all until it is asked to do something by the Zabbix server. The communication originates from the Zabbix server and requires port 10050[1] to open on the FileMaker Server to allow that incoming traffic.

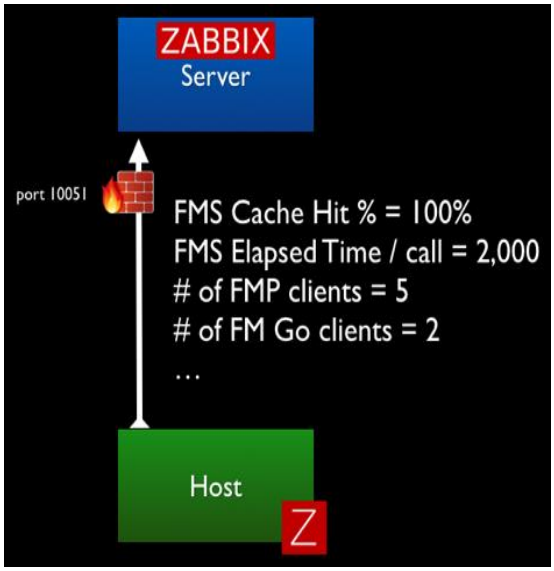

Figure 5. Zabbix Server – active move

In Active mode, the agent collects all of the required data on its own (based on the interval set for each item it collects data for) and sends that data to the Zabbix server. In this scenario all communication originates from the FileMaker Server; no ports need to be opened on the FileMaker Server. The Zabbix port 10051 needs to be open on the Zabbix server to accept the incoming data.

To use all of Zabbix's functionality, including the ability for Zabbix server to send remote commands to your FileMaker Server (for instance to restart the scripting engine), you'll need to allow traffic in both directions.

[1] These ports can be customized as we will show later.

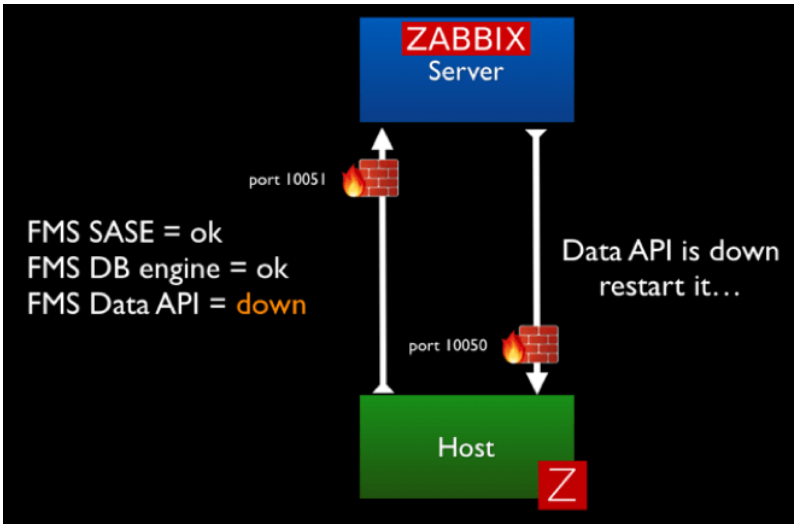

Figure 6. Allow traffic in both directions

Based on your security requirements, you can decide to forego some of the functionality around triggering remote actions and opt for a more locked-down deployment.

Installing the Agent

For macOS and Windows you can download the agent from the Zabbix download page. For FileMaker Cloud the installation is done through the CentOS software manager command line.

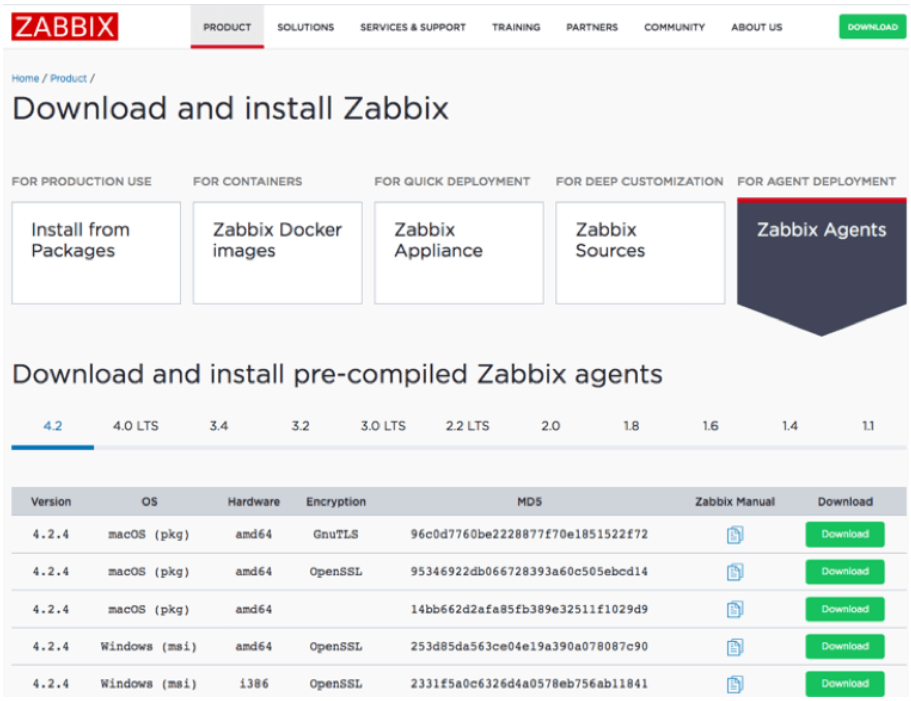

Figure 7. Zabbix download page

Note that you have multiple choices per platform depending on the encryption engine (GnuTLS, OpenSSL, no encryption). The main reason for offering different encryption engine options is so that if a vulnerability were to be discovered in one encryption platform, we can fairly seamlessly switch to another. In that sense, you can pick whichever one you prefer. There is no functional difference between the choices.

Installing on Windows

The Zabbix agent for Windows comes as a standard installer with the usual wizard that will walk you through some of the needed basic configuration details.

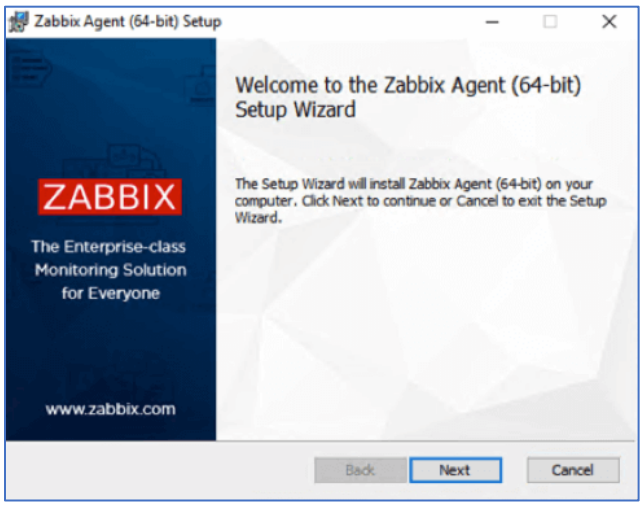

Figure 8. Zabbix Agent Windows installer

The choices you make on the next screen can all be modified in the Zabbix agent config file as will be shown later in this guide.

The Host Name gets set by default to the host name of your Windows machine. You can change it to something meaningful, provided that it is unique. The Host Name will be shown on the Zabbix server dashboard and is used when you set up a new host to monitor on your Zabbix server.

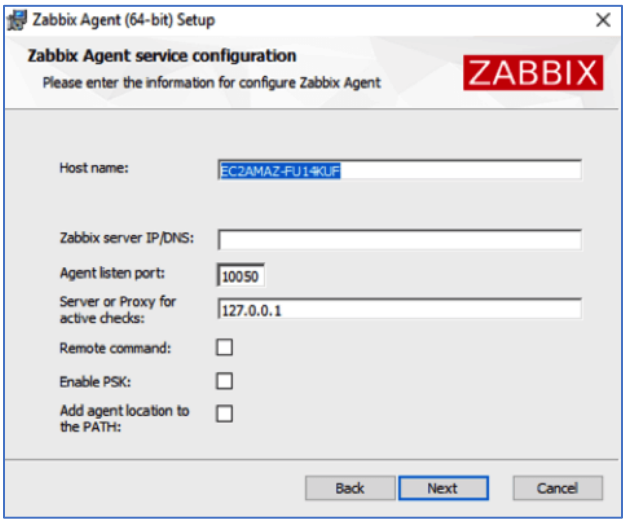

Figure 9. You can change the default host name

The **Zabbix server DNS name or IP address** will be used to inform the agent where to send data and as a security measure so that the agent will only respond to incoming traffic (passive requests for data, remote commands) from the Zabbix server(s) listed.

Port 10050 is the default port on the Agent-side to listen to those incoming requests. This port needs to be opened on your FileMaker Server's firewall or your perimeter firewall for your network and then forwarded from your router to your FileMaker Server. If your FileMaker Server is hosted on AWS or a similar provider, remember to adjust the inbound rules there. If you would rather not use the default port, you can adjust it here (or later by modifying the config file).

Typically, you would use the same DNS name or IP address for the **Server or Proxy for Active Checks** as you have for the Zabbix server DNS name earlier. This setting decides where the Agent will send the data it collects for Active items (where the agent does not get prompted by the server to collect data). In complex deployments, you could use a different Zabbix server or a Zabbix proxy for these active checks.

The **Remote command** toggle is to decide whether you will allow this Agent to accept remote commands from the Zabbix server listed. We do use this functionality in our FileMaker Server templates to restart processes like the FileMaker Server scripting engine, Data API, or Web Publishing Engine if they have stopped running.

By **enabling PSK**, you encrypt the traffic between the Agent and the Server through a Pre-shared Key. This security scheme is similar to how most Wi-Fi networks work.

Enabling the option to **add the agent location to the PATH** will ensure that you can use the Zabbix agent command line commands from anywhere on the machine without first having to navigate to where those executables are. That is similar to how "fmsadmin" works on your FileMaker Server; the FileMaker Server installer does this automatically.

For our deployment, the configuration looks like Figure 10.

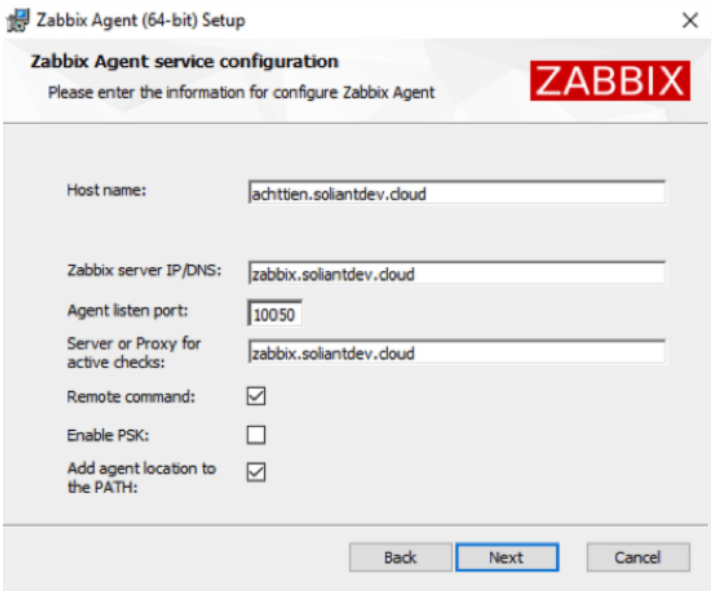

Figure 10. Deployment configuration

The core of the Zabbix agent is the 'Agent Daemon'. By default, the installer will also install the Zabbix Sender and Zabbix Get, which are command line tools to manually initiate sending data to the Zabbix server or retrieve information from the Zabbix server about what active items for which to collect data.

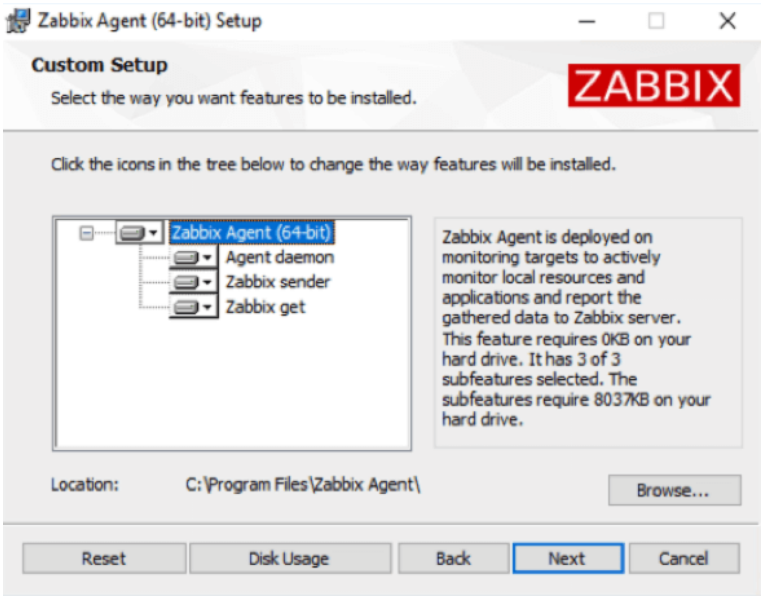

Figure 11. Custom setup

When the installer has completed, you will find the Zabbix agent listed among the Windows services. Like most background services, it runs under the "local system" account.

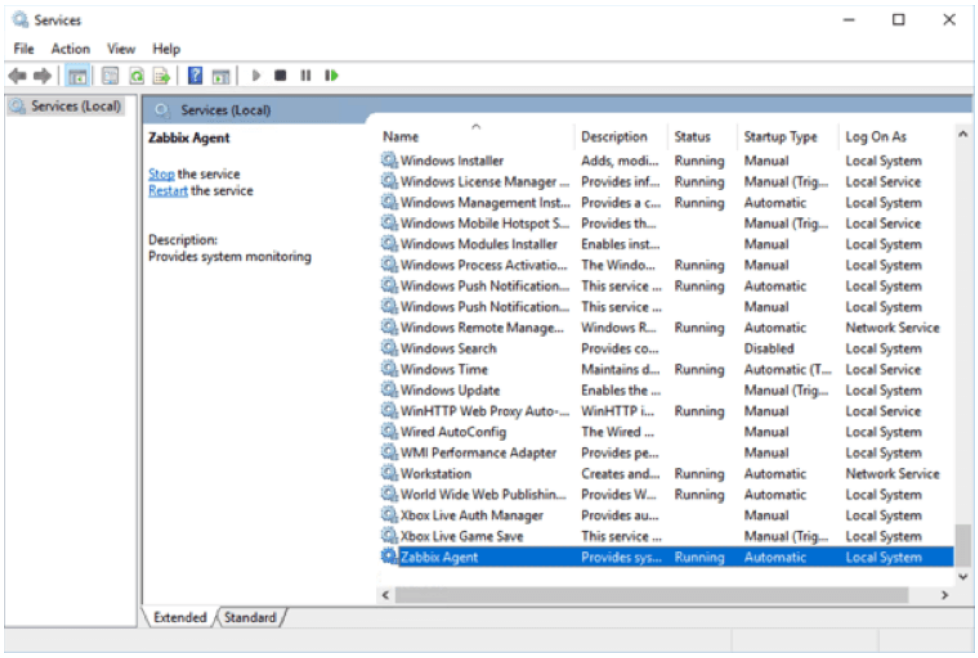

Figure 12. Zabbix Agent shown under Windows services

The log file for troubleshooting is in the Zabbix agent install location under “Program Files”:

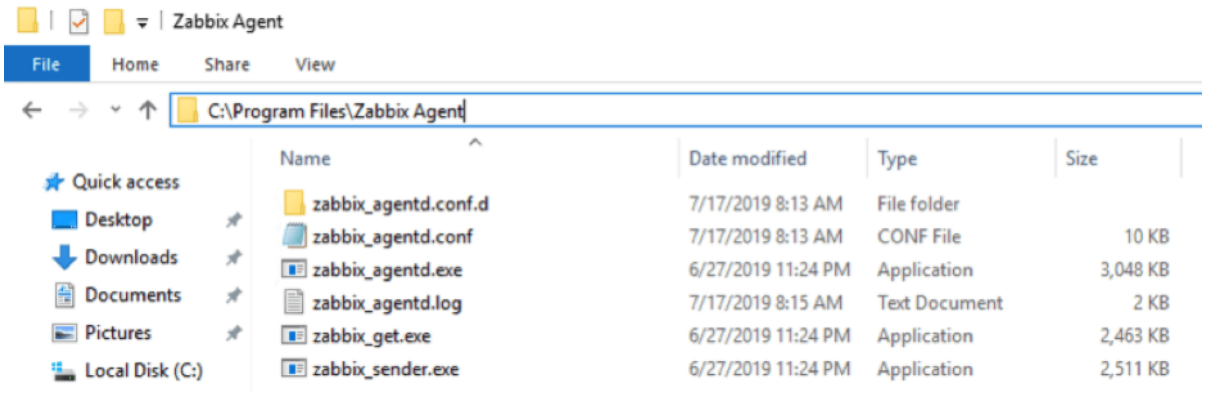

Figure 13. Log file

Later in this guide, we will make some modifications to the **zabbix_agentd.conf** file, located in this same folder, to further tweak our deployment.

Installing on macOS

Similarly, on macOS the Agent’s installer, will walk you through the standard wizard:

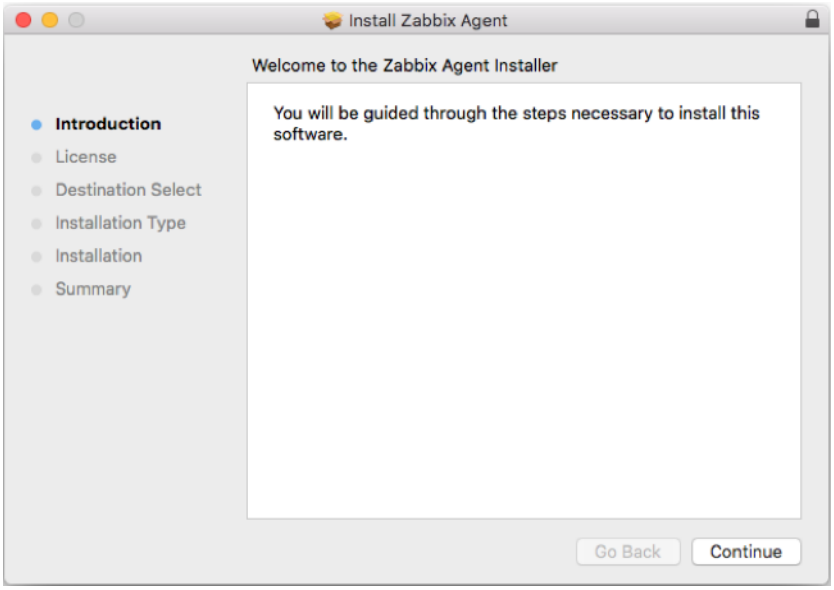

Figure 14. Zabbix Agent macOS installer

But it will not provide any options to change configuration settings up-front. We will show you how to modify the config file to set the relevant options.

The macOS installer adds a Zabbix user account responsible for running the daemon. This will be relevant later on when we make our configuration changes.

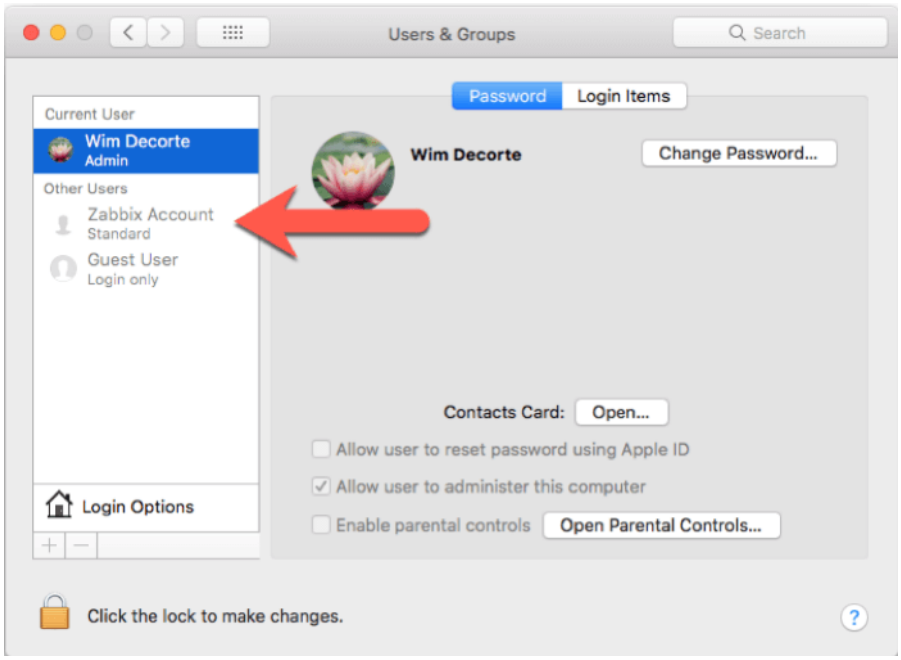

Figure 15. Zabbix user account is added during installation

Adding the zabbix user to sudoers

As part of our Zabbix template, we use some of the macOS and FileMaker Server command line functionality to collect (and take action on) data for items we monitor. As such, the Zabbix agent user needs the right level of privileges to execute those commands.

To make this work seamlessly through the security features available in macOS, we will use the sudoers file.

First off, open Terminal and type in this command to create a new file in the sudoers folder:

```
sudo nano /etc/sudoers.d/zabbix_nopasswd
```

In the nano text editor window:

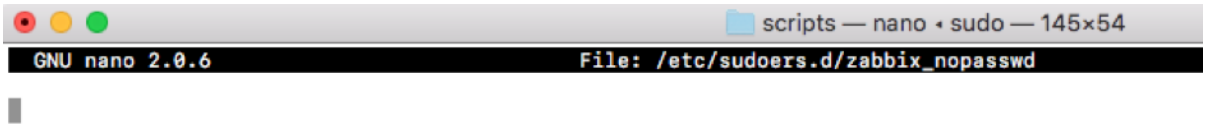

Figure 16. Nano text editor

Type in or paste in the following line:

```
zabbix ALL=(ALL) NOPASSWD: ALL
```

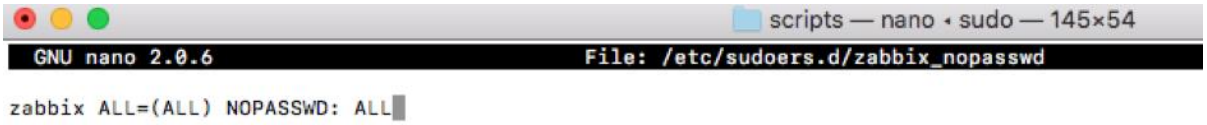

Figure 17. Edit file added to sudoers folder

Hit control-o and then enter to save the file and then control-x to quit out of the nano text editor and return back to the command line.

Type in the following command to restrict that new file's access level so that it is read-only for the owner of the file and the group to which the owner belongs. (This further protects it from inadvertent changes.)

```
sudo chmod 0440 /etc/sudoers.d/zabbix_nopasswd
```

With this done, we'll instruct macOS to read this new file when evaluating the rights of a certain user to run commands as Super-User (aka the **su** in sudo).

The safest way to do this is through the command line utility **visudo**. **vi** is another text editor, and **visudo** is made specifically to be the text editor for everything that has to do with changes to the super-user do (sudo) levels.

```
sudo visudo
```

The Terminal window will change to the **vim** text editor (a variant of the **vi** text editor) – which most people find difficult to work with unless you use it very often.

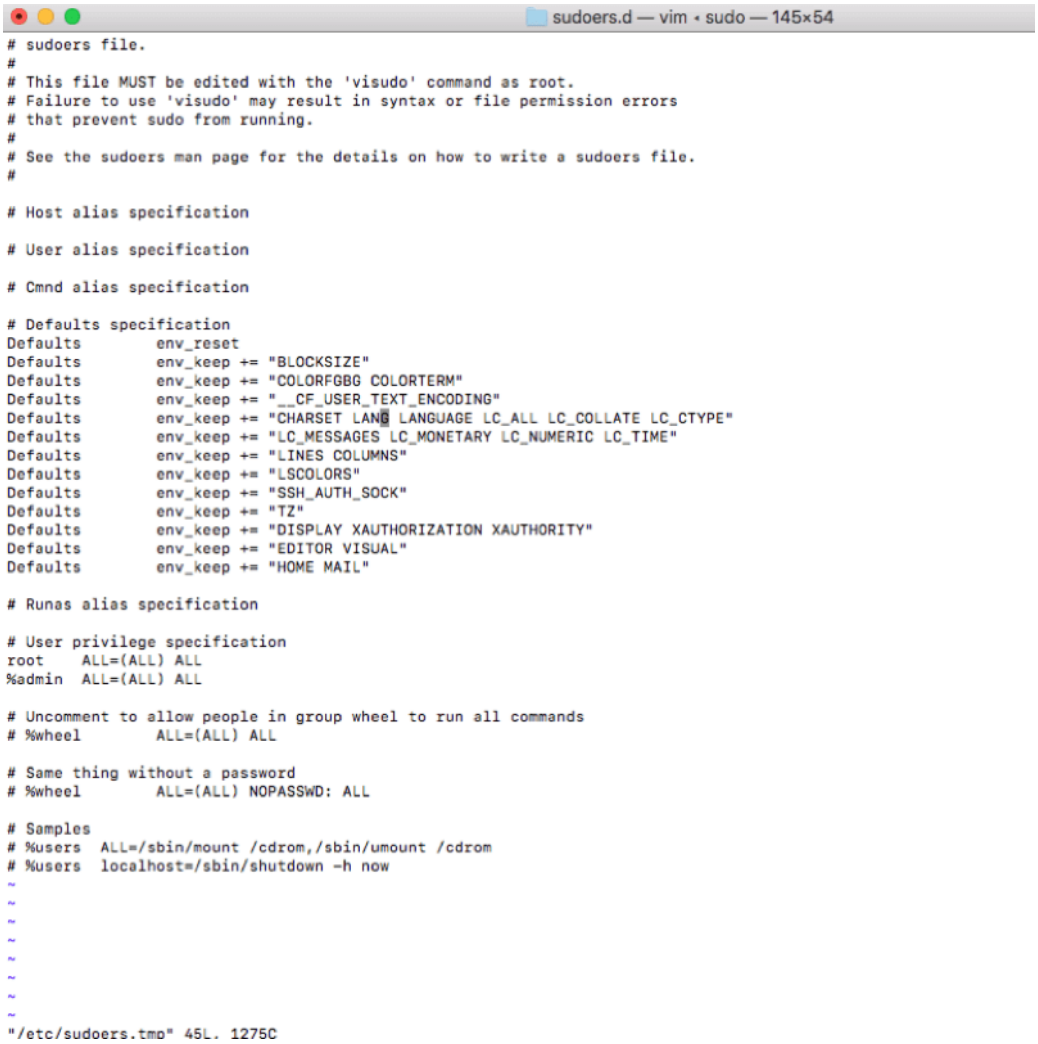

Figure 18. Vim text editor

Using the arrow keys, scroll down to the bottom of the file and hit **i** on your keyboard to enter edit (insert) mode. The bottom of the window will confirm that you are in edit mode:

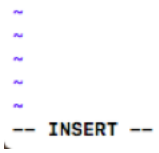

*Figure 19. Scroll to the bottom of the file and hit the "**i**" key*

Add the following two lines:

Read drop-in files from /etc/sudoers.d (## indicates a comment line; # does not)
#includedir /etc/sudoers.d

To exit edit mode, hit **escape** on your keyboard and type in **:wq** and then enter to save the document and quit vim. This will place you back on the command line.

The syntax of the file we have just added to the sudoers folder is crucial to the operating system. To ensure you did not make any syntax errors use this command:

sudo visudo -cf /etc/sudoers.d/zabbix_nopasswd

When all is well, you should see "parsed OK" in the result of that command:

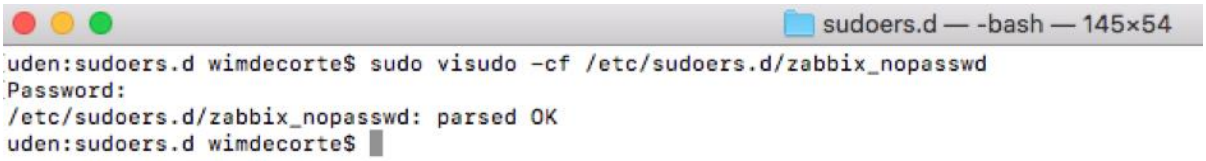

Figure 20. "parsed OK" is shown when done

Python requests module

As part of our template, we will use a small Python script on the FileMaker Server machine (PowerShell on Windows) to communicate with the FileMaker Server Admin API and retrieve configuration settings. The Admin API is only available in FileMaker Server 18 (and in 17 until its expiry on September 27, 2019).

Recent versions of macOS have Python 2.7.10 installed by default[2] so we made sure that the Python script is compatible with that – somewhat old – version of Python. To make the REST request to the Admin API, we want to use Python's requests module.

[2] See https://opensource.apple.com/, for each version of macOS, you can click through to see what version of Python was installed. Python 2.7.10 is included in all versions since 10.10 (Yosemite). Because FileMaker Server 18 requires macOS 10.13 (High Sierra) or 10.14 (Mojave) and 17 requires macOS 10.12 (Sierra) or 10.13 (High Sierra), we know that the right version of Python is available on all macOS servers running FileMaker Server that support the Admin API.

That module, however, is missing from the standard macOS Python installation and so is Python's software package installer (pip).

First, we need to install **pip**:

sudo python -m ensurepip --default-pip

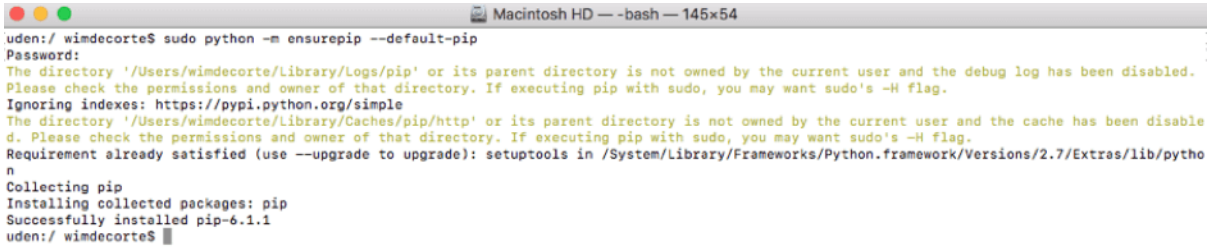

Figure 21. Installing pip

And with pip installed, we can install the requests module:

sudo python -m pip install requests

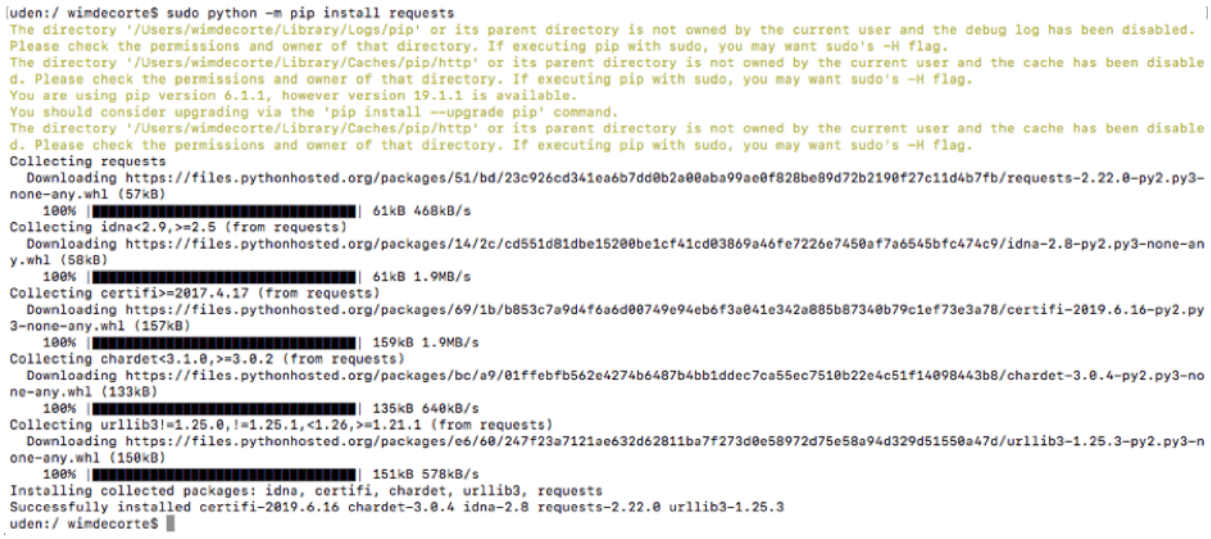

Figure 22. Installing the request module

Starting, Stopping the agent and where to find the log file

To start the agent, use this command in Terminal:

sudo launchctl start com.zabbix.zabbix_agentd

Or, use **stop** to stop the agent, particularly after making changes to the Zabbix agent config file which necessitates an agent restart.

The log file is in this folder: **/var/log/Zabbix/Zabbix_agentd.log** and contains very useful troubleshooting information.

Installing on FileMaker Cloud

FileMaker Cloud runs on Linux CentOS. The Zabbix downloads page does not offer a pre-compiled agent for that operating system. Instead, all software installations on CentOS are done through its built-in command line software package manager: yum[3].

Since we need access to the command line, we need to establish an SSH connection to the server. FileMaker Cloud instances do not allow this by default, so we need to change the inbound rules in the AWS EC2 console. Select your FileMaker Cloud instance and click on the security group that applies to it:

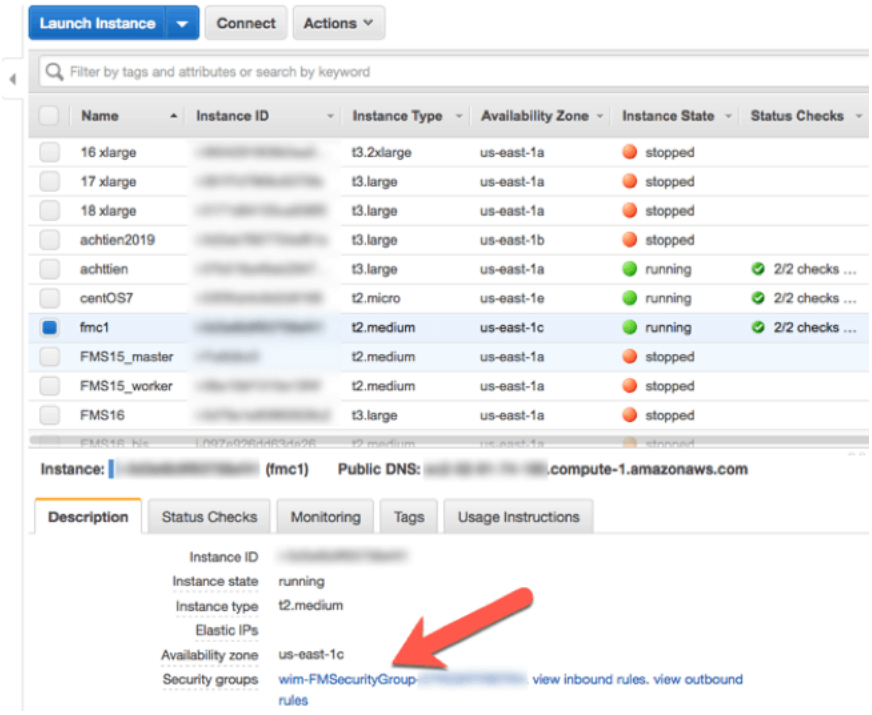

Figure 23. Click on the security group for the selected FileMaker Cloud instance

In the security group settings, select inbound rules and adjust them so that:

- Port 22 (SSH) is allowed but only from your IP address
- Port 10050 is allowed but only from the IP address of your Zabbix server

[3] Yum = Yellow dog Updater, Modified. If you are familiar with other flavors of Linux, it is the equivalent of "apt-get".

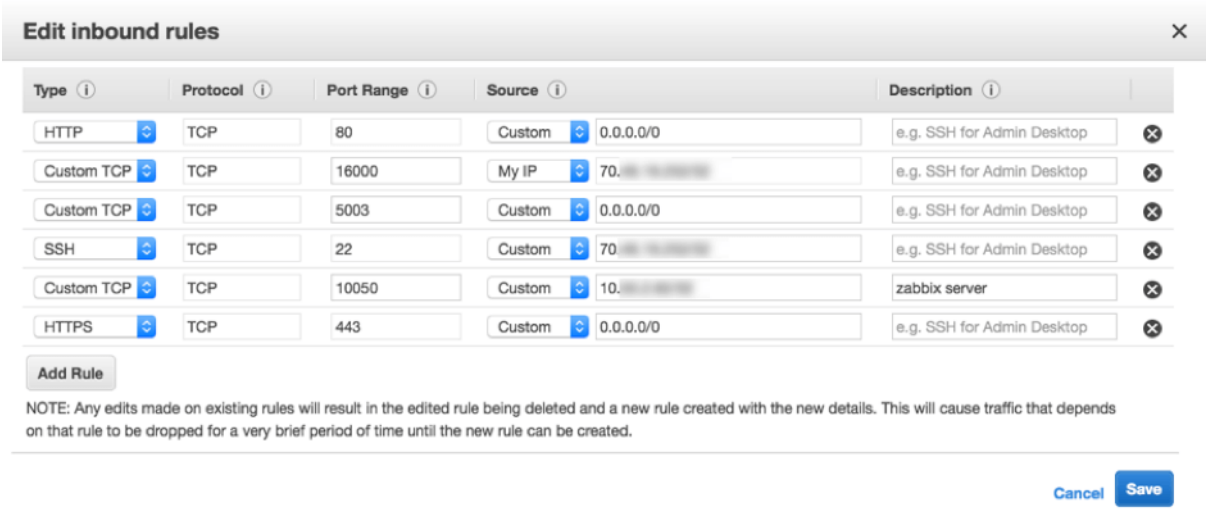

Figure 24. Inbound rules

With this done, we can now open Terminal on macOS or your favorite SSH client on Windows and connect to the FileMaker Cloud instance:

ssh -i /Users/wimdecorte/Documents/projects/ETS18/zabbix_resources/wim_ets_15.pem centos@<IP or DNS name of your FileMaker Cloud instance>

All SSH connections to AWS instances require the use of the pem file (certificate) that was used to create the instance. You can do this by specifying the **-i** and the path to that pem file. **centos** is the default user name to log into CentOS Linux.

Before we go on, we have to mention a big caveat: any and all configuration changes that we make from this point forward may get lost through the automatic updates that happen on FileMaker Cloud instances. There is nothing that can be done about this, since that is the architecture of FileMaker Cloud. A FileMaker Cloud instance consists of a number of drives, one of which holds your FileMaker Data and all the FileMaker Server configuration settings. The other drives hold the Linux operating system and its configuration, and those drives get replaced from time to time with Linux system updates.

What does this mean for Zabbix monitoring? After a FileMaker Cloud upgrade, you may have to redo the steps in this section, so it is a good idea to save a copy of the configuration file after making changes to it.

The very first thing we will do is install nano, our favorite Linux text editor. We will need it to change the Zabbix agent configuration.

Type in:

sudo yum install nano

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:

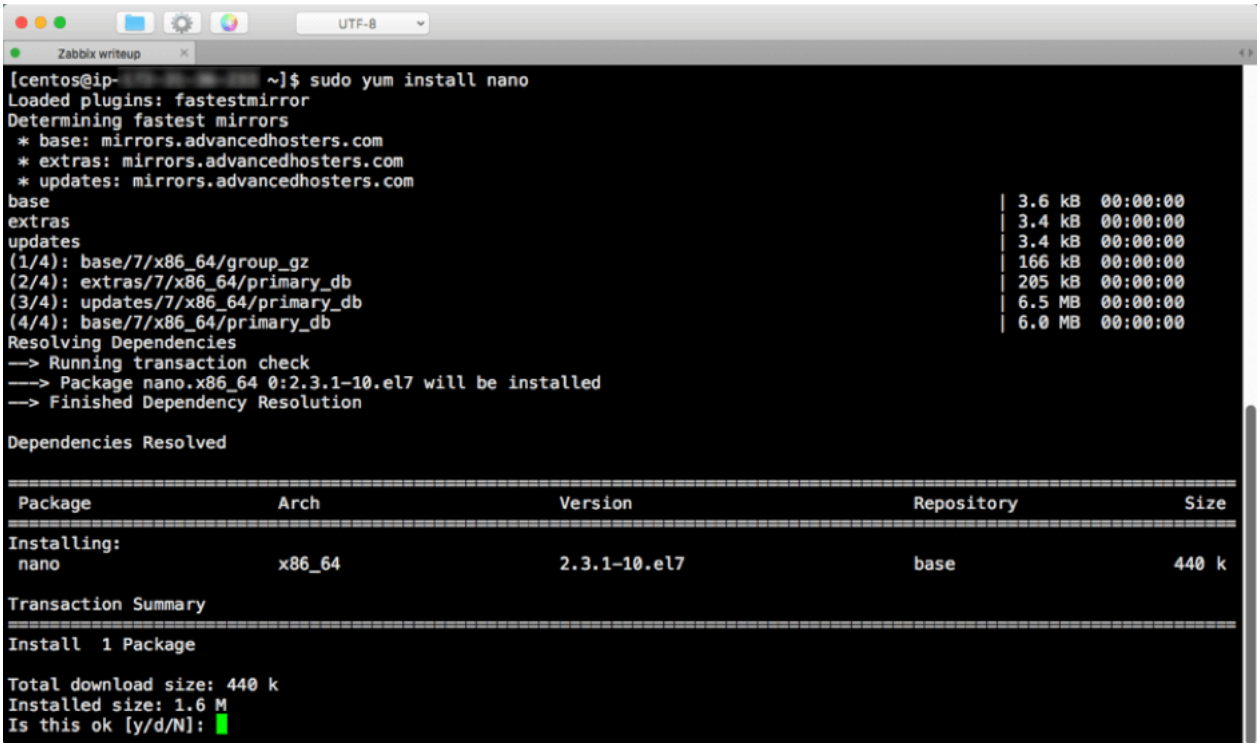

Figure 25. Type “Y” to proceed

A few seconds later, we will be done:

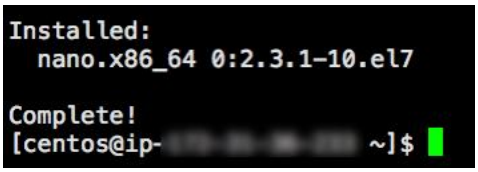

Figure 26. Nano installation completed

Type in these two commands:

sudo firewall-cmd --zone=public --add-port=10050/tcp –permanent

sudo firewall-cmd --reload

You can check what ports are open with this command, to confirm that the port is now open:

sudo firewall-cmd --list-ports

Figure 27. View ports that are open

Yum, the software package manager used by CentOS, keeps a list of repositories with available software that can be installed. The Zabbix repository is not listed by default, so we will need to add it with this command:

sudo rpm -Uvh https://repo.zabbix.com/zabbix/4.2/rhel/7/x86_64/zabbix-release-4.2-1.el7.noarch.rpm

followed by this command to tell yum to do some internal housekeeping:

sudo yum clean all

And finally, we can run the command to install the Zabbix agent:

sudo yum install -y zabbix-agent

And these two commands to start it and set it to auto-start whenever the machine boots:

sudo systemctl start zabbix-agent

sudo systemctl enable zabbix-agent

The next section of this guide will step you through the Zabbix agent configuration.

Configuration changes for Zabbix agent

On Windows, the configuration file will be in **C:\Program Files\Zabbix Agent** unless you changed the installation location during the install. On macOS you will find the configuration file in **/usr/local/etc/zabbix/**. And on FileMaker Cloud it is located in **/etc/zabbix/**.

The configuration file is always named **zabbix_agentd.conf**, and its content is the same on all platforms.

On Windows, the installer will have asked for some configuration options already. However, this will not have happened on macOS and FileMaker Cloud, so we will review all the changes here that make our Zabbix server installation work, specifically for monitoring a FileMaker Server.

On Windows, we usually install Notepad++, which allows us to create a custom 'language' that colors all the comments in green for easy reading:

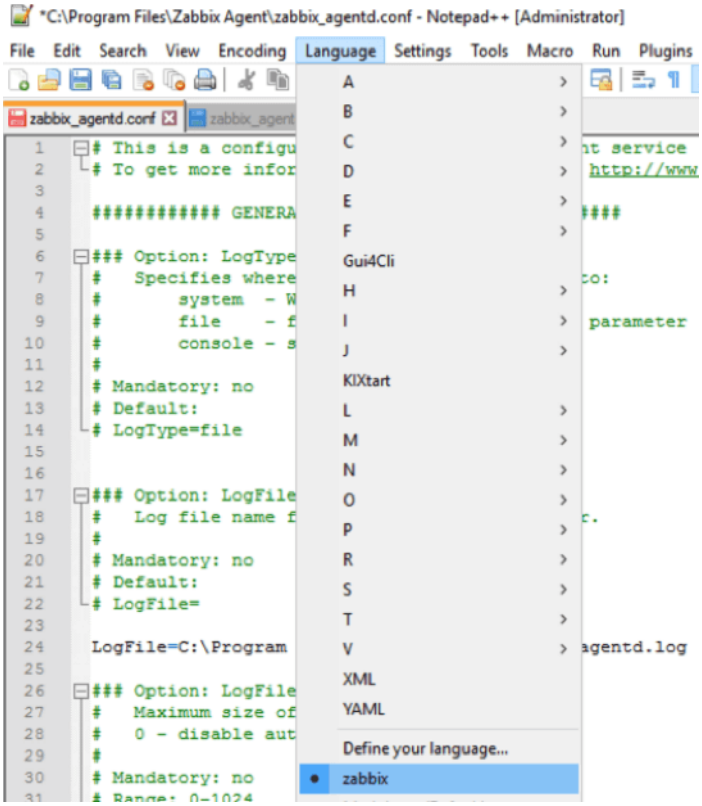

Figure 28. Notepad ++ on. Windows

You can download that language file at https://github.com/soliantconsulting/FileMaker-Server-Zabbix-Templates.

On macOS and FileMaker Cloud, you can edit the config file from the command line by using the nano text editor:

macOS:

```
sudo nano /usr/local/etc/zabbix/zabbix_agentd.conf
```

FileMaker Cloud:

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

Using the command line on both macOS and FileMaker Cloud ensures that the privileges on the file do not change. On macOS, you could certainly use your favorite text editor but make sure that the privileges do not change from what they need to be for the Zabbix agent to work properly:

```
uden:zabbix wimdecorte$ pwd
/usr/local/etc/zabbix
uden:zabbix wimdecorte$ ls -al
total 24
drwxr-xr-x  4 root  wheel    136 Jul 17 08:30 .
drwxr-xr-x  3 root  wheel    102 Jun 27 05:32 ..
drwxr-xr-x  4 root  wheel    136 Jul 17 08:30 zabbix_agentd
-rw-r--r--  1 root  wheel  10837 Jun 27 05:32 zabbix_agentd.conf
uden:zabbix wimdecorte$
```

Figure 29. Ensure privileges do not change

Enable Remote Commands

This setting allows Zabbix server to send commands to the FileMaker server as part of a configured Action; for instance, to restart the FileMaker Server Scripting Engine process when it fails. If you enable this setting, we also recommend enabling the setting that logs each executed remote command. But note that doing so will result in the FileMaker Server admin console credentials being included in the agent log for all of the items and remote actions that rely on the fmsadmin utility. (We will cover items and actions in more detail in the Zabbix Configuration white paper.)

Note that from a security point of view, the Zabbix agent will only accept remote commands from servers listed in the “Active” section (see later).

```
55
56  ### Option: EnableRemoteCommands
57  #   Whether remote commands from Zabbix server are allowed.
58  #   0 - not allowed
59  #   1 - allowed
60  #
61  # Mandatory: no
62  # Default:
63  # EnableRemoteCommands=0
64  EnableRemoteCommands=1
65
66  ### Option: LogRemoteCommands
67  #   Enable logging of executed shell commands as warnings.
68  #   0 - disabled
69  #   1 - enabled
70  #
71  # Mandatory: no
72  # Default:
73  LogRemoteCommands=1
74
```

Figure 30. Remote commands enabled

Set Zabbix server & the port that the Agent listens to

These settings are relevant for passive checks, where Zabbix server talks to the agent to ask it to collect data for a monitored item or to run a remote command.

We have left the port setting at its default of 10050, but this is where you can change it. The port is also specified in the Zabbix frontend and, as was discussed earlier, in the firewall settings. If you end up changing it in the configuration file, don't forget to also change it in those other places.

```
77  ### Option: Server
78  #   List of comma delimited IP addresses, optionally in CIDR notation, or DNS
79  #   Incoming connections will be accepted only from the hosts listed here.
80  #   If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.
81  #   '0.0.0.0/0' can be used to allow any IPv4 address.
82  #   Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.domain
83  #
84  # Mandatory: yes, if StartAgents is not explicitly set to 0
85  # Default:
86  # Server=
87
88  Server=zabbix.soliantdev.cloud
89
90  ### Option: ListenPort
91  #   Agent will listen on this port for connections from the server.
92  #
93  # Mandatory: no
94  # Range: 1024-32767
95  # Default:
96  # ListenPort=10050
97
```

Figure 31. This is where the port setting can be changed

The Zabbix agent will only listen to requests from the server that is listed here.

Set Zabbix server to send data to

The previous section determines which Zabbix server the agent will listen to, and this section defines which Zabbix server the agent will send its data to for Active[4] items.

```
117
118  ### Option: ServerActive
119  #    List of comma delimited IP:port (or DNS name:por
120  #    If port is not specified, default port is used.
121  #    IPv6 addresses must be enclosed in square bracke
122  #    If port is not specified, square brackets for IP
123  #    If this parameter is not specified, active check
124  #    Example: ServerActive=127.0.0.1:20051,zabbix.dom
125  #
126  # Mandatory: no
127  # Default:
128  # ServerActive=
129
130  ServerActive=zabbix.soliantdev.cloud
131
```

Figure 32. Send data for Active items to the specified Zabbix server

Hostname

The hostname will be used to reference the FileMaker server on which the Agent is running. The same name will be used when setting up the monitored host in the Zabbix frontend. It needs to be unique among all the servers monitored by the Zabbix server. Using the DNS name of the FileMaker Server is an easy way to ensure that.

```
131
132  ### Option: Hostname
133  #    Unique, case sensitive hostname.
134  #    Required for active checks and must match hostname as configured on the server.
135  #    Value is acquired from HostnameItem if undefined.
136  #
137  # Mandatory: no
138  # Default:
139  # Hostname=
140
141  Hostname=achttien.soliantdev.cloud
142
```

Figure 33. Define the host name

[4] Active vs. Passive is described earlier in this document.

Advanced Parameters – Timeout

The timeout setting is located a lot further down in the config file, and it specifies how long the Zabbix agent is going to spend on any one request. The default is three seconds, but we will ask it to do some things that could take longer as you will see later.

```
223
224 ### Option: Timeout
225 #    Spend no more than Timeout seconds on processing.
226 #
227 # Mandatory: no
228 # Range: 1-30
229 # Default:
230 Timeout=30
231
```

Figure 34. Setting the timeout

User Defined Monitored Parameters – allow unsafe parameters

This setting sounds scarier than it is. It allows us to send certain characters which Zabbix considers unsafe – such as slashes and spaces – as parameters to remote commands that the Agent will execute.

```
258
259 ####### USER-DEFINED MONITORED PARAMETERS #######
260
261 ### Option: UnsafeUserParameters
262 #   Allow all characters to be passed in arguments to user-defined parameters.
263 #   The following characters are not allowed:
264 #   \ ' " ` * ? [ ] { } ~ $ ! & ; ( ) < > | # @
265 #   Additionally, newline characters are not allowed.
266 #   0 - do not allow
267 #   1 - allow
268 #
269 # Mandatory: no
270 # Range: 0-1
271 # Default:
272 UnsafeUserParameters=1
273
```

Figure 35. Allow unsafe parameters

User Defined Monitored Parameters – UserParameter

This configuration option will be discussed at length in the Zabbix Configuration white paper.

The “scripts” folder and the "fms_config.ps1" PowerShell script referenced in the screenshot are items that we deployed to the FileMaker Server machine; they are not part of the default Zabbix agent installation.

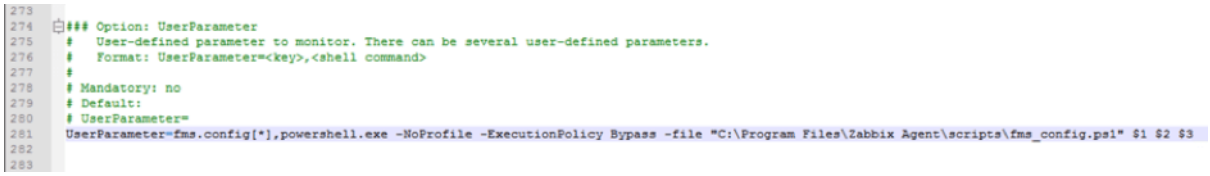

Figure 36. Set the UserParameter

The UserParameter configuration is largely the same on macOS and FileMaker Cloud, except that a Python script is called instead of a PowerShell script. The path to the scripts folder and the call syntax are also different between macOS and FileMaker Cloud.

FileMaker Cloud

```
### Option: UserParameter
#       User-defined parameter to monitor. There can be several user-defined parameters.
#       Format: UserParameter=<key>,<shell command>
#       See 'zabbix_agentd' directory for examples.
#
# Mandatory: no
# Default:
# UserParameter=

UserParameter=fms.config[*],/etc/zabbix/scripts/fms_config.py $1 $2 $3
```

Figure 37. UserParameter configuration in FileMaker Cloud

macOS

```
### Option: UserParameter
#       User-defined parameter to monitor. There can be several user-defined parameters.
#       Format: UserParameter=<key>,<shell command>
#       See 'zabbix_agentd' directory for examples.
#
# Mandatory: no
# Default:
# UserParameter=

UserParameter=fms.config[*],python /usr/local/etc/zabbix/scripts/fms_config.py $1 $2 $3
```

Figure 38. UserParameter in macOS

The PowerShell and Python script files are available https://github.com/soliantconsulting/FileMaker-Server-Zabbix-Templates.

Restart Zabbix agent Service

Whenever you make changes to the configuration file, you will need to restart the agent for those changes to take effect.

On Windows, use the Windows Services Control Panel to restart the agent. On macOS, use these commands:

```
sudo launchctl stop com.zabbix.zabbix_agentd

sudo launchctl start com.zabbix.zabbix_agentd
```

And on CentOS (FileMaker Cloud) use this command:

```
sudo systemctl restart zabbix-agent
```

The next guide in the series will walk you through how to import the FileMaker Server templates into the Zabbix admin console and configure your first FileMaker Server to be monitored.