CONFIGURATION GUIDES › WEB SERVERS

Apache server 2.4 Monitoring for windows

Applicare 10.0 · Updated August 22, 2024 · apache · **Pending technical review**

Follow the steps for installing Applicare SingleAgent

Step 1: To enable server-status and custom log modules in Apache by navigating to the Apache installed directory and editing the conf file.

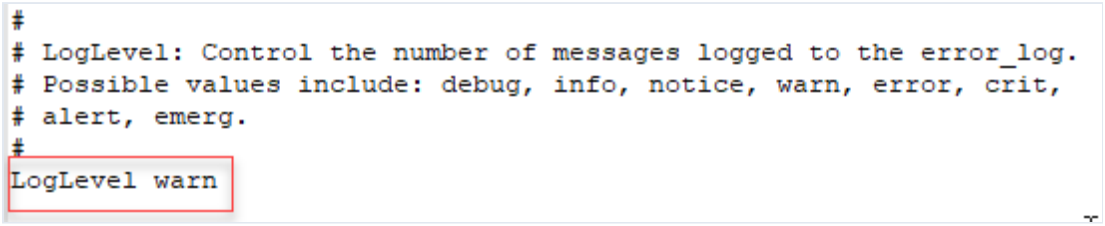
Step 2: Ensure the following modules are enabled in the conf file. If they are not, enable them by removing '#' at the beginnings of the lines.

Some of the modules may not be present in the Apache conf file by default. In that cases, manually add the following modules below the Apache Listen line in the Apache conf file.

```text
#Listen 12.34.56.78:80
Listen 80

LoadModule status_module modules/mod_status.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule unique_id_module modules/mod_unique_id.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule headers_module modules/mod_headers.so
```

Step 3: Ensure the **LogLevel** is set to **warn**. If it is set to a **LogLevel** is different value, comment out existing **LogLevel** line.



Step 4: Create a new log folder within the Apache home directory.

Step 5: Append the following five lines to the end of the **log_config_module** and section in the **CustomLog** and leave the existing **log_config_module** path unchanged.

```
TEXT
SetEnvIfNoCase Request_URI "^/server-status/pingFromApplicare" dontlog
SetEnvIfNoCase Request_URI "^/applicare_agent/*" dontlog
SetEnvIf _ap_trace_id ^[a-zA-Z0-9].* HAVE_AP_ID
RequestHeader set _ap_trace_id "%{UNIQUE_ID}e" env=!HAVE_AP_ID
RequestHeader set _ap_apache_id "<Apache_servername>"
CustomLog "|<RotateLogPath> -n 1440 <ApacheInstalledDir>/newlogfolder/apl_log 60" "%t %h %s
%B %D %U :%{_ap_trace_id}i" env=!dontlog
```

1. Replace **<Apache_servername>** with a custom name (without spaces) and include double quotes **(Need to remove the angle brackets too)**. The name can be **<Apache_servername>**. But it must be unique.

2. Replace **<RotateLogPath>** with the path to your **rotatelogs.exe** file. **(and remove the angle brackets).** By default, it is located in the Apache installation directory **<ApacheInstalledDir>/bin/rotatelogs.exe**

3. Replace **"<ApacheInstalledDir>"** with the path to your Apache installation **(and remove the angle brackets).**

4. Replace **<newlogfolder>**with the name of the folder you created inside the Apache installation directory **(add remove the angle brackets too)**.

```
<IfModule log_config_module>
    #
    # The following directives define some format nicknames for use with
    # a CustomLog directive (see below).
    #
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %b" common

    <IfModule logio_module>
      # You need to enable mod_logio.c to use %I and %O
      LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
    </IfModule>

    #
    # The location and format of the access logfile (Common Logfile Format).
    # If you do not define any access logfiles within a <VirtualHost>
    # container, they will be logged here.  Contrariwise, if you *do*
    # define per-<VirtualHost> access logfiles, transactions will be
    # logged therein and *not* in this file.
    #
    #CustomLog "logs/access.log" common

    #
    # If you prefer a logfile with access, agent, and referer information
    # (Combined Logfile Format) you can use the following directive.
    #

    CustomLog "logs/access.log" combined "expr=%{REQUEST_STATUS} > 200"
    #CustomLog "logs/access.log" combined

    SetEnvIfNoCase Request_URI "^/server-status/pingFromApplicare" dontlog
    SetEnvIf _ap_trace_id ^[a-zA-Z0-9].* HAVE_AP_ID
    RequestHeader set _ap_trace_id "%{UNIQUE_ID}e" env=!HAVE_AP_ID
    RequestHeader set _ap_apache_id "Apache"
    CustomLog "|E:/Apache/Apache24/bin/rotatelogs.exe -n 1440 E:/Apache/Apache24/newlogfolder/apl_log 86400" "%t %h %s %B %D %U :%{_ap_trace_id}i" env=!dontlog
</IfModule>
```

Step 6: Append the following modules to the end of the **conf** file.

```xml
<IfModule mod_status.c>
<Location /server-status>
ProxyPass !
SetHandler server-status
Require all granted
</Location>
</IfModule>
```

Step 7: Verify the configuration for syntax errors. If the output shows **Syntax OK,** the configuration is correct.

For httpd

```text
httpd -t
```

Step 8: Finally, **Restart Apache server**.

After restarting the Apache server, Enter the following URL in your browser. This will display information about the apache server.

To check in browser:

```xml
<Apache_Protocol>://<Apache_IP>:<Apache_Port>/server-status?auto
Example: http://127.0.0.1:80/server-status?auto
```

To check using curl:

```bash
curl <Apache_Protocol>://<Apache_IP>:<Apache_Port>/server-status?auto
Example: curl http://127.0.0.1:80/server-status?auto
```

<Apache_Protocol>://<Apache_IP>:<Apache_Port>/server-status?auto

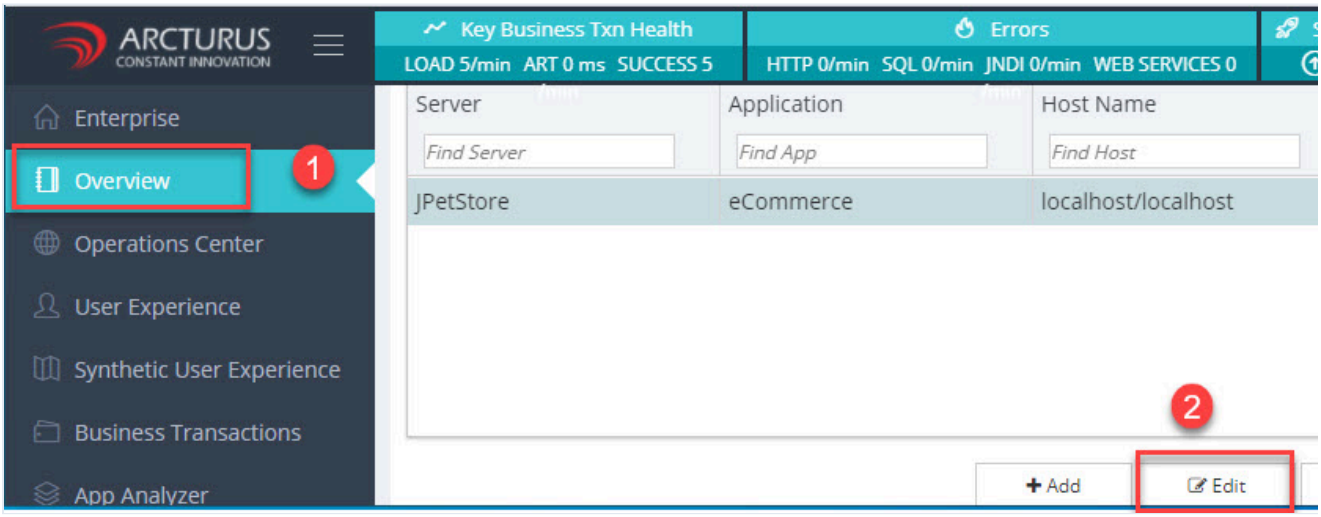
<Apache_Protocol> - http (or) https **<Apache_IP> -** Apache server running IP **<Apache_Port> -** Apache server running port

```yaml
ServerMPM: event
Server Built: Jan 27 2021 07:22:47
CurrentTime: Monday, 28-Jun-2021 18:11:40 EDT
RestartTime: Monday, 28-Jun-2021 18:08:24 EDT
ParentServerConfigGeneration: 1
ParentServerMPMGeneration: 0
ServerUptimeSeconds: 195
ServerUptime: 3 minutes 15 seconds
```

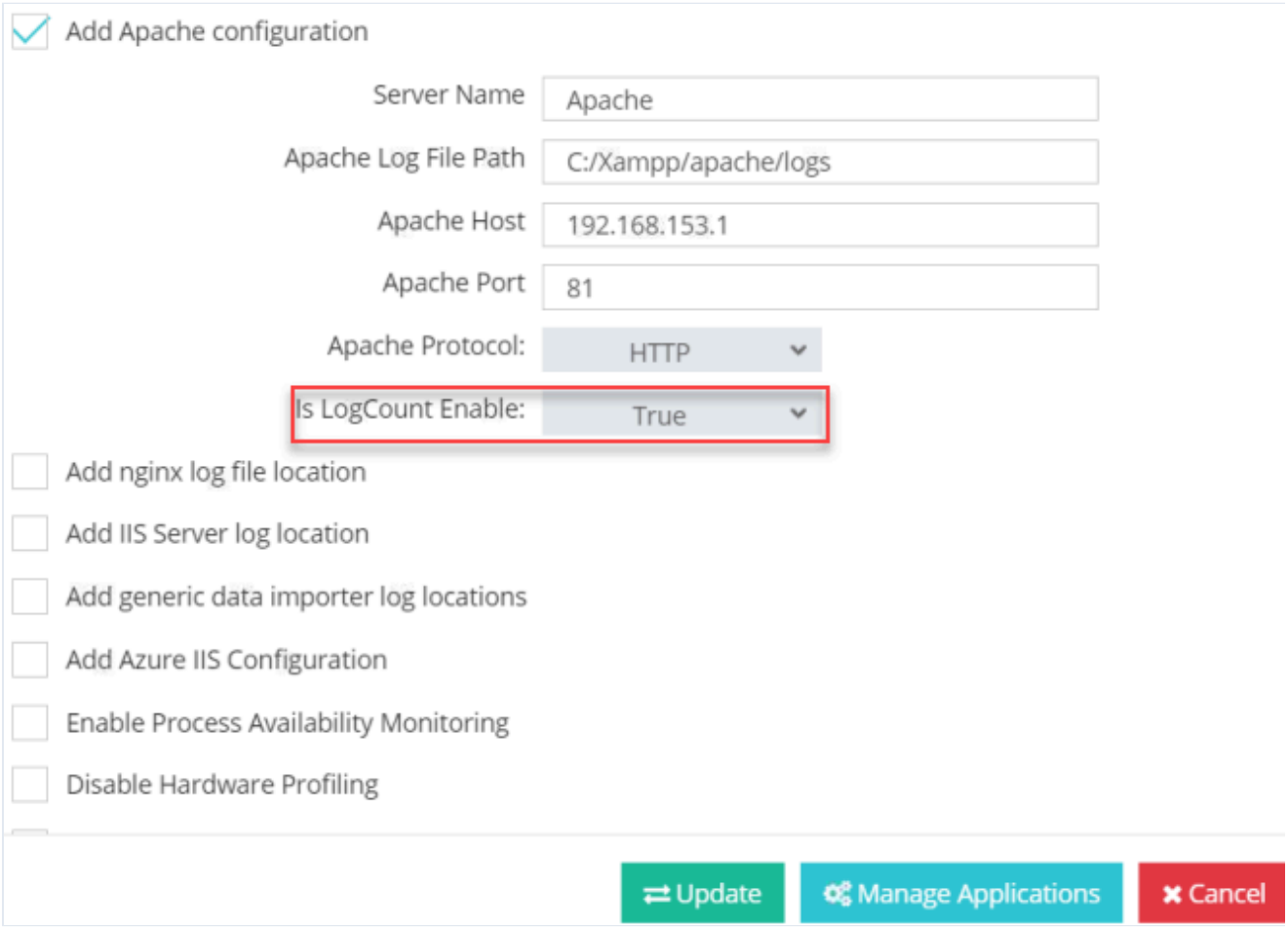
Step 9: Start the agent by executing the **startAgent.bat** in the **<Agent_Home_Directory>**.

Step 10: After the agent starts, It will be automatically added to the Applicare controller. Log in to the **Applicare console** and navigate to the **Overview** menu.

Step 11: Choose the newly added agent and click the **Edit** button.



Step 12: Check the "**Add Apache Configuration**" box and Set the configure the setting as shown as below. If the Apache is running over **HTTPS,** select **HTTPS** from the **Apache Protocol dropdown**. Click the **update** button to save the change.



1. Use the following command to check the configuration and virtual host details of Apache server.

httpd -S and for version **httpd -V**

2. If the new log configuration isn't taking effect, check whether a custom configuration is being referenced in the default configuration. Custom configuration may be included using the "**IncludeOptional**" keyword.

3. If the customer is wants to log only error logs to their existing log files. Add the following syntax to the end of their CustomLog configuration.

```
TEXT
"expr=%{REQUEST_STATUS} > 200"
```

e.g.

CustomLog "logs/access.log" combined **"expr=%{REQUEST_STATUS} > 200"**

```
#
# If you prefer a logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#

CustomLog "logs/access.log" combined "expr=%{REQUEST_STATUS} > 200"
```

Docs home