Vultr Docs

Search articles

How to Install Icinga 2 and Icinga Web 2 on Ubuntu 16.04

Modified on: Fri, Mar 31, 2017 at 11:24 am EST

Linux Guides (/docs/category/linux-guides/) **Server Apps (/docs/category/server-apps/)**

System Admin (/docs/category/system-admin/) **Ubuntu (/docs/category/ubuntu/)**

Icinga 2 is a widely used open source network resource monitoring system, and Icinga Web 2 is the official web interface for Icinga 2.

In this tutorial, I will explain how to install both of them on an Ubuntu 16.04 server.

Prerequisites

- A newly deployed Vultr Ubuntu 16.04 server instance.
- A sudo user. To learn more about creating a sudo user on Ubuntu, see instructions on Debian in another Vultr tutorial (https://www.vultr.com/docs/how-to-use-sudo-on-debian-centos-and-freebsd).

Step 1: Update the system

Log in from an SSH terminal as a sudo user, and then update the system to the latest stable status using the following commands:

```
sudo apt-get update -y
sudo apt-get upgrade -y
sudo shutdown -r now
```

After the reboot, use the same sudo user to log in.

Step 2: Install Apache

Install Apache using the following command:

```
sudo apt-get install apache2 -y
```

Delete the default Ubuntu Apache welcome page:

```
sudo rm /var/www/html/index.html
```

For security purposes, you should prohibit Apache from exposing files and directories within the web root directory `/var/www/html` to visitors:

```
sudo sed -i "s/Options Indexes FollowSymLinks/Options FollowSymLinks/" /etc/apache2/apache2.conf
```

Start the Apache service and get it started on boot:

```
sudo systemctl start apache2.service
sudo systemctl enable apache2.service
```

Step 3: Configure the UFW firewall

By default, the UFW firewall is disabled on a newly deployed Vultr Ubuntu 16.04 server instance. Use the following commands to enable the UFW firewall and to allow inbound traffic of SSH, HTTP, and HTTPS:

```
sudo ufw app list
sudo ufw allow OpenSSH
sudo ufw allow in "Apache Full"
sudo ufw enable
```

Step 4: Install MariaDB

4.1) Use the following command to install MariaDB:

```
sudo apt-get install mariadb-client mariadb-server -y
```

4.2) Start the MariaDB service:

```
sudo systemctl start mysql.service
sudo systemctl enable mysql.service
```

4.3) Secure the installation of MariaDB:

```
sudo /usr/bin/mysql_secure_installation
```

During the interactive process, answer questions one by one as below:

```
Enter current password for root (enter for none): Enter
Set root password? [Y/n]: Y
New password: <your-password>
Re-enter new password: <your-password>
Remove anonymous users? [Y/n]: Y
Disallow root login remotely? [Y/n]: Y
Remove test database and access to it? [Y/n]: Y
Reload privilege tables now? [Y/n]: Y
```

Note: Replace `<your-password>` with your own MySQL root password.

4.4) Modify the authentication plugin of MySQL `root` user:

```
sudo mysql -u root -p
```

Use the MariaDB root password you set earlier to log in.

In the MySQL shell:

```
UPDATE mysql.user SET authentication_string=PASSWORD('<your-password>'), plugin='mysql_native_pa
FLUSH PRIVILEGES;
EXIT;
```

Note: Replace `<your-password>` with your own MySQL root password.

Step 5: Install PHP

Install PHP 7.0 and several extensions for Icinga 2 and Icinga Web 2:

```
sudo apt-get install php7.0 libapache2-mod-php7.0 php7.0-gd php7.0-intl php7.0-xml php7.0-ldap p
```

Install the current version of Composer:

```
cd
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('SHA384', 'composer-setup.php') === 'e115a8dc7871f15d853148a7fbac7da27d6c0
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

Note: The above commands may be out of date in the future, so you should always get the latest version from the Composer official website (https://getcomposer.org/download/).

As a matter of convenience, move the Composer script `composer.phar` to `/usr/local/bin` and rename it `composer` :

```
sudo mv ~/composer.phar /usr/local/bin/composer
```

Install zip and unzip:

```
sudo apt-get install zip unzip -y
```

Install the ZendFramework Db component using Composer:

```
composer require zendframework/zend-db
```

Then you need to setup the proper timezone for your machine, which can be determined from the PHP official website (https://secure.php.net/manual/en/timezones.php). For example, if your server instance resides in the Vultr Los Angeles datacenter, then the timezone value for it is `America/Los_Angeles` .

Open the PHP configuration file with the vi editor:

```
sudo vi /etc/php/7.0/apache2/php.ini
```

Find the line:

```
;date.timezone =
```

Change it to:

```
date.timezone = America/Los_Angeles
```

Save and quit:

```
:wq!
```

Restart the Apache service in order to put new settings into effect:

```
sudo systemctl restart apache2.service
```

Step 6: Install Icinga 2 and its plugins

Setup the Icinga APT repo:

```
cd
wget -O - http://packages.icinga.org/icinga.key | sudo apt-key add -
sudo add-apt-repository 'deb http://packages.icinga.org/ubuntu icinga-xenial main'
sudo apt-get update
```

Install Icinga 2 and several plugins using the Icinga APT repo:

```
sudo apt-get install icinga2 nagios-plugins -y
```

To learn more about Icinga 2 plugins, please visit the Monitoring Plugins Project (https://www.monitoring-plugins.org/) website.

Start the Icinga 2 service:

```
sudo systemctl start icinga2.service
sudo systemctl enable icinga2.service
```

By default, the Icinga 2 program will enable three features: checker, mainlog, and notification. You can confirm that using the following command:

```
sudo icinga2 feature list
```

Step 7: Setup the Icinga 2 IDO modules

7.1) Install the IDO (Icinga Data Output) modules for MySQL

```
sudo apt-get install icinga2-ido-mysql
```

In the `Configuring icinga2-ido-mysql` wizard, when being asked whether you want to enable Icinga 2's ido-mysql feature, choose `<No>`. We will manually enable this feature later.

When being asked whether you want to configure a database for icinga2-ido-mysql, choose `<No>`. Instead, you can manually create a database as explained in step 7.2.

7.2) Create a database for Icinga 2

Log into the MySQL shell as root:

```
sudo mysql -u root -p
```

Use the MariaDB root password you set in step 4 to log in.

In the MySQL shell, create a database named `icinga` and a database user named `icinga` with the password `icinga`, and then grant privileges on this database to this database user.

```
CREATE DATABASE icinga;
GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icinga.* TO 'icinga'@
FLUSH PRIVILEGES;
EXIT;
```

7.3) Import the Icinga 2 IDO schema

```
sudo mysql -u root -p icinga < /usr/share/icinga2-ido-mysql/schema/mysql.sql
```

When prompted, input the MariaDB root password to finish the job.

7.4) Enable the IDO MySQL module

```
sudo vi /etc/icinga2/features-available/ido-mysql.conf
```

Find these lines:

```
user = "icinga2",
password = "",
host = "localhost",
database = "icinga2"
```

Modify them as below:

```
user = "icinga"
password = "icinga"
host = "localhost"
database = "icinga"
```

Save and quit:

```
:wq!
```

Enable the ido-mysql feature:

```
sudo icinga2 feature enable ido-mysql
sudo systemctl restart icinga2.service
```

Step 8: Install Icinga Web 2

8.1) Setup external command pipe

```
sudo icinga2 feature enable command
sudo systemctl restart icinga2.service
sudo icinga2 feature list
```

Before you can send commands to Icinga 2 using a web interface, you need to add the `www-data` user to the `icingacmd` group:

```
sudo groupadd icingacmd
sudo usermod -a -G icingacmd www-data
```

Use the following command to confirm your setup:

```
id www-data
```

8.2) Install Icinga Web 2 packages

```
sudo apt-get install icingaweb2 icingaweb2-module-monitoring icingaweb2-module-doc icingacli -y
```

Point the Apache web root directory to a location specified by Icinga Web 2:

```
sudo icingacli setup config webserver apache --document-root /usr/share/icingaweb2/public
sudo systemctl restart apache2.service
```

Load the Icinga Web 2 database schema:

```
mysql -p icingaweb2 < /usr/share/doc/icingaweb2/schema/mysql.schema.sql
```

8.3) Generate a setup token for later use in the Icinga Web 2 web installation wizard

```
sudo icingacli setup token create
```

8.4) Initiate the Icinga 2 installation wizard in the web interface

Point your web browser to the following URL:

```
http://<your-serve-ip>/icingaweb2/setup
```

8.5) On the Welcome page, input the setup token you generated earlier, and then click the `Next` button.

8.6) On the Modules page, select one or more modules you want to enable (at least, the `Monitoring` module is required), and then click the `Next` button.

8.7) On the Requirements page, make sure that every required item is satisfied, and then click the `Next` button.

8.8) On the Authentication page, you need to choose the authentication method when accessing Icinga Web 2. Here, you can choose `Database`, and then click the `Next` button.

8.9) On the Database Resource page, fill out all required fields as below, and then click the `Next` button.

- Resource Name*: icingaweb_db
- Database Type*: MySQL
- Host*: localhost
- Database Name*: icingaweb2
- Username*: root
- Password*: <MariaDB-root-password>

8.10) On the Authentication Backend page, using the default backend name `icingaweb2`, click the `Next` button to move on.

8.11) On the Administration page, setup the first Icinga Web 2 administrative account (say it is `icingaweb2admin`) and password (say it is `icingaweb2pass`), and then click the `Next` button.

8.12) On the Application Configuration page, you can adjust application- and logging-related configuration options to fit your needs. For now, you can use the default values listed below and click the `Next` button to proceed.

- Show Stacktraces: Checked
- User Preference Storage Type*: Database
- Logging Type*: Syslog
- Logging Level*: Error
- Application Prefix*: icingaweb2

8.13) On the Review page, double check your configuration, and then click the `Next` button.

8.14) On the Monitoring Module Configuration Welcome page, click the `Next` button.

8.15) On the Monitoring Backend page, use the default backend name `icinga` and backend type `IDO`, and then click the `Next` button.

8.16) On the Monitoring IDO Resource page, input IDO database details you setup earlier, and then click the `Next` button.

- Resource Name*: icinga_ido
- Database Type*: MySQL
- Host*: localhost
- Database Name*: icinga

- Username*: icinga
- Password*: icinga

8.17) On the Command Transport page, still use these default values listed below. Click the `Next` button to move on.

- Transport Name*: icinga2
- Transport Type*: Local Command File
- Command File*: /var/run/icinga2/cmd/icinga2.cmd

8.18) On the Monitoring Security page, still use the default value:

- Protected Custom Variables: *pw*,*pass*,community

Click the `Next` button to go to next page.

8.19) On the review page, double check your configuration, and then click the `Finish` button.

8.20) On the Congratulations! page, click the `Login to Icinga Web 2` button to jump to the Icinga Web 2 login page. Use the Icinga Web 2 administrative account and password you setup earlier to log in. Feel free to explore the Icinga Web 2 dashboard.

That concludes our tutorial. Thank you for reading.