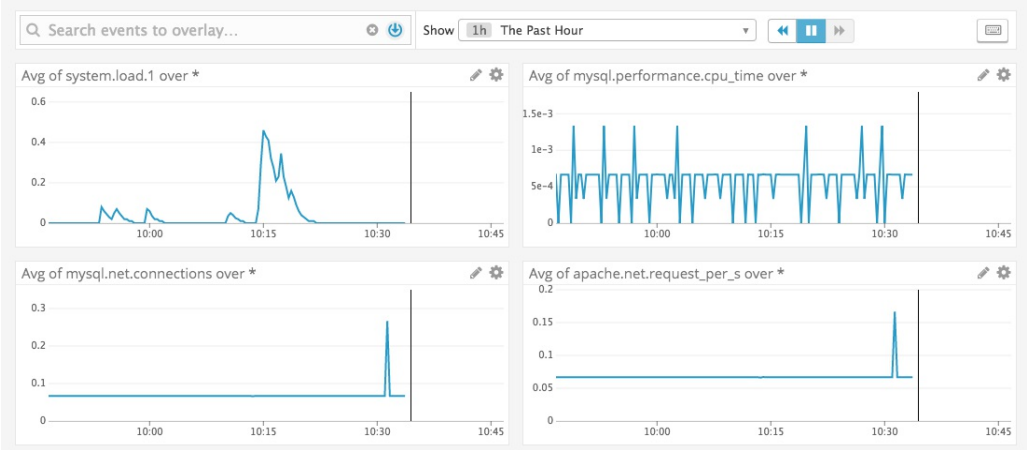
Monitoring the LAMP stack with Datadog

The LAMP stack is a set of open-source software used for creating websites and web applications. Lamp is named for its four original components — Linux, Apache, MySQL, and PHP — and, though evolving to include alternatives like Python and Perl, it has retained its open-source nature. LAMP's cost-effective and flexible approach to web infrastructure makes it the most popular solution stack for hosting websites and web apps.

By providing real-time collection and vizualization of key metrics across each layer of the LAMP stack, Datadog offers a flexible and customizeable way to monitor your web app's performance and usage.



A custom dashboard created with Datadog

This post will walk you through the integration and configuration of each LAMP stack layer. You'll learn how to:

- Install the Datadog agent
- Integrate each layer of your LAMP stack with Datadog
- Configure Datadog to collect the desired metrics
- Customize, organize, and optimize your monitoring experience

Installing the Datadog Agent

After you've created your Datadog account (or signed up for a 14-day free trial), you'll recieve an application key that you'll use to install the Datadog agent. You can find the key in your account .

Run the following script to install the agent on your machine:

```
DD_API_KEY=<YOUR_API_KEY> bash -c "$(curl -L https://raw.githubusercontent.com/DataDog/datadog-agent/master/cmd/agent/install_script.sh)"
```

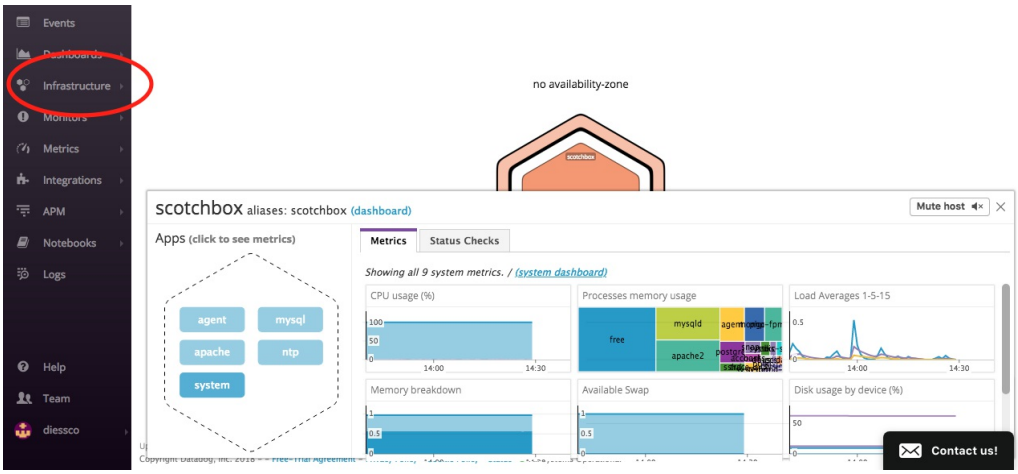
To check the success of the installation, run:

```
sudo datadog-agent status
```

If the installation was successful, you should see information about the running agent:

```
Getting the status from the agent. ============== Agent (v6.2.0) ==============
Status date: 2018-05-16 19:45:31.773123 UTC Pid: 26471 Python Version: 2.7.12 Logs:
Check Runners: 1 Log Level: info Paths ===== Config File: /etc/datadog-agent/datadog.yaml conf.d: /etc/datadog-agent/conf.d checks.d: /etc/datadog-agent/checks.d [...]
```

After that, you're ready to view your host's metrics in real-time via the Datadog dashboard: go to Infrastructure, HostMap, and click on the system panel that appears after your host has been selected.



the host's system metrics:
Infrastructure->HostMap->your Host->System

Now that you've installed the Datadog Agent on your host, you can begin integrating the rest of your lamp stack. The remainder of the article will walk you through the steps necessary to

use the agent to monitor each software component in LAMP.

(Check out the documentation to see a full list of commands for the Datadog agent.)

Monitoring the Apache web server

Integration

Select the Apache tile in the Integrations section of the Datadog website. Following the directions in the tile, you may need to install mod-status on your Apache server.

Apache's status Module

Your server's status page should be available at:
`http://yourpage@example.com/server-status`
If this link throws a forbidden access error, then you'll need to enable mod-status manually.

Configuration

To configure Apache to report metrics to the Datadog agent, you'll need to navigate to your a apache.d/conf.yaml.example file. Create a conf.yaml file with the following code:
`sudo cp conf.yaml.example conf.yaml`

Edit the new conf.yaml file to match the following configuration:

```
init_config:

instances:
- apache_status_url: http://example.com/server-status?auto
# apache_user: example_user
# apache_password: example_password
tags:
- instance:foo
```

When the agent runs, it will look for a conf.yaml file, and begin pulling metrics from Apache when it finds it.

To check that the agent is collecting data from Apache run the following two commands to: restart the agent:
`sudo service datadog-agent restart`

check that Apache has been integrated successfully

```
sudo service datadog-agent restart
```

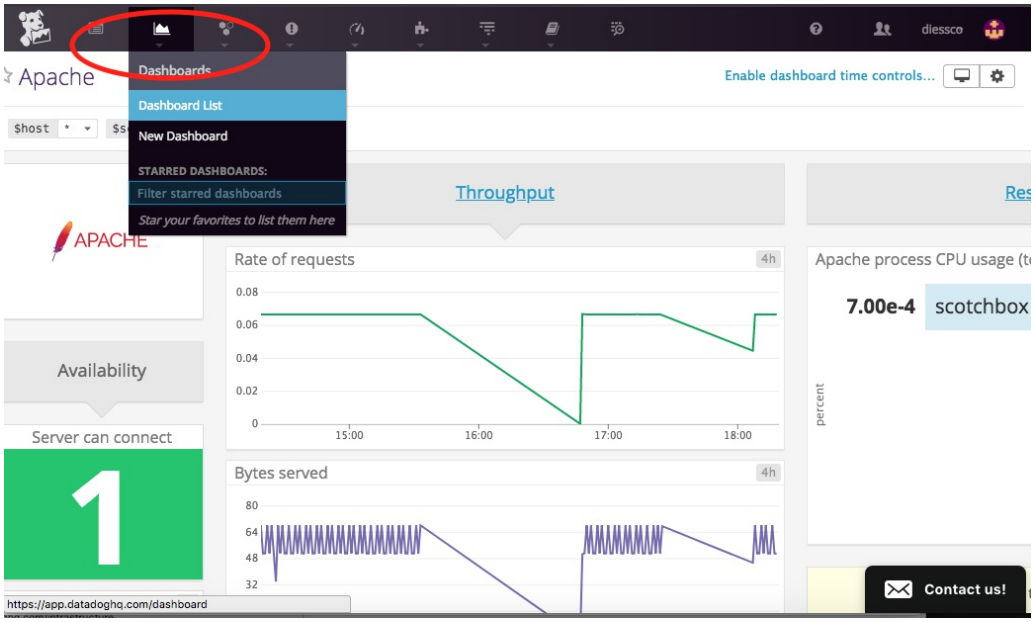
If Apache has been integrated succesfully, the output should contain a section similar to this:

```
Checks ======
[...]


apache
------
- instance #0 [OK] -
- Collected 8 metrics & 0 events
```

Monitoring

Now that you have your Apache server integrated and configured properly, visit your Apache dashboard on Datadog to view your server's metrics in real time.



The Apache Dashboard:
Dashboards -> DasboardList -> Apache

Monitoring the MySQL Database

Integration

Select the MySQL tile in the Integrations

section of the Datadog website. Following the directions on the tile, use the following commands to:
create a Datadog user on your MySQL server

```
sudo mysql -e "CREATE USER 'datadog'@'localhost' IDENTIFIED BY 'yourpassword';"
```

grant that user replication rights:

```
sudo mysql -e "GRANT REPLICATION CLIENT ON *.* TO 'datadog'@'localhost' WITH MAX_USER_CONNECTIONS 5;"
```

grant all additional permissions in order to obtain the full metrics catalog:

```
sudo mysql -e "GRANT PROCESS ON *.* TO 'datadog'@'localhost';"
sudo mysql -e "GRANT SELECT ON performance_schema.* TO 'datadog'@'localhost';"
```

Use the following commands to check if:

a Datadog user with replication rights has been created on the MySQL server

```
mysql -u datadog --password='TFs1v1AEY9>kv84btq29n6Yy' -e "show status" | \ grep Uptime && echo -e "\033[0;32mMySQL user - OK\033[0m" || \ echo -e "\033[0;31mCannot connect to MySQL\033[0m" mysql -u datadog --password='TFs1v1AEY9>kv84btq29n6Yy' -e "show slave status" && \ echo -e "\033[0;32mMySQL grant - OK\033[0m" || \ echo -e "\033[0;31mMissing REPLICATION CLIENT grant\033[0m"
```

Any additional priveleges have been granted

```
mysql -u datadog --password='TFs1v1AEY9>kv84btq29n6Yy' -e "SELECT * FROM performance_schema.threads" && \ echo -e "\033[0;32mMySQL SELECT grant - OK\033[0m" || \ echo -e "\033[0;31mMissing SELECT grant\033[0m" mysql -u datadog --password='TFs1v1AEY9>kv84btq29n6Yy' -e "SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST" && \ echo -e "\033[0;32mMySQL PROCESS grant - OK\033[0m" || \ echo -e "\033[0;31mMissing PROCESS grant\033[0m"
```

Configuration

Just like Apache, to configure mySQL to report metrics to the Datadog agent, you'll need to navigate to your a mysql.d/conf.yaml.example file. Create a conf.yaml file with the following code:

```
sudo cp conf.yaml.example conf.yaml
```

Edit the conf.yaml file to match the following configuration:

```
 init_config: instances: - server: localhost user: datadog pass:
TFs1v1AEY9>kv84btq29n6Yy tags: - optional_tag1 - optional_tag2 options: replication: 0
galera_cluster: 1
```

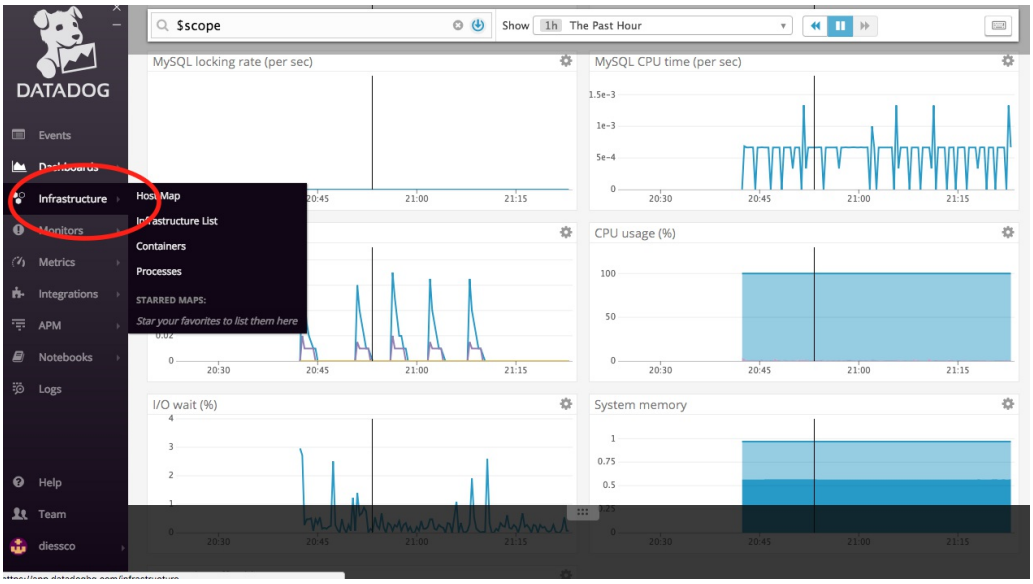
When the agent runs, it will look for a conf.yaml file, and begin pulling metrics from MySQL.

To check that the agent is collecting data from MySQL run the following two commands to:
restart the agent:
`sudo service datadog-agent restart`
check that mySQL has been integrated successfully
`sudo service datadog-agent restart`
If MySQL has been integrated succesfully, the output should contain a section similar to this:
`Checks ======`
`[...]`

```
mySQL
------
- instance #0 [OK] -
- Collected 8 metrics & 0 events
```

Monitoring

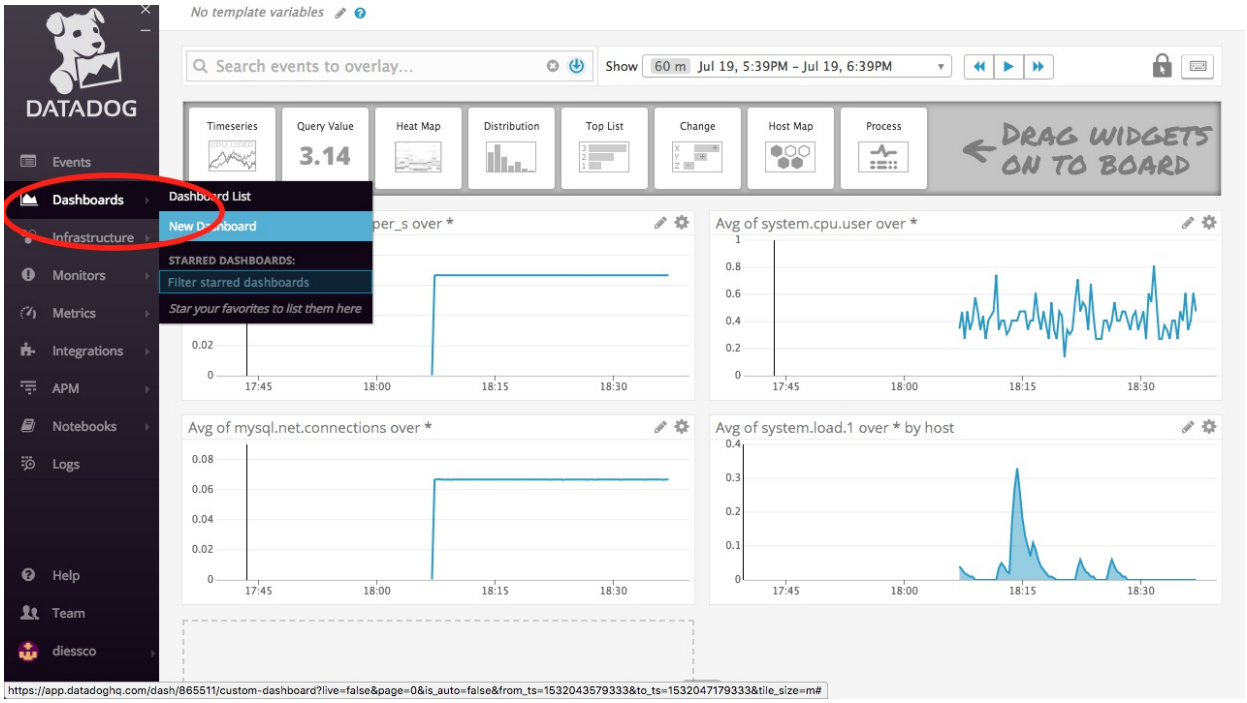
Now that you have your MySQL server integrated and configured properly, visit your MySQL dashboard on Datadog to view your server's metrics in real time.



The MySQL Dashboard:
Dashboards -> DasboardList -> MySQL

Custom Dashboards

Datadog dashboards allow you to gather customized vizualizations for any obtained metrics in one place. To create a custom dashboard, click on the "new dashboard" option in the Dashboards pane. Click on "add graphs" at the top of the page to drag and drop real-time vizualized metrics from all your integrations. For a full list of dashboard features, visit the Dashboarding page.



A Custom Dashboard:
Dashboards -> New Dashboard

Full LAMP Observability with Datadog

The LAMP solution stack is the most popular approach for hosting web apps and websites. We've shown you just a few ways to collect and visualize key metrics from each layer of your LAMP stack.

Datadog integrates with more than 250 other tools and services to provide a comprehensive, customizeable, and intuitive monitoring experience. Start monitoring your LAMP stack today with a free trial.