

CHAPTER 1:

1.1 What is Datadog?

Datadog is a monitoring service for cloud-scale applications, providing monitoring of servers, databases, tools, and services, through a SaaS-based data analytics platform.

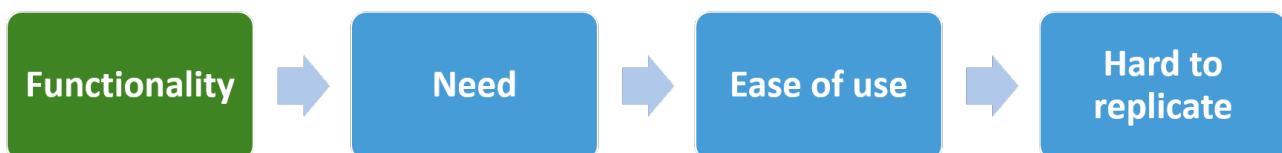
Datadog was designed as a cloud infrastructure monitoring service, with a dashboard, alerting, and visualizations of metrics.

Datadog was listed in Forbes' Cloud 100 and was ranked in the top ten fastest growing companies in North America in Deloitte's 2016 Fast 500 List.

Datadog was founded in 2010 by Olivier Pomel and Alexis Lê-Quôc.

1.2 Why Datadog?

- Providing functionality
- We need
- In an easy-to-use manner
- That would be difficult to build and maintain ourselves



- **The agent:** It gathers system metrics, integrates with key software we use, and provides a standard interface to which our applications can send custom metrics. The agent is deployable via a chef cookbook datadog wrote for us. It requires minimal configuration. It knows which system and application metrics are worth gathering.
- **Integrations:** Datadog has prebuilt integrations to pull data from almost every important service we use. Integrating with all the data sources is literally a few clicks.
- **Events:** Through the integrations datadog generates a consolidated event stream that we can filter and search as needed. The interface makes searching and filtering events straightforward.
- **Dashboards:** Datadog lets us build dashboards that combine metrics from many different sources. We can combine and transform metrics to make them more useful. It also provides a powerful interface for interactive exploration of metrics.

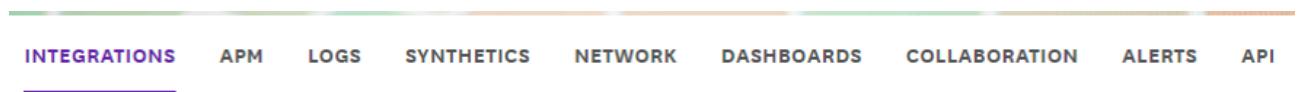
There are prebuilt dashboards for lots of things we care about. Snazzy features like autocomplete and templating make building our own dashboards easy.

- **Alerting:** Datadog has nice stream processing capabilities for generating alerts, and it can surface them in services we use like pagerduty and slack.

1.3 Datadog High-level Architecture

1. Datadog Backend: Backend is built using a number of open and closed source technologies including D3, Apache Cassandra, Kafka, PostgreSQL, etc.
2. Datadog Agent: Datadog uses a Go based agent, rewritten from scratch since its major version 6.0.0 released on February 28, 2018. It was formerly Python based.

1.4 Features



More than 450 built-in integrations

See across all your systems, apps, and services



<https://www.datadoghq.com/product/platform/integrations/#all>

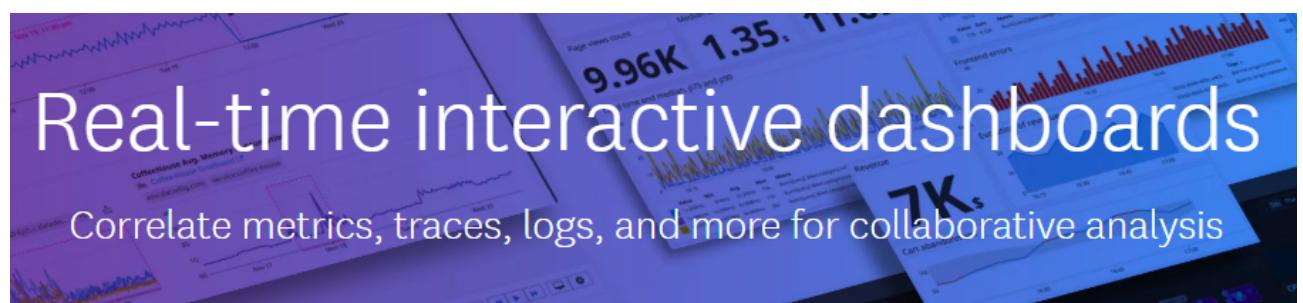


fig 1.1 Real-time Interactive Dashboards

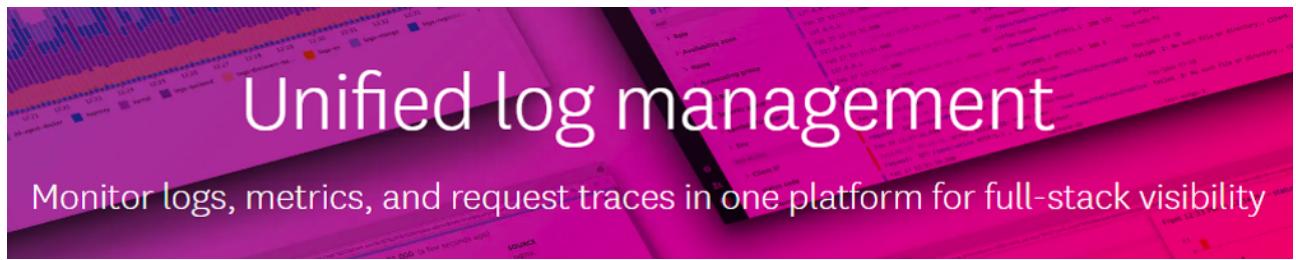


fig 1.2 Unified Log Management

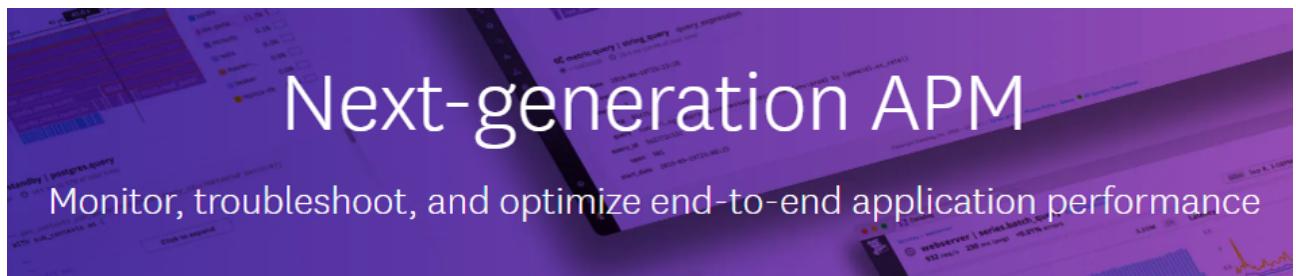


fig 1.3 Next-generation APM

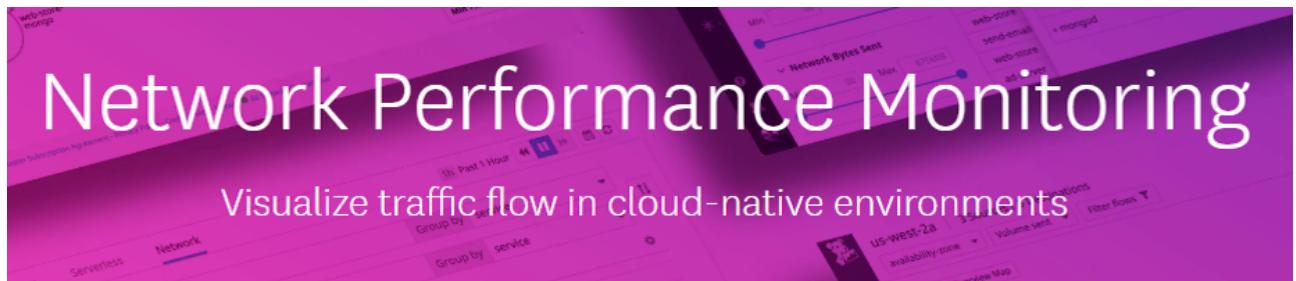


fig 1.4 Network Performance Monitoring



fig 1.5 Simplified Synthetic Monitoring



fig 1.6 Real User Monitoring

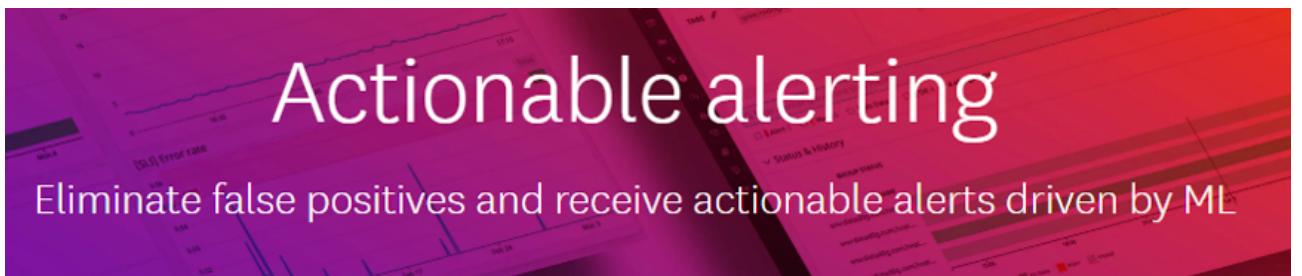


fig 1.7 Actionable alerting

CHAPTER 2:

2.1 Getting Started

Create a Datadog account. Free for 14 days

Use this link to open an account: <https://www.datadoghq.com/>

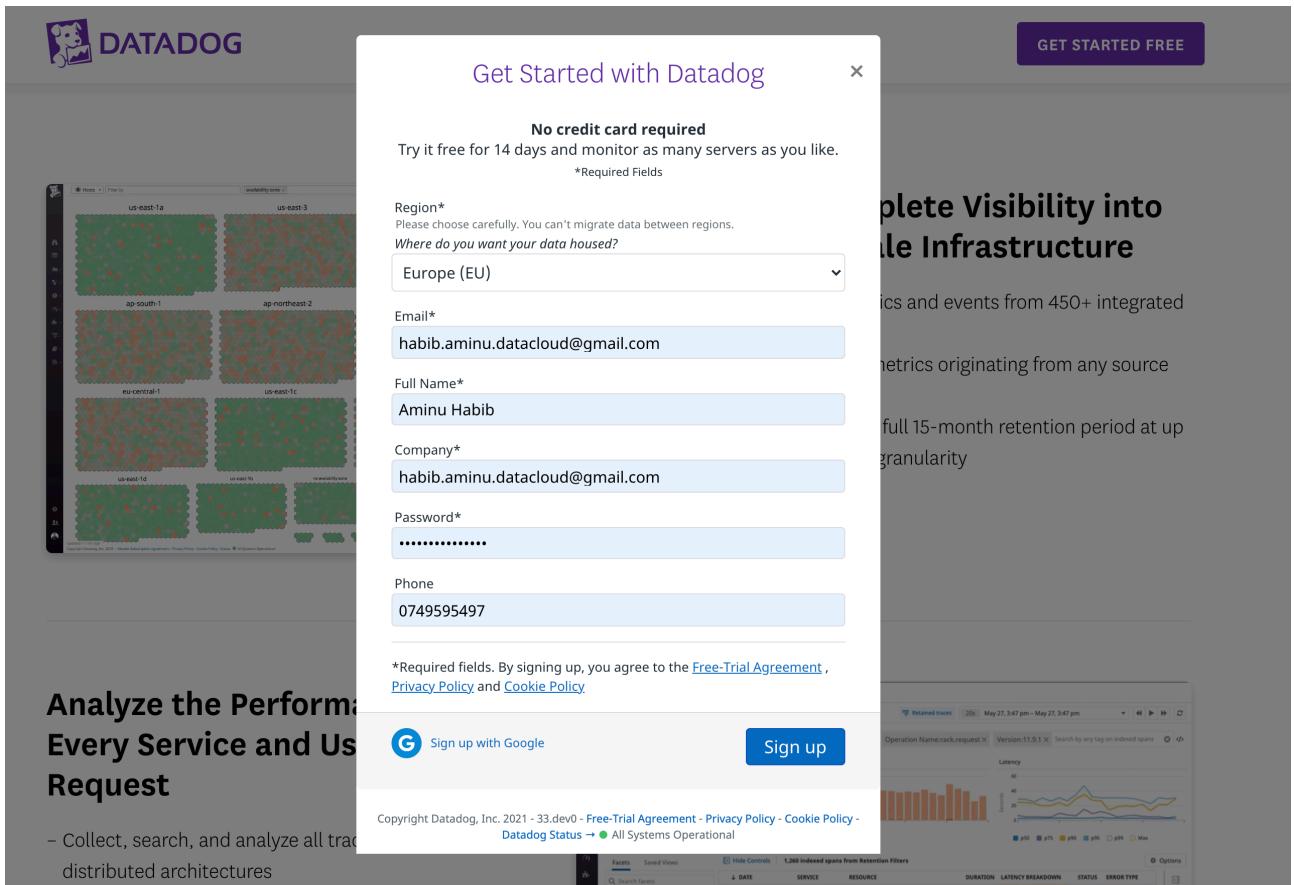


fig 2.1 Creating an account

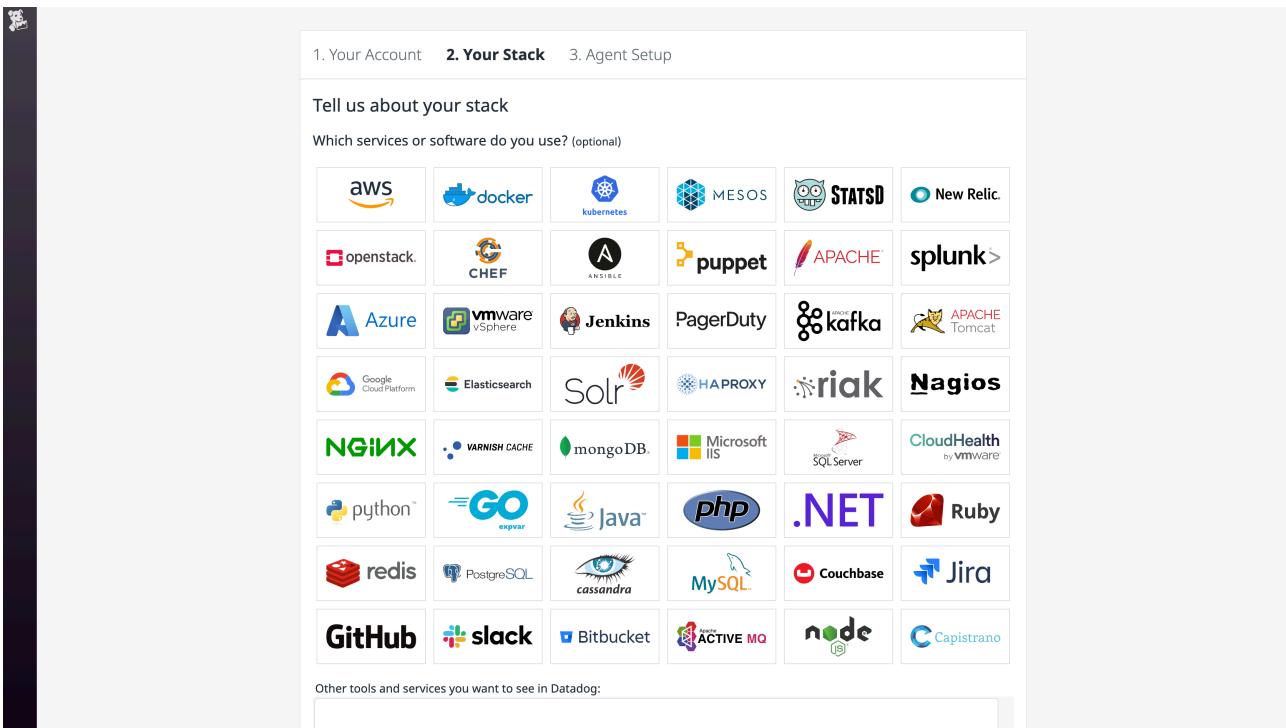


fig 2.2 Choose your stack

2.2 Datadog Terminology

- Agent
- DogStatsD
- StatsD
- Metrics

2.2.1 Datadog Agent

The Agent is lightweight software installed on your hosts. It reports metrics and events from your host to Datadog via....

- Integrations (<https://docs.datadoghq.com/integrations>)
- DogStatsD, or
- The API (<https://docs.datadoghq.com/api/>)

With additional setup, the Agent can report live processes, logs, and traces.

2.2.1.2 How Datadog Agent Works?

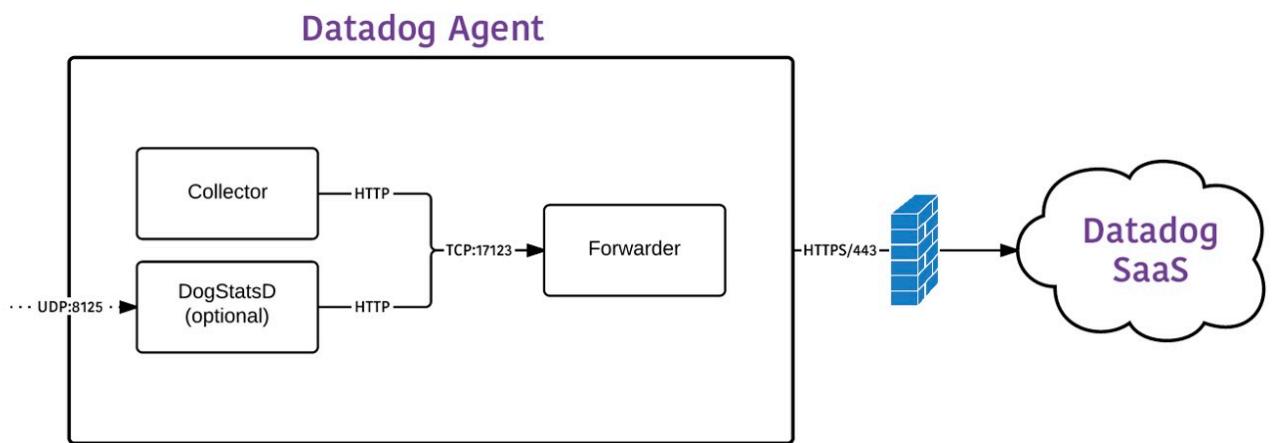


fig 2.3 Datadog Agent Architecture

2.2.1.3 Datadog Agent: Platforms

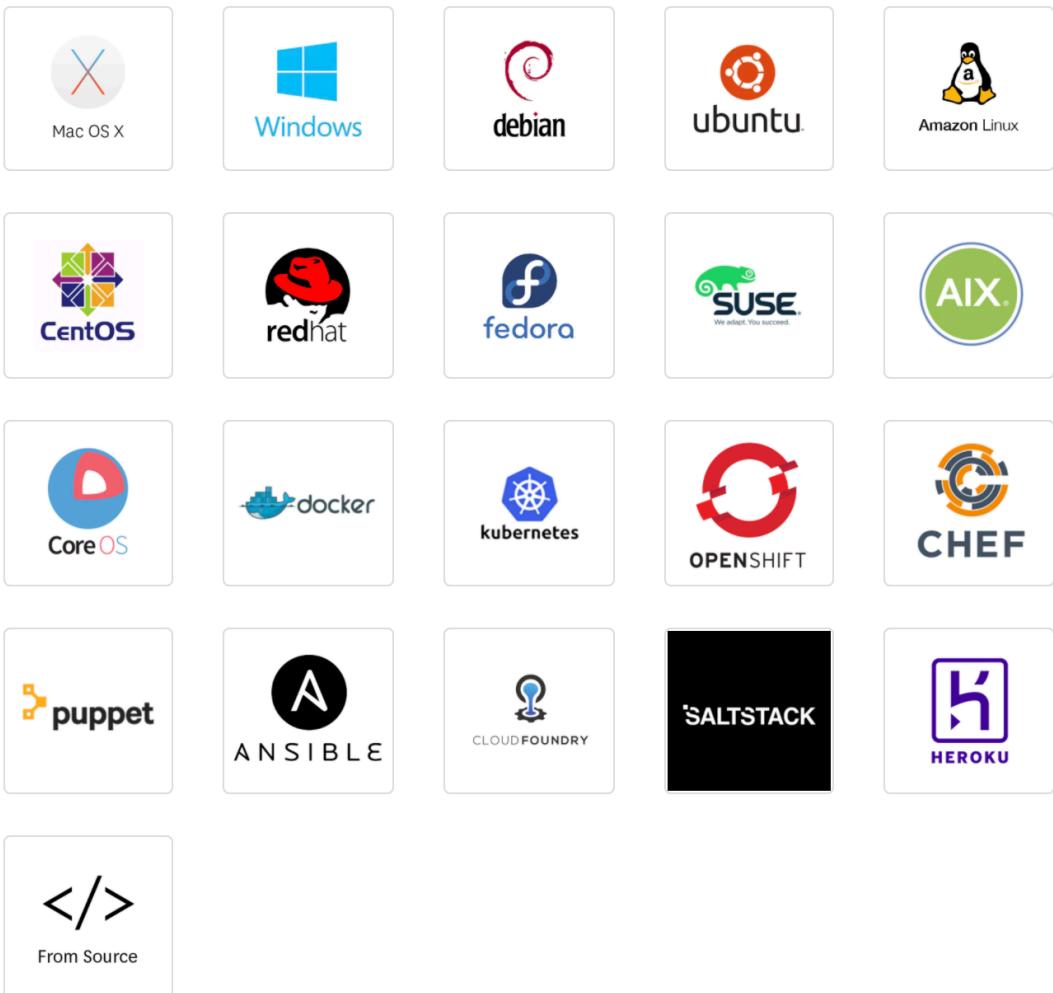


fig 2.4 Datadog Agent Platforms

2.2.2 Datadog Agent: DogStatsD

DogStatsD accepts all three of the major Datadog data types: metrics, events, and service checks.

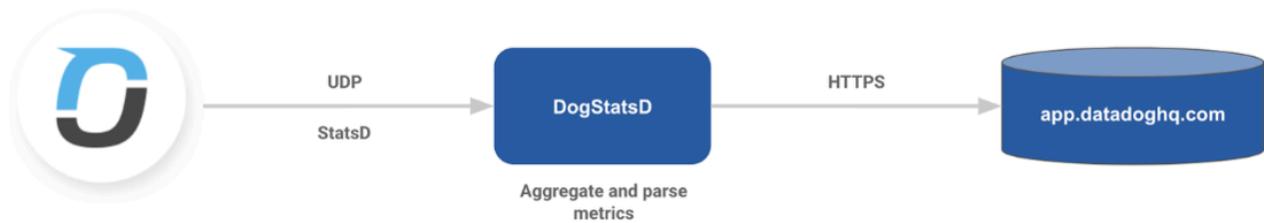


fig 2.5 DogStatsD Architecture

In v6, DogStatsD is a Golang implementation of Etsy's StatsD metric aggregation daemon. It is used to receive and roll up arbitrary metrics over UDP or Unix socket, thus allowing custom code to be instrumented without adding latency.

2.2.2.2 How does DogStatsD Work?

- DogStatsD accepts custom metrics, events, and service checks over UDP and periodically aggregates and forwards them to Datadog.
- Because it uses UDP, your application can send metrics to DogStatsD and resume its work without waiting for a response. If DogStatsD ever becomes unavailable, your application won't experience an interruption.
- As it receives data, DogStatsD aggregates multiple data points for each unique metric into a single data point over a period of time called the flush interval (ten seconds, by default).

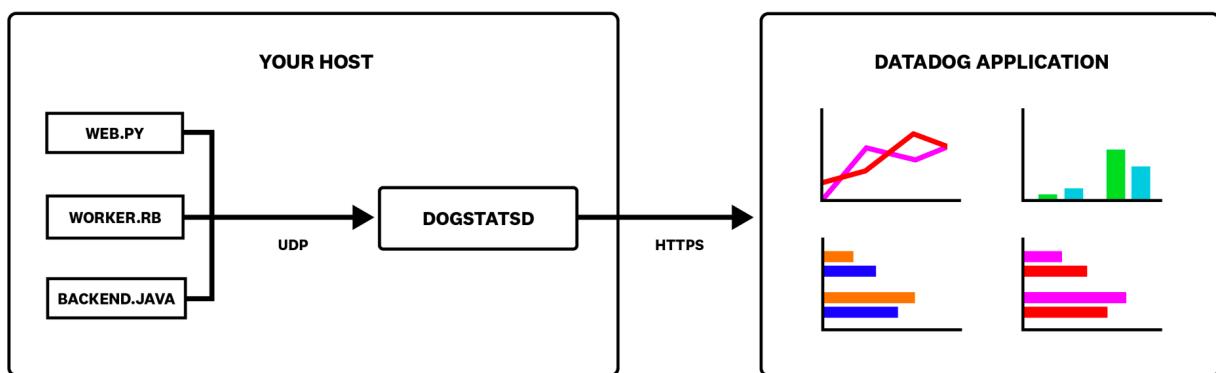


fig 2.6 DogStatsD Functional Model

2.2.2.3 Datadog Agent: The Collector

The collector gathers all standard metrics every 15 seconds. Agent v6 embeds a Python 2.7 interpreter to run integrations and custom checks.

2.2.2.4 Datadog Agent: The Forwarder

The Agent forwarder send metrics over HTTPS to Datadog. Buffering prevents network splits from affecting metric reporting.

Metrics are buffered in memory until a limit in size or number of outstanding send requests are reached.

Afterwards, the oldest metrics are discarded to keep the forwarder's memory footprint manageable. Logs are sent over an SSL-encrypted TCP connection to Datadog.

2.3. Agent v6 & v7

Agent v6 and v7 are composed of a main process responsible for collecting infrastructure metrics, logs, and receiving DogStatsD metrics. The main components to this process are:

- A. The Collector is in charge of running checks and collecting metrics.
- B. The Forwarder sends payloads to Datadog.

Two optional processes are spawned by the Agent if enabled in the `datadog.yaml` configuration file:

- C. The APM Agent is a process to collect traces (enabled by default).
- D. The Process Agent is a process to collect live process information. By default, it only collects available containers, otherwise it is disabled.

2.4 Datadog Agent: Agent GUI

You can configure the port on which the GUI runs in the `datadog.yaml` file. To disable the GUI, set the port's value to `-1`. For Windows and macOS, the GUI is enabled by default and runs on port 5002. For Linux, the GUI is disabled by default.

When the Agent is running, use the `datadog-agent launch-gui` command to open the GUI in your default web browser.

2.4.2 Datadog Agent: Agent GUI: Requirements

Cookies must be enabled in your browser. The GUI generates and saves a token in your browser which is used for authenticating all communications with the GUI server.

To start the GUI, the user must have the required permissions. If you are able to open `datadog.yaml`, you are able to use the GUI.

For security reasons, the GUI can only be accessed from the local network interface (`localhost/127.0.0.1`), therefore you must be on the same host that the Agent is running. That is, you can't run the Agent on a VM or a container and access it from the host machine.

2.6 Datadog Agent: CLI

With Agent v6+, the command line interface is based on subcommands. To run a subcommand, first invoke the Agent binary:

```
<AGENT_BIN_PATH> <SUB_COMMAND> <OPTIONS>
```

| SUBCOMMAND | NOTES |
|-------------|---|
| check | Run the specified check. |
| configcheck | Print all configurations loaded & resolved of a running Agent. |
| diagnose | Execute connectivity diagnosis on your system. |
| flare | Collect a flare and send it to Datadog. |
| health | Print the current Agent health. |
| help | Help about any command. |
| hostname | Print the hostname used by the Agent. |
| import | Import and convert configuration files from previous versions of the Agent. |

| | |
|-----------------|---|
| installservice | Install the Agent within the service control manager. |
| launch-gui | Start the Datadog Agent GUI. |
| regimport | Import the registry settings into datadog.yaml. |
| remove-service | Remove the Agent from the service control manager. |
| restart | Restart the Agent. |
| restart-service | Restart the Agent within the service control manager. |
| start | Start the Agent. |
| start-service | Start the Agent within the service control manager. |
| status | Print the current Agent status. |
| stop | Stop the Agent. |
| stopservice | Stop the Agent within the service control manager. |
| version | Print version info. |

2.8 Datadog Agent: `datadog.yaml`

Agent Configuration Files: The Agent v6 configuration file uses YAML to better support complex configurations, and to provide a consistent configuration experience, as Checks also use YAML configuration files.

https://github.com/DataDog/datadog-agent/blob/master/pkg/config/config_template.yaml

Agent configuration directory:

| PLATFORM | COMMAND |
|--------------------------------------|--|
| AIX | <code>/etc/datadog-agent/datadog.yaml</code> |
| Linux | <code>/etc/datadog-agent/datadog.yaml</code> |
| CentOS | <code>/etc/datadog-agent/datadog.yaml</code> |
| Debian | <code>/etc/datadog-agent/datadog.yaml</code> |
| Fedora | <code>/etc/datadog-agent/datadog.yaml</code> |
| macOS | <code>~/.datadog-agent/datadog.yaml</code> |
| RedHat | <code>/etc/datadog-agent/datadog.yaml</code> |
| Source | <code>/etc/datadog-agent/datadog.yaml</code> |
| Suse | <code>/etc/datadog-agent/datadog.yaml</code> |
| Ubuntu | <code>/etc/datadog-agent/datadog.yaml</code> |
| Windows Server 2008, Vista and newer | <code>\ProgramData\Datadog\datadog.yaml</code> |
| Windows Server 2003, XP or older | <i>unsupported platform</i> |

| PLATFORM | COMMAND |
|--------------------------------------|--|
| AIX | <code>/etc/datadog-agent/conf.d/</code> |
| Linux | <code>/etc/datadog-agent/conf.d/</code> |
| CentOS | <code>/etc/datadog-agent/conf.d/</code> |
| Debian | <code>/etc/datadog-agent/conf.d/</code> |
| Fedora | <code>/etc/datadog-agent/conf.d/</code> |
| macOS | <code>~/.datadog-agent/conf.d/</code> |
| RedHat | <code>/etc/datadog-agent/conf.d/</code> |
| Source | <code>/etc/datadog-agent/conf.d/</code> |
| Suse | <code>/etc/datadog-agent/conf.d/</code> |
| Ubuntu | <code>/etc/datadog-agent/conf.d/</code> |
| Windows Server 2008, Vista and newer | <code>\ProgramData\Datadog\conf.d</code> |
| Windows Server 2003, XP or older | <i>unsupported platform</i> |

CHAPTER 3:

3.1 Install your first Datadog Agent

The Datadog Agent collects metrics and events from your systems and apps. Install at least one Agent anywhere, even on your workstation.

Select your platform to see Agent 7 Installation Instructions.

Installs in seconds - 20 supported OS - 100% open-source

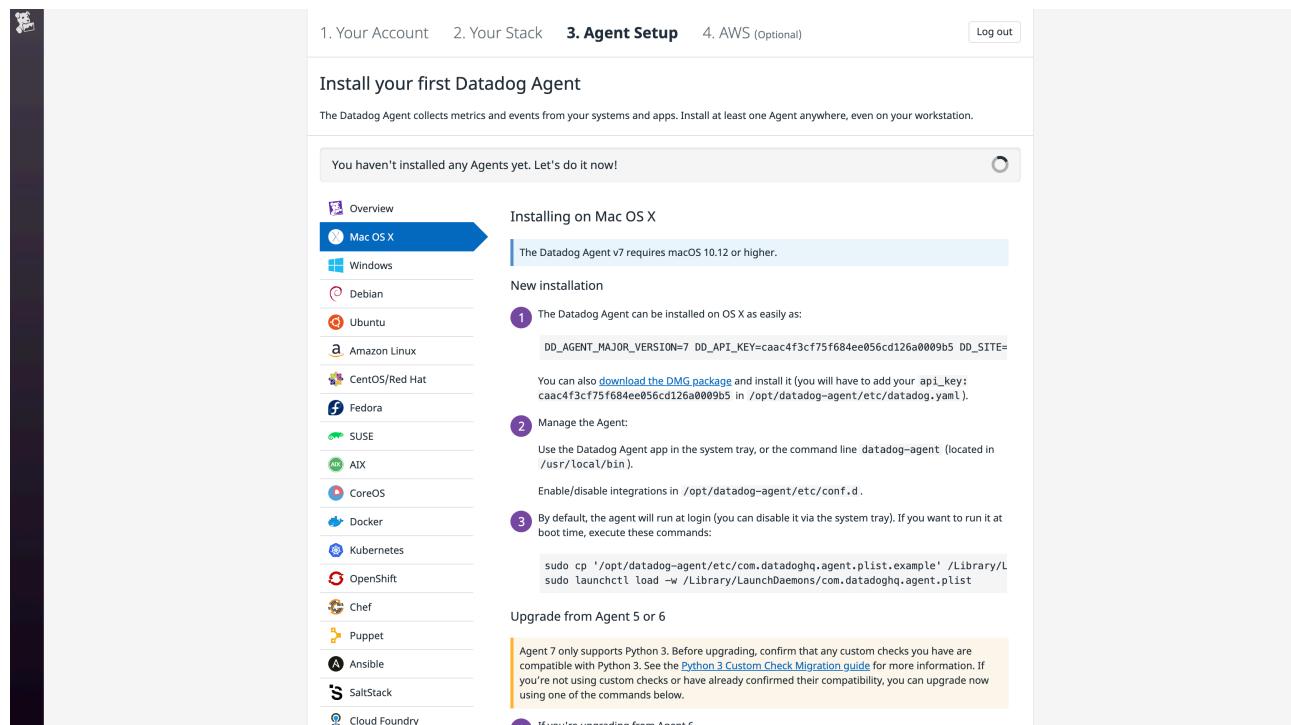


fig 3.1 Datadog agent installation

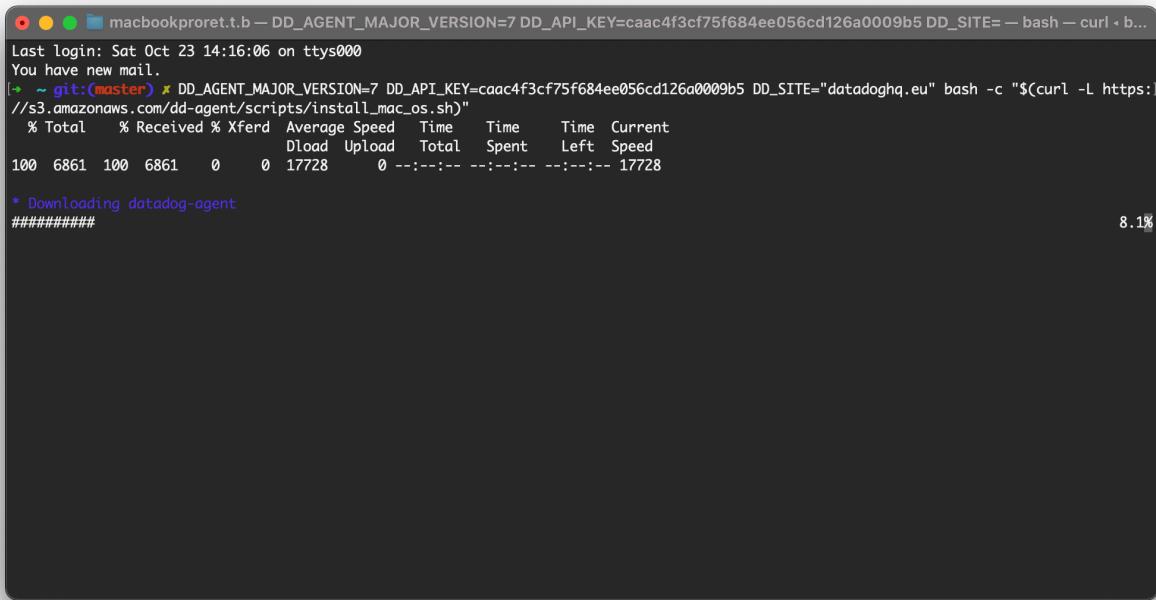
3.1.2 Installing on macOS Big Sur (v11.2.3)

The Datadog Agent v7 requires macOS 10.12 or higher.

3.1.2.2 New installation

1. The Datadog Agent can be installed on OS X as easily as:

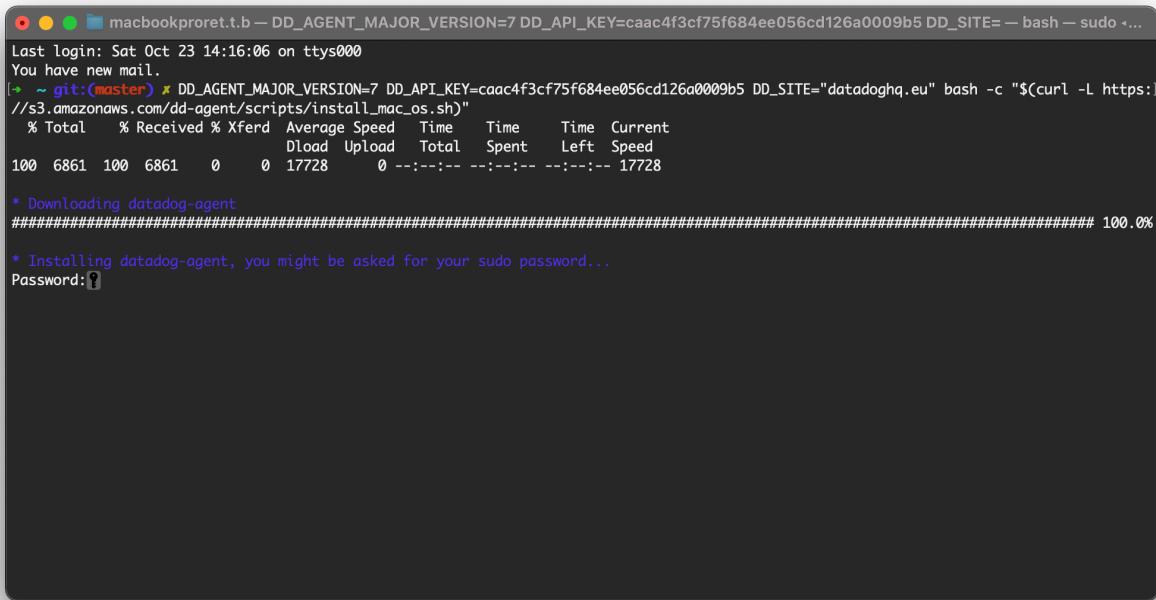
```
DD_AGENT_MAJOR_VERSION=7 DD_API_KEY=caac4f3cf75f684ee056cd126a0009b5
DD_SITE="datadoghq.eu" bash -c "$(curl -L https://s3.amazonaws.com/dd-agent/scripts/install_mac_os.sh)"
```



```
macbookproret.t.b - DD_AGENT_MAJOR_VERSION=7 DD_API_KEY=caac4f3cf75f684ee056cd126a0009b5 DD_SITE= -- bash -- curl -L https://s3.amazonaws.com/dd-agent/scripts/install_mac_os.sh"
Last login: Sat Oct 23 14:16:06 on ttys000
You have new mail.
[~ - git:(master) x DD_AGENT_MAJOR_VERSION=7 DD_API_KEY=caac4f3cf75f684ee056cd126a0009b5 DD_SITE="datadoghq.eu" bash -c "$(curl -L https://s3.amazonaws.com/dd-agent/scripts/install_mac_os.sh)"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100  6861  100  6861    0      0  17728      0 --:--:-- --:--:-- 17728
* Downloading datadog-agent
#####
8.1%
```

fig 3.2 Datadog Agent Installation on macOS

You might be asked for your password:



```
macbookproret.t.b - DD_AGENT_MAJOR_VERSION=7 DD_API_KEY=caac4f3cf75f684ee056cd126a0009b5 DD_SITE= -- bash -- sudo ...
Last login: Sat Oct 23 14:16:06 on ttys000
You have new mail.
[~ - git:(master) x DD_AGENT_MAJOR_VERSION=7 DD_API_KEY=caac4f3cf75f684ee056cd126a0009b5 DD_SITE="datadoghq.eu" bash -c "$(curl -L https://s3.amazonaws.com/dd-agent/scripts/install_mac_os.sh)"
% Total    % Received % Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100  6861  100  6861    0      0  17728      0 --:--:-- --:--:-- 17728
* Downloading datadog-agent
#####
* Installing datadog-agent, you might be asked for your sudo password...
Password: [REDACTED]
```

fig 3.3 Password Requirement

```

macbookproret.t.b — macbookproret.t.b@MBP-de-MacBook — ~ — zsh — 137x29
      Dload Upload Total Spent Left Speed
100  6861  100  6861    0    0 17728    0 ---:--- ---:--- ---:--- 17728
* Downloading datadog-agent
#####
* Installing datadog-agent, you might be asked for your sudo password...
[Password: ]
- Mounting the DMG installer...
- Unpacking and copying files (this usually takes about a minute) ...
- Unmounting the DMG installer ...
* Restarting the Agent...

Your Agent is running properly. It will continue to run in the
background and submit metrics to Datadog.

You can check the agent status using the "datadog-agent status" command
or by opening the webui using the "datadog-agent launch-gui" command.

If you ever want to stop the Agent, please use the Datadog Agent App or
the launchctl command. It will start automatically at login.

+ ~ git:(master) ✘

```

fig 3.4 Successful Installation of Datadog Agent

You can also **download the DMG package** and install it (you will have to add your `api_key: caac4f3cf75f684ee056cd126a0009b5` in `/opt/datadog-agent/etc/datadog.yaml`).

2. Manage the Agent:

Use the Datadog Agent app in the system tray, or the command line `datadog-agent` (located in `/usr/local/bin`).

Enable/disable integrations in `/opt/datadog-agent/etc/conf.d`.

By default, the agent will run at login (you can disable it via the system tray). If you want to run it at boot time, execute these commands:

```

sudo cp '/opt/datadog-agent/etc/com.datadoghq.agent.plist.example'/
/Library/LaunchDaemons/com.datadoghq.agent.plist

```

```

sudo launchctl load -w /Library/LaunchDaemons/com.datadoghq.agent.plist

```

3.1.2.4 Upgrade from Agent 5 or 6

NB: Agent 7 only supports Python 3. Before upgrading, confirm that any custom checks you have are compatible with Python 3. See the [Python 3 Custom Check Migration guide](#) for more information. If you're not using custom checks or have already confirmed their compatibility, you can upgrade now using one of the commands below.

1. If you're upgrading from Agent 6.

```
DD_AGENT_MAJOR_VERSION=7
DD_API_KEY=caac4f3cf75f684ee056cd126a0009b5 DD_SITE="datadoghq.eu"
bash -c "$(curl -L https://s3.amazonaws.com/dd-agent/scripts/
install_mac_os.sh)"
```

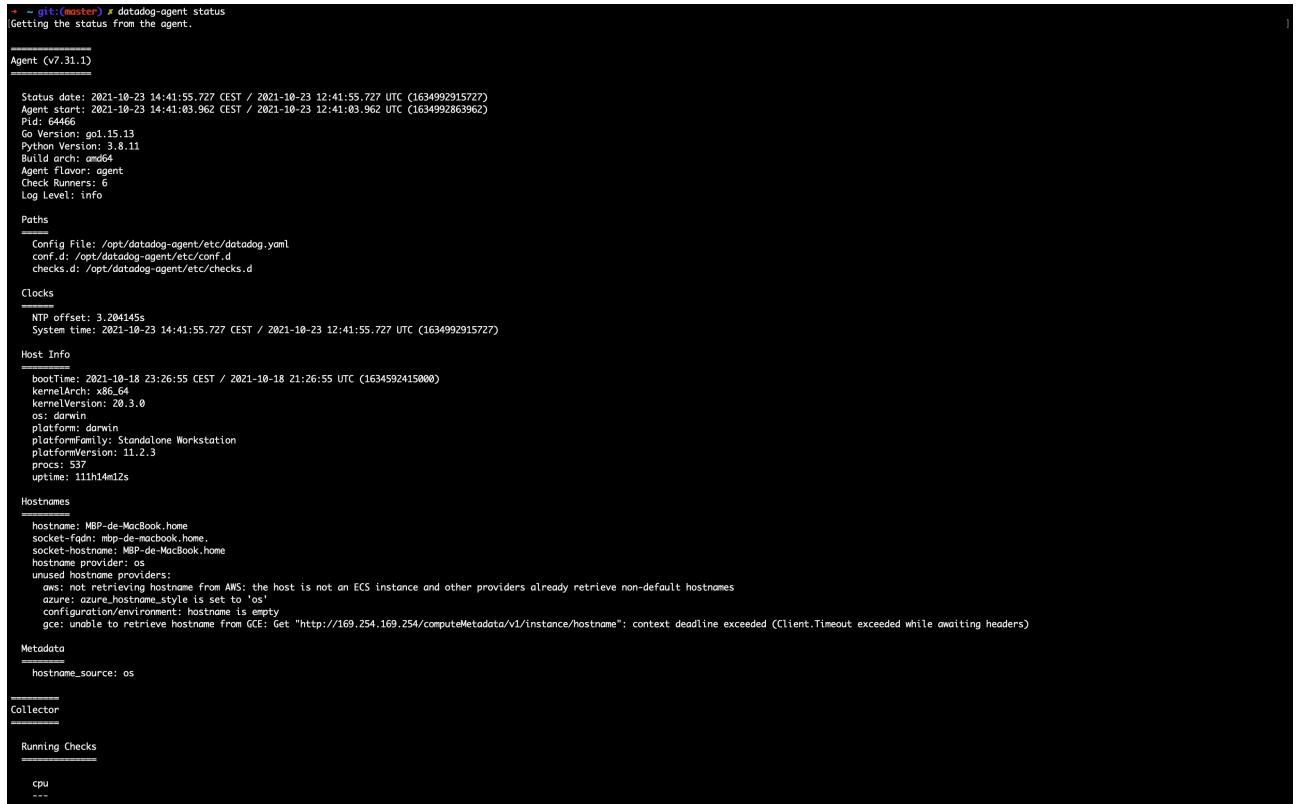
This will install the agent, similarly to what is described above. The existing Agent 6 configuration will be used by Agent 7.

2. If you're upgrading from Agent 5.

```
DD_AGENT_MAJOR_VERSION=7 DD_UPGRADE=true
DD_SITE="datadoghq.eu" bash -c "$(curl -L https://s3.amazonaws.com/dd-
agent/scripts/install_mac_os.sh)"
```

This will install the agent, similarly to what is described above, but we will also import your existing Agent 5 configuration so that you can get up and running immediately.

After you are done installing the agent, you can check the agent status using the “**datadog-agent status**” command



```
+ ~ git:(master) ✘ datadog-agent status
Getting the status from the agent.

=====
Agent (v7.31.1)

=====
Status date: 2021-10-23 14:41:55.727 CEST / 2021-10-23 12:41:55.727 UTC (1634992915727)
Agent start: 2021-10-23 14:41:03.962 CEST / 2021-10-23 12:41:03.962 UTC (1634992863962)
Pid: 64466
Go Version: go1.15.13
Python Version: 3.11
Build Arch: x64
Agent Flavor: agent
Check Runners: 6
Log Level: info

Paths
=====
Config File: /opt/datadog-agent/etc/datadog.yaml
conf.d: /opt/datadog-agent/etc/conf.d
checks.d: /opt/datadog-agent/etc/checks.d

Clocks
=====
NTP offset: 3.204145s
System time: 2021-10-23 14:41:55.727 CEST / 2021-10-23 12:41:55.727 UTC (1634992915727)

Host Info
=====
bootTime: 2021-10-18 23:26:55 CEST / 2021-10-18 21:26:55 UTC (1634592415000)
kernelArch: x86_64
kernelVersion: 20.3.0
os: darwin
platform: darwin
platformFamily: Standalone Workstation
platformVersion: 11.2.3
procs: 557
uptime: 111h14m12s

Hostnames
=====
hostname: MBP-de-MacBook.home
socket: /tmp/datadog.sock
socket-hostname: MBP-de-MacBook.home
hostname provider: os
unused hostname providers:
aws: not retrieving hostname from AWS: the host is not an ECS instance and other providers already retrieve non-default hostnames
azure: azure_hostname_style is set to 'os'
configuration/environment: hostname is empty
gce: unable to retrieve hostname from GCE: Get "http://169.254.169.254/computeMetadata/v1/instance/hostname": context deadline exceeded (Client.Timeout exceeded while awaiting headers)

Metadata
=====
hostname_source: os

=====
Collector

=====
Running Checks
=====
cpu
---
```

fig 3.5 Datadog-agent status

```

Running Checks
-----
cpu
-- 
  Instance ID: cpu [OK]
  Configuration Source: file:/opt/datadog-agent/etc/conf.d/cpu.d/conf.yaml.default
  Total Runs: 4
  Metric Samples: Last Run: 8, Total: 25
  Events: Last Run: 0, Total: 0
  Service Checks: Last Run: 0, Total: 0
  Average Execution Time : 1ms
  Last Execution Date : 2021-10-23 14:41:52 CEST / 2021-10-23 12:41:52 UTC (1634992912000)
  Last Successful Execution Date : 2021-10-23 14:41:52 CEST / 2021-10-23 12:41:52 UTC (1634992912000)

  disk (4.4.0)
-- 
  Instance ID: disk:e5dfb8bef24336f [OK]
  Configuration Source: file:/opt/datadog-agent/etc/conf.d/disk.d/conf.yaml.default
  Total Runs: 3
  Metric Samples: Last Run: 76, Total: 228
  Events: Last Run: 0, Total: 0
  Service Checks: Last Run: 0, Total: 0
  Average Execution Time : 62ms
  Last Execution Date : 2021-10-23 14:41:44 CEST / 2021-10-23 12:41:44 UTC (1634992904000)
  Last Successful Execution Date : 2021-10-23 14:41:44 CEST / 2021-10-23 12:41:44 UTC (1634992904000)

  io
-- 
  Instance ID: io [OK]
  Configuration Source: file:/opt/datadog-agent/etc/conf.d/io.d/conf.yaml.default
  Total Runs: 3
  Metric Samples: Last Run: 13, Total: 30
  Events: Last Run: 0, Total: 0
  Service Checks: Last Run: 0, Total: 0
  Average Execution Time : 0s
  Last Execution Date : 2021-10-23 14:41:51 CEST / 2021-10-23 12:41:51 UTC (1634992911000)
  Last Successful Execution Date : 2021-10-23 14:41:51 CEST / 2021-10-23 12:41:51 UTC (1634992911000)

  load
-- 
  Instance ID: load [OK]
  Configuration Source: file:/opt/datadog-agent/etc/conf.d/load.d/conf.yaml.default
  Total Runs: 3
  Metric Samples: Last Run: 6, Total: 18
  Events: Last Run: 0, Total: 0
  Service Checks: Last Run: 0, Total: 0
  Average Execution Time : 0s
  Last Execution Date : 2021-10-23 14:41:43 CEST / 2021-10-23 12:41:43 UTC (1634992903000)
  Last Successful Execution Date : 2021-10-23 14:41:43 CEST / 2021-10-23 12:41:43 UTC (1634992903000)

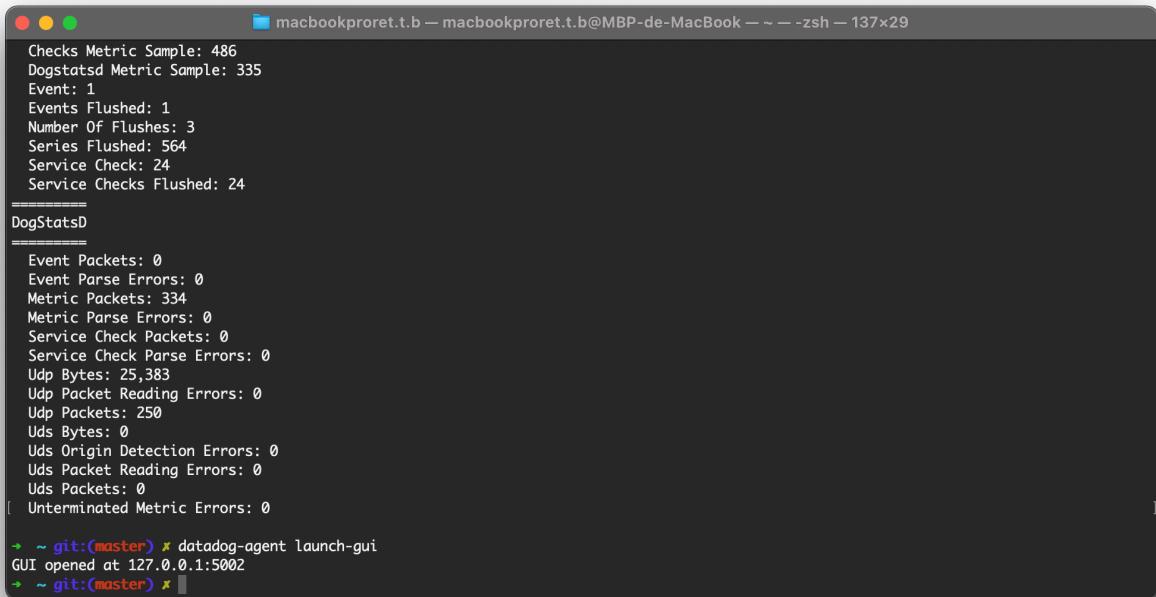
  memory
-----
  Instance ID: memory [OK]
  Configuration Source: file:/opt/datadog-agent/etc/conf.d/memory.d/conf.yaml.default
  Total Runs: 3
  Metric Samples: Last Run: 9, Total: 27
  Events: Last Run: 0, Total: 0
  Service Checks: Last Run: 0, Total: 0
  Average Execution Time : 0s

```

fig 3.6 Datadog-agent status details

or by opening the webui using the “**“datadog-agent launch-gui”** command.

fig 3.7 Datadog Agent Manager



```

Checks Metric Sample: 486
Dogstatsd Metric Sample: 335
Event: 1
Events Flushed: 1
Number Of Flushes: 3
Series Flushed: 564
Service Check: 24
Service Checks Flushed: 24
=====
DogStatsD
Event Packets: 0
Event Parse Errors: 0
Metric Packets: 334
Metric Parse Errors: 0
Service Check Packets: 0
Service Check Parse Errors: 0
Udp Bytes: 25,383
Udp Packet Reading Errors: 0
Udp Packets: 250
Uds Bytes: 0
Uds Origin Detection Errors: 0
Uds Packet Reading Errors: 0
Uds Packets: 0
Unterminated Metric Errors: 0
→ ~ git:(master) ✘ datadog-agent launch-gui
GUI opened at 127.0.0.1:5002
→ ~ git:(master) ✘

```

fig 3.8 Datadog-agent launch-gui

We could go back to our data account on our webpage to check if our host has been detected on our account.

Scroll down and click next.

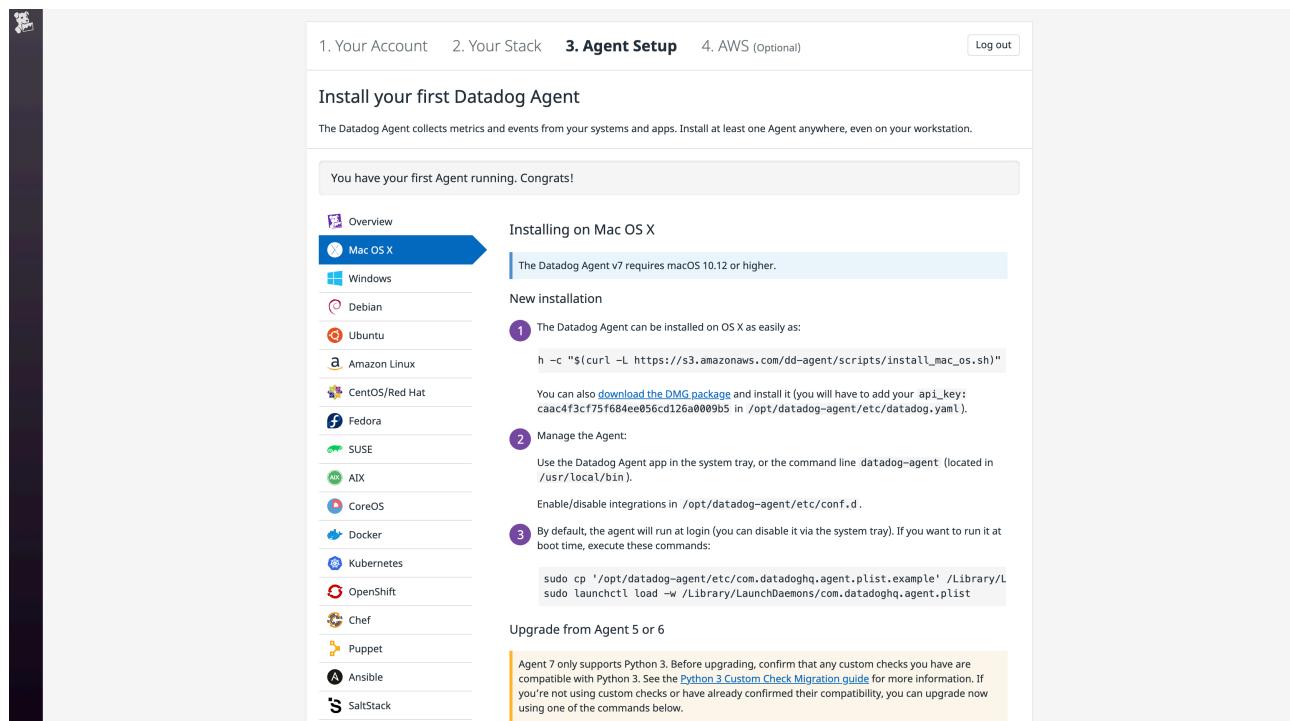


fig 3.9 Datadog-agent setup

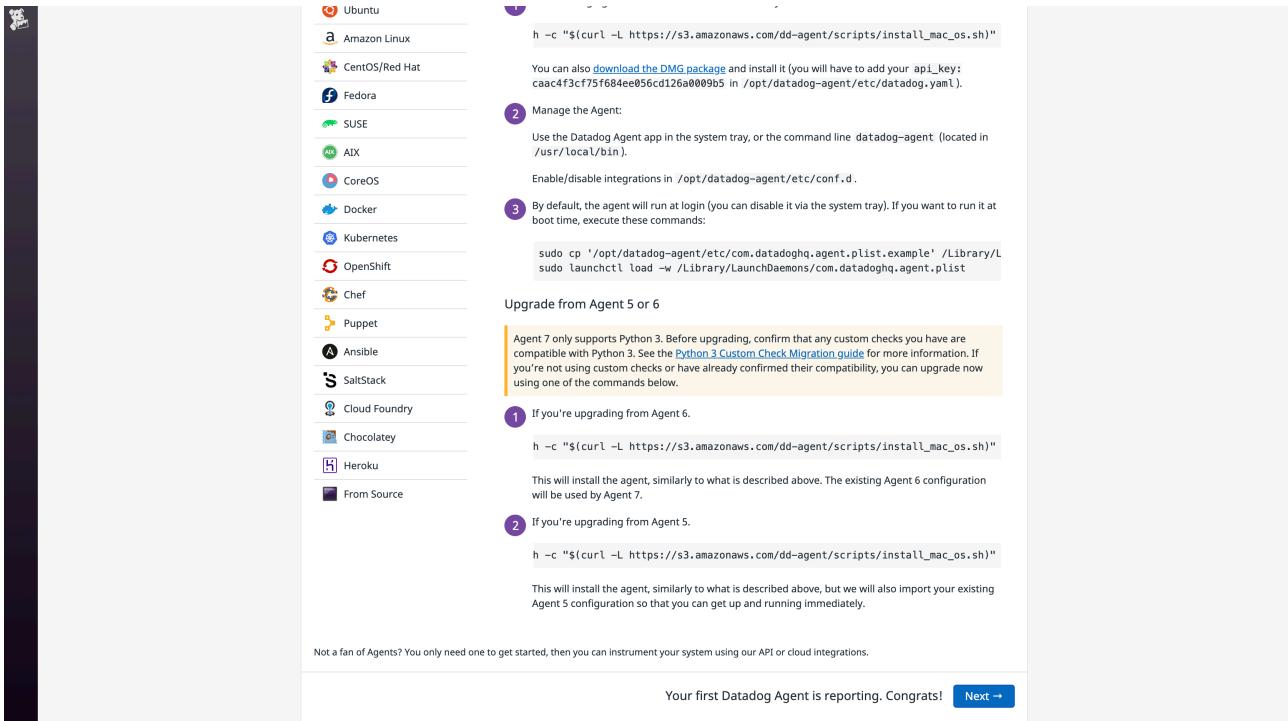


fig 3.10 Datadog-agent setup

Skip this for now since we are installing the agent on our host.

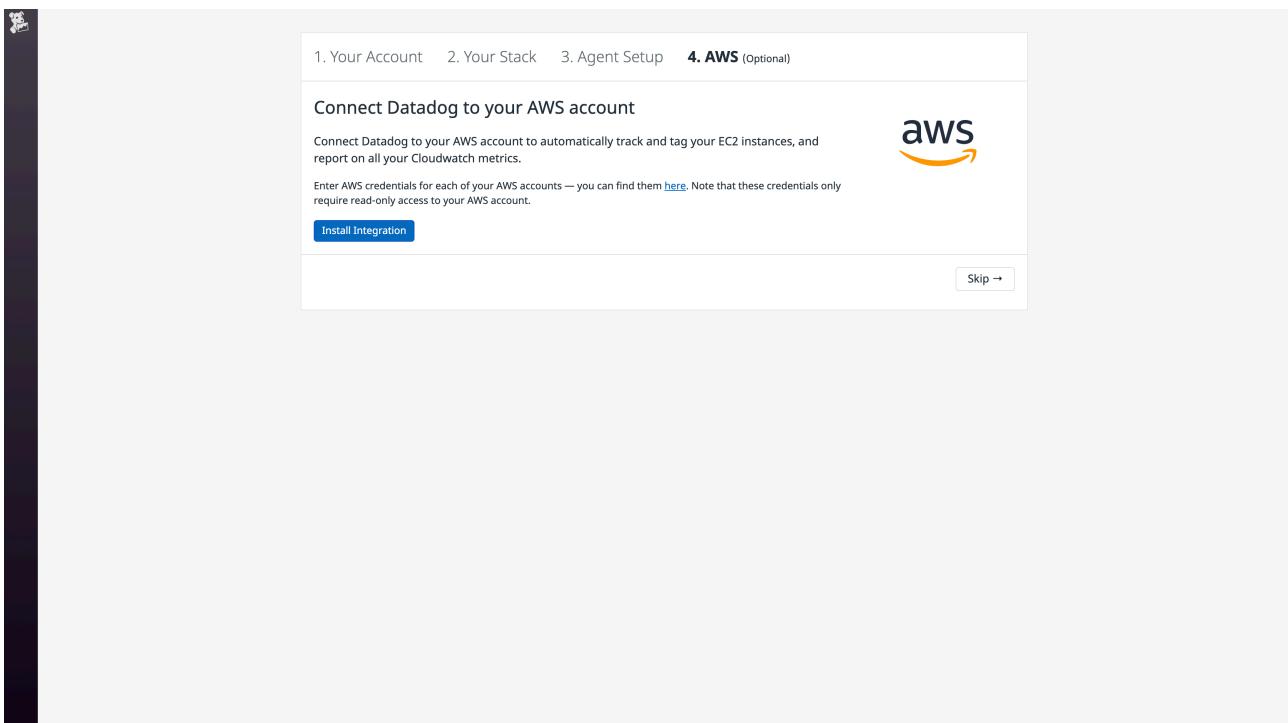


fig 3.11 AWS option

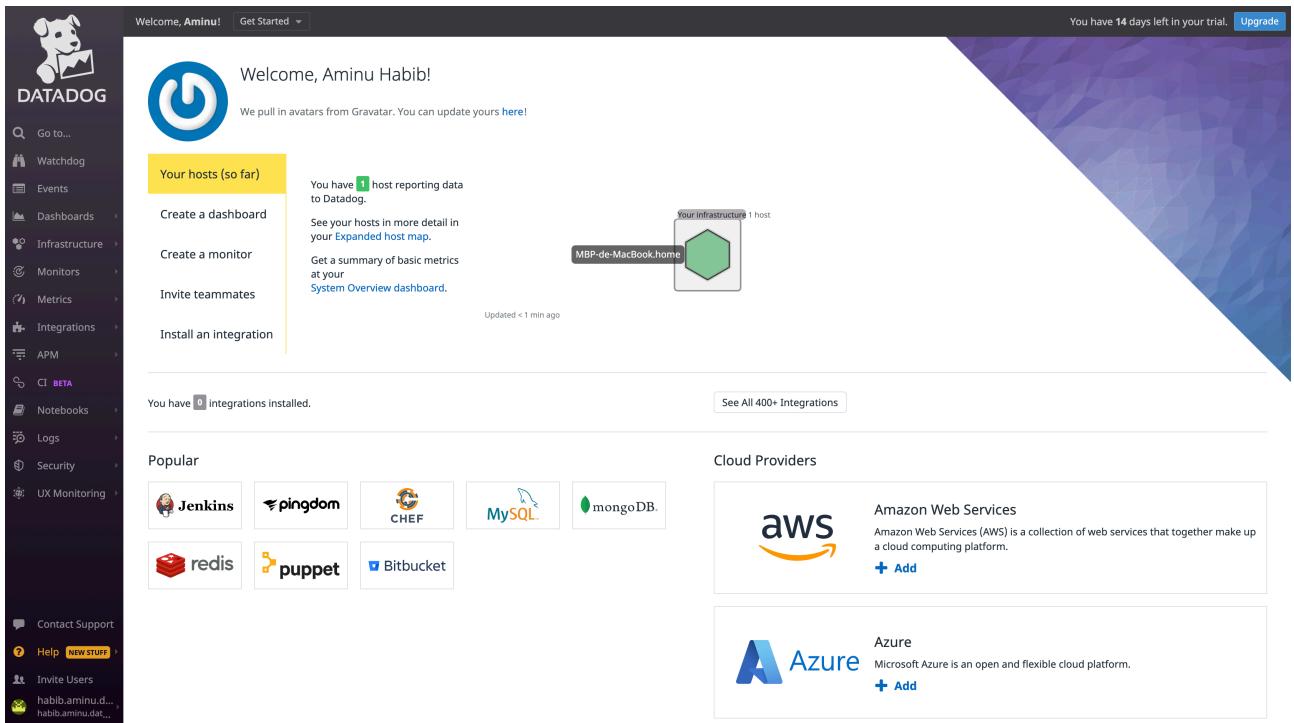


fig 3.12 Datadog Working Environment

This is our datadog working environment. The green node represents our connected machine which is the MacBook.

Click on “Expanded host map” to see your host in more detail.

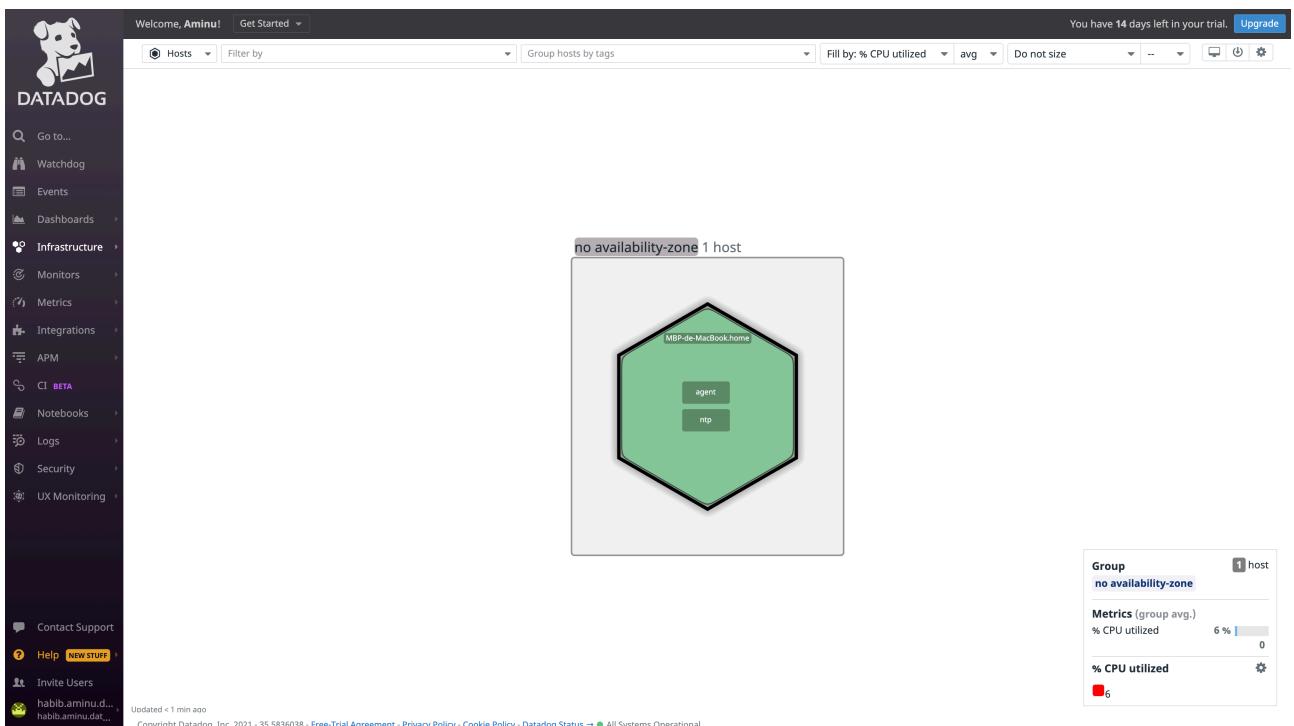


fig 3.13 Host Machine

At the bottom of the right hand side of the screen, we could see some metrics about the CPU usage, click on the node to get more options.

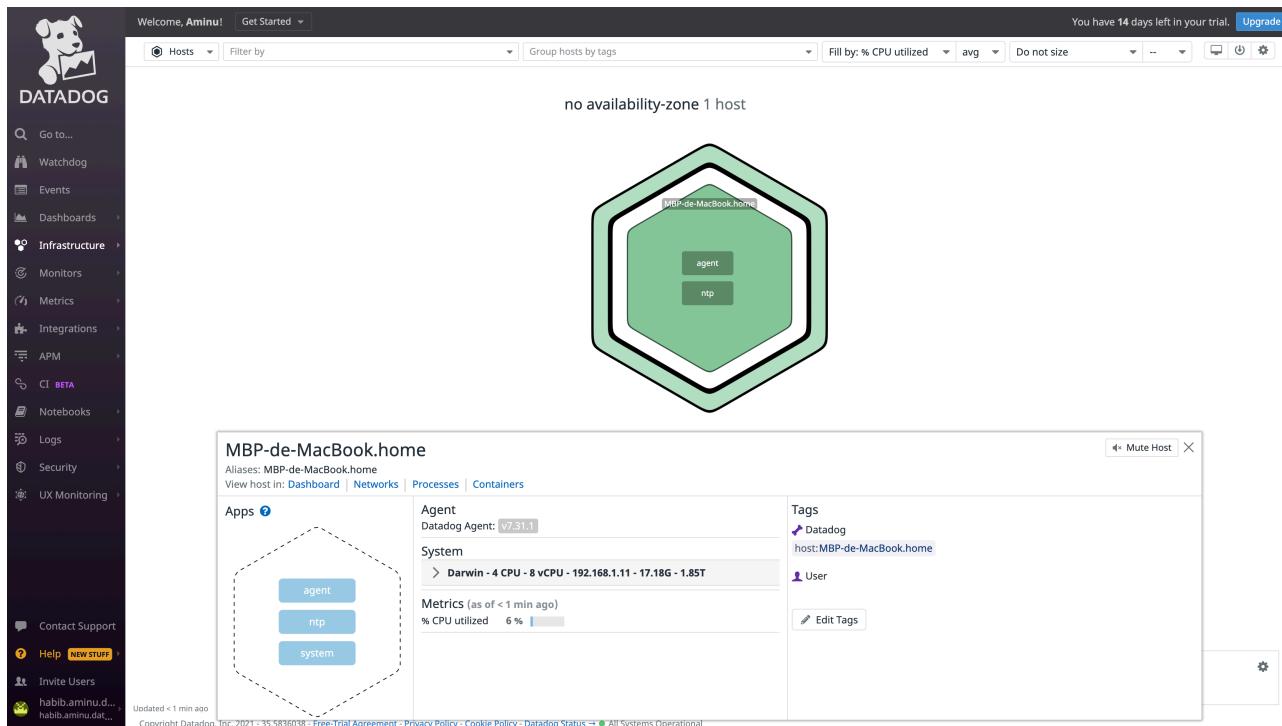


fig 3.14 Host Machine CPU Metric

We could monitor the host in: 1. Dashboard Monitoring

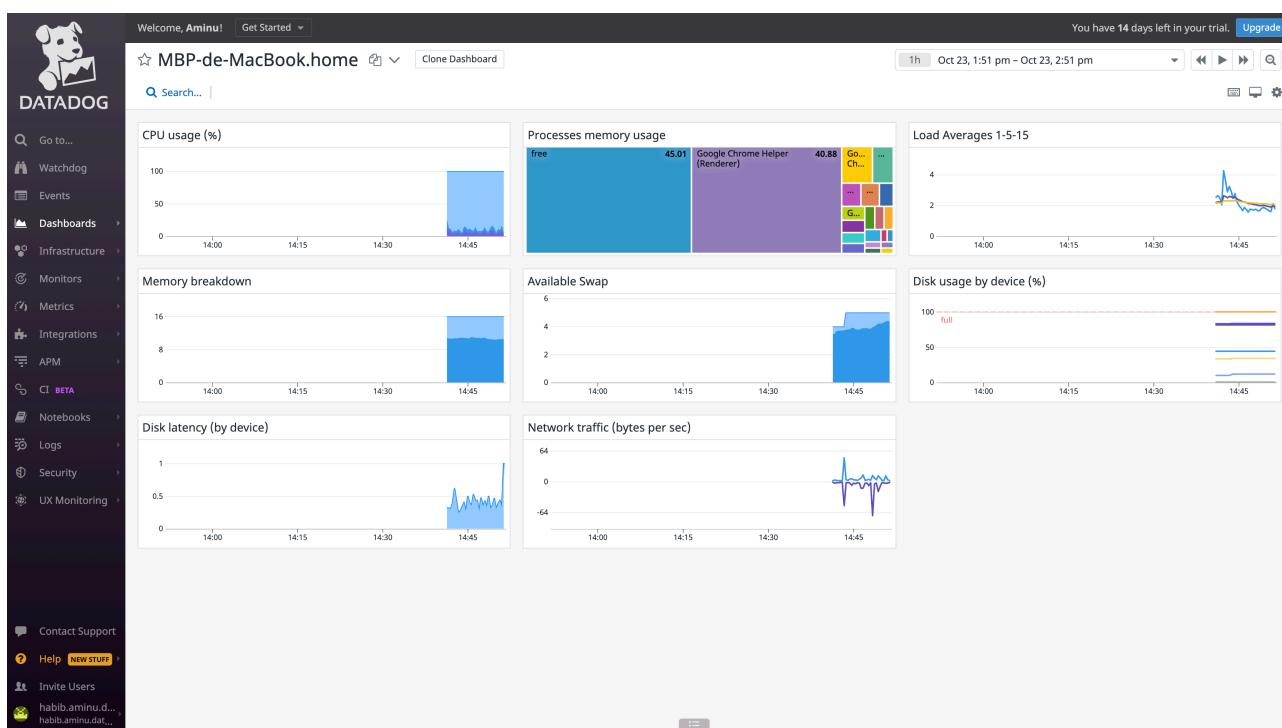


fig 3.14 Host Machine Dashboard Monitoring

2. Network Monitoring

fig 3.15 Network Performance Monitoring

3. Process Monitoring

fig 3.16 Process Monitoring

4. Container Monitoring

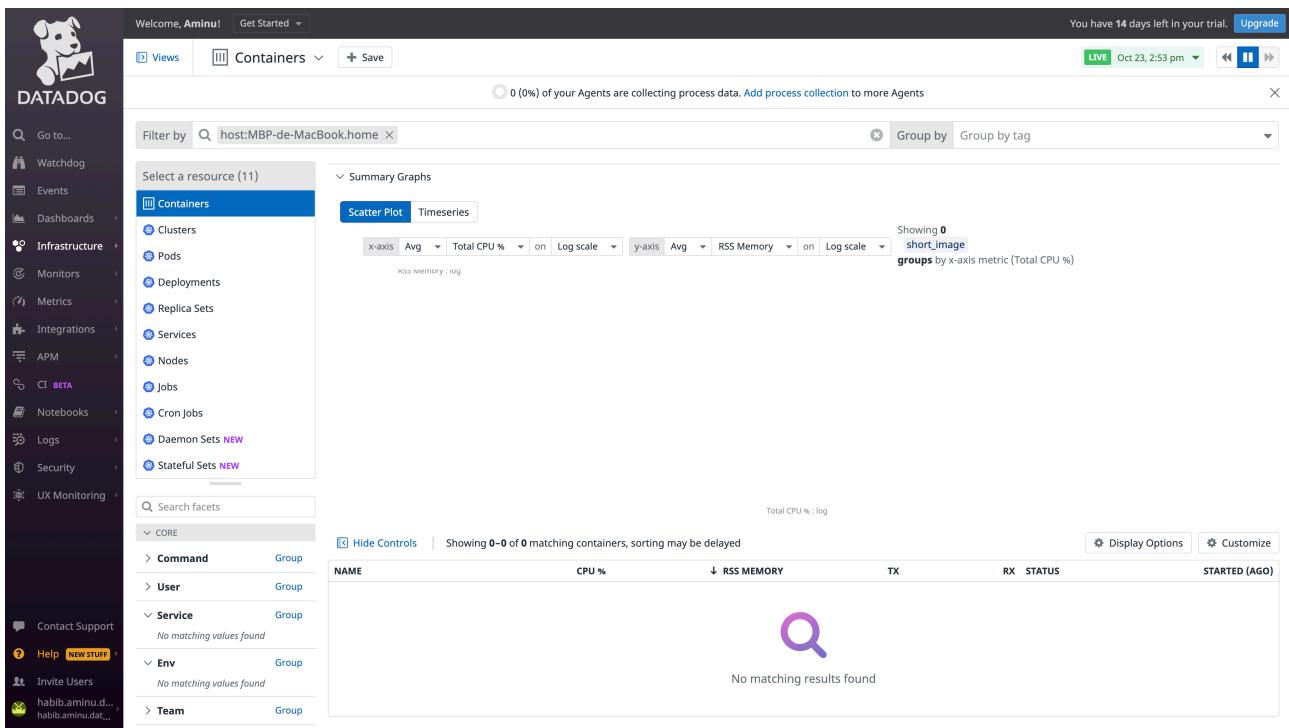


fig 3.17 Container Monitoring

You can stop the agent using the “**datadog-agent stop**” command.

```
macbookproret.t.b ~ macbookproret.t.b@MBP-de-MacBook ~ - zsh - 137x29
Event: 1
Events Flushed: 1
Number Of Flushes: 3
Series Flushed: 564
Service Check: 24
Service Checks Flushed: 24
=====
DogStatsD
=====
Event Packets: 0
Event Parse Errors: 0
Metric Packets: 334
Metric Parse Errors: 0
Service Check Packets: 0
Service Check Parse Errors: 0
Udp Bytes: 25,383
Udp Packet Reading Errors: 0
Udp Packets: 250
Uds Bytes: 0
Uds Origin Detection Errors: 0
Uds Packet Reading Errors: 0
Uds Packets: 0
Unterminated Metric Errors: 0
+ ~ git:(master) ✘ datadog-agent launch-gui
GUI opened at 127.0.0.1:5002
+ ~ git:(master) ✘ datadog-agent stop
Agent successfully stopped
+ ~ git:(master) ✘
```

fig 3.18 Agent Successfully Stopped

CHAPTER 4:

4.1 Datadog with AWS

To monitor a Cloud based host, you would need to create a cloud provider account(AWS, Azure, GCP).

Use this link to create an account with Amazon Web Service: [https://signin.aws.amazon.com/signin?](https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fportal.aws.amazon.com%2Fbilling%2Fsignup%2Fresource&client_id=signup&code_challenge_method=SHA-256&code_challenge=MY73UXmvW_MWcfB7zdQ6XF7djkyD-OeX3K29zazFOf4)

https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fportal.aws.amazon.com%2Fbilling%2Fsignup%2Fresource&client_id=signup&code_challenge_method=SHA-256&code_challenge=MY73UXmvW_MWcfB7zdQ6XF7djkyD-OeX3K29zazFOf4

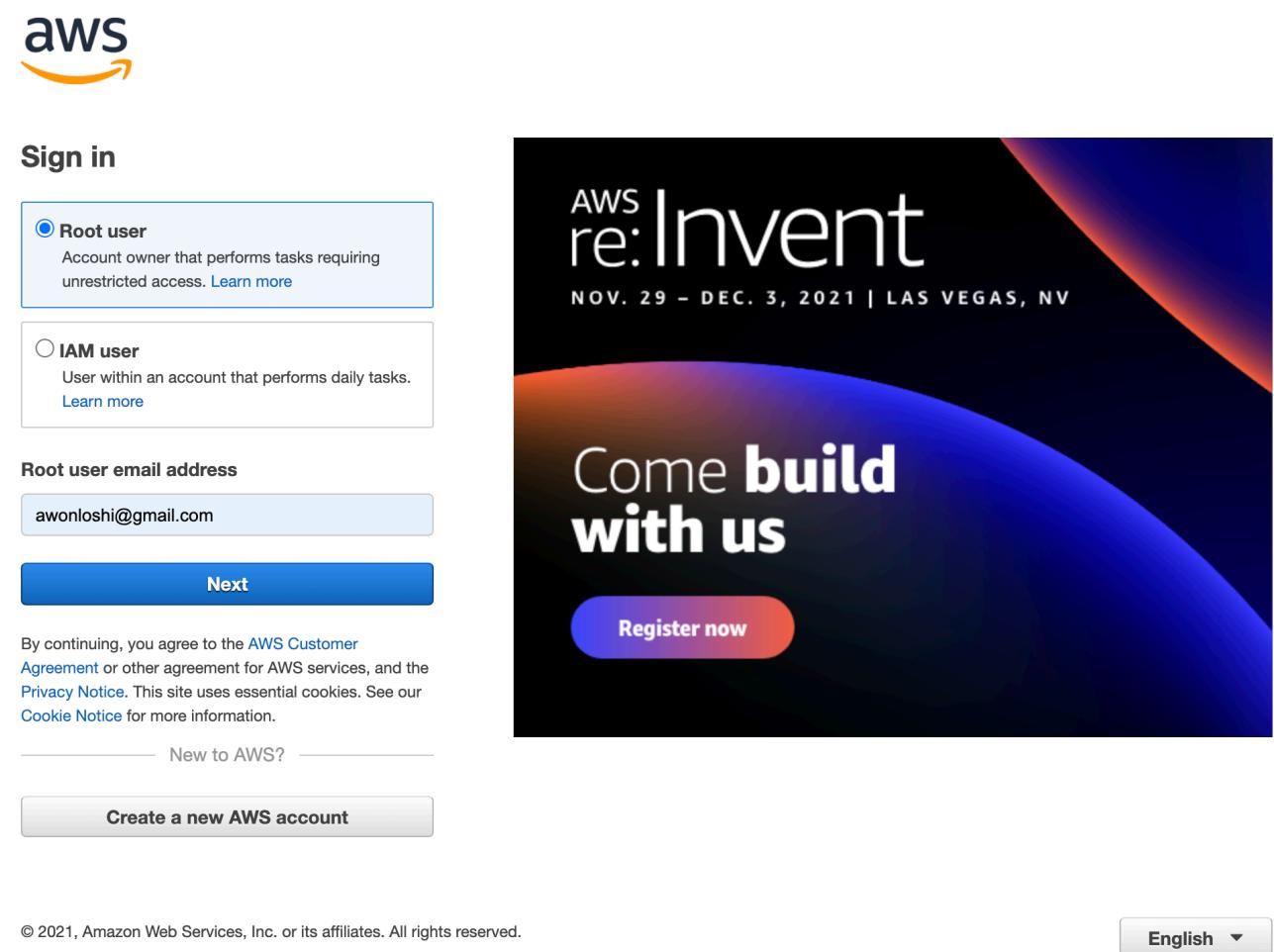
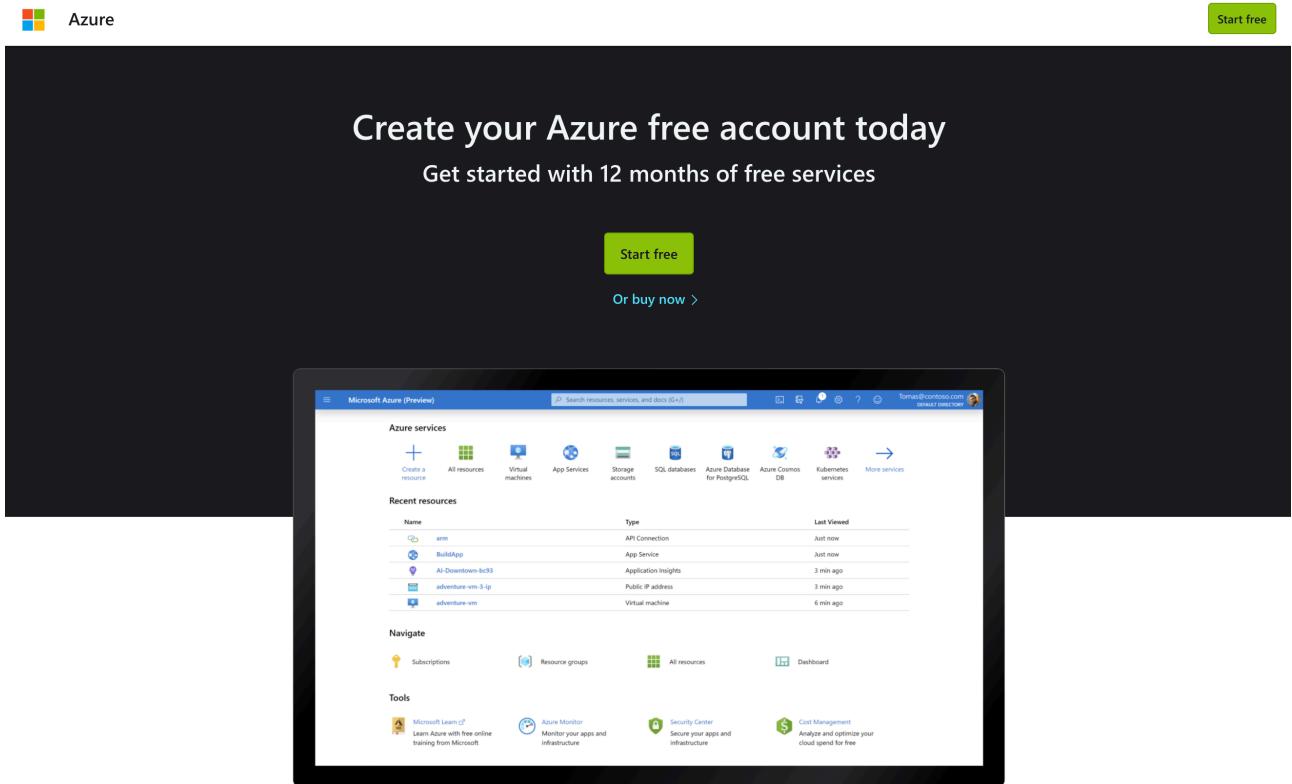


fig 4.1 Create a new AWS account

- Use this link to create an account with Microsoft Azure: https://azure.microsoft.com/en-us/free/search/?&ef_id=CjwKCAjw5c6LBhBdEiwAP9ejG4L0m1aBa_wLm7gYd5dbqVFzSBwWGSwONZoVx-0lkNgCwbv1MMKp2xoCav0QAvD_BwE:G:s&OCID=AID2200187_SEM_CjwKCAjw5c6LBhBdEiwAP9ejG4L0m1aBa_wLm7gYd5dbqVFzSBwWGSwONZoVx-0lkNgCwbv1MMKp2xoCav0QAvD_BwE:G:s&gclid=CjwKCAjw5

c6LBhBdEiwAP9ejG4L0m1aBa_wLm7gYd5dbqVFzSBwWGSwONZoVx-0lkNg
Cwbv1MMKp2xoCav0QAvD_BwE



What do I get?

With your Azure free account, you get all of this—and you won't be charged until you choose to upgrade.

fig 4.2 Create a new Azure account

In this tutorial, we shall be using Amazon Web Service to accomplish the following tasks after creation your account to:

- Spin up a server on the cloud
- Install the agent on the server
- Set up the cloud provider integration
- Create a custom log file for the Agent to tail and display the log lines in the UI
- Create a monitor to alert us on something that needs our attention

Go to your AWS account and launch an EC2 instance (Elastic Cloud Compute) by following this process:

- Click on services and you would locate EC2 on the compute service or you could search for EC2 on the search for services.

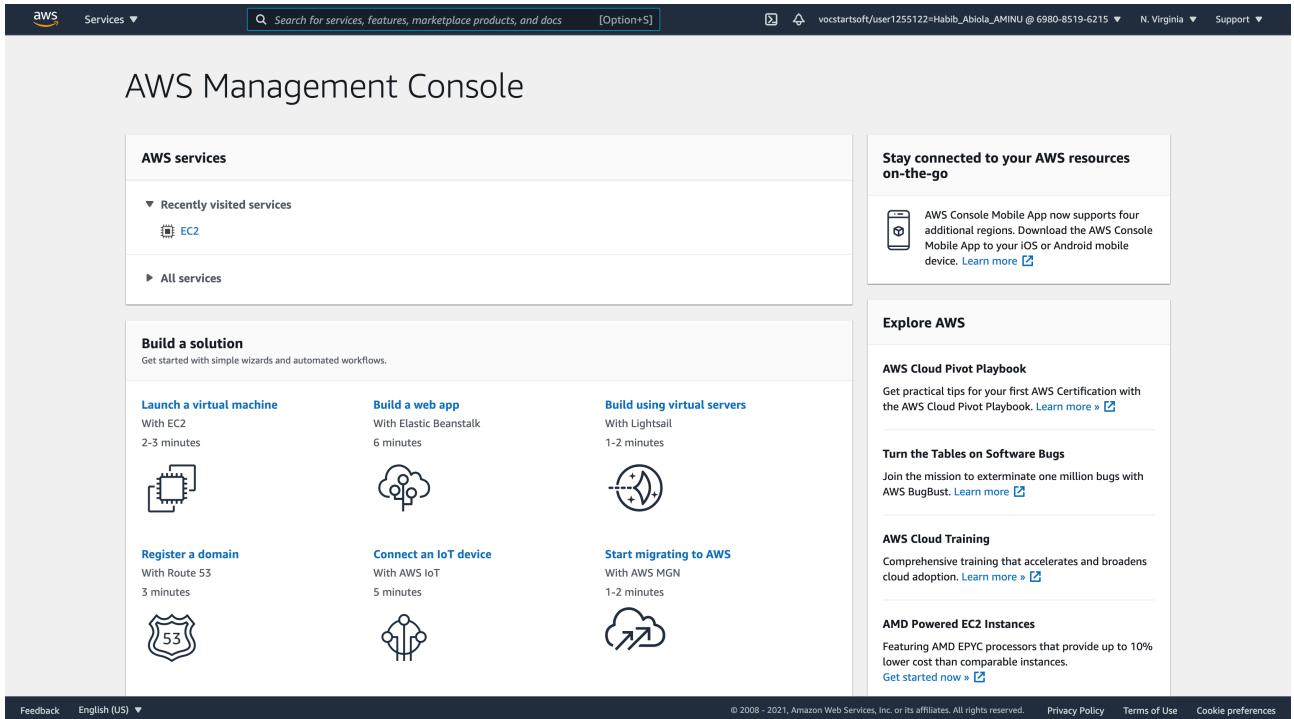


fig 4.3 AWS Management Console

- Click **Launch EC2 instance**
- On the AMI page, select the **Amazon Linux 2 AMI**
- Leave t2.micro selected, and click **Next: Configure Instance Details**.

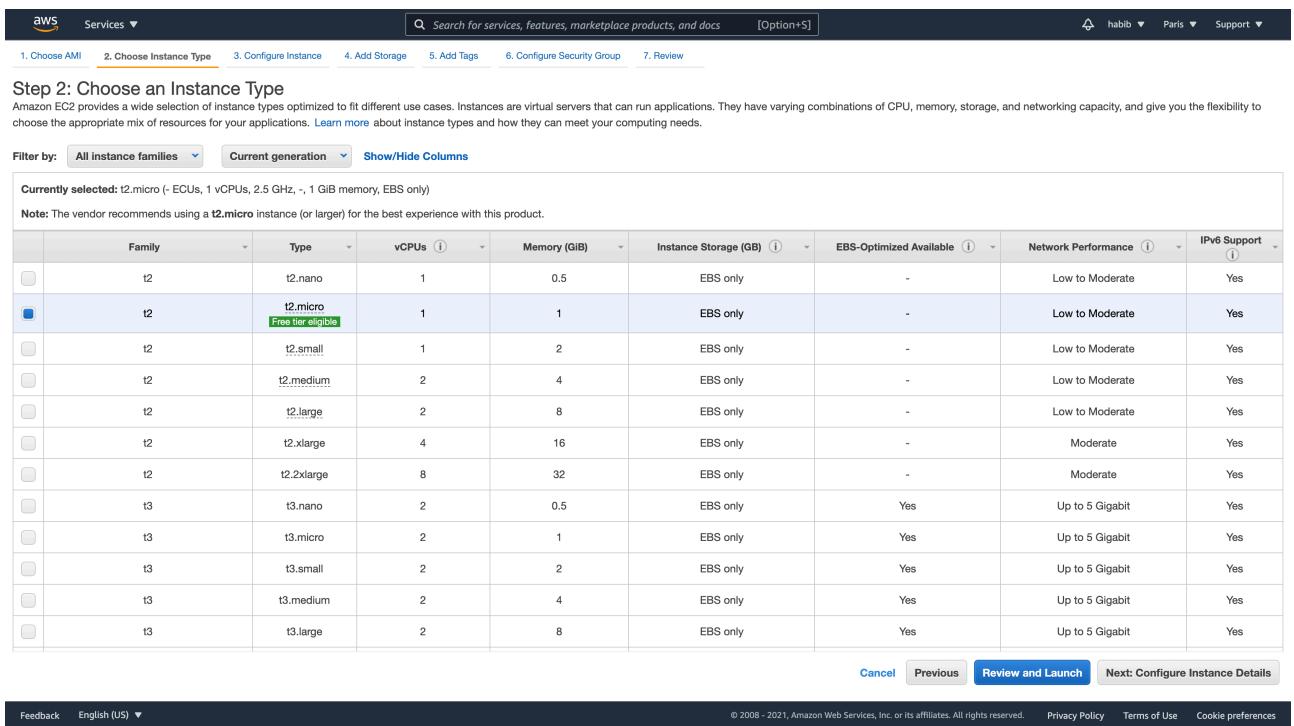


fig 4.4 EC2 Type

- On the Configure Instance Details page:
 - Network: **Default**
 - Subnet: **No preference**
 - Auto-assign Public IP: **Use Subnet(Enable)**
- Expand *Advanced details*, and paste the following into the user data box:

```
#!/bin/bash
```

```
yum update -y
```

```
yum install -y httpd
```

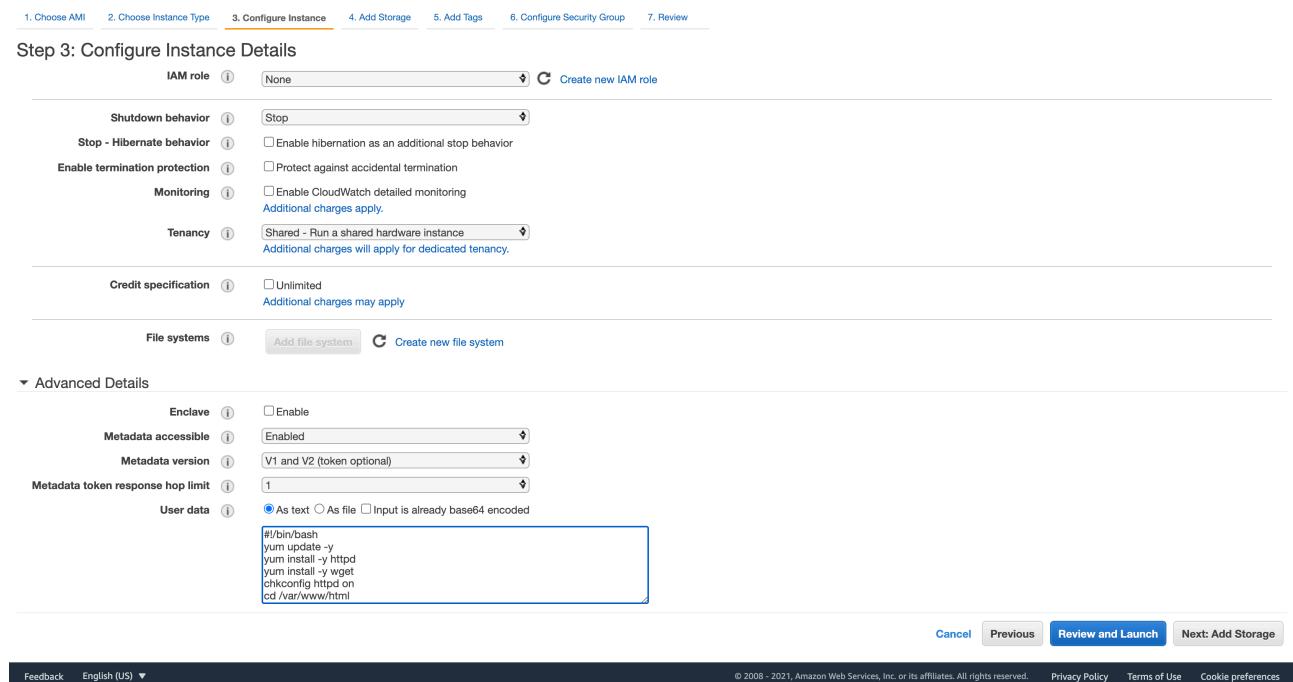
```
yum install -y wget
```

```
chkconfig httpd on
```

```
cd /var/www/html
```

```
service httpd start
```

This command installs an apache server on our EC2 instance.



The screenshot shows the AWS CloudFormation 'Step 3: Configure Instance Details' page. The 'User data' field contains the following shell script:

```
#!/bin/bash
yum update -y
yum install -y httpd
yum install -y wget
chkconfig httpd on
cd /var/www/html
service httpd start
```

The 'Review and Launch' button is highlighted in blue at the bottom of the page.

fig 4.5 Instance Configuration

- Click **Next: Add Storage**, and then click **Next: Add tags**

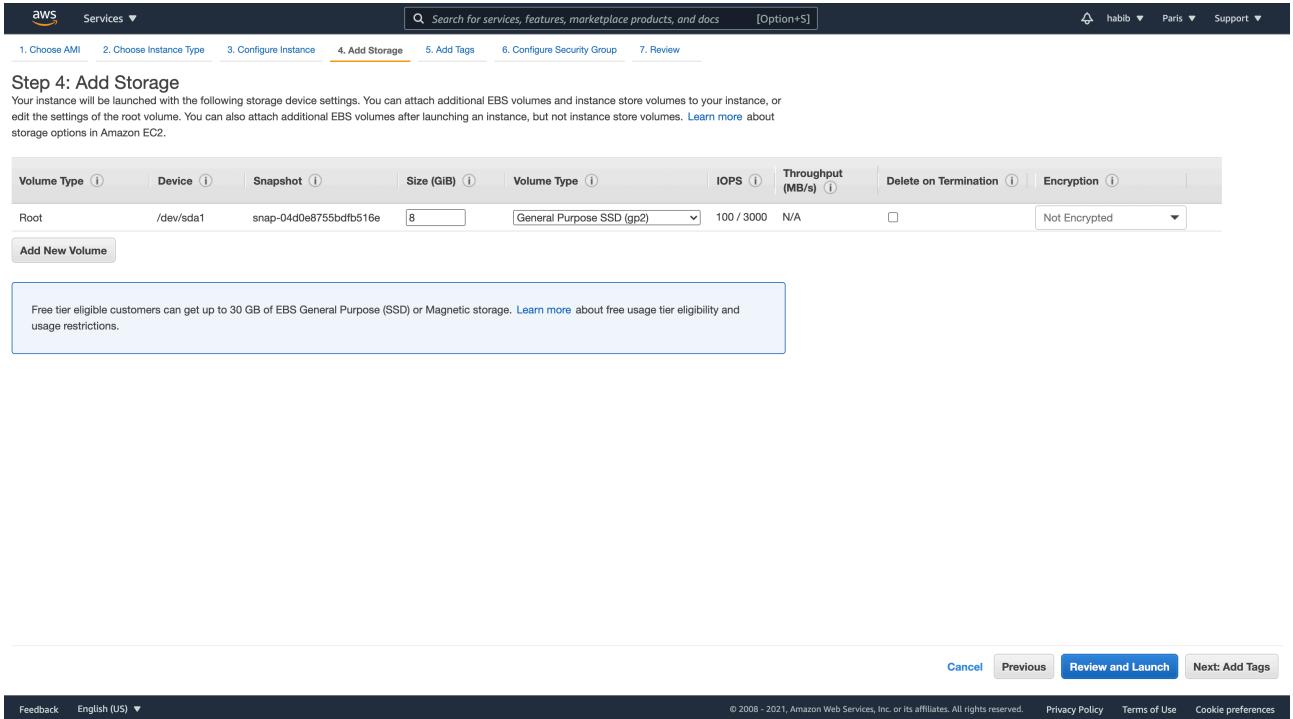


fig 4.6 Add Storage

- On the Add Tags page, select **Add Tag** then add the following:

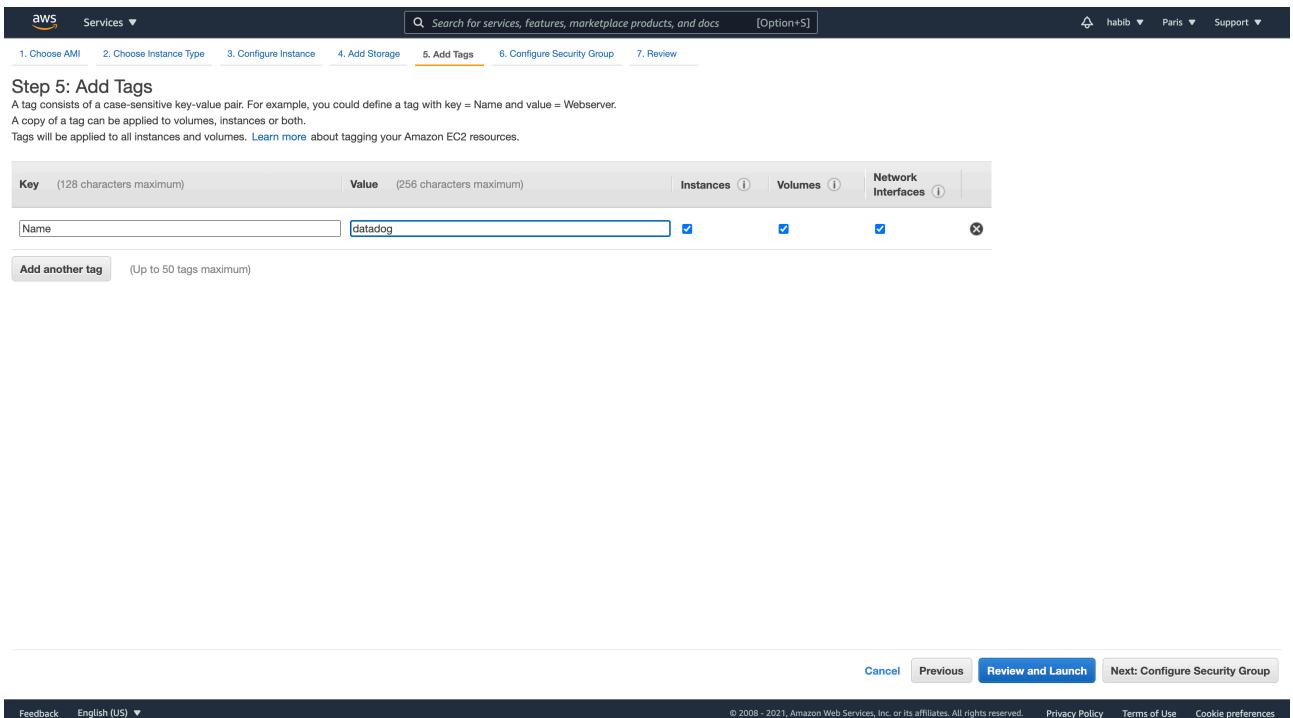


fig 4.7 Add Tag

- Key: Name
- Value: **datadog**
- Click Next: **Configure Security Group**
- On the Configure Security Group page, click **Create a new security group**

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group
 Select an existing security group

Security group name:
Description:

| Type | Protocol | Port Range | Source | Description |
|------|----------|------------|--------|---|
| SSH | TCP | 22 | Custom | 0.0.0.0/0 e.g. SSH for Admin Desktop |

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous **Review and Launch**

Feedback English (US) ▾ © 2008 – 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Cookie preferences

fig 4.8 Create a new Security Group

- Click **Review and Launch**, and then **Launch**.
- In the key pair dialog, select **Create a new key pair**.
- Give it a key pair name of “**datadog**”.
- Click **Download Key Pair**, and then **Launch Instances**.
- Click **View Instances**, and give it a few minutes to enter the running state.
- Click **Instance ID** to get more details.
- Click **Connect**, we would connect to the machine through SSH client.

4.2 Manage the EC2 instance

Once the instance is running with 2/2 status checks:

- Copy the instance’s **public DNS (IPv4)**

ec2-user@52.47.101.218

- On the EC2 instances console, right-click the instance, and select **Connect**.

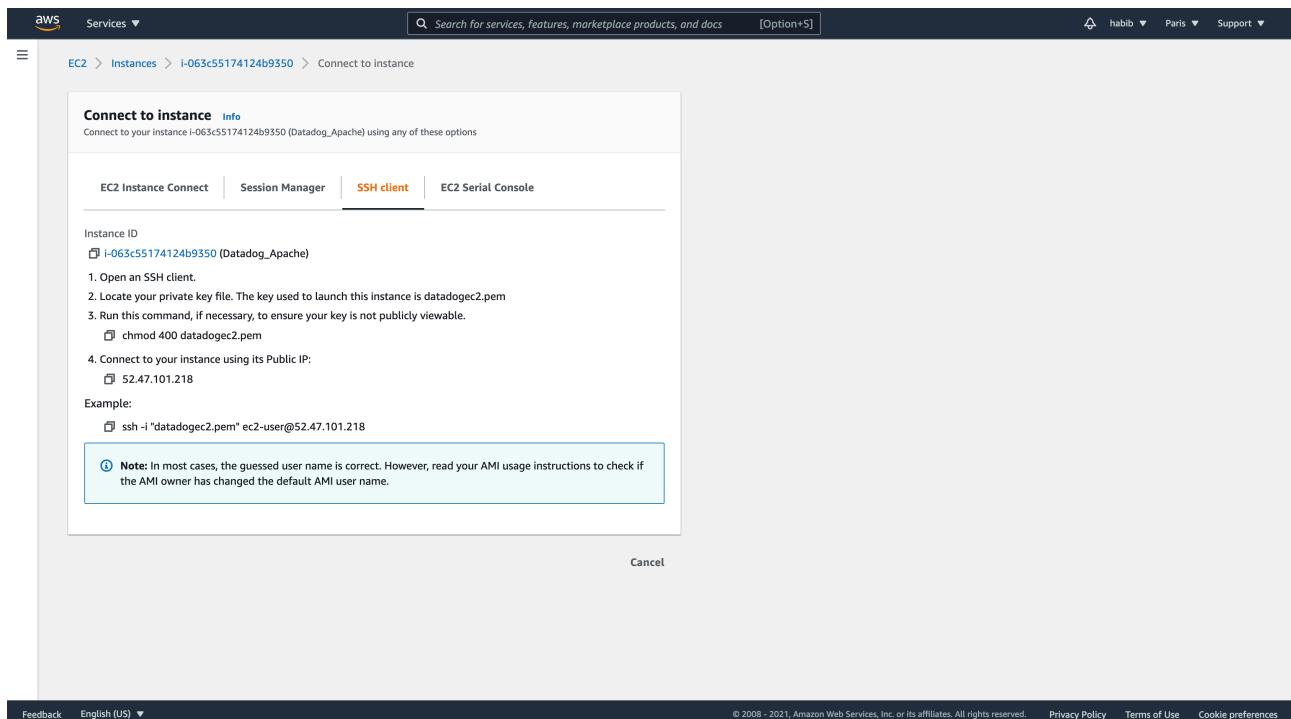


fig 4.9 Connect to Instance

- On the MacBook terminal, change to your Downloads directory (e.g., `cd Downloads`).
- Run `chmod 400 datadog.pem` to adjust the permissions on the file.
- Run `ssh -i "datadogec2.pem" ec2-user@15.237.117.84`
- If using Windows, you will need to follow the instructions found here.
- Connect to the instance using the ssh command provided in the dialog when you click **Connect** (or using the PuTTY instructions).
- Install the agent on the machine

`DD_AGENT_MAJOR_VERSION=7`

`DD_API_KEY=caac4f3cf75f684ee056cd126a0009b5 DD_SITE="datadoghq.eu"`
`bash -c "$(curl -L https://s3.amazonaws.com/dd-agent/scripts/install_script.sh)"`

- To stop the agent, run:

`sudo systemctl stop datadog-agent`

- To start the agent, run:

`sudo systemctl start datadog-agent`

```

Dependencies Resolved
=====
==== Packages =====
Package          Arch      Version       Repository      Size
=====
Installing:
datadog-agent    x86_64    1:7.31.1-1    datadog        222 M
=====
Transaction Summary
=====
Install 1 Package
=====
Total download size: 222 M
Installed size: 731 M
Downloading packages:
warning: /var/cache/yum/x86_64/2/datadog/packages/datadog-agent-7.31.1-1.x86_64.rpm: Header V3 RSA/SHA256 Signature, key ID e09422b3: NOKEY
Public key for datadog-agent-7.31.1-1.x86_64.rpm is not installed
Retrieving key from https://keys.datadoghq.com/DATADOG_RPM_KEY_CURRENT.public
Importing GPG key 0x09422b3:
  Userid : "DataDog Agent <spckage@datadoghq.com>"
  Fingerprint: c655 9b69 0ca8 82f0 23bd f3f6 3f4d 1729 f46b f915
  From   : https://keys.datadoghq.com/DATADOG_RPM_KEY_CURRENT.public
Retrieving key from https://keys.datadoghq.com/DATADOG_RPM_KEY_E09422b3.public
Retrieving key from https://keys.datadoghq.com/DATADOG_RPM_KEY_FD4BF915.public
Importing GPG key 0x0d4bf915:
  Userid : "DataDog, Inc. RPM key (2020-09-08) <package+pmkey@datadoghq.com>"
  Fingerprint: c655 9b69 0ca8 82f0 23bd f3f6 3f4d 1729 f46b f915
  From   : https://keys.datadoghq.com/DATADOG_RPM_KEY_FD4BF915.public
Running transaction test
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:datadog-agent-7.31.1-1.x86_64
  Enabling service datadog-agent
  Created symlink from /etc/systemd/system/multi-user.target.wants/datadog-agent.service to /usr/lib/systemd/system/datadog-agent.service.
  No datadog.yaml file detected, not starting the agent
  Verifying  : 1:datadog-agent-7.31.1-1.x86_64
  1/1
=====
Installed:
  datadog-agent.x86_64 1:7.31.1-1
=====
Complete!
* Adding your API key to the Agent configuration: /etc/datadog-agent/datadog.yaml
* Setting SITE in the Agent configuration: /etc/datadog-agent/datadog.yaml
/usr/bin/systemctl
* Starting the Agent...
[ec2-user@ip-10-0-1-46 ~]$ 

Your Agent is running and functioning properly. It will continue to run in the
background and submit metrics to Datadog.

If you ever want to stop the Agent, run:
  sudo systemctl stop datadog-agent

And to run it again run:
  sudo systemctl start datadog-agent
[ec2-user@ip-10-0-1-46 ~]$ systemctl status datadog-agent
● datadog-agent.service - Datadog Agent
  Loaded: loaded (/usr/lib/systemd/system/datadog-agent.service; enabled; vendor preset: disabled)
  Active: active (running) since Sat 2021-10-23 21:48:53 UTC; 12min ago
    Main PID: 3490 (Agent)
   CGroup: /system.slice/datadog-agent.service
          └─3490 /opt/datadog-agent/bin/agent/run -p /opt/datadog-agent/run/agent.pid

Oct 23 21:50:19 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:50:19 UTC | CORE | INFO | (pkg/collector/worker/check_logger.go:37 in CheckStarted) | check:memory | Running check...
Oct 23 21:50:20 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:50:20 UTC | CORE | INFO | (pkg/collector/worker/check_logger.go:37 in CheckStarted) | check:io | Running check...
Oct 23 21:50:21 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:50:21 UTC | CORE | INFO | (pkg/collector/worker/check_logger.go:37 in CheckStarted) | check:disk | Running check...
Oct 23 21:50:26 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:50:26 UTC | CORE | INFO | (pkg/collector/worker/check_logger.go:37 in CheckStarted) | check:network | Running check...
Oct 23 21:50:27 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:50:27 UTC | CORE | INFO | (pkg/collector/worker/check_logger.go:37 in CheckStarted) | check:load | Running check...
Oct 23 21:50:28 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:50:28 UTC | CORE | INFO | (pkg/collector/worker/check_logger.go:37 in CheckStarted) | check:cpu | Running check...
Oct 23 21:53:54 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:53:54 UTC | CORE | INFO | (pkg/serializer/serializer.go:395 in SendProcessesMetadata) | Sent processes metadata payload, size (raw/compressed): 1050/286 bytes.
Oct 23 21:53:59 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:53:59 UTC | CORE | INFO | (pkg/serializer/serializer.go:395 in SendProcessesMetadata) | Sent processes metadata payload, size (raw/compressed): 874/284 bytes.
Oct 23 21:58:58 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:58:58 UTC | CORE | INFO | (pkg/serializer/serializer.go:371 in sendMetadata) | Sent metadata payload, size (raw/compressed): 874/284 bytes.
Oct 23 21:58:59 ip-10-0-1-46.eu-west-3.compute.internal agent[3490]: 2021-10-23 21:58:59 UTC | CORE | INFO | (pkg/serializer/serializer.go:395 in SendProcessesMetadata) | Sent processes metadata payload, size: 1361 bytes.
[ec2-user@ip-10-0-1-46 ~]$ 

```

fig 4.10 Successful Agent Installation

- Run `systemctl status datadog-agent` to check the status.
- Go to your datadog account to verify if the new host has been detected.
- You could create a dashboard, monitor, invite teammates and install an integration.
- Click infrastructure.
- Run `ls` on your terminal to verify if we have the `ddagent-install.log`
- Run `vi /etc/datadog-agent/datadog.yaml` if you want to customize your configuration.
- On the Datadog webpage, click on the hostname you want to inspect
- We could inspect the:
 - metrics

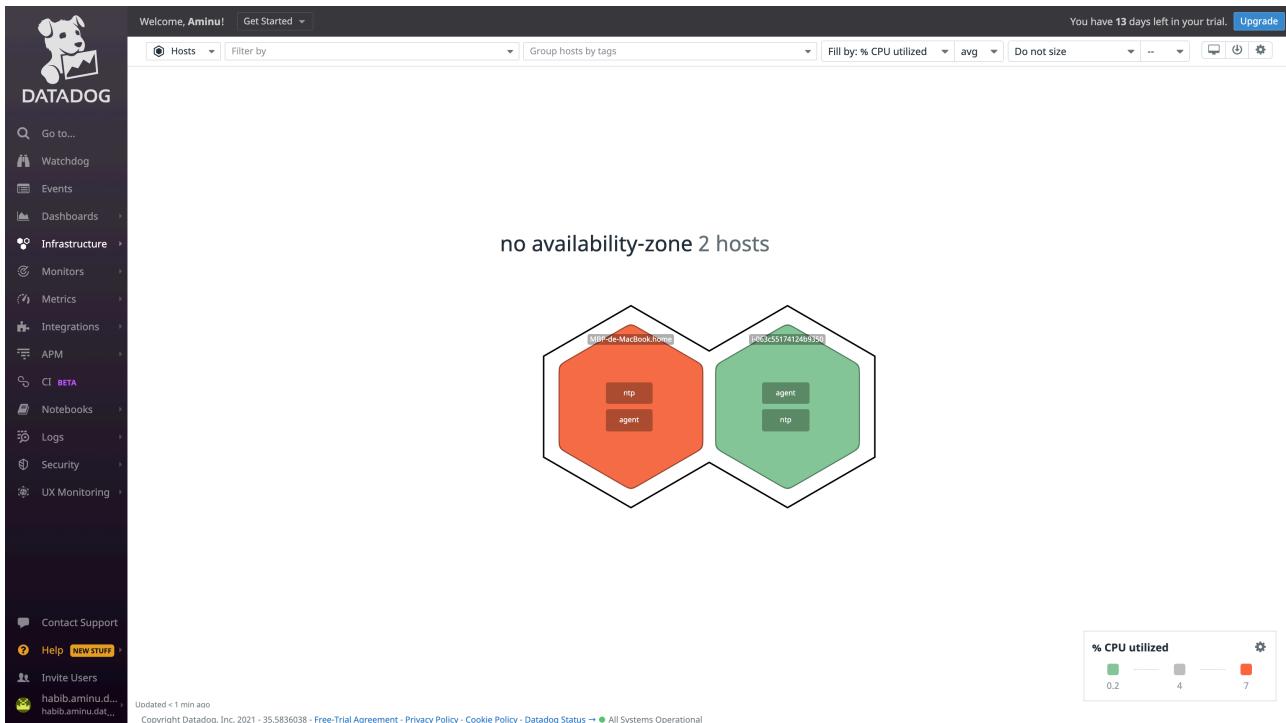


fig 4.11 Connected Host

- Containers
- Processes
- Network

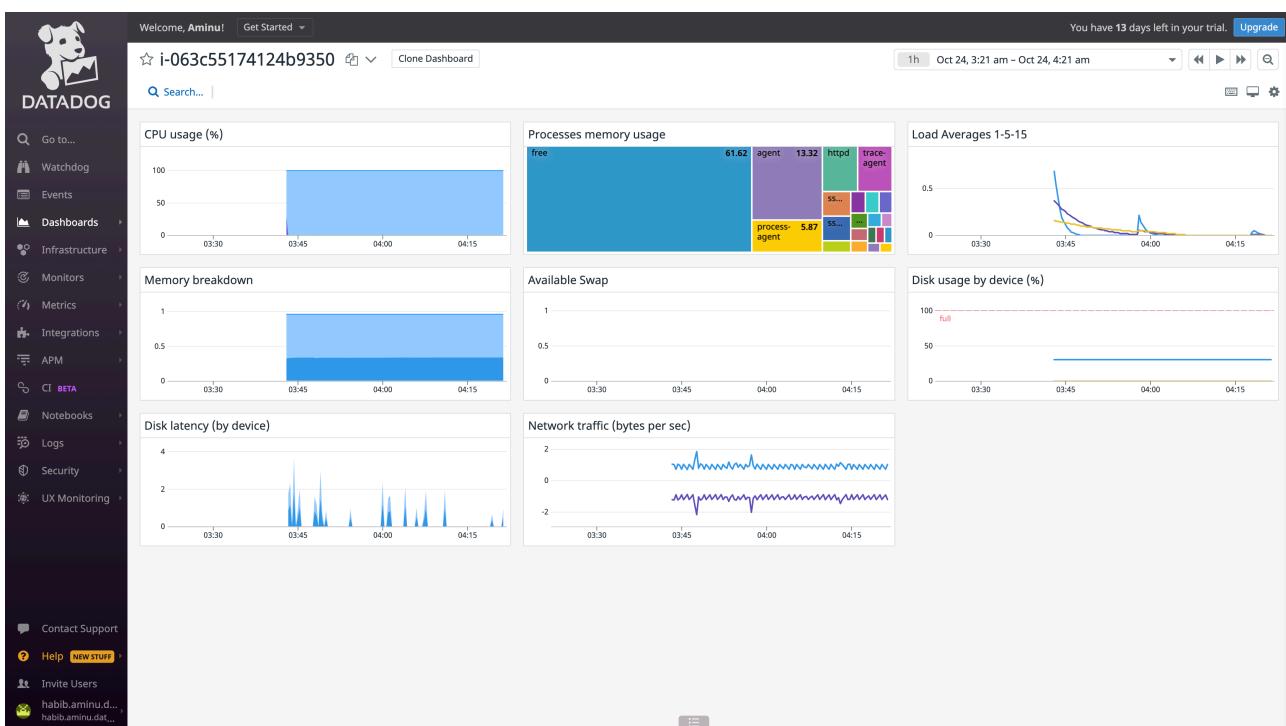


fig 4.12 Host Dashboard

- Log
- Traces
- Security
- Click **Integrations** to install a service to monitor

fig 4.13 Integration Service

- Search for Apache Integration. The Apache check tracks requests per second, bytes served, number of worker threads, service uptime, and more.

Apache Integration
Track requests per second, bytes served, worker threads, uptime, and more.

AVAILABLE

Overview **Configuration** **Metrics** **Monitors**

Installation
The Apache check is packaged with the Agent. To start gathering your Apache metrics and logs, you need to:

1. [Install the Agent](#) on your Apache servers.
2. Install `mod_status` on your Apache servers and enable `ExtendedStatus`.

Configuration

Host
To configure this check for an Agent running on a host:

Metric collection
1. Edit the `apache.d/conf.yaml` file in the `conf.d` folder at the root of your [Agent's configuration directory](#) to start collecting your Apache metrics. See the [sample apache.d/conf.yaml](#) for all available configuration options.

```
init_config:
  instances:
    ## #param apache_status_url - string - required
    ## Status url of your Apache server.
    #
    - apache_status_url: http://localhost/server-status?auto
```

Log collection
Available for Agent versions >6.0

1. Collecting logs is disabled by default in the Datadog Agent. Enable it in `datadog.yaml`:

```
logs_enabled: true
```

- Use this link to install the mod_status on your Apache servers and enable ExtendedStatus: <https://www.tecmint.com/monitor-apache-web-server-load-and-page-statistics/>
- Run vi /etc/httpd/conf/httpd.conf
- Search for the word “mod_status” or keep scrolling down until you find a line containing.
- If you could not find it, run :q! then **more /etc/httpd/conf/httpd.conf | grep status** to verify if it not truly not there.
- We can introduce the whole module if it is not available to us
- Run again vi /etc/httpd/conf/httpd.conf and insert LoadModule status_module modules/mod_status.so then :wq
- Configure mod_status: vi /etc/httpd/conf/httpd.conf and insert

<Location /server-status>

SetHandler server-status

Order allow,deny

Deny from all

Allow from all

Allow from <Host Private Ip Address>

</Location>

ExtendedStatus On

- Restart the apache server: systemctl stop httpd
- Run systemctl start httpd
- Now make sure that you've correctly enabled and configured the Apache server-status page. You can also check for the errors in the **httpd.conf** configuration using the following command.

httpd -t

fig. Apache Configuration

- To activate the Metric Collection, run **cd /etc/datadog-agent/conf.d/** and **ls** to see the available services
- Run **cd apache.d** and **ls**
- Run **cp conf.yaml.example conf.yaml** to copy and modify the configuration
- Run **vi conf.yaml** to see if it is activated
- Restart the agent

4.3 Log collection

- Collecting logs is disabled by default in the Datadog Agent. Enable it in **datadog.yaml**:

logs_enabled: true

- Add this configuration block to your **apache.d/conf.yaml** file to start collecting your Apache logs, adjusting the **path** and **service** values to configure them for your environment:

```

- type: file
  path: /path/to/your/apache/access.log
  source: apache
  service: apache
  sourcename: http_web_access

- type: file
  path: /path/to/your/apache/error.log
  source: apache
  service: apache
  sourcename: http_web_error

```

- Click **Install** on the Datadog webpage.

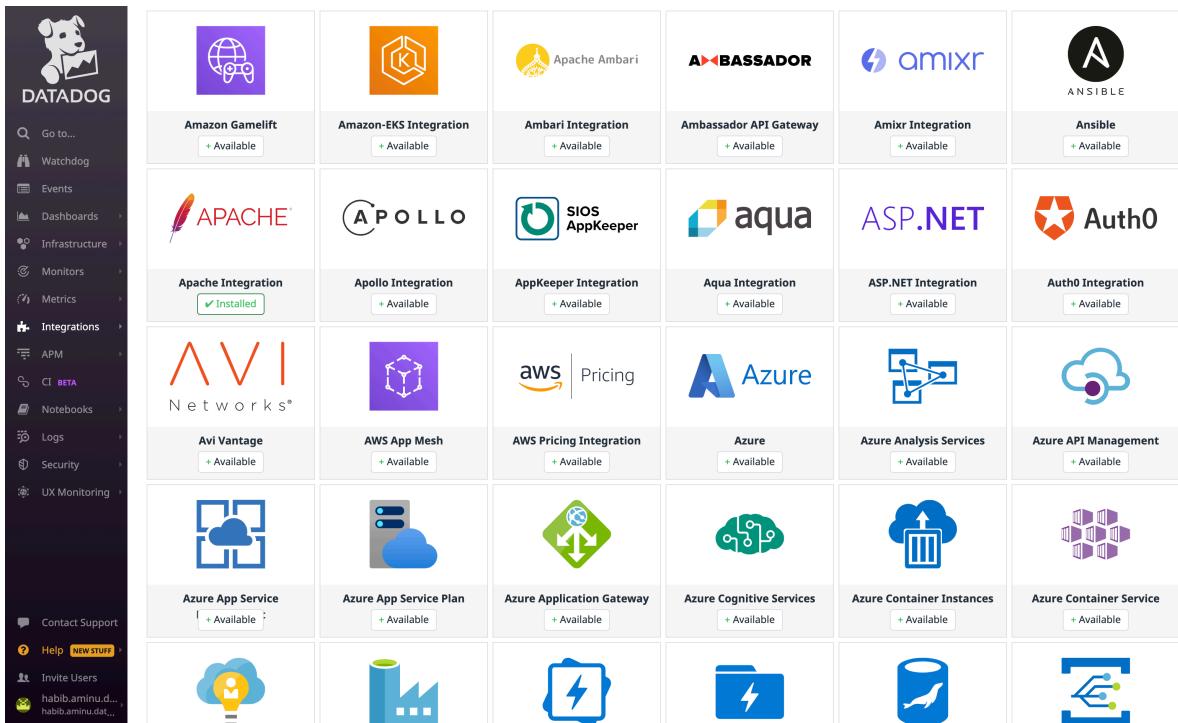


fig 4.14 Apache Installed

- Click **Metrics** and select the metric you want from the list

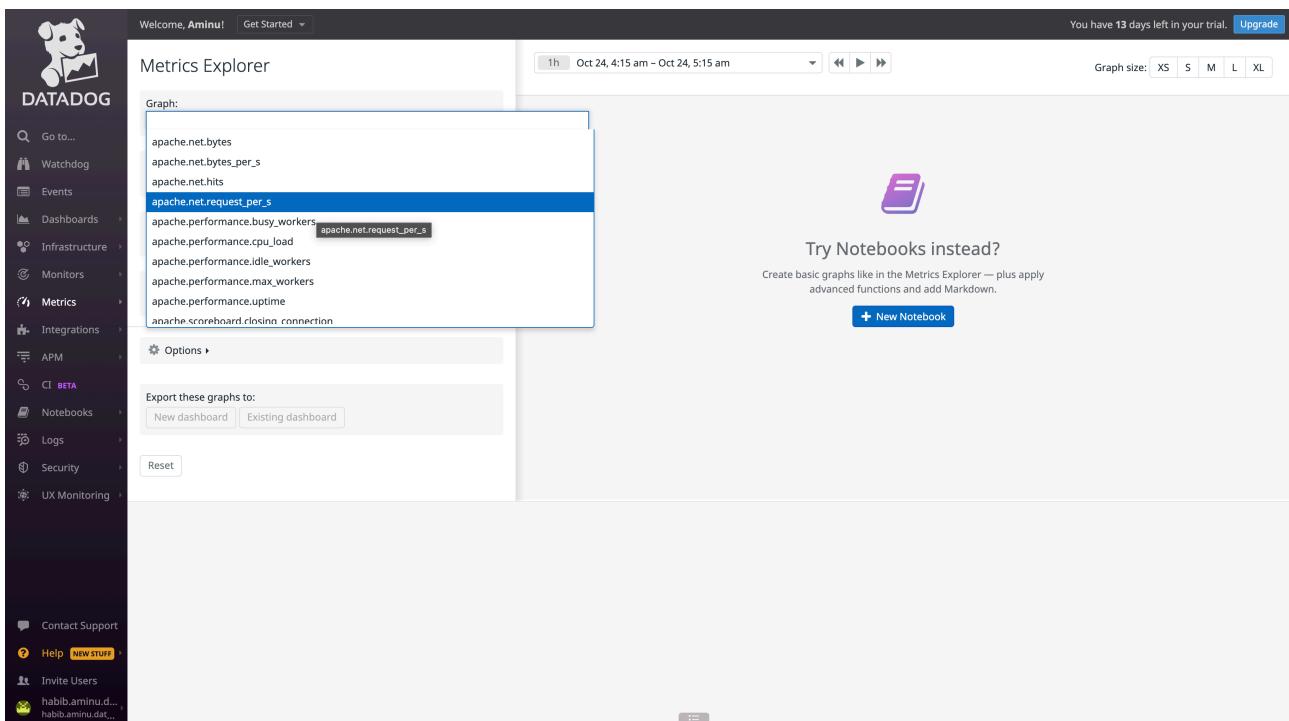


fig 4.15 Metrics

- Run `dd if=/dev/zero of=/dev/null` on the terminal to create a workload for the cloud host

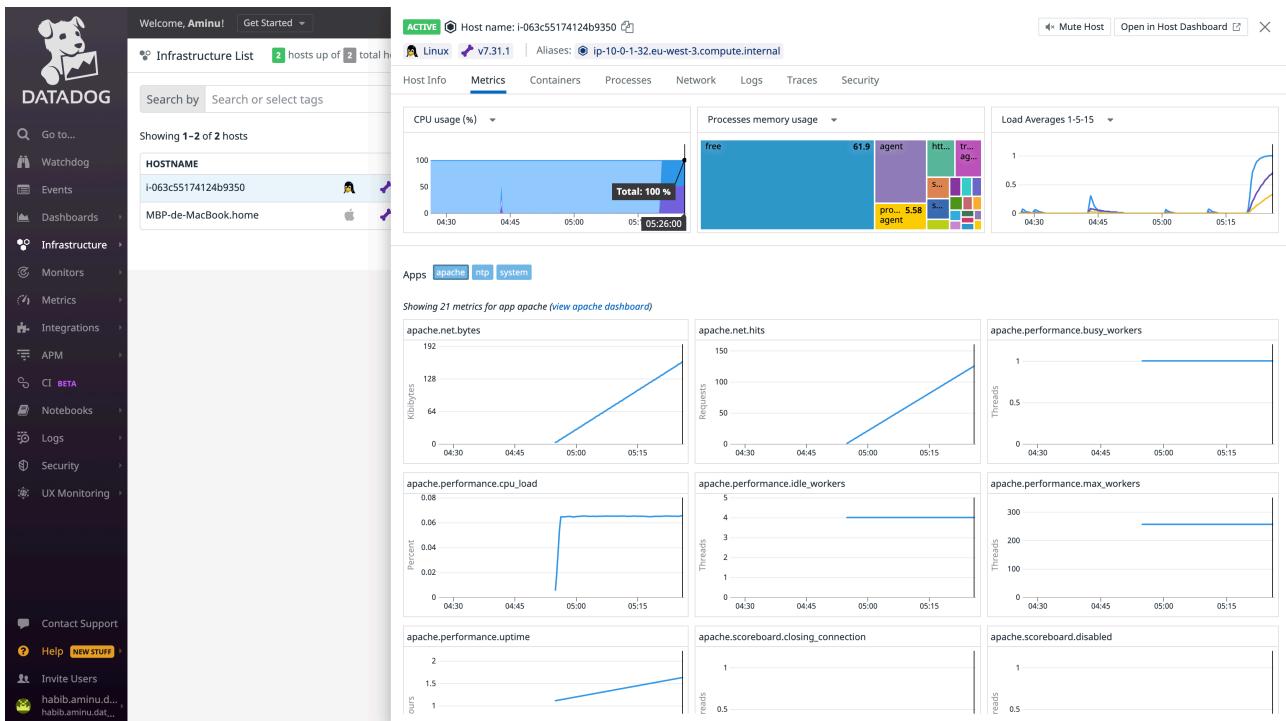


fig 4.16 Host Infrastructure Metrics

- The CPU is getting more load due to the previous command

```
Downloads -- root@ip-10-0-1-32:/etc/datadog-agent -- ssh -i datadogec2.pem ec2-user@52.47.101.218 — 152x38
 airflow.d          directory.d      hive.d          lighttpd.d      pan_firewall.d  statsd.d
 amazon_msk.d      disk.d          hivemq.d       linker.d       pgbouncer.d    supervisord.d
 ambari.d          dns_check.d    http_check.d  linux_proc_extras.d  php_fpm.d    system_core.d
 apache.d          docker.d       ibm_db2.d      load.d        postfix.d    system_swap.d
 avi_vantage.d     druid.d        ibm_mq.d      mapr.d        postgres.d   systemd.d
 azure_iot_edge.d  eks_fargate.d  ibm_was.d     mapreduce.d   powerdns_recursor.d  tcp_check.d
 btrfs.d          eks_fargate.d  ignite.d      marathon.d   presto.d     tcp_queue_length.d
 cacti.d          elastic.d      io.d          marklogic.d  process.d    teamcity.d
 cassandra.d      envoy.d        istio.d       mcache.d     prometheus.d  tenable.d
 cassandra_nodetool.d  etcd.d      jboss_wildfly.d  memory.d    proxysql.d  tls.d
 ceph.d          external_dns.d  jetson.d     mesos_master.d  rabbitmq.d  tomcat.d
 cilium.d         file_handle.d  jmx.d        mesos_slave.d  redisdb.d   twemproxy.d
 cisco_aci.d      flink.d        journald.d   mongo.d      rethinkdb.d  twistlock.d
 clickhouse.d     fluentd.d     kafka.d       mysql.d      riak.d      uptime.d
 cloud_foundry_api.d  gearmand.d  kafka_consumer.d  nagios.d    riakcs.d    varnish.d
 cockroachdb.d    gitlab.d      kong.d       network.d   sop_hana.d  vault.d
 confluent_platform.d  gitlab_runner.d  kube_apiversion_metrics.d  nfsstat.d  scylla.d    vertical.d
 consul.d         glusterfs.d    kube_controller_manager.d  nginx.d    sidekiq.d   voltdb.d
 containerd.d     go_metro.d    kube_dns.d    nginx_ingress_controller.d  redisdb.d  vsphere.d
 coredns.d        go_exvar.d    kube_metrics_server.d  ntp.d      snmp.d      yarn.d
 couch.d          gunicorn.d    kube_proxy.d  oom_kill.d    snowflake.d  zk.d
 couchbase.d     haproxy.d     kube_scheduler.d  openldap.d   solr.d      spark.d
 cpu.d          harbor.d      kubelet.d    openmetrics.d
 [root@ip-10-0-1-32 conf.d]# cd ..
 [root@ip-10-0-1-32 datadog-agent]# ls
 auth_token_compliance.d  datadog.yaml      install_info      security-agent.yaml.example  system-probe.yaml.example
 checks.d  conf.d          datadog.yaml.example  runtime-security.d  selinux
 [root@ip-10-0-1-32 datadog-agent]# cd datadog.yaml
 bash: cd: datadog.yaml: Not a directory
 [root@ip-10-0-1-32 datadog-agent]# vi datadog.yaml
 [root@ip-10-0-1-32 datadog-agent]# systemctl stop datadog-agent
 [root@ip-10-0-1-32 datadog-agent]# systemctl start datadog-agent
 [root@ip-10-0-1-32 datadog-agent]# dd if=/dev/zero of=/dev/null
```

fig 4.17 Terminal

- Click **Dashboard** to get details about the cloud host

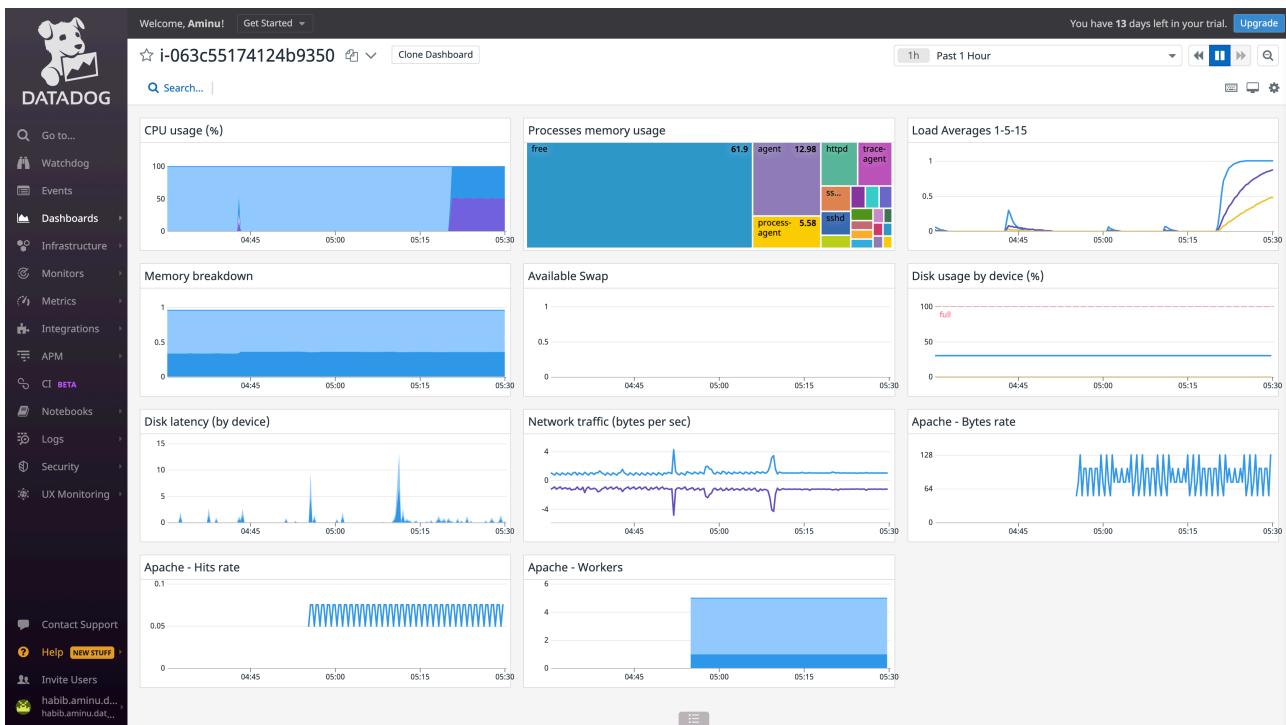


fig 4.18 Host Dashboard

- Click **Logs > Get Started**
- Select the source of your logs

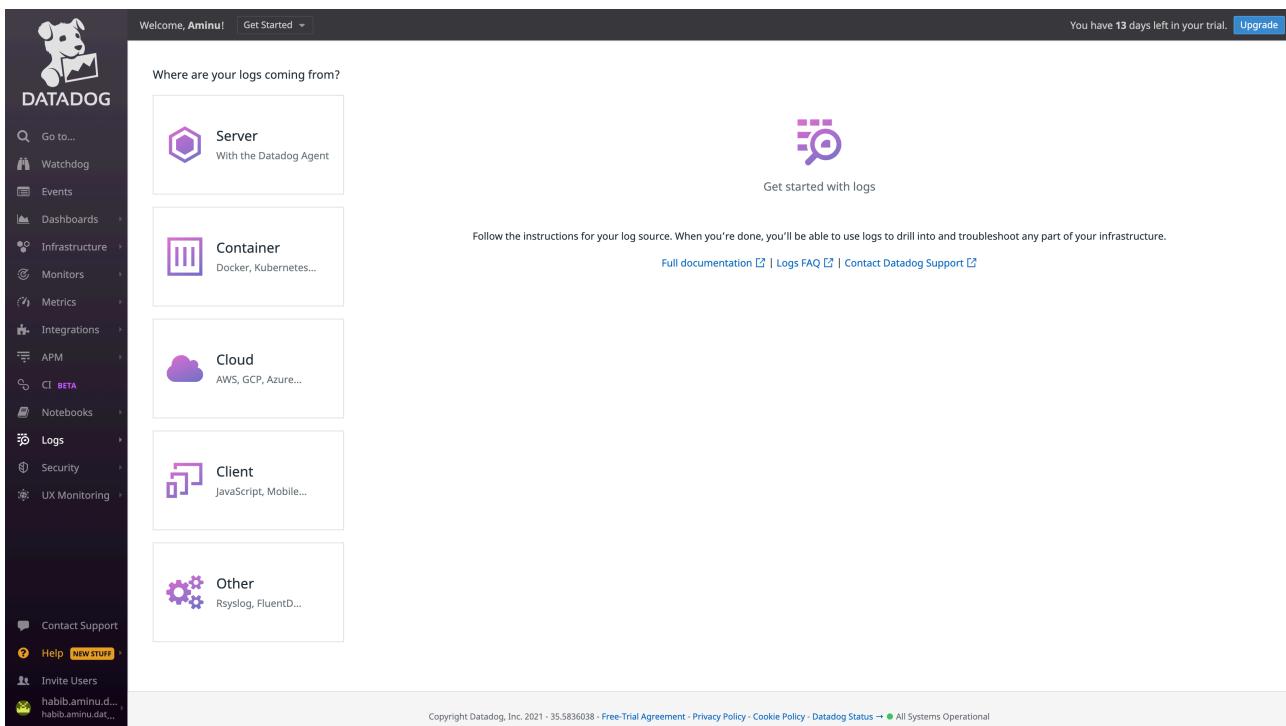


fig 4.19 Host Logs

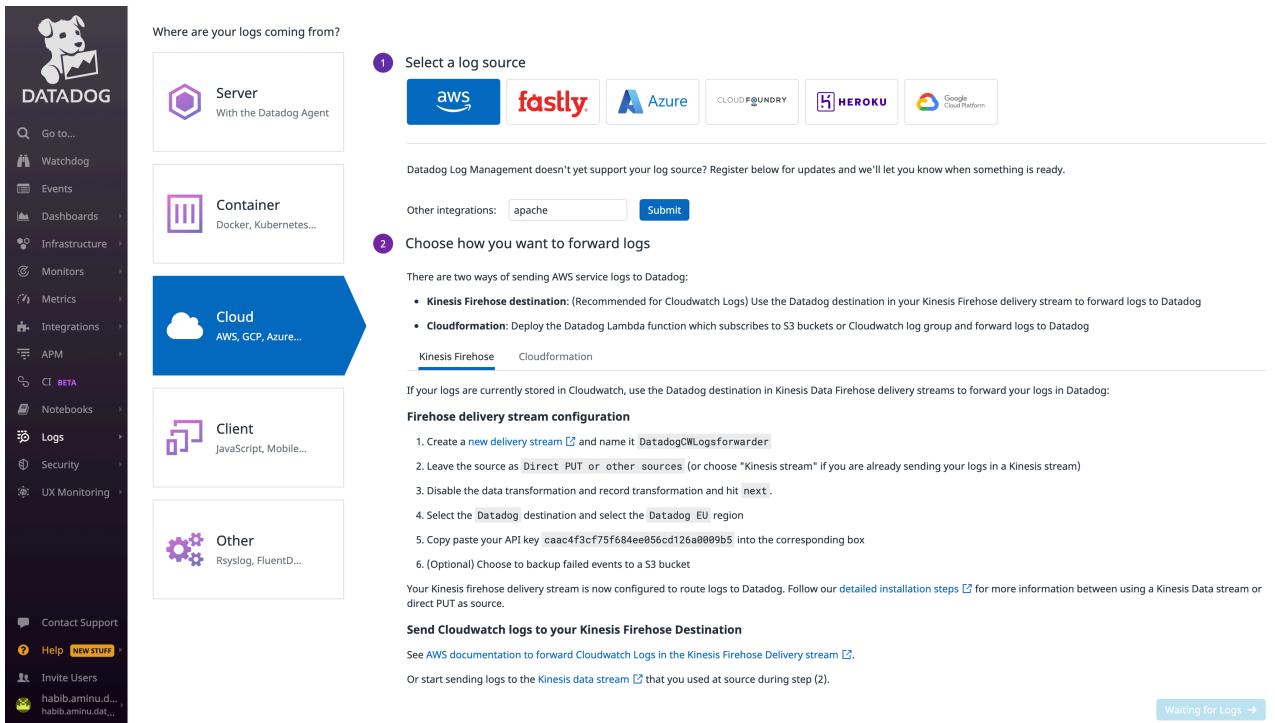


fig 4.20 AWS Logs Configuration

- Click Monitor

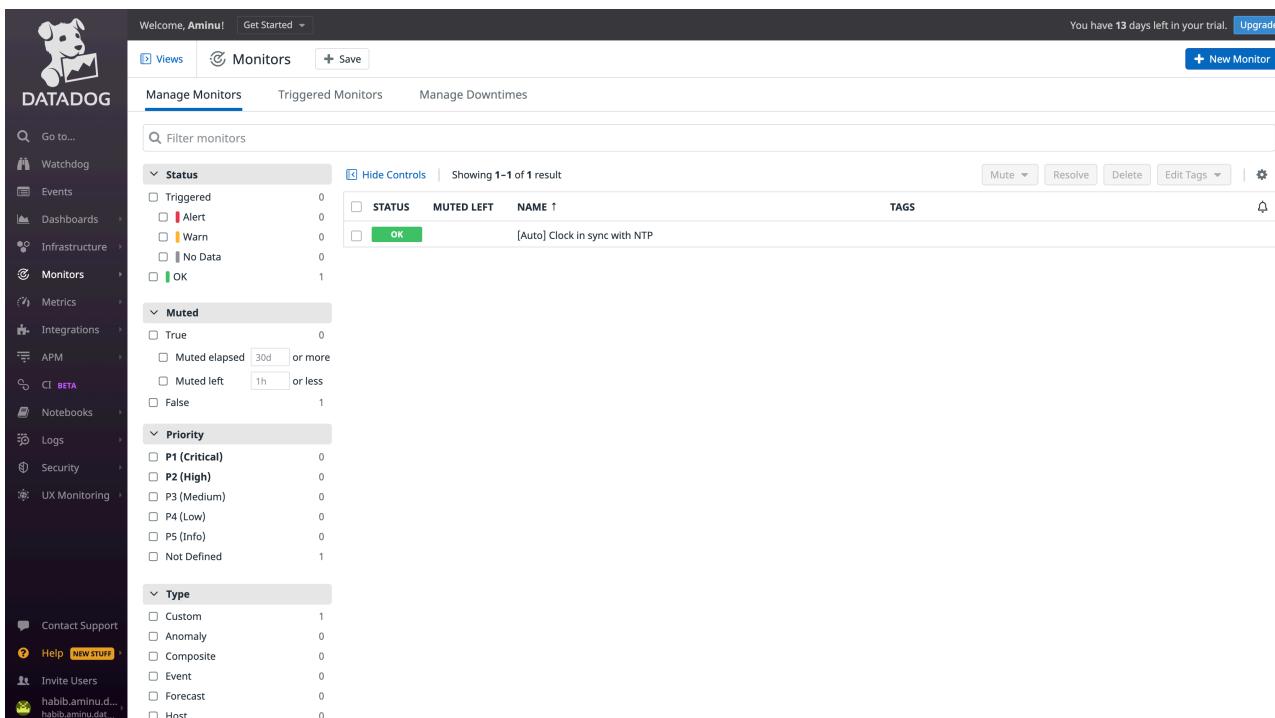


fig 4.21 Host Monitor

- Click **New Monitor** to create a new customized monitor for the host machine
- Click **Host Monitor type** > Choose the host to be monitored and set it up

Manage Monitors Triggered Monitors Manage Downtime

New Monitor / Host

1. Pick hosts by name or tag

host:063c55174124b9350 excluding Select tags to exclude

2. Set alert conditions

Check Alert Cluster Alert An alert is triggered when a host stops reporting. [?](#)

Trigger a separate alert for each host reporting your check. [?](#)

Notify if data is missing for more than 2 minutes. [?](#)

[Never] automatically resolve this event from a no data state. [?](#)

For new groups, wait 15 seconds before evaluating this monitor [?](#)

3. Notify your team

@habib.aminu.datacloud@gmail.com

Do not notify alert recipients when this alert is modified [?](#)

If this monitor stays in Select status renotify every Select time frame [?](#)

4. Say what's happening

Include triggering tags in notification title

Edit [?](#) Preview [?](#)

Markdown Help | [Use Message Template Variables](#)

Markdown Help | [Use Message Template Variables](#)

Markdown Help | [Use Message Template Variables](#)

Tags: Select or add related tags [?](#)

Priority: Not Defined

Select value [?](#) editing this monitor to its creator or administrators [?](#)

Test Notifications Export Monitor Save

Copyright Datadog, Inc. 2021 - 35.5836038 - [Free Trial Agreement](#) - [Privacy Policy](#) - [Cookie Policy](#) - [Datadog Status](#) [?](#) All Systems Operational

fig 4.22 Host Machine Monitor Configuration

- Click **Save** to save the host machine configuration
- Click **Monitors**

fig 4.23 Available Monitors

- Create **New Monitor** to create a monitor for the request per second coming from apache **apache.net.request_per_s** metric
- Configure the monitor with the following metrics:

The screenshot shows the Datadog interface for creating a new monitor. The left sidebar includes links for Go to..., Watchdog, Events, Dashboards, Infrastructure, Monitors (selected), Metrics, Integrations, APM, CI (BETA), Notebooks, Logs, Security, UX Monitoring, Contact Support, Help (NEW STUFF), Invite Users, and a search bar.

Monitor Configuration:

- 4. Notify your team:**
 - Alert recovery threshold: <= Optional
 - Warning recovery threshold: <= Optional
 - Require a full window of data for evaluation. [?](#)
 - Note: We highly recommend you select "Do Not Require" for sparse metrics, otherwise some evaluations will be skipped.
- 5. Say what's happening:**
 - Edit [Preview](#)
 - Markdown Help [?](#) Use Message Template Variables
 - APACHE IS HAVING SOME PROBLEM
 - @habib.aminu.datacloud@gmail.com APACHE IS HAVING SOME PROBLEM

Monitor Details:

- Delay evaluation by 0 seconds [?](#)
- Notify your team: @habib.aminu.datacloud@gmail.com
- Do not notify [?](#) alert recipients when this alert is modified [?](#)
- Do not restrict [?](#) editing this monitor to its creator or administrators [?](#)
- If this monitor stays in Select status [?](#) renotify every 0 minutes [?](#)

Tags: Select or add related tags [?](#)

Priority: Not Defined

Buttons: Test Notifications, Export Monitor, Save

Copyright Datadog, Inc. 2021 - 35.5836038 - [Free-Trial Agreement](#) - [Privacy Policy](#) - [Cookie Policy](#) - [Datadog Status](#) - ● All Systems Operational

fig 4.24 Apache Req/Sec Configuration Monitor

- Click **Save**
 - Create a **New Monitor** to monitor the CPU workload
 - Click **Save**

Alert recovery threshold: <input type="text" value="Optional"/>

Warning recovery threshold: <input type="text" value="Optional"/>

Require a full window of data for evaluation. [?](#)

Note: We highly recommend you select "Do Not Require" for sparse metrics, otherwise some evaluations will be skipped.

Do not notify if data is missing. [?](#)

Note: we recommend the missing data window be at least 2x the evaluation period above

[Never] automatically resolve this event from a triggered state. [?](#)

Delay evaluation by seconds [?](#)

4 **Notify your team**

[X](#)

Do not notify alert recipients when this alert is modified [?](#)

Do not restrict editing this monitor to its creator or administrators [?](#)

If this monitor stays in renotify every [?](#)

5 **Say what's happening**

[Edit](#) [Preview](#)

CPU load is very HIGH

CPU load is very HIGH @habib.aminu.datacloud@gmail.com

[Markdown Help](#) | [Use Message Template Variables](#)

```
(6/20): httpd-1.3.42-2.51-1.0rn2.x86_64.rpm | 88 kB 00:00:00
(7/20): httpd-2.4.51-1.0rn2.x86_64.rpm | 1.3 kB 00:00:00
(8/20): mod_http2-1.15-19-1.0rn2.0.1.x86_64.rpm | 149 kB 00:00:00
(9/20): mailcap-2.1.41-2.0rn2.noarch.rpm | 31 kB 00:00:00
```

Total 7.0 MB/s | 1.9 MB 00:00:00

fig 4.25 CPU Workload Monitor



Welcome, Aminu! Get Started ▾ You have 13 days left in your trial. Upgrade

Views Monitors + Save

Manage Monitors Triggered Monitors Manage Downtimes

Filter monitors

Status

- Triggered 1
 - Alert 0
 - Warn 0
 - No Data 1
- OK 3

Muted

- True 0
 - Muted elapsed 30d or more
 - Muted left 1h or less
- False 4

Priority

- P1 (Critical) 0
- P2 (High) 0
- P3 (Medium) 0
- P4 (Low) 0
- P5 (Info) 0
- Not Defined 4

Type

- Custom 1
- Host 1
- Integration 1
- Metric 1
- Anomaly 0
- Composite 0
- Event 0

APACHE IS HAVING SOME PROBLEM

CPU load is very HIGH

[Auto] Clock in sync with NTP

'{{ node.name }} is NOT doing well'

Mute ▾ Resolve Delete Edit Tags ▾

Watchdog Events Dashboards Infrastructure Metrics Integrations APM CI BETA Notebooks Logs Security UX Monitoring

Contact Support Help NEW STUFF

Invite Users habib.aminu.d... habib.aminu.dat...

fig 4.26 Available Monitors

- Click **Dashboard** to create a new dashboard
- Select **New Dashboard**
- Add **widget**
- Drag the widget you would love to see on the dashboard e.g **Query Value** with a CPU workload metric

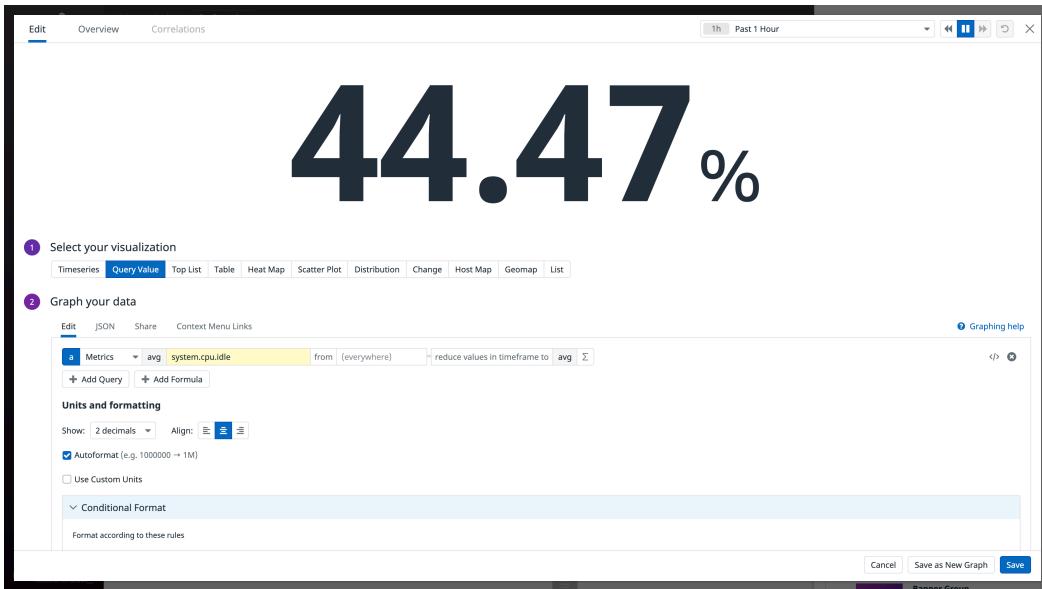


fig 4.27 CPU Workload Query Value

- Click **Save**
- Drag another **Query Value** to see the apache request **apache.net.request_per_s**

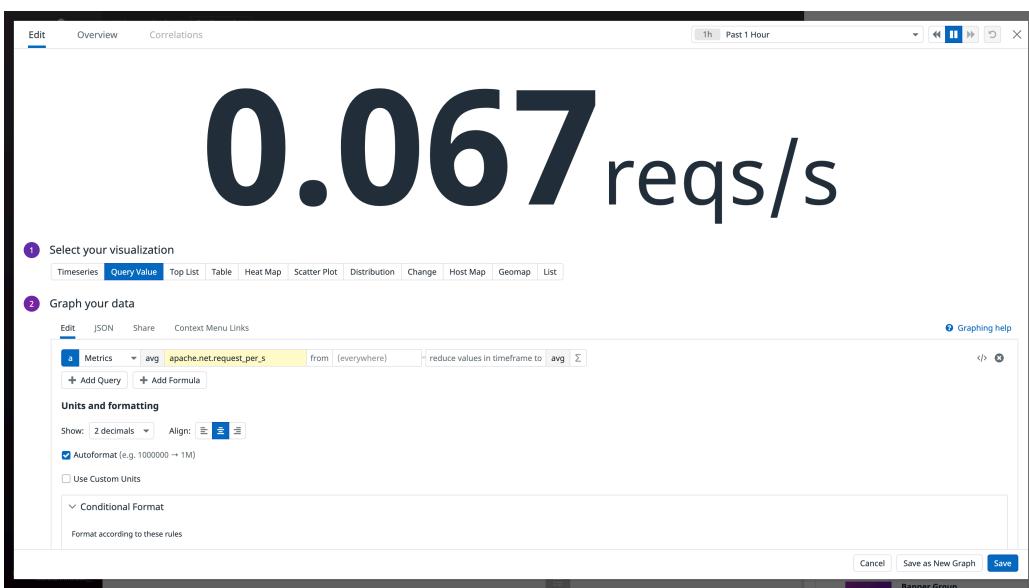


fig 4.28 Apache Req/Sec Query Value Dashboard

- Drag **Time series** to see the memory available memory on the host machine



fig 4.29 Available Host Memory Time Series Dashboard

- Drag **Top List**

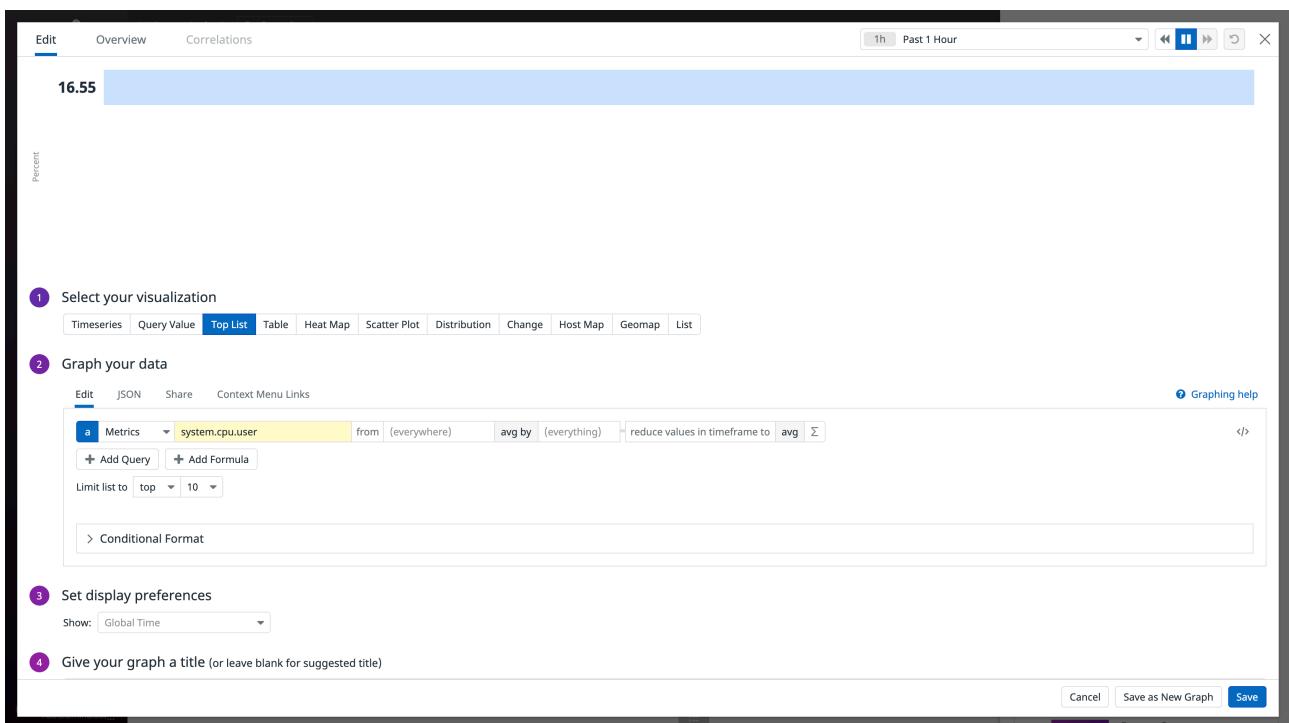


fig 4.30 CPU User

- Click Save

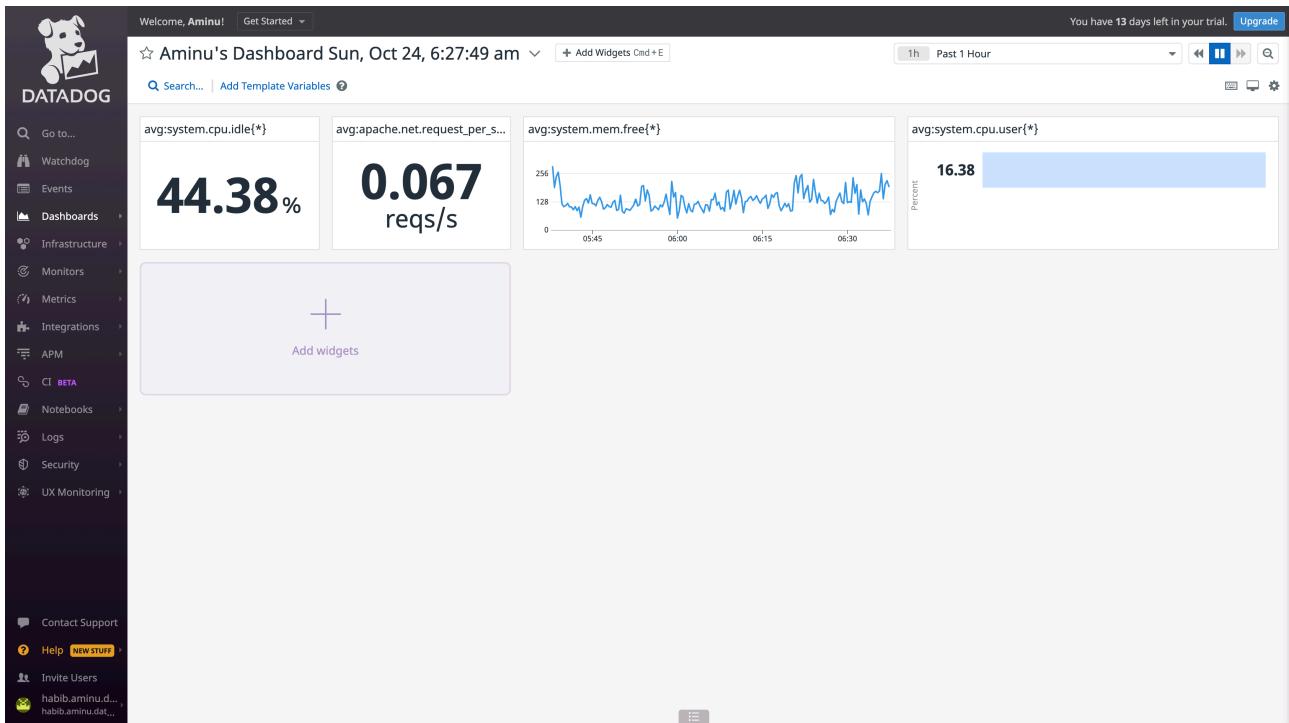


fig 4.31 Dashboard

- We have managed to create monitors, logs, dashboard, and alerts for better understanding of our infrastructure, and easy troubleshooting due to the available datadog services.

CHAPTER 5:

5.1 Customer Questions: Integration Not Working

I appreciate you letting me know about the issue! I definitely will make sure that it gets sorted.

This output is from the diagnose command which checks connectivity with various components that the agent can interact with.

These failures are expected if you don't run on all of these platforms, and can be safely ignored.

Regarding cloud providers (azure, gce, aws) it tries to query their local metadata api and once that failed it stops trying to guess the cloud provider.

These failures only show up when you run `./agent diagnose` right? This command is explicitly for debugging purpose to troubleshoot connectivity issues. Failures are fine.

If you just installed the Agent, it may take a few moments before you start seeing metrics appear. The first place you should check for metrics is the [Metrics Explorer](#).

You may have already tried these, but can you just confirm you've gone through the following troubleshooting steps:

1. Is your host connected to the internet or able to access it through a proxy?
2. If using a proxy: is your Agent configured for this proxy?
3. Is the Datadog API key set up in your `datadog.yaml` configuration file the API key corresponding to your Datadog platform?
4. Is the site configured in your `datadog.yaml` configuration file matching the one from your organization?
5. Is there only one Datadog Agent running on your host?
6. Did you restart the Datadog Agent after editing a yaml configuration file?

If the answer to all questions above is yes, then [run the status command](#) for more details about your Agent and its integrations status. You can also check the [Agent logs](#) directly and enable debug mode to [get more logging from the Agent](#).

If you're still unsure about the issue, you may reach out to the [Datadog support team with a flare](#) from your Agent.

Datadog has a few integrations that are set up through YAML files in the Agent.

Here is a quick guide for troubleshooting getting integrations installed:

1. Run the [info command](#).
2. Is the integration showing up in the [info command](#)?

No, it's not:

1. Check the configuration file, make sure it is in the right location and named correctly.
2. Check it in a YAML parser to make sure it has the correct syntax.
3. If you moved or changed the file, [restart the Agent](#) and then rerun the info command to see if it is now showing up.

Yes, it's there:

1. Check the Metrics Explorer to see if system metrics are showing up from the host. For example, look for **system.disk.directory.files.bytes** from the host that is running the Agent and has that integration setup.
2. If there are still no metrics, check the logs for errors and send them along with the info command output to [Datadog support](#).

Thanks again for your patience!

5.1.2 Metric API Help

Thanks so much for reporting this to us. This definitely isn't what we would expect to see, so I want to get this fixed up for you quickly. I understand how frustrating this must be, so bear with me as we walk through some steps to get this fixed together.

I've gone through and checked that everything looks as it should but while looking at the error message, I noticed that the problem is with the input parameter in 'to'.

{"errors": ["The value provided for parameter 'to' is invalid"]}%

That means I need your help to investigate further.

Could you please start by:

- Trying a supported fixed date syntax and input a valid parameter in 'to'. This should help the curl query to connect to the API and get the metrics.

- Use this link https://docs.datadoghq.com/dashboards/guide/custom_time_frames/ for more details on supported syntaxes.

Can you see the metrics?

- No, go to your datadog account, if you have created a monitor for system.load.1, select the monitor and you could select or customize the particular date you would love to see the metrics.

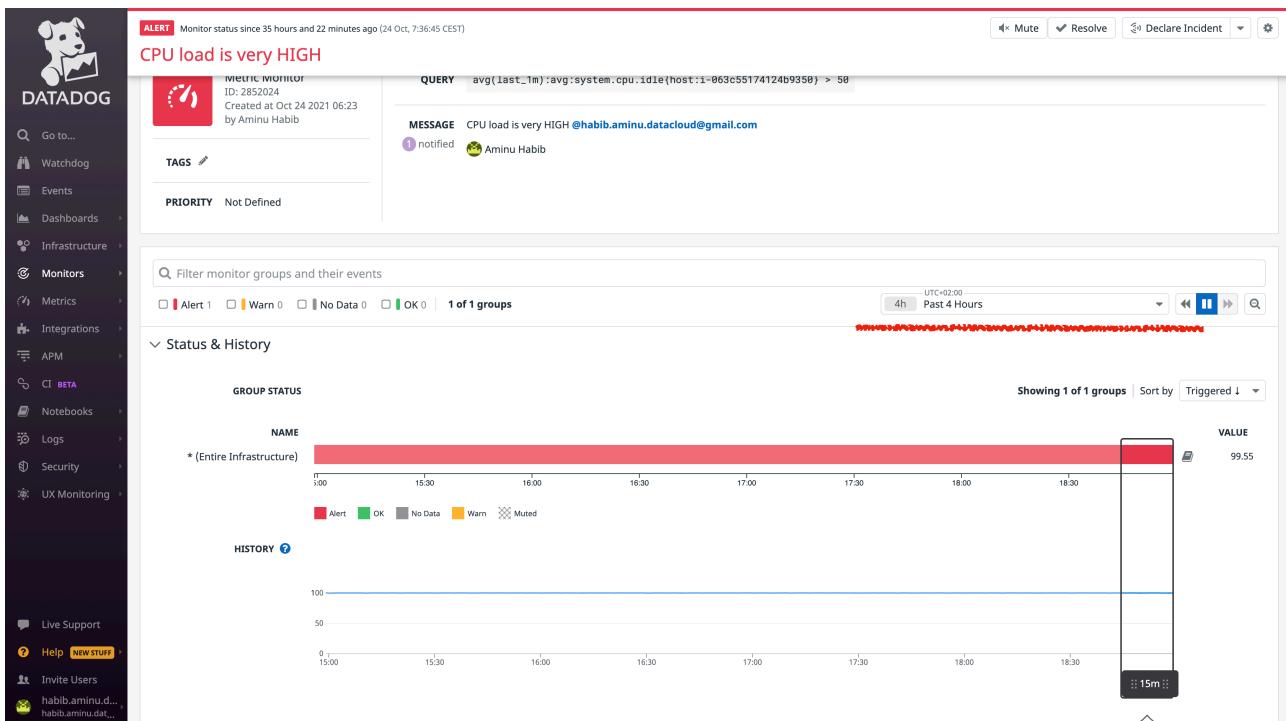


fig 5.1 CPU Metric Monitor

Let me know if this resolves the problem or if you have any questions.

Thanks again for your patience!