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.

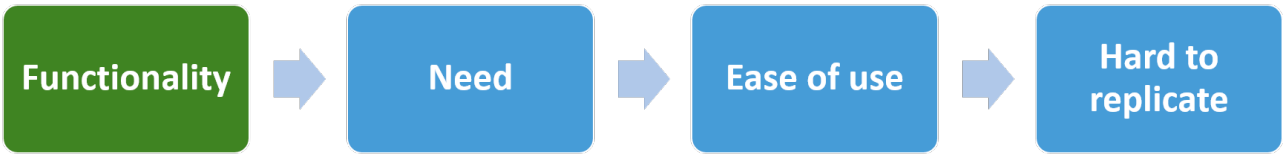
Datadog was designed as a cloud infrastructure monitoring service, with a dashboard, alerting, and visualizations of metrics.

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.

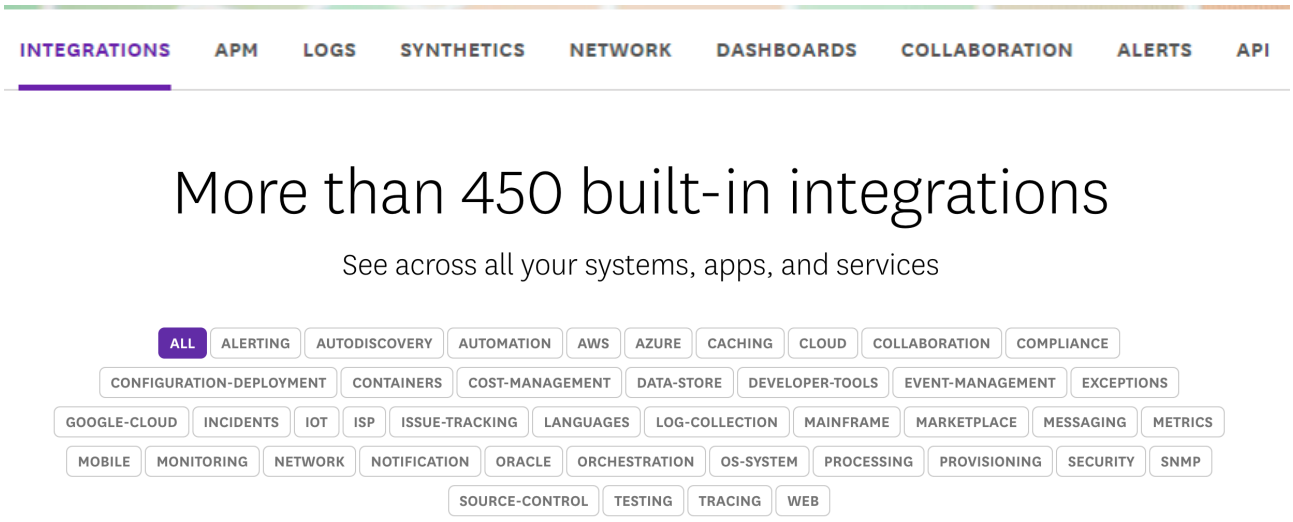
There are prebuilt dashboards for lots of things we care about. Snazzy features like autocomplete and templating make building our own dashboards easy.

- **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



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

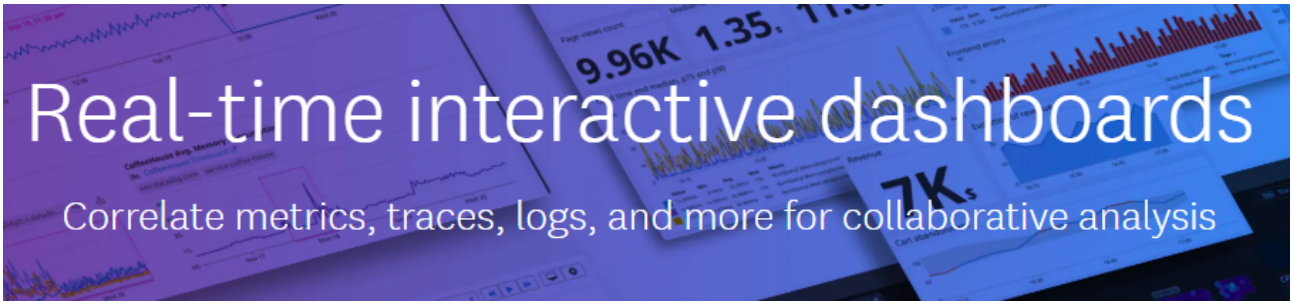


fig 1.1 Real-time Interactive Dashboards

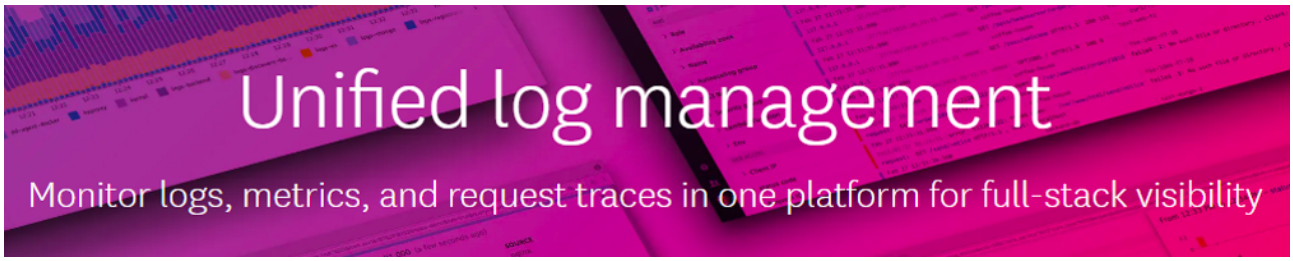


fig 1.2 Unified Log Management

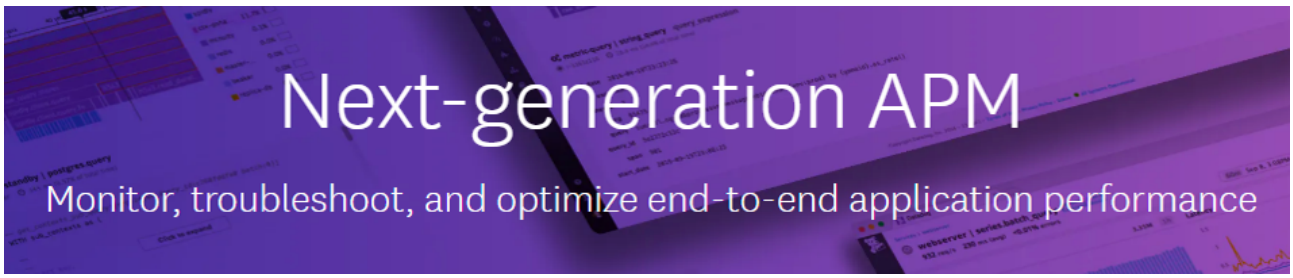


fig 1.3 Next-generation APM

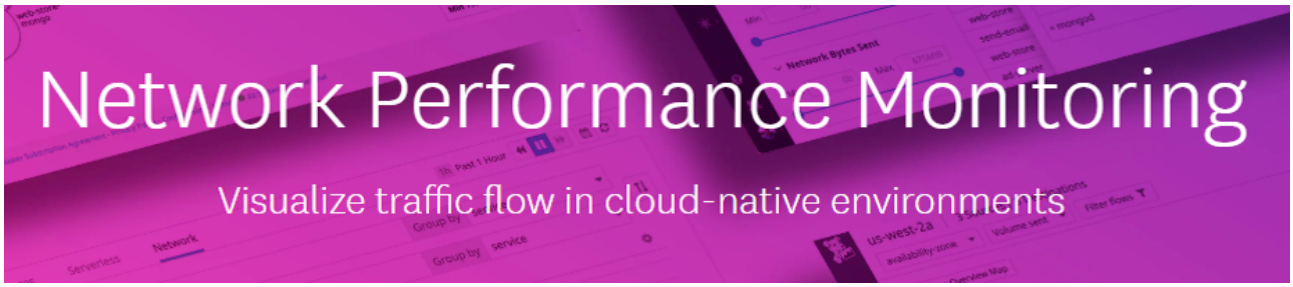


fig 1.4 Network Performance Monitoring

fig 1.5 Simplified Synthetic Monitoring

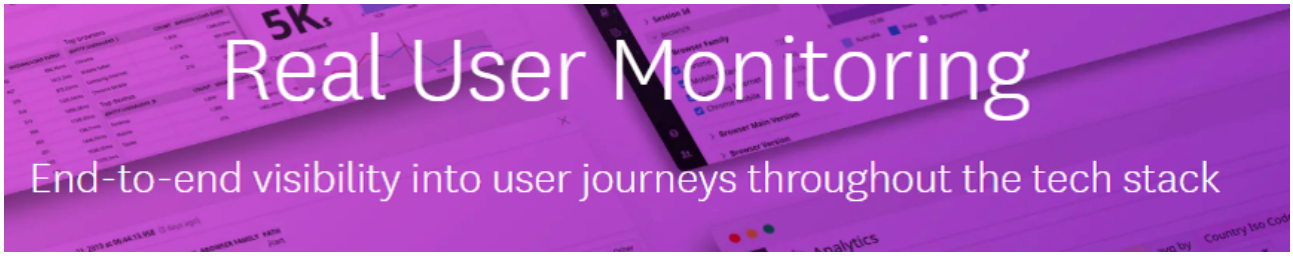


fig 1.6 Real User Monitoring

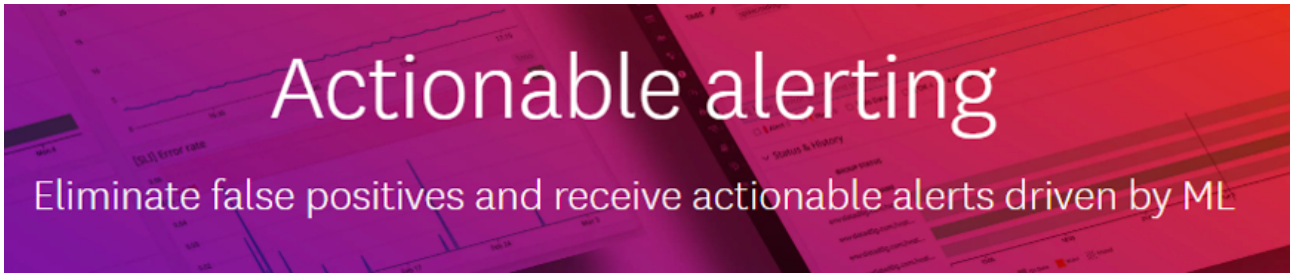


fig 1.7 Actionable alerting

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/

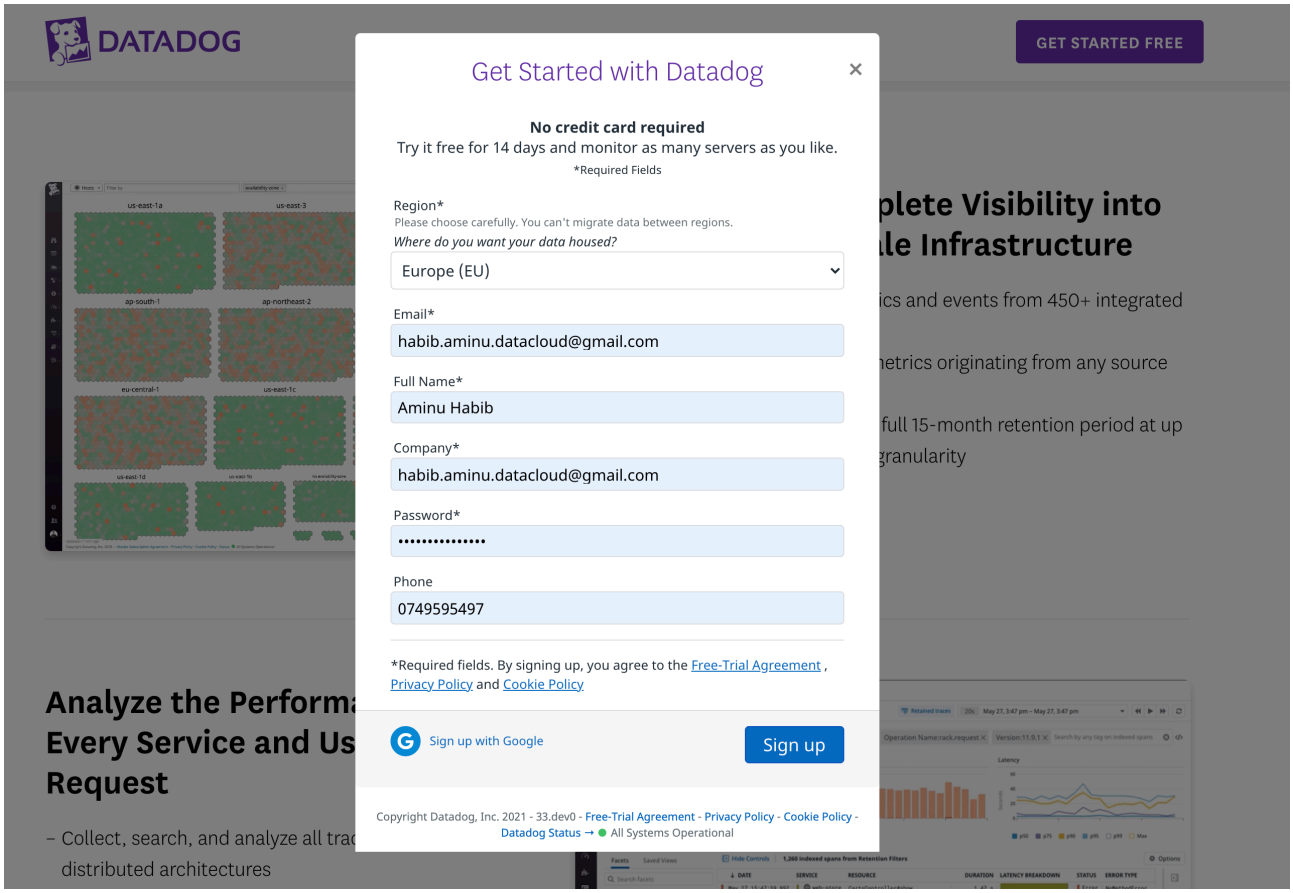


fig 2.1 Creating an account

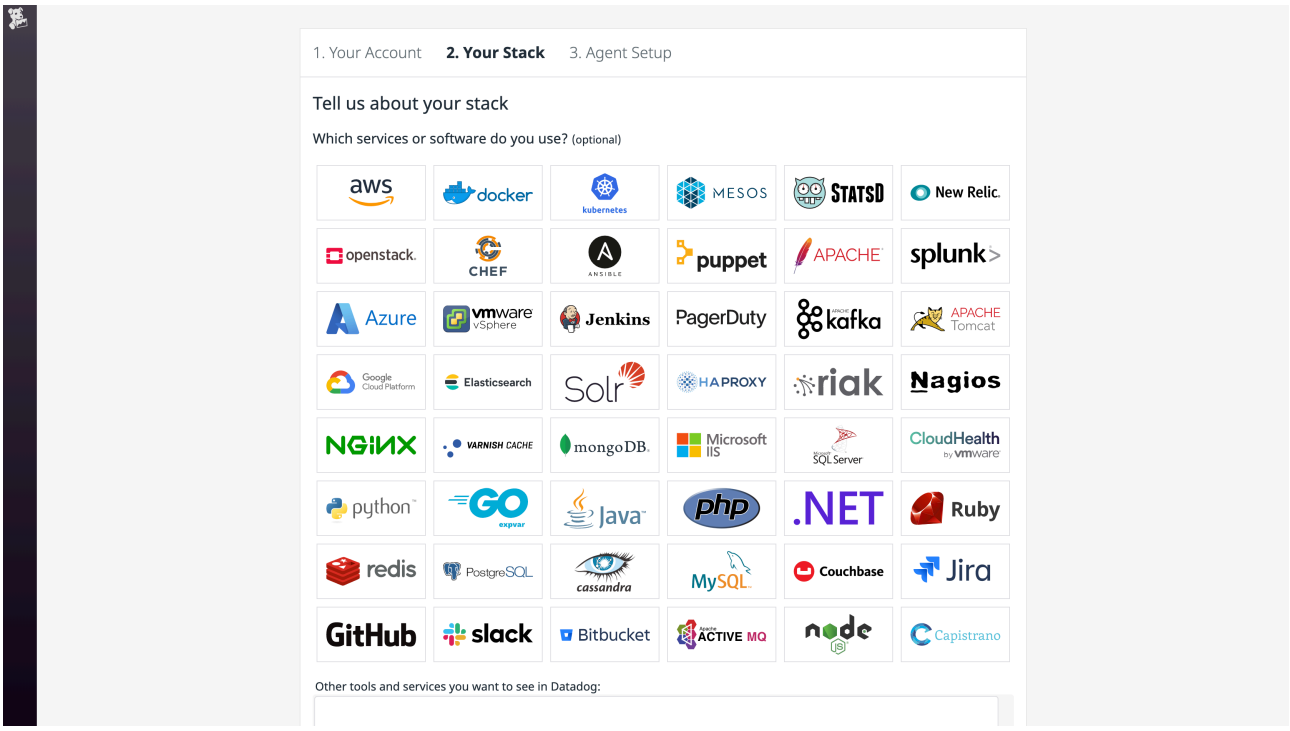


fig 2.2 Choose your stack

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?

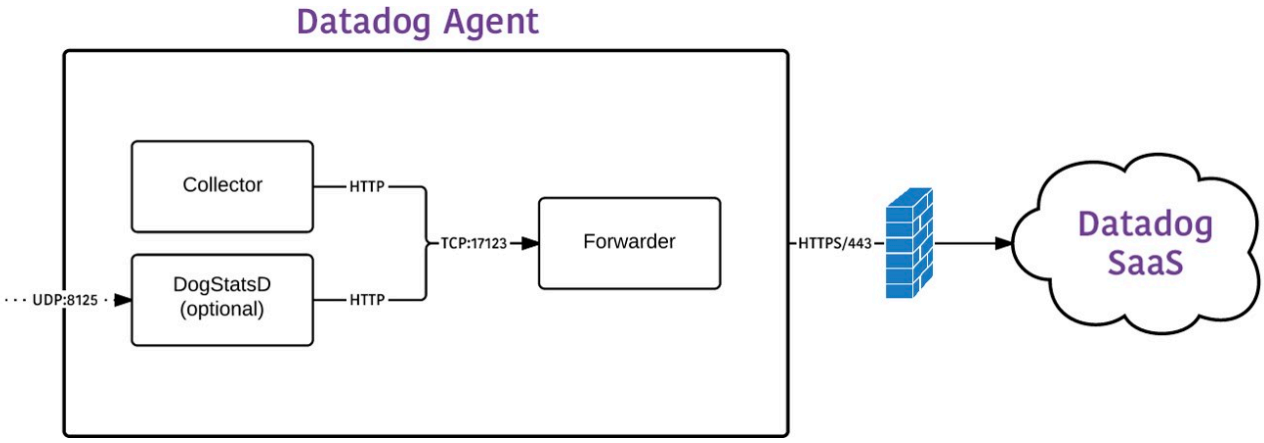


fig 2.3 Datadog Agent Architecture

2.2.1.3 Datadog Agent: Platforms

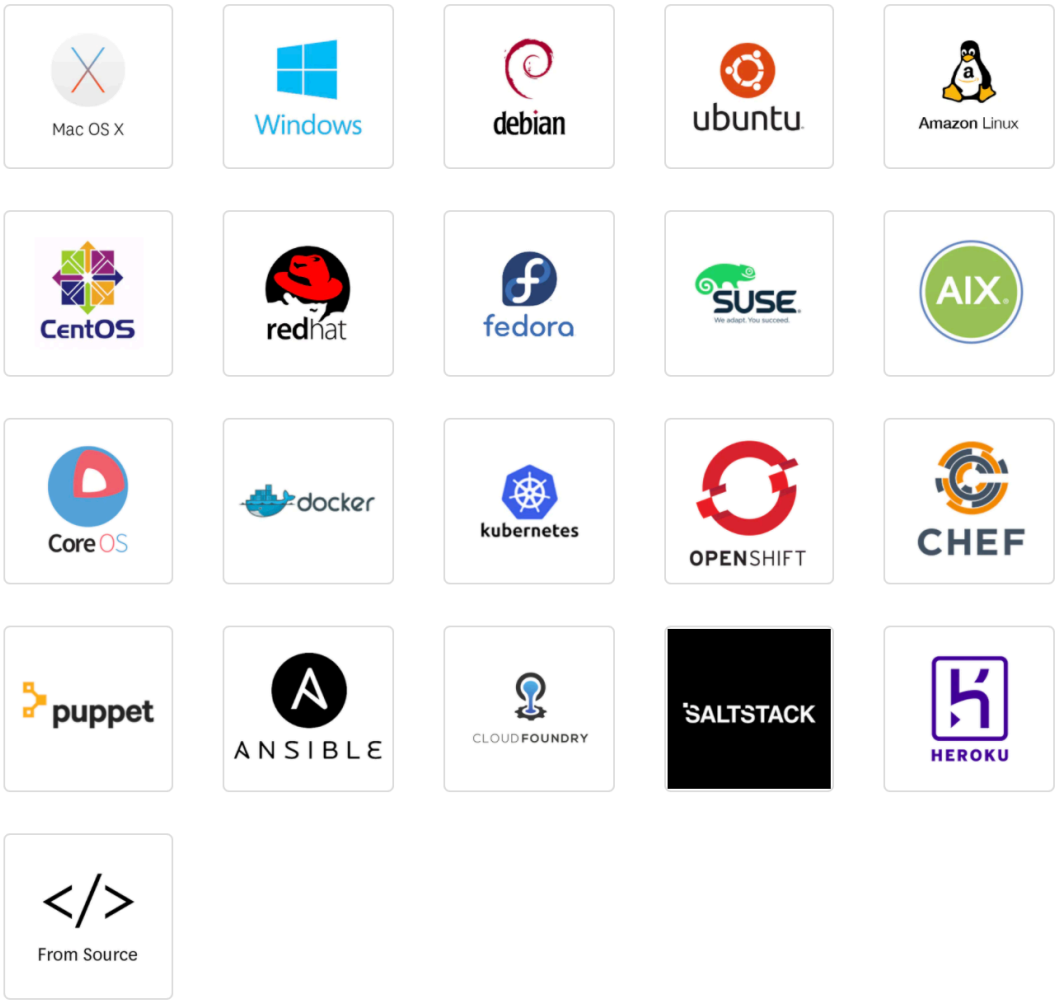


fig 2.4 Datadog Agent Platforms

2.2.2 Datadog Agent: DogStatsD

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

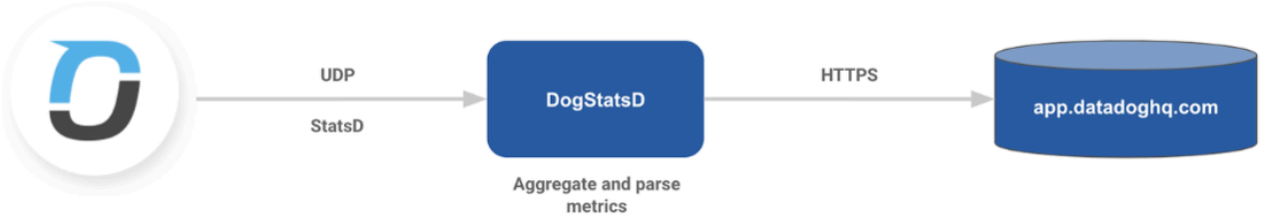


fig 2.5 DogStatsD Architecture

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).

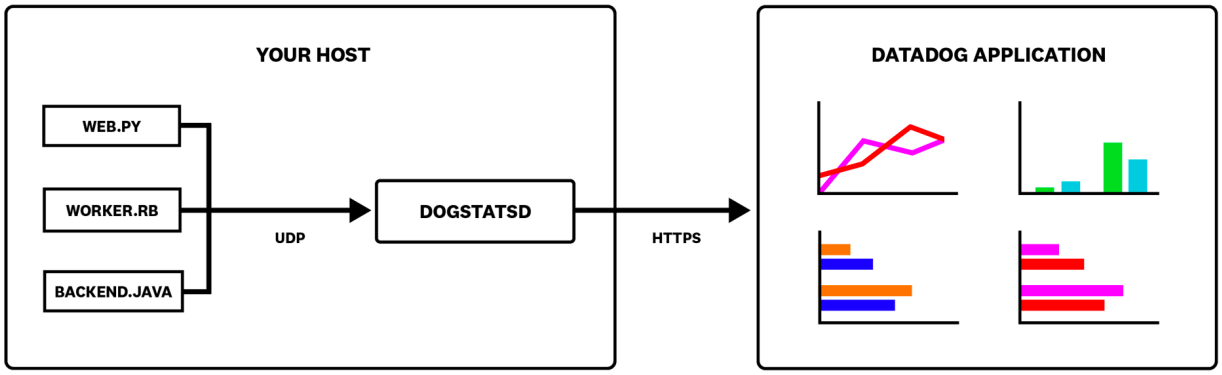


fig 2.6 DogStatsD Functional Model

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	`/etc/datadog-agent/datadog.yaml`
Linux	`/etc/datadog-agent/datadog.yaml`
CentOS	`/etc/datadog-agent/datadog.yaml`
Debian	`/etc/datadog-agent/datadog.yaml`
Fedora	`/etc/datadog-agent/datadog.yaml`
macOS	`~/.datadog-agent/datadog.yaml`
RedHat	`/etc/datadog-agent/datadog.yaml`
Source	`/etc/datadog-agent/datadog.yaml`
Suse	`/etc/datadog-agent/datadog.yaml`
Ubuntu	`/etc/datadog-agent/datadog.yaml`
Windows Server 2008, Vista and newer	`\\ProgramData\Datadog\datadog.yaml`
Windows Server 2003, XP or older	*unsupported platform*

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

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

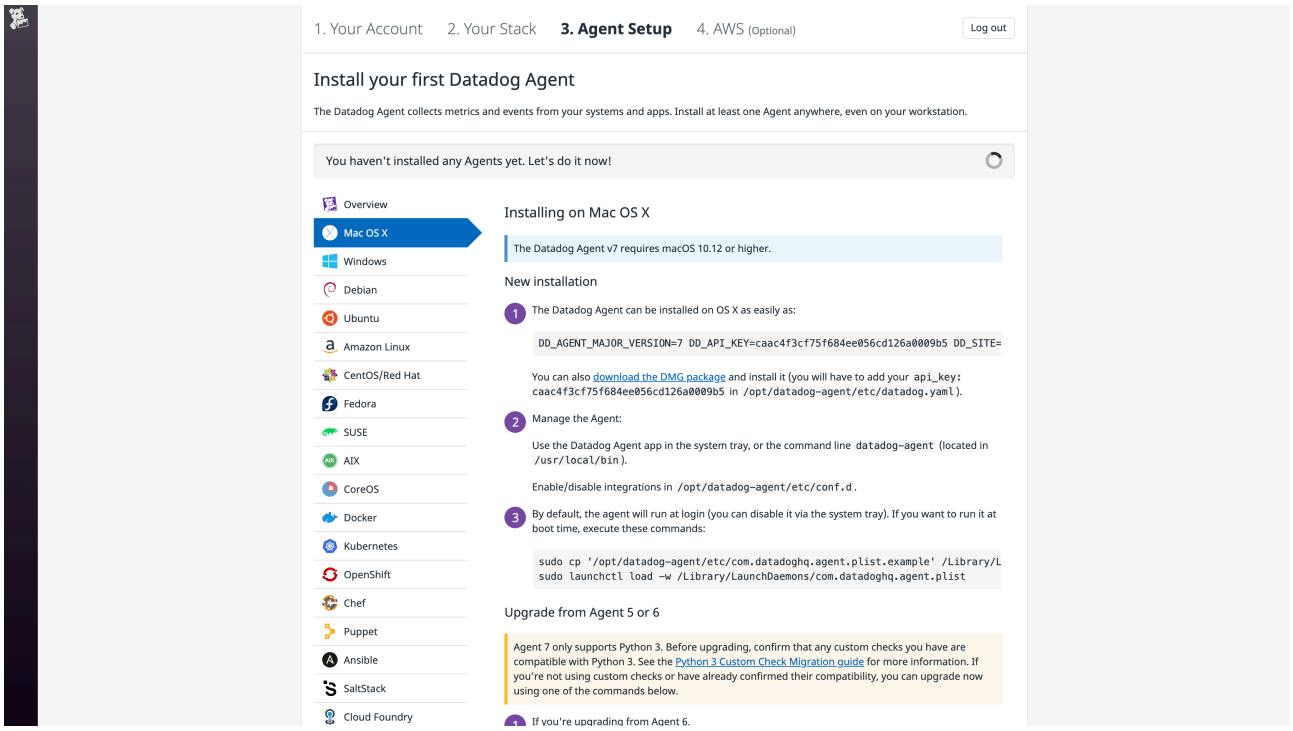


fig 3.1 Datadog agent installation

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)"

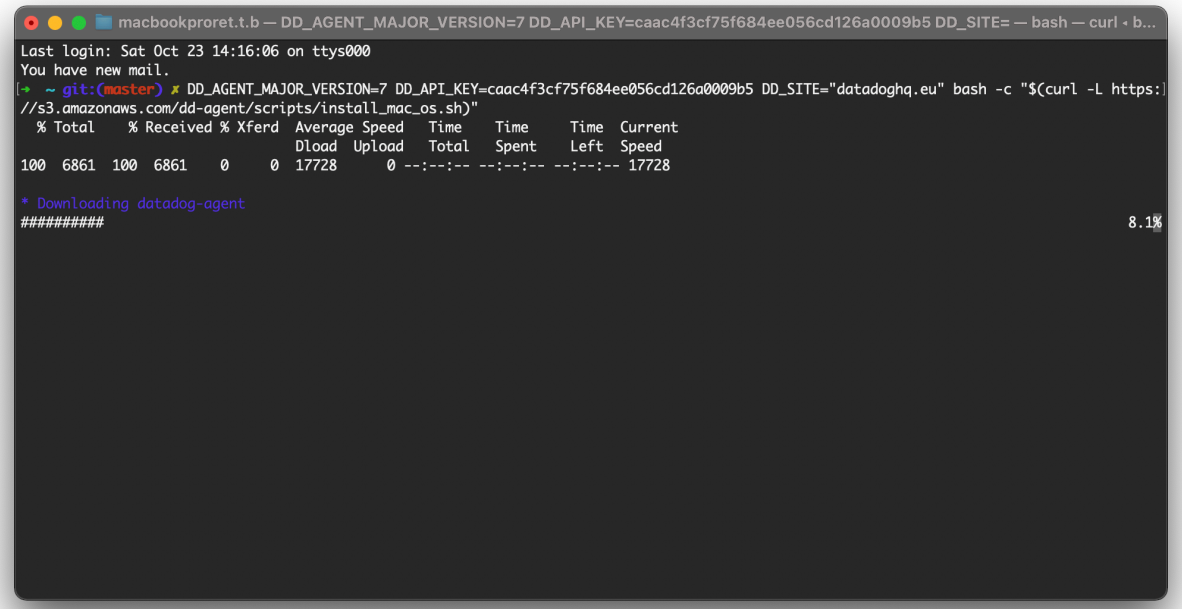


fig 3.2 Datadog Agent Installation on macOS

You might be asked for your password:

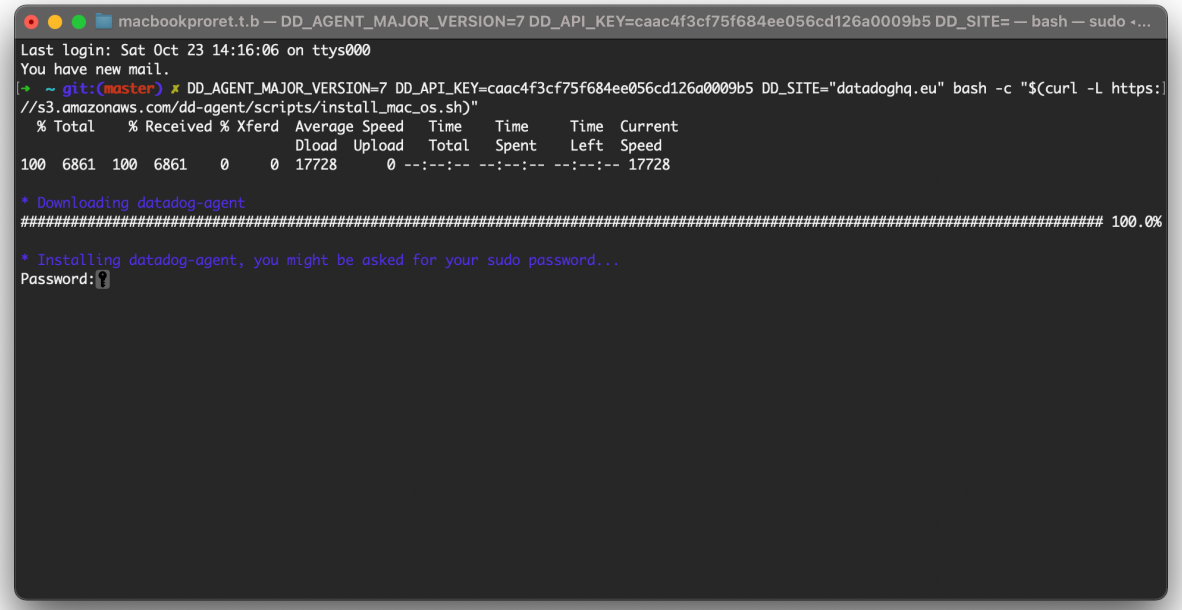


fig 3.3 Password Requirement

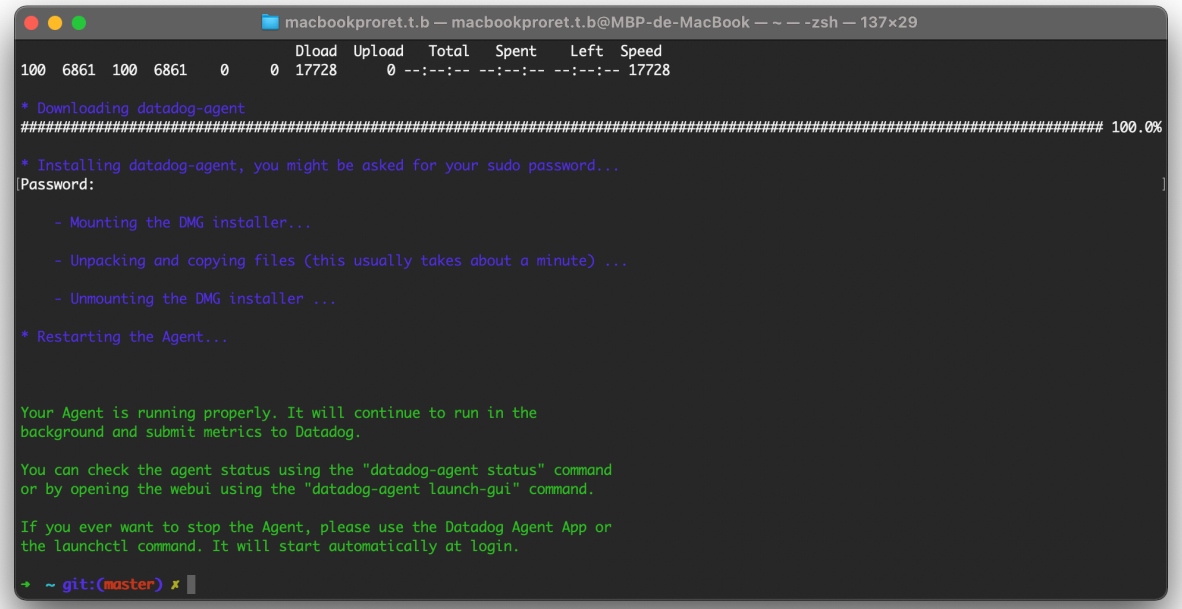


fig 3.4 Successful Installation of Datadog Agent

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

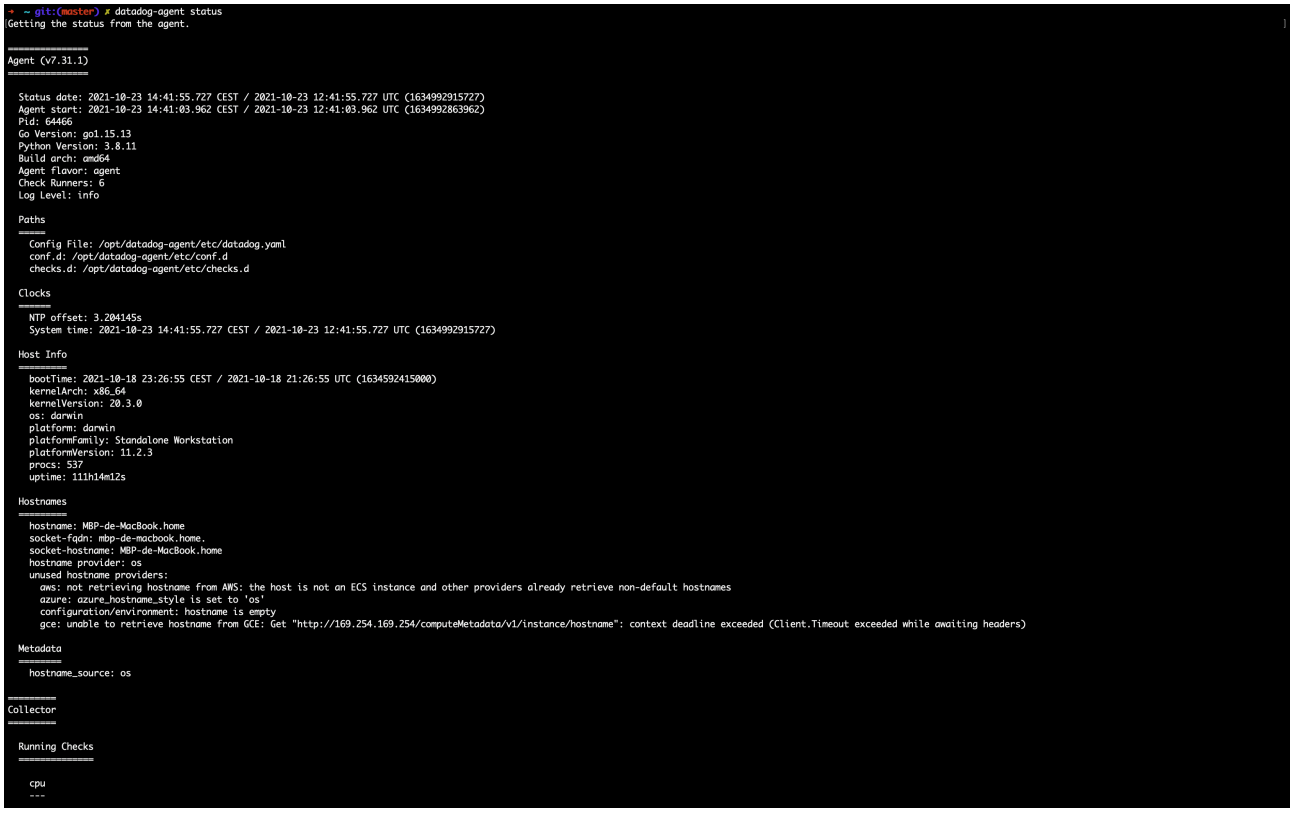


fig 3.5 Datadog-agent status

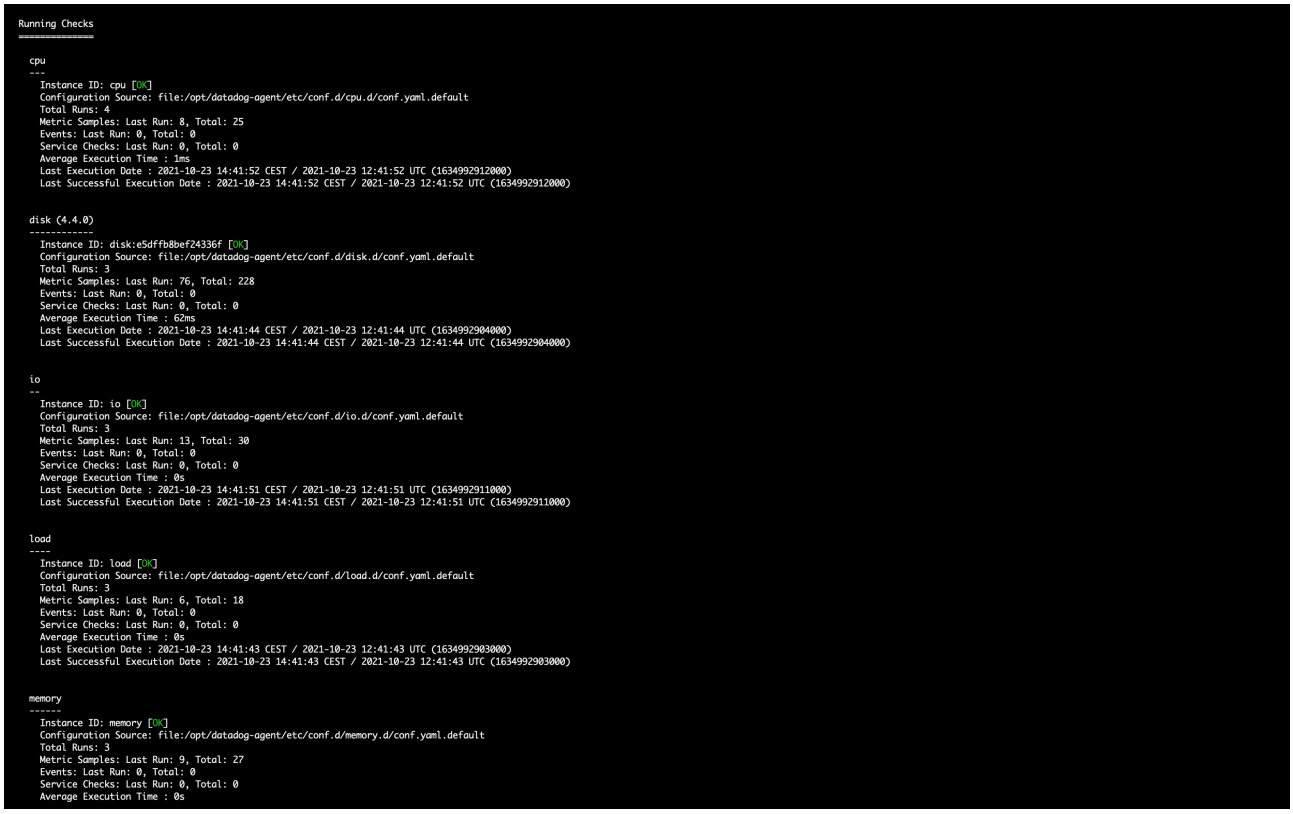


fig 3.6 Datadog-agent status details

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

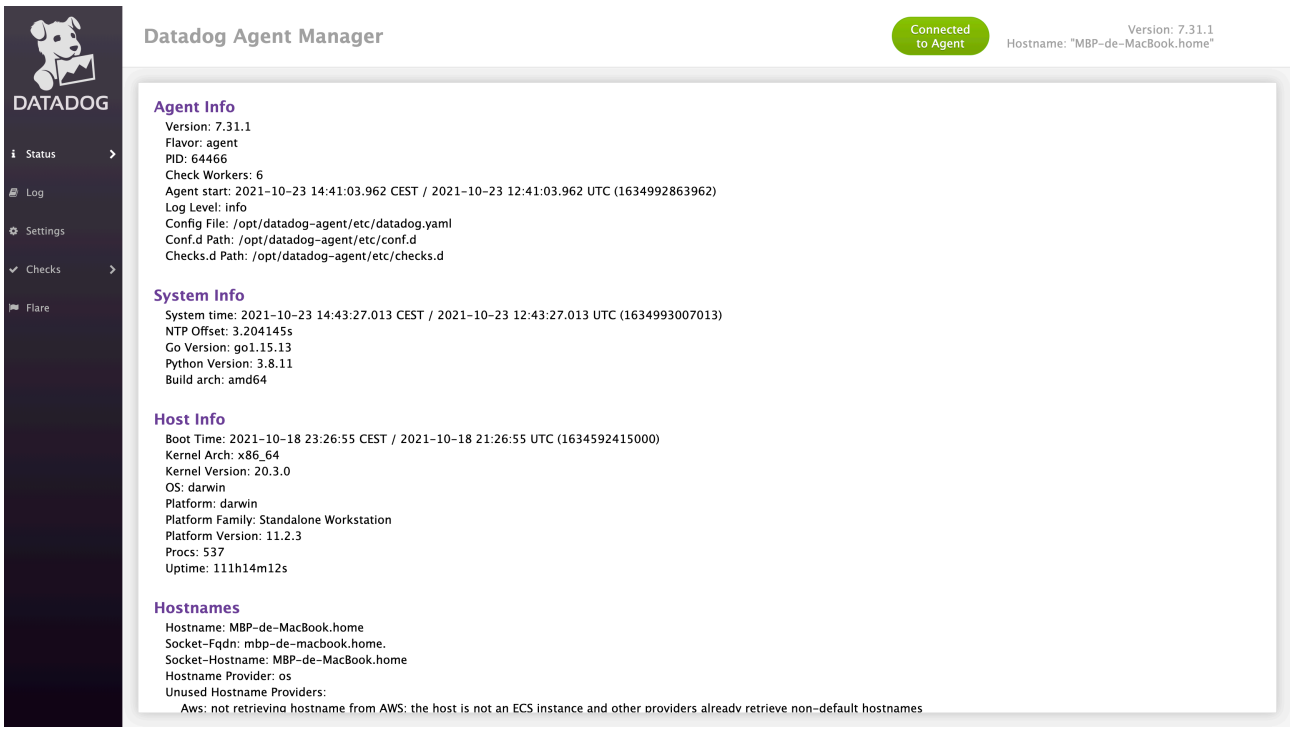


fig 3.7 Datadog Agent Manager

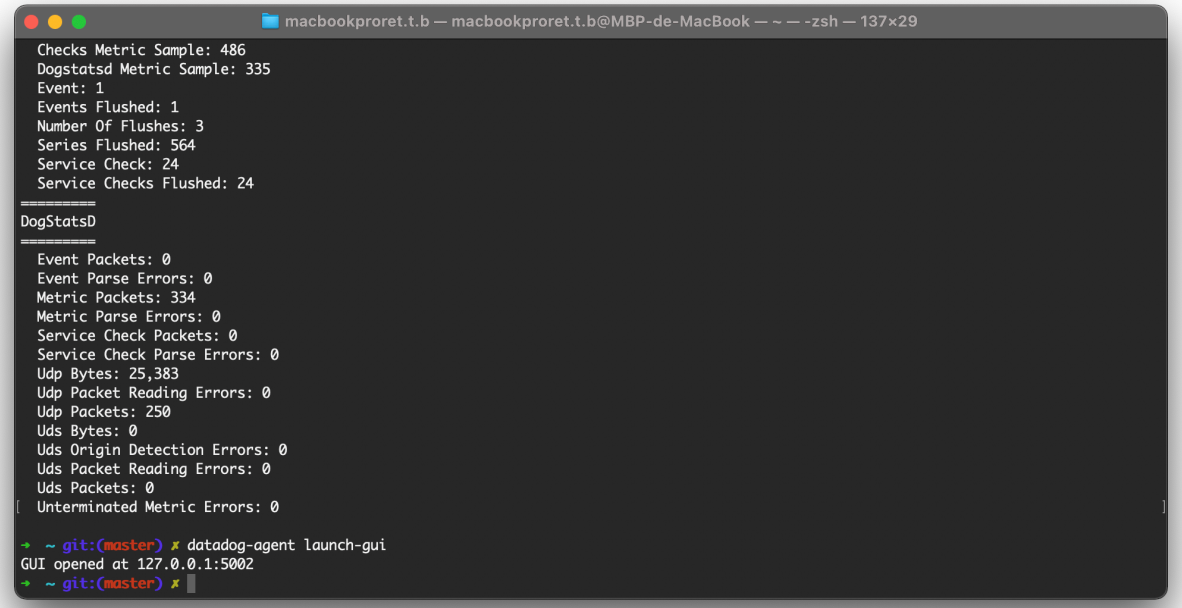


fig 3.8 Datadog-agent launch-gui

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.

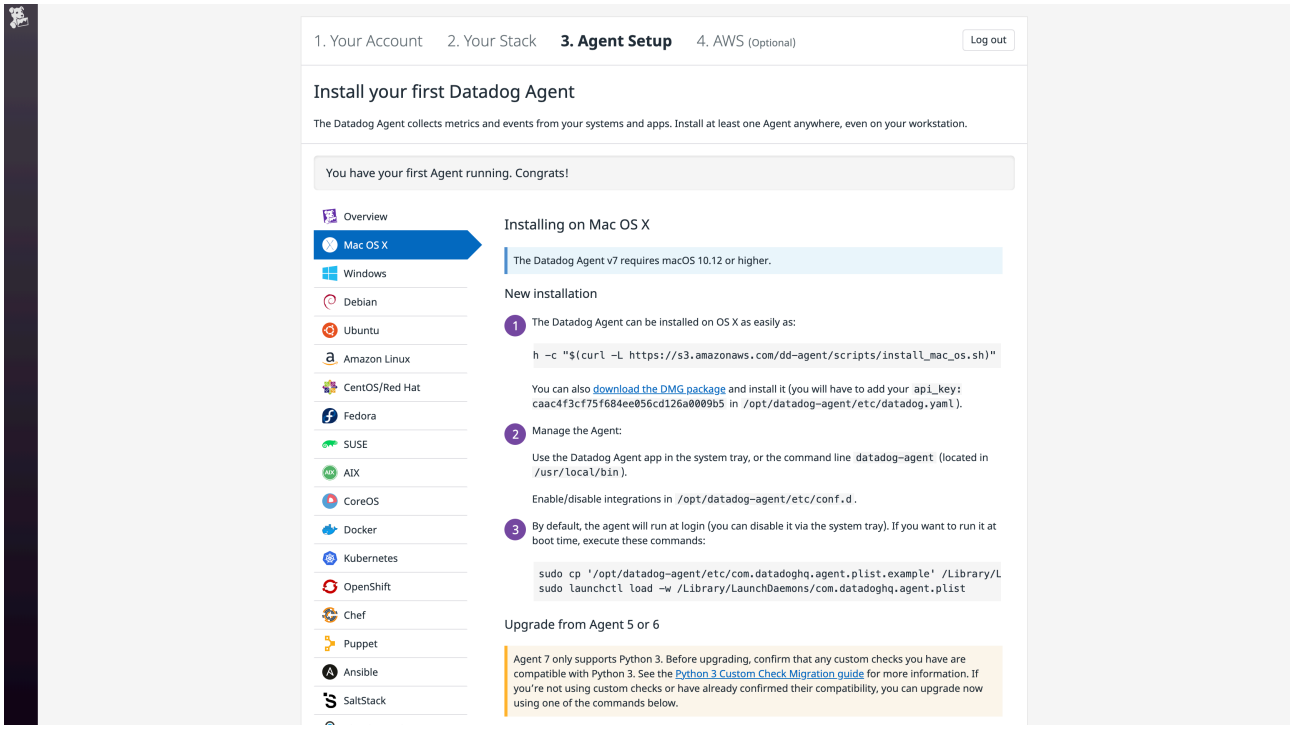


fig 3.9 Datadog-agent setup

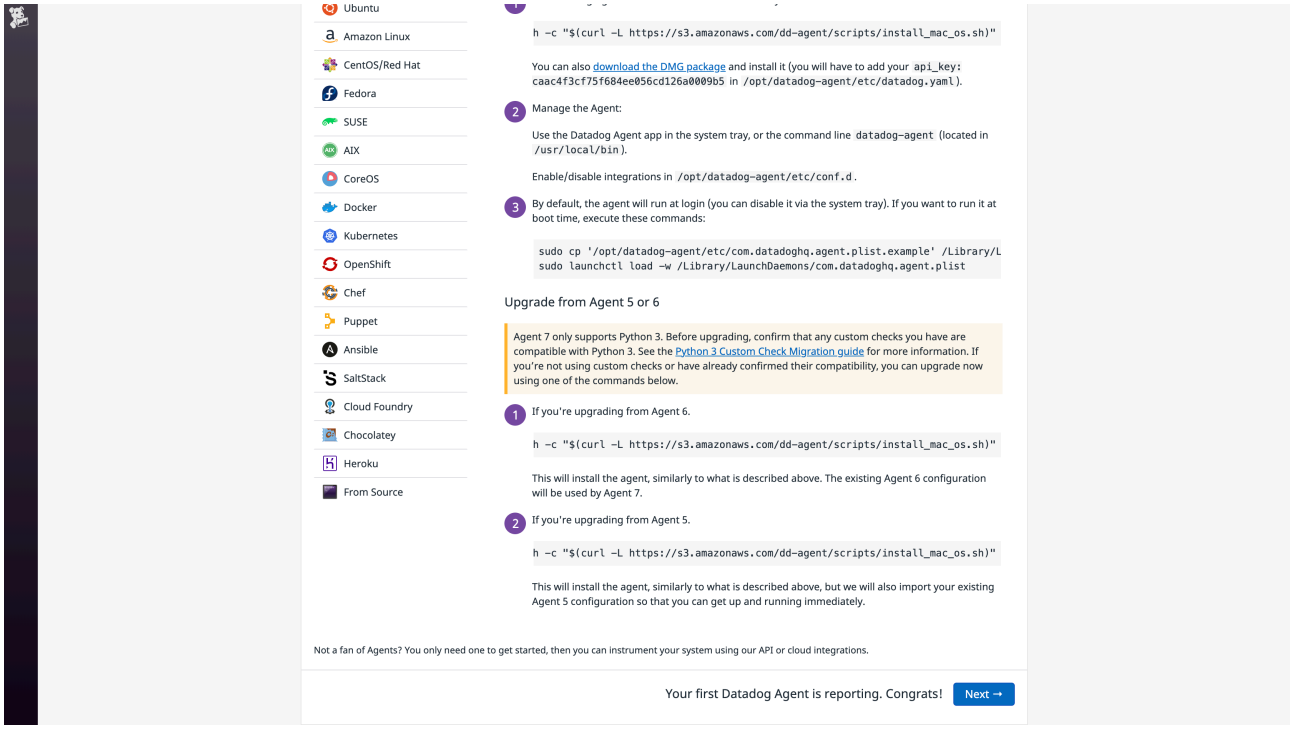


fig 3.10 Datadog-agent setup

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

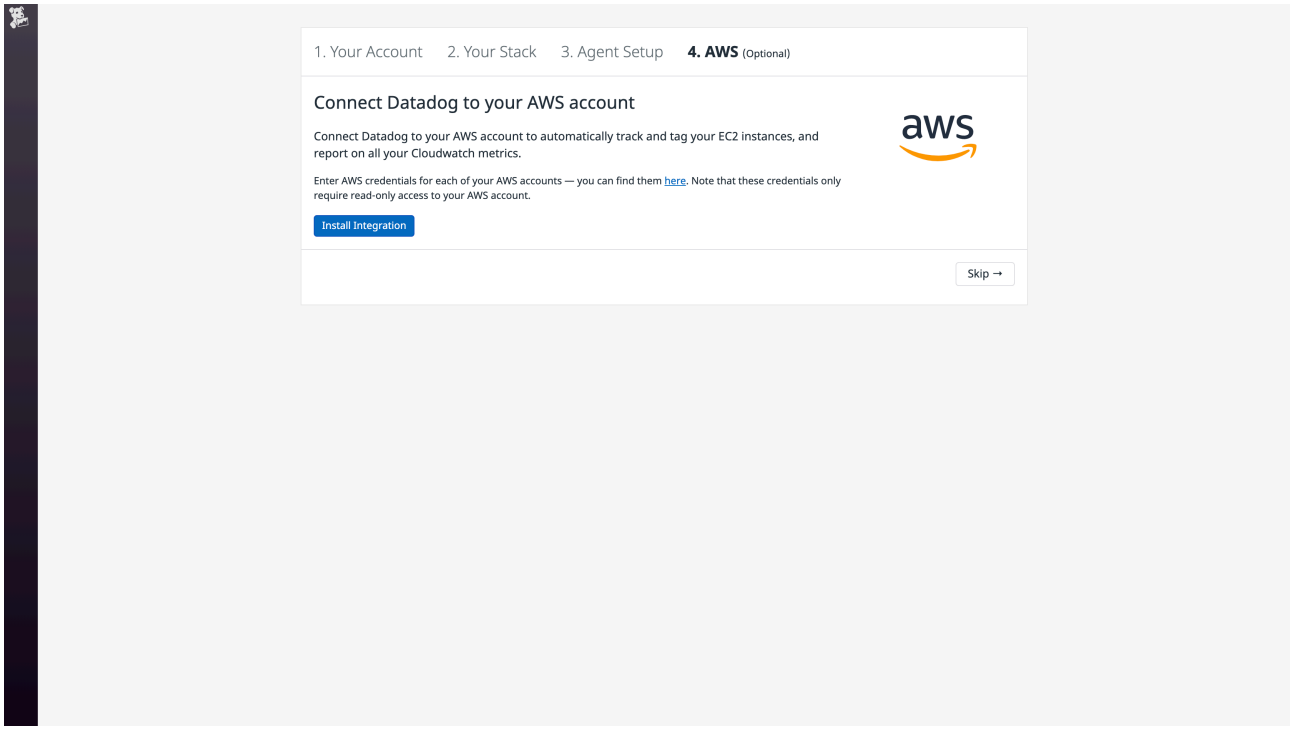


fig 3.11 AWS option

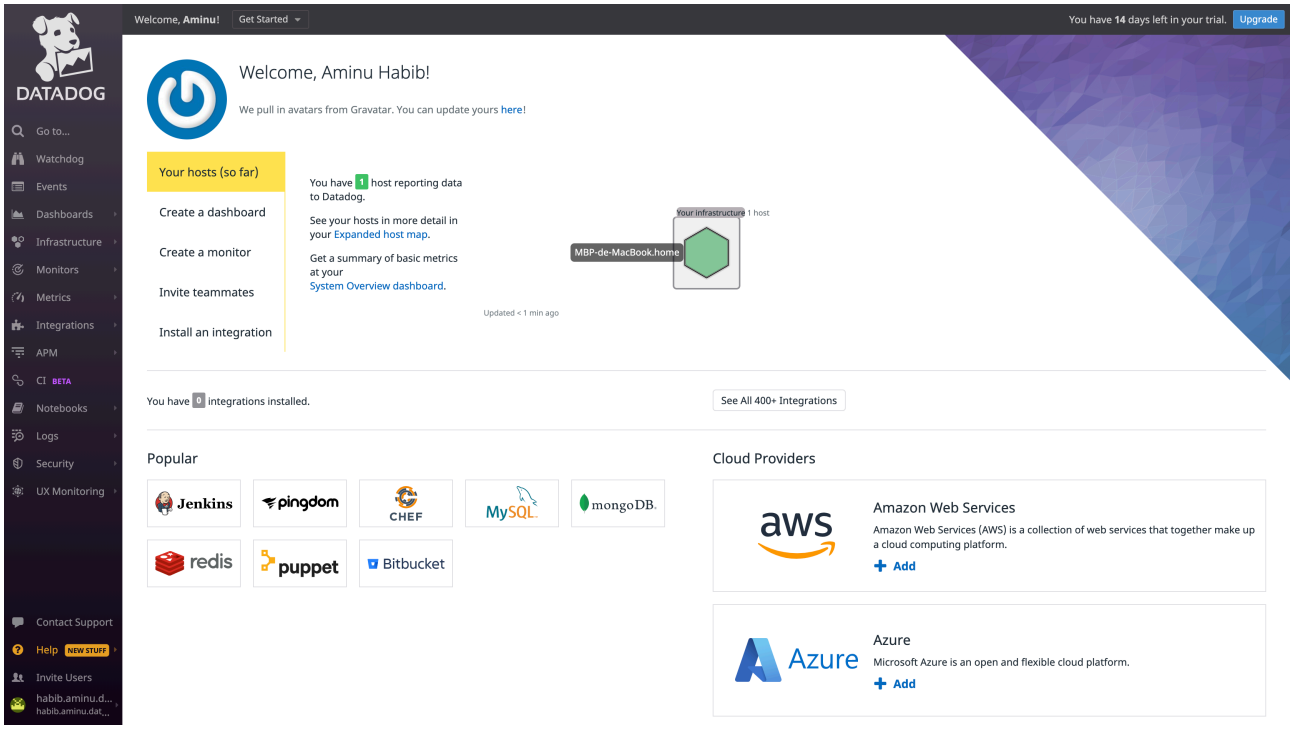


fig 3.12 Datadog Working Environment

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.

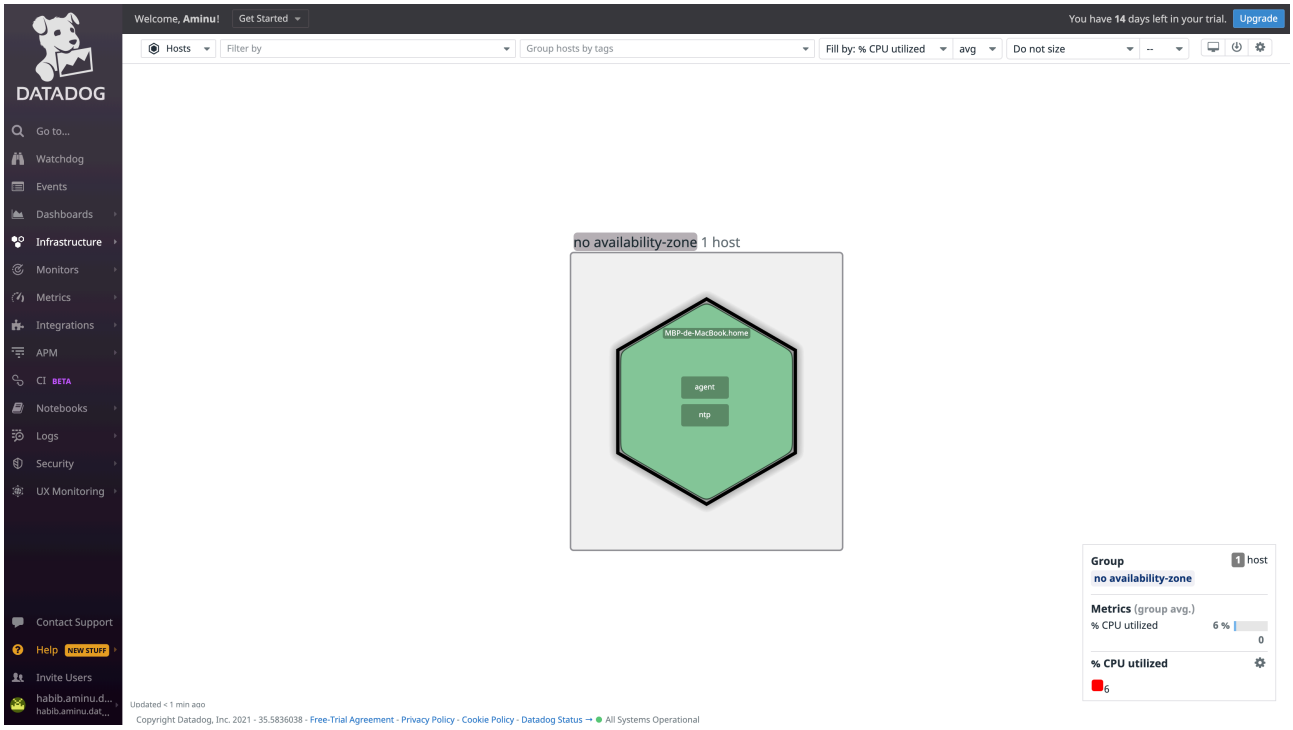


fig 3.13 Host Machine

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.

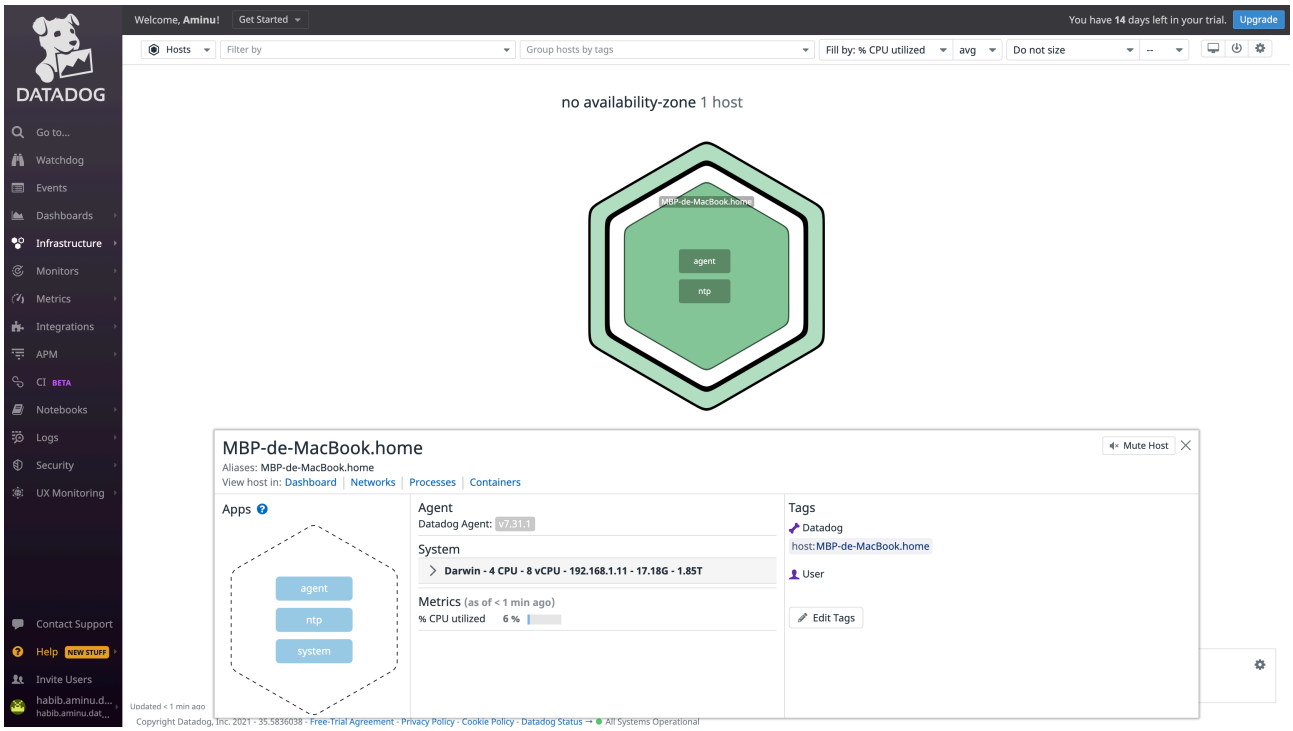


fig 3.14 Host Machine CPU Metric

We could monitor the host in: 1. Dashboard Monitoring

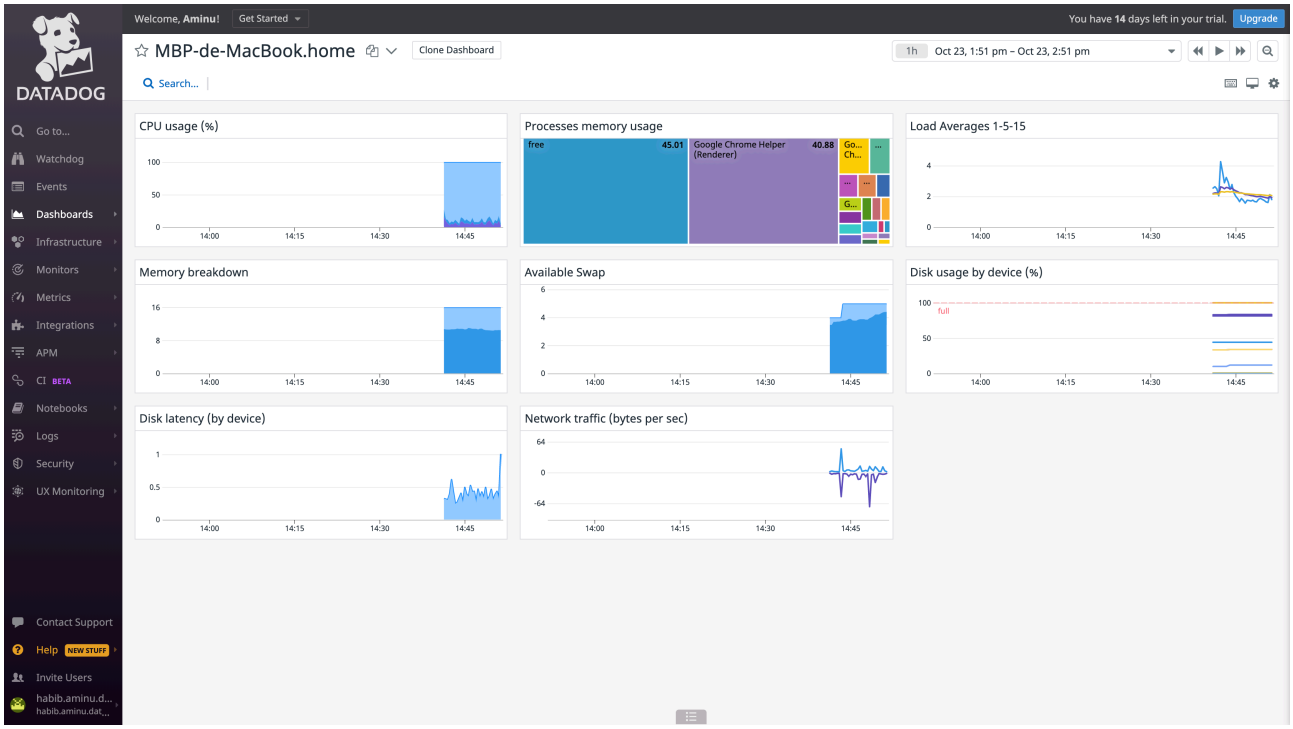


fig 3.14 Host Machine Dashboard Monitoring

2. Network Monitoring

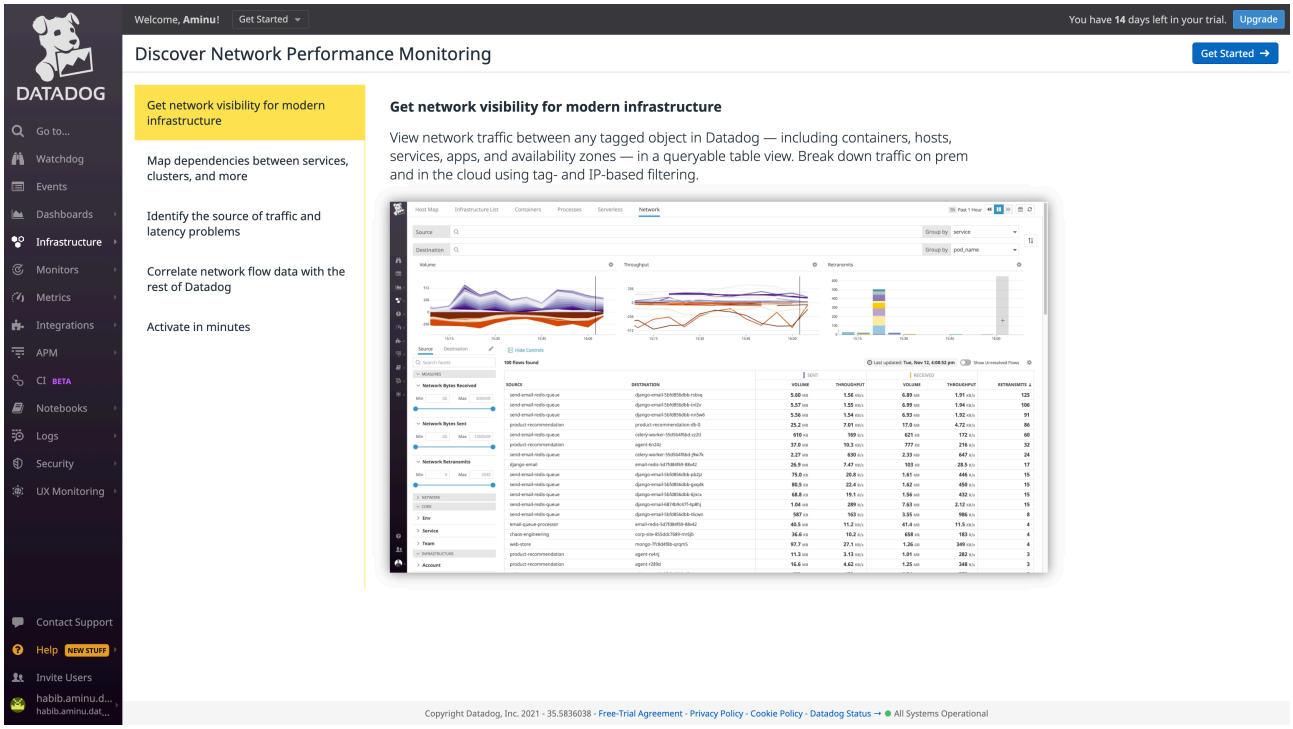


fig 3.15 Network Performance Monitoring

3. Process Monitoring

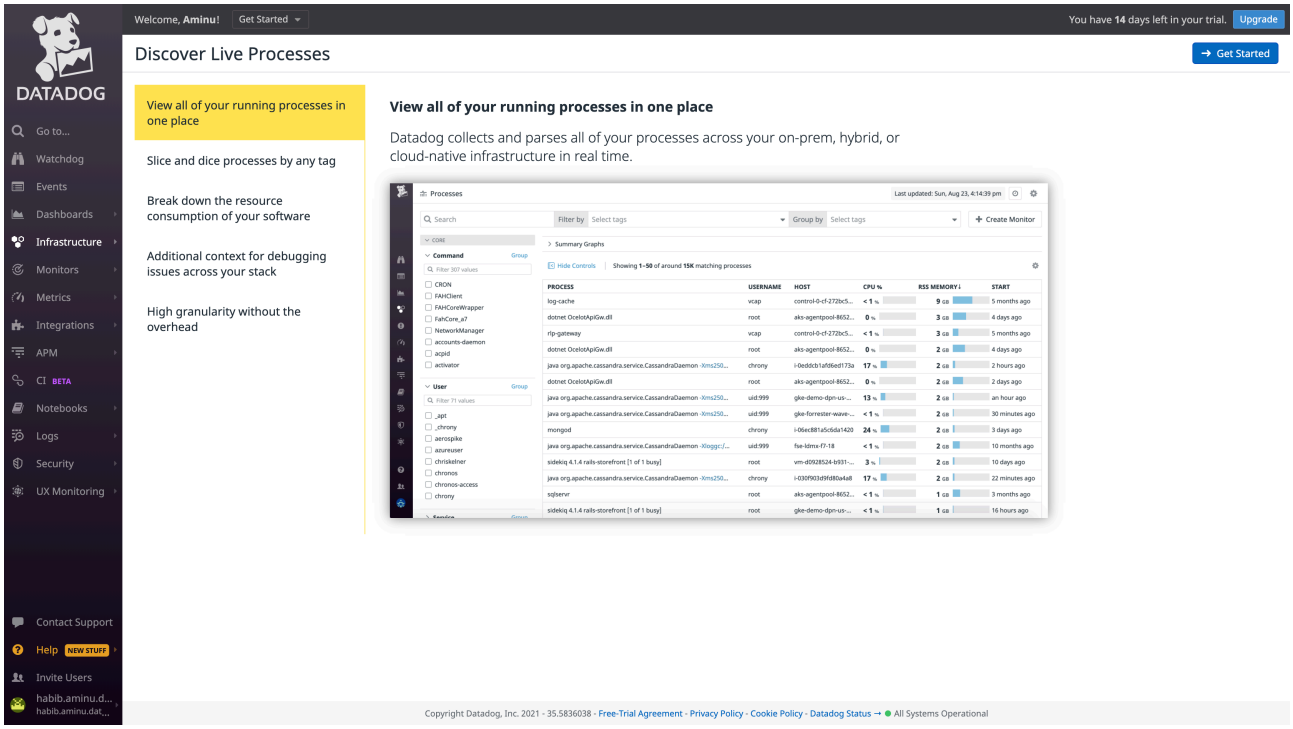


fig 3.16 Process Monitoring

4. Container Monitoring

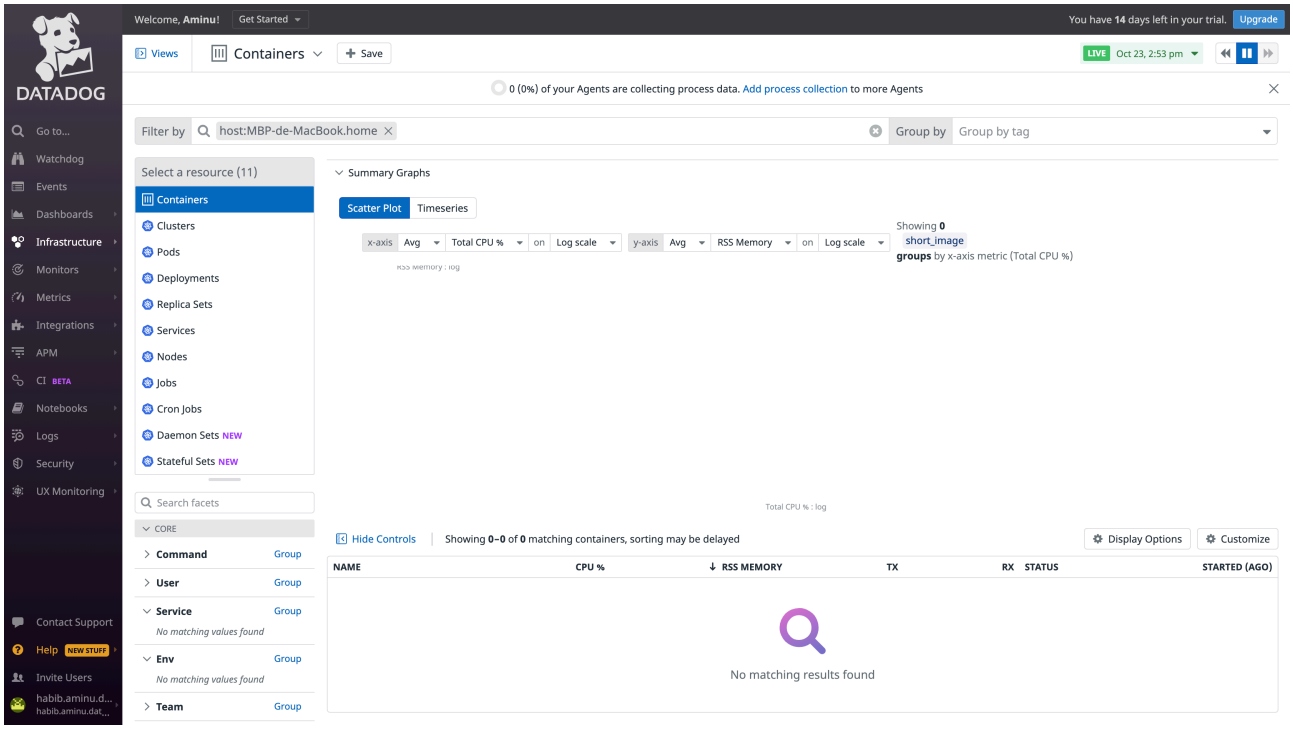


fig 3.17 Container Monitoring

You can stop the agent using the "**datadog-agent stop**" command.

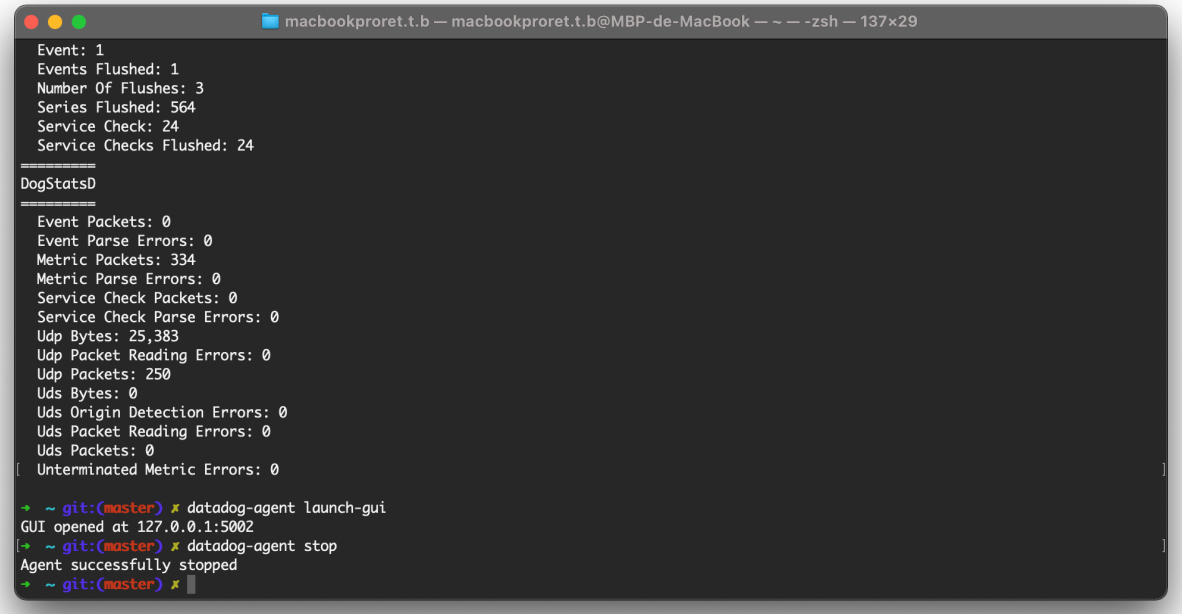


fig 3.18 Agent Successfully Stopped

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

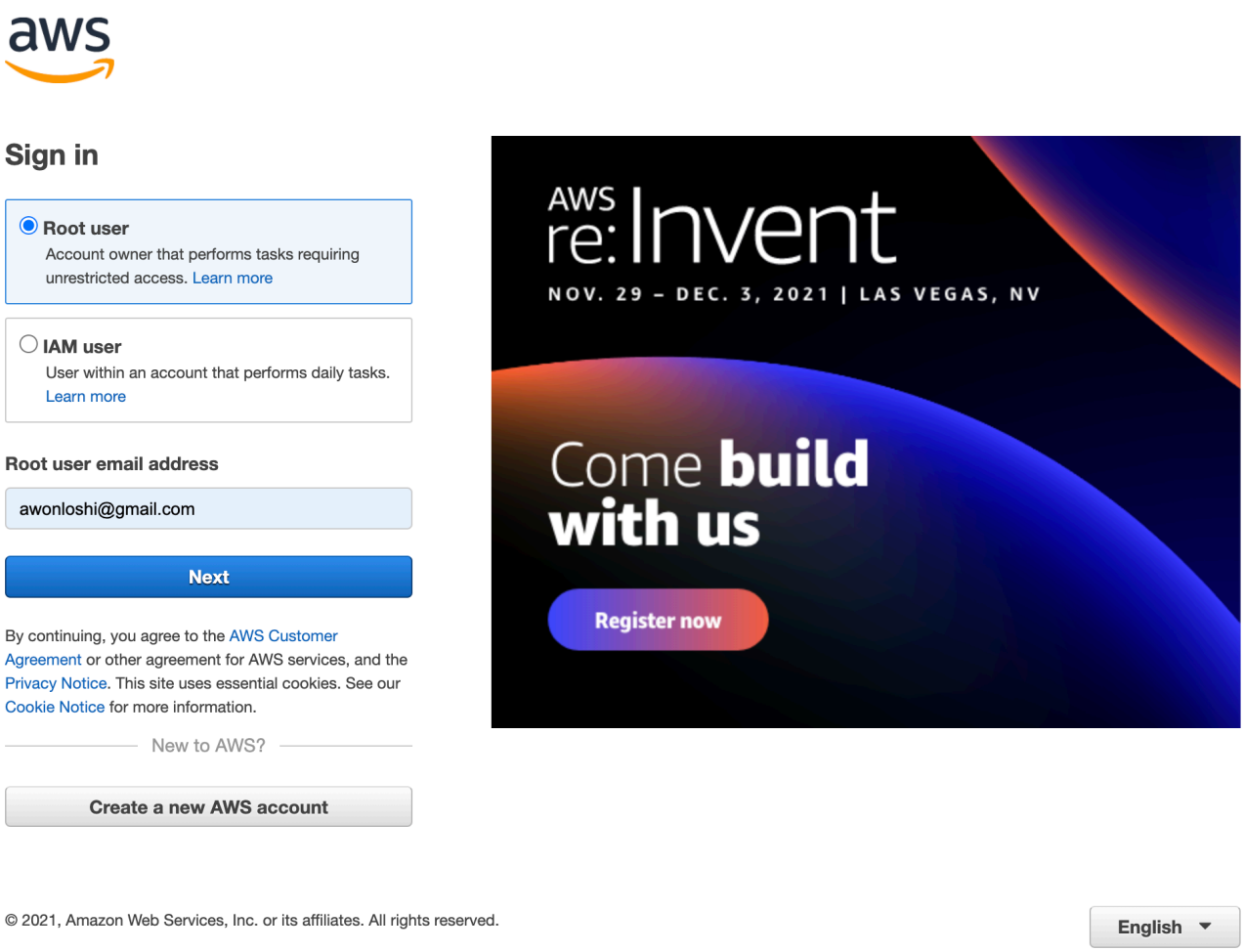


fig 4.1 Create a new AWS account

- 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-0lkNgCwbv1MMKp2xoCav0QAvD_BwE

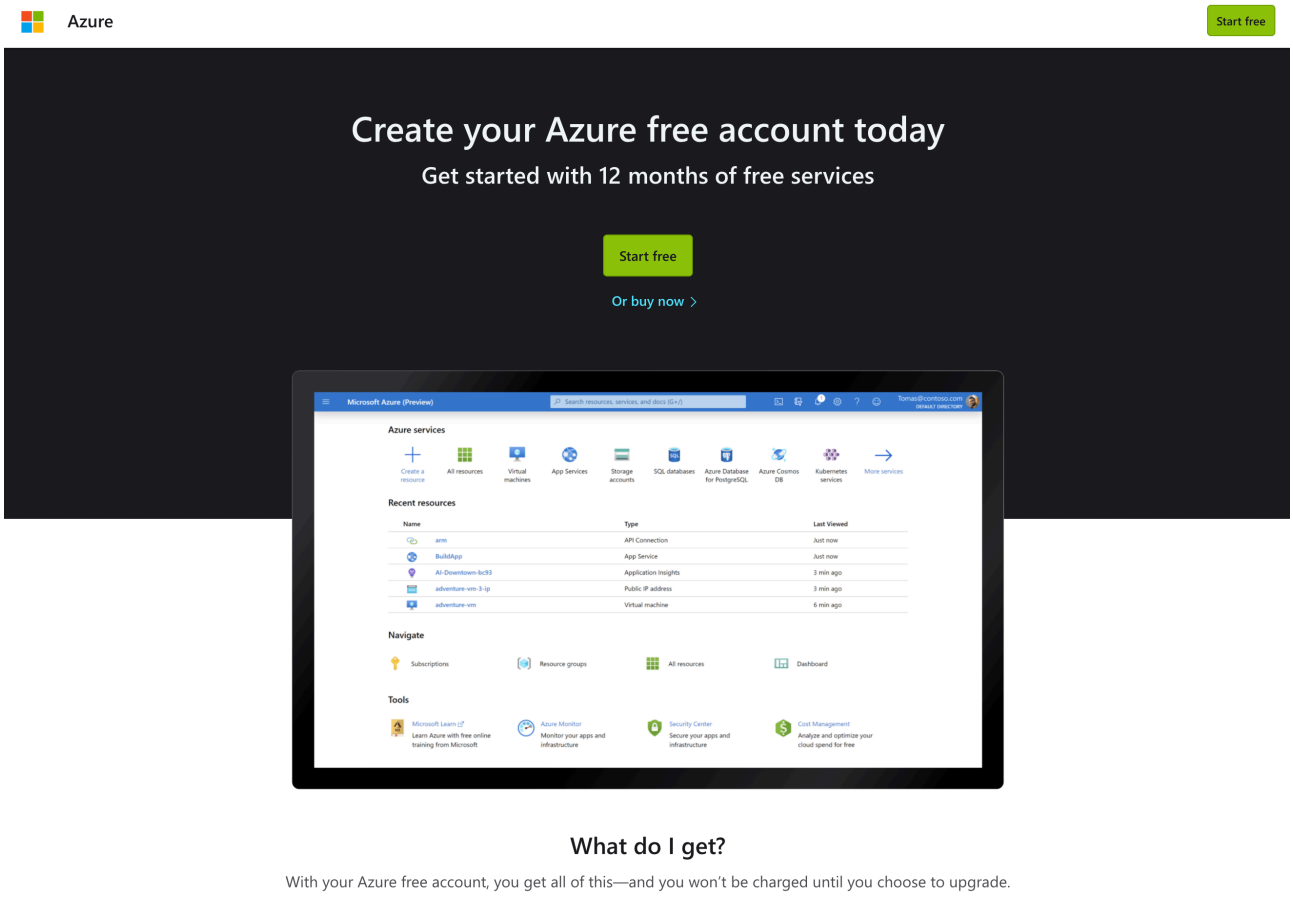


fig 4.2 Create a new Azure account

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.

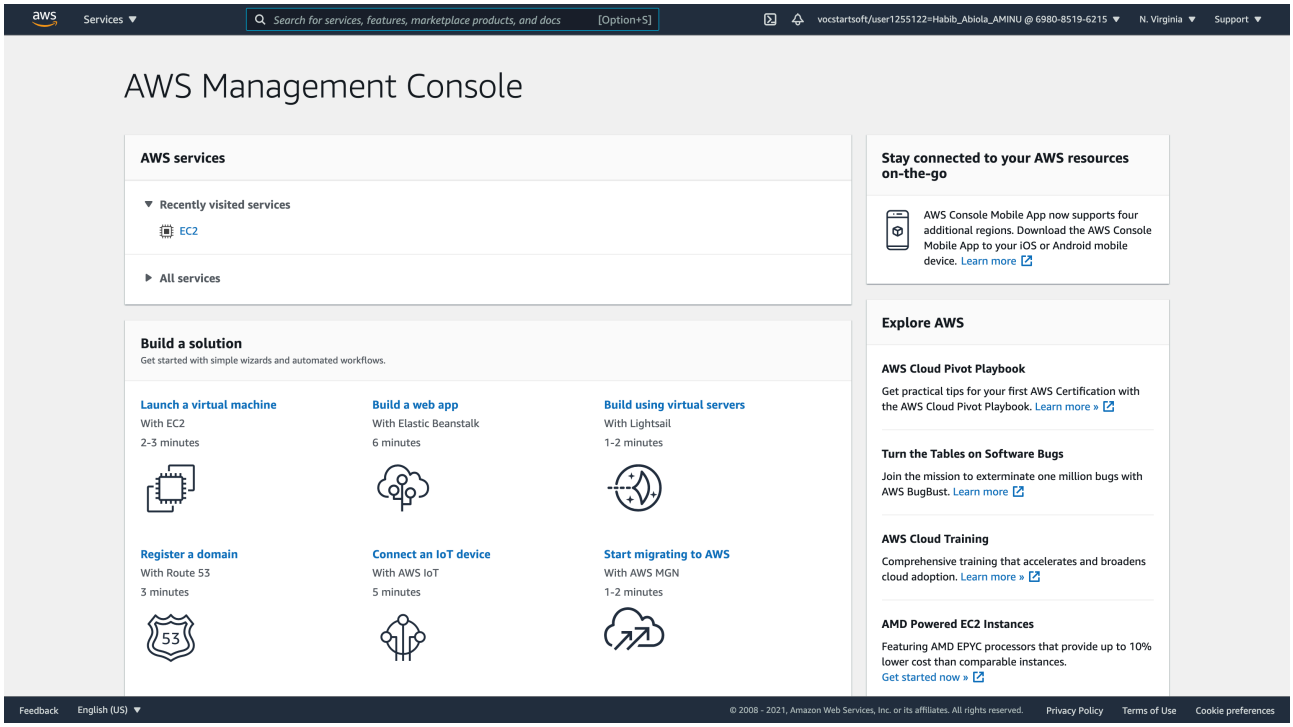


fig 4.3 AWS Management Console

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

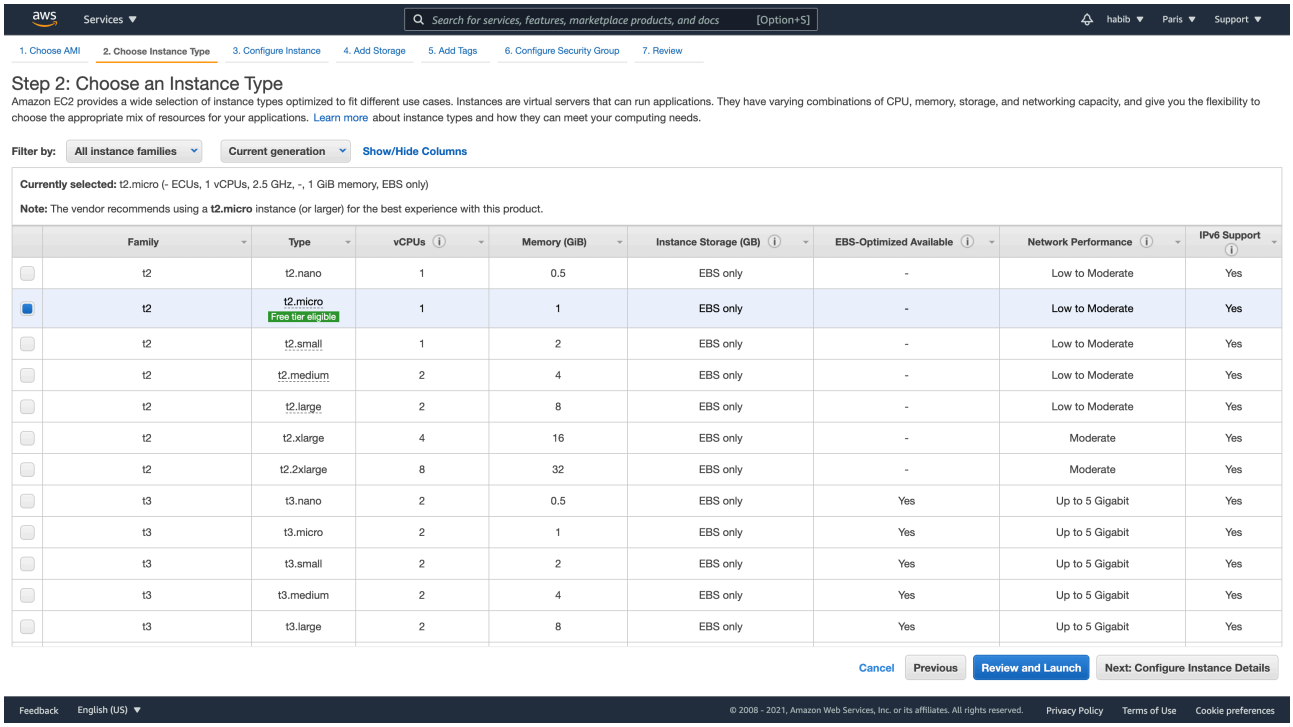


fig 4.4 EC2 Type

- 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.

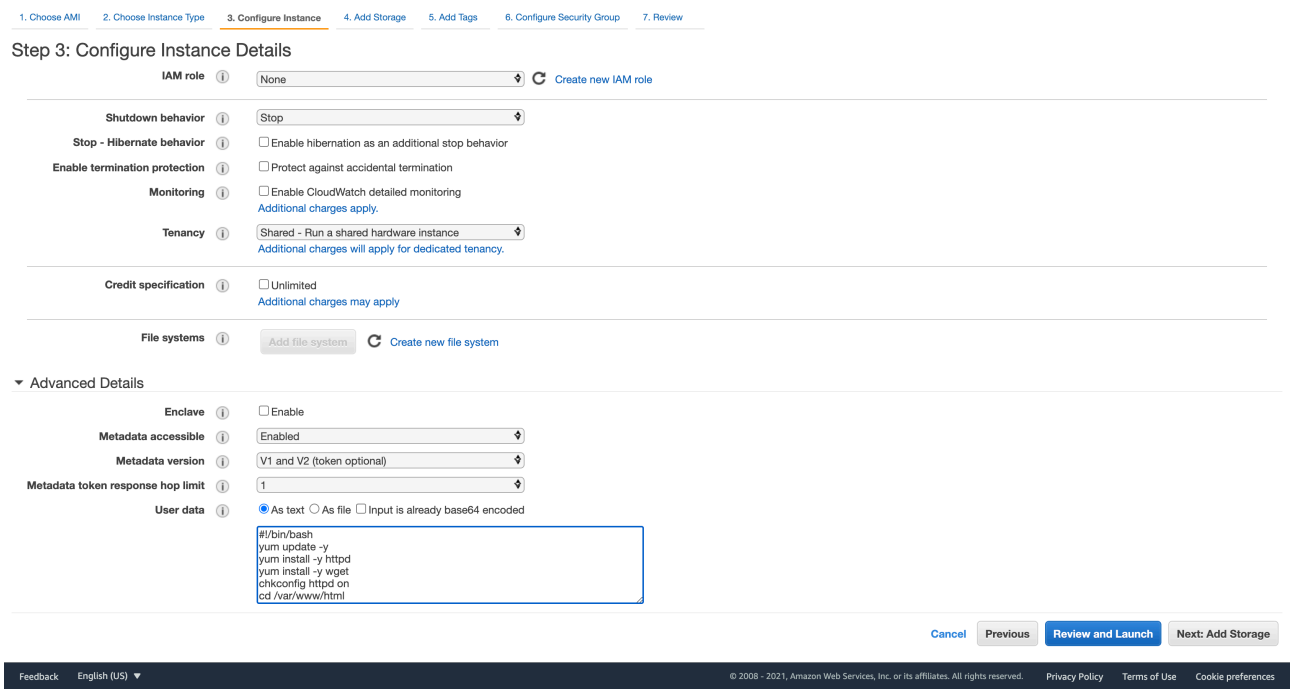


fig 4.5 Instance Configuration

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

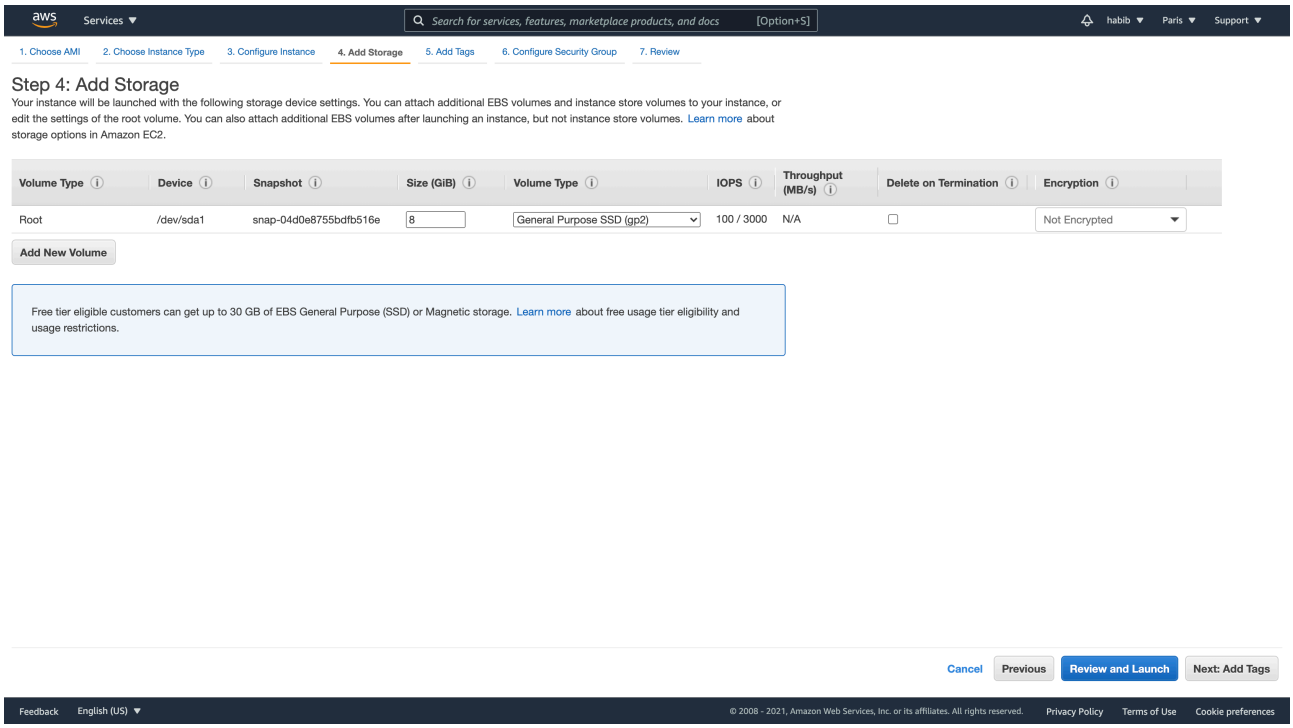


fig 4.6 Add Storage

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

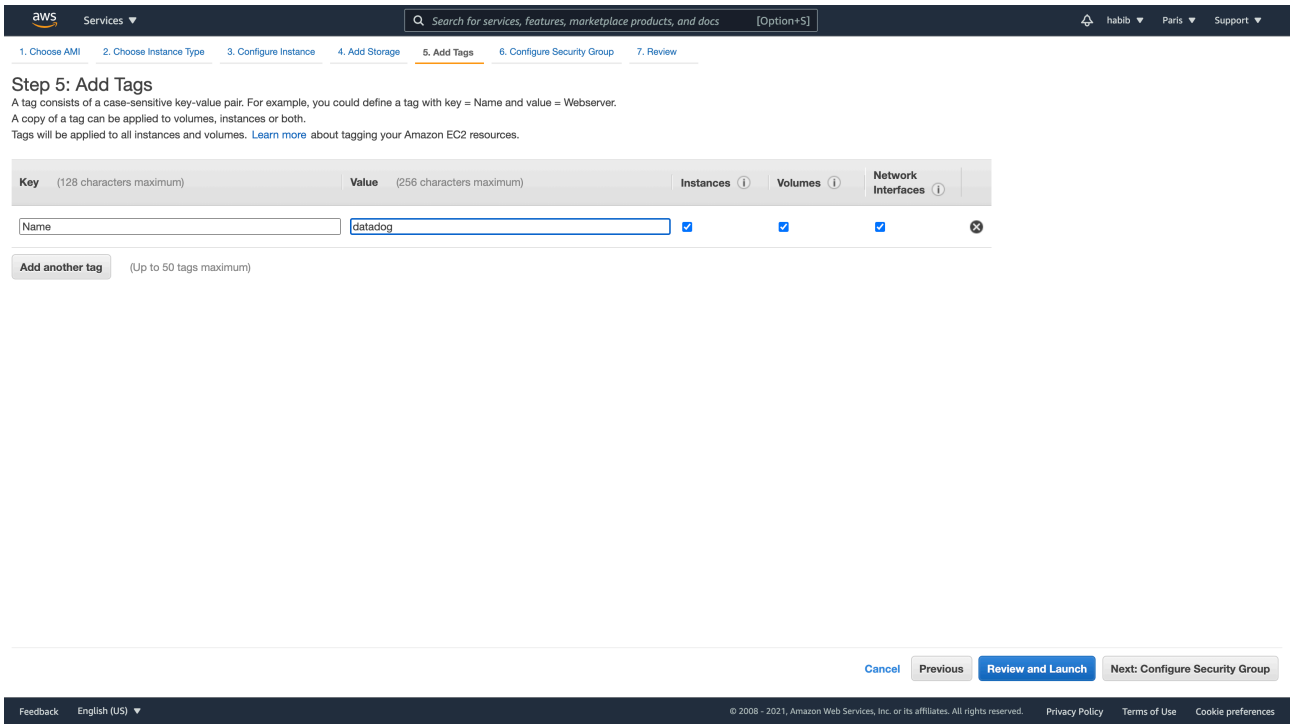


fig 4.7 Add Tag

- Key: **Name**
- Value: **datadog**

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

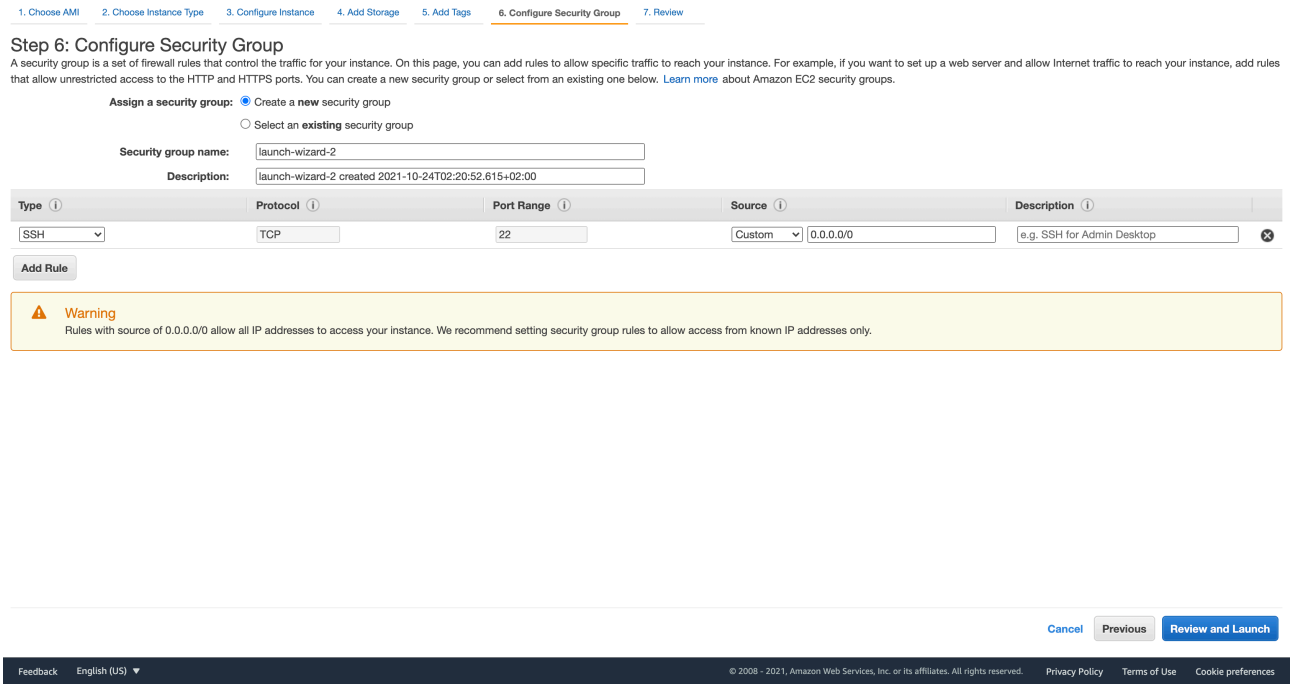


fig 4.8 Create a new Security Group

- 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**.

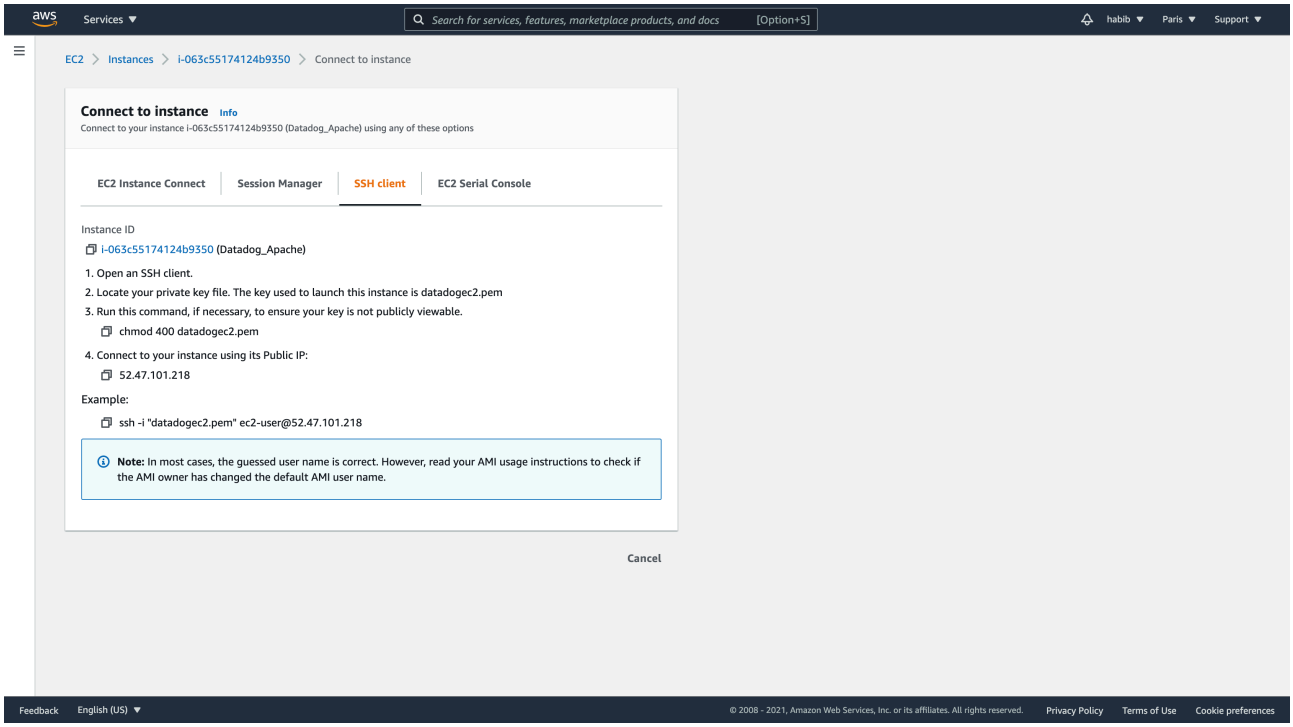


fig 4.9 Connect to Instance

- 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

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

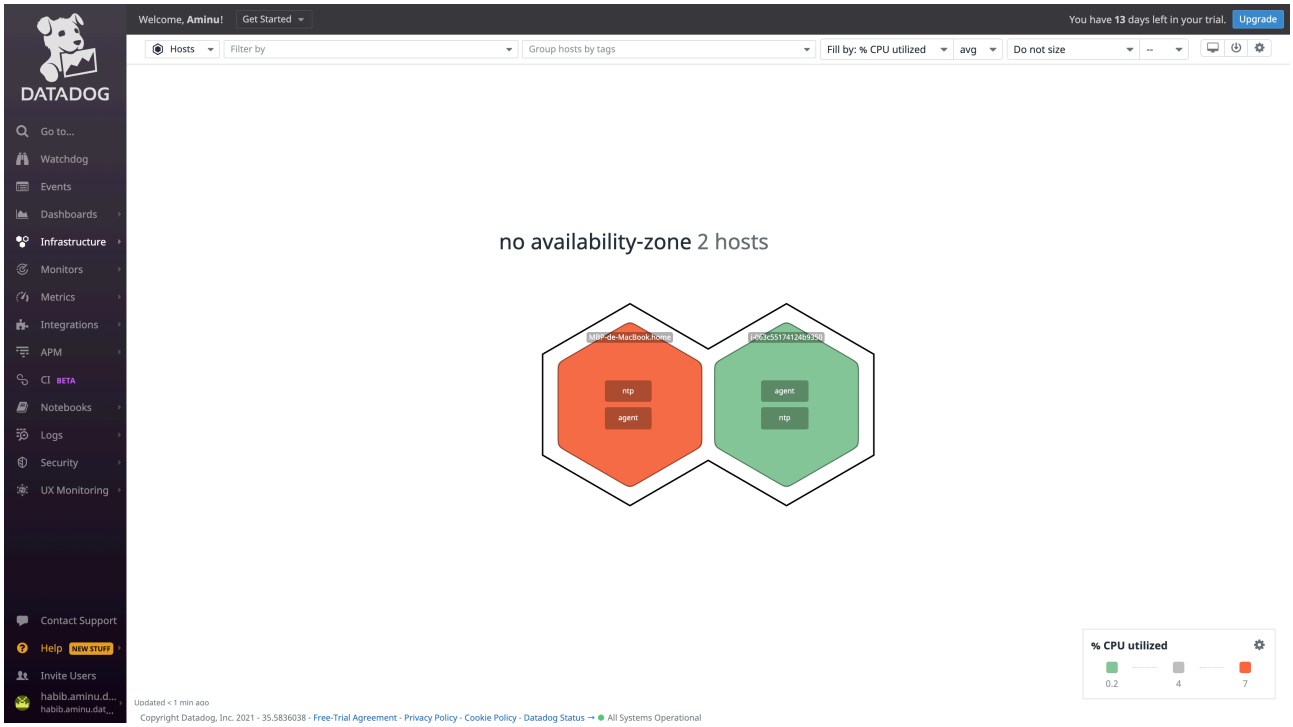


fig 4.11 Connected Host

- Containers
- Processes
- Network

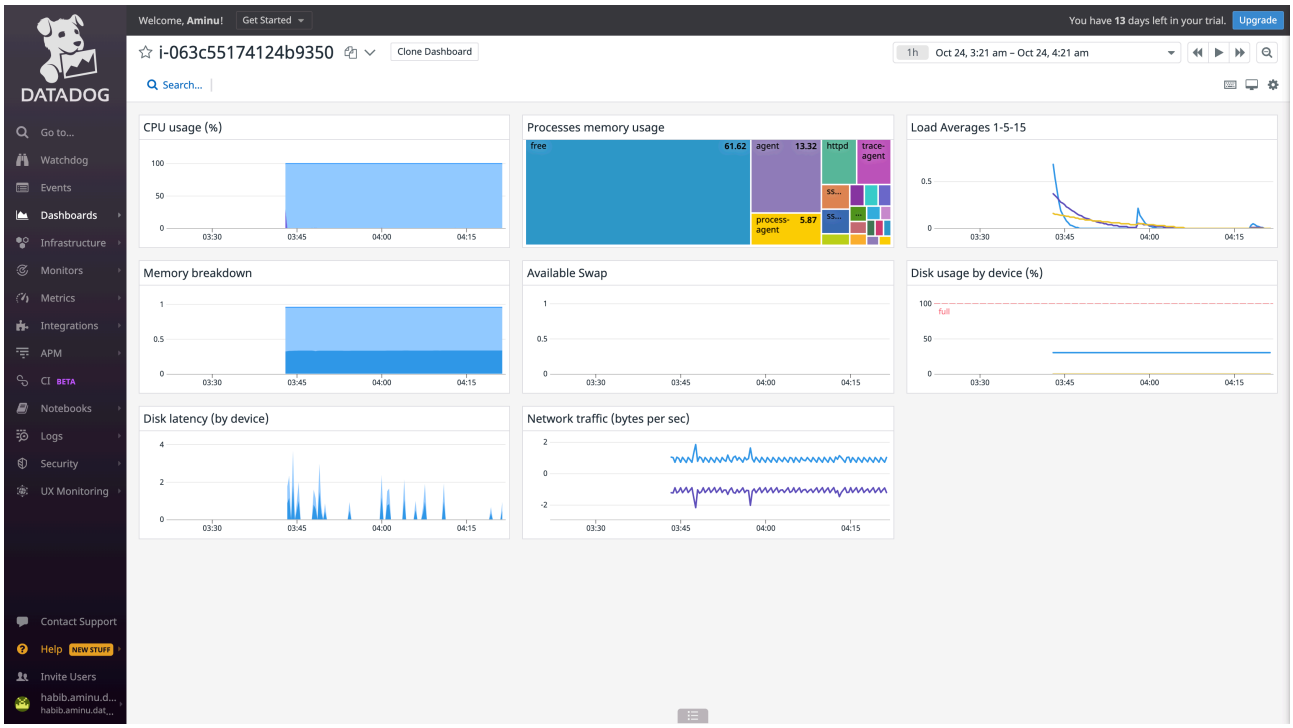


fig 4.12 Host Dashboard

- Log
- Traces
- Security

- Click **Integrations** to install a service to monitor

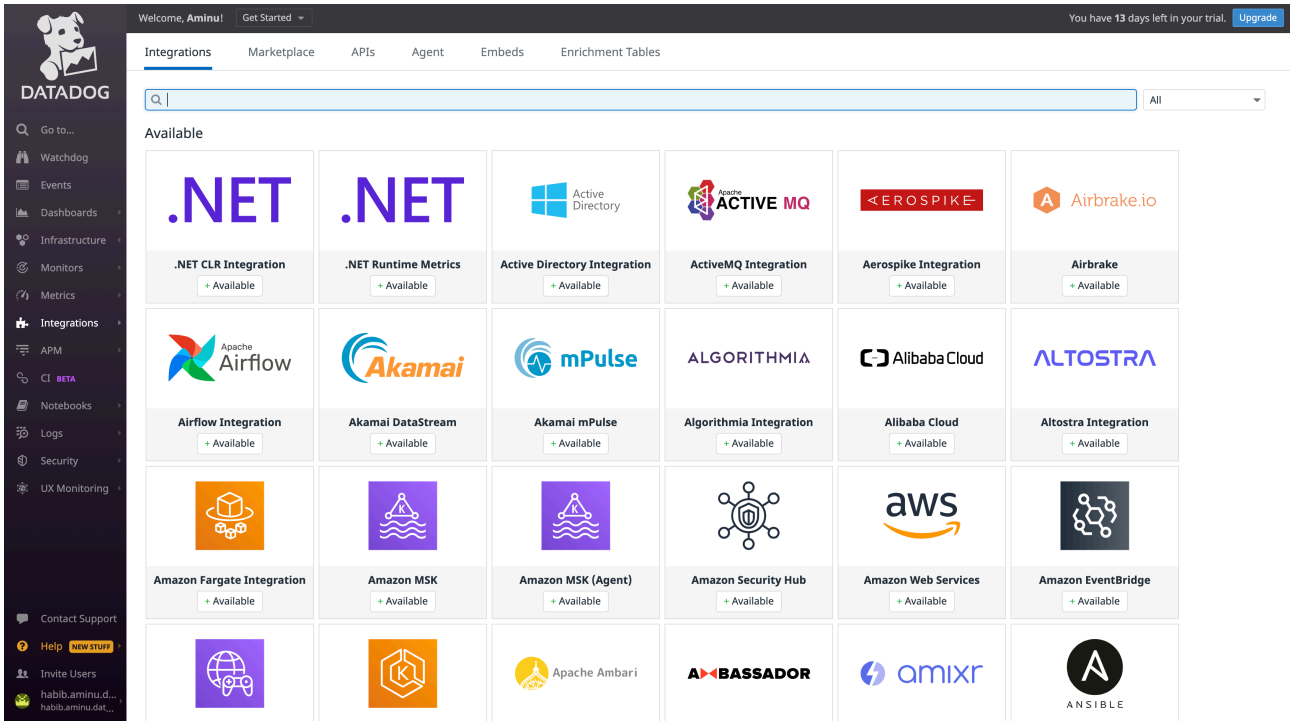
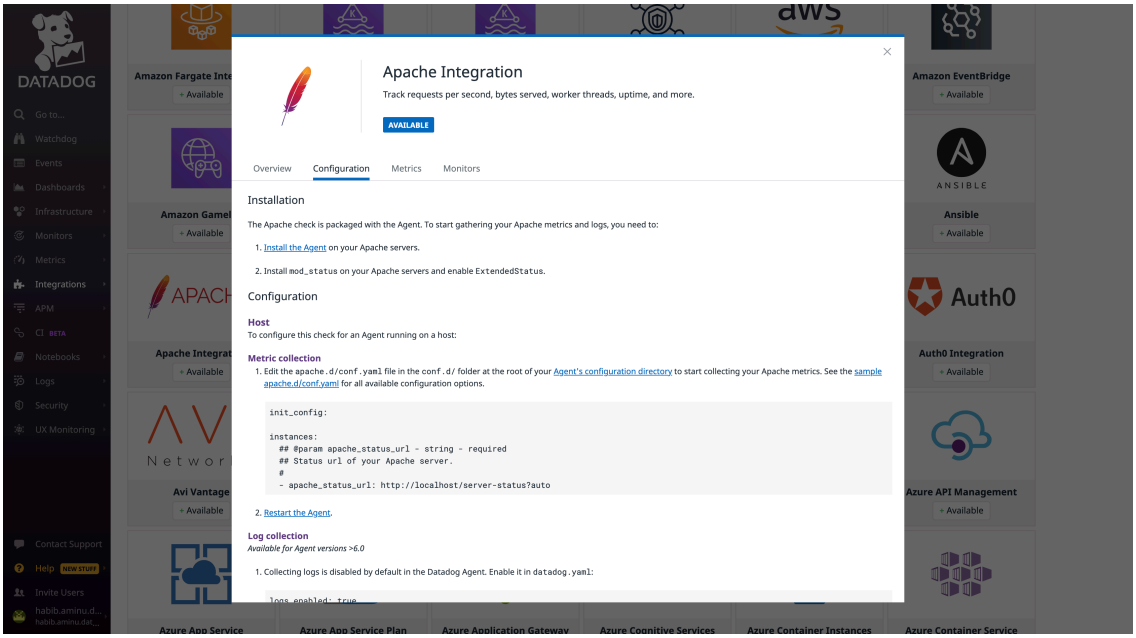


fig 4.13 Integration Service

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



- 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

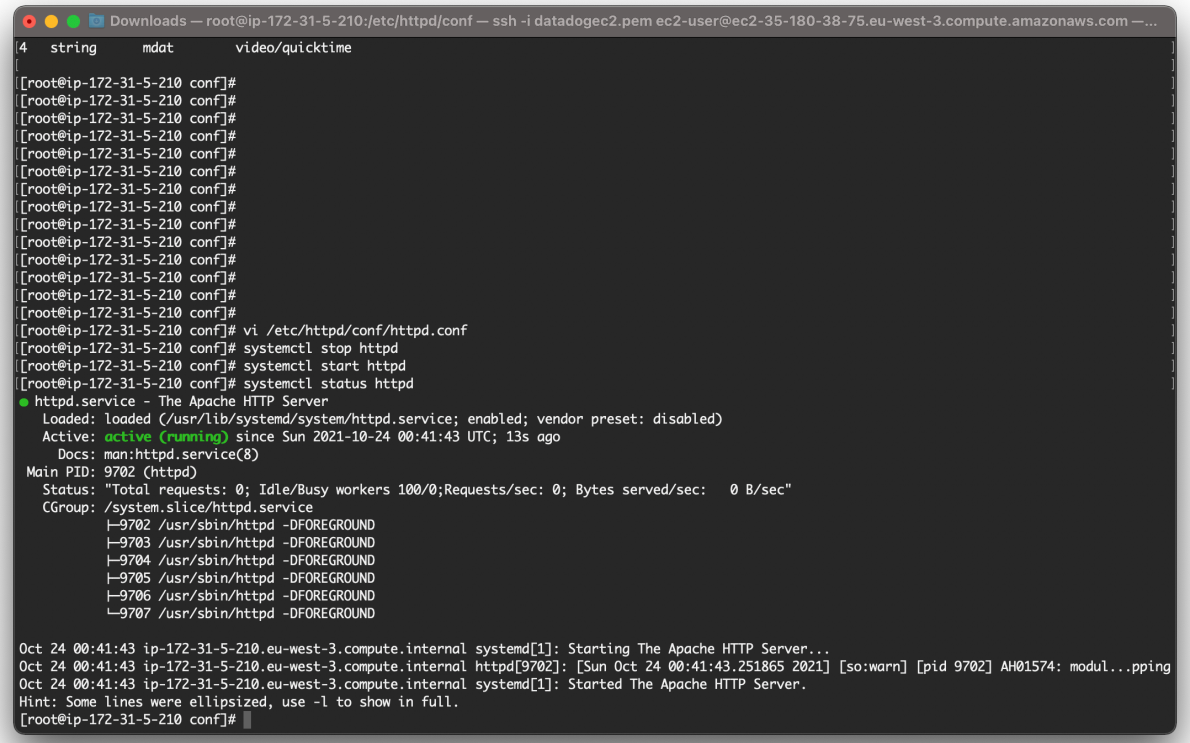


fig. Apache Configuration

- 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
    sourcecategory: http_web_access

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

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

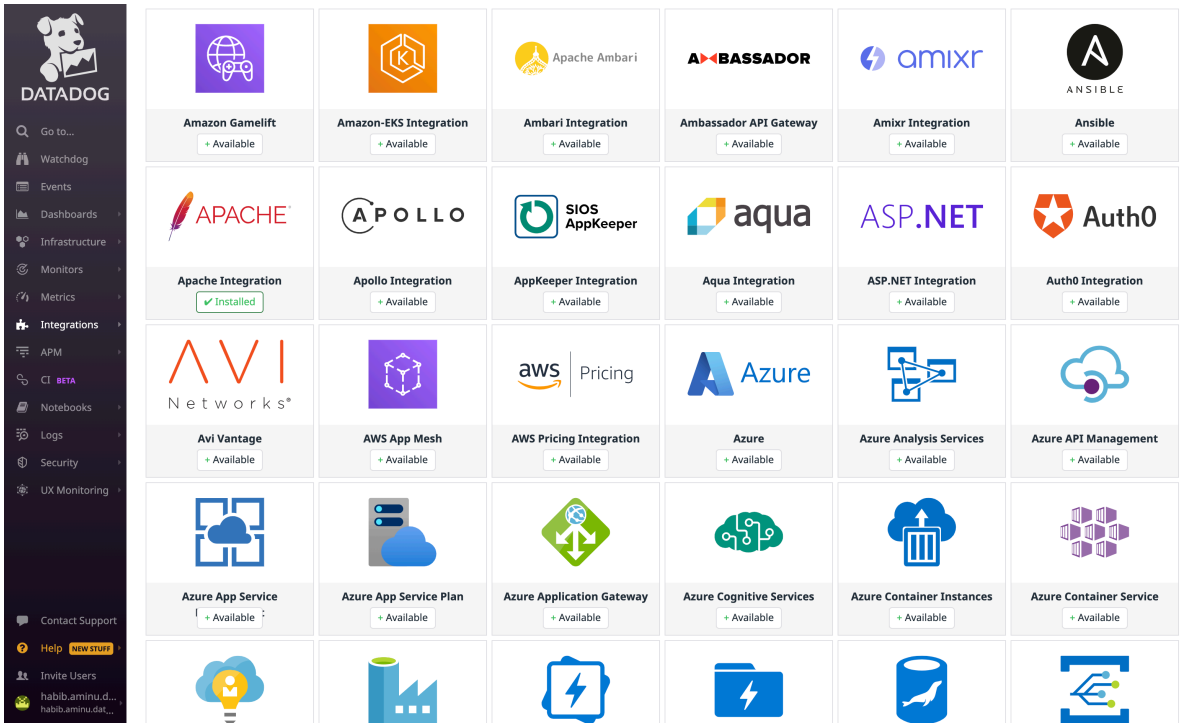


fig 4.14 Apache Installed

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

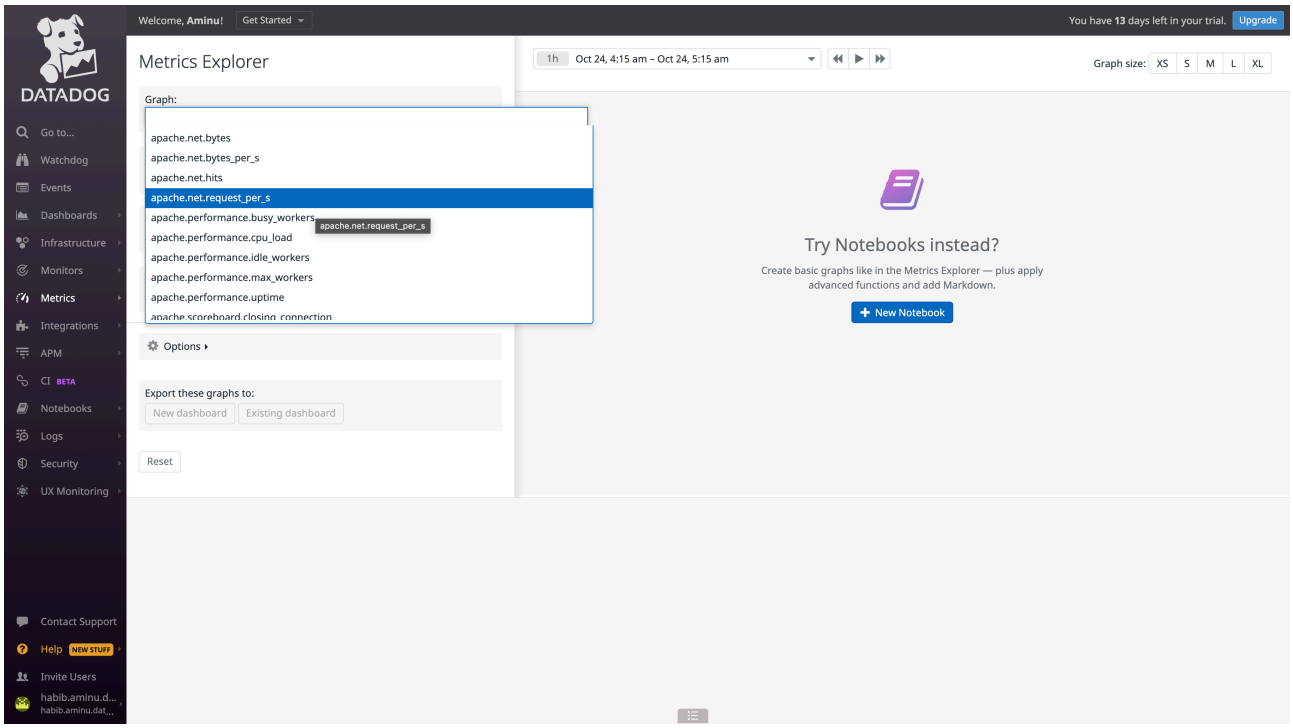


fig 4.15 Metrics

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

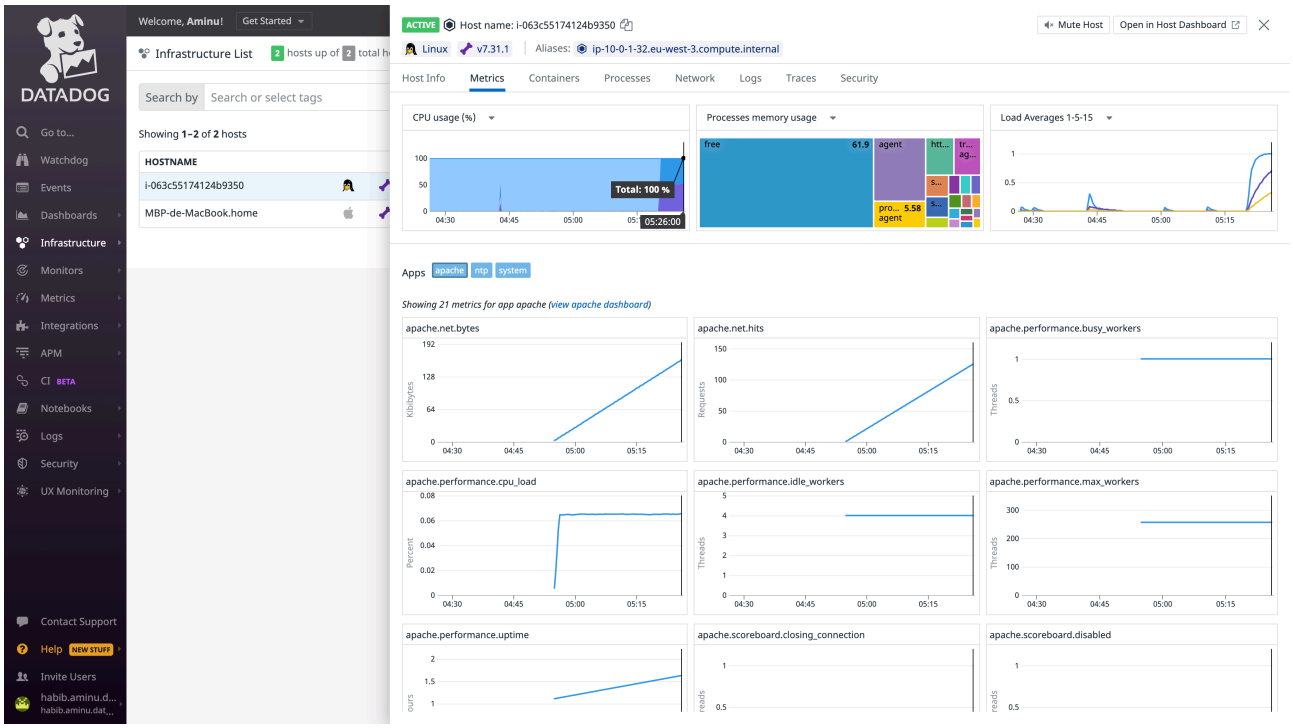


fig 4.16 Host Infrastructure Metrics

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

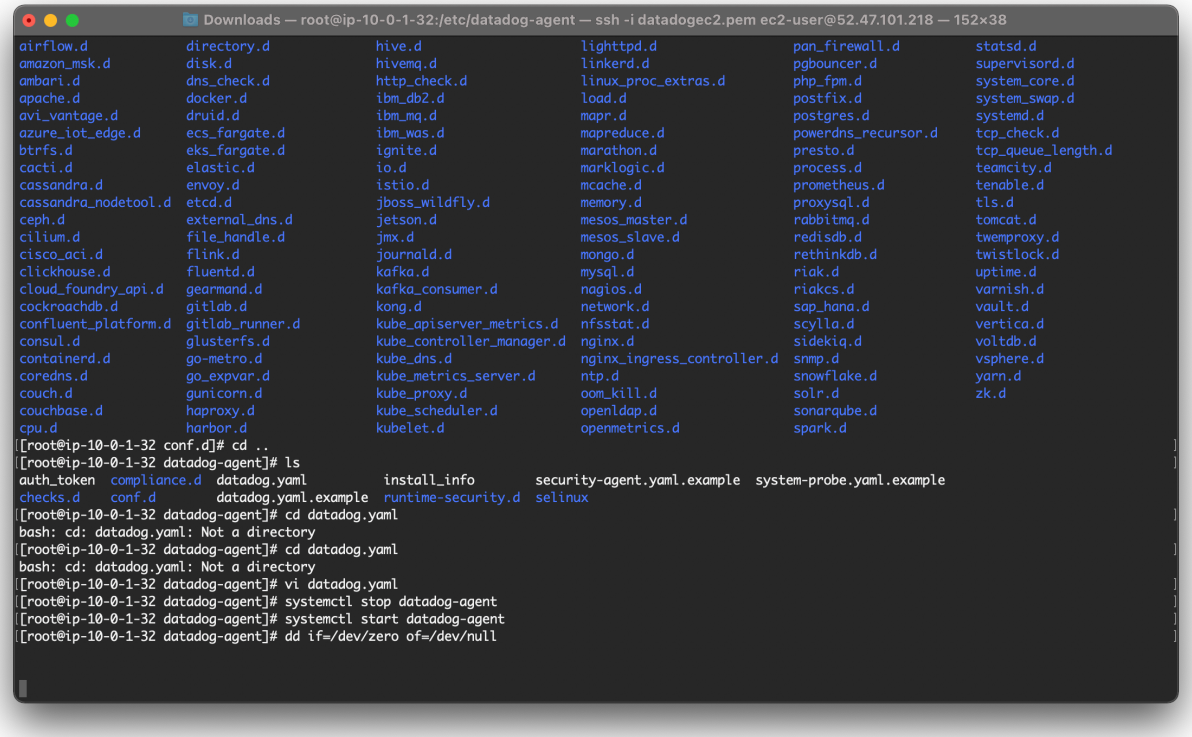


fig 4.17 Terminal

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

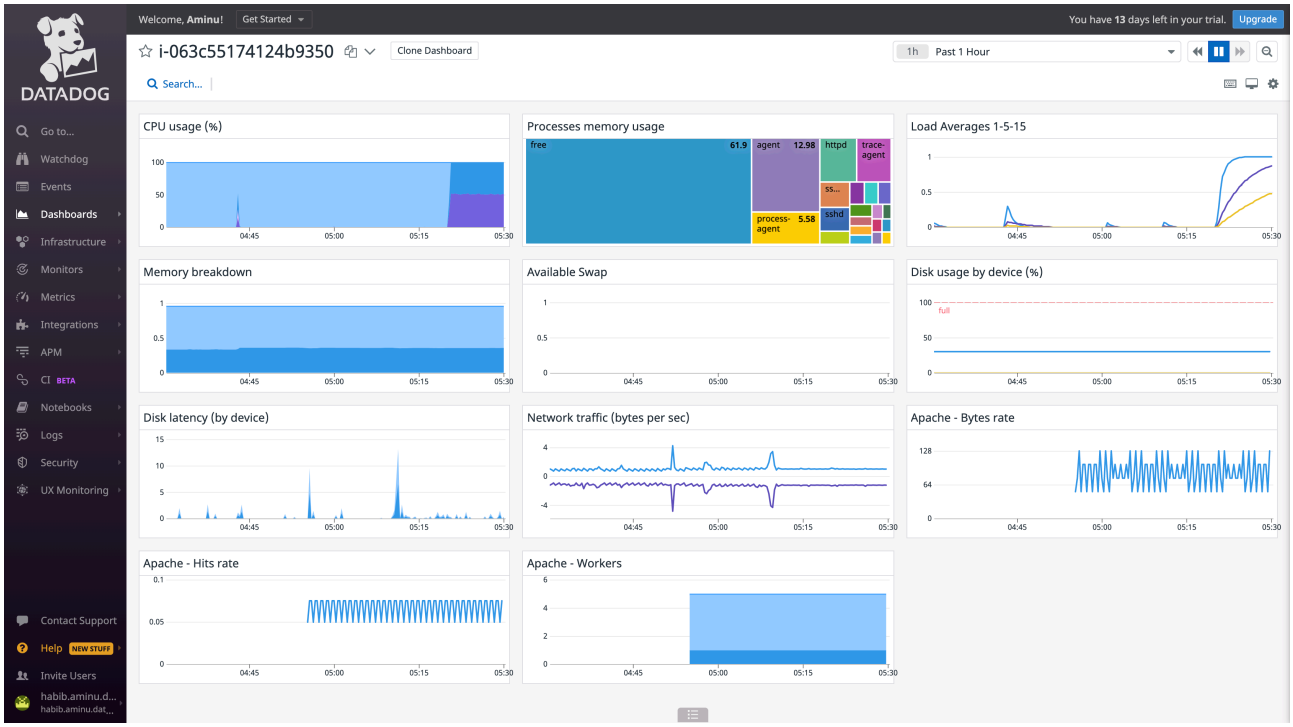


fig 4.18 Host Dashboard

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

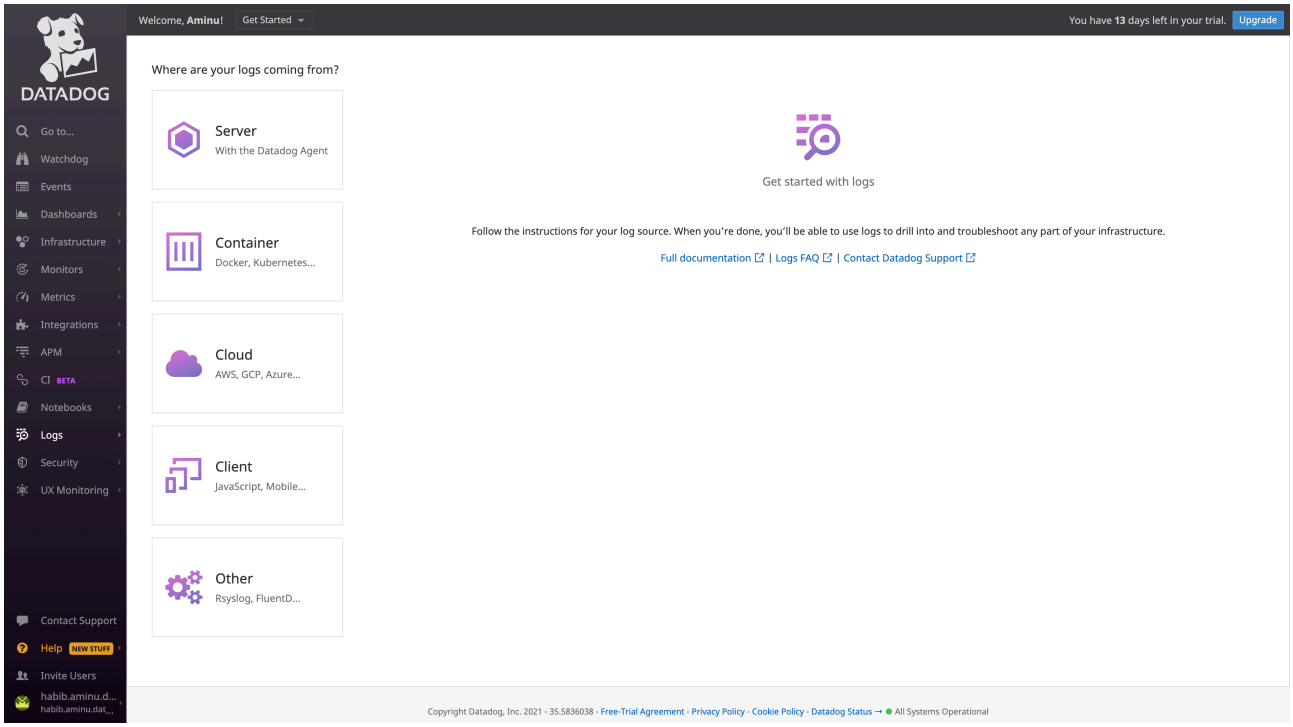


fig 4.19 Host Logs

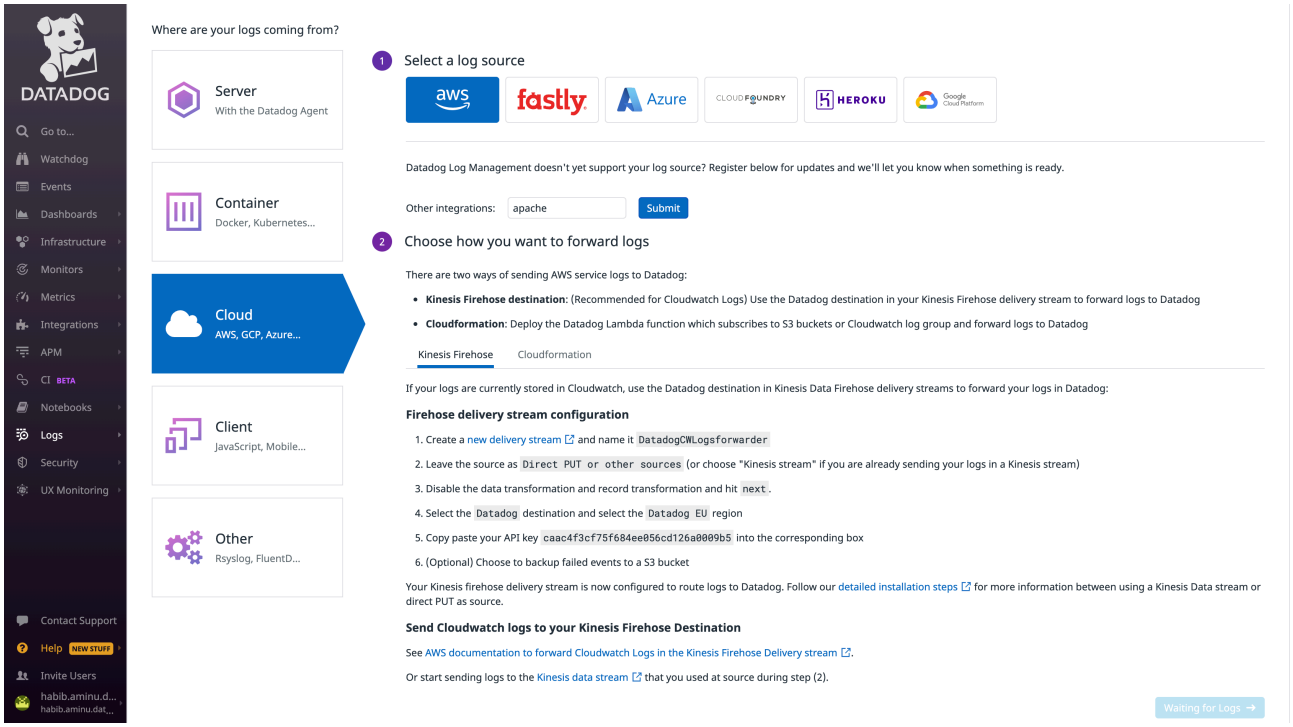


fig 4.20 AWS Logs Configuration

- Click **Monitor**

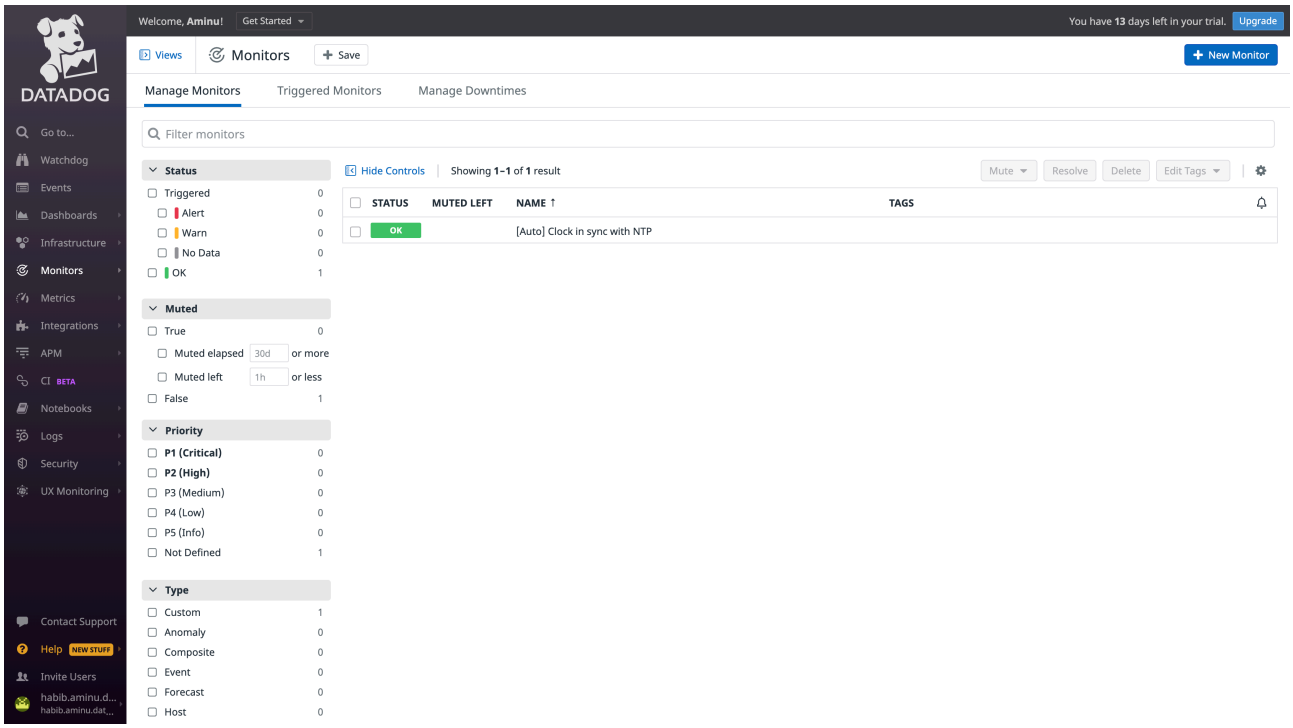


fig 4.21 Host Monitor

- 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

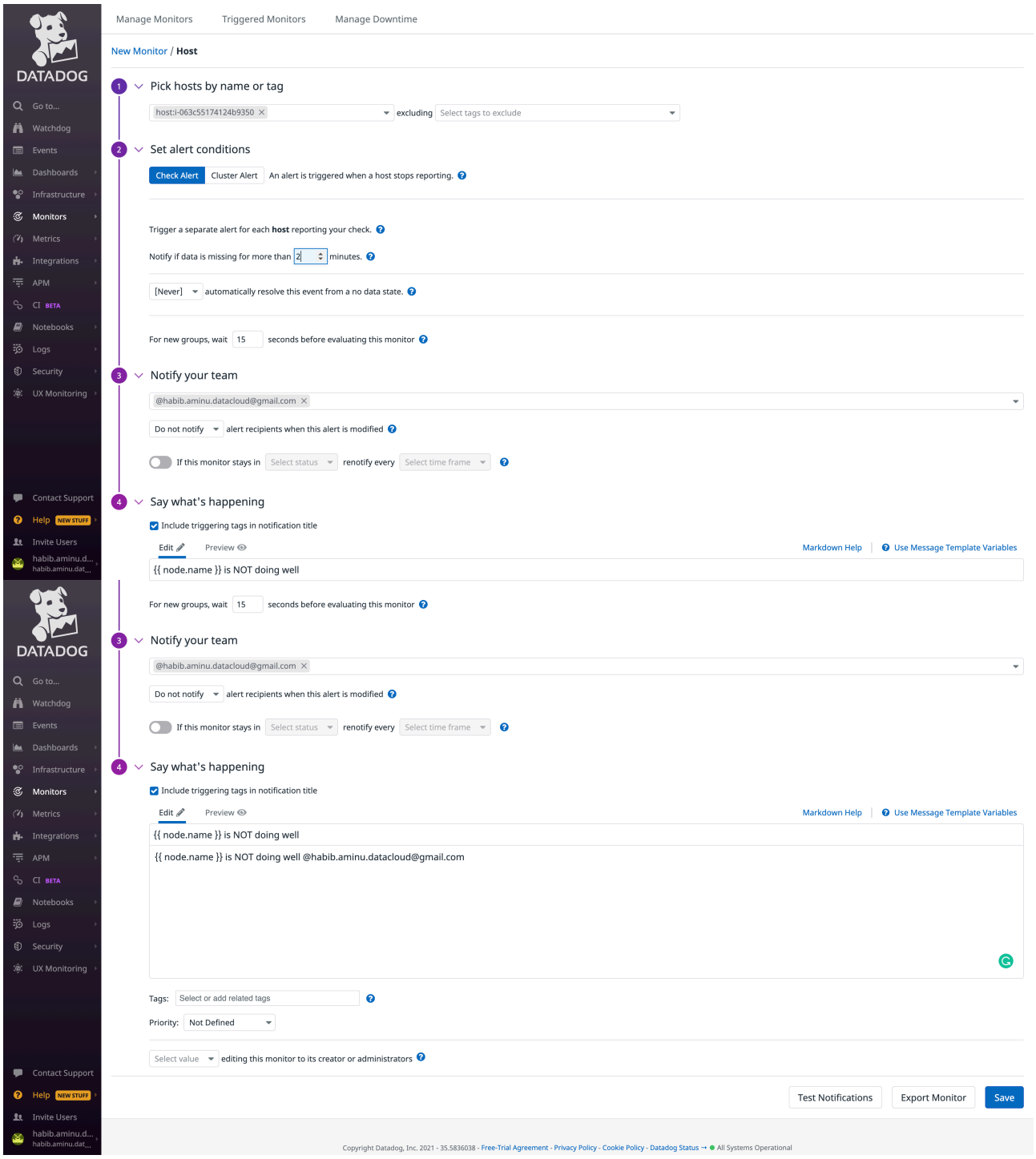


fig 4.22 Host Machine Monitor Configuration

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

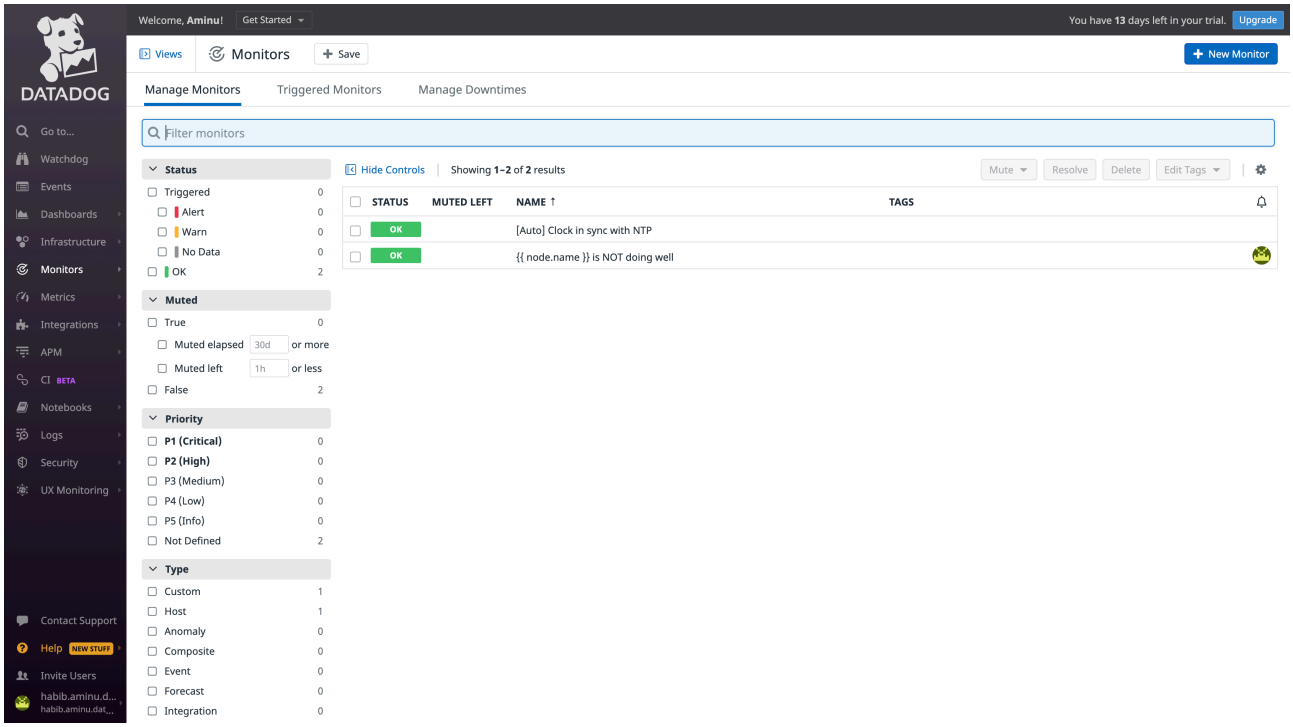
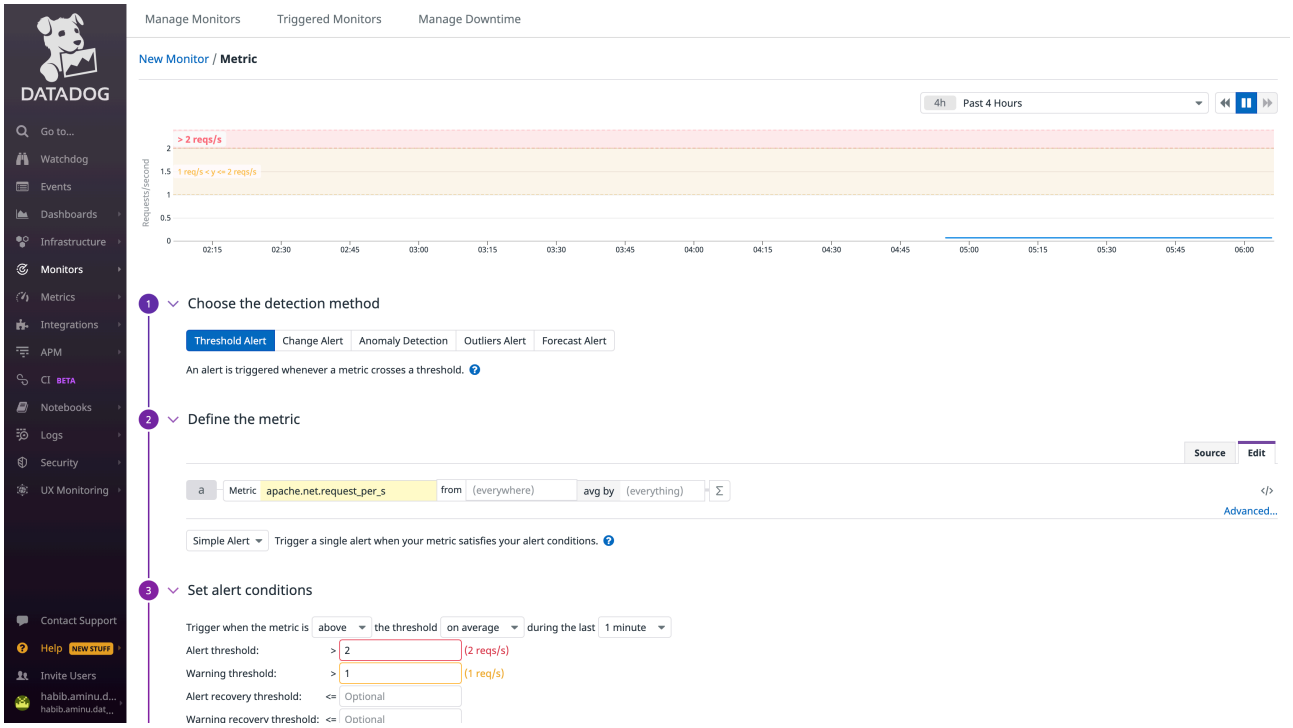


fig 4.23 Available Monitors

- 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:



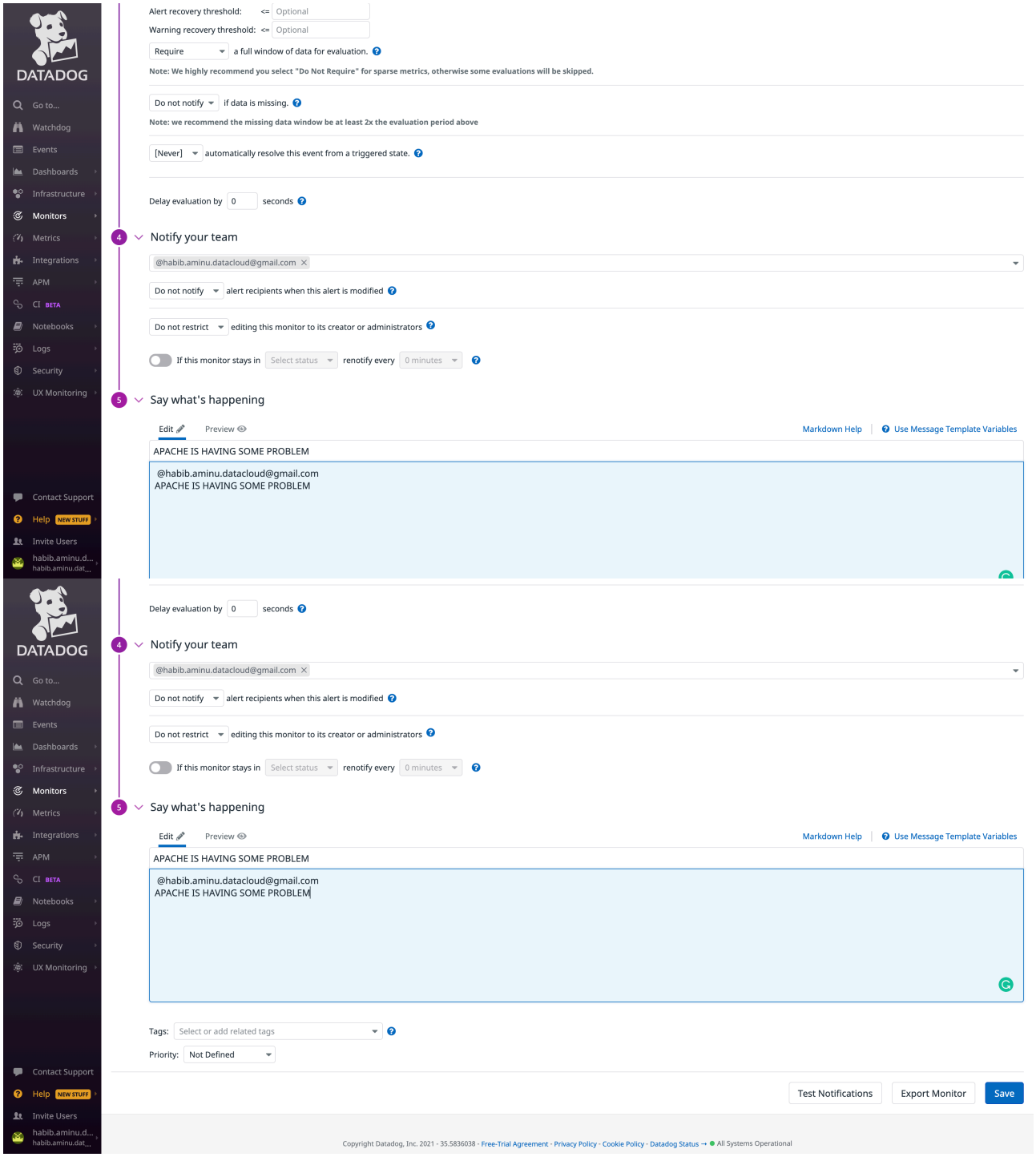


fig 4.24 Apache Req/Sec Configuration Monitor

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

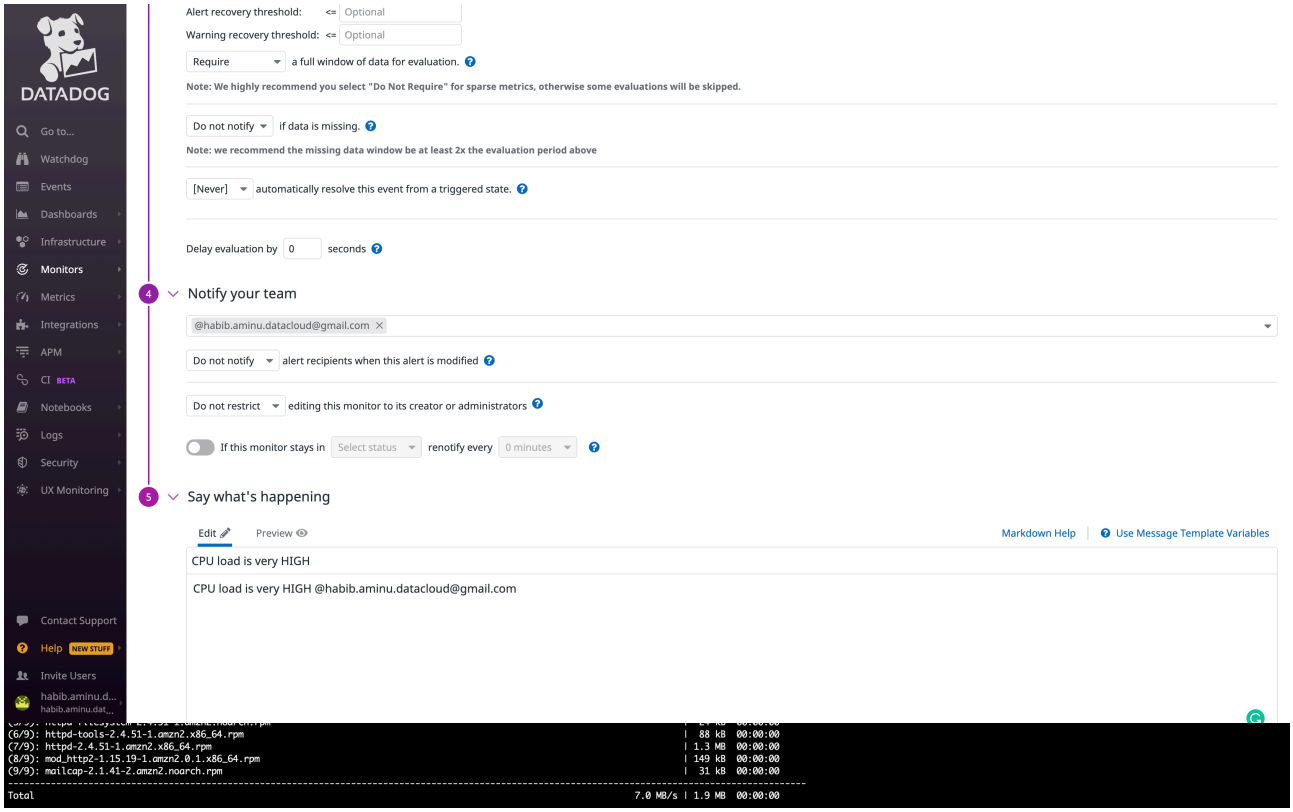


fig 4.25 CPU Workload Monitor

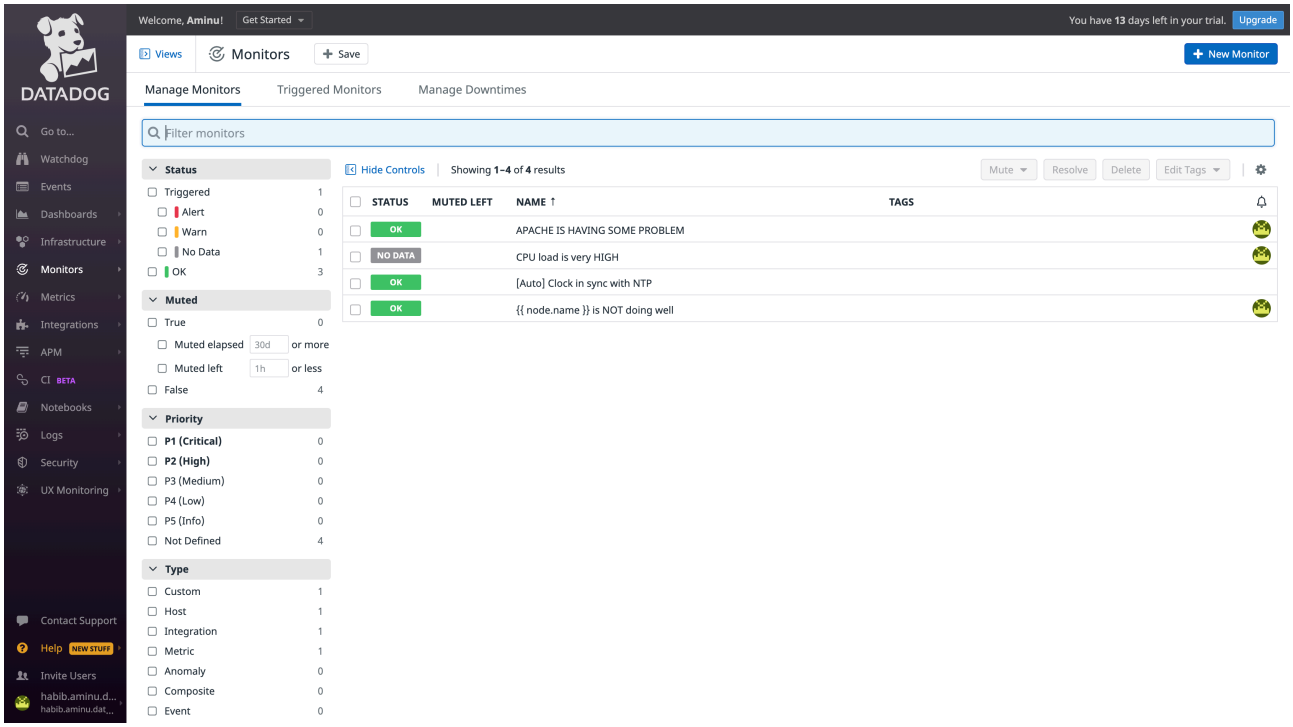


fig 4.26 Available Monitors

- 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

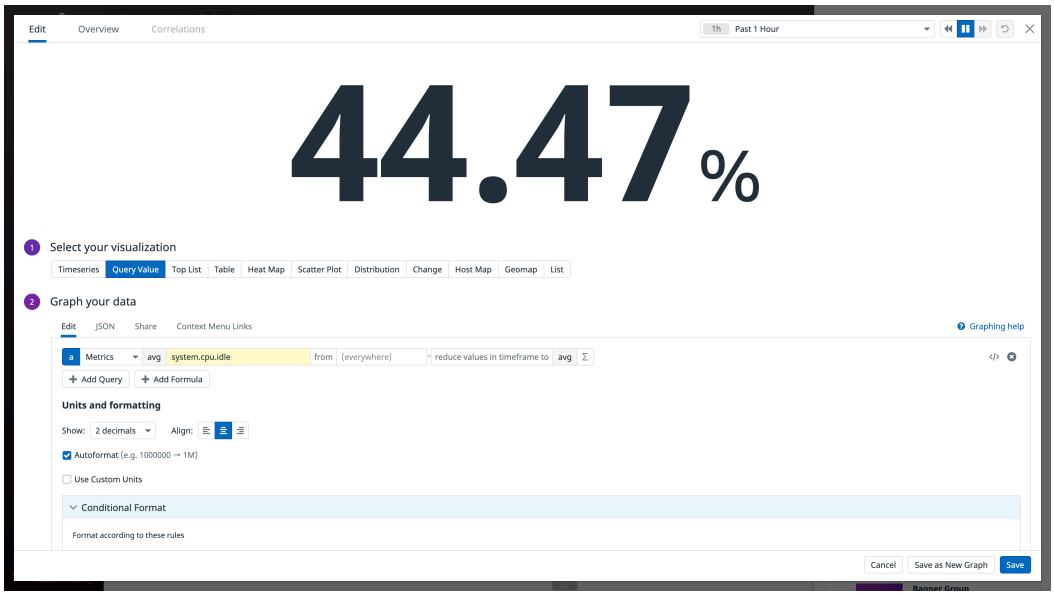


fig 4.27 CPU Workload Query Value

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

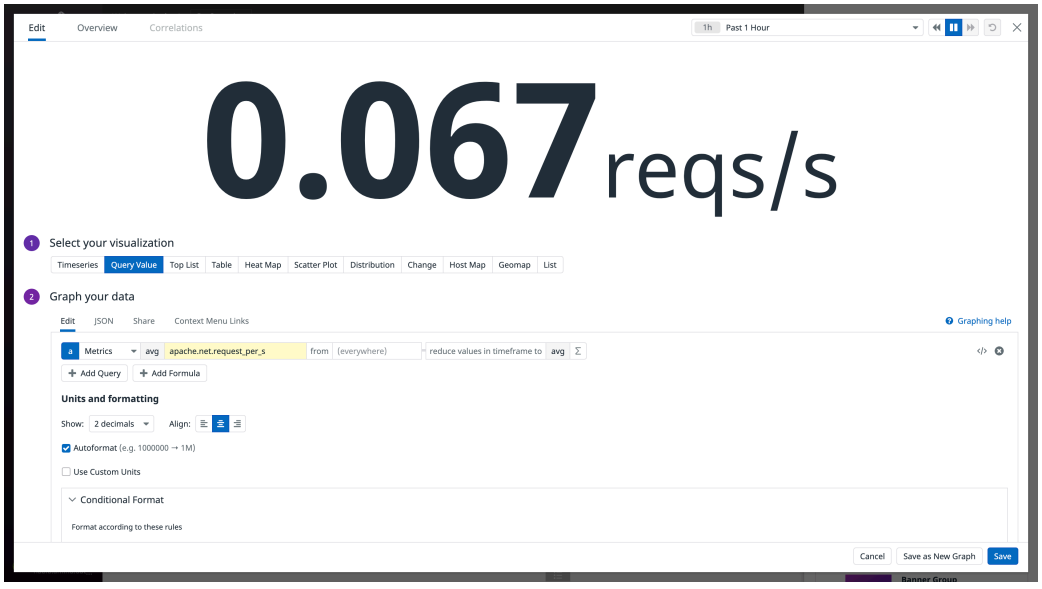


fig 4.28 Apache Req/Sec Query Value Dashboard

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

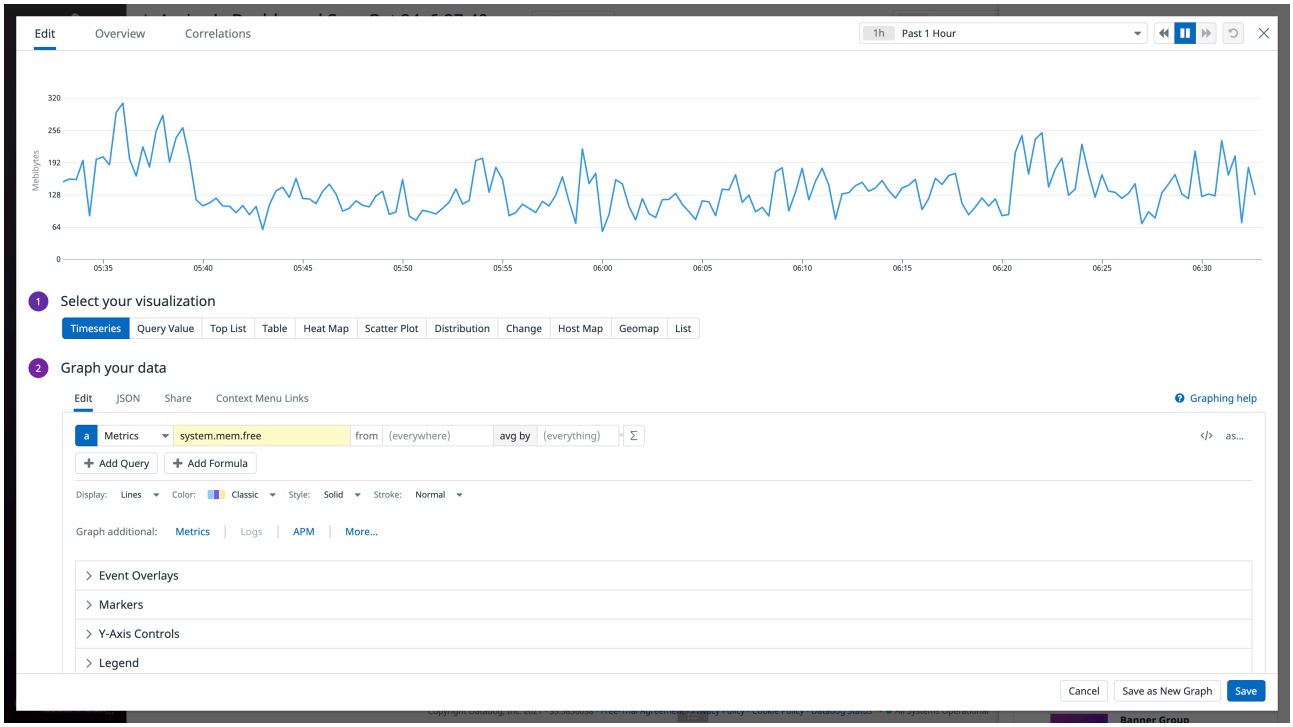


fig 4.29 Available Host Memory Time Series Dashboard

- Drag Top List

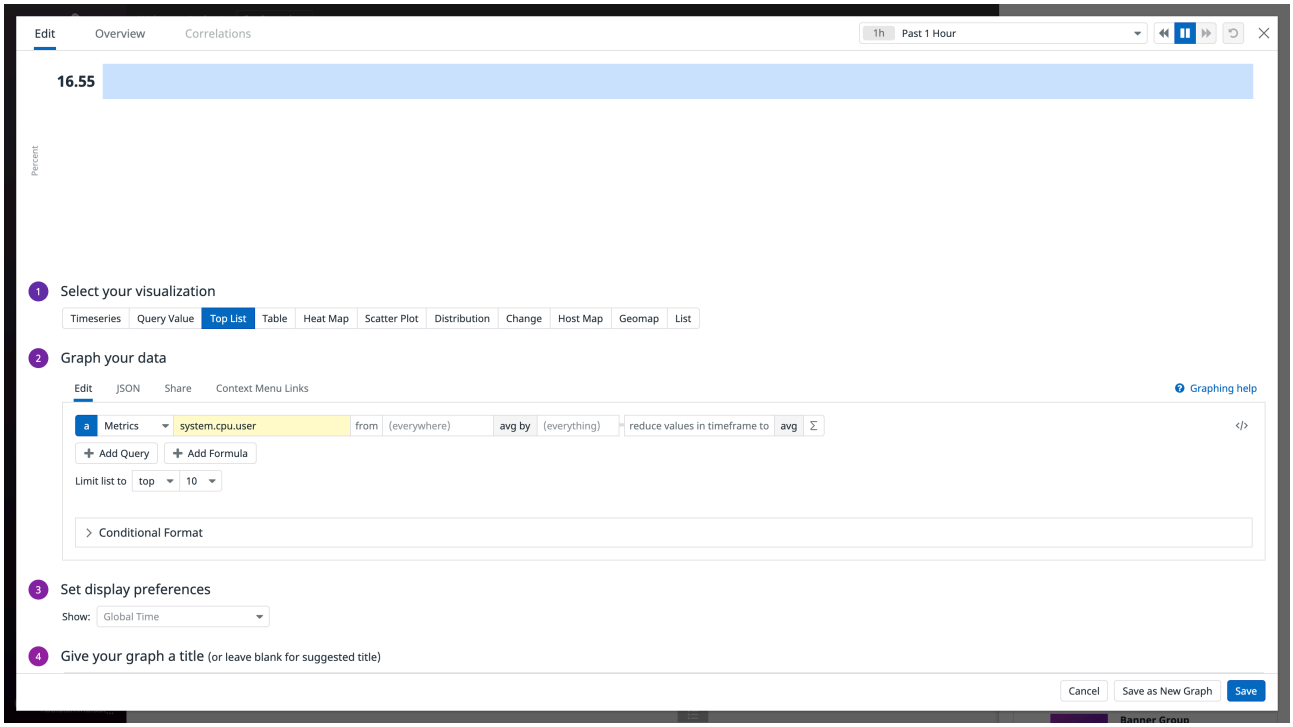


fig 4.30 CPU User

- Click **Save**

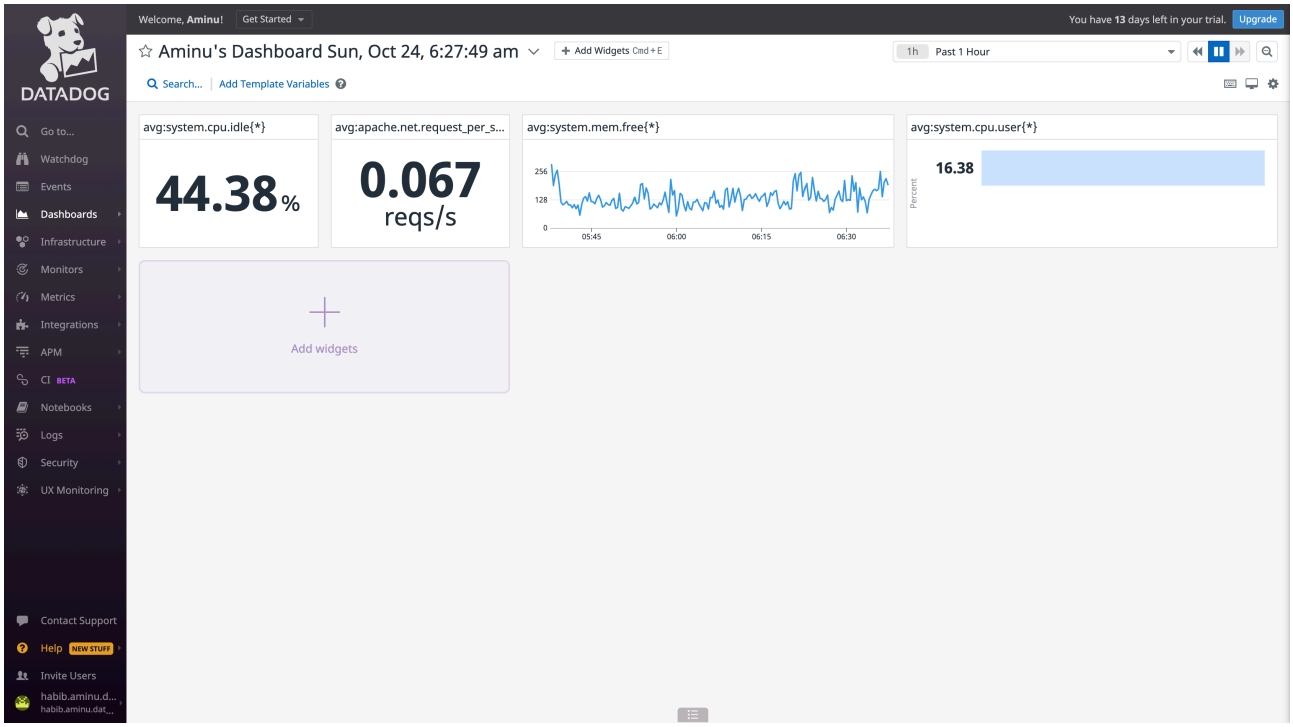


fig 4.31 Dashboard

- 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.

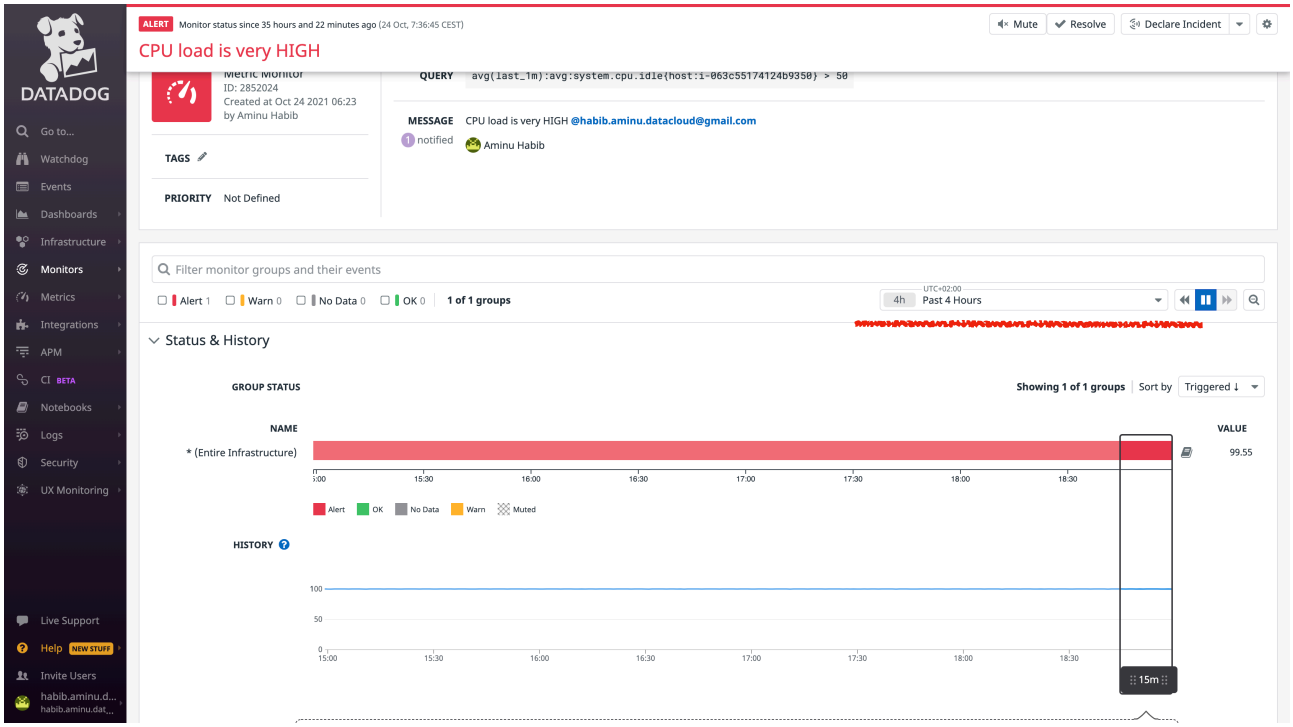


fig 5.1 CPU Metric Monitor

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

Thanks again for your patience!