


Large Scale Computing

Oscar Teeninga

1. Stworzenie i połączenie z instancją AWS

Zacząłem od stworzenia AMI. Wybrałem Ubuntu Server 20.04 LTS.



Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-083654bd07b5da81d (64-bit x86) / ami-04fe9398b2a27a600 (64-bit Arm)

Free tier eligible

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select

☒ 64-bit (x86)

☐ 64-bit (Arm)

Następnie wybrałem rodzaj instancji t2.micro

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	t3	t3.large	2	8	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel

Previous

Review and Launch

Next: Configure Instance Details

Skonfigurowałem grupę w taki sposób, aby był dostęp zarówno po SSH i HTTP

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: launch-wizard-1

Description: launch-wizard-1 created 2021-11-28T22:44:52.377+01:00

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Add Rule

Udało mi się połączyć (wygenerowałem wcześniej klucz)

```
[ubuntu@ip-172-31-91-148:~]$ oscar teeninga to ja
```

Zainstalowałem serwer HTTP

```
[ubuntu@ip-172-31-91-148:~]$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-11-28 21:49:51 UTC; 12s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2097 (apache2)
     Tasks: 55 (limit: 1154)
    Memory: 4.8M
    CGroup: /system.slice/apache2.service
            └─2097 /usr/sbin/apache2 -k start
              └─2099 /usr/sbin/apache2 -k start
                └─2100 /usr/sbin/apache2 -k start

Nov 28 21:49:51 ip-172-31-91-148 systemd[1]: Starting The Apache HTTP Server...
Nov 28 21:49:51 ip-172-31-91-148 systemd[1]: Started The Apache HTTP Server.
```

W widoku AWS stworzyłem AMI

Instances (1/2) Info

Connect

Instance state

Actions

Launch instances

Search

	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	-	i-0cb2ed05abe555a52	Terminated	t2.micro
<input checked="" type="checkbox"/>	-	i-0a9998aee64126560	Running	t2.micro

Connect

View details

Manage instance state

Instance settings

Networking

Security

Image and templates

Monitor and troubleshoot

Create image

Create template from instance

Launch more like this

Teraz AMI jest dostępne

Amazon Machine Images (AMIs) (1) Info

EC2 Image Builder

Actions

Launch instance from image

Owned by me

Search

	Name	AMI ID	AMI name	Source	Owner	Visibility
<input type="checkbox"/>	-	ami-09aadfed2aa851f04	oscarteeninga-image-http-server	365105900035/oscarteeninga-image-h...	365105900035	Private

2. Program

Sam kod nie pozostawia wątpliwości, dokładność pomiaru co do sekundy powinno wystarczyć.

```
2 import requests
3 import time
4 import boto3
5
6 ec2_client = boto3.client('ec2')
7 instance_id = 'i-0a9998aee64126560'
8
9 start_t = time.time()
10
11 res = ec2_client.start_instances(InstanceIds=[instance_id])
12 print(res)
13
14 while res != 'running':
15     time.sleep(0.1)
16     res_call = ec2_client.describe_instances(InstanceIds=[instance_id])
17     instance = res_call['Reservations'][0]['Instances'][0]
18     res = instance['State']['Name']
19     print(res)
20
21 print(res)
22 print("Start time:", time.time() - start_t, "s")
23
24 ip = instance['PublicDnsName']
25 print("IP Address:", ip)
26 f = requests.get('http://' + ip, allow_redirects=True)
27 print(f.content)
28
29 stop = ec2_client.stop_instances(InstanceIds=[instance_id])
30 print(stop)
```


Musiłem uzupełnić credsy w ~/.aws/credentials. Uruchomienie instancji trwało 56 sekund.

```
Start time: 56.16326570510864 s
IP Address: ec2-52-91-252-112.compute-1.amazonaws.com
```

Zatrzymanie instancji na końcu

```
{'StoppingInstances': [{'CurrentState': {'Code': 64, 'Name': 'stopping'}, 'InstanceId': 'i-0a9998aee64126560', 'Previous State': {'Code': 16, 'Name': 'running'}}], 'ResponseMetadata': {'RequestId': 'b3ba4df4-7336-408d-90ba-ce0b97556262', 'HT TPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'b3ba4df4-7336-408d-90ba-ce0b97556262', 'cache-control': 'no-cac he, no-store', 'strict-transport-security': 'max-age=31536000; includeSubDomains', 'content-type': 'text/xml; charset=UTF -8', 'content-length': '579', 'date': 'Sun, 28 Nov 2021 22:42:03 GMT', 'server': 'AmazonEC2'}, 'RetryAttempts': 0}}
```

Sprawdziłem również działanie strony apache


ubuntu

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf  
|
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/`