Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

Лабораторна робота №1

з дисципліни «Проектування розподілених систем»

за темою «Розгортання веб-порталу ввикористовуючи підхід IaaS»

**Виконав:**

Студент групи ІA-11мн

Новиков Данило Михайлович

**Перевірив:**

доц. Волокина Артем Миколайович

Київ 2022

**Тема:** Розгортання веб-порталу ввикористовуючи підхід IaaS.

**Завдання:**

* Застосунок має бути розгорнутим в Azure Cloud або в Digital Ocean на віртуальному сервері (VPS)
* Застосунок має розгортатися за допомогою копіювання скомпільованого коду на сервер за допомогою SFTP або SSH

**Хід роботи**

1. По-перше, була створена віртуальна машина зі стандартними налаштуваннями, використовуючи Amazon Linux. Для неї був створений SSH ключ та нова Security Group з можливістю доступу по SSH. Далі відображені скріншоти створення EC2 – віртуальної машини.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

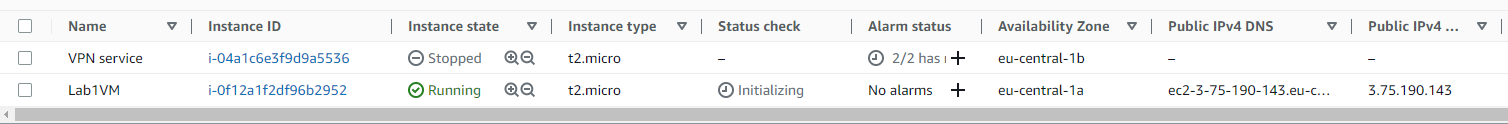
Description automatically generated

1. Дані файлу зі згенерованим SSH ключем

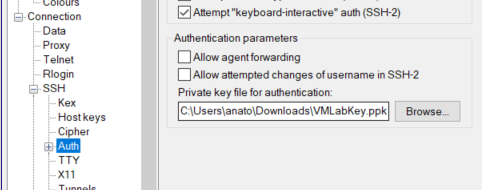
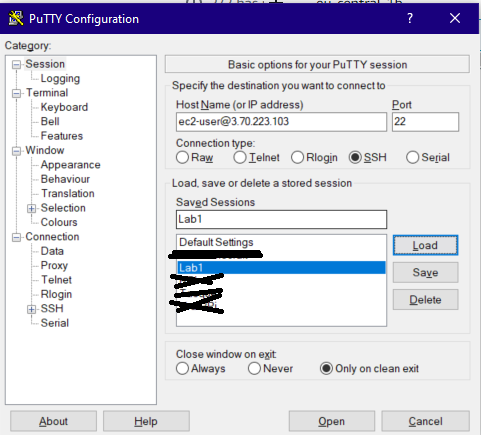
A screenshot of a computer

Description automatically generated with medium confidence

1. Ініціалізація створення віртуальної машини та отриманий результат при завершенні даного процесу.



1. Після того, як була створена віртуальна машина, використаємо Putty для підключення за SSH до неї. Для цього, введемо IP адресу отриманої машини та ім’я користувача ec2-user. Також, додамо наш отриманий файл з приватним ключем до авторизації.



1. Також, використаємо WinSCP як SFTP клієнт, введено ті самі дані.

Graphical user interface

Description automatically generated

1. Після цього, знайдемо мінімальний .NET проект. В даному випадку, був використаний проект з офіційного github репозиторію Azure, що використовує .NET та InMemoryDb.

Посилання на гітхаб проекту: <https://github.com/Azure-Samples/dotnet-core-api>

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Після локального запуску та перевірки, що все працює, запаблішимо проект у папку, в даному випадку, була використана папка res.

Text

Description automatically generated

1. Використовуючи WinSCP клієнт, перемістимо отриманий білд на віртуальну машину у папку lab1/Build.

Graphical user interface, text, application

Description automatically generated

1. Перейдемо до налаштування EC2. Для цього, встановимо dotnet-sdk-3.1. Але, оскільки по-дефолту даного пакету не існує, додамо посилання на репозиторій з пакетами дотнету. Після цього вже можна встановити sdk.

Graphical user interface, text

Description automatically generated

1. Перевіримо коректність встановлення, виконавши наступну команду:

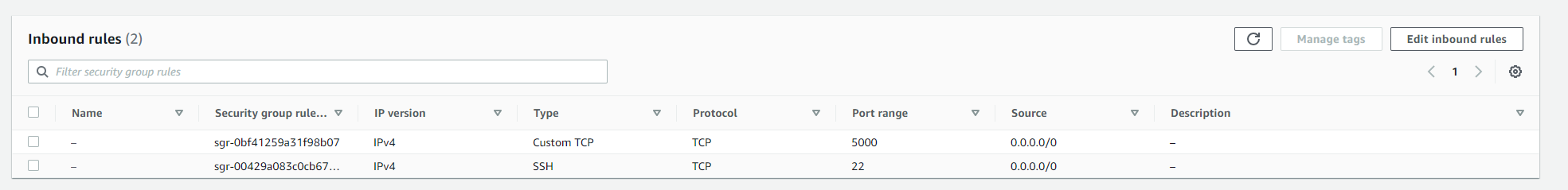


1. Запустимо застосунок, перейшовши у папку build та запустивши .dll файл.

Text

Description automatically generated

1. Додамо Inbound Security Rule для можливості достукатися до порту 5000 із зовні для віртуальної машини.



1. Перевіримо можливість отримання даних локально (на віртуальній машині) та за допомогою http запитів з комп’ютера користувача. Перший варіант у даному випадку буде працювати без проблем (використовуючи команду wget). А ось другий буде повертати помилки через неможливість підконектитись до серверу. Щоб прибрати цю помилку, додаємо строчку .UseUrls(“http://0.0.0.0:5000”) до Program.cs.

Text

Description automatically generated

1. Перезапустимо застосунок

Text

Description automatically generated

1. Перевіримо роботу роутів, використовуючи Insomnia.

Graphical user interface, application, Teams

Description automatically generated Graphical user interface, text, application, Teams

Description automatically generated Graphical user interface, application, Teams

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, application, Teams

Description automatically generated

1. В результаті був отриманий працюючий застосунок, що захощений на AWS EC2 використовуючи підхід IaaS.

**Висновок:**

В результаті виконання лабораторної роботи було розгорнуто застосунок на AWS EC2 за допомогою підключення через SSH для налаштування та підключення по SFTP для завантаження заздалегідь підготовленого застосунку.