Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи магістра

на тему: «Проєкт хмарної лабораторії на основі платформи MicroStack/OpenStack»

Виконала студентка 2-го курсу групи КІм-22-1
спеціальності 123 «Комп'ютерна інженерія»
_____ О.С. Головня

Керівник професор кафедри комп'ютерної інженерії та кібербезпеки, доктор технічних наук, доцент
_____ В.В. Воротніков

Рецензент доцент кафедри комп'ютерної інженерії та кібербезпеки, доктор технічних наук, доцент
_____ Ю.Б. Бродський

Житомир – 2023

Державний університет «Житомирська політехніка»

Факультет інформаційно-комп'ютерних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки

Андрій Єфіменко

13 жовтня 2023 р.

ЗАВДАННЯ

на випускну кваліфікаційну роботу магістра

Студент: Головня Олена Сергіївна

Тема роботи: **«Проєкт хмарної лабораторії на основі платформи MicroStack/OpenStack»**

затверджена Наказом університету від 13 жовтня 2023 р. № 571/с

Термін здачі студентом закінченої роботи 11 грудня 2023 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

1. **Об'єктом** дослідження є процес розроблення хмарної мережної інфраструктури та хмарних лабораторій.
2. **Предметом** дослідження є технології та засоби розроблення хмарної мережної інфраструктури та хмарних лабораторій.

Обладнання та програмне забезпечення:

1. Ноутбук HP 250 G5 з ОС Ubuntu Desktop 22.04 LTS (для розгортання проєкту хмари).

2. Хмарне програмне забезпечення Microstack Ussuri (на базі OpenStack).

3. Операційні системи CirrOS та Ubuntu Server 22.04 LTS (хмарний образ) для хмарних інстансів.

4. Фізичне обладнання наявної домашньої мережі (маршрутизатор Tp-Link Archer C20, тестова робоча станція з ОС Windows 10, тестова робоча станція з ОС macOS Monterey).

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1			
2			
3			

Студент _____ О.С. Головня

Керівник _____ В.В. Воротніков

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літературних джерел. Постановка мети та завдань дослідження	15.10.2023 р.	*виконано*
2	Вибір технології для реалізації	17.10.2023 р.	*виконано*
3	Розгортання тестового прототипу	21.10.2023 р.	*виконано*
4	Розроблення моделі віртуальної мережної інфраструктури для проєкту хмарної лабораторії	15.11.2023 р.	*виконано*
5	Налаштування та тестування проєкту хмарної лабораторії	07.12.2023 р.	*виконано*
6	Завершення оформлення пояснювальної записки	11.12.2023 р.	*виконано*

Студент _____ О.С. Головня

Керівник _____ В.В. Воротніков

АНОТАЦІЯ

Кваліфікаційна робота складається з проєкту хмарної лабораторії для навчання ОС Linux на базі MicroStack/OpenStack та пояснювальної записки до нього. Пояснювальна записка до кваліфікаційної роботи складається з анотації, вступу, трьох розділів, висновків, списку використаної літератури та додатків. Текст пояснювальної записки викладено на 73 сторінках та містить 64 ілюстрації та 3 таблиці. Список використаної літератури налічує 37 найменувань.

Метою кваліфікаційної роботи є розроблення проєкту хмарної лабораторії на основі платформи MicroStack/OpenStack.

Досліджено теоретичні аспекти побудови хмарної лабораторії. Визначено основні підходи до організації хмарної інфраструктури і розглянуто основні хмарні топології. Здійснено огляд хмарних платформ для розгортання хмарної інфраструктури (Amazon EC2, Google Compute Engine, Azure Virtual Machines; Eucalyptus, OpenStack, CloudStack, OpenNebula).

Виконано проєктування хмарної лабораторії. Сформульовано вимоги до проєкту хмарної лабораторії. Здійснено порівняльний аналіз можливостей популярних хмарних платформ та спеціалізованого вільно поширюваного хмарного програмного забезпечення. Внаслідок порівняльного аналізу наявних платформ для реалізації проєкту хмарної лабораторії обрано рішення MicroStack/OpenStack.

Здійснено налаштування та тестування проєкту хмарної лабораторії. Побудовано хмарну мережну інфраструктуру, яка забезпечує доступ до хмарних інстансів з інших фізичних пристроїв мережі. Запущено хмарні інстанси (ОС CirrOS та ОС Ubuntu Server). Перевірено працездатність мережного з’єднання з інстансами, а також доступ до інстансу за протоколом SSH.

Розроблений проєкт хмарної лабораторії може бути покладено в основу робочої хмарної лабораторії для навчання ОС Linux у Державному університеті “Житомирська політехніка”.

КЛЮЧОВІ СЛОВА: ХМАРНА ЛАБОРАТОРІЯ, ХМАРНА ІНФРАСТРУКТУРА, LINUX, OPENSTACK, MICROSTACK, IAAS.

ANNOTATION

The qualification work consists of a MicroStack/OpenStack-based project of the cloud laboratory for teaching the Linux operating system and an explanatory note. The explanatory note of the qualification work contains an annotation, introduction, three sections, conclusions, a list of references, and appendices. The text of the explanatory note is presented on 73 pages and includes 64 illustrations and 63 tables. The list of references consists of 37 items.

The purpose of the qualification work was to develop a MicroStack/OpenStack-based cloud laboratory project.

The work researches theoretical aspects of cloud laboratory development. The main approaches were determined, and the basic cloud topologies were considered. The study gives an overview of cloud platforms for cloud infrastructure deployment (Amazon EC2, Google Compute Engine, Azure Virtual Machines; Eucalyptus, OpenStack, CloudStack, OpenNebula).

The design of the cloud laboratory was created. The work formulates the requirements for the project of the cloud laboratory. A comparative analysis of features of popular cloud platforms and specialized free and open-source cloud software has been done. As a result of the comparative analysis, the MicroStack/OpenStack solution was selected for the cloud laboratory implementation.

Configuring and testing of the cloud laboratory project had been done. The cloud network infrastructure was built to provide access to cloud instances from the other physical devices connected to the same physical network. The cloud instances were launched (CirrOS and Ubuntu Server). The network connection with instances was checked, as well as the access to the instances via SSH protocol.

The project of the cloud laboratory could provide the basis for a working cloud laboratory for teaching the Linux operating system at Zhytomyr Politechnic State University.

KEYWORDS: CLOUD LABORATORY, CLOUD INFRASTRUCTURE, LINUX, OPENSTACK, MICROSTACK, IAAS.

ЗМІСТ

ВСТУП.....5

РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ ПОБУДОВИ ХМАРНОЇ ЛАБОРАТОРІЇ.....8

1.1 Хмарні технології.....8

1.2 Хмарні лабораторії.....11

1.3 Основні підходи до організації хмарної інфраструктури та хмарні топології.....12

1.4 Огляд хмарних платформ для розгортання хмарної інфраструктури.....15

РОЗДІЛ 2 ПРОЄКТУВАННЯ ХМАРНОЇ ЛАБОРАТОРІЇ.....24

2.1 Вимоги до проєкту віртуальної хмарної лабораторії.....24

2.2 Добір хмарної платформи для проєкту хмарної лабораторії.....25

2.3 Топологія та схема адресації моделі комп'ютерної мережі.....29

РОЗДІЛ 3 НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ ПРОЄКТУ ХМАРНОЇ ЛАБОРАТОРІЇ.....32

3.1 Розгортання проєкту хмарної лабораторії.....32

3.2 Перевірка роботи проєкту хмарної лабораторії.....56

ВИСНОВКИ.....64

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....65

ДОДАТКИ.....69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AWS – Amazon Web Services

GCP – Google Cloud Platform

HP – Hewlett-Packard

IaaS – Infrastructure as a Service – інфраструктура як сервіс

MS – Microsoft

PaaS – Platform as a Service – платформа як сервіс

RHEL – Red Hat Enterprise Linux

SaaS – Software as a Service – програмне забезпечення як сервіс

VM – virtual machine – віртуальна машина

ОС – операційна система

ПЗ – програмне забезпечення

ПК – персональний комп’ютер

ВСТУП

Актуальність теми. Хмарні технології (хмарні обчислення, англ. cloud computing) набули значного поширення протягом останніх двох десятків років. По суті, хмарні технології є продовженням клієнт-серверної технології, але з максимальним відокремленням клієнта від використовуваних ним фізичних потужностей, які абстрагуються у вигляді хмари. За допомогою хмари може надаватися доступ як до окремих програмних застосунків, так і до цілої мережної інфраструктури клієнта. На думку Е. Таненбаума та Х. Боса [32], хмарні технології є «можливо, найважливішим та найактуальнішим із сучасних застосувань віртуалізації». Серед факторів популярності хмарних технологій – оплата лише використаного обсягу ресурсів, гнучкість у виборі місця фізичного розташування обчислювальних потужностей, висока масштабованість та, потенційно, можливість забезпечення високої надійності та захищеності. На початку 2020-х подальшому розповсюдженню хмарних сервісів також сприяла пандемія Covid-19, а для багатьох українських організацій рушієм для міграції в хмару стала повномасштабна російсько-українська війна і прагнення захистити свою інфраструктуру, сервіси та дані від фізичної втрати і хакерських атак. Усе перелічене вище робить хмарні мережні інфраструктури важливим напрямом досліджень, а здатність обирати засоби для розроблення такої інфраструктури, проєктувати, розгортати, налаштовувати її та керувати нею є необхідною навичкою для сучасних комп'ютерних інженерів. Окремий інтерес становлять хмарні лабораторії, які через спеціальний інтерфейс користувача (часто – веб-інтерфейс) забезпечують досвід практичної взаємодії з об'єктами, що вивчаються. Хмарні лабораторії можуть мати найрізноманітніше освітнє, технічне та наукове застосування. Наприклад, на основі хмарної інфраструктури може бути розроблено хмарні лабораторії для навчання операційних систем, зокрема ОС Linux.

Метою кваліфікаційної роботи є розроблення проєкту хмарної лабораторії на основі платформи MicroStack/OpenStack.

Визначена мета обумовлює наступні завдання:

- дослідити теоретичні аспекти побудови хмарних лабораторій та хмарної мережної інфраструктури;
- обрати хмарну платформу для проєкту хмарної лабораторії;
- розробити проєкт хмарної лабораторії;
- здійснити налаштування та тестування проєкту хмарної лабораторії.

Об'єктом дослідження є процес розроблення хмарної мережної інфраструктури та хмарних лабораторій.

Предметом дослідження є технології та засоби розроблення хмарної мережної інфраструктури та хмарних лабораторій.

У процесі виконання кваліфікаційної роботи було розроблено проєкт хмарної лабораторії для навчання ОС Linux на базі платформи MicroStack/OpenStack. Розроблено модель віртуальної мережної інфраструктури для хмарної лабораторії. Здійснено розгортання проєкту хмарної лабораторії на базі одновузлової хмарної інфраструктури. Виконане тестування розробленого проєкту хмарної лабораторії з під'єднанням віртуальної мережної інфраструктури до реальної фізичної мережі. Проаналізовано можливості застосування та подальшого розвитку хмарної лабораторії.

Апробація результатів:

1. Holovnia O. Linux online virtual environments in teaching operating systems. Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. 2020. Vol. 2. pp. 964-973.

2. Holovnia O., Oleksiuk V. Selecting cloud computing software for a virtual online laboratory supporting the Operating Systems course. Proceedings of the 9th Workshop on Cloud Technologies in Education. 2021. pp. 216-227.

3. Головня О. С., Магурін О. О., Дехтяр М. С. Добір віртуалізаційного програмного забезпечення для прототипу хмарної мережної інфраструктури. Тези доповідей VI Всеукраїнської науково-технічної конференції «Комп’ютерні технології: інновації, проблеми, рішення», м. Житомир, 29-30 листопада 2023 р. – Житомир: Житомирська політехніка, 2023.

РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ ПОБУДОВИ ХМАРНОЇ ЛАБОРАТОРІЇ

1.1 Хмарні технології

Відповідно до означення Національного інституту стандартів та технологій, США (National Institute of Standards and Technology, NIST) [31], хмарні технології (хмарні обчислення, cloud computing) – це модель, що дозволяє повсюдний, зручний мережний доступ на вимогу (англ. on-demand access) до області спільно використовуваних налаштовуваних обчислювальних ресурсів (мереж, серверів, сховищ, застосунків, сервісів тощо), які може бути швидко надано та швидко вивільнено з мінімальними управлінськими затратами та мінімальною взаємодією з постачальником сервісу.

Основні характеристики хмарних технологій, за NIST, є наступними [31]:

– *самообслуговування на вимогу* (on-demand self-service) – споживач може власноруч визначати та змінювати обчислювальні потужності на зразок серверного часу чи обсягу мережного сховища, за допомогою автоматичних засобів, не маючи потреби безпосередньо взаємодіяти з представником постачальника ресурсів;

– *універсальний мережний доступ* (broad network access) – обчислювальні потужності доступні по мережі через стандартні механізми, тому для доступу до цих потужностей може бути використано різноманітні пристрої, зокрема ПК, ноутбуки, планшети, смартфони тощо;

– *об'єднання ресурсів* (resource pooling) – обчислювальні ресурси провайдера для обслуговування багатьох споживачів об’єднуються на основі багатокористувацької моделі; користувач не зобов’язаний турбуватися про географічне розміщення використовуваних ним ресурсів, проте нерідко може вибирати країну та регіон, у яких розташовуватимуться ці ресурси;

– *еластичність* (rapid elasticity) – обчислювальні потужності динамічно виділяються та вивільняються згідно із запитами користувача, і у деяких випадках обсяг ресурсів може збільшуватися і зменшуватись автоматично;

– *вимірюваність обсягів використання сервісу* (measured service) – автоматичний контроль та оптимізація обсягів споживання ресурсів, зазвичай з оплатою за принципом *pay-per-use* (оплата лише використаного обсягу ресурсів) та з наданням користувачу можливості відстежування обсягів використаних ним ресурсів.

Виокремлюють три основні моделі надання хмарних послуг.

SaaS (Software as a Service – програмне забезпечення як сервіс). Споживач використовує програмні засоби постачальника, які працюють у хмарі. Робота з такими програмними засобами часто доступна з різних пристроїв через веб-переглядач або спеціалізовані застосунки. Споживач не має контролю над серверами, операційними системами, сховищами та іншими складовими інфраструктури, у якій працює використовуваний ним програмний засіб, і може змінювати обмежений набір користувацьких налаштувань цього програмного засобу.

Приклади: Gmail, Google Docs, Dropbox, Spotify, Pictochart та багато інших.

PaaS (Platform as a Service – платформа як сервіс). Споживач розміщує у хмарній інфраструктурі власні програмні засоби (розроблені власноруч або на замовлення, використовувані за вільними ліцензіями, придбані тощо). Для розгортання програмних засобів споживачеві надаються мови програмування, бібліотеки, сервіси та інші інструменти. Як це було і для SaaS, у PaasS споживач не має контролю над інфраструктурою, натомість у PaaS споживач може керувати програмними засобами, що розгортаються, а також налаштовувати середовище для роботи цих програмних засобів.

Приклади: AWS Elastic Beenstalk, Microsoft Azure Cloud Services, Google App Engine, Oracle Cloud PaaS, Open Shift, Force.com та ін.

IaaS (Infrastructure as a Service – інфраструктура як сервіс). Споживачеві надаються засоби для опрацювання та зберігання даних, мережні інструменти та інші базові обчислювальні ресурси, на основі яких споживач може розміщувати довільне ПЗ. Хоча споживач не керує всією хмарною інфраструктурою, він має контроль над значною частиною цієї інфраструктури, наприклад, операційними системами, сховищами, програмними засобами, а також, можливо, контроль над окремими складовими мережі.

Приклади: Amazon EC2, Google Compute Engine, GoGrid, Microsoft Azure Virtual Machines та ін., а також так зване хмарне ПЗ (OpenStack, CloudStack, OpenNebula, Eucalypus та ін.).

Хмарна інфраструктура (не лише у моделі IaaS) може розгортатися відповідно до чотирьох моделей розгортання: приватна хмара, громадська хмара, публічна хмара та гібридна хмара.

Приватна хмара (private cloud) використовується лише тією організацією, для потреб якої її було розгорнуто. Водночас, приватна хмара може бути власністю цієї організації, а може належати хмарному провайдеру, а організації постачатися як послуга.

Громадська хмара (спільна хмара, community cloud) може використовуватися кількома організаціями, якщо ці організації мають спільні потреби.

Публічна хмара (public cloud) є загальнодоступною, хоч і не обов'язково безкоштовною.

Гібридна хмара (hybrid cloud) поєднує дві та більше хмарних інфраструктур різних моделей.

Хмарні сервіси можуть надаватися як безкоштовно, так і платно. Розповсюдженим є варіант, коли базовий тарифним план доступний безоплатно, натомість додатковий функціонал уже потребує оплати. Також існує підхід, за якого сервіс платний, але новим користувачам надається безкоштовний пробний період

(від одного до кількох місяців, рік тощо). Наприклад, подібні умови пропонують сервіси моделі IaaS (наприклад, Amazon EC2, Google Compute Engine, Azure Virtual Machines).

1.2 Хмарні лабораторії

Хмарна лабораторія може бути означена як “цифрова система, у якій завдяки віртуалізації та вебінтерфейсу формуються мережні віртуальні об’єкти, що відповідають реальним комп’ютерним системам” [37].

Тобто хмарна лабораторія передбачає наявність певних віртуальних середовищ, доступ до яких здійснюється за допомогою хмарних технологій. Розміщення сервісів у хмарі, і створення у хмарі повноцінної інфраструктури є популярним сучасним напрямом. Водночас, хмарні лабораторії вирізняє дослідницьке і (або) освітнє застосування.

Хмарні лабораторії застосовуються для навчань та досліджень не лише у напрямку інформаційних технологій, а й з фізики, хімії, біології, астрономії, математики, не оминаючи й соціально-гуманітарних дисциплін, у яких також є місце для експериментальних досліджень, спільної навчальної діяльності та спільних дослідницьких проєктів. Хмарні лабораторії дають змогу працювати з предметами та явищами, що вивчаються, за умови, коли реальний доступ до них є складним, має зависоку вартість, є небезпечним для здоров’я та життя або наразі неможливий. Завдяки хмарній лабораторії одержують можливість спільного навчання та досліджень здобувачі освіти, які перебувають у різних географічних локаціях. Окремий різновид хмарних лабораторій — автоматизовані та роботизовані лабораторії з віддаленим доступом для дослідників за принципом хмарного сервісу, які можуть використовуватися, наприклад, у сучасній біології [2], але такі лабораторії лежать поза межами предмету даної кваліфікаційної роботи.

Важливо відзначити, що поняття хмарної лабораторії не обов’язково передбачає доступ до елементів хмарної мережної інфраструктури (зокрема, віртуальних машин) відповідно до моделі IaaS. Послуги користувачам хмарної ла-

бораторії можуть також надаватися і у межах моделі SaaS, і моделі PaaS, і у межах інших гібридних моделей.

Крім того, хмарна лабораторія, як і хмара загалом, може мати різні моделі розгортання, розглянуті у п. 1.2 (приватна хмара, громадська хмара, публічна хмара та гібридна хмара).

Хмарна інфраструктура, що лежить в основі хмарної лабораторії, може належать провайдеру хмарних послуг або належати організації, яка використовує хмарну. Залежно від кількості використаних у хмарі серверів, хмарна лабораторія також може спиратися на одновузлову чи багатовузлову топологію. У наступному параграфі розглянемо перелічені підходи та топології детальніше.

1.3 Основні підходи до організації хмарної інфраструктури та хмарні топології

Основні підходи до організації хмарної інфраструктури. Належний контроль над хмарною інфраструктурою надають хмарні сервіси моделі IaaS – інфраструктура як сервіс. Виокремимо два основні підходи до організації такої інфраструктури:

– використання для побудови хмарної інфраструктури платформ, наданих провайдером хмарних сервісів (Amazon EC2, Google Compute Engine, Azure Virtual Machines та ін.);

– використання для побудови хмарної інфраструктури спеціалізованого хмарного ПЗ (OpenStack, CloudStack, OpenNebula, Eucalyptus та ін.).

Крім того, ці підходи можуть поєднуватися: частину інфраструктури може бути організовано за одним підходом, а частину – за іншим.

Перевагою застосування хмарних сервісів IaaS, наданих провайдером, є те, що про значну частину робіт дбає провайдер, зокрема про фізичне обладнання та його обслуговування, образи операційних систем тощо. Також клієнт платить за принципом pay-per-use, тобто лише за використаний обсяг ресурсів і зазвичай ли-

ше у ті періоди, коли ці ресурси безпосередньо споживаються (окрім сервісів, які передбачають оплату за моделлю підписки).

Захист інфраструктури також до певної міри забезпечується провайдером. Однак важливо розуміти, що тут хмарний провайдер може відповідати лише за надання безпекових механізмів (наприклад, реалізувати доступ до віртуальних машин з ОС Linux на основі протоколу SSH), проте провайдер не може нести відповідальність за некоректні політики безпеки своїх клієнтів чи за недотримання цих політик. Так, у прикладі з SSH-з'єднанням з віртуальною машиною провайдер не може проконтролювати, аби клієнт не передавав ключ для доступу стороннім особам – випадково чи умисно. З іншого боку, дискусії щодо безпеки використання хмарних сервісів також мають місце.

Іншою потенційною проблемою опертя на сервіси хмарних провайдерів є потенційна зміна умов використання. Це особливо стосується пільгових тарифних планів, які передбачають найменше зобов'язань провайдера перед клієнтом. Наприклад, так сталося у кінці 2010-х із сервісом Cloud9. Сервіс припинив надавати безкоштовні віртуальні контейнери з Linux, лишивши тільки можливість підключати наявні віртуальні машини з інших хмарних сервісів, а зрештою став частиною Amazon Web Services [3]. Описані зміни торкнулися й індивідуальних користувачів, і підписників тарифного плану для освітян. Останні більше не могли надавати Linux-контейнери з Cloud9 своїм студентам. Зміни відбулися посеред навчального семестру, що спричинило додаткові ускладнення.

У випадку використання спеціалізованого хмарного ПЗ для самостійної побудови хмарної інфраструктури маємо високий рівень контролю над даними, що добре для чутливих даних, вузькоспеціалізованих завдань (які складно або неможливо реалізувати на основі пропонованих провайдером сервісів), а також для досліджень у галузі хмарних технологій. Організація власної хмари надає значну свободу щодо деталей реалізації інфраструктури. Водночас, потрібно самостійно дбати про обладнання, програмне забезпечення та його обслуговування, і це потребує роботи кваліфікованого персоналу та інших капіталовкладень.

Як бачимо, обидва підходи мають свої переваги та недоліки. Тому на практиці також може застосовуватися їх поєднання. Крім того, існують платформи, які дозволяють орендувати фізичні віддалені сервери (bare-metal servers), хмарну інфраструктуру на базі визначеного хмарного програмного забезпечення (наприклад, на базі OpenStack).

Хмарні топології. Хмара може розгортатися відповідно одновузлової та багатовузлової топології. Ключова структурна відмінність між двома топологіями полягає у розміщенні основних ролей – керування (control), обчислення (compute) та зберігання (storage).

Одновузлова топологія (single-node topology) передбачає розміщення всієї інфраструктури на одному вузлі (сервері, ноутбуці, настільному ПК). Тобто всі три основні ролі (керування, обчислення, зберігання) розміщуються на одному комп'ютері. Така модель зазвичай використовується передусім для локального доступу і потребує менше ресурсів.

Багатовузлова топологія (multi-node topology) є результатом розвитку одновузлової топології, зростання потреб в обчислювальних потужностях, в дисковому просторі та в загальній стійкості інфраструктури. У багатовузловій топології зазвичай використовується принаймні три окремі вузли – для ролей керування, обчислення та зберігання. У високодоступних системах застосовується лише багатовузлова топологія.

На вибір хмарної топології впливають наявний обсяг ресурсів, розміри хмари та її призначення. Одновузлову хмарна топологія можна згодом розвинути у багатовузлову.

1.4 Огляд хмарних платформ для розгортання хмарної інфраструктури

У цьому підрозділі буде розглянуто низку популярних хмарних рішень відповідно до обох підходів до організації хмарної інфраструктури, досліджених у п. 1.3. Визначення критеріїв для порівняння, порівняльний аналіз та вибір на

основі його результатів хмарного рішення для реалізації даного проєкту здійснюватиметься у розділі 2.

Спершу розглянемо **платформи для побудови хмарної інфраструктури, надані провайдерами хмарних сервісів**. Серед найпопулярніших таких платформ, які пропонують послуги за моделлю IaaS, – Amazon EC2, Microsoft Azure Virtual Machines та Google Compute Engine (див. Додаток А).

Amazon EC2 (Amazon Elastic Compute Cloud) є частиною Amazon Web Services [31], відомих також під абревіатурою AWS. В Amazon EC2 окремі обчислювальні одиниці мають назву *інстанси* (англ. instances – досл. екземпляри, але зазвичай не перекладається). Доступні операційні системи Linux (Amazon Linux 2, Ubuntu, RHEL, CentOS, SUSE, Debian) та Windows (Windows Server). Сервіс пропонує інстанси багатьох типів різної потужності та за різну ціну (наприклад, інстанси t2.nano мають 1 віртуальний ЦП і 0,5 ГіБ оперативної пам'яті, а інстанси i4g.16xlarge – 64 віртуальних ЦП і 512 ГіБ оперативної пам'яті). Сервери розміщені у низці регіонів: US East, US West, Africa, Asia Pacific, Canada, Europe, Israel, Middle East, South America, що діляться на дрібніші регіони, наприклад, US West (N. California) та US East (Oregon). Від регіону залежить доступність сервісів (окремі сервіси можуть бути доступні не у всіх регіонах), перелік типів віртуальних машин, цінова політика, правові моменти. Нерідко компанії можуть приймати рішення розмістити свою інфраструктуру у більш ніж одному регіоні, задля підвищення надійності.

Для нових користувачів Amazon EC2 доступний рівень безкоштовного користування, AWS Free Tier, у межах якого надається 750 годин на місяць роботи інстансів типу t2.micro з ОС Linux та Windows. AWS Free Tier діє протягом 1 року. Важливо ретельно контролювати використання ресурсів інстансами, оскільки нарахування плати за їх використання за межами визначеного обсягу здійснюється автоматично. Питання відстежування обсягів користування та встановлення відповідних попереджень (alerts) важливе як для безкоштовних

послуг, так і для платних, а також бути актуальним і для решти розглянутих тут хмарних сервісів.

Microsoft Azure Virtual Machines доступні у межах хмарної платформи Microsoft Azure [35]. Окремі обчислювальні одиниці тут називають віртуальними машинами (virtual machines, VMs). Доступні машини з Linux (RHEL, SUSE, Ubuntu, CentOS, Debian, CoreOS) та Windows (Windows Server). Типи віртуальних машин також різняться за можливостями та вартістю користування: від найекономнішої A-серії (A-Series) до потужніших серій на зразок Ls-серії (Ls-Series) чи H-серії (H-Series). MS Azure Virtual Machines доступна в багатьох регіонах, зокрема Asia-Pacific, Australia, Brazil, Canada, Chine, Europe, France, Germany, India, Japan, Korea, Norway, South Africa, Sweden, Switzerland, United Kingdom, United States, United Arab Emirates та ін. Варто зазначити, що для Azure наведено назви так званих регіональних пар. Кожна з них містить два, чотири або більше окремих регіонів. Наприклад, регіональна пара для Європи – це North Europe (Ireland) та West Europe (Netherlands) [4].

Безкоштовний обліковий запис MS Azure пропонує по 750 годин на місяць роботи віртуальних машин з Windows та віртуальних машин з Linux (типи B1s, B2pts v2, B2ats) протягом 1 року користування. Як і у випадку з Amazon EC2, в MS Azure Virtual Machines важливо відстежувати обсяги користування, аби лишатися у межах безкоштовного тарифного плану або запланованого бюджету.

Google Compute Engine входять до складу Google Cloud Platform (GCP) [7]. Окремі обчислювальні одиниці на GCP називають віртуальними машинами (virtual machines) або інстансами (instances), терміни рівносильні. Серед типів віртуальних машин на GCP є, наприклад, серія загального призначення (C3, E2, N2, N2D, N1), серія для інтенсивних обчислювальних навантажень (C2, C2D) та ін. Сервіс доступний у багатьох регіонах, серед яких Asia East, Asia Northeast, Asia South, Asia Southeast, Australia Southeast, Europe Central, Europe North, Europe Southwest, Europe West, Middle East Central, Middle East West, North America

Northeast, South America East, South America West, US Central, US Central, US West (ці великі регіони, у свою чергу, діляться на велику кількість дрібніших регіонів).

Нові користувачі Google Cloud мають у своєму розпорядженні сервісів на 300 доларів протягом перших 90 днів. Також протягом одного місяця можна безкоштовно користуватися одним інстансом типу e2-micro.

Перейдімо до **платформ для побудови хмарної інфраструктури на основі спеціалізованого хмарного ПЗ**. Для огляду обрано чотири популярні хмарні платформи – Eucalyptus, OpenStack, CloudStack та OpenNebula.

Eucalyptus (Eucalyptus Systems) є хмарним ПЗ з відкритим кодом для побудови хмарної інфраструктури на основі моделей приватної та гібридної хмари [12]. Eucalyptus поширюється під ліцензією GNU GPL версії 3. Безкоштовна підтримка забезпечується спільнотою, доступна також платна підтримка.

Eucalyptus має високий рівень сумісності з AWS, зокрема підтримує подібні типи інстансів, AWS API, а також інструменти, використовувані AWS.

Eucalyptus підтримує завантаження з командного рядка готових образів з ОС Linux (CentOS, CentOS Atomic Host, Fedora, Fedora Core OS, Ubuntu). Дане віртуалізаційне ПЗ має інтерфейс командного рядка (для адміністрування), веб-інтерфейс для керування інстансами (рис. 1.1).

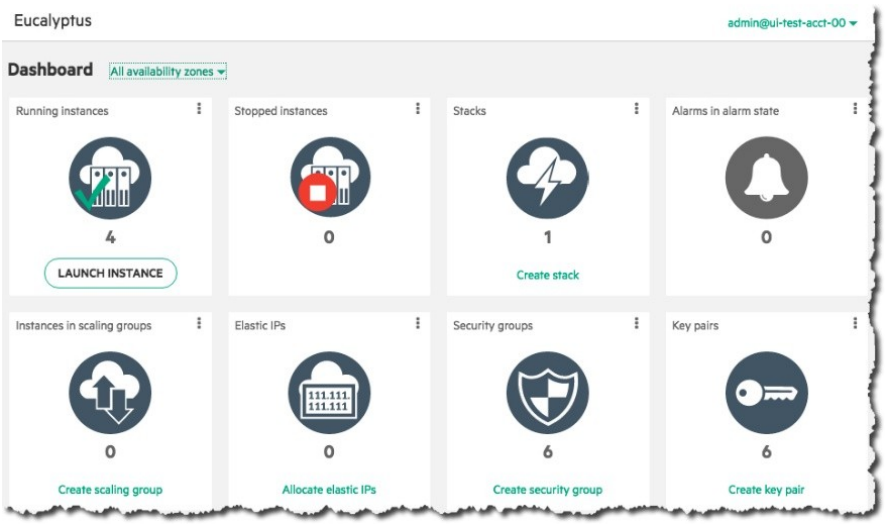


Рисунок 1.1 – Панель керування Eucalyptus

Підтримується з’єднання з інстансами засобами командного рядка за допомогою Euca2ools (для ОС Cent OS, RHEL, Mac OS X) [11] та AWS CLI (для ОС Linux, Windows, macOS та для Docker) [20].

Системні вимоги для Eucalyptus включають процесор Intel / AMD з чотирма ядрами по 2 ГГц кожне, принаймні 16 Гб оперативної пам’яті для кожної віртуальної машини, щонайменше 100 Гб місця на диску для кожного хоста. Водночас, рекомендовано мати 200 Гб на контролері вузів (Node Controller), де працюють віртуальні машини з Linux, 500 Гб на контролерах сховищ (Storage Controller), а також 500 Гб на шлюзі сховища об’єктів Walrus (за умови його використання). Машини, на яких працюватимуть компоненти (крім Cloud Controller та Node Controller), повинні підтримувати групову передачу UDP для IP-адреси 239.193.7.3. Дуже важливо врахувати, що всі сервіси встановлюються на фізичні комп’ютери, а не віртуальні машини [29].

Для початківців Eucalyptus пропонує рішення FastStart, яке розгортається на CentOS 7.9 і потребує кількох вільних IP-адрес у підмережі.

OpenStack (початкові розробники: Rackspace Hosting, NASA; подальша розробка: OpenStack Foundation, згодом – OpenInfra Foundation). OpenStack є вільно поширюваною платформою для хмарних обчислень, орієнтованою на відкриті стандарти. Поширюється під ліцензією Apache License версії 2.0.

OpenStack може працювати на базі ОС Windows, Solaris, Linux (зокрема CentOS, Debian, Fedora, openSUSE, RHEL, Ubuntu) та гіпервізора Vmware ESXi [33]. Дане хмарне ПЗ має веб-інтерфейс Horizon (рис. 1.2). OpenStack підтримує інструменти сторонніх розробників, зокрема Kubernetes, CloudFoundry, Terraform.

Системні вимоги до OpenStack залежать від призначення майбутньої хмари, обраних компонентів OpenStack та інших факторів. Наприклад, інсталяцію OpenStack на базі ОС Ubuntu може бути здійснено згідно з одним із трьох основних сценаріїв: одновузлове розгортання, багатовузлове розгортання та розгортання кластера дата-центру.

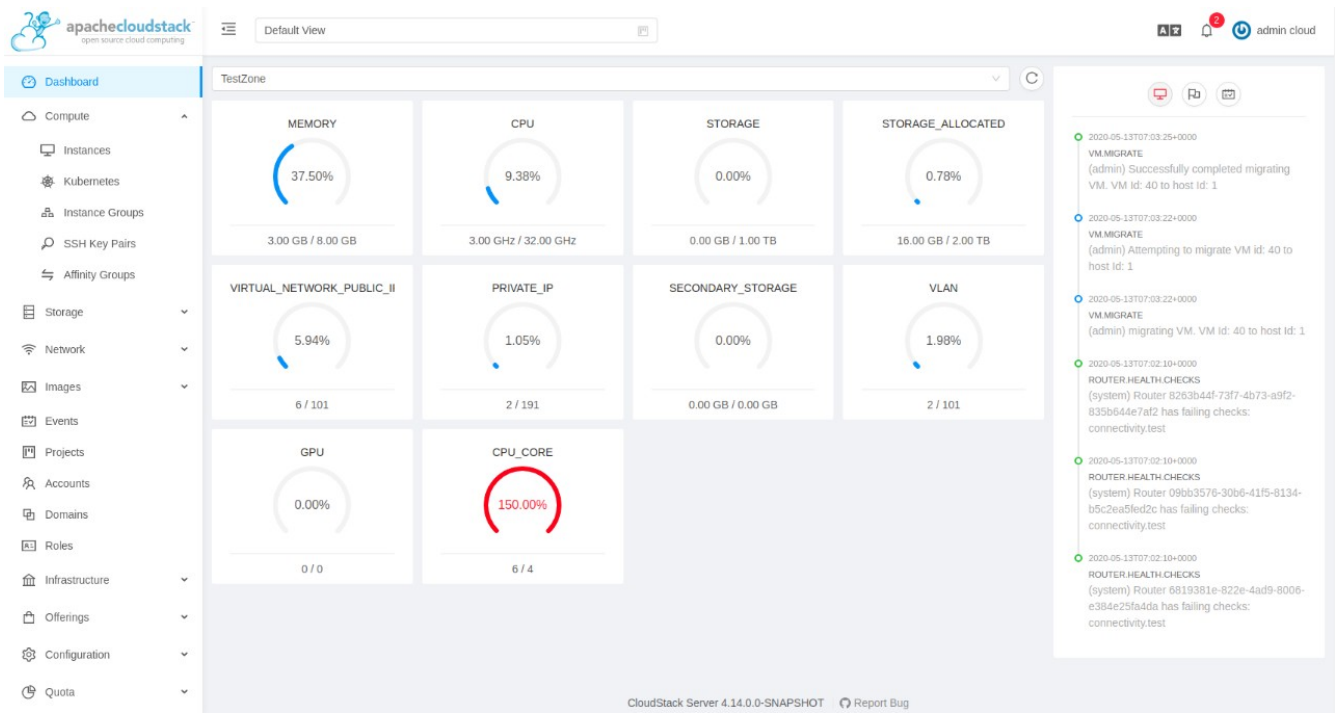


Рисунок 1.2 – Панель керування OpenStack

Залежно від обраного сценарію, OpenStack потребує принаймні 1 фізичної машини з 16 Гб оперативної пам’яті, багатоядерним процесором та 50 Гб вільного місця на диску (для одновузлового розгортання); 2 фізичні машини, кожна з 16 Гб оперативної пам’яті, багатоядерним процесором та 50 Гб місця на диску (для багатовузлового розгортання); 6 фізичних серверів, кожний з 8 Гб оперативної пам’яті, IPMI BMC, подвійними мережними адаптерами, підтримкою високодоступних архітектур, мережним комутатором та Інтернет-шлюзом (для розгортання кластера дата-центру) [16]. Мінімальна інсталяція для OpenStack Yoga (вийшла у березні 2022 року) складається з сервісу ідентичностей Keystone, сервісу образів Glance, сервісу розміщення Placement, обчислювального сервісу Nova, мережного сервісу Neurone, а також у більшості випадків включатиме інформаційну панель Horizon та сервіс блокових накопичувачів Cinder [17].

Для нових користувачів доступні сценарії Training Labs, які надають можливість автоматизованого розгортання кластеру на базі віртуальних машин в VirtualBox або KVM. Для роботи зі сценаріями Training Labs достатньо ПК або ноутбука

з основною ОС Linux, MacOS чи Windows, щонайменше 8 Гб оперативної пам’яті та 20 Гб вільного місця на диску [9].

Серед інших рішень OpenStack для початківців – MicroStack [22], а також одно-вузловий кластер на основі MicroStack на базі Sunbeam від компанії Canonical [18].

На базі MicroStack можна розгорнути невеликий одновузловий кластер з інстансами під управлінням CirrOS – дистрибутива Linux з низькими системними вимогами, спеціально для тестових образів. Системні вимоги до машини, на якій здійснюватиметься розгортання такого кластера, включають процесор з 2 ядрами (рекомендовано 4 ядра), 8 Гб оперативної пам’яті (рекомендовано 16 Гб), основна ОС Ubuntu (чи інший дистрибутив Linux з підтримкою технології Snap) [22]. Також рекомендовано мати 100 Гб вільного місця на диску, але можливий тестовий запуск з і меншим обсягом – все залежить від кількості інстансів.

Офіційний сайт Canonical пропонує більш вимогливе до ресурсів рішення на базі технологій MicroStack та Sunbeam. Таке рішення потребує машини з основною ОС Ubuntu 22.04, процесора з принаймні 4 ядрами, щонайменше 16 Гб вільної оперативної пам’яті (загалом потрібно більше) та мінімум 50 Гб вільного місця на диску [18].

CloudStack (початкові розробники: Cloud.com, подальша розробка: Apache Foundation) [1]. CloudStack – вільно поширюване хмарне ПЗ, розроблене для великих мереж, але також застосовне і до менших мереж. Розповсюджується під ліцензією Apache License версії 2.0.

CloudStack має веб-інтерфейс CloudStack UI (рис. 1.3), а також підтримку інтерфейсів AWS API та Open Cloud Computing Interface [36]. Мінімальні системі вимоги для сервера керування CloudStack, бази даних та сховища – 64-розрядний процесор архітектури x86, 4 Гб оперативної пам’яті, 250 Гб вільного місця на диску (рекомендовано – 500 Гб), один мережний адаптер, статична IP-адреса та повне доменне ім’я для кожного фізичного хоста. Сервер керування OpenStack

можна встановлювати на віртуальну машину. Фізичні комп'ютери, на яких працюватимуть гіпервізори і віртуальні машини, повинні мати 64-розрядні процесори архітектури x86 з підтримкою апаратної віртуалізації (Intel-VT або AMD-V), 4 Гб оперативної пам'яті, 36 Гб вільного місця на диску та один мережний адаптер [19].

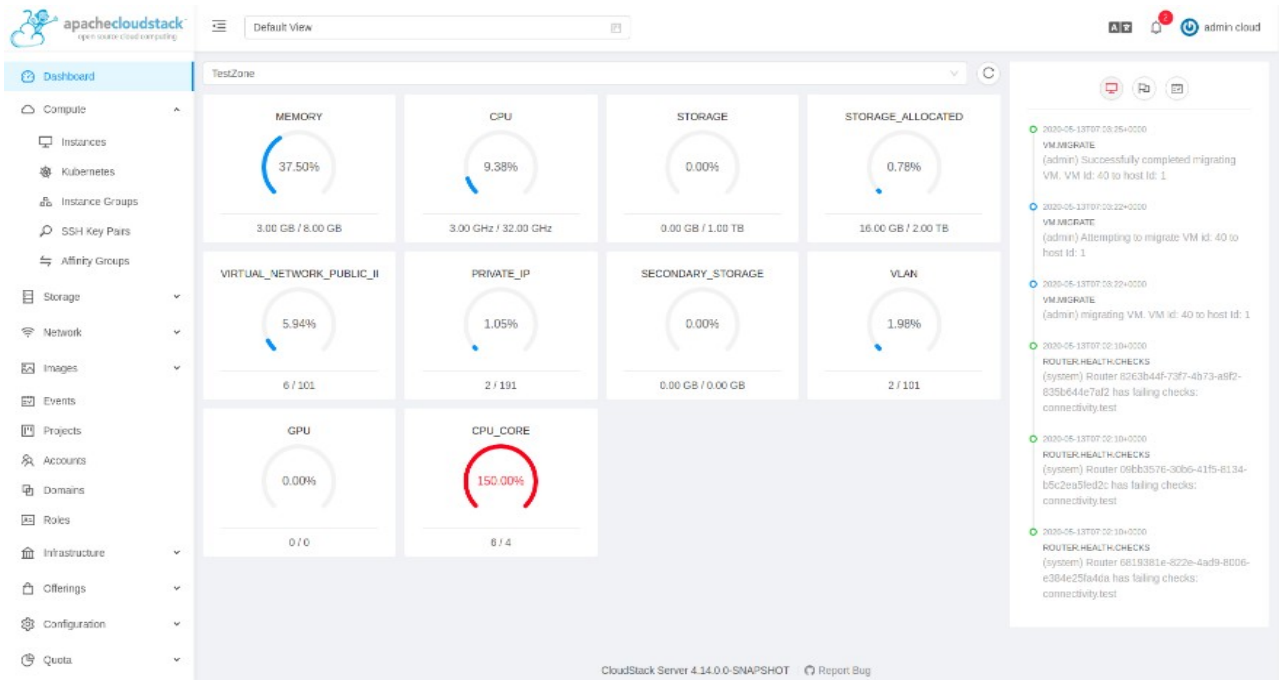


Рисунок 1.3 – Панель керування CloudStack

OpenNebula (розробники: OpenNebula Systems, OpenNebula community) [27]. OpenNebula є вільнопоширюваною хмарною платформою моделі IaaS. Розповсюджується під ліцензією Apache License версії 2. OpenNebula має підтримку спільноти, а також платну підтримку на основі підписки (OpenNebula EE).

OpenNebula має інтеграцію з Amazon AWS та Microsoft Azure. Програмний засіб має вбудований графічний інтерфейс користувача (Sunstone GUI, FireEdge GUI), а також підтримує інструменти сторонніх виробників, зокрема Terraform, Kubernetes, Ansible і Docker [28]. На рис. 1.4 зображено Sunstone GUI.

Для нових користувачів OpenNebula має інструмент під назвою miniONE, який дає змогу розгорнути хмару на базі OpenNebula на віртуальних машинах на базі гіпервізора KVM, встановлює та налаштовує всі необхідні компоненти. Си-

стемні вимоги для OpenNebula minion включають фізичний або віртуальний сервер з 4 Гб оперативної пам’яті, 30 ГіБ вільного місця на диску та відкритими портами для SSH (22), Sunstone (80) і FireEDGE (2616) [13].

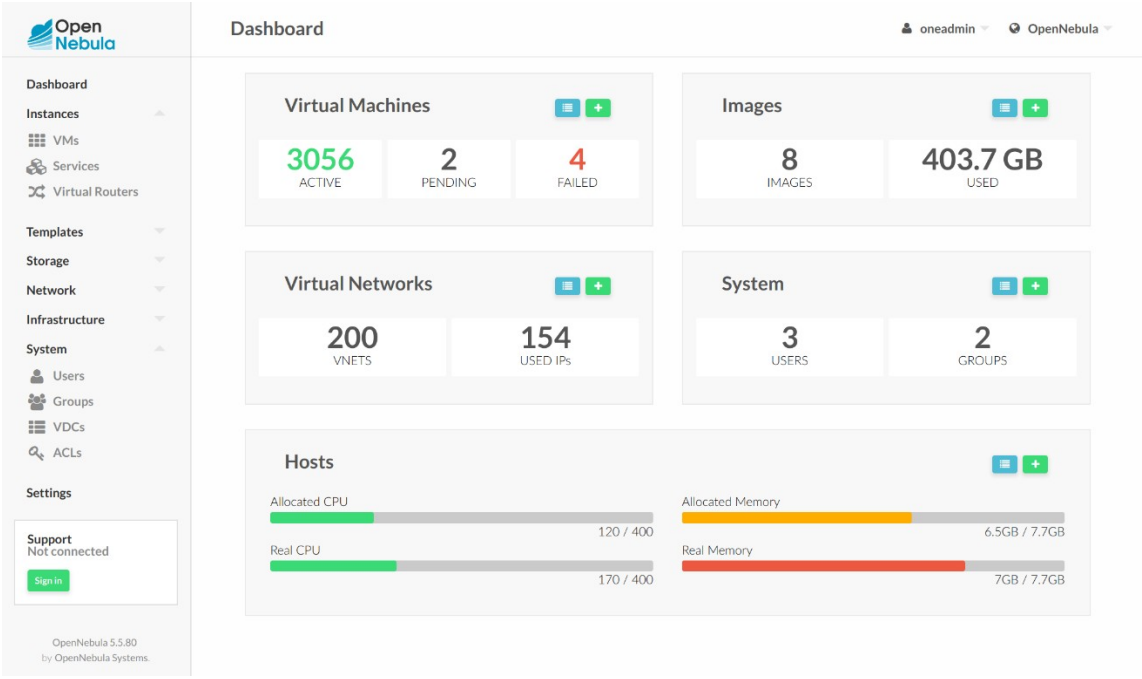


Рисунок 1.4 – Панель керування OpenNebula

Також варто відзначити, що всі проаналізовані платформи мають детальну онлайн-документацію з покроковими інструкціями.

Висновки з розділу 1

У розділі 1 було досліджено теоретичні аспекти побудови хмарної лабораторії. Досліджено поняття хмарних технологій, основних моделей, згідно з якими може розгортатися хмара. Розглянуто поняття хмарної лабораторії та напрями їх потенційного освітнього та дослідницького використання. Визначено основні підходи до організації хмарної інфраструктури – використання платформ, наданих провайдером хмарних сервісів, та використання спеціалізованого хмарного ПЗ. Розглянуто основні хмарні топології — одновузлова та багатовузлова. Здійснено огляд хмарних платформ для розгортання хмарної інфраструктури (Amazon EC2, Google Compute Engine, Azure Virtual Machines; Eucalyptus, OpenStack, CloudStack, OpenNebula).

РОЗДІЛ 2 ПРОЄКТУВАННЯ ХМАРНОЇ ЛАБОРАТОРІЇ

2.1 Вимоги до проєкту віртуальної хмарної лабораторії

Як було вказано у вступі, предметом цієї кваліфікаційної роботи є технології та засоби розроблення хмарної мережної інфраструктури та хмарних лабораторій, а метою – розроблення проєкту хмарної лабораторії на основі платформи MicroStack/OpenStack.

Дослідженню, описаному у даній кваліфікаційній роботі, передували попередні наші дослідження технологій віртуалізації. Зокрема, у роботі [14] було визначено потребу розроблення власного віртуального середовища з ОС Linux для використання у навчанні операційних систем у закладах вищої освіти та сформульовано вимоги до системи таких середовищ, яку надалі називатимемо хмарною лабораторією.

Вимоги до хмарної лабораторії для навчання ОС Linux є наступними.

– ***Реалізм.*** Поведінка ОС Linux у віртуальних середовищах повинна бути максимально наближеною до поведінки ОС Linux на фізичному комп'ютері.

– ***Релевантність***. Хмарна лабораторія має відповідати потребам курсу з операційних систем, тобто надавати змогу повноцінно виконувати практичні завдання курсу.

– ***Доступність.*** Хмарна лабораторія має бути доступна всім студентам, які проходять даний курс, після того, які ті успішно підтвердили свою ідентичність.

– ***Надійність.*** Потрібні механізми для вирішення проблем у роботі хмарної лабораторії протягом передбачуваного періоду часу.

– ***Масштабованість.*** Має бути змога розширювати та вдосконалювати хмарну лабораторію, зокрема переходити на потужніше обладнання, додавати новий функціонал (наприклад, практичні завдання, які перевіряються та оцінюються автоматично).

– ***Безпека.*** Мінімізація загроз та наявність механізмів для ефективного реагування на порушення безпеки хмарної лабораторії.

Зауважимо, що перелічені вимоги визначають напрями розроблення та розвитку лабораторії і не передбачають реалізації у всього функціоналу на етапі проєкту, запропонованого у даній кваліфікаційній роботі. Водночас, важливо, аби розроблюваний проєкт та обрані для його реалізації технології та засоби не суперечили означеним вимогам та давали змогу поетапно реалізувати їх повною мірою у майбутньому.

Отже, зважаючи на перелічені вище вимоги до хмарної лабораторії для навчання ОС Linux, **вимоги до проєкту хмарної лабораторії** є наступними.

– використання віртуальних середовищ з ОС Linux, об'єднаних за допомогою віртуальних комп'ютерних мереж;

– розгортання за моделлю приватної хмари (оскільки хмарна лабораторія використовуватиметься виключно студентами університету) [15];

– невеликі потреби в ресурсах (проєкт розгортатиметься на невеликих апаратних потужностях);

– можливість масштабування у майбутньому.

2.2 Добір хмарної платформи для проєкту хмарної лабораторії

Як було зазначено у п. 1.2, платформи для побудови хмарної інфраструктури, які надаються провайдерами хмарних сервісів, позбавляють проблем із купівлею та обслуговуванням фізичного обладнання і дають змогу оплачувати лише використаних обсяг ресурсів. Водночас, такі платформи не завжди забезпечують гнучкість, необхідну для дослідницького використання. Також умови користування сервісами може бути змінено, й рішення, раніше безкоштовні чи недорогі, згодом можуть суттєво подорожчати. До подібних змін особливо чутливі заклади освіти, які нерідко оперують порівняно невеликим бюджетом для таких потреб.

Тому, на наш погляд, для хмарної лабораторії, яка використовуватиметься закладом освіти, доцільно обрати хмарну інфраструктуру на основі спеціального хмарного ПЗ. У майбутньому можлива як її підтримка власними силами, так і перенесення на сервери провайдера, що підтримує відповідну хмарну платформу.

У статті [15] сформульовано вимоги до хмарного ПЗ, на основі якого може бути розроблено хмарну лабораторію для курсу з операційних систем, тобто хмарну лабораторію. *Вимоги до хмарного ПЗ для створення проєкту хмарної лабораторії* у даній кваліфікаційній роботі є аналогічними і наступними:

– підтримка розгортання на основі моделі приватної хмари;

– підтримка ОС Linux як гостьової ОС;

– розповсюдження під ліцензіями для вільно поширюваного ПЗ;

– підтримка віртуальних комп'ютерних мереж;

– наявність веб-інтерфейсу з можливість створювати, адмініструвати, видаляти віртуальні середовища і налаштовувати мережні з'єднання між ними;

– доступність детальної документації;

– інтеграція з протоколами автентифікації.

Перелічені вище вимоги врахуємо у порівняльному аналізі популярного вільно поширюваного хмарного ПЗ, огляд якого наведено у п. 1.3.

У табл. 2.1 наведено основні характеристики платформ Eucalyptus, OpenStack, CloudStack та OpenNebula.

Таблиця 2.1 – Порівняння вільно поширюваного хмарного ПЗ

Характеристики	**Eucalyptus**	**OpenStack**	**CloudStack**	**OpenNebula**
Модель хмарного сервісу	IaaS	IaaS	IaaS	IaaS
Ліцензія	GNU GPL v3	Apache License 2.0	Apache License 2.0	Apache License 2.0
Модель розгортання	приватна і гібридна хмара	приватна, публічна та гібридна хмара	приватна, публічна та гібридна хмара	приватна, публічна та гібридна хмара

Характеристики	Eucalyptus	OpenStack	CloudStack	OpenNebula
Технології віртуалізації	віртуальні машини (гіпервізори KVM, Xen, Vmware)	віртуальні машини (зокрема Vmware ESXi), віртуальні контейнери	віртуальні машини (гіпервізори KVM, Vmware vSphere, XenServer/XCP, VirtualBox), віртуальні контейнери (LXC)	віртуальні машини, (гіпервізори Vmware vCenter, KVM, VirtualBox), віртуальні контейнери (LXD), мікровіртуальні машини (AWS Firecrackers)
Основні ОС	Linux (Cent OS 7.9, RHEL 7.9)	Windows, Solaris, Linux	Linux, Windows	Linux (зокрема CentOS та Ubuntu)
Гостьові ОС	всі ОС, які підтримує використовувана технологія віртуалізації	всі ОС, які підтримує використовувана технологія віртуалізації	всі ОС, які підтримує використовувана технологія віртуалізації	всі ОС, які підтримує використовувана технологія віртуалізації
Назви віртуальних середовищ для гостьових ОС	інстанси / instances, віртуальні машини / virtual machines	інстанси / instances, віртуальні машини / virtual machines, віртуальні контейнери / virtual containers	інстанси / instances, віртуальні машини / virtual machines, віртуальні контейнери / virtual containers	інстанси / instances, віртуальні машини / virtual machines, віртуальні контейнери / virtual containers, мікровіртуальні машини / microVMs
Інтерфейс для адміністрування	веб-інтерфейс та CLI	веб-інтерфейс (Horizon)	веб-інтерфейс (CloudStack UI)	Sunstone GUI, FireEdge GUI CLI
Інтерфейс для гостьового доступу	веб-інтерфейс, Euca2ools, AWS CLI	Horizon, OpenStackClient	веб-інтерфейс (CloudStack UI)	Sunstone GUI, CLI
Віртуальні мережі	ASW VPC	neutron	віртуальні мережі AWS, Nicira NVP	Linux bridge, 802.1Q, VXLAN, Open-vSwitch

Характеристики	Eucalyptus	OpenStack	CloudStack	OpenNebula
Інтеграція з LDAP	так	так	так	так

Як видно з порівняльної таблиці, усі розглянуті хмарні платформи відповідають наведеним вище вимогам до хмарного ПЗ для створення проєкту хмарної лабораторії. Біль ґрунтовне практичне дослідження технічних характеристик цих хмарних платформ здійснено у дослідженні [5], автори якого здійснили порівняння хмарних платформ Eucalyptus, OpenStack, CloudStack та OpenNebula і дійшли висновків, що всі ці системи може бути успішно використано для розгортання хмари, водночас OpenStack та Eucalyptus краще оптимізовано для хмарної інфраструктури, а CloudStack та OpenNebula – для віртуалізації центрів обробки даних.

Беручи до уваги все вищезгадане, для реалізації проєкту хмарної лабораторії було обрано платформу OpenStack, якій притаманна висока гнучкість та відповідність відкритим стандартам. Крім того, OpenStack пропонує рішення, які підійдуть саме для невеликих проєктів завдяки невисоким системним вимогам – OpenStack Training Labs [9], MicroStack на базі Sunbeam [18], невеликий кластер на основі Microstack з інстансами CirrOS [22]. Водночас, OpenStack дає змогу масштабувати проєкт у майбутньому.

Після попереднього тестування трьох перелічених вище OpenStack-рішень для подальшої реалізації проєкту хмарної лабораторії було обрано останнє – невеликий кластер на основі Microstack з інстансами CirrOS [22]. OpenStack Training Labs, розроблені хронологічно раніше, мають проблеми сумісності з Ubuntu 22.04 й навіть після адаптації скриптів на Python 2 відповідно до вимог Python 3, не запрацювали на належному рівні. Рішення MicroStack на базі Sunbeam потребує 16 Гб вільної оперативної пам’яті, а тому не працює на ноутбуці з 16 Гб оперативної пам’яті загалом, навіть за умови завантаження лише інтерфейсу командного рядка (Ubuntu 22.04, текстовий термінал номер 3).

Детальніші відомості про основні компоненти OpenStack наведено у додатку Б.

2.3 Топологія та схема адресації моделі комп'ютерної мережі

Враховуючи порівняно невеликі апаратні ресурси (розгортання здійснюватиметься на ноутбуці з 16 ГіБ RAM), а також дослідницьку мету, для проєкту хмарної лабораторії обрано одновузлову топологію.

Топологію віртуальної мережі, використаної у проєкті хмарної інфраструктури, показано на рис. 2.1. У моделі використано дві мережі: внутрішню (*lab-network*) та зовнішню (*public*). Мережі поєднує віртуальний маршрутизатор *public-router*. У внутрішній мережі розміщуються віртуальні інстанси (*test-instance*, *ubuntu-test*). Зовнішня мережа використовується для забезпечення зовнішнього доступу до інстансів.

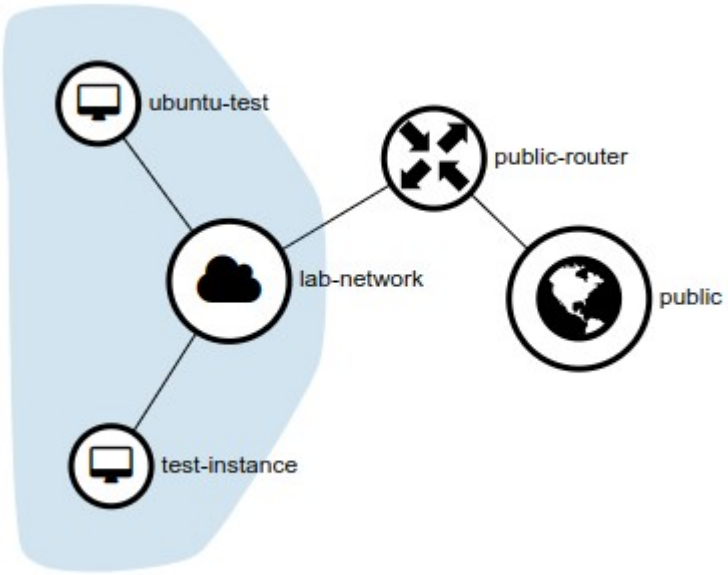


Рисунок 2.1 – Топологія мережі проєкту хмарної інфраструктури

Модель мережі використовує схему адресації, наведену у табл. 2.2.

Таблиця 2.2 – Схема адресації мережі проєкту хмарної інфраструктури

Мережа / Пристрій	**Інтерфейс / Мережний адаптер / Шлюз**	**IP-адреса**	**Маска**	**Префікс**
Мережа ***public***	–	192.168.0.0	255.255.255.0	/24
Маршрутизатор *public-router*	(463bb0b7-78b9)	192.168.0.32	255.255.255.0	/24
Мережа ***lab-network***	–	10.0.10.0	255.255.255.0	/24
Маршрутизатор *public-router*	(dba7c3f8-b473)	10.0.10.1	255.255.255.0	/24
Інстанс *test-instance* (CirrOS)	Мережний адаптер	10.0.10.47	255.255.255.0	/24
	Шлюз за замовчуванням	10.0.10.1	–	–
	Плаваюча адреса	192.168.0.37	–	–
Інстанс *ubuntu-test* (Ubuntu 22.04)	Мережний адаптер	10.0.10.22	255.255.255.0	/24
	Шлюз за замовчуванням	10.0.10.1	–	–
	Плаваюча адреса	192.168.0.16	–	–

Загалом, схема адресації для мережі 192.168.0.0/24 (*public*) має 254 доступні для використання адреси у діапазоні від 192.168.0.1 до 192.168.0.254, а схема адресації для мережі 10.0.10.0/24 (*lab-network*) – 254 доступні для використання адреси у діапазоні від 10.0.10.1 до 10.0.10.254. У межах мережі *public* виділятимуться плаваючі (floating) адреси для інстансів, завдяки чому до них уможливиться доступ з інших вузлів фізичної домашньої мережі (наприклад, з іншого ноутбука). Важливо також, аби адреси, використовувані як плаваючі для інстансів, не конфліктували з адресами, які маршрутизатор у фізичній мережі призначає іншим фізичним простоям – це потрібно врахувати у налаштуваннях віртуальних мереж у хмарі. У мережі *lab-network* використано невелику частину адрес, але за потреби у модель може бути додано більшу кількість вузлів. Наразі у табл. 2.1 є два хмарні інстанси — *test-instance* та *ubuntu-test*. Їхні внутрішні адреси (10.0.10.47,

10.0.10.22) та їхні плаваючі адреси (192.168.0.37, 192.168.0.16) відповідають адресам, автоматично виділених для них платформою MicroStack. Відповідні налаштування описані у п. 3.1.

Для досягнення більшої реалістичності проєкту хмарної лабораторії необхідно реалізувати доступ до хмарних інстансів не лише з ноутбука, на якому розгорнуто хмару, а й з інших пристроїв фізичної домашньої мережі. Для зв'язку між віртуальною мережею *public* та фізичною домашньою мережею (також 192.168.0.0/24) застосовано спеціальні налаштування одного з вбудованих віртуальних мережних інтерфейсів MicroStack/OpenStack — *br-ex*, які детальніше описано у п. 3.1.

Висновки з розділу 2

У розділі 2 здійснено проєктування хмарної лабораторії. Сформульовано вимоги до хмарної лабораторії, на основі яких виконано добір хмарної платформи для реалізації прототипу. Згідно з результатами порівняльного аналізу, обрано хмарну платформу OpenStack, а саме рішення MicroStack. Розроблено топологію та схему адресації моделі мережі для проєкту хмарної лабораторії.

РОЗДІЛ 3 НАЛАШТУВАННЯ ТА ТЕСТУВАННЯ ПРОЄКТУ ХМАРНОЇ ЛАБОРАТОРІЇ

3.1 Розгортання проєкту хмарної лабораторії

Розгортання проєкту хмарної лабораторії здійснювалося на ноутбуці HP 250 G5 з процесором Intel Core i5-6200U @ 2.30GHz × 4, 16 ГіБ оперативної пам’яті та основною ОС Ubuntu 22.04.3 LTS з підтримкою технології Snap.

Передусім, на комп'ютері, на якому розгортатиметься хмара, варто мати два фізичних мережних інтерфейси. Перший мережний інтерфейс застосовуватиметься для виходу в Інтернет, оскільки мати такий доступ корисно під час розгортання для вирішення можливих проблем. Роль першого мережного інтерфейсу у даній моделі виконуватиме бездротовий мережний інтерфейс *wlo1* на ноутбуці HP. Другий мережний інтерфейс буде використовуватися для забезпечення зв'язку між віртуальною мережею та фізичного мережею. У даній моделі роль другого мережного інтерфейсу виконуватиме дротовий мережний інтерфейс *eno1* на ноутбуці HP. На початку обидва інтерфейси матимуть доступ до Інтернету, згодом інтерфейс *eno1* буде переприв’язаний до віртуальної мережі. Інтерфейс *eno1* буде використовуватися як мережний інтерфейс для Microstack, тому встановлення та ініціалізація Microstack здійснюватиметься одразу на ньому.

Під час розгортання використано мінімальну тестову конфігурацію MicroStack на основі рекомендацій, наведених у статті [22], та налаштування віртуальних мереж, запропонованих у статті [24].

Перед початком роботи тимчасово відключаємо бездротовий мережний інтерфейс (*wlo1*) і перемикаємося на дротове мережне з'єднання (*eno1*), аби подальші налаштування Microstack виконувалися для *eno1*.

Встановлено snap-пакунок *microstack* (рис. 3.1).



Рисунок 3.1 – Встановлення MicroStack

Далі проініціалізовано хмарну інфраструктуру на базі MicroStack. Налаштування обрано автоматичні, окрім налаштувань компоненту Cinder, які було змінено задля зручнішої роботи з віртуальними накопичувачами через веб-інтерфейс Horizon (рис. 3.2). Ініціалізація інфраструктури потребувала часу.

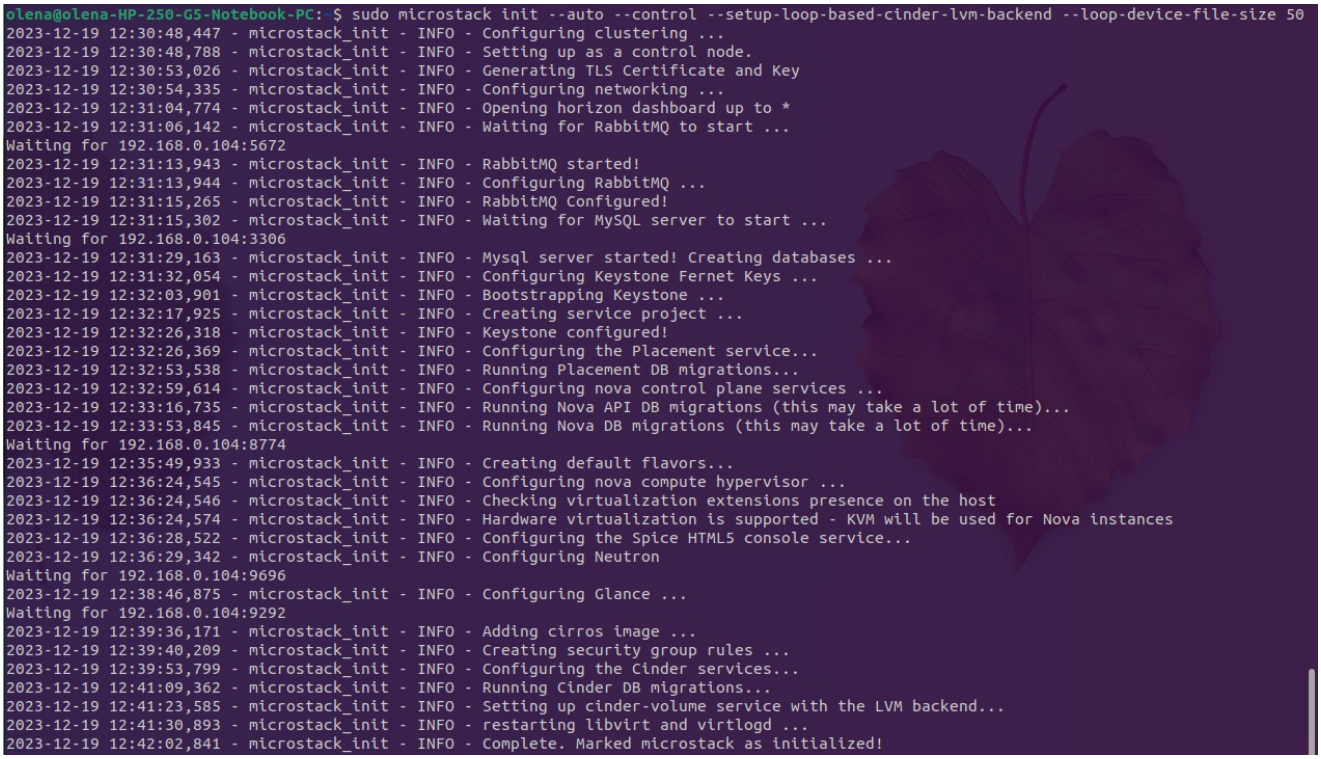


Рисунок 3.2 – Ініціалізація хмарної інфраструктури

MicroStack успішно встановлено та проініціалізовано. Поточна версія OpenStack – 5.2.0 (рис. 3.3).

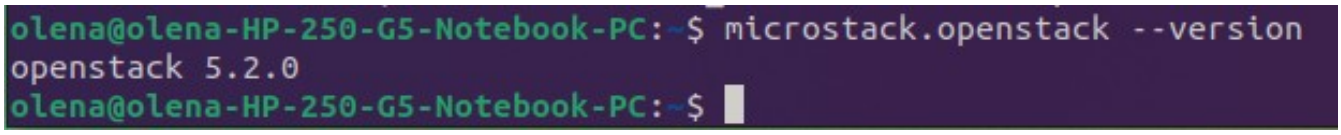


Рисунок 3.3 – Використовувана версія OpenStack

З'ясовуємо реквізити для входу в Horizon (рис. 3.4).

```
olena@olena-HP-250-G5-Notebook-PC:~$ sudo snap get microstack config.credentials.keystone-password
```

Рисунок 3.4 – Виведення на екран паролю адміністратора для Horizon

За допомогою веб-браузера переходимо за адресою https://10.20.20.1 та входимо до веб-інтерфейсу Horizon (рис. 3.5) під обліковим записом *admin* та паролем, одержаним вище.

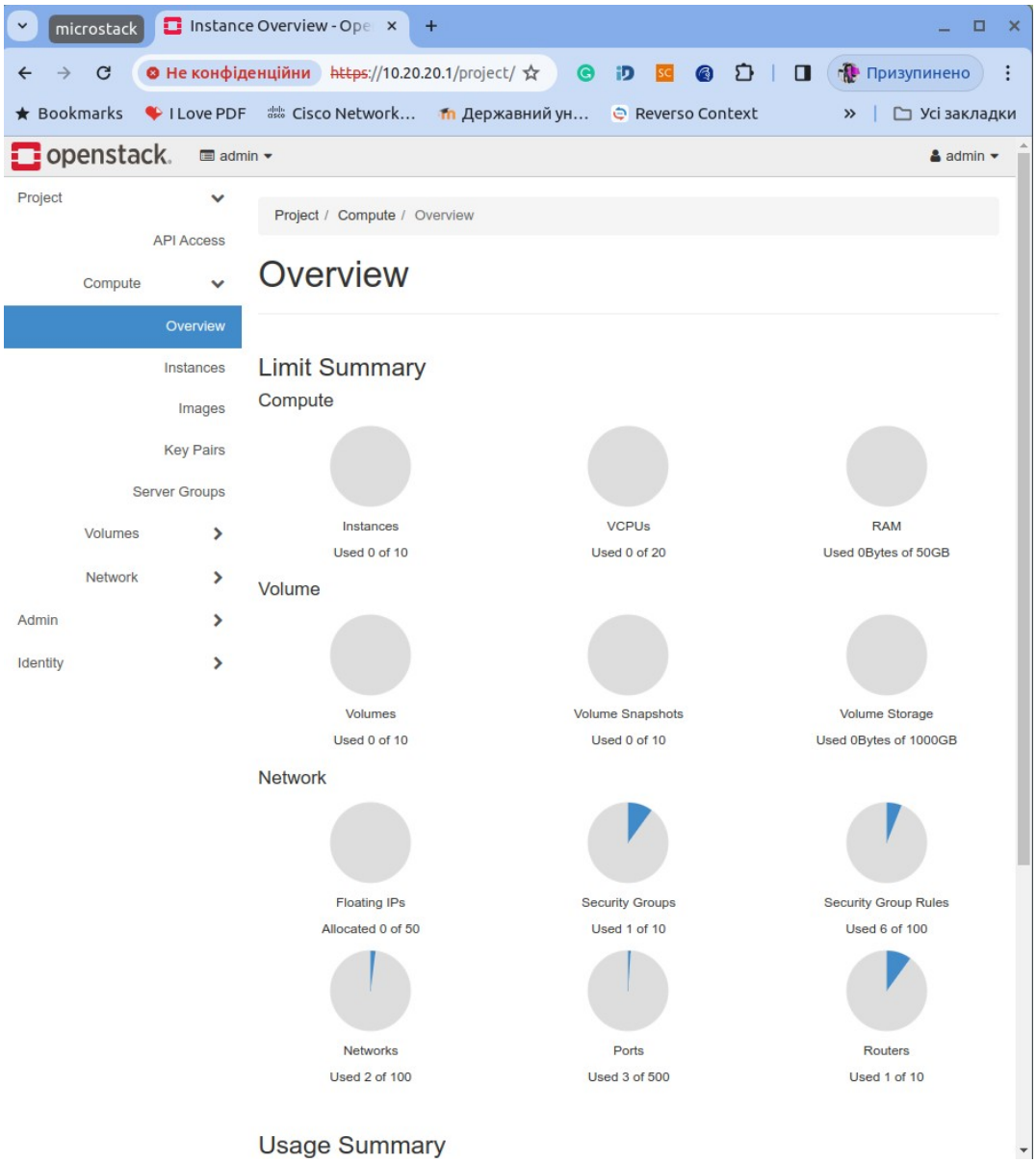


Рисунок 3.5 – Веб-інтерфейс Horizon

Якщо потрібний лише тестовий запуск MicroStack із подальшим доступом до інстансів лише з даного хоста, то жодних спеціальних мережних налаштувань не потрібно. Однак для хмарної лабораторії зазвичай важливо мати доступ до

хмарних інстансів з інших фізичних комп'ютерів мережі через протокол SSH. Тому у даній моделі застосовуватимуться налаштування фізичної та віртуальних мереж, відмінні від створюваних за замовчуванням (відповідну схему адресації описано у п. 2.3).

Налаштовуємо псевдонім для *microstack.ovs-vsctl* (рис. 3.6). Це не обов'язковий крок, але він зробить команди з віртуальними мережними інтерфейсами коротшими.

```
olena@olena-HP-250-G5-Notebook-PC:~$ sudo snap alias microstack.ovs-vsctl ovs-vsctl
Додано:
  - microstack.ovs-vsctl як ovs-vsctl
```

Рисунок 3.6 – Налаштування псевдоніму *ovs-vsctl*

Видаляємо тестовий маршрутизатор та тестові мережі через інтерфейс Horizon, категорія *Admin* > *Networks* (рис. 3.7-3.8).

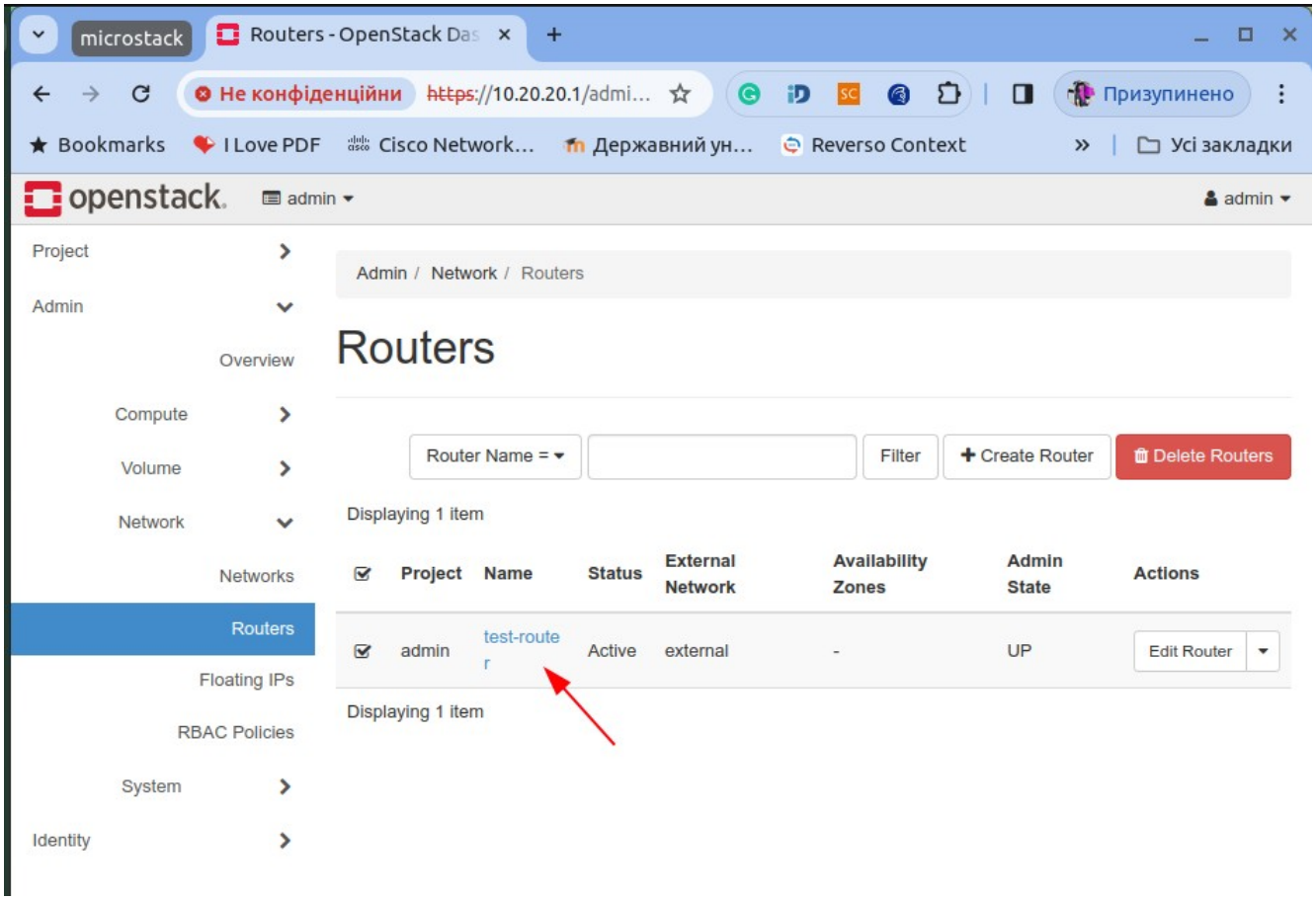


Рисунок 3.7 – Видалення тестового віртуального маршрутизатора

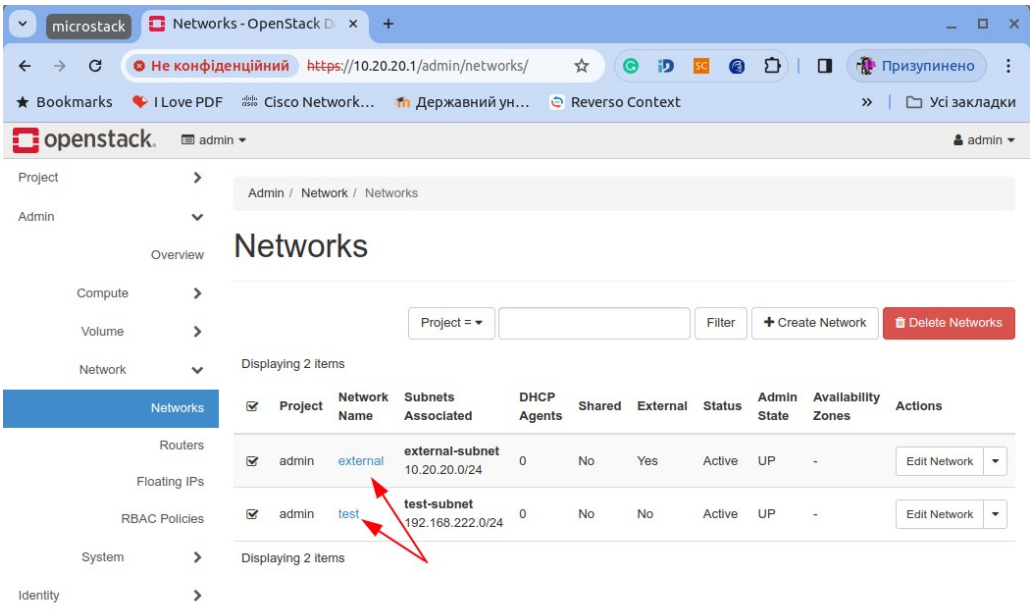


Рисунок 3.8 – Видалення тестових віртуальних мереж

У Horizon створюємо нову віртуальну мережу *public*, яка буде відповідати фізичній мережі, до котрої під'єднано ноутбук через інтерфейс Ethernet. Вона має бути зовнішня (*external*), мати тип фізичної реалізації *flat*. Прапорець DHCP для віртуальної мережі знімаємо для уникнення конфліктів з DHCP на фізичному обладнанні домашньої мережі (рис. 3.9-3.11).

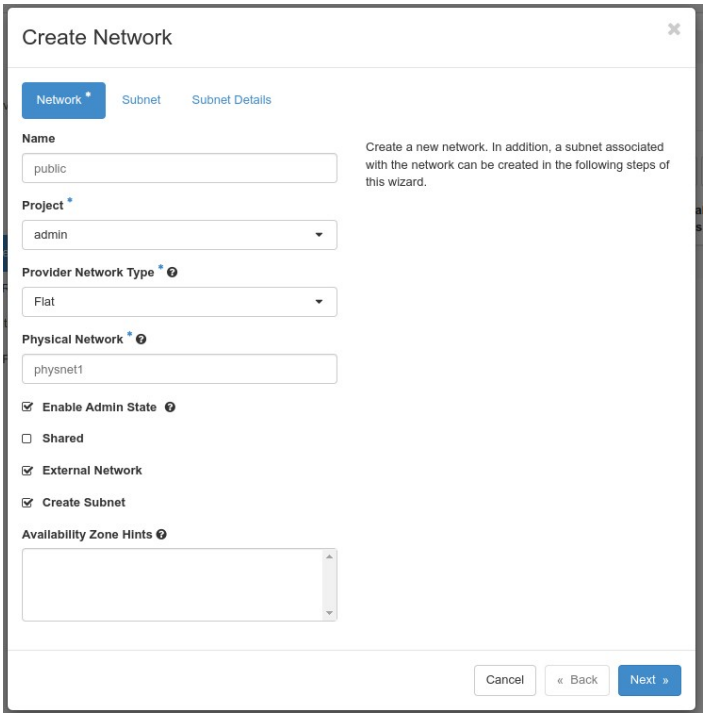


Рисунок 3.9 – Створення зовнішньої віртуальної мережі *public*

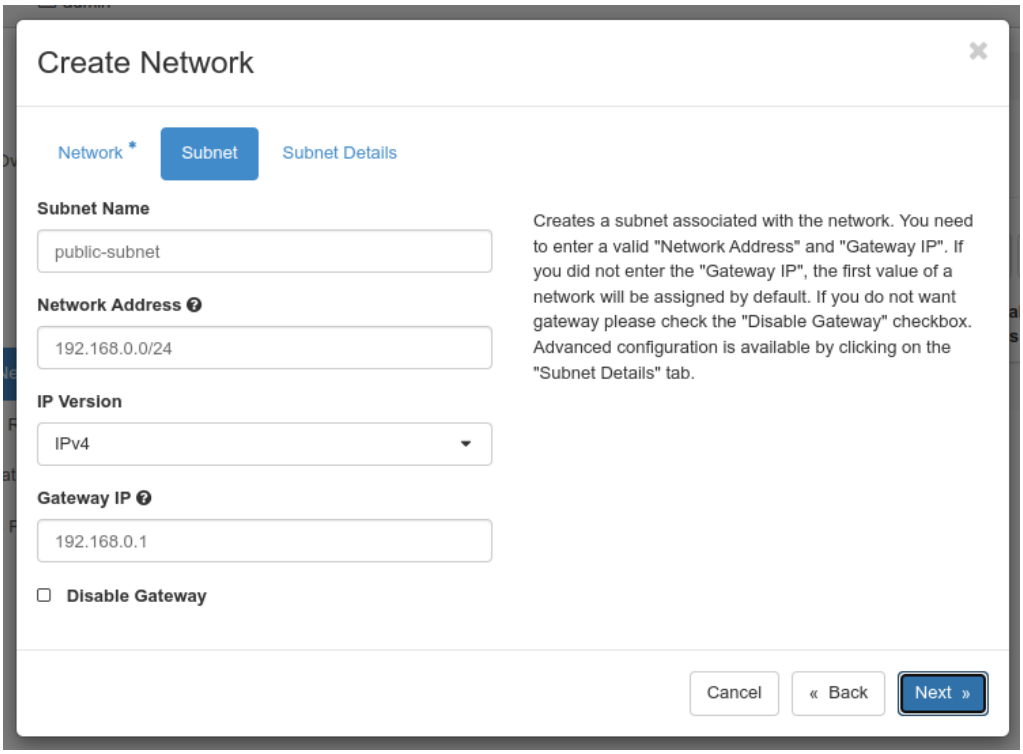


Рисунок 3.10 – Створення зовнішньої віртуальної підмережі *public-subnet*

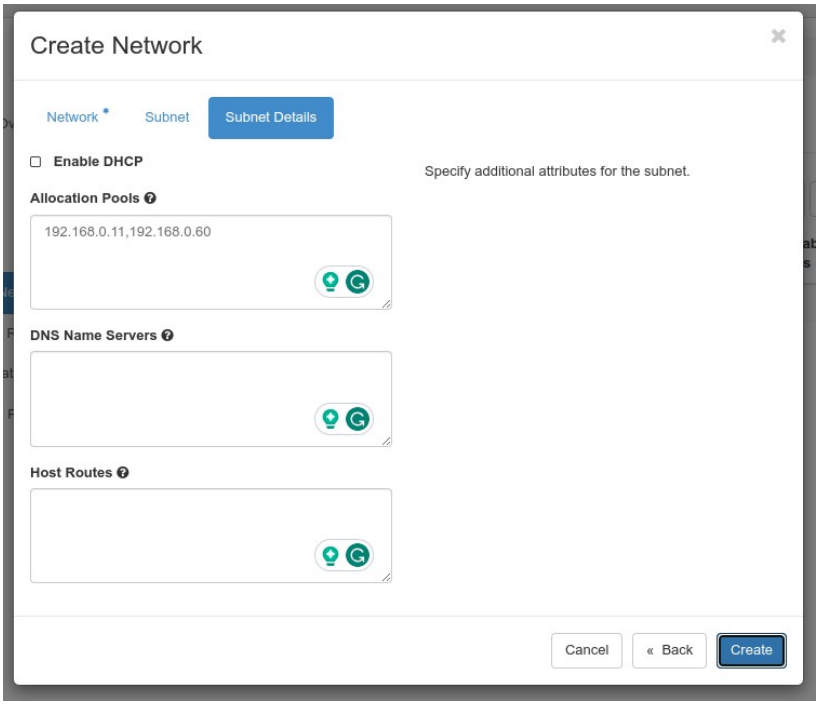


Рисунок 3.11 – Налаштування діапазону віртуальних зовнішніх адрес

Пул для адрес, які використовуватиме мережа *public*, обрано від 192.168.0.11 до 192.168.0.60. Ці адреси не повинні бути серед тих, які фізичний

маршрутизатор присвоїв іншим пристроям у мережі. Під час введення у відповідне поле між ними не ставимо пробіл, інакше налаштування не вдасться зберегти.

Додаємо інтерфейс *eno1* як порт для інтерфейсу зовнішнього мосту OpenStack, який має ім'я *br-ex* і був автоматично налаштований під час ініціалізації MicroStack (рис. 3.12).

```
olena@olena-HP-250-G5-Notebook-PC:~$ sudo -s
root@olena-HP-250-G5-Notebook-PC:/home/olena# ovs-vsctl add-port br-ex eno1
root@olena-HP-250-G5-Notebook-PC:/home/olena# ip addr flush dev eno1
root@olena-HP-250-G5-Notebook-PC:/home/olena# ip addr add 192.168.0.10/24 dev br-ex
root@olena-HP-250-G5-Notebook-PC:/home/olena# ip link set br-ex up
root@olena-HP-250-G5-Notebook-PC:/home/olena#
exit
```

Рисунок 3.12 – Зміна налаштувань інтерфейсу *br-ex*

Щоб зміни мережної конфігурації були постійними, скористаємося рекомендаціями, запропонованими у статті [24]. Спершу створимо bash-скрипт з відповідними мережними налаштуваннями (рис. 3.13).

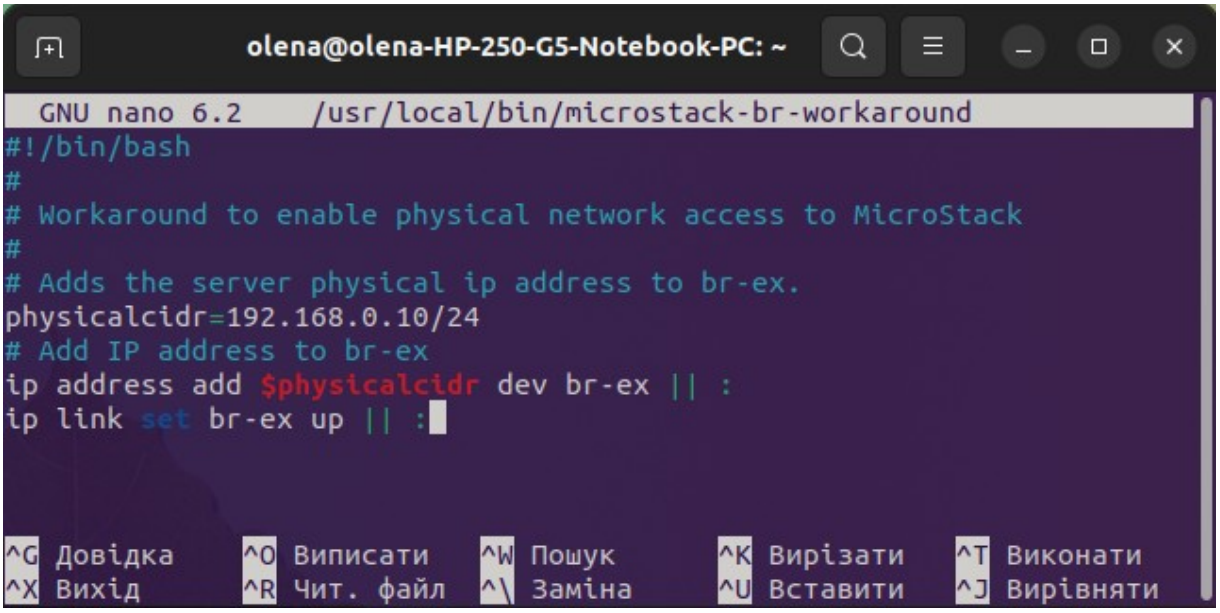


Рисунок 3.13 – Файл */usr/local/bin/microstack-br-workaround*

Потім додамо фоновий процес, який буде застосовувати налаштування з файлу */usr/local/bin/microstack-br-workaround,* коли ноутбук, на якому розгорнуто хмару, перезавантажуватиметься (рис. 3.14).

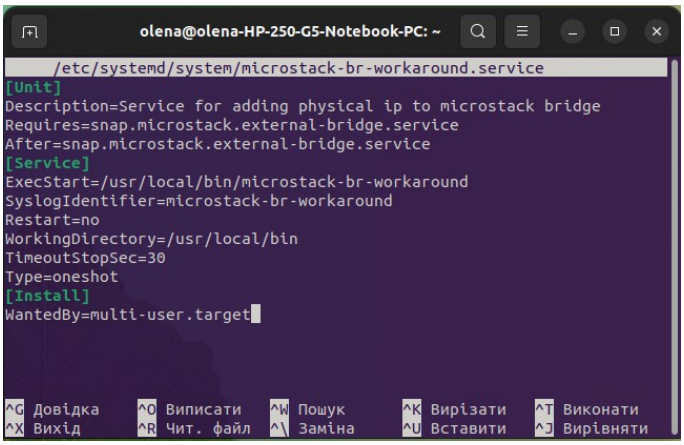


Рисунок 3.14 – Файл */etc/systemd/system/microstack-br-workaround.service*

Вмикаємо фоновий процес *microstack-br-workaround.service* (рис. 3.15)

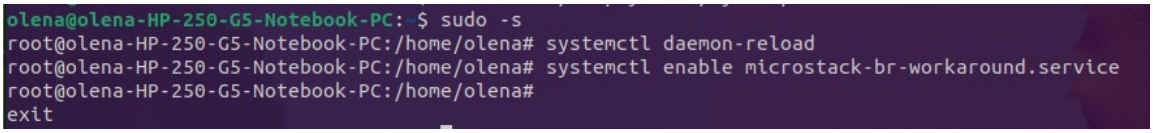


Рисунок 3.15 – Активація фонового процесу *microstack-br-workaround.service*

У налаштуваннях DHCP на фізичному маршрутизаторі фіксуємо за дротовим інтерфейсом ноутбуку його поточну адресу (192.168.0.104) як зарезервовану, після чого перезавантажуємо ноутбук. Це перезавантаження є обов’язковим і має передувати створенню нової хмарної мережної інфраструктури.

Далі створюємо внутрішню мережу *lab-network* (рис. 3.16-3.18).

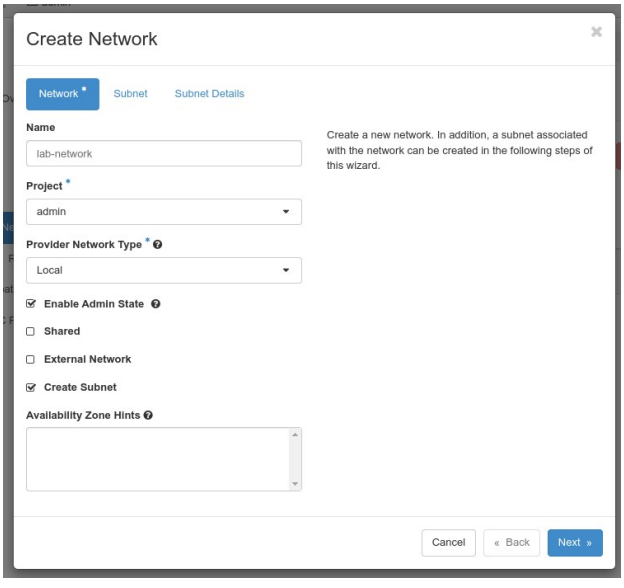


Рисунок 3.16 – Створення внутрішньої віртуальної мережі *lab-network*

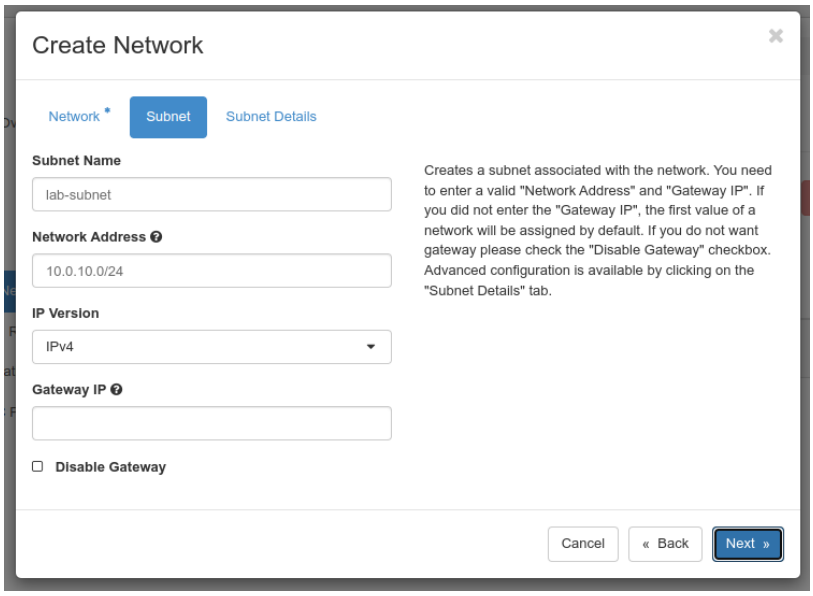


Рисунок 3.17 – Створення внутрішньої віртуальної підмережі *lab-subnet*

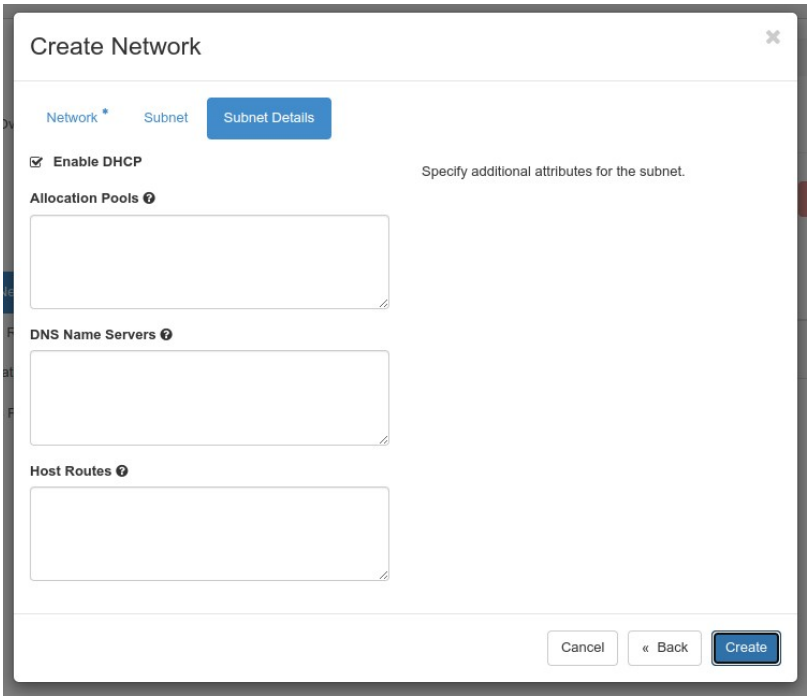


Рисунок 3.18 – Увімкнення DHCP для підмережі *lab-subnet*

На даному етапі маємо дві віртуальні мережі – *lab-network* та *public* (рис. 3.19).

Для приєднання мережі *lab-network* до мережі *public* створюємо віртуальний маршрутизатор *public-router* (рис. 3.20).

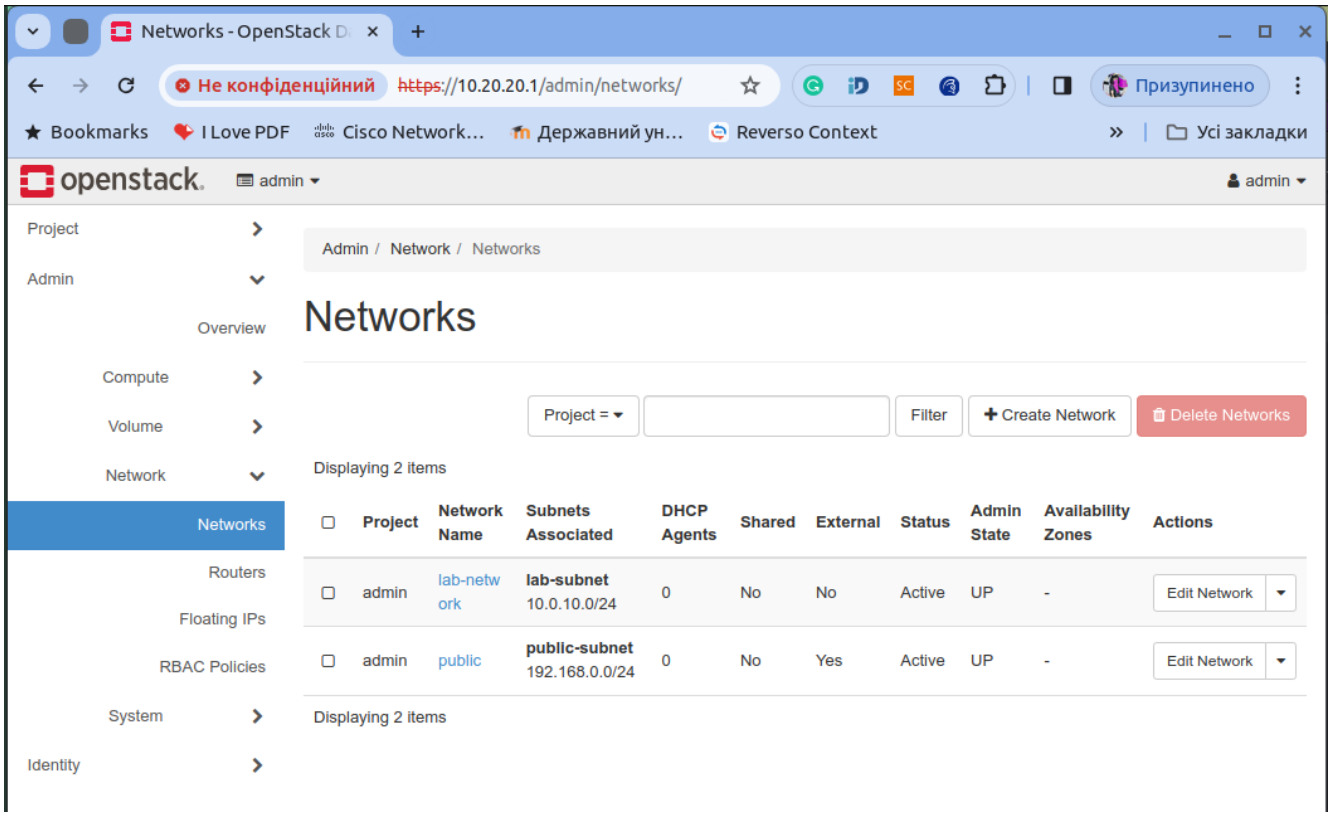


Рисунок 3.19 – Мережі *lab-network ma public*

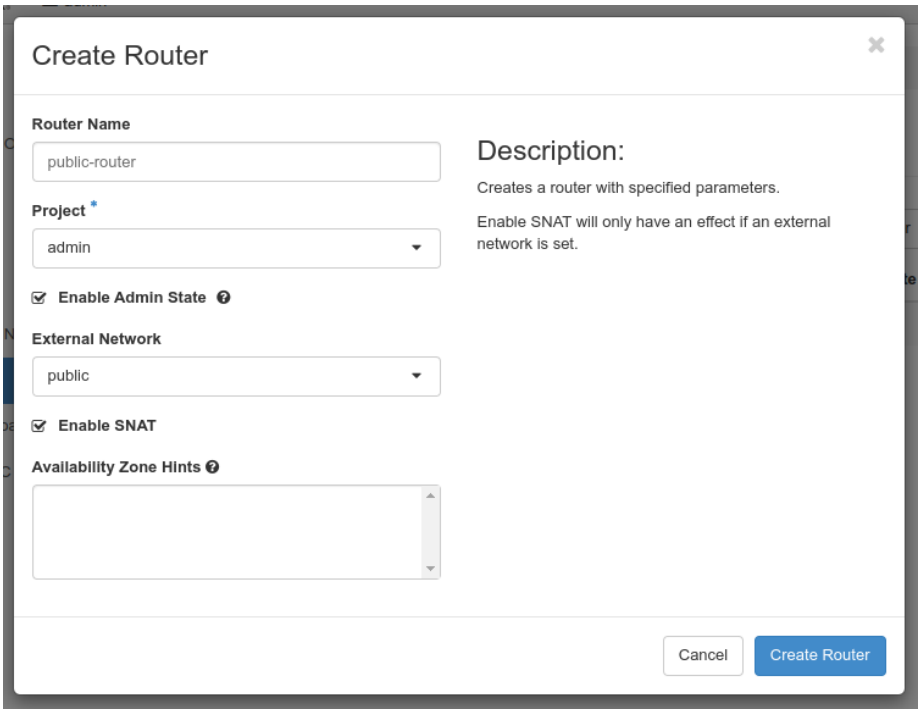


Рисунок 3.20 – Створення віртуального маршрутизатора *public-router*

Новостворений маршрутизатор *public-router* наразі під'єднаний лише до мережі *public* (рис. 3.21).

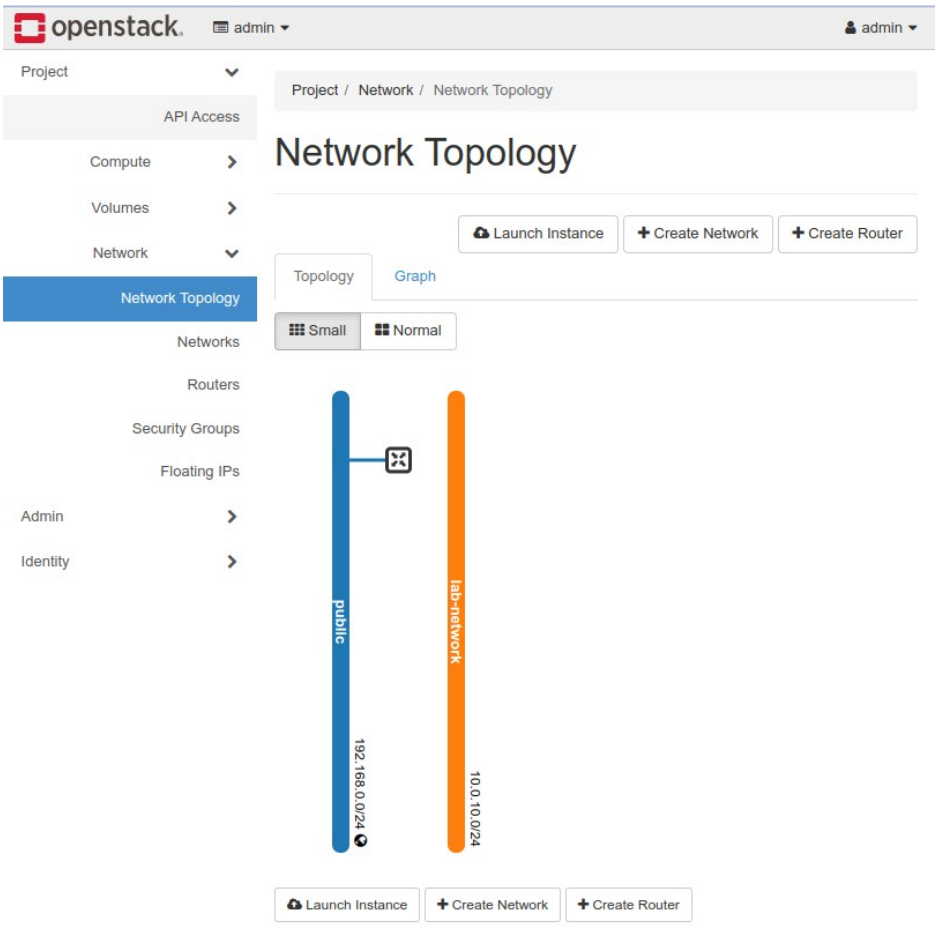


Рисунок 3.21 – Топологія мережі одразу після додавання маршрутизатора

Для приєднання маршрутизатора також і до мережі *lab-network* переходимо в *Project* > *Network* > *Network Topology* і на маршрутизаторі натискаємо *Add Interface* (рис. 3.22).

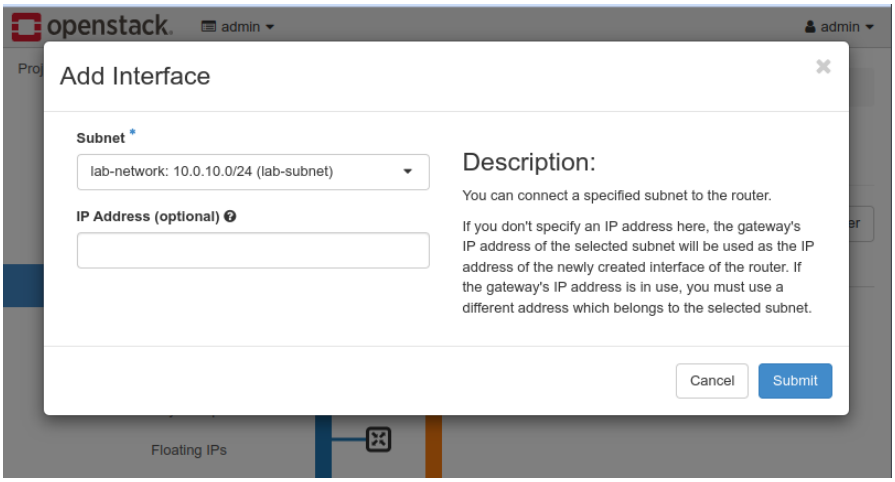


Рисунок 3.22 – Під’єднання маршрутизатора до мережі *lab-network*

Тепер маршрутизатор під'єднаний до обох віртуальних мереж, обидва відповідні інтерфейси активні (рис. 3.23, 3.24).

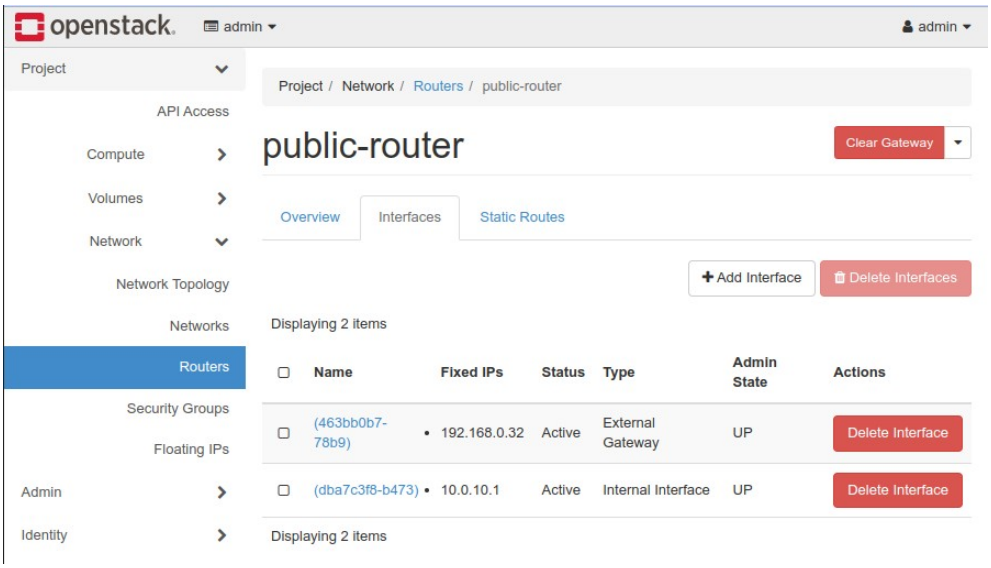


Рисунок 3.23 – Властивості маршрутизатора *public-router* після під'єднання до обох мереж



Рисунок 3.24 – Маршрутизатор *public-router* після під'єднання до обох мереж на топології хмарної мережної інфраструктури

Після створення нової мережної інфраструктури, яка відповідає потребам проєкту хмарної лабораторії, переходимо до створення інстансів.

Для майбутнього інстансу потрібна плаваюча IP-адреса (floating IP). Вона, разом зі здійсненими вище мережними налаштуваннями, забезпечить доступ до цього інстансу з інших фізичних пристроїв домашньої мережі. Переходимо у категорію *Project* > *Network* > *Floating IPs* та натискаємо *Allocate IP to Project* (рис. 3.25).

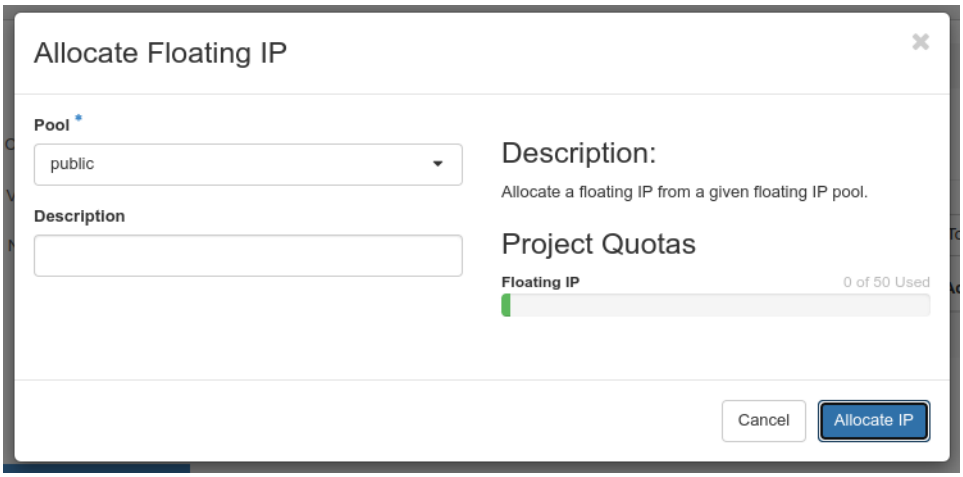


Рисунок 3.25 – Запит на виділення плаваючої IP-адреси у мережі *public*

Плаваючу IP-адресу успішно виділено (рис. 3.26).

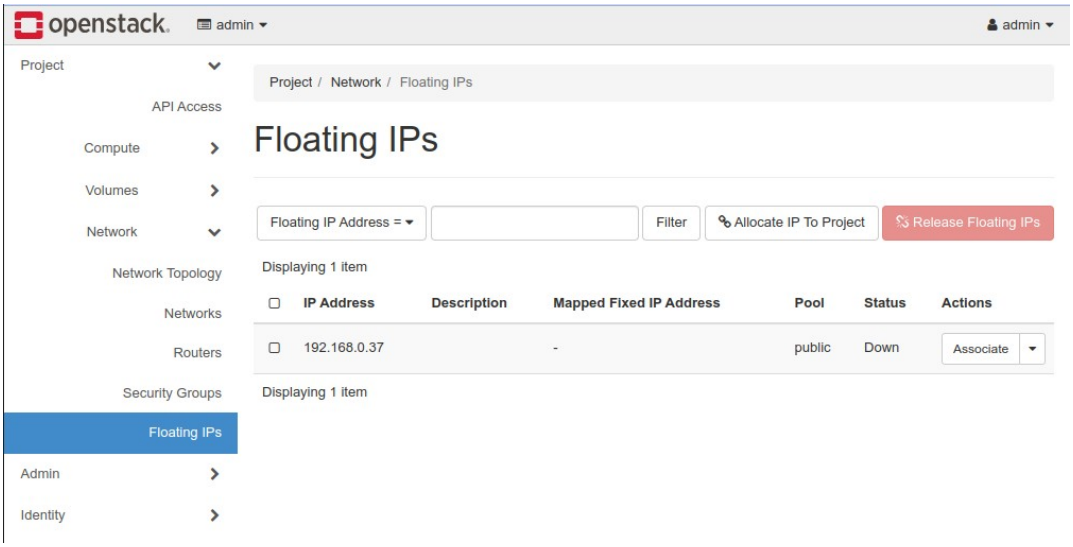


Рисунок 3.26 – Нова плаваюча IP-адреса (192.168.0.37)

Далі створюємо безпекову групу. Вона міститиме безпекові налаштування мережного доступу до інстансів за різними мережними протоколами. Обираємо *Project* > *Network* > *Security Groups* > *Create Security Group* (рис. 3.27).

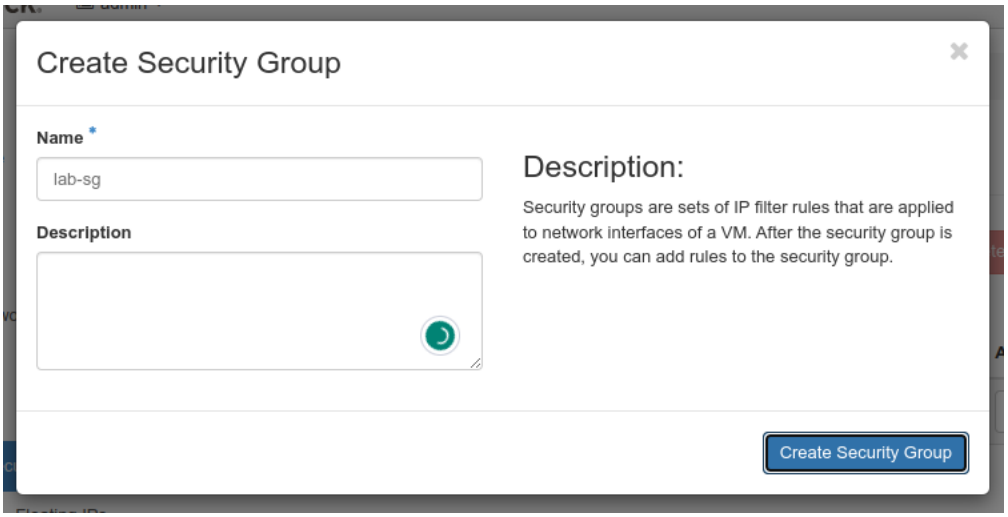


Рисунок 3.27 – Створення нової безпекової групи *lab-sg*

Для новоствореної безпекової групи додаємо правила, які уможливлять використання команди *ping* для перевірки з'єднання з інстансом (тип правила – *All ICMP*) та для доступу до інстансу через протокол SSH (тип правила – *SSH*). У Horizon це можна зробити, перейшовши у категорію *Project* > *Network* > *Security Groups* та обравши *Manage Rules* (рис. 3.28-3.29).

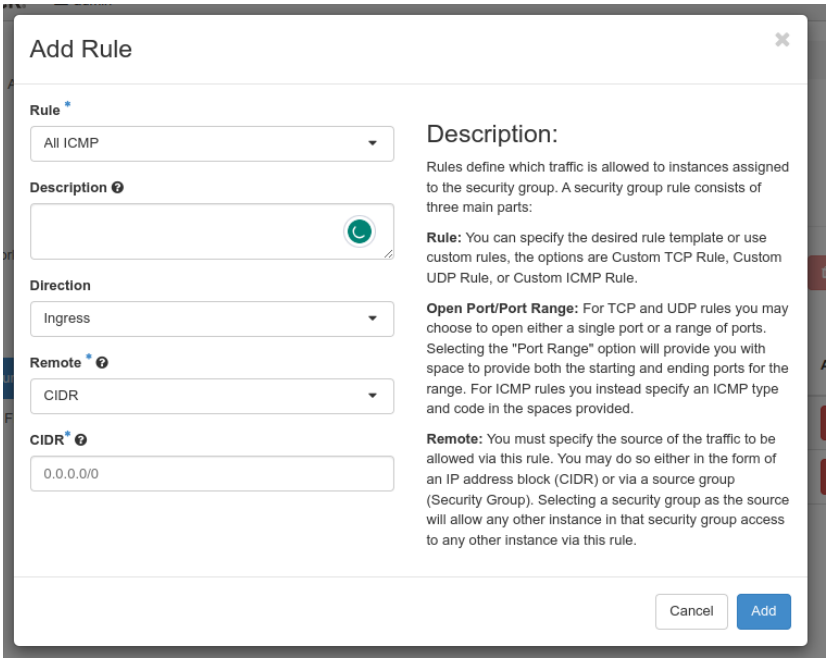


Рисунок 3.28 – Додавання правила для протоколу ICMP

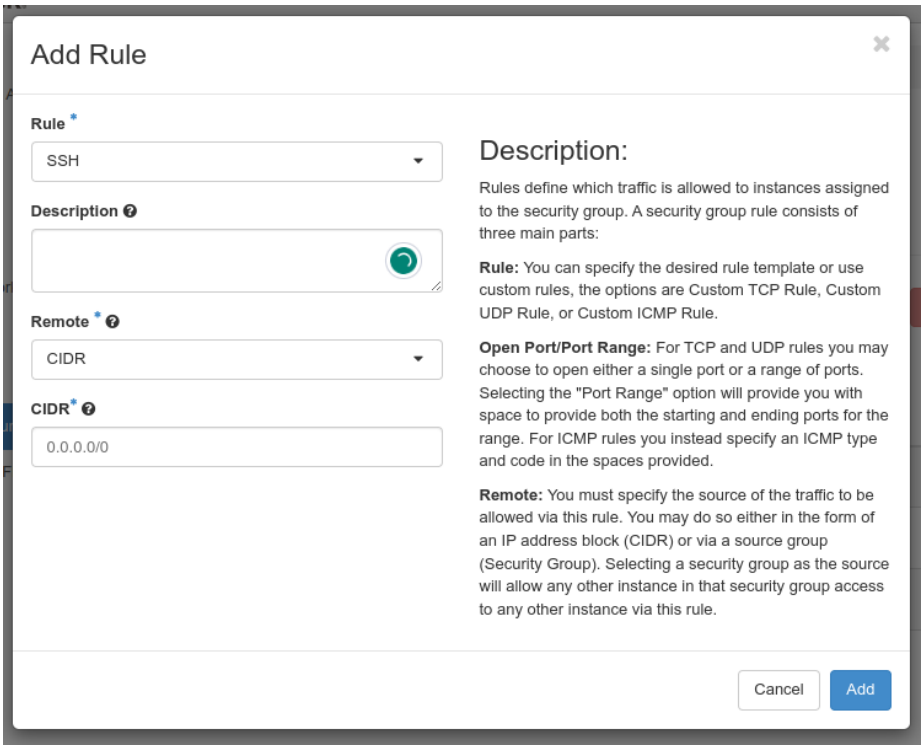


Рисунок 3.29 – Додавання правила для протоколу SSH

І нарешті, створюємо та запускаємо віртуальний інстанс. Це тестовий інстанс на базі CirrOS, полегшеного дистрибутиву Linux для тестових образів.

Спершу задамо ім’я (*test-instance*), як це показано на рис. 3.30.

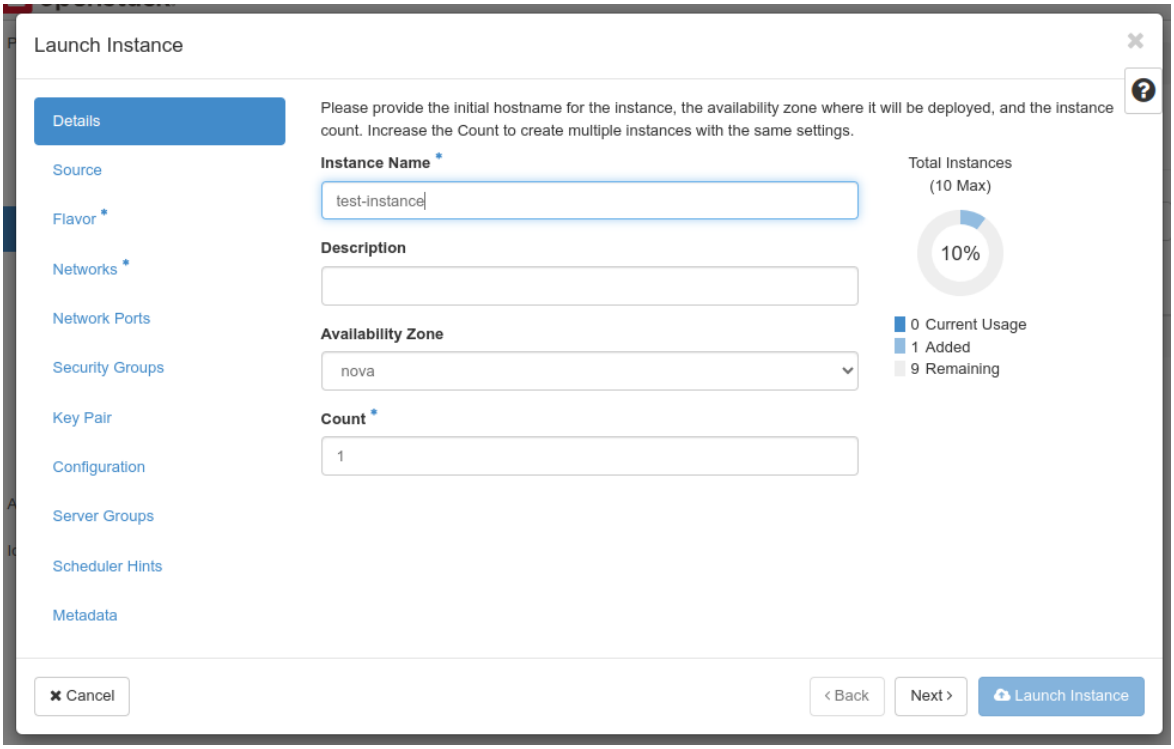


Рисунок 3.30 – Задання імені для інстансу з CirrOS

Далі оберемо образ віртуальної машини, з якого необхідно розгорнути інстанс. На даному етапі доступний лише один образ – це саме CirrOS (рис. 3.31). Інші образи потрібно додавати самостійно залежно від потреб. Важливо враховувати, що файли образів також займають дисковий простір.

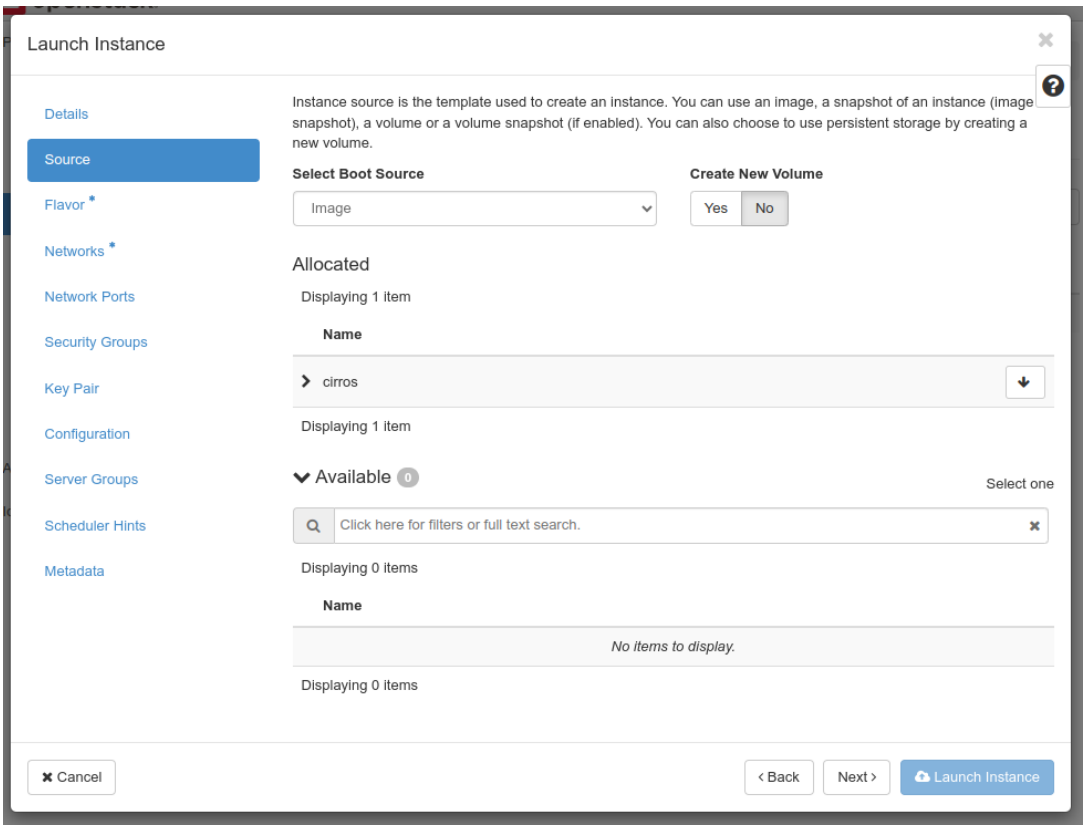


Рисунок 3.31 – Вибір образу CirrOS

Далі обираємо тип інстансу (*instance flavour*, у термінології OpenStack). Різні типи передбачають різні обсяги ресурсів для віртуальної машини. Наприклад, тип *m1.tiny* споживає небагато ресурсів, потребуючи виділення одного віртуального ЦП, 512 Мб оперативної пам'яті та 1 Гб місця на диску. Такі обсяги цілком відповідають потребам тестового інстансу з CirrOS, однак для сучасних повнофункціональних ОС часто може знадобитися більше ресурсів.

Різні типи інстансів доступні за замовчуванням, однак можна створювати власні – за допомогою Horizon чи засобів командного рядка.

Обираємо тип інстансу *m1.tiny* (рис. 3.32) та переходимо до наступного кроку.

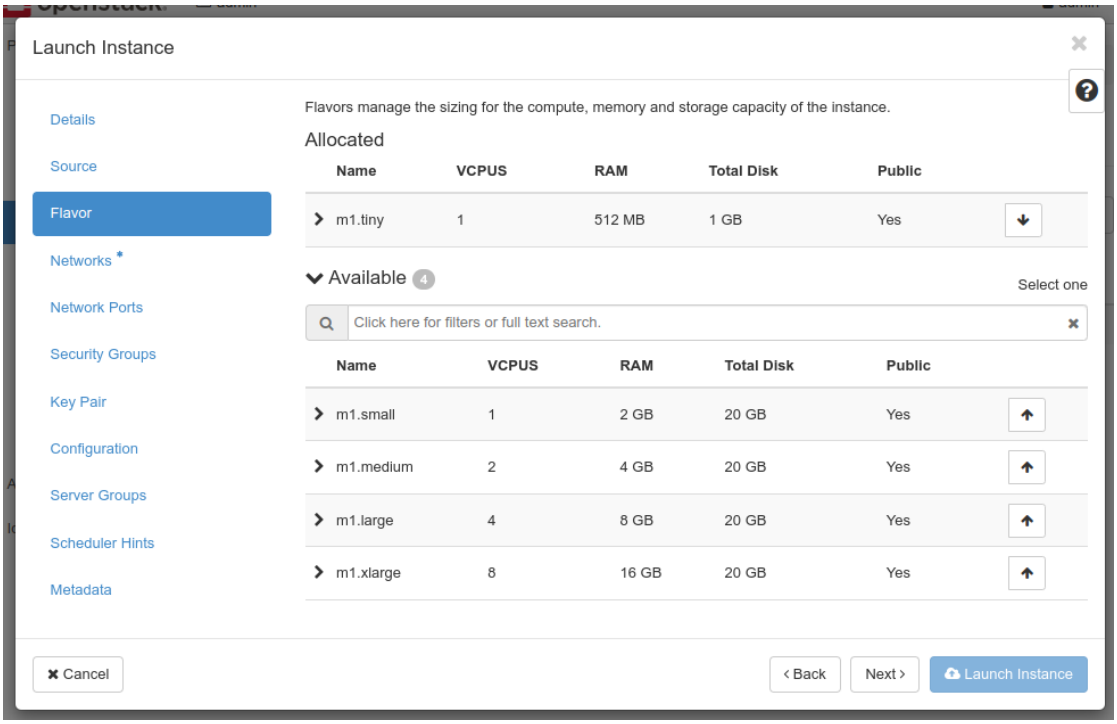


Рисунок 3.32 – Вибір типу інстансу

Віртуальна мережа, до якої буде підключено інстанс – це створена раніше внутрішня мережа *lab-network* (рис. 3.33).

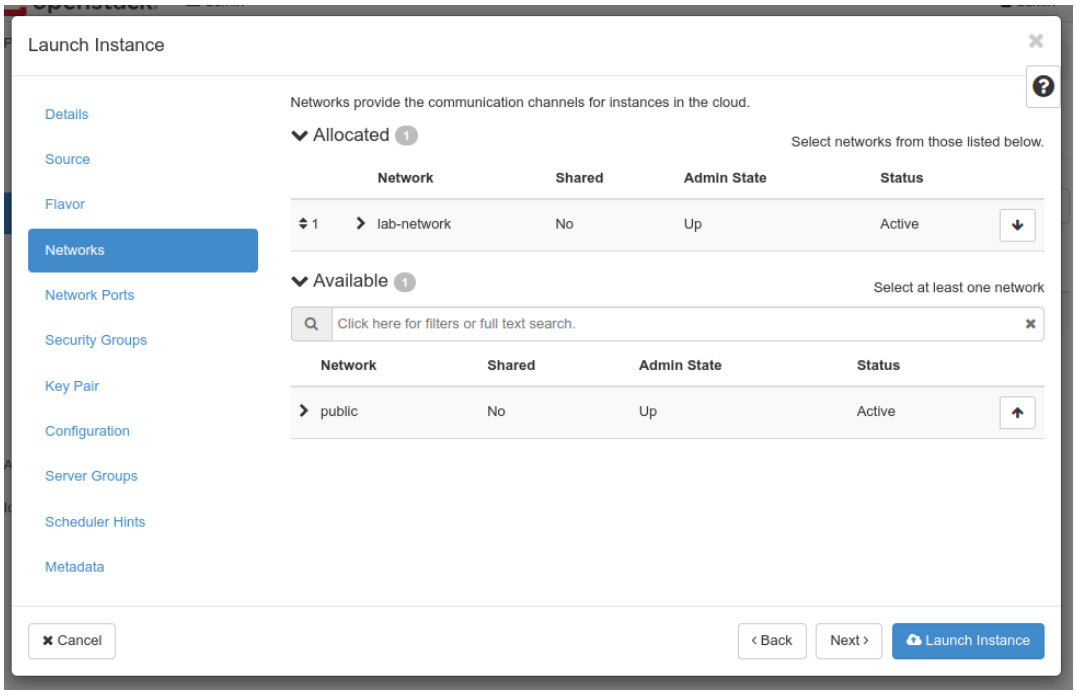


Рисунок 3.33 – Вибір віртуальної мережі для інстансу

Безпекову групу також було створено вище – *lab-sg* (рис. 3.34).

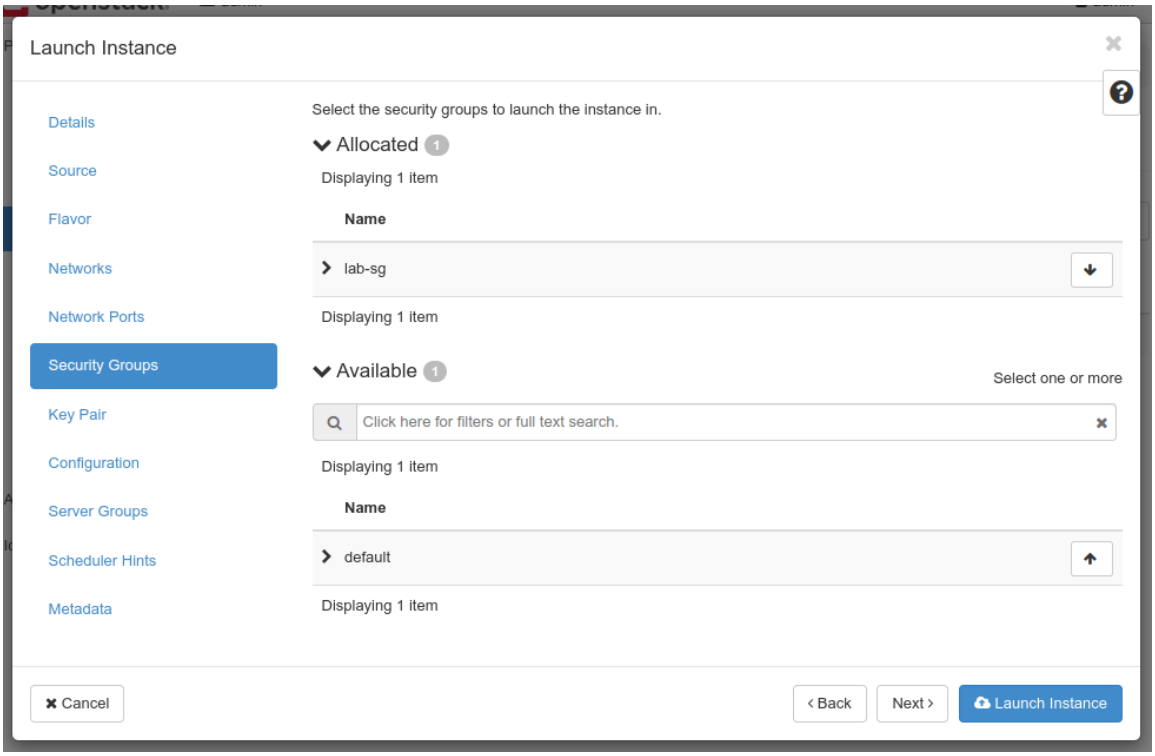


Рисунок 3.34 – Вибір безпекової групи для інстансу

Решту налаштувань для тестового інстансу лишаємо без змін і запускаємо інстанс. Коли інстанс успішно створиться, прив’язуємо до нього раніше виділену плаваючу адресу (рис. 3.35-3.36).

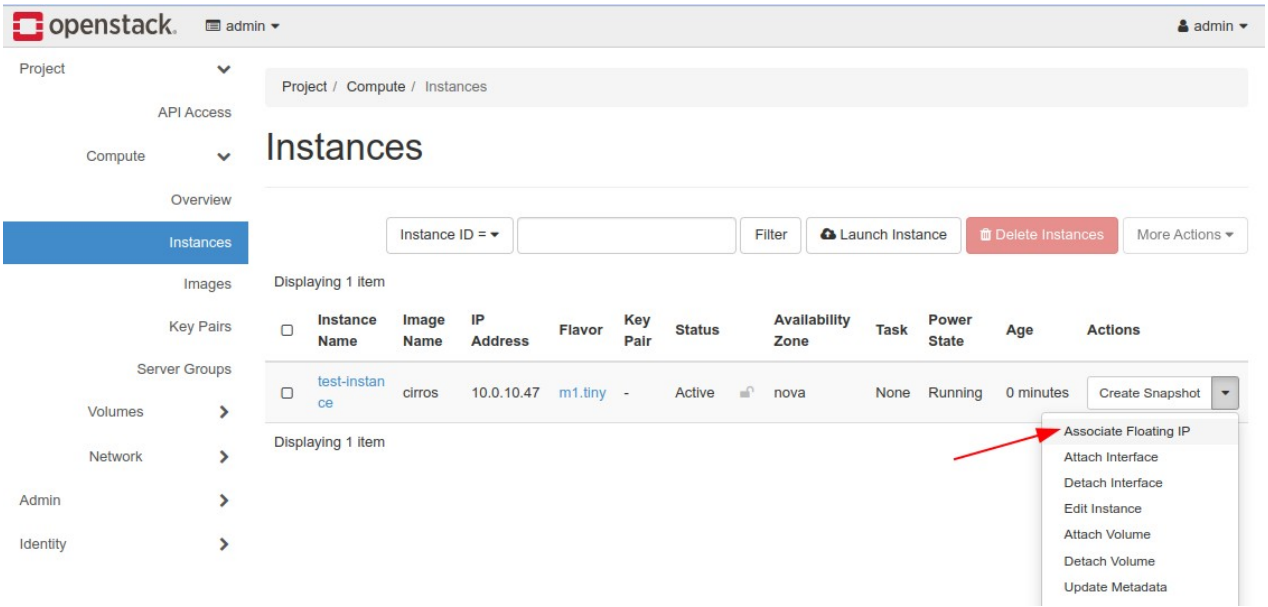


Рисунок 3.35 – Вибір опції прив’язування плаваючої адреси до інстансу

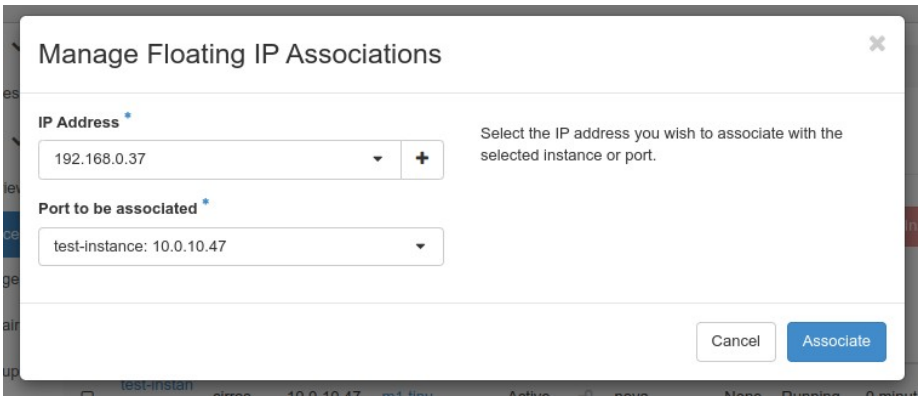


Рисунок 3.36 – Прив’язування попередньо виділеної плаваючої адреси до інстансу

За замовчуванням, інстанс запускається одразу після того, як його було успішно створено. Прив’язування плаваючої адреси може відбуватися і коли інстанс ще не запустився до кінця, однак для повноцінної роботи з інстансом потрібно зачекати, доки гостьова ОС інстансу повністю завантажиться. Перевірити, чи це так, можна через веб-інтерфейс Horizon, на вкладці інстансу *Console*).

Коли інстанс завантажився, перевіряємо наявність з’єднання з ним. На рис. 3.26 показано тестування з’єднання з інстансом з ноутбука, на якому розгорнуто хмару. Для команди *ping* у цьому випадку важливо вказати не той мережний інтерфейс, до якого було раніше прив’язано інтерфейс *br-ex*, а інший активний фізичний інтерфейс. У даному випадку це бездротовий інтерфейс *wlo1* (рис. 3.37), його потрібно попередньо ввімкнути. Як бачимо, перевірка з’єднання з інстансом успішна.

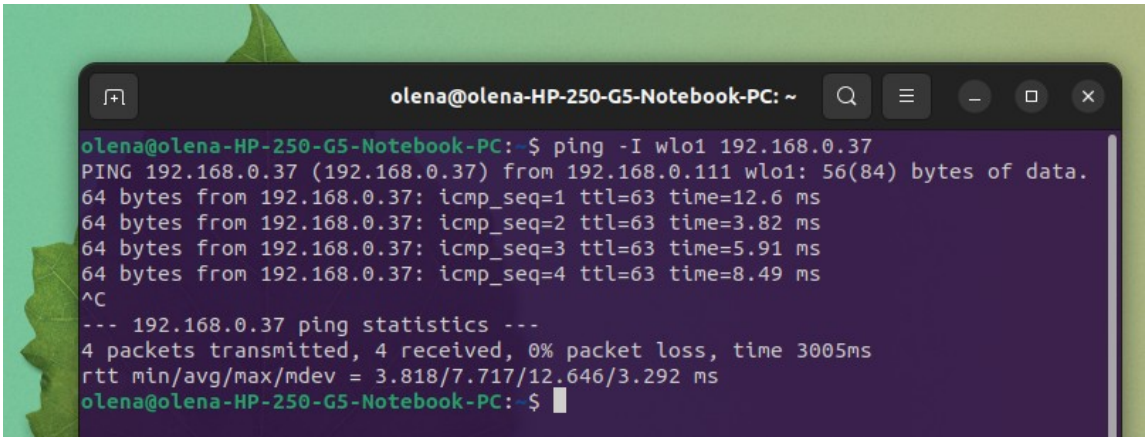


Рисунок 3.37 – Перевірка з’єднання з інстансом *test-instance* з ноутбука з ОС Ubuntu Desktop 22.04

Далі здійснюємо перевірку з’єднання з інстансом з іншого фізичного пристрою мережі – ноутбука з ОС Windows 10. Перевірка також успішна (рис. 3.38).

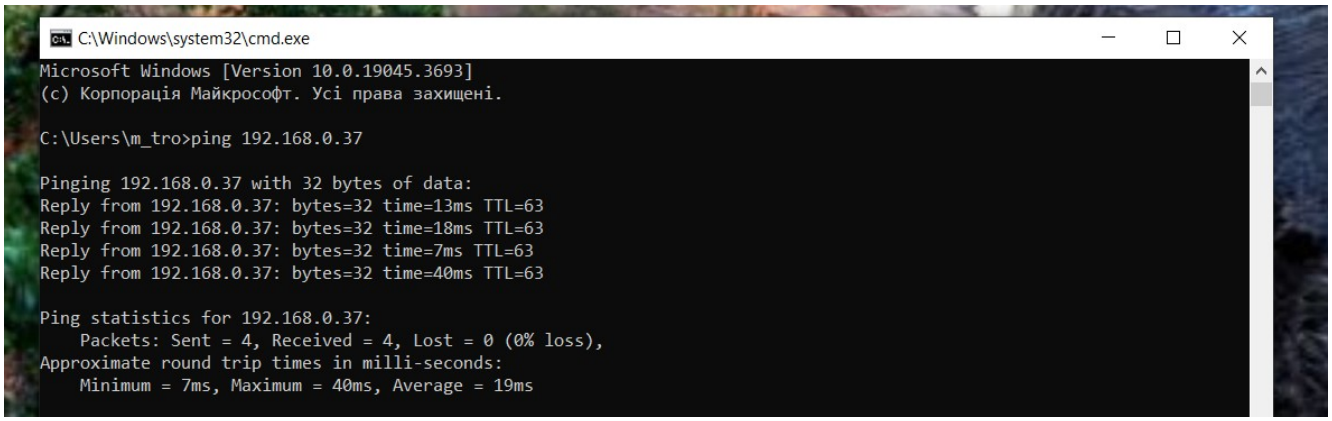


Рисунок 3.38 – Перевірка з’єднання з інстансом *test-instance* з ноутбука з ОС Windows 10

Отже, основні мережні налаштування працюють. Зокрема, було досягнуто важливого для даного проєкту хмарної лабораторії результату: доступ до хмарного інстансу є не лише з ноутбука, на якому розгорнуто хмару, а й з іншого фізичного пристрою мережі.

Водночас, необхідно також провести тестування з’єднання з інстансом з інших фізичний пристроїв з іншими ОС, а також можливість доступу до інстансу за допомогою протоколу SSH. Ці перевірки описано у п. 3.2.

Важливо врахувати, що ОС CirrOS не є повнофункціональним дистрибутивом Linux і розрахована передусім на тестування хмарної інфраструктури. І, хоч у CirrOS можна успішно виконувати базові команди Linux, все ж таки задля досягнення більшого реалізму доцільно використати для інстансів й інші дистрибутиви.

Для прикладу обрано дистрибутив Ubuntu, випуск Ubuntu Server 22.04 LTS. Це популярний сучасний Debian-сумісний дистрибутив, серверна версія якого встановлюється з інтерфейсом командного рядка. Додатковим аргументом на користь вибору Ubuntu є й те, що Ubuntu є наразі базовим дистрибутивом Linux, який використовується для навчання операційних систем у Державному універси-

теті “Житомирська політехніка”, для якого й призначено хмарну лабораторію. Водночас, у хмарі на базі MicroStack/OpenStack можна запустити й інші дистрибутиви Linux.

Готові образи Ubuntu у даній хмарі наразі відсутні, тому потрібно спершу створити власний. Завантажимо один зі спеціальних хмарних образів Ubuntu з офіційного ресурсу Canonical [34] (рис. 3.39).

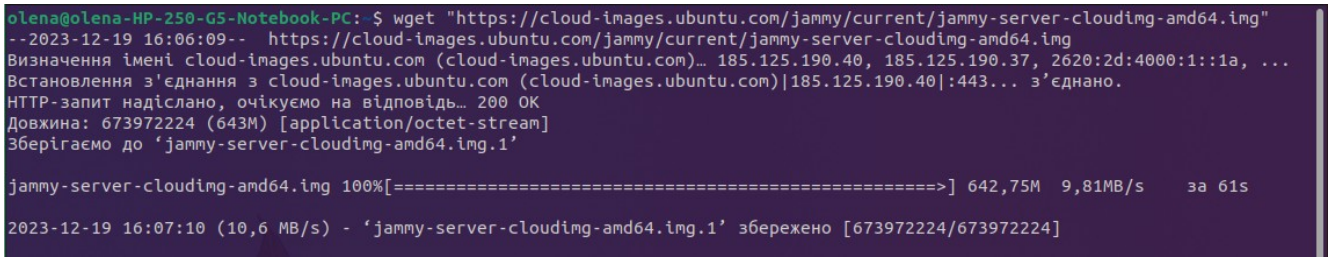


Рисунок 3.39 – Завантаження хмарного образу Ubuntu Server 22.04 LTS

За замовчуванням даний образ не дозволяє підключення через протокол SSH за допомогою логіну та паролю, а передбачає лише використання ключа. І, хоч підхід з відкритим і закритим ключем підвищить безпеку хмарної інфраструктури, для початкового налаштування і тестування хмари буде зручніше мати доступ за логіном та паролем. Для цього кастомізуємо завантажений образ Ubuntu за допомогою інструменту *virt-customize*. Спершу підготуємо файл з паролем *~/ubuntu-pass*, зберігши у ньому пароль, що відповідає сучасним вимогам складності, та заборонивши доступ до цього файлу всім, крім користувача-власника *olena* (рис. 3.40).

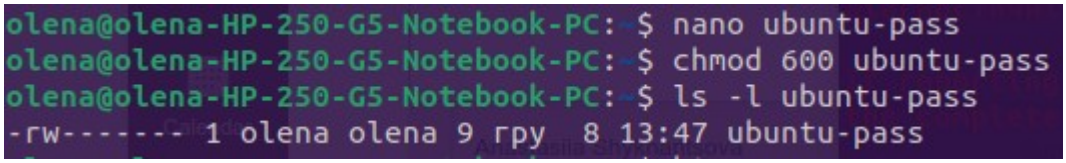


Рисунок 3.40 – Підготовка файлу з паролем для Ubuntu Server 22.04

Далі створимо новий образ Ubuntu Server 22.04 на основі наявного, встановивши підготовлений пароль для користувача *root* (рис. 3.41).

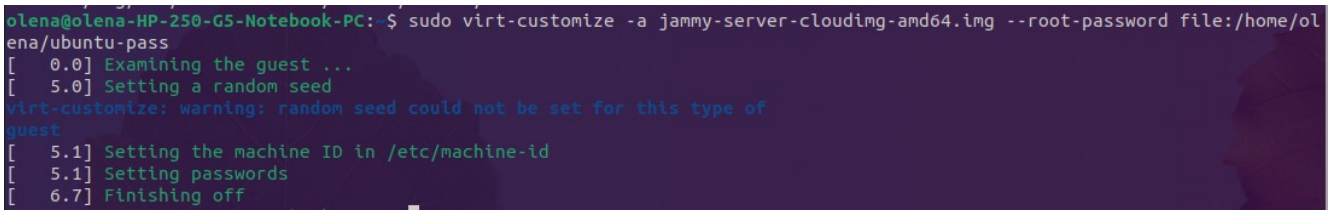


Рисунок 3.41 – Підготовка файлу з паролем для Ubuntu Server 22.04

На основі кастомізованого інсталяційного образу створюємо хмарний образ *ubuntu-jammy* для MicroStack/OpenStack (рис. 3.42).

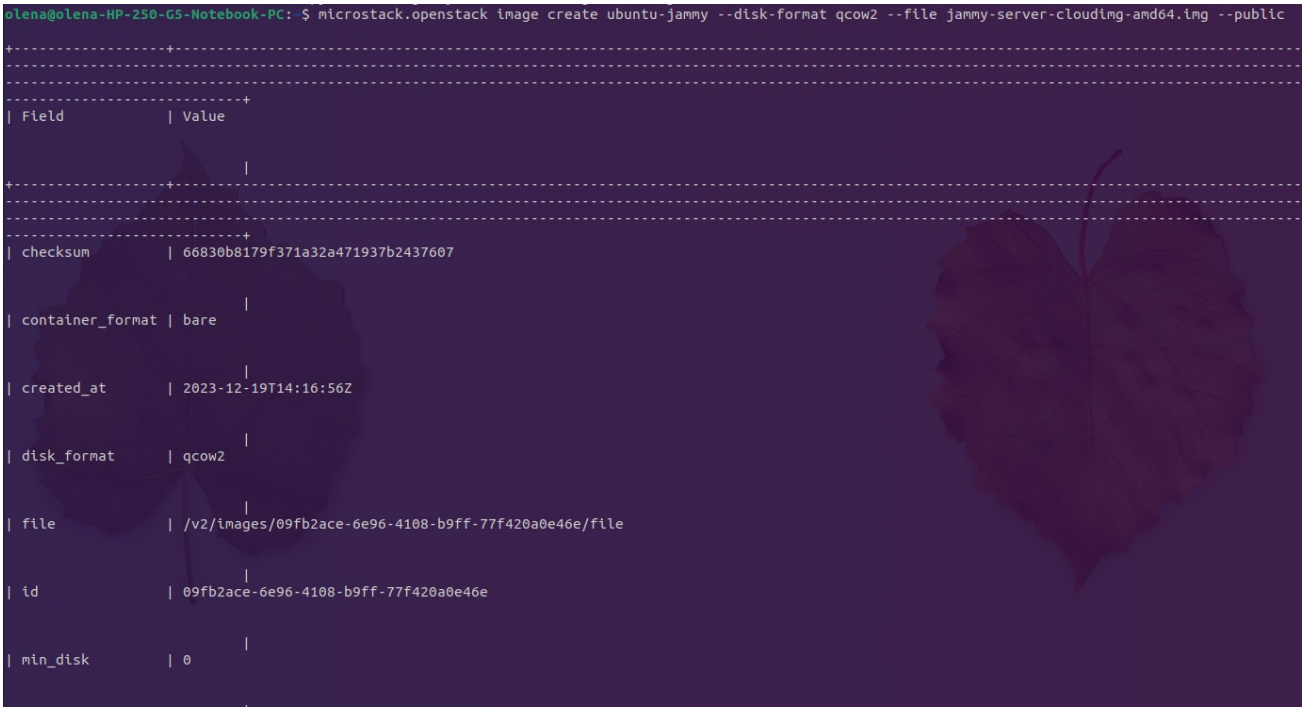


Рисунок 3.42 – Створення хмарного образу *ubuntu-jammy* (скриншот обрізано, насправді вивід довший)

Тепер у хмарі доступно два образи – *cirros* та *ubuntu-jammy* (рис. 3.43).

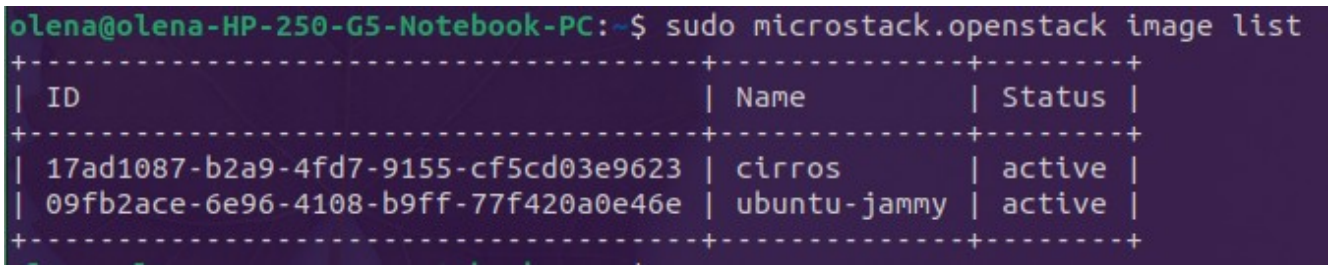


Рисунок 3.43 – Образи, наразі доступні в хмарі

Інстанс з Ubuntu з ім’ям *ubuntu-test* створимо засобами командного рядка (рис. 3.44).

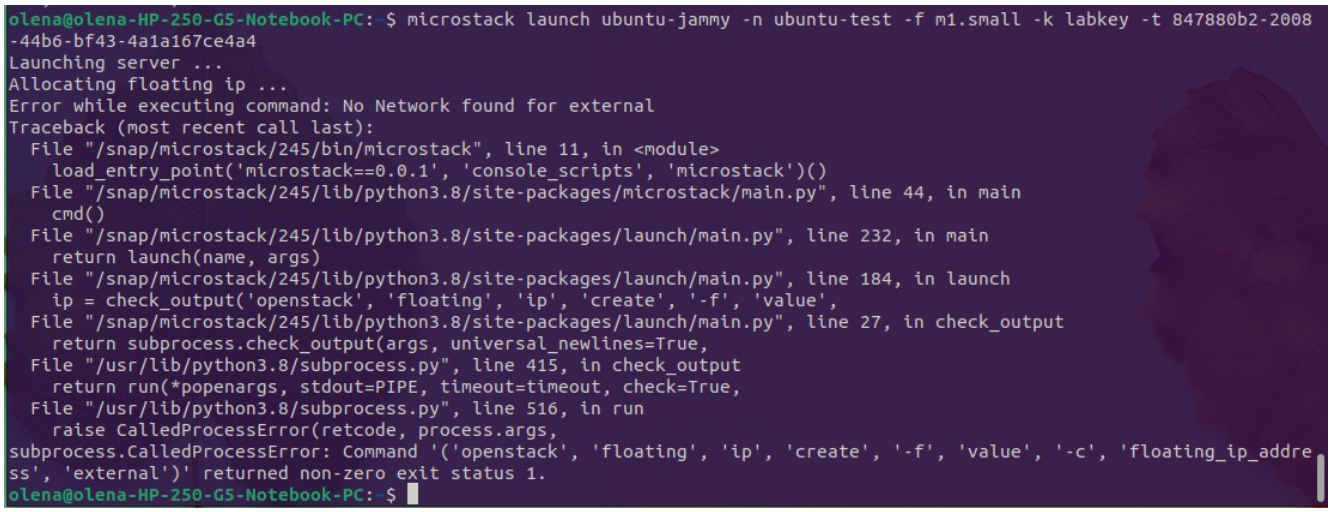


Рисунок 3.44 – Запуск інстансу *ubuntu-test*

Створення загалом відбулося успішно, за винятком плаваючої IP-адреси. Виділяємо ще одну плаваючу адресу (192.168.0.16) та прив’язуємо її до інстансу *ubuntu-test* вручну (рис. 3.45).

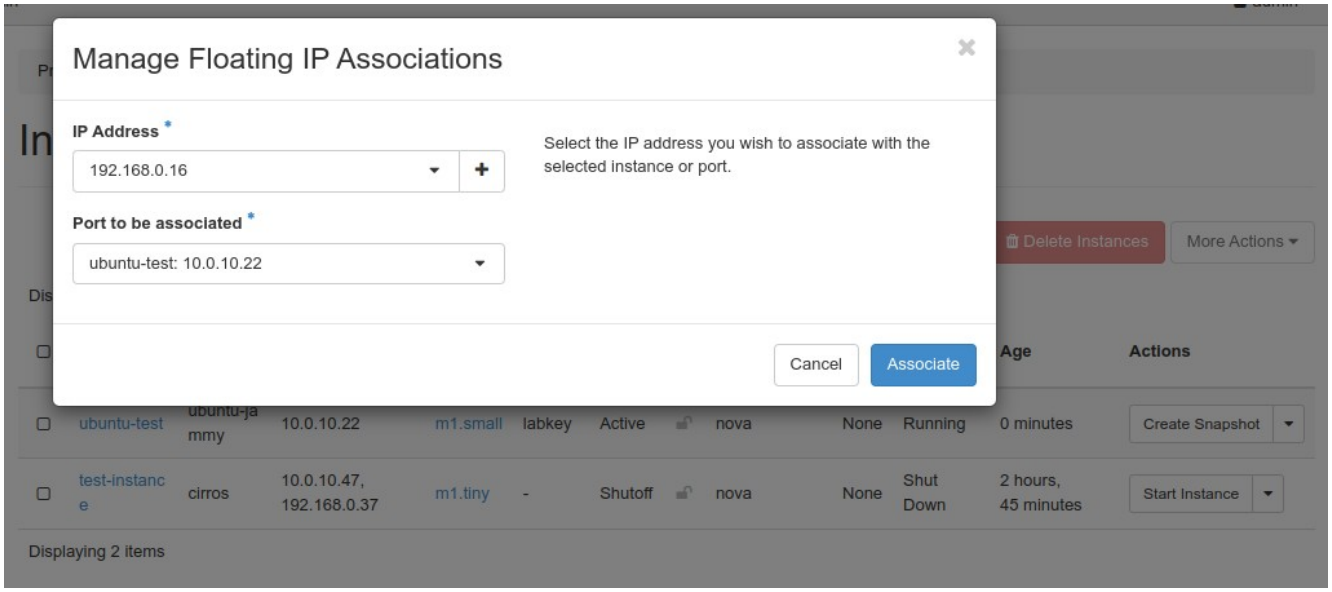


Рисунок 3.45 – Прив’язування плаваючої IP-адреси до інстансу *ubuntu-test*

Тепер інстанс з Ubuntu має не лише локальну адресу 10.0.10.22, а й плаваючу адресу 192.168.0.16, за якою до нього можна звертатися ззовні (рис. 3.46).

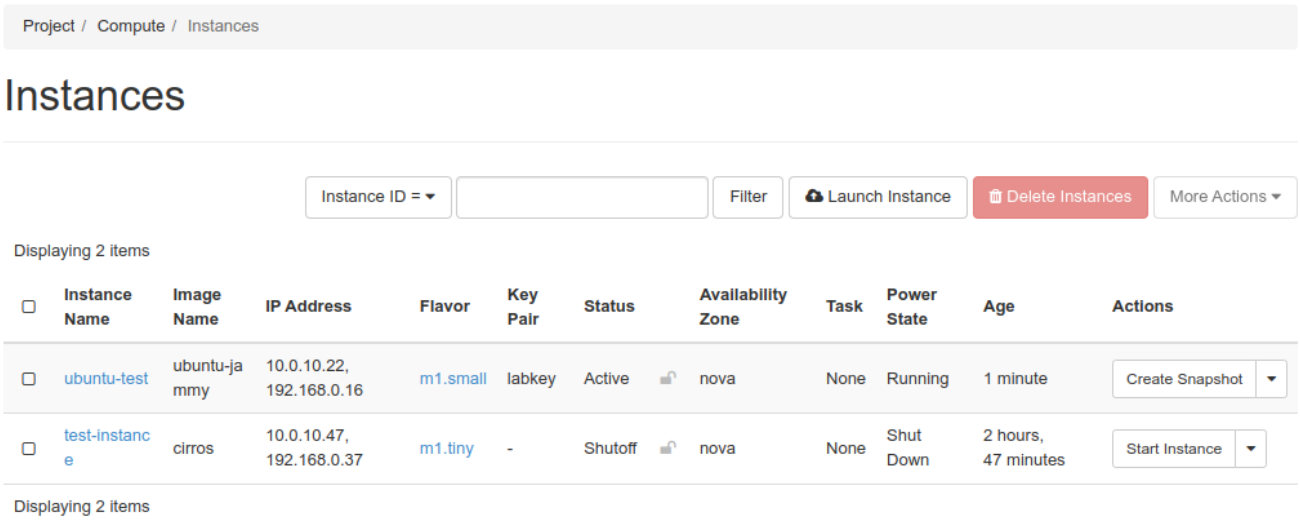


Рисунок 3.46 – Інстанси *ubuntu-test* та *test-instance* в інтерфейсі Horizon

За допомогою розділу Console у властивостях інстансу *ubuntu-test* маємо змогу переконатися, що інстанс успішно запустився і працює (рис. 3.47).

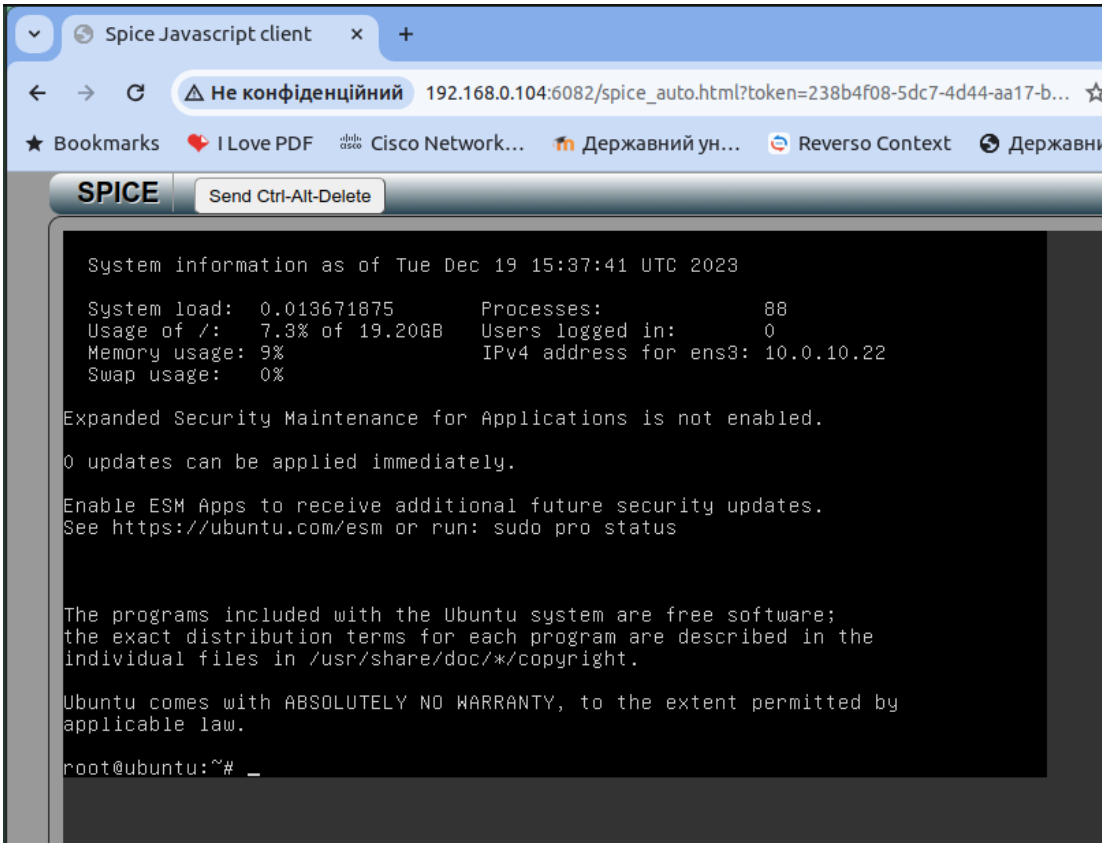


Рисунок 3.47 – CLI-термінал інстансу *ubuntu-test* в засобі *Console* веб-інтерфейсу Horizon

Подальші перевірки роботи проєкту хмарної лабораторії описано у п. 3.2.

3.2 Перевірка роботи проєкту хмарної лабораторії

Протестуємо доступ до інстансів за допомогою протоколу SSH. Доступ через протокол SSH з ноутбука з ОС Windows 10 працює коректно (рис. 3.48).

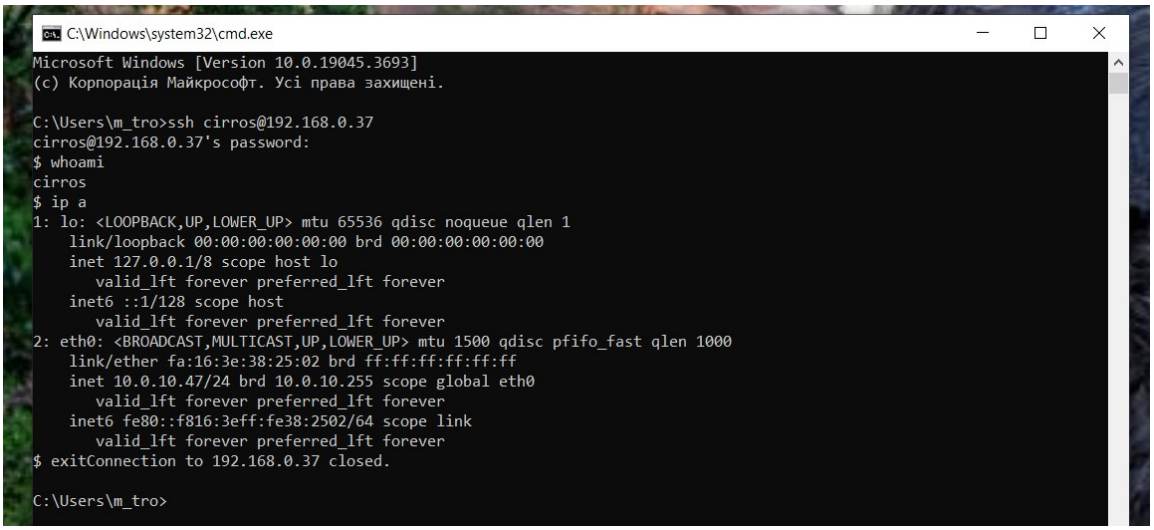


Рисунок 3.48 – Перевірка доступу до інстансу *test-instance* з ноутбука з ОС Windows 10 через протокол SSH

Для наступної перевірки використано ноутбук з macOS Monterey. Перевірка зв’язку за допомогою команди *ping* успішна, доступ через протокол SSH наявний (рис. 3.49).

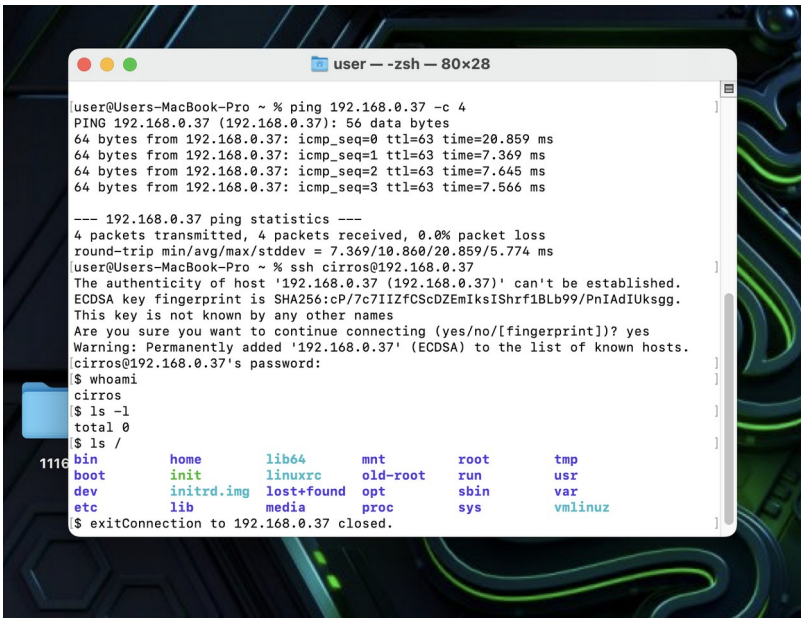


Рисунок 3.49 – Перевірка з’єднання з інстансом *test-instance* та доступу до нього через протокол SSH з ноутбука з macOS

Для тестування доступу з мобільних пристроїв використано смартфон Nokia 5.3 з ОС Android 12 та застосунок Termius. Налаштування профілю з’єднання з інстансом у мобільному застосунку Termius показано на рис. 3.50.

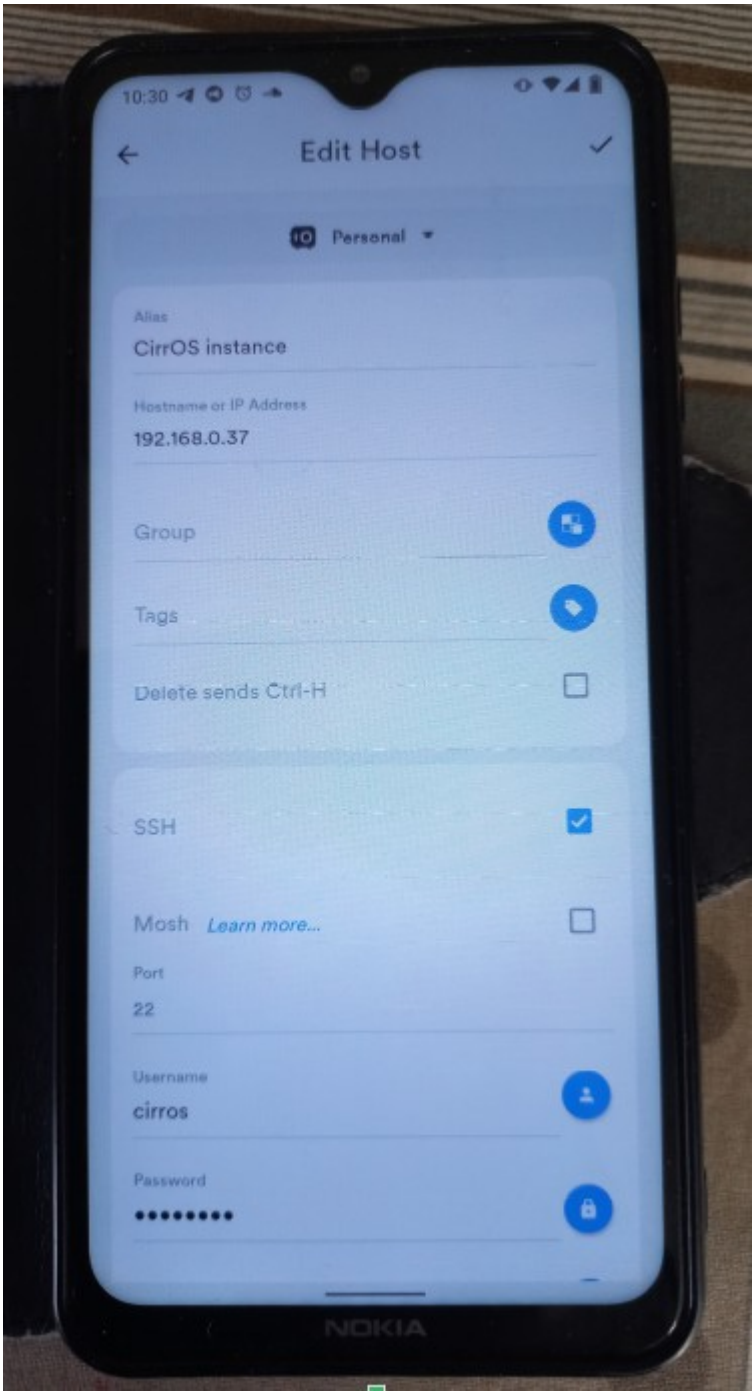


Рисунок 3.50 – Перевірка доступу до інстансу *test-instance* через протокол SSH зі смартфону Nokia 5.3 (налаштування)

На рис. 3.51 показано роботу у терміналі інстансу після успішного з’єднання засобами протоколу SSH.

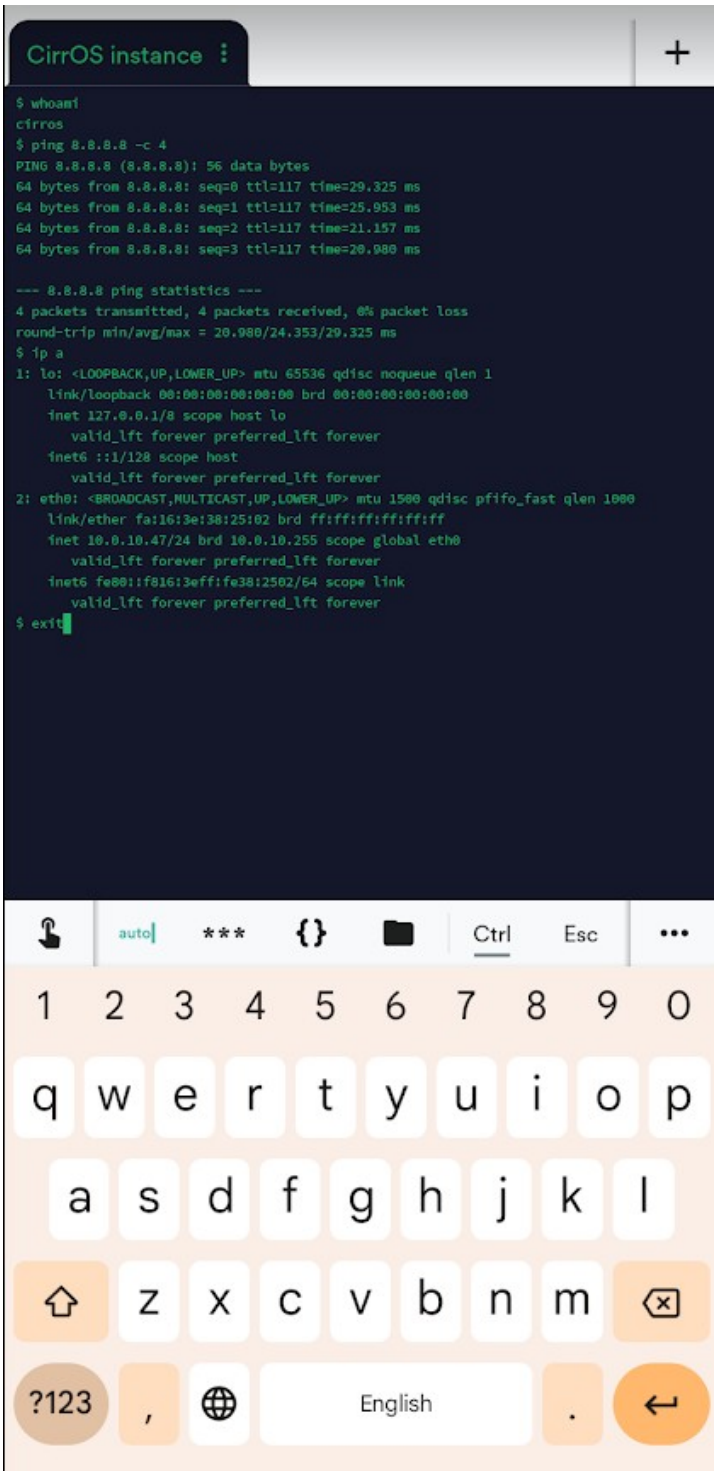


Рисунок 3.51 – Перевірка доступу до інстансу *test-instance* через протокол SSH зі смартфону Nokia 5.3 (термінал)

Також, як видно з рис. 3.51 інстанси також мають доступ до Інтернету, що розширює їхні можливості (наприклад, дозволяє навчати студентів використанню команд, які потребують такого доступу), але водночас створює безпекові виклики, що їх необхідно вирішити у майбутньому.

На рис. 3.52, 3.53 продемонстровано результати перевірки з'єднання всередині хмари між інстансами *test-instance* та *ubuntu-test*. Як бачимо, з'єднання працює як за внутрішніми адресами, так і за плаваючими адресами, що у майбутньому може бути використано для вивчення мережних функцій Linux.

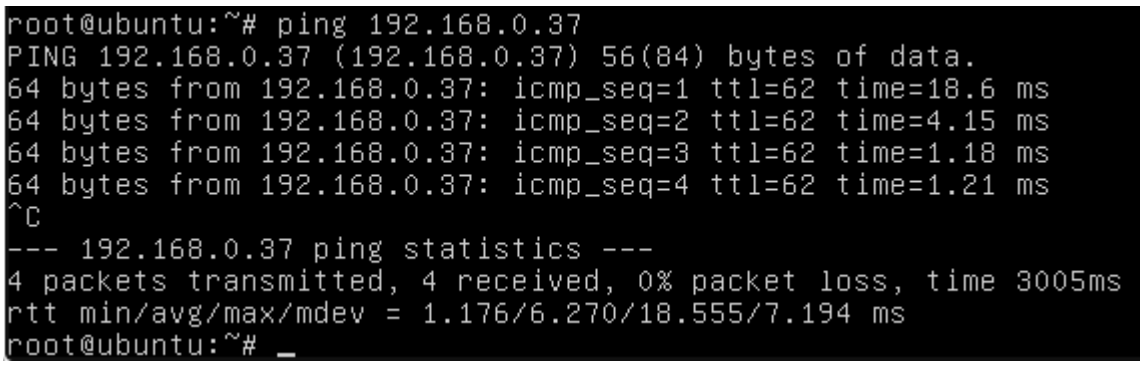


Рисунок 3.52 – Перевірка з'єднання інстансу *ubuntu-test* з інстансом *test-instance* за плаваючою IP-адресою

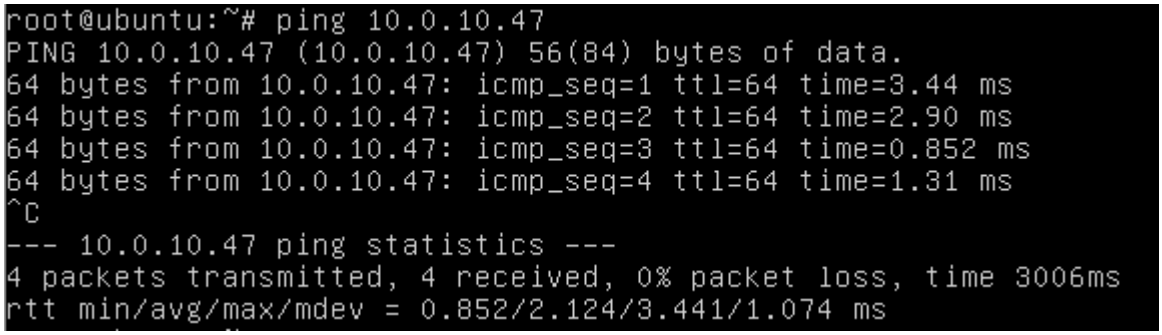


Рисунок 3.53 – Перевірка з'єднання інстансу *ubuntu-test* з інстансом *test-instance* за внутрішньою IP-адресою

Крім того, можливе також з'єднання між інстансами за протоколом SSH (рис. 3.54), що також може бути використано для навчання мережних засобів.

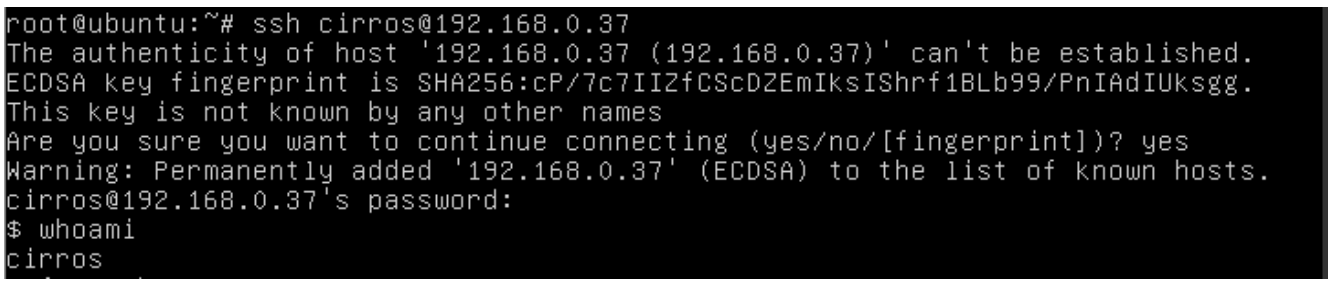


Рисунок 3.54 – Перевірка доступу з інстансу *ubuntu-test* до інстансу *test-instance* за протоколом SSH

Веб-інтерфейс Horizon доступний за адресою https://10.20.20.1 (рис. 3.55).

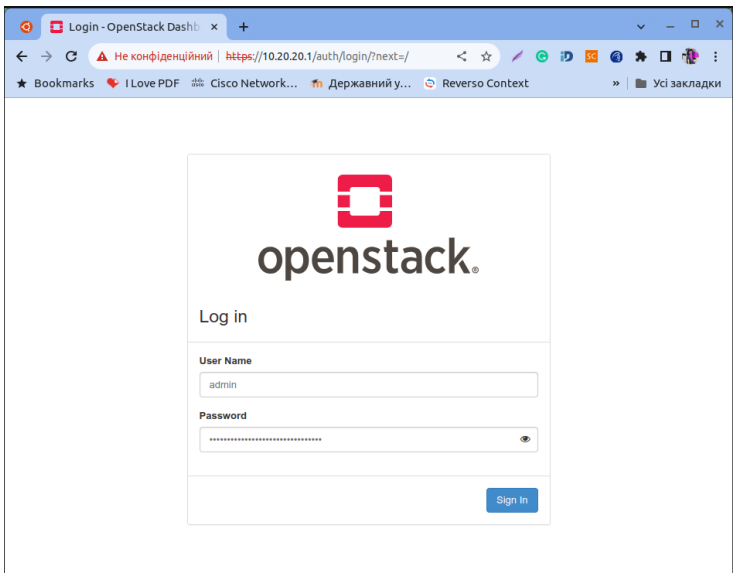


Рисунок 3.55 – Вхід у веб-інтерфейс Horizon

Після успішного входу в Horizon відображається панель керування OpenStack, де можна переглянути узагальнені відомості про хмару, а також детальну інформацію про інстанси, віртуальні диски, мережну інфраструктуру, образи операційних систем тощо. На рис. 3.56 показано вигляд інтерфейсу Horizon після створення першого тестового інстансу.

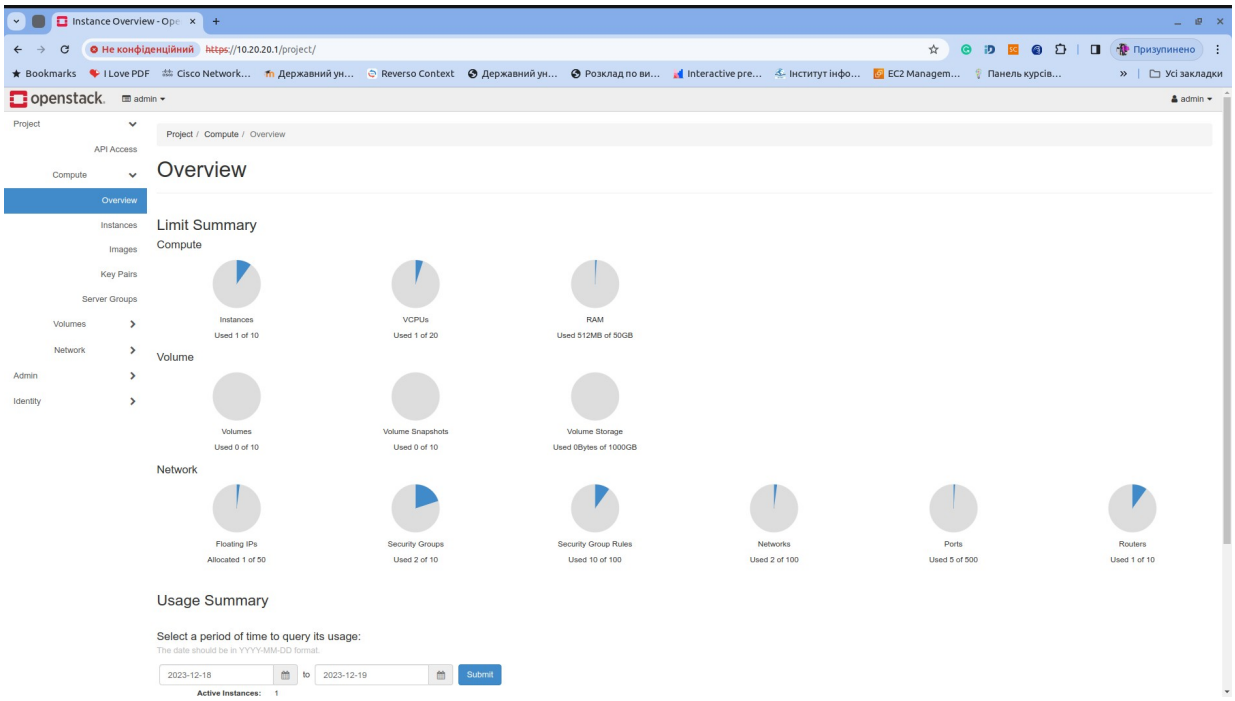


Рисунок 3.56 – Панель керування хмарною інфраструктурою у Horizon

Інтерфейс Horizon також дає змогу відстежувати обсяги використання ресурсів. Наприклад, на рис. 3.57 показано відомості про використання віртуальних процесорів, диску, оперативної пам’яті та інших ресурсів за період з 18.12.2023 по 19.12.2023. Також унизу скриншота видно відомості про максимальні обсяги ресурсів, доступні для інстансу *test-instance* (1 віртуальний процесор, 1 Гб дискового простору, 512 Мб оперативної пам’яті).

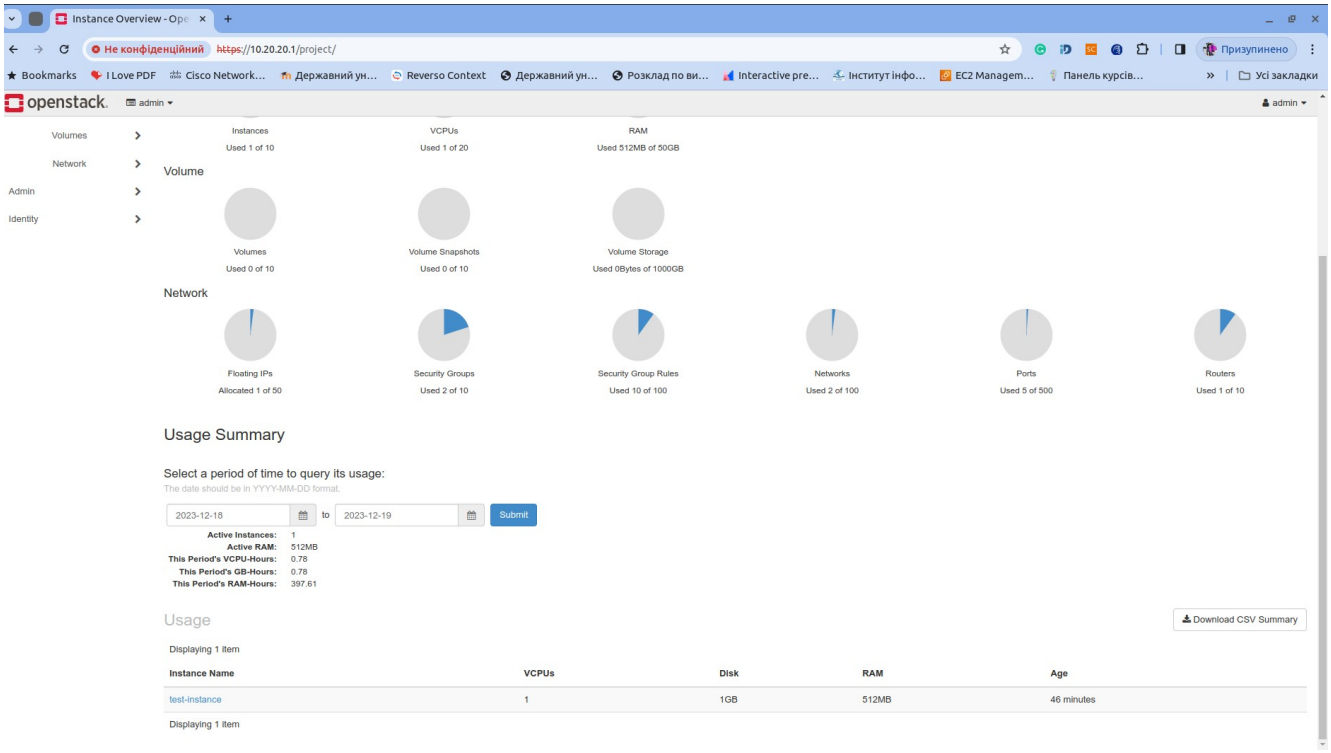


Рисунок 3.57 – Інформація про обсяги використання ресурсів

У випадку повномасштабної реалізації проєкту хмарної лабораторії доцільно також налаштувати максимальні доступні обсяги ресурсів у хмарі, залежно від наявних апаратних потужностей.

У категорії *Networks* відображається топологія хмарної інфраструктури, яку можна використовувати не лише з метою візуалізації, а й з метою перегляду та зміни основних мережних налаштувань (рис. 3.58). На рис. 3.58 показано обидві мережі – *public* та *lab-network*, а також віртуальний роутер *public-router* між ними та обидва інстанси, під’єднані до мережі *lab-network*.

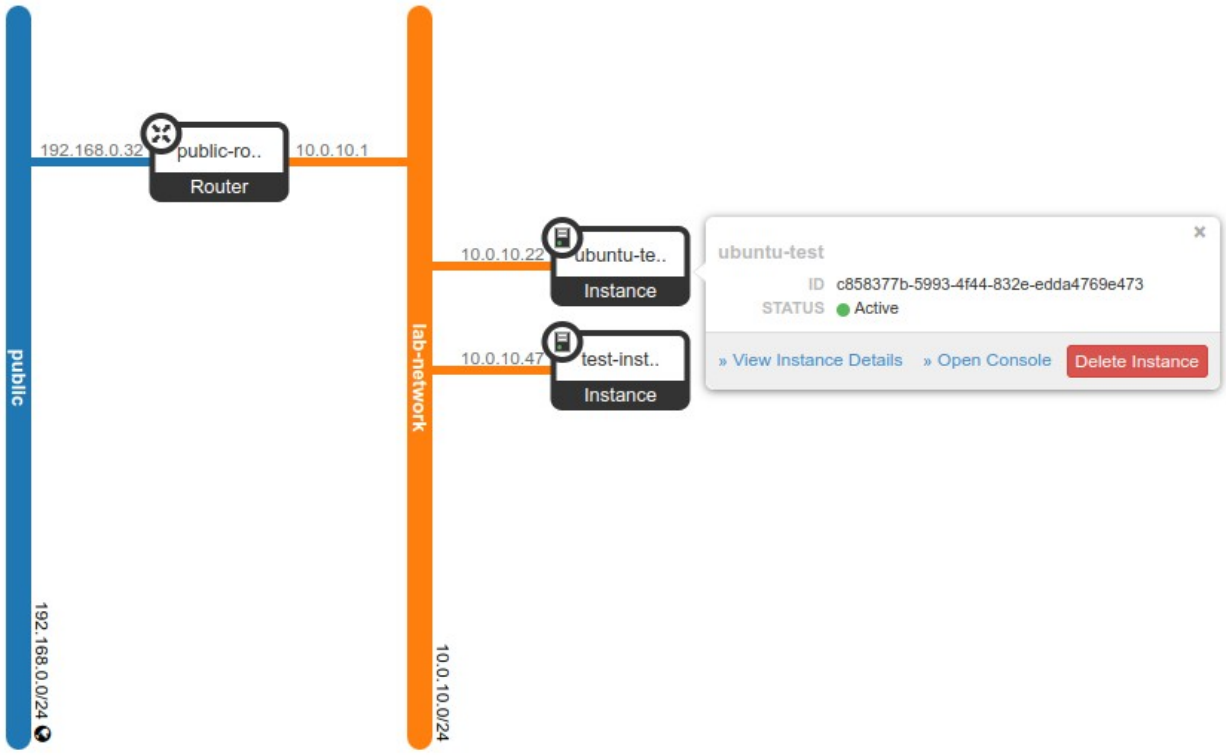


Рисунок 3.58 – Інтерактивна візуалізація топології хмарної інфраструктури у Horizon

Також важливо вимикати інстанси, коли вони не використовуються, та видаляти непотрібні інстанси.

Таким чином, бачимо, що розгортання проєкту хмарної лабораторії відбулося успішно. Інстанси працюють коректно, мають з'єднання одне з одним за протоколами ICMP та SSH, з'єднання з мережею Інтернет. Також наявне з'єднання з інстансами та доступ до інстансу за протоколом SSH з інших фізичних пристроїв мережі (ноутбук з ОС Windows 10, ноутбук з ОС mac OS Monterey, смартфон Nokia 3.5 з ОС Andriod 12 та встановленим застосунком Termius.

Подальший розвиток проєкту хмарної лабораторії доцільно здійснювати за напрямом дослідження та впровадження інструментів для посилення захисту хмарних інстансів від несанкціонованого доступу.

Висновки з розділу 3

У розділі 3 здійснено налаштування та тестування проєкту хмарної лабораторії. Було встановлено та проініцілізовано хмару на основі рішення MicroStack. Створено хмарну мережну інфраструктуру, відмінну від стандартної, задля забезпечення доступу до хмарних інстансів з інших фізичних пристроїв мережі. Окрім стандартного тестового інстансу з CirrOS, також запущено інстанс з ОС Ubuntu Server 22.04 LTS на основі хмарного інсталяційного образу, попередньо кастомізованого для потреб хмарної лабораторії. Успішно протестовано мережне з'єднання з інстансами та доступ до інстансу за протоколом SSH. Досліджено та випробувано можливості веб-інтерейсу Horizon.

Розгортання проєкту хмарної лабораторії відбулося успішно. Напрямом подальшого розвитку хмарної лабораторії визначено дослідження та впровадження інструментів захисту хмарних інстансів від несанкціонованого доступу.

ВИСНОВКИ

У межах кваліфікаційної роботи розроблено проєкт хмарної лабораторії для навчання ОС Linux на базі MicroStack/OpenStack.

У першому розділі кваліфікаційної роботи розглянуто теоретичні особливості побудови хмарної лабораторії. Досліджено поняття хмарних технологій та основні моделі, за якими може бути розгорнуто хмару. Розглянуто поняття хмарної лабораторії, здійснено огляд можливостей її освітнього та дослідницького застосування. Виокремлено основні підходи до організації хмарної інфраструктури (використання платформ, наданих провайдером хмарних сервісів; використання спеціалізованого хмарного програмного забезпечення). Досліджено ключові відмінності основних хмарних топологій (одновузлової та багатовузлової). Проаналізовано можливості для розгортання хмарної інфраструктури як хмарних платформ наданих провайдерами (Amazon EC2, Google Compute Engine, Azure Virtual Machines), так і спеціалізованого хмарного програмного забезпечення (Eucalyptus, OpenStack, CloudStack, OpenNebula).

У другом розділі описано проєктування хмарної лабораторії. Сформульовано вимоги до проєкту хмарної лабораторії. Вивчено можливості популярних хмарних платформ та спеціалізованого вільно поширюваного хмарного ПЗ. Внаслідок порівняльного аналізу наявних платформ, обрано рішення MicroStack (платформа OpenStack) для подальшої реалізації проєкту хмарної лабораторії. Розроблено топологію та схему адресації моделі мережі для проєкту хмарної лабораторії.

У третьому розділі кваліфікаційної роботи здійснено налаштування та тестування проєкту хмарної лабораторії. Побудовано хмарну мережну інфраструктуру, яка дозволяє отримувати доступ до хмарних інстансів з інших фізичних пристроїв мережі. Запущено інстанс з тестовою ОС CirrOS на основі вбудованого образу та інстанс з ОС Ubuntu Server 22.04 LTS на основі хмарного інсталяційного образу, попередньо кастомізованого для потреб хмарної лабораторії. Перевірено працездатність мережного з’єднання з інстансами, а також доступ до інстансу за протоколом SSH.

Розроблений проєкт хмарної лабораторії може бути покладено в основу робочої хмарної лабораторії для навчання ОС Linux у Державному університеті “Житомирська політехніка”. Подальші дослідження доцільно спрямувати та вивчення та впровадження інструментів захисту інстансів хмарної лабораторії від несанкціонованого доступу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Apache CloudStack – Open Source Cloud Computing. – Access mode: https://cloudstack.apache.org/.
2. Arnold C. Cloud labs: where robots do the research. Nature, 2022 Jun, 606(7914):612-613. doi: 10.1038/d41586-022-01618-x.
3. AWS Cloud9 Amazon Web Services. – Access mode: https:// aws.amazon.com/cloud9
4. Azure paired regions. – Access mode: https://learn.microsoft.com/en-us/ azure/reliability/cross-region-replication-azure#azure-paired-regions
5. Bedi P., Deep B., Kumar P., Sarna P. Comparative Study of OpenNebula, CloudStack, Eucalyptus and OpenStack. International Journal of Distributed and Cloud Computing 6(1), 37–42 (2018)
6. Chapter 1. Components Red Hat OpenStack Platform 9 | Red Hat Customer Portal. – Access mode: https://access.redhat.com/documentation/en-us/ red_hat_openstack_platform/9/html/architecture_guide/components
7. Compute Engine – Google Cloud. – Access mode: https://cloud.google.com/ compute/
8. Create Your Azure Free Account Today – Microsoft Azure. – Access mode: https://azure.microsoft.com/en-us/free#all-free-services
9. Documentation/training-labs – OpenStack. – Access mode: https:// wiki.openstack.org/wiki/Documentation/training-labs
10. Donald Sebastian Leung. Introduction to OpenStack with MicroStack. – Access mode: https://dev.to/donaldsebleung/introduction-to-openstack-with-microstack-1f5j
11. Euca2ools 3.4.1 Reference Guide. – Access mode: https:// docs.eucalyptus.cloud/eucalyptus/4.4.3/euca2ools-guide-3.4.1.pdf.

12. Eucalyptus Documentation,. – Access mode: https://docs.eucalyptus.cloud/eucalyptus/5/.
13. GitHub – OpenNebula_minione – Easy to use deployment tool for an OpenNebula evaluation environment. – Access mode: https://github.com/OpenNebula/minione.
14. Holovnia O. Linux online virtual environments in teaching operating systems. Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. 2020. Vol. 2. pp. 964-973.
15. Holovnia O., Oleksiuk V. Selecting cloud computing software for a virtual online laboratory supporting the Operating Systems course. Proceedings of the 9th Workshop on Cloud Technologies in Education. 2021. pp. 216-227
16. Install OpenStack – Ubuntu. – Access mode: https://ubuntu.com/openstack/install.
17. Install OpenStack services – Installation Guide documentation. – Access mode: https://docs.open-stack.org/install-guide/openstack-services.html.
18. Install OpenStack yourself on Ubuntu – Ubuntu. – Access mode: https://ubuntu.com/openstack/install
19. Installation overview – Apache CloudStack Installation Documentation 4.6.0 documentation. – Access mode: http://docs.cloudstack.apache.org/projects/cloudstack-installation/en/4.6/overview/.
20. Installing, updating, and uninstalling the AWS CLI version 2 – AWS Command Line Interface, https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html.
21. Introduction to OpenStack – Security Guide documentation. – Access mode: https://docs.openstack.org/security-guide/introduction/introduction-to-open-stack.html
22. Introduction to OpenStack with MicroStack – DEV Community. – Access mode: https://dev.to/donaldsebleung/introduction-to-openstack-with-microstack-1f5j

23. Mell P., Grance T. The NIST Definition of Cloud Computing. National Institute of Standards and Technology, 2011. – Access mode: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.
24. Nascimento R. OpenStack Single-Node (MicroStack). – Access mode: https://rnascimento.com/openstack-single-node-microstack/
25. Navigate the Dashboard. – Access mode: https://docs.eucalyptus.cloud/eucalyptus/4.4.5/shared/console_dashboard.html
26. Open Source Cloud Computing Platform Software – OpenStack. – Access mode: https://www.openstack.org/software/
27. OpenNebula – Open Source Cloud & Edge Computing Platform. – Access mode: https://opennebula.io/.
28. OpenNebula Overview – Datasheet – OpenNebula Customer Portal. – Access mode: https://support.opennebula.pro/hc/en-us/articles/360036935791-OpenNebula-Overview-Datasheet.
29. OpenStack – Access mode: https://wiki.openstack.org/wiki/Main_Page
30. OpenStack dashboard – openstackdocstheme 2.2.8.dev10 documentation, https://docs.openstack.org/openstackdocstheme/latest/demo/dashboard_demo.html.
31. Secure and resizable cloud compute – Amazon EC2 – Amazon Web Services. – Access mode: https://aws.amazon.com/ec2/?nc2=h_ql_prod_fs_ec2
32. Tanenbaum A. S., Bos H. Modern operating systems. 4th ed. Pearson, 2014.
33. TechnologyIntegrationPrinciples – OpenStack. – Access mode: https://wiki.openstack.org/wiki/TechnologyIntegrationPrinciples
34. Ubuntu Cloud Images – the official Ubuntu images for public clouds, Openstack, KVM and LXD. – Access mode: https://cloud-images.ubuntu.com/
35. Virtual Machines (VMs) for Linux and Windows – Microsoft Azure. – – Access mode: https://azure.microsoft.com/en-us/products/virtual-machines/

36. Welcome to Apache CloudStack's Documentation – Apache CloudStack 4.15.0.0 documentation. – Access mode: https://docs.cloudstack.apache.org/en/latest/index.html.
37. Олексюк В. П. Теоретико-методичні основи проєктування, адміністрування та використання хмаро орієнтованого середовища навчання майбутніх учителів інформатики. Дис. доктора пед. наук, Інститут цифровізації освіти, К., 2023.

ДОДАТКИ

Додаток А – **Порівняння основних характеристик популярних сервісів IaaS, які надаються провайдерами**

Таблиця А.1 – Платформи для побудови хмарної інфраструктури, надані провайдерами хмарних сервісів

Характеристики	**Amazon EC2**	**MS Azure Virtual Machines**	**Google Compute Engine**
Назва хмарної платформи	Amazon Web Services	Microsoft Azure	Google Cloud Platform
Модель надання хмарних послуг	IaaS	IaaS	IaaS
Назва окремої обчислювальної одиниці	інстанс / instance	віртуальна машина / virtual machine	інстанс / instance, віртуальна машина / virtual machine
Доступні операційні системи	Linux (Amazon Linux 2, Ubuntu, RHEL, CentOS, SUSE, Debian), Windows (Windows Server)	Linux (RHEL, SUSE, Ubuntu, CentOS, Debian, CoreOS), Windows (Windows Server)	Linux (Debian, Ubuntu, SUSE, RHEL, CoreOS), Windows (Windows Server)
Основні доступні регіони	US East, US West, Africa, Asia Pacific, Canada, Europe, Israel, Middle East, South America	Asia-Pacific, Australia, Brazil, Canada, Chine, Europe, France, Germany, India, Japan, Korea, Norway, South Africa, Sweden, Switzerland, United Kingdom, United States, United Arab Emirates	Asia East, Asia Northeast, Asia South, Asia Southeast, Australia Southeast, Europe Central, Europe North, Europe Southwest, Europe West, Middle East Central, Middle East West, North America Northeast, South America East, South America West, US Central, US Central, US West
Доступність рівня безкоштовного користування	так	так	так

Характеристики	Amazon EC2	MS Azure Virtual Machines	Google Compute Engine
Обсяг ресурсів, доступний на рівні безкоштовного користування	750 годин на місяць інстансів типу t2.micro з ОС Linux та Windows протягом 1 року	750 годин на місяць віртуальних машин з Windows та Linux (типи B1s, B2pts v2, B2ats) протягом 1 року	300 доларів на використання сервісів протягом перших 90 днів; один інстансом типу e2-micro протягом 1 місяця

Додаток Б – **Огляд основних компонентів хмарної платформи OpenStack**

Платформа OpenStack складається з низки компонентів, кожний з яких має власну спеціалізацію та надає послуги через відповідний API [6], [26], [21].

На рис. Б.1 показано основні компоненти OpenStack, насправді їх значно більше. Повний перелік сервісів доступний на офіційному сайті OpenStack [26].

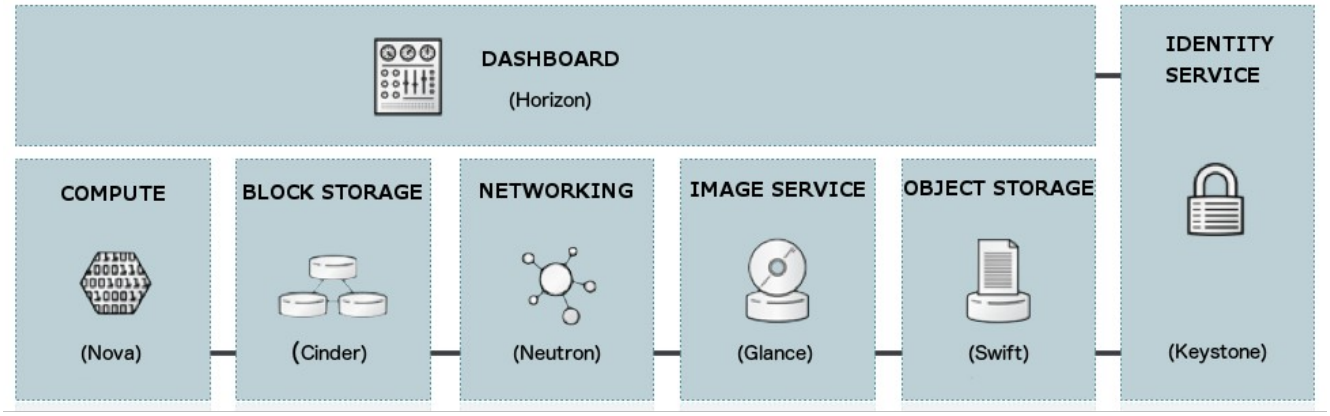


Рисунок Б.1 – Діаграма основних компонентів OpenStack [37]

Nova (сервіс обчислень, compute service). Керування інстансами на базі віртуальних машин, які працюють на базі гіпервізорів.

Swift (сервіс сховища об'єктів, object storage service). Підтримує зберігання та одержання обов'язкових даних хмари. Організація сховища об'єктів відрізняється від традиційної файлової системи.

Cinder (сервіс блокового сховища, block storage service). Керує розділами накопичувачів для віртуальних машин.

Horizon (панель керування, dashboard). Веб-інтерфейс для керування сервісами OpenStack.

Keystone (сервіс ідентичності, identity service). Централізований сервіс автентифікації та авторизації для інших сервісів OpenStack, для керування користувачами, проєктами, ролями.

Neutron (мережний сервіс, networking service). Забезпечує з'єднання між інтерфейсами сервісів OpenStack.

Glance (сервіс образів, image service). Зберігає такі ресурси, як образи віртуальних машин та зрізи (snapshots) томів.

Ceilometer (сервіс телеметрії, telemetry service). Забезпечує вимірювання обсягів використання хмарних ресурсів.

Heat (сервіс оркестрування, orchestration service). Забезпечує оркестрування ресурсів. Підтримує автоматичне створення стеків ресурсів.