Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Розгортання хмарної інфраструктури з OpenStack

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-5
Євген БАЧИНСЬКИЙ
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп'ютерна інженерія
(повна назва освітньої програми)

Керівник: ас. Дар'я ТИМОШЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Бачинському Євгену Олександровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Розгортання хмарної інфраструктури з OpenStack

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи Тема дослідження
Офіційна документація OpenStack, OpenStack-Ansible, Ansible та Linux Containers
Скрипти автоматизованого розгортання сервісів OpenStack
Серверне середовище на базі Ubuntu Server 22.04 LTS

4. Перелік питань, що потрібно опрацювати у роботі
Аналіз теоретичних основ хмарних обчислень та моделей розгортання інфраструктури
Огляд платформи OpenStack: компоненти, функціональність, архітектура
Запуск сценаріїв Ansible для поетапного розгортання інфраструктури
Тестування працездатності компонентів OpenStack та взаємодії між ними
Аналіз переваг та недоліків використання OpenStack-Ansible

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 13

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	27.05.25-30.05.25	
2	Вибір технологій та інструментів	31.05.25-01.06.25	
3	Розробка конфігурацій	02.06.25-06.06.25	
4	Тестування та відлагодження	07.06.25-09.06.25	
5	Оформлення матеріалів кваліфікаційної роботи	10.06.25-11.06.25	
6	Подання кваліфікаційної роботи керівникові та	12.06.25-13.06.25	
	її попередній захист		
7	Подання кваліфікаційної роботи на рецензу-	14.06.25-16.06.25	
	вання		

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Дар'я ТИМОШЕНКО.
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 55 с., 13рис., 0 табл., 1дод., 15 джерел.

ХМАРНІ ТЕХНОЛОГІЇ, OPENSTACK, ІНФРАСТРУКТУРА, АНСІБЛ, LINUX-КОНТЕЙНЕРИ, ВІРТУАЛІЗАЦІЯ, КОМПОНЕНТИ OPENSTACK, МЕРЕЖЕВА АРХІТЕКТУРА, СКРИПТИЗАЦІЯ, АВТОМАТИЗАЦІЯ.

Метою кваліфікаційної роботи є дослідження та практичне розгортання хмарної інфраструктури на базі OpenStack із використанням інструментів автоматизації, зокрема OpenStack-Ansible, а також оцінка ефективності такого підходу в умовах сучасної IT-інфраструктури.

У ході виконання кваліфікаційної роботи було проведено комплексний аналіз сучасних хмарних технологій, розглянуто принципи побудови відкритої платформи OpenStack, її ключові компоненти та можливості масштабування. Особлива увага приділена засобам автоматизації з використанням Ansible як рушія для конфігурації та керування розгортанням сервісів OpenStack. Практична частина роботи передбачає підготовку серверного середовища, налаштування інвентарних файлів, виконання необхідних скриптів для ініціалізації контейнеризованого середовища, а також повне розгортання основних сервісів OpenStack. Результатом роботи стало створення працездатного хмарного середовища з можливістю масштабування та централізованого управління. Отримані результати можуть бути використані для побудови власних приватних хмар у межах підприємств, дослідницьких установ або освітніх закладів.

ABSTRACT

Bachelor’s thesis: 55 pages, 13 figures, 0 tables, 0 appendices, 15 sources.

CLOUD TECHNOLOGIES, OPENSTACK, INFRASTRUCTURE AS CODE, ANSIBLE, LINUX CONTAINERS, VIRTUALIZATION, AUTOMATION, CONFIGURATION MANAGEMENT, IAAS, OPEN SOURCE.

The major goal of this thesis is to design, configure, and deploy a cloud infrastructure using OpenStack with the help of the OpenStack-Ansible automation toolkit. The project aims to demonstrate the practical implementation of cloud services based on open-source technologies while ensuring scalability, security, and manageability.

In order to achieve this, the thesis explores the architecture and components of the OpenStack platform, emphasizing its modular structure and compatibility with infrastructure-as-code principles. The practical part includes the preparation of the control server, installation of required software packages, cloning of the OpenStack-Ansible repository, inventory configuration, password generation, and the execution of Ansible playbooks to deploy cloud services. As a result, a fully functional private cloud environment was created with automated service management and flexible resource allocation, suitable for enterprise or academic use.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – програмний інтерфейс прикладного програмування (Application Programming Interface)

CI/CD – безперервна інтеграція та розгортання (Continuous Integration / Continuous Deployment)

CLI – інтерфейс командного рядка (Command Line Interface)

DBMS – система керування базами даних (Database Management System)

DHCP – протокол динамічного конфігурування хостів (Dynamic Host Configuration Protocol)

DNS – система доменних імен (Domain Name System)

GUI – графічний інтерфейс користувача (Graphical User Interface)

HDD/SSD – жорсткий/твердотільний диск (Hard Disk Drive / Solid State Drive)

LXC – контейнери на рівні операційної системи (Linux Containers)

NIC – мережевий інтерфейсний контролер (Network Interface Card)

RAM – оперативна пам’ять (Random Access Memory)

VM – віртуальна машина (Virtual Machine)

WDS – бездротова розподільча система (Wireless Distribution System)

YAML – формат розмітки конфігураційних файлів (YAML Ain’t Markup Language)

ВСТУП

У сучасну цифрову епоху, що характеризується стрімким технологічним розвитком і постійним зростанням обсягу інформації, хмарні обчислювальні технології набули статусу ключового чинника трансформації IT-інфраструктур. Вони забезпечують можливість швидкого й зручного доступу до обчислювальних потужностей, прикладного програмного забезпечення, сховищ даних і мережевих сервісів у режимі реального часу за допомогою мережі Інтернет. Такий підхід дозволяє не лише оптимізувати ресурси організації, а й повністю переглянути модель надання IT-послуг, переходячи від класичної інфраструктури з високими капітальними витратами до гнучкої, орієнтованої на споживання моделі, де витрати розподіляються за фактом використання.

Упровадження хмарних технологій суттєво змінює правила гри в бізнесі, науці, охороні здоров’я, освіті та навіть державному управлінні. Вони дозволяють організаціям зменшити залежність від фізичних потужностей, підвищити швидкість розгортання нових сервісів, а також мінімізувати витрати на технічне обслуговування. У практиці цифрової трансформації все частіше спостерігається тенденція до переходу на гібридні та мультихмарні моделі, що дає змогу поєднувати переваги різних хмарних провайдерів, уникати прив’язки до одного постачальника та покращувати стійкість до відмов.

Однією з провідних технологій для побудови відкритої хмарної інфраструктури є OpenStack — масштабований, модульний і повністю відкритий проєкт, розроблений як спільно технологічними компаніями, так і незалежними розробниками з усього світу. Цей програмний комплекс дозволяє створювати власні приватні або публічні хмари, які функціонують за тими ж принципами, що й сервіси найбільших світових хмарних гігантів. OpenStack надає інтегрований набір сервісів для керування віртуальними

машинами, мережами, блоковими та об’єктними сховищами, автентифікацією користувачів, а також автоматизованим оркеструванням ресурсів. Основна сила платформи полягає в її модульності: компоненти можна розгортати незалежно, адаптуючи систему до специфічних потреб.

У порівнянні з комерційними аналогами, такими як VMware або Microsoft Azure, OpenStack вирізняється відкритим вихідним кодом, відсутністю ліцензійної плати та широкими можливостями кастомізації. Це робить його надзвичайно привабливим для навчальних закладів, науково-дослідних інститутів, стартапів і державних структур. Більш того, завдяки активній міжнародній спільноті, платформа постійно оновлюється, удосконалюється і розвивається у відповідь на сучасні виклики в галузі обчислень, зберігання даних і кібербезпеки.

Інноваційність OpenStack полягає також у можливості інтеграції з іншими інструментами DevOps — такими як Ansible, Terraform, Helm — що відкриває шлях до побудови повністю автоматизованих CI/CD-процесів. Розгортання інфраструктури стає відтворюваним, контрольованим і масштабованим, що відповідає принципам інфраструктури як коду (Infrastructure as Code). Крім того, OpenStack підтримує взаємодію з різними типами гіпервізорів, включаючи KVM, Xen та Hyper-V, а також забезпечує багатокомпонентну автентифікацію та багаторівневий контроль доступу, що є критично важливим у сучасному світі інформаційної безпеки.

Варто також зазначити, що хмарні технології, зокрема ті, що побудовані на основі OpenStack, поступово стають платформою для реалізації більш складних концепцій, таких як edge computing, fog computing і хмарне машинне навчання. Завдяки гнучкості та відкритості, OpenStack використовується не лише для побудови приватних дата-центрів, а й у великих телекомунікаційних компаніях, які інтегрують його у свої мережеві ядра, забезпечуючи послуги на базі NFV (Network Function Virtualization).

Таким чином, хмарні технології на сьогодні — це не просто спосіб оптимізації IT-витрат, а стратегічний інструмент забезпечення

конкурентоспроможності, безпеки, інноваційності та цифрової стійкості бізнесу. OpenStack, у свою чергу, виступає не лише як платформа, а як екосистема, що відкриває нові горизонти в управлінні ресурсами, автоматизації та цифровій трансформації організацій будь-якого масштабу. Його роль у сучасній IT-парадигмі продовжує зростати, і з кожною новою ітерацією він набуває дедалі більшої актуальності для наукової, освітньої й індустріальної спільнот.

1 ХМАРНІ ТЕХНОЛОГІЇ

Хмарні технології дозволяють отримувати комп'ютерні ресурси для обробки та зберігання даних у вигляді послуг. Іншими словами: користувачі не повинні купувати дороге обладнання, а отримують кінцевий продукт у вигляді програми, яку можна використовувати на своєму пристрої або звичайному комп'ютері. Доступ до цього програмного забезпечення здійснюється через Інтернет.

Хмарні технології включають кілька обов'язкових елементів. До них, наприклад, належить доступність центрів обробки даних, в яких знаходяться фізичні сервери. До цього додається віртуалізація, яка дозволяє розподіляти обчислювальну потужність центрів обробки даних для створення дискового простору або віртуальних машин.

Хмарні технології з'явилися в повсякденному житті підприємств і приватних осіб відносно недавно, але без них багато корисних і практичних інструментів були б немислимими. Необмежене зберігання даних, недорогі та легкодоступні додатки, онлайн-ігри – все це стало можливим завдяки хмарним технологіям. Завдяки їм підприємства значно зменшили проблеми, пов'язані зі створенням необхідної інфраструктури та зберіганням інформації.

Хмарні технології забезпечують доступ до комп'ютерних ресурсів для обробки та зберігання даних у вигляді послуг. Це означає, що користувачам не потрібно купувати дороге комп'ютерне обладнання: кінцевий продукт поставляється у вигляді програми для використання на стандартному пристрої або комп'ютері. Доступ до цієї послуги здійснюється через Інтернет.

Хмарні технології мають кілька обов'язкових елементів. Наприклад, повинні існувати центри обробки даних з фізичними серверами. Крім того, існує віртуалізація, яка дозволяє використовувати обчислювальну потужність центрів обробки даних для зберігання або створення віртуальних машин.

Технології хмарних обчислень тільки недавно увійшли в повсякденне життя підприємств і приватних осіб, але без них вже неможливо уявити багато корисних і практичних інструментів.

Необмежене зберігання даних, прості та практичні додатки, онлайн-ігри – все це стало можливим завдяки технологіям хмарних обчислень. Завдяки застосуванню цих технологій підприємства значно зменшили проблеми, пов'язані із забезпеченням необхідної інфраструктури та зберіганням інформації. Наразі хмарні технології можна розділити на різні типи залежно від способу надання комп'ютерних ресурсів користувачеві.

1.1 Публічні хмари

Публічні Хмари — це модель обробки даних у хмарі, в якій всі користувачі мають доступ до ресурсів через Інтернет. Ці ресурси можуть включати програмне забезпечення, пам'ять даних або сервери. Послугами хмари може користуватися кожен, зберігаючи конфіденційність своїх персональних даних. Жоден користувач не може отримати несанкціонований доступ до дискового простору інших клієнтів. Це може статися тільки в разі необережності користувача.

Перед використанням публічної хмари слід пам'ятати, що параметри, визначені постачальником послуг, можуть бути змінені виключно за його згодою.

Послуги публічної хмари мають кілька важливих переваг, які роблять їх дуже популярними в сучасному IT-середовищі. Однією з головних особливостей є можливість використання комп'ютерних ресурсів через Інтернет, що забезпечує гнучке використання інфраструктури незалежно від географічного розташування користувача. Завдяки великій масштабованості ресурси можна швидко адаптувати до поточних потреб, що має особливе значення в разі змінного навантаження або динамічного розвитку проекту.

Публічна хмара доступна для широкого кола користувачів, незалежно

від їх технічного рівня або організації, до якої вони належать, що розширює можливості цифрової трансформації також для малих підприємств або індивідуальних програмістів. Крім того, модель оплати за використання дозволяє оптимізувати фінансові витрати та точно спланувати бюджет, уникаючи надмірних витрат, пов'язаних з невикористаною інфраструктурою. Всі ці переваги роблять рішення публічної хмари привабливими з точки зору продуктивності, економічності та доступності.і.

1.2 Приватні хмари

Приватна хмара — це модель хмарних обчислень, в якій всі ресурси зарезервовані виключно для однієї організації. Це означає, що ніхто інший не має доступу до приватного середовища. Воно контролюється та управляється виключно його власником. Ємність приватної хмари може знаходитися в локальному середовищі клієнта або бути орендованою в центрі обробки даних.

Слід зазначити, що така архітектура ідеально підходить для зберігання конфіденційних даних, які не можуть зберігатися на публічних серверах. Однак для створення робочої інфраструктури компанія повинна інвестувати багато часу та коштів.

Приватні хмарні середовища надають компаніям розширені функції управління IT-ресурсами завдяки повному контролю над інфраструктурою, навіть якщо фізичне обладнання знаходиться в орендованих центрах обробки даних. Ця модель забезпечує виняткову автономність в управлінні та конфігурації ресурсів відповідно до внутрішніх директив та технічних вимог компанії. Високий рівень безпеки досягається завдяки централізованому контролю даних, мережевої архітектури та доступу до послуг, що є надзвичайно важливим для компаній, які обробляють конфіденційну або стратегічну інформацію.

Крім того, приватна хмара пропонує гнучку багаторівневу систему

доступу, яка дозволяє чітко визначити повноваження співробітників відповідно до їхніх ролей та обов'язків. Це значно зменшує ризик несанкціонованого доступу та підвищує загальний рівень безпеки інформації. Таким чином, приватна хмара є надійним рішенням для підприємств, які бажають зберегти повний контроль над своїми цифровими ресурсами, гарантуючи стабільність, конфіденційність та відповідність нормативним вимогам.Приватна хмара — це модель хмарних обчислень, в якій всі ресурси зарезервовані виключно для однієї організації. Це означає, що ніхто інший не має доступу до приватного середовища. Воно контролюється та управляється виключно власником. Ємність приватної хмари може знаходитися в локальному середовищі клієнта або бути орендованою в центрі обробки даних.

Слід зазначити, що така архітектура ідеально підходить для зберігання конфіденційних даних, які не можуть зберігатися на публічних серверах. Однак для створення робочої інфраструктури компанія повинна інвестувати багато часу та коштів.

Приватні хмарні середовища надають компаніям розширені функції управління інформаційними ресурсами завдяки повному контролю над інфраструктурою, навіть якщо фізичне обладнання знаходиться в орендованих центрах обробки даних. Ця модель забезпечує виняткову автономність в управлінні та конфігурації ресурсів відповідно до внутрішніх директив та технічних вимог підприємства. Високий рівень безпеки досягається завдяки централізованому контролю даних, мережевої архітектури та доступу до послуг, що є надзвичайно важливим для підприємств, які обробляють конфіденційну або стратегічну інформацію.

Крім того, приватна хмара пропонує гнучку багаторівневу систему доступу, яка дозволяє чітко визначити повноваження співробітників відповідно до їхніх ролей та обов'язків. Це значно зменшує ризик несанкціонованого доступу та підвищує загальний рівень безпеки інформації. Таким чином, приватна хмара є надійним рішенням для підприємств, які

бажають зберегти повний контроль над своїми цифровими ресурсами, гарантуючи стабільність, конфіденційність та відповідність нормативним вимогам.

1.3 Гібридні хмари

Гібридна хмара - це багатофункціональна модель, що поєднує приватні та публічні хмари. Це гнучка модель, яка дозволяє зберігати конфіденційну інформацію в приватній зоні, а іншу - в публічній. Під час інсталяції налаштовуються спеціальні шлюзи для передачі даних з приватного в публічне середовище і навпак

Гібридна хмара поєднує в собі гнучкість публічного хмарного середовища з контрольованістю та безпекою приватної інфраструктури, що дозволяє підприємствам адаптувати ресурси до змін у робочих навантаженнях. Завдяки цьому компанії мають змогу оперативно масштабувати власну інфраструктуру шляхом інтеграції додаткових обчислювальних потужностей із публічної хмари, не втрачаючи при цьому контроль над ключовими компонентами. Такий підхід забезпечує ефективне управління ресурсами в умовах зростаючої динаміки бізнесу та змін у вимогах до продуктивності.

Інтеграція гібридної моделі також дає змогу органічно поєднувати вже наявну локальну інфраструктуру з ресурсами публічної хмари, що дозволяє зберігати раніше налагоджені технічні процеси та водночас розширювати можливості за рахунок зовнішніх сервісів. Це забезпечує гнучкість у розміщенні робочих навантажень і сприяє ефективному використанню обчислювальних ресурсів відповідно до конкретних завдань.

Окремо варто відзначити можливість вільного переміщення додатків і даних між приватним і публічним середовищем, що дозволяє адаптувати інфраструктуру до оперативних потреб користувачів та бізнес-процесів. Така мобільність є особливо актуальною в умовах цифрової трансформації, коли

швидкість реакції на зміни є критичним фактором успішності.

Гібридна хмара також гарантує підвищений рівень безпеки завдяки можливості розміщення конфіденційної інформації виключно у приватному середовищі, тоді як менш чутливі компоненти можуть ефективно оброблятися у публічному просторі. Це доповнюється механізмами резервного копіювання, які дозволяють розміщувати дублікати даних на різних фізичних майданчиках, що значно знижує ризики втрати інформації в разі аварійних ситуацій.

1.4 Мультихмара

Мультихмара — це гнучка інфраструктура, яка використовує послуги декількох постачальників хмарних послуг. Таким чином, клієнт отримує мультихмару, яка ідеально відповідає потребам його підприємства і здатна виконувати безліч специфічних завдань. Під час налаштування слід пам'ятати, що ця модель вимагає більше адміністративної роботи для досягнення необхідного рівня безпеки та продуктивності. Гіперконвергентна інфраструктура поєднує віртуальну обчислювальну потужність, програмно-визначену пам'ять і віртуальну мережу на одному сервері. Ці вузли можна швидко адаптувати до потреб.

Зростаючий попит на хмарні послуги, а також необхідність заміни застарілого обладнання та зниження експлуатаційних витрат спонукають підприємства впроваджувати гіперконвергентні інфраструктури в своїх центрах обробки даних. Гіперконвергентна інфраструктура дозволяє виконувати обчислювальні та сховища на одному сервері. Сервери об'єднуються у віртуальний кластер обчислювальних і сховищ ресурсів через стандартну мережу Ethernet.

Мультихмарна стратегія полягає в одночасному використанні послуг декількох хмарних постачальників, що значно підвищує надійність і відмовостійкість системи зберігання та обробки даних. Розподіл

навантаження між різними платформами захищає від потенційних збоїв одного постачальника та гарантує безперебійний доступ до критично важливих IT-послуг та ресурсів. Такий підхід є ефективним засобом забезпечення надійності інформації та мінімізації ризику втрати даних.

Крім того, мультихмара відкриває можливості для фінансової оптимізації, оскільки підприємства можуть вибирати постачальників, що пропонують найвигідніші умови для конкретних завдань або послуг. Це створює умови для гнучкого управління витратами, уникнення залежності від одного постачальника та зниження загальної вартості хмарної інфраструктури.

Мультихмарна модель також сприяє підвищенню технологічної адаптивності підприємства. Можливість залучення різних постачальників, що спеціалізуються на останніх інноваціях, дозволяє компаніям швидко впроваджувати сучасні технології у власні робочі процеси. Це забезпечує високу конкурентоспроможність на ринку та дозволяє швидше реагувати на зміни в технологічному середовищі.

В цілому, мультихмара створює більш стійку, масштабовану та гнучку IT-інфраструктуру, здатну задовольнити багато технічних та економічних вимог.

1.5 Еволюція хмарних технологій

Історія хмарних технологій починається ще з ідей розподілених обчислень у 1960-х роках, коли Джон Маккарті припустив, що обчислення можуть колись надаватися як комунальні послуги. У 1990-х роках із появою віртуалізації компанії на кшталт VMware почали розвивати програмні рішення, що дозволяли ізольовано запускати кілька операційних систем на одному фізичному сервері. Це стало фундаментом для сучасних хмарних платформ.

Справжній прорив відбувся на початку 2000-х, коли Amazon Web

Services (AWS) запропонував першу комерційну IaaS-платформу з такими сервісами, як Amazon EC2 (Elastic Compute Cloud) та S3 (Simple Storage Service). Успіх AWS зумовив появу конкурентів: Microsoft Azure, Google Cloud Platform, а також відкритих платформ, зокрема OpenStack.

Згодом концепція хмари еволюціонувала від IaaS до PaaS (Platform as a Service) та SaaS (Software as a Service). Такі сервіси, як Heroku чи Google App Engine, надали розробникам змогу зосередитися на коді, а не на інфраструктурі. Ще пізніше з’явились поняття FaaS (Function as a Service) і serverless-обчислення, які ще більше абстрагують користувача від інфраструктури.

Зараз ми спостерігаємо перехід до мультихмарного та гібридного підходу, де підприємства поєднують кілька хмарних провайдерів і локальні ресурси, аби досягти більшої гнучкості, відповідності до нормативних вимог та зниження вартості.

2 OPENSTACK ТА ЙОГО КОМПОНЕНТИ

У цьому розділі розглядаються основні компоненти хмарної платформи OpenStack з відкритим вихідним кодом. Аналізуються функціональні можливості, архітектура та взаємодія окремих сервісів, які формують основу для побудови масштабованої, гнучкої та керованої хмарної інфраструктури.

2.1 OpenStack

OpenStack — це набір модулів програмного забезпечення типу open source, що служать для створення послуг інфраструктури хмарних обчислень та систем зберігання даних у хмарі. Система характеризується високою гнучкістю, масштабованістю та економічністю, завдяки чому підходить для віртуалізації центрів обробки даних на основі відкритих технологій. OpenStack можна ефективно інтегрувати в гетерогенні середовища, а також у корпоративні або інші рішення з відкритим кодом, що розширює сферу його застосування.

Завдяки сумісності API OpenStack з інтерфейсами таких постачальників, як Amazon EC2 і S3, клієнтські додатки можуть бути перенесені в середовище OpenStack без значних змін, що має особливе значення для завдань, пов'язаних з обробкою великих обсягів даних і розвитком Інтернету речей (IoT). Платформа відносно проста у впровадженні, а її архітектура може бути адаптована до вимог приватних хмар різного розміру.

OpenStack відіграє важливу роль у модернізації центрів обробки даних, оскільки дозволяє автоматизувати управління фізичними ресурсами та віртуальними послугами. Тому все більше компаній та постачальників послуг інтегрують OpenStack у свою інфраструктуру, щоб підвищити ефективність та адаптуватися до сучасних викликів у сфері інформаційних

технологій.

OpenStack постійно розвивається, і кожне оновлення додає нові функції. Відкритий характер і активна підтримка спільноти сприяють зростанню популярності платформи серед розробників і компаній, зацікавлених у гнучких і масштабованих хмарних рішеннях.

«OpenStack» включає шість основних послуг, які керують обчислювальною потужністю, мережами, зберіганням даних, ідентифікацією користувачів та образами системи.

Ці основні послуги становлять інфраструктуру, на якій базується багато додаткових модулів, що розширюють функціональність хмарної платформи. Ці додатки включають координацію, автоматизацію процесів, моніторинг, оповіщення, резервування ресурсів та послуги розрахунків – загалом понад 30. Завдяки гнучкій архітектурі OpenStack дозволяє інтегрувати різні компоненти, які взаємодіють через відкриті API-інтерфейси та забезпечують ефективний доступ до ресурсів інфраструктури.

Під час впровадження OpenStack кожна послуга відповідає за певну частину інфраструктурного середовища. Наприклад, «Nova» виконує роль базової IT-послуги, тоді як «Zun» спеціалізується на створенні контейнерів. «Ironic» використовується для управління фізичними серверами, а «Cyborg» забезпечує підтримку спеціалізованого обладнання. У сфері зберігання даних OpenStack пропонує різні методи: Swift для зберігання об'єктів, Cinder для блокового доступу та Manila для роботи з розподіленими файловими системами.

Мережева комунікація в хмарі здійснюється за допомогою служби Neutron, яка забезпечує гнучке управління віртуальними мережами. Octavia відповідає за балансування навантаження, а Designate бере на себе функції системи доменних імен. Серед загальних послуг важливу роль відіграють Keystone (аутентифікація та авторизація), Placement (розподіл ресурсів) та Glance (управління віртуальними машинами). Додаткові функції забезпечують Barbican (управління ключами), Heat (координація

інфраструктури), Senlin (кластеризація), Mistral, Zaqar, Blazar та інші служби.

Інфраструктура OpenStack також включає послуги створення, виконання та обслуговування навантажень. Magnum призначений для координації контейнерів, Sahara — для обробки великих обсягів даних, а Trove забезпечує інтерфейс для служби управління базами даних. Masakari забезпечує високу доступність, а Murano та Solum спрощують впровадження, обслуговування та автоматизацію протягом усього життєвого циклу додатків. Freezer відповідає за створення резервних копій та відновлення даних у надзвичайних ситуаціях. Управління ресурсами OpenStack візуалізується за допомогою веб-інтерфейсів Horizon та Skyline, які спрощують використання та управління хмарним середовищем.

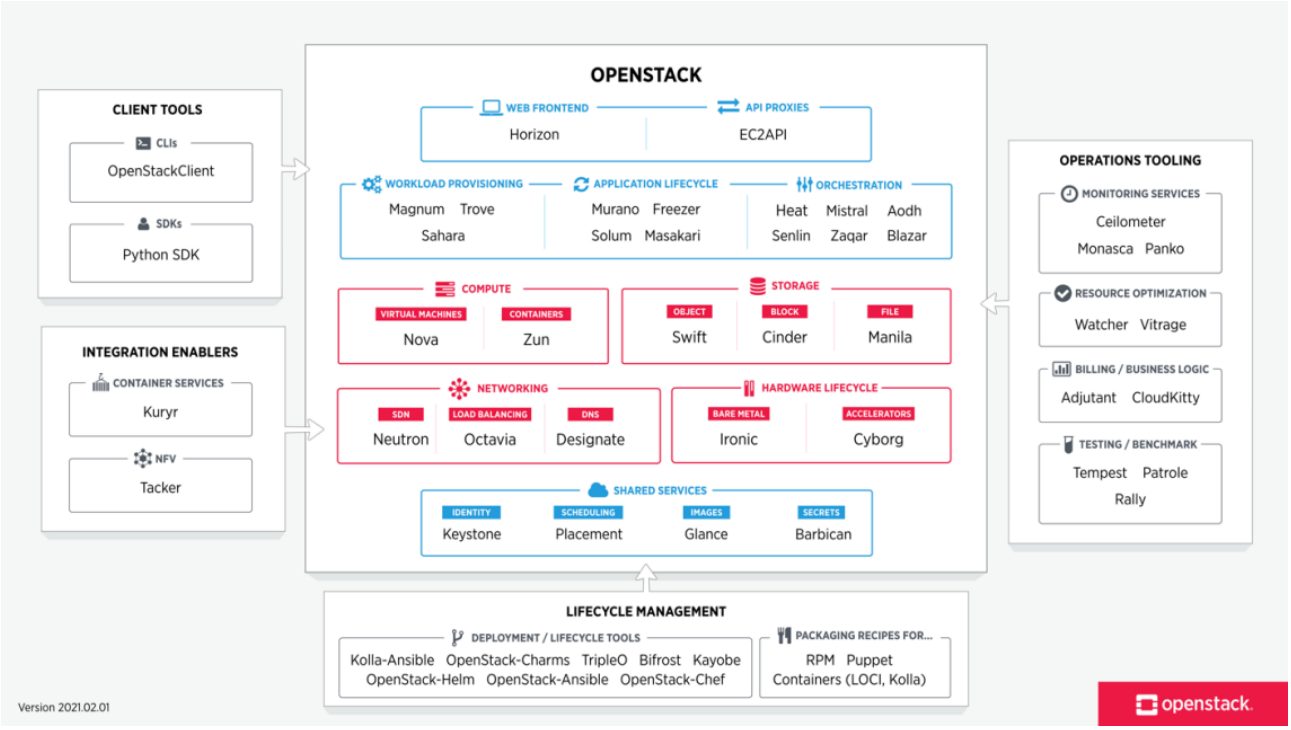


Рисунок 2.1 - Компоненти OpenStack

2.2 Keystone

OpenStack Keystone — це ключовий компонент хмарної платформи OpenStack, який виконує функції служби ідентифікації. Він забезпечує

автентифікацію користувачів, авторизацію доступу до ресурсів та управління каталогом сервісів. Keystone є першим сервісом, який обробляє запити в OpenStack, перевіряючи облікові дані користувачів і видаючи токени для подальшої взаємодії з іншими компонентами системи [11].

Служба підтримує різні методи автентифікації, включаючи токени, пару ім'я користувача та пароль, а також інтеграцію з зовнішніми системами, такими як LDAP. Це дозволяє гнучко налаштовувати політики доступу та забезпечує централізоване управління ідентифікацією в багатокористувацькому середовищі.

Keystone також відповідає за надання каталогу сервісів, який містить інформацію про доступні сервіси OpenStack та їхні API-ендпоїнти. Це дозволяє клієнтам динамічно виявляти та взаємодіяти з необхідними сервісами в хмарному середовищі.

Завдяки своїй архітектурі та підтримці сучасних протоколів автентифікації, таких як OAuth, SAML та OpenID, Keystone забезпечує високий рівень безпеки та масштабованості, що робить його невід'ємною частиною будь-якого розгортання OpenStack.

2.3 Nova

OpenStack Nova — це основний компонент хмарної платформи OpenStack, який відповідає за створення, управління та масштабування обчислювальних ресурсів, зокрема віртуальних машин (VM), фізичних серверів (bare metal) та контейнерів. Nova реалізує модель інфраструктури як сервісу (IaaS), надаючи інтерфейси для автоматизованого розгортання та адміністрування обчислювальних середовищ [12].

Архітектура Nova побудована на модульному принципі та включає низку сервісів, кожен з яких виконує специфічні функції. Сервіс nova-api обробляє зовнішні API-запити користувачів (рисунок 2.3), ініціюючи процеси створення та управління інстансами. nova-compute взаємодіє з гіпервізорами,

такими як KVM, Xen або VMware, для запуску та завершення віртуальних машин. nova-scheduler визначає оптимальний хост для розміщення нового інстансу, враховуючи доступні ресурси. Модуль nova-conductor забезпечує безпечну взаємодію між nova-compute та базою даних, мінімізуючи прямий доступ до неї. Додаткові сервіси, такі як nova-novncproxy та nova-spicehtml5proxy, надають можливість віддаленого доступу до інстансів через VNC або SPICE протоколи .

Nova тісно інтегрується з іншими компонентами OpenStack для забезпечення повноцінного функціонування хмарної інфраструктури. Зокрема, Keystone відповідає за автентифікацію та авторизацію користувачів, Glance надає доступ до образів віртуальних машин, Neutron забезпечує мережеву взаємодію, а Placement відстежує та розподіляє ресурси між інстансами .

Завдяки своїй гнучкості та масштабованості, Nova дозволяє розгортати обчислювальні ресурси як у віртуалізованому середовищі, так і на фізичних серверах, що робить її універсальним інструментом для побудови приватних, публічних та гібридних хмарних рішень. Підтримка різноманітних гіпервізорів та можливість інтеграції з іншими сервісами

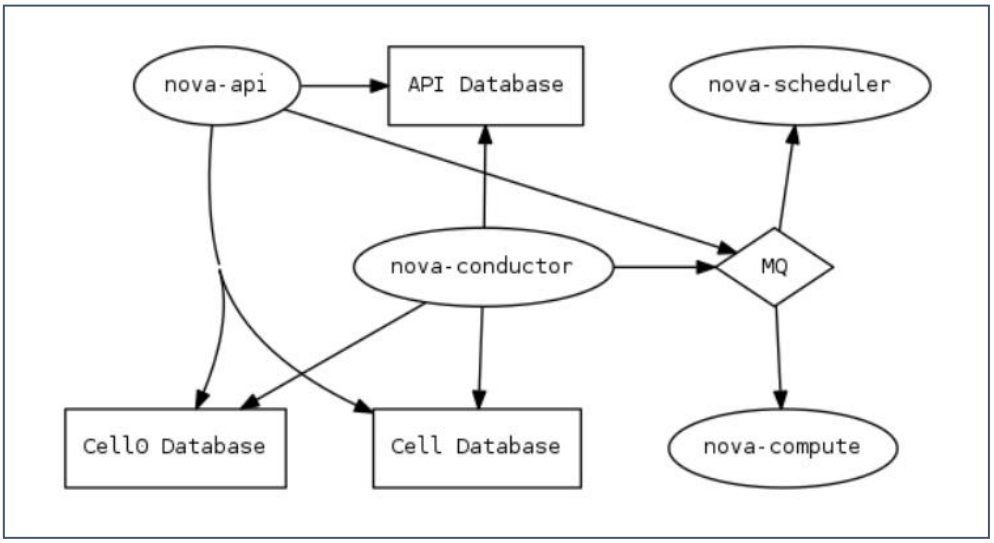


Рисунок 2.3 – Сервіси Nova

2.4 Neutron

OpenStack Neutron — це ключовий компонент хмарної платформи OpenStack, який забезпечує функціональність «мережа як сервіс» (NaaS). Він дозволяє користувачам створювати, налаштовувати та керувати віртуальними мережевими структурами, включаючи мережі, підмережі, порти, маршрутизатори та інші мережеві ресурси. Neutron надає API, що дозволяє визначати мережеву топологію та адресацію в хмарному середовищі, забезпечуючи гнучкість та масштабованість мережевої інфраструктури [13] .

Архітектура Neutron побудована на модульному принципі та включає такі основні компоненти:

neutron-server обробляє API-запити та передає їх відповідним плагінам для виконання.

Плагіни та агенти відповідають за реалізацію мережевих функцій, таких як створення мереж, підмереж, портів, а також управління IP-адресацією. Плагіни можуть бути адаптовані до різного мережевого обладнання та програмного забезпечення, що забезпечує гнучкість у розгортанні OpenStack.

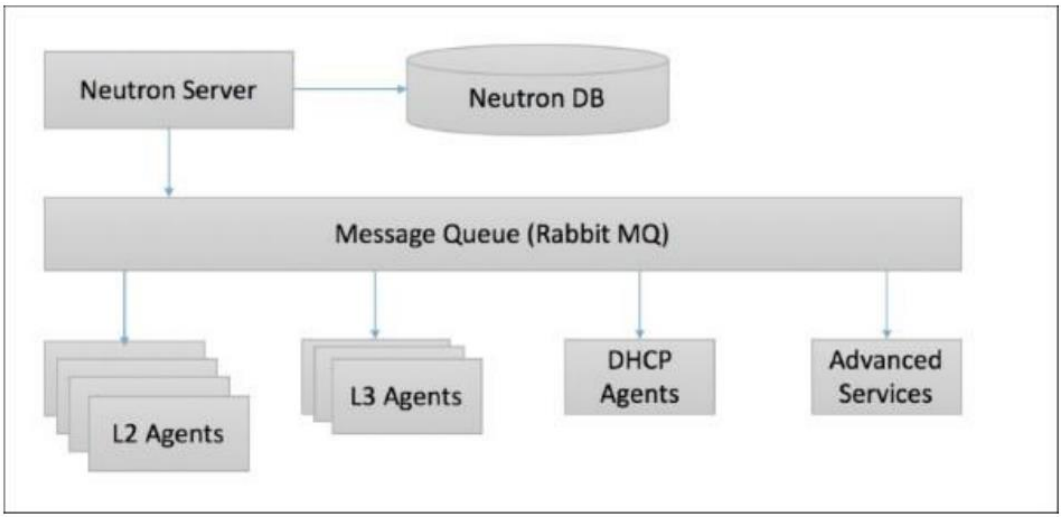


Рисунок 2.4 – Сервіси Neutron

Neutron підтримує різноманітні мережеві технології, включаючи Open vSwitch, Linux Bridge, а також інтеграцію з SDN-контролерами. Це дозволяє створювати складні мережеві топології, включаючи багаторівневі маршрутизатори, брандмауери, VPN та балансувальники навантаження. Крім того, Neutron забезпечує підтримку як статичної, так і динамічної IP-адресації, включаючи можливість використання плаваючих IP-адрес для забезпечення доступу до інстансів ззовні хмарного середовища.

Завдяки своїй гнучкості та розширюваності, Neutron є невід'ємною частиною OpenStack, що забезпечує ефективне управління мережевими ресурсами в хмарних інфраструктурах різного масштабу та складності. Його модульна архітектура та підтримка широкого спектру мережевих технологій роблять його потужним інструментом для побудови сучасних хмарних рішень.

2.5 Cinder

OpenStack Cinder — це модуль системи OpenStack, що відповідає за надання блочного зберігання як послуги (Block Storage as a Service). Його основна функція — управління життєвим циклом блочних пристроїв, які можуть бути підключені до віртуальних машин (інстансів), створених у середовищі OpenStack через компонент Nova. На відміну від об'єктного або файлового зберігання, блочне зберігання забезпечує високопродуктивний доступ до даних, подібний до фізичних жорстких дисків, що робить його особливо корисним для баз даних, транзакційних систем і критично важливих додатків [14] .

Cinder дозволяє створювати, видаляти, підключати та відключати блочні томи, а також здійснювати знімки (snapshots) томів, створювати резервні копії і клонувати томи. Він підтримує широкий спектр бекендів — як апаратних (наприклад, NetApp, Dell EMC, HPE), так і програмних (Ceph, LVM, NFS).

2.6 Glance

OpenStack Glance-це служба образів, яка забезпечує швидкий і зручний спосіб копіювання і запуску екземплярів. За допомогою Glance користувачі можуть завантажувати, виявляти, реєструвати і витягувати образи віртуальних машин з швидкістю і легкістю. Glance забезпечує безпечне завантаження і безпечне вивантаження завдяки перевірці зображень з підписом. Це означає, що дані можуть бути перевірені через Glance перед збереженням у вашій хмарі. Тому, якщо перевірка не пройшла, завантаження буде невдалою, а зображення буде видалено. Те ж саме стосується всіх завантажень образів. Якщо дані не можуть виконати відповідну перевірку при завантаженні зображення, то воно не буде зберігатися у вашому хмарі. Що стосується сумісності, Glance не обмежується конкретними серверами, оскільки він може завантажувати віртуальні машини поряд з Cinder і Ironic. Завдяки RESTful API Glance можливий запит метаданих образу віртуальної машини, а також отримання фактичного образу. Нарешті, однією з переваг передових технологій OpenStack є проста інтеграція Glance з блоковим сховищем Cinder в рамках інфраструктури. Це дозволяє забезпечити зберігання і просту у використанні віртуалізацію управління блоковим сховищем.

Компонентна архітектура Cinder включає:

- cinder-api — приймає REST-запити та передає їх для обробки;
- cinder-scheduler — визначає, на якому вузлі створювати новий том;
- cinder-volume — основний процес, який взаємодіє з обраною системою зберігання;
- cinder-backup — опціональний компонент для створення резервних копій томів у зовнішніх сховищах.

Інтеграція з Nova дає змогу динамічно підключати томи до віртуальних машин, що забезпечує гнучкість у розподілі ресурсів. Cinder також підтримує механізми політик доступу, шифрування даних і контроль квот, що робить

його важливою частиною безпечної та масштабованої хмарної інфраструктури. Завдяки цьому Cinder є ключовим елементом побудови продуктивних хмарних рішень з можливістю тонкого управління сховищем.

2.7 Horizon

Horizon — це офіційний веб-інтерфейс (Dashboard) для керування сервісами OpenStack, який надає користувачам та адміністраторам зручну візуальну платформу для взаємодії з хмарною інфраструктурою. Horizon дозволяє виконувати більшість базових операцій, що доступні через API OpenStack, без потреби в глибоких технічних знаннях або взаємодії з командним рядком[15].

Цей інтерфейс реалізований як модуль Django-додатка на Python та інтегрується з основними службами OpenStack: Nova (обчислення), Cinder (блочне зберігання), Neutron (мережі), Glance (образи систем), Keystone (ідентифікація) тощо. Через Horizon користувачі можуть створювати і запускати віртуальні машини, налаштовувати мережі, керувати томами зберігання, образами, обліковими записами, ролями, безпековими групами та іншими ресурсами.

Horizon має модульну архітектуру, яка дозволяє додавати нові панелі для підтримки додаткових сервісів або розширення функціональності. Це робить його гнучким інструментом як для публічних, так і для приватних хмарних рішень.

Horizon є ключовим інструментом для візуального адміністрування хмарної інфраструктури OpenStack, спрощуючи доступ до її функціональних можливостей та підвищуючи ефективність роботи з хмарними сервісами.

2.8 Порівняння OpenStack з іншими платформами віртуалізації та хмарної інфраструктури

У сучасному IT-середовищі організації мають змогу обирати з-поміж

широкого спектра платформ для розгортання віртуалізованих середовищ і хмарної інфраструктури. Вибір між такими рішеннями, як OpenStack, VMware vSphere та Proxmox VE, залежить від цілей, бюджету, масштабу і рівня технічної підготовки команди. Кожна з платформ має свої особливості, підходи до побудови інфраструктури, модель розширюваності та керування.

OpenStack, як вже згадувалося, є відкритим комплексом для створення приватних і публічних хмар. Його ключова особливість полягає у повній модульності та масштабованості, що дозволяє вибудувати інфраструктуру будь-якої складності — від невеликих приватних кластерів до гіпермасштабованих хмар. OpenStack підтримує повноцінну реалізацію моделі IaaS і взаємодіє з широким набором компонентів: від керування віртуальними машинами до надання мережевих послуг, зберігання даних, аутентифікації користувачів і моніторингу.

На відміну від OpenStack, який орієнтований на інженерів із глибоким розумінням Linux-систем і принципів DevOps, платформа VMware vSphere є комерційним продуктом, що пропонує високий рівень зручності, стабільності та технічної підтримки. Рішення VMware розроблене насамперед для корпоративного ринку й надає потужний набір інструментів для керування ресурсами через централізовану консоль. Інтерфейс користувача максимально орієнтований на візуальне адміністрування, що спрощує роботу для системних адміністраторів без глибокої підготовки у сфері автоматизації. Проте, попри зручність і стабільність, VMware обмежена у своїй гнучкості та потребує значних фінансових вкладень, оскільки вся екосистема базується на ліцензіях.

Proxmox VE — ще одне популярне рішення для побудови інфраструктури віртуалізації. На відміну від OpenStack, яке працює з контейнерами LXC або віртуальними машинами KVM у відокремлених компонентах, Proxmox поєднує обидва підходи в єдиному інтерфейсі. Це дозволяє швидко розгортати як легкі контейнери, так і повноцінні віртуальні машини. Однією з переваг Proxmox є простота впровадження: базову

інфраструктуру можна підняти буквально за кілька годин. Веб-інтерфейс платформи інтуїтивно зрозумілий, що робить її привабливою для малих і середніх підприємств. Утім, Proxmox має обмеження щодо масштабування і не надає такого рівня абстракції, як OpenStack, що робить її менш придатною для побудови багатокористувацьких або мультиорендованих хмар.

Щодо розширюваності та кастомізації, OpenStack не має рівних серед згаданих рішень. Його архітектура дозволяє підключати сторонні модулі, інтегрувати з CI/CD-пайплайнами, використовувати Helm-чарти або Terraform для автоматизації, а також будувати hybrid- або multicloud-рішення із сумісністю з Kubernetes, Ceph, Prometheus тощо. Хоча цей підхід потребує значних знань і часу на впровадження, він виправданий у великих інфраструктурах, які вимагають гнучкості, контролю над ресурсами, відкритих API та відсутності vendor lock-in.

З іншого боку, VMware позиціонує себе як рішення «все-в-одному», де інструменти, як-от vCenter, vSAN, NSX, забезпечують інтеграцію всіх аспектів віртуалізації в єдину екосистему. Але ця замкнена архітектура означає, що міграція з VMware на іншу платформу є складною, що стримує розвиток і експерименти з інноваційними технологіями. Proxmox, у свою чергу, частково відкритий і підтримує API для автоматизації, але без такої глибини можливостей, яку надає OpenStack.

Таким чином, OpenStack — це платформа для тих, хто готовий інвестувати час у розгортання та підтримку, прагнучи отримати гнучкість і контроль. VMware — ідеальний вибір для підприємств, що шукають надійність і підтримку «з коробки». А Proxmox VE — золота середина для малого бізнесу, освітніх закладів або тестових середовищ, які потребують простоти, але водночас хочуть зберегти базову гнучкість і відмову від складної комерційної моделі

Ще одним важливим аспектом, що суттєво впливає на вибір платформи, є модель ліцензування і відповідно — витрати на впровадження та обслуговування. OpenStack, як повністю відкрите рішення, поширюється

за ліцензією Apache 2.0 і не потребує жодних комерційних платежів за використання. Це дозволяє не тільки вільно змінювати код і розширювати функціональність, а й виключає додаткові витрати при масштабуванні середовища. Водночас проєкти на основі OpenStack можуть інтегрувати сторонні рішення та розширення, які також мають відкритий код, що значно знижує сукупну вартість володіння інфраструктурою.

На відміну від цього, VMware базується на закритому ліцензуванні, яке включає оплату за кожен фізичний хост, додаткові компоненти (vSAN, NSX, vCenter), а також платну технічну підтримку. При зростанні обсягу інфраструктури, витрати пропорційно збільшуються, і це може стати критичним фактором для компаній, що обмежені в бюджеті. Водночас, комерційна підтримка та надійність VMware, безумовно, є її сильною стороною, особливо для підприємств, де критично важлива доступність систем і відповідальність постачальника.

Proxmox, хоч і позиціонує себе як open-source рішення, у своїй бізнес-моделі застосовує модель підписки. Саме програмне забезпечення можна використовувати безкоштовно, але для доступу до стабільних оновлень та офіційної підтримки пропонується придбати комерційну ліцензію. Це дозволяє зберегти баланс між відкритістю коду і фінансуванням подальшого розвитку проєкту, але водночас створює деяку невизначеність для великих організацій щодо довгострокової стратегії використання.

Щодо технічної спільноти, варто зазначити, що OpenStack має одну з найбільших екосистем у світі відкритого ПЗ. Його розвиток підтримують такі технологічні гіганти, як Red Hat, Canonical, SUSE, Intel, NVIDIA, які не тільки сприяють розширенню функціоналу, а й забезпечують високу якість коду, регулярні оновлення та безпекові патчі. Активна спільнота та доступна документація є важливою перевагою для адміністраторів, які стикаються з нетиповими проблемами, адже часто вирішення можна знайти в обговореннях, офіційних дискусіях або репозиторіях проєкту.

Також слід згадати про рівень відповідності сучасним стандартам

хмарних технологій. OpenStack повністю підтримує автоматизацію за допомогою Terraform, Helm, Kolla, інтеграцію з Kubernetes, Prometheus, Ceph і багатьма іншими інструментами сучасного DevOps-ландшафту. Це дозволяє створювати хмари, які не тільки є функціональними, але й відповідають принципам "cloud-native" інфраструктури. У той час як VMware також поступово впроваджує сумісність із контейнерними середовищами (через Tanzu), її інтеграційні можливості обмежені рамками власної екосистеми. Proxmox підтримує базову автоматизацію через API або сторонні скрипти, але не має повноцінної підтримки таких комплексних сценаріїв.

З урахуванням цих факторів можна зробити висновок, що OpenStack є найбільш економічно доцільним вибором для організацій, які прагнуть отримати повний контроль над своєю інфраструктурою, уникнути залежності від конкретного постачальника та впроваджувати сучасні практики DevOps. При цьому успішне розгортання потребує відповідного рівня компетенції та досвіду. Для компаній, що не мають внутрішніх ресурсів для розгортання OpenStack або мають обмежений кадровий потенціал, платформи на кшталт VMware або Proxmox можуть стати швидшим способом досягнення результату, хоча й зі своїми обмеженнями та додатковими витратами.

2.9 Безпека та мультиорендність в OpenStack

Одним з найважливіших аспектів хмарної інфраструктури є безпека. OpenStack спроектований як мультиорендне середовище, тобто одночасне використання ресурсів кількома клієнтами (тенантами) з ізоляцією даних і політик доступу. Це дозволяє хостити кілька незалежних середовищ на одній фізичній інфраструктурі.

OpenStack використовує компонент Keystone для автентифікації, авторизації та управління політиками. Keystone підтримує інтеграцію з LDAP, SAML та іншими механізмами єдиного входу (SSO), що важливо для корпоративного використання.

Ще одним критичним компонентом є Neutron — мережева служба, яка забезпечує ізоляцію мереж на рівні віртуальних приватних мереж (VLAN, VXLAN) або з використанням безпечних тунелів. Крім того, завдяки гнучким правилам безпеки (security groups) OpenStack дозволяє granularний контроль над трафіком між віртуальними машинами.

Для збереження даних компоненти, як Glance (зберігання образів), Swift (об’єктне сховище) та Cinder (блочне сховище) реалізують шифрування, контроль доступу та аудит операцій. OpenStack також підтримує інтеграцію з зовнішніми засобами моніторингу, сканування вразливостей та журнального аудиту (наприклад, OpenStack Security Notes, OSSN).

Загалом, безпека OpenStack базується на гнучкій архітектурі, яка дозволяє застосовувати як вбудовані засоби захисту, так і сторонні рішення, що робить платформу надійним вибором для підприємств із високими вимогами до конфіденційності та захисту даних.

2.10 Тренди в хмарних технологіях

Сучасні хмарні технології перебувають у стані постійної еволюції, що зумовлено як технологічними інноваціями, так і змінами в потребах бізнесу та користувачів. Одним із ключових напрямів трансформації стала поступова заміна традиційних віртуальних машин на більш легкі й гнучкі контейнери. Контейнеризація стала наріжним каменем сучасних IT-архітектур, і такі інструменти, як Kubernetes, фактично стали стандартом для їх оркестрації. Платформи, що надають інфраструктуру як послугу, дедалі частіше інтегрують засоби управління контейнерами, дозволяючи розгортати мікросервісні додатки з високою масштабованістю та стійкістю.

Ще одним стратегічним напрямом розвитку є периферійні обчислення (edge computing). Оскільки все більше пристроїв генерують дані на краю мережі — від індустріальних сенсорів до автономних автомобілів — виникає

потреба виконувати обчислення ближче до джерела даних. Це зменшує затримки, розвантажує центральні дата-центри і підвищує ефективність роботи критичних систем. OpenStack відповідає на цей запит через спеціалізовані проєкти, такі як StarlingX, що дозволяють ефективно розгортати та керувати інфраструктурою в edge-середовищах.

Іншим вектором розвитку хмар є поєднання обчислювальних ресурсів із технологіями штучного інтелекту та машинного навчання. З огляду на потребу в потужних обчисленнях і великих обсягах даних, хмара стала природним середовищем для експериментів, тренування моделей і запуску AI-додатків у продакшн. Платформи оптимізуються для роботи з графічними процесорами, використовують спеціалізовані апаратні прискорювачі та пропонують середовища для розробки з готовими фреймворками.

Крім цього, екологічні виклики спонукають до розвитку так званої "зеленої хмари". Хмарні провайдери все більше уваги приділяють енергоефективності, автоматично масштабуючи ресурси, знижуючи споживання електроенергії в моменти низького навантаження та використовуючи поновлювані джерела енергії. Цей підхід дозволяє не лише знизити вартість інфраструктури, а й відповідати вимогам сталого розвитку.

Нарешті, важливим трендом залишається автоматизація та підхід Infrastructure as Code (Інфраструктура як код). Завдяки використанню таких інструментів, як Ansible, Terraform чи Helm, інфраструктура описується у вигляді декларативних конфігурацій, що підвищує передбачуваність, прискорює розгортання та полегшує інтеграцію в CI/CD процеси.

OpenStack, як відкрита платформа, здатна адаптуватися до всіх зазначених трендів. Його модульність, підтримка інтеграцій, активна спільнота та орієнтація на масштабованість роблять цю технологію актуальною у швидкоплинному технологічному середовищі сучасного цифрового світу.

3 РОЗГОРТУВАННЯ OPENSTACK

OpenStack є однією з наймасштабованіших і найгнучкіших платформ для побудови хмарної інфраструктури, і її успішне впровадження починається з детально спланованого підготовчого етапу. На цьому етапі формуються апаратні та програмні передумови для майбутнього розгортання, здійснюється проєктування мережевої топології, визначається роль кожного вузла в інфраструктурі та проводиться базове налаштування середовища.

3.1. Підготовчий етап

Перш за все, необхідно підготувати фізичні або віртуальні сервери, які будуть задіяні у майбутньому кластері. Мінімальна рекомендована конфігурація для кожного вузла — процесор з чотирма ядрами, 8 Гб оперативної пам’яті, а також принаймні 50 Гб вільного простору на диску. Особлива увага приділяється мережевим інтерфейсам: для повноцінної роботи OpenStack потрібно щонайменше два мережевих інтерфейси — один для зовнішнього трафіку, інший для внутрішньої взаємодії між контейнерами та сервісами.

На даному етапі також проводиться встановлення базової операційної системи на всі вузли (рисунок 3.1). У даній роботі використовується Ubuntu 22.04 LTS як стабільна та підтримувана версія з широкою сумісністю із OpenStack-компонентами. Після встановлення ОС виконується базове налаштування: встановлення SSH-доступу, синхронізація часу через NTP, додавання користувача з правами sudo, налаштування статичної IP-адреси, а також оновлення системних пакетів до актуальних версій.

Паралельно необхідно визначити логічну структуру майбутнього хмарного середовища: які вузли будуть виконувати роль інфраструктурних (control plane), які — обчислювальних (compute), які — сховищ даних

(storage), і яким чином ці вузли будуть взаємодіяти між собою. Важливо також врахувати, чи планується використання одного сервера у режимі All-in-One, чи буде реалізовано повноцінне розділення на декілька фізичних машин або віртуальних інстансів.

Завершальним кроком підготовчого етапу є перевірка мережевої доступності між усіма вузлами, встановлення всіх необхідних утиліт, таких як curl, git, lvm2, python3, bridge-utils, та підготовка до завантаження та конфігурування середовища OpenStack-Ansible. Саме цей інструмент у подальших кроках буде використовуватись для автоматизованого розгортання OpenStack-компонентів у контейнерах, забезпечуючи стандартизацію, масштабованість та швидке розгортання.

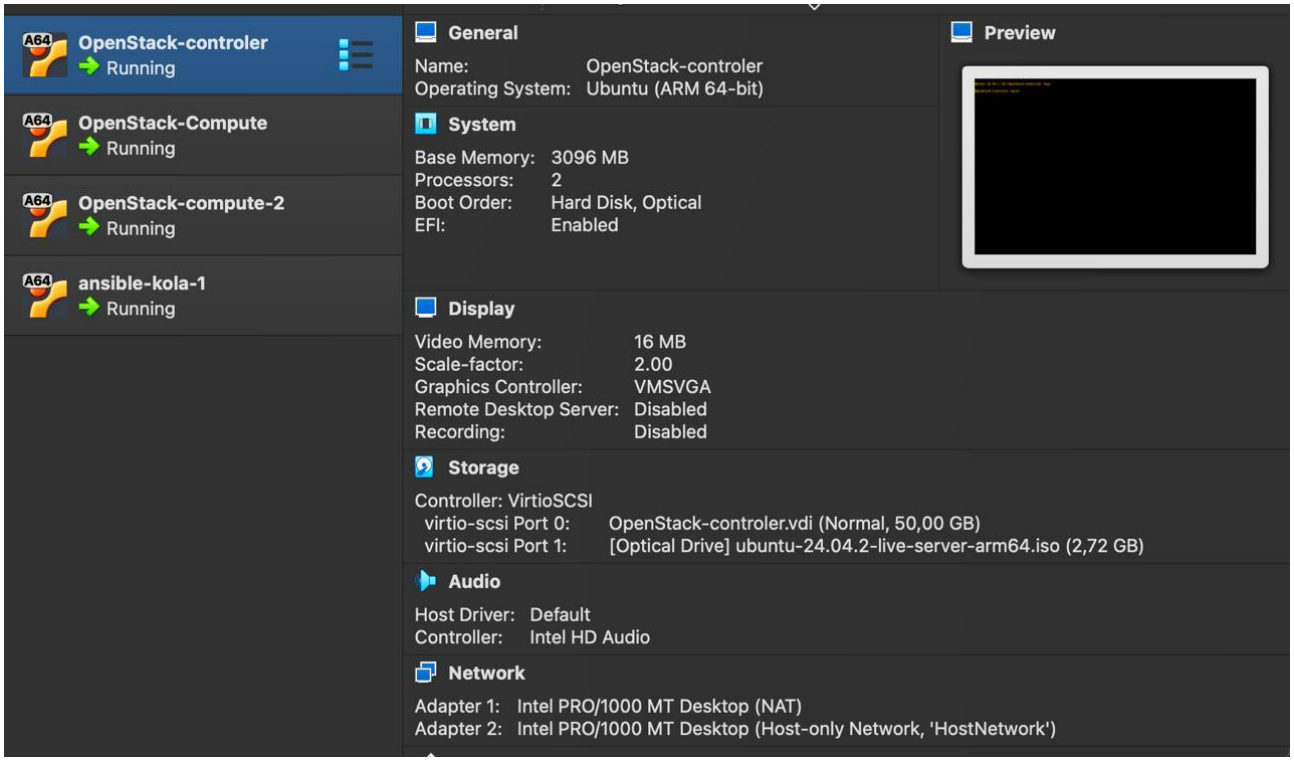


Рисунок 3.1 – Віртуальні машини з налаштуваннями у VirtualBox або VMware.

3.2. Встановлення необхідних пакетів на сервері управління

Спочатку на сервері управління встановлюється (рисунок 3.2)

необхідне програмне забезпечення (лістинг 3.1).

Лістинг 3.1 – Оновлення пакетного менеджера та встановлення базових утиліт для керування репозиторіями

```
sudo apt update
sudo apt install -y git software-properties-common
```

- git – для роботи з репозиторієм OpenStack-Ansible.
- software-properties-common – для зручного керування репозиторіями та додатковими пакетами.

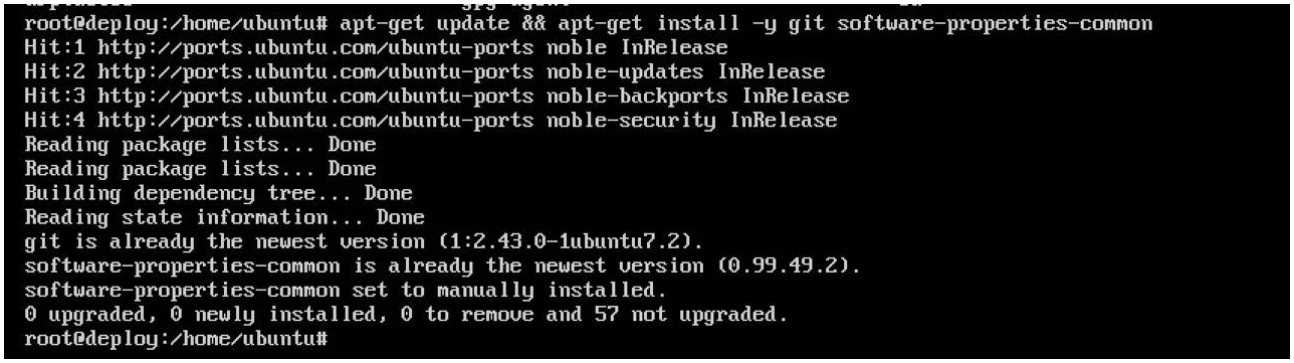


Рисунок 3.2 – Консоль з результатами виконання команд.

3.3. Завантаження репозиторію OpenStack-Ansible

Репозиторій містить необхідні playbook-и для автоматизованого розгортання і нам потрібно заванатажити його на машину (лістинг 3.2)

Лістинг 3.2 – Клонування репозиторію OpenStack-Ansible та перехід до робочого каталогу

```
git clone https://opendev.org/openstack/openstack-ansible
/opt/openstack-ansible
cd /opt/openstack-ansible
```

Репозиторій OpenStack-Ansible (рисунок 3.3) включає в себе всі необхідні скрипти і конфігурації, що дозволяють автоматизовано розгорнути

повноцінний OpenStack-кластер, забезпечуючи стандартизовану конфігурацію середовища.

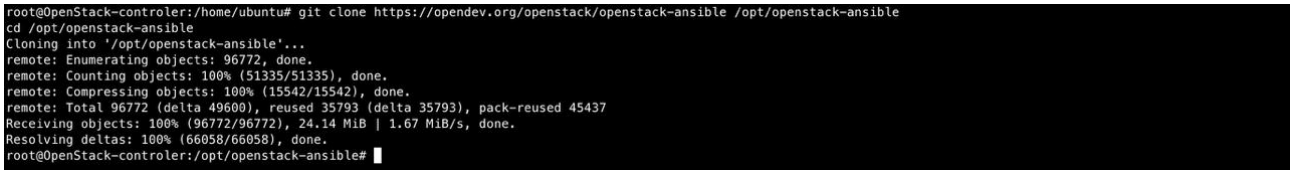


Рисунок 3.3 – Список файлів після завантаження репозиторію.

3.4. Вибір необхідної версії та встановлення залежностей

Для робочого клауду потрібен перехід на стабільну версію (лістинг 3.3)

Лістинг 3.3 – Перехід до стабільної версії OpenStack-Ansible та встановлення залежностей

```
git checkout yoga-eol
scripts/bootstrap-ansible.sh
scripts/bootstrap-aio.sh
```

Рисунок 3.4 – Виведення командної консолі під час виконання скриптів.

Скрипт bootstrap-ansible.sh встановлює Ansible з усіма необхідними модулями та ролями (рисунок 3.4), а bootstrap-aio.sh готує базову конфігурацію контейнерів LXC для розгортання компонентів OpenStack.

3.5 Конфігурація інвентарю

Файл інвентарю /etc/openstack_deploy/openstack_user_config.yml описує сервери і мережі (лістинг 3.4):

Лістинг 3.4 – Приклад інвентаризації серверів та мереж у OpenStack-Ansible

```
infra_hosts:
  infra1:
    ip: 192.168.56.2
compute_hosts:
  compute1:
    ip: 192.168.56.3
storage_hosts:
  storage1:
    ip: 192.168.56.4

provider_networks:
  - network:
      container_bridge: "br-mgmt"
      container_interface: "eth1"
      ip_from_q: "management"
      type: "raw"
      is_management_address: true
```

Файл конфігурації визначає типи серверів (рисунок 3.5) та мережі,

якими вони пов'язані, забезпечуючи гнучку мережеву конфігурацію для компонентів OpenStack.

```
---
cidr_networks:
  management: 192.168.56.0/24
  tunnel: 172.29.240.0/22
  storage: 172.29.244.0/22

used_ips:
  - "172.29.240.1,172.29.240.50"
  - "172.29.244.1,172.29.244.50"
  - "172.29.248.1,172.29.248.50"

global_overrides:
  # The internal and external VIP should be different IPs, however they
  # do not need to be on separate networks.
  external_lb_vip_address: 192.168.56.2
  internal_lb_vip_address: 192.168.56.2
  management_bridge: "br-mgmt"
  provider_networks:
    - network:
        container_bridge: "br-mgmt"
        container_type: "veth"
        container_interface: "eth1"
        ip_from_q: "management"
        type: "raw"
        group_binds:
          - all_containers
          - hosts
        is_management_address: true
    - network:
        container_bridge: "br-vxlan"
        container_type: "veth"
        container_interface: "eth10"
        ip_from_q: "tunnel"
        type: "vxlan"
        range: "1:1000"
        net_name: "vxlan"
        group_binds:
          - neutron_openvswitch_agent
    - network:
        container_bridge: "br-vlan"
        container_type: "veth"
        container_interface: "eth12"
        host_bind_override: "eth12"
        type: "flat"
        net_name: "physnet1"
        group_binds:
          - neutron_openvswitch_agent
    - network:
        container_bridge: "br-vlan"
        container_type: "veth"
        container_interface: "eth11"
        type: "vlan"
        range: "101:200,301:400"
        net_name: "physnet2"
"/etc/openstack_deploy/openstack_user_config.yml" 149L, 3180B
```

Рисунок 3.5. – Вміст файлу конфігурації.

3.6. Генерація паролів

На цьому етапі (лістинг 3.5) відбувається автоматизоване створення унікальних паролів для всіх компонентів майбутньої хмарної інфраструктури. Це надзвичайно важливий процес, який дозволяє гарантувати безпечну взаємодію між різними службами OpenStack. Генерація паролів виконується за допомогою вбудованого скрипту, який входить до складу OpenStack-Ansible і забезпечує створення файлу з обліковими даними

Лістинг 3.5 – Генерація паролів для сервісів OpenStack за допомогою скрипту

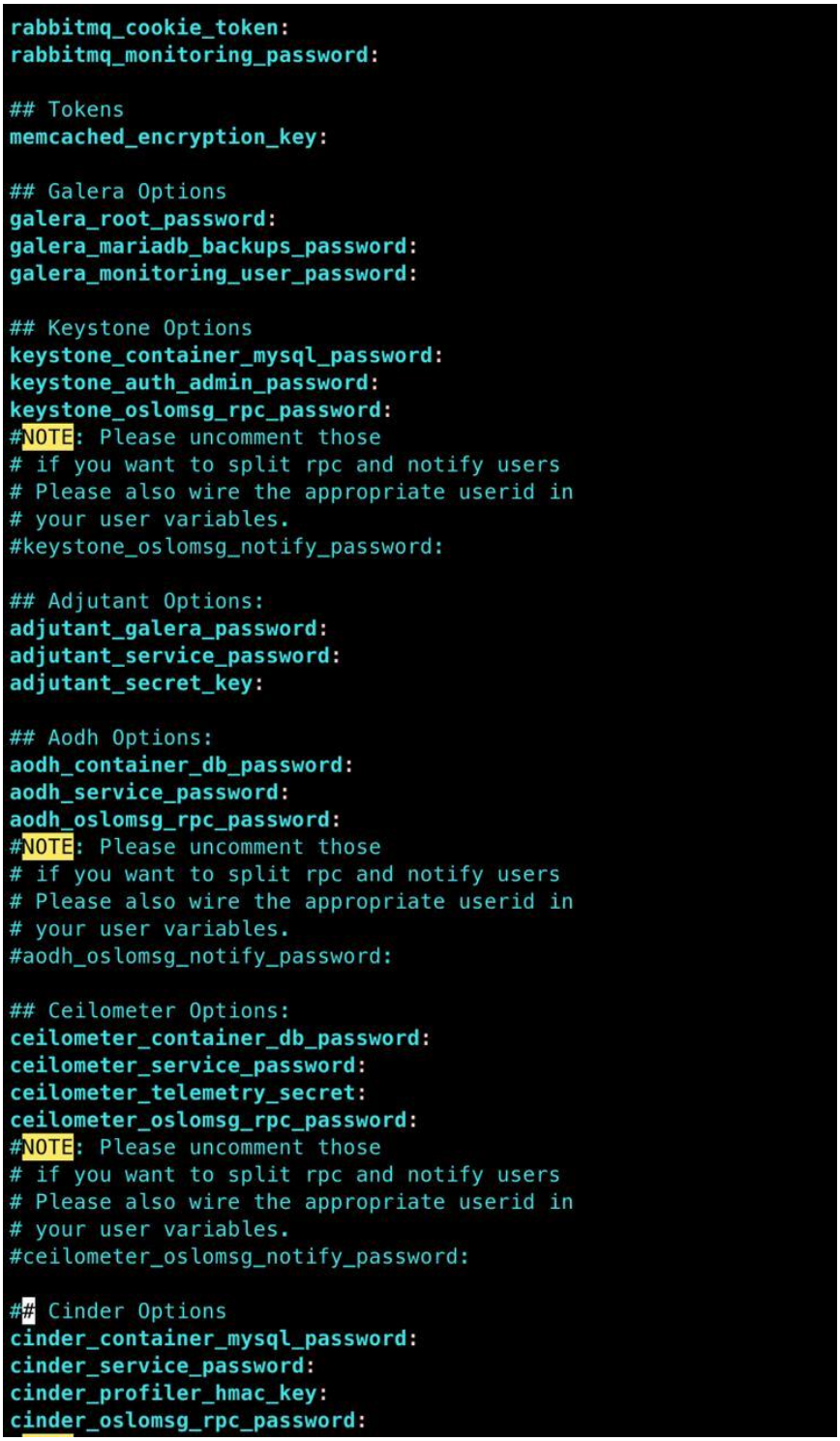
```
./scripts/pw-token-gen.py --file
/etc/openstack_deploy/user_secrets.yml
```

Результатом виконання цієї команди є файл `user_secrets.yml` (рисунок 3.6), що містить згенеровані паролі для таких ключових сервісів, як Keystone (ідентифікація та аутентифікація), MariaDB (база даних), RabbitMQ (черга повідомлень), Horizon (веб-інтерфейс), Glance (сховище образів), Nova (обчислювальні ресурси) та багатьох інших. Цей файл зберігається у захищеному вигляді у каталозі `/etc/openstack_deploy/` і використовується на наступних етапах конфігурації та розгортання.

Автоматична генерація облікових даних дозволяє уникнути помилок, які часто виникають при ручному введенні складних паролів, а також забезпечує унікальність кожного запису. Завдяки цьому знижується ризик витоку або компрометації даних, що особливо важливо для систем, які працюють у мережах з підвищеними вимогами до безпеки. Крім того, автоматизоване створення паролів полегшує подальше адміністрування системи — у разі необхідності паролі можна перевизначити чи оновити централізовано.

Цей підхід є рекомендованою практикою при розгортанні OpenStack,

оскільки дозволяє стандартизувати процес безпеки для середовищ різного масштабу — від тестових стендів до великих промислових кластерів.



```
rabbitmq_cookie_token:
rabbitmq_monitoring_password:

## Tokens
memcached_encryption_key:

## Galera Options
galera_root_password:
galera_mariadb_backups_password:
galera_monitoring_user_password:

## Keystone Options
keystone_container_mysql_password:
keystone_auth_admin_password:
keystone_oslomsg_rpc_password:
#NOTE: Please uncomment those
# if you want to split rpc and notify users
# Please also wire the appropriate userid in
# your user variables.
#keystone_oslomsg_notify_password:

## Adjutant Options:
adjutant_galera_password:
adjutant_service_password:
adjutant_secret_key:

## Aodh Options:
aodh_container_db_password:
aodh_service_password:
aodh_oslomsg_rpc_password:
#NOTE: Please uncomment those
# if you want to split rpc and notify users
# Please also wire the appropriate userid in
# your user variables.
#aodh_oslomsg_notify_password:

## Ceilometer Options:
ceilometer_container_db_password:
ceilometer_service_password:
ceilometer_telemetry_secret:
ceilometer_oslomsg_rpc_password:
#NOTE: Please uncomment those
# if you want to split rpc and notify users
# Please also wire the appropriate userid in
# your user variables.
#ceilometer_oslomsg_notify_password:

## Cinder Options
cinder_container_mysql_password:
cinder_service_password:
cinder_profiler_hmac_key:
cinder_oslomsg_rpc_password:
```

Рисунок 3.6 – Частина згенерованих паролів (з прихованими значеннями).

3.7. Виконання розгортання OpenStack

Запуск розгортання виконується у три етапи:

- Налаштування хоста openstack-ansible setup-hosts.yml (рисунок 3.7)

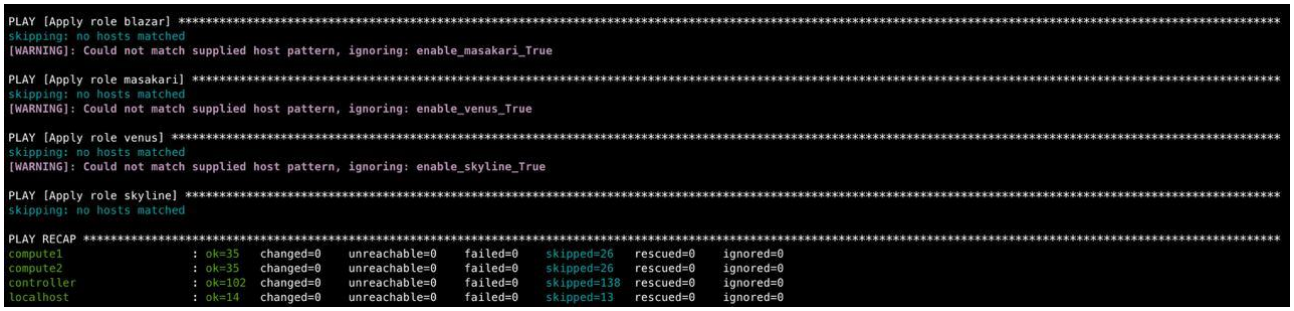


Рисунок 3.7 – : Вивід етапу розгортання openstack-ansible setup-hosts.yml

- Розгортання інфраструктури (база даних, RabbitMQ): openstack-ansible setup-infrastructure.yml (рисунок 3.8)

- Встановлення всіх сервісів OpenStack (Nova, Neutron, Glance тощо): openstack-ansible setup-openstack.yml

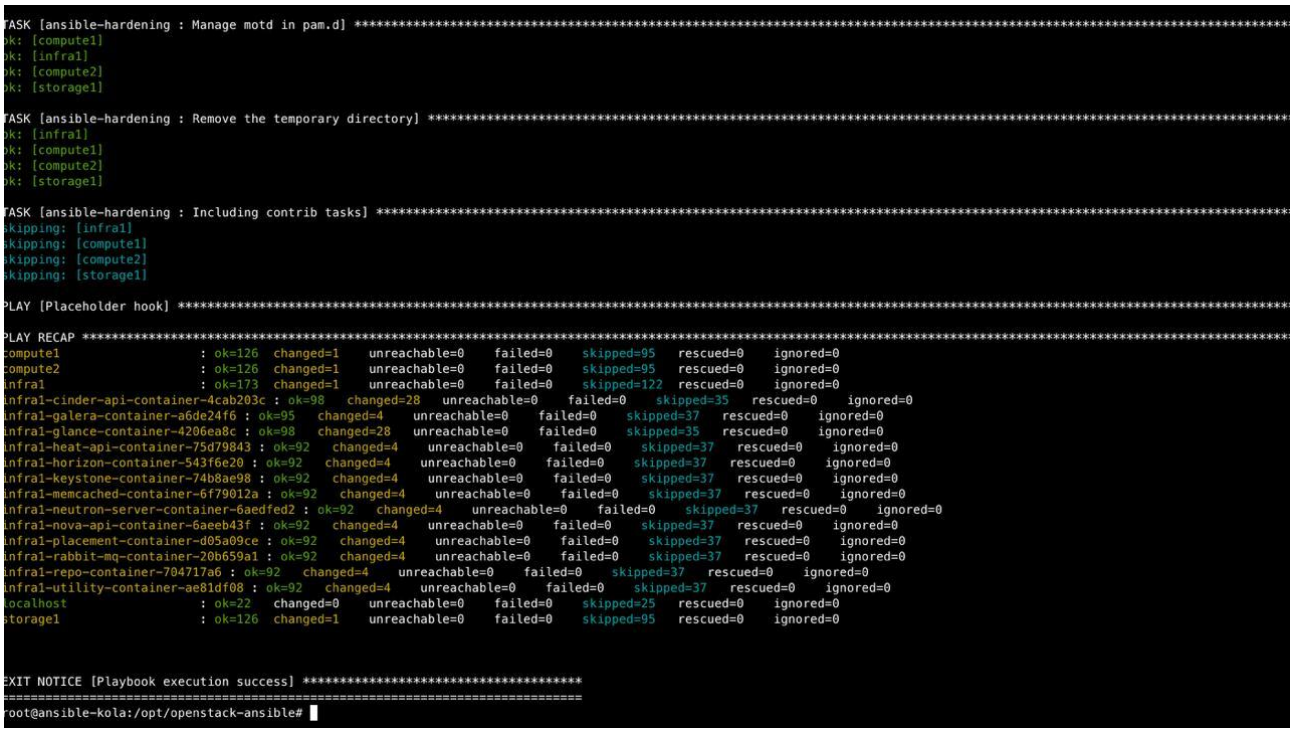


Рисунок 3.8 – : Вивід етапу розгортання openstack-ansible setup-infrastructure.yml

Кожен етап відповідає за розгортання різних компонентів, починаючи від підготовки інфраструктури (контейнери, база даних, черга повідомлень)

до повноцінного встановлення всіх сервісів OpenStack.

3.8. Перевірка стану розгорнутої хмарної інфраструктури

Перевіряємо доступність OpenStack за допомогою Horizon веб-інтерфейсу (рисунок 3.9).

Перевірка працездатності через веб-інтерфейс Horizon дозволяє переконатися у правильному функціонуванні всіх сервісів OpenStack, а також оперативно виявити можливі проблеми.

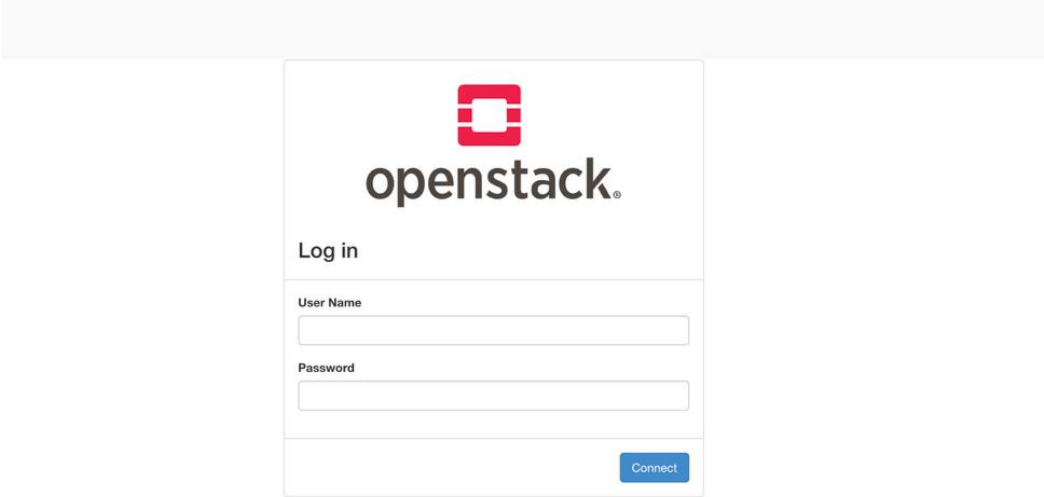


Рисунок 3.9 – Веб-інтерфейс Horizon, CLI-вивід списків сервісів.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було здійснено комплексне дослідження сучасних підходів до проєктування та розгортання хмарної інфраструктури з використанням платформи OpenStack. OpenStack, як відкрите та масштабоване рішення, зарекомендувало себе як одне з найпопулярніших середовище для побудови приватних і публічних хмарних систем. У роботі розглянуто ключові компоненти OpenStack, що забезпечують роботу хмарної інфраструктури за моделлю IaaS. Особливу увагу приділено аналізу модулів Nova, Neutron, Cinder, Swift та Keystone, кожен з яких виконує критично важливу роль в управлінні ресурсами, мережею, сховищами та безпекою доступу користувачів.

Визначено, що ефективне розгортання та експлуатація OpenStack потребує інтеграції з інструментами автоматизації. У цьому контексті було розглянуто та порівняно два сучасних інструменти керування конфігураціями — Ansible та Terraform. Ansible виявився особливо корисним у завданнях конфігурації компонентів OpenStack та оркестрації процесів, тоді як Terraform забезпечує зручний підхід до опису інфраструктури за допомогою декларативного синтаксису та ефективного управління її станом. Проведений аналіз підтвердив, що використання цих інструментів суттєво підвищує відтворюваність, масштабованість та надійність розгортання хмарної платформи.

Завдяки експериментальному моделюванню було перевірено працездатність ключових компонентів OpenStack у віртуалізованому середовищі. Особливу увагу було приділено інтеграції між окремими службами через API-інтерфейси, а також забезпеченню доступу до ресурсів кінцевим користувачам. У результаті дослідження встановлено, що платформа OpenStack має високий потенціал до адаптації в межах різноманітних сценаріїв використання, включаючи обробку великих обсягів

даних, підтримку CI/CD, а також створення приватних хмар для внутрішніх потреб підприємств.

Крім того, розглянуто аспекти безпеки, багатокористувацького доступу та масштабування інфраструктури, що дозволило більш повно оцінити можливості платформи OpenStack у сучасному IT-середовищі. Встановлено, що реалізація багаторівневих політик доступу та гнучке управління обчислювальними й мережевими ресурсами забезпечують високий рівень керованості хмарного середовища. Також проаналізовано архітектурні особливості OpenStack, які забезпечують її стійкість до відмов та розширюваність за рахунок модульної структури.

Отримані результати дозволяють стверджувати, що OpenStack у поєднанні з інструментами автоматизації розгортання, такими як Terraform та Ansible, є ефективним рішенням для побудови надійної хмарної інфраструктури. Така інфраструктура здатна забезпечити гнучке масштабування ресурсів, оперативне реагування на зміни у навантаженні та зручне адміністрування. Це особливо важливо для підприємств, установ та організацій, які функціонують у швидкозмінному інформаційному середовищі й потребують високого рівня технологічної гнучкості.

Таким чином, поставлені у роботі цілі були досягнуті. Здійснено ґрунтовний теоретичний аналіз архітектури та компонентів OpenStack, проведено порівняльне дослідження засобів автоматизації інфраструктури, продемонстровано приклад практичного розгортання хмари та доведено ефективність запропонованого підходу. Отримані результати можуть бути використані як основа для подальших досліджень у галузі хмарних технологій, а також для впровадження практичних рішень у сфері приватних та гібридних хмар.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is IBM Cloud? Services Offered, Features & Pricing. Datamation. URL: https://www.datamation.com/cloud/ibm-cloud/ (дата звернення: 09.04.2025).

2. Getting started with Red Hat CodeReady Containers. Red Hat. URL: https://access.redhat.com/documentation/en-us/red_hat_codeready_containers/1.0/html/getting_started_guide/getting-started-with-codeready-containers_gsg (дата звернення: 12.04.2025).

3. Operator pattern. Kubernetes Documentation. URL: https://kubernetes.io/docs/concepts/extend-kubernetes/operator/ (дата звернення: 15.04.2025).

4. Kubernetes Service Mesh Definition. Avi Networks. URL: https://avinetworks.com/glossary/kubernetes-service-mesh/ (дата звернення: 18.04.2025).

5. Serverless on Kubernetes. Medium. URL: https://medium.com/appvia/serverless-on-kubernetes-63b49aeaf4ef (дата звернення: 21.04.2025).

6. Kubernetes extension. Quarkus. URL: https://quarkus.io/guides/deploying-to-kubernetes (дата звернення: 24.04.2025).

7. CNCF. Cloud Native Computing Foundation. URL: https://www.cncf.io/ (дата звернення: 27.04.2025).

8. Terraform. Terraform.io. URL: https://www.terraform.io/ (дата звернення: 30.04.2025).

9. HashiCorp. Terraform Documentation. URL: https://developer.hashicorp.com/terraform/docs (дата звернення: 03.05.2025).

10. Creating a Highly Available Architecture with Terraform. HashiCorp. URL: https://www.hashicorp.com/ (дата звернення: 05.05.2025).

11. OpenStack Keystone – Identity service. OpenStack Documentation.

URL: https://docs.openstack.org/keystone/latest/ (дата звернення: 23.06.2025).

12. OpenStack Nova – Compute service. OpenStack Documentation. URL: https://docs.openstack.org/nova/latest/ (дата звернення: 23.06.2025).

13. OpenStack Neutron – Networking service. OpenStack Documentation. URL: https://docs.openstack.org/neutron/latest/ (дата звернення: 23.06.2025).

14. OpenStack Cinder – Block Storage service. OpenStack Documentation. URL: https://docs.openstack.org/cinder/latest/ (дата звернення: 23.06.2025).

15. OpenStack Horizon – Web UI Dashboard. OpenStack Documentation. URL: https://docs.openstack.org/horizon/latest/ (дата звернення: 23.06.2025).