МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет будівництва і архітектури

В. М. Хроленко, В. Г. Голенков

Хмарні та GRID-технології

Рекомендовано вченою радою Київського національного університету будівництва і архітектури як навчальний посібник для здобувачів першого (бакалаврського) рівня вищої освіти галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки»

Київ 2025

УДК 004
Х 94

Рецензенти: І. А. Ачкасов, д-р техн. наук, професор, КНУБА;
О. В. Федусенко, канд. техн. наук, доцент,
КНУ ім. Тараса Шевченка;
В. І. Кудін, д-р техн. наук, професор,
КНУ ім. Тараса Шевченка

Затверджено на засіданні вченої ради Київського національного університету будівництва і архітектури, протокол № 3 від 8 жовтня 2024 року.

Хроленко В. М.

Х 94 Хмарні та GRID-технології: навчальний посібник / В. М. Хроленко, В. Г. Голенков. – Київ : КНУБА, 2025. – 100 с.

ISBN 978-966-627-270-9

Містить відомості з дисципліни «Хмарні та GRID-технології» щодо вивчення, застосування й реалізації способів побудови хмарного сервісу та використання його для подальшого розвитку інформаційних систем у будівництві, SOA-технологій, крос-платформних віртуальних технологій, технологій віртуалізації, а також щодо вирішення проблем використання послуг, наданих хмарними операторами. У посібнику також розглянуто питання переносимості мовної та платформної незалежності, прозорості місцезнаходження об’єктів, а також придбання практичних навичок із створення віртуальної машини.

Призначено для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 122 «Комп’ютерні науки».

УДК 004

ISBN 978-966-627-270-9

Зміст

Вступ

Метою дисципліни є придбання студентами теоретичних і практичних знань, навичок, методів і засобів побудови та використання хмарних сервісів і платформ.

Основні завдання: вивчення, ефективне застосування й реалізація способів побудови хмарного сервісу та використання спеціалізованих хмарних сервісів для подальшого розвитку інформаційних систем у будівництві, а також вирішення проблем використання послуг, наданих хмарними операторами, переносимості мовної та платформної незалежності, прозорості місцезнаходження об’єктів і придбання практичних навичок із застосування крос-платформних віртуальних технологій.

Розглядаються питання щодо вирішення проблем використання послуг, наданих хмарними операторами, переносимості мовної та платформної незалежності, прозорості місцезнаходження об’єктів, придбання практичних навичок із застосування крос-платформних віртуальних технологій, створення віртуальної машини, а також теоретичні знання про види Amazon Web Services, сценарії, акторів та ролі у хмарних обчисленнях, технології віртуалізації, SOA-технології, що слугували основою для хмар тощо.

Під час вивчення дисципліни студенти повинні опрацьовувати в лабораторних роботах лекційний матеріал, самостійно вивчати додаткову літературу, здійснювати підготовку до лабораторних занять.

Поточний і підсумковий контроль здійснюється за результатами усних опитувань на лекціях, проведення модульних письмових контрольних робіт, виконання індивідуальних лабораторних завдань за допомогою відповідного програмного забезпечення на комп’ютерах, а також здійснюється підсумковий комплексний контроль.

Студенти повинні знати архітектуру й моделі хмарних досліджень, технології віртуалізації та її різновиди. Студенти повинні вміти створювати віртуальну машину й орієнтуватися в хмарних сервісах Amazon.

Тема 1. Хмарні середовища

1.1. Технологія хмарних середовищ

1.2.1. Хмарні обчислення (англ. Cloud computing)

Це модель забезпечення повсюдного доступу до мережі на вимогу до загального пулу (англ. Pool) обчислювальних ресурсів, наприклад мереж передачі даних, серверів, пристроїв зберігання даних, додатків і сервісів як разом, так і окремо, які можуть бути оперативно надані та звільнені з мінімальними експлуатаційними витратами та/або зверненнями до провайдера.

При цьому в користувача (клієнта) фактично залишається лише інтерфейс його інформаційної системи, а його дані, програмні засоби, які він використовував, та інформаційна інфраструктура перебувають у провайдера, залежність від якого стає фатальною.

Основними причинами виникнення та просування хмарних технологій є:

• природний надлишок обчислювальних потужностей і пам'яті суперкомп'ютерів, підключених до мережі каналами високої пропускної здатності (пошукові системи, потужні хостинги та ін.);

• бажання власників цих потужностей отримати від них прибуток;

• бажання власників цих потужностей і політичних сил, що стоять за ними, зробити масового користувача по всьому світу залежним від цих послуг та істотно керованим.

• бажання побудувати новий інформаційний порядок на планеті шляхом впливу на політику держав через Інтернет.

1.1.2. Історичні міфи і реальність

Спочатку концепція використання обчислювальних ресурсів за принципом системи поділу часу була запропонована в 1960-ті роки, і до 1990-х років широко використовувалася у великих організаціях, де

були «мейнфрейми» – великі обчислювальні машини, і де багато користувачів були підключені до цих ЕОМ.

Це була ера мейнфреймів і колективного використання ЕОМ. У 1980-х роках з'явилися перші персональні комп'ютери: Apple, IBM, і з 1990-х почалася ера індивідуальних обчислювальних засобів, яка триває донині.

Паралельний розвиток мереж і виникнення мережі Інтернет наповнили персональні комп'ютери морем інформації і зробили їх основним інструментом епохи інформатизації суспільства.

Нова хвиля усуспільнення обчислювальні ресурсів виникла в останнє десятиліття й активно пропагується корпораціями – власниками суперкомп'ютерів і великих мереж.

Вони стали активно підкидати цю ідею в маси через інтернет-конференції. Під час дискусій висувалися різні версії, за однією з яких термін Cloud був уперше використаний главою компанії Google Еріком Шмідтом у виступі й набув поширення в засобах масової інформації.

Інша популярна версія припускає, що термін cloud computing став широко вживатися в США з 2005 року після запуску компанією Amazon.com проєкту Elastic Compute Cloud (Amazon EC2) і широко поширився в бізнесі, серед постачальників інформаційних технологій і в науково-дослідній середовищі.

Термін «хмара» використовується як метафора, заснована на зображенні інтернету на діаграмі комп'ютерної мережі, або як образ складної інфраструктури, за якою ховаються всі технічні деталі.

1.1.3. Міфи про хмарні обчислення

Концепція хмарних обчислень із публічної моделлю піддавалася критиці з боку співтовариства вільного програмного забезпечення і, зокрема, з боку Річарда Столлмана: «Використовувати вебдодатки для своїх обчислювальних процесів не слід, наприклад, тому, що ви втрачаєте над ними контроль. І це не краще, ніж використовувати будь-яку пропрієтарну програму. Робіть свої обчислення на своєму комп'ютері, використовуючи програми, які поважають вашу свободу. Якщо ви використовуєте будь-яку пропрієтарну програму або чужий вебсервер, ви стаєте беззахисними. Ви стаєте іграшкою в руках того,

хто розробив це ПЗ» (Cloud computing is a trap, warns GNU founder Richard Stallman, інтерв'ю газеті The Guardian [6] англійською).

Існує імовірність, що з повсюдним приходом цієї технології стане очевидною проблема створення неконтрольованих даних, коли інформація, залишена користувачем, зберігатиметься роками, або без його відома, або він буде не в змозі змінити якусь її частину. Прикладом того можуть слугувати сервіси Google, де користувач не може видалити невикористовувані їм сервіси й навіть видалити окремі групи даних, створені в деяких з них (FeedBurner, Google Friend Connect і, можливо, інші).

Як альтернатива «очищення» свого профілю пропонується створити новий. Однак не варто забувати про те, що ім'я користувача вже зайнято попередньої обліковим записом, а нові – на кшталт John22441 – влаштовують не всіх. Оскільки хмарні обчислення будуть повністю пропрієтарні (відкритий API не виправляє ситуацію), поки немає надії на те, що користувачеві нададуть засіб для видалення своїх же даних на подібних серверах.

Крім того, деякі аналітики припускали появу у 2010 році проблем зі хмарними обчисленнями. Так, наприклад, Марк Андерсон, керівник галузевого IT-видання Strategic News Service, вважав, що через значний приплив користувачів сервісів, що використовують хмарні обчислення (наприклад, Flickr або Amazon), росте вартість помилок і витоків інформації з подібних ресурсів, а 2010 року мали відбутися великі «катастрофи типу виходу з ладу, або катастрофи, пов'язані з безпекою». Так, наприклад, у 2009 році сервіс для зберігання закладок Magnolia втратив всі свої дані. Проте багато експертів дотримуються позиції, що переваги і зручності переважують можливі ризики використання подібних сервісів.

1.1.3. Міф про загальне у переході в хмарні середовища

Хоча перехід до хмарного середовища ще не став загальним, немає ніяких сумнівів, що це переважаюча тенденція, яка буде рости й розвиватися.

За даними опитування, проведеного компанією Advanced Micro Devices у 2011 році, вже близько 37 % компаній користуються хмарною

інфраструктурою. І 24 % учасників опитування, проведеного у вересні 2011 року компанією Emerson Network Power серед менеджерів ЦОД, реселерів та інженерів, підтвердили, що вони планують впровадити або розробити стратегію використання хмарних обчислень у найближчі 18 місяців. За оцінками компанії IDC, обсяги продажів на ринку продуктів і послуг публічного хмарного середовища виростуть з 16 млрд дол. США у 2010 році до 56 млрд у 2014 році. Фахівці Gartner були ще більш оптимістичні: за їхніми оцінками, обсяг продажів хмарних послуг до 2013 року досягне 150 млрд дол. Незалежно від того, чиї прогнози виявилися більш точними, очевидно те, що хмарні технології розвиваються.

Що очікує ринок хмарних технологій у 2023–2028 рр., які виклики та тенденції?

Аналітична компанія MarketsandMarkets™ опублікувала глобальне дослідження, у якому визначила ринкову частку хмарних обчислень (cloud computing market share), а також тенденції ринку (cloud computing market trends). Згідно із зібраними даними, до 2028 року, розмір глобального ринку зросте до $1266,4 млрд, якщо збережеться середньорічний темп приросту 15,1 %. Для порівняння: у 2023 році ринкова частка становила $626,4 млрд.

Чому ж деякі ще сумніваються й вагаються? Хоч з моменту виникнення хмарного середовища провайдери значно поліпшили механізми захисту даних і збереження конфіденційності, побоювання щодо доступу до інформації, її розміщення та передачі зберігаються, що перешкоджає повсюдному застосуванню хмарних технологій.

Питання гнучкості системи загалом – ще одна проблема публічного хмарного середовища. Крім того, перебої в роботі деяких відомих провайдерів хмарних послуг, імовірно, є досить вагомою причиною стриманості деяких компаній, діяльність яких цілком залежить від стабільної доступності мережі.

1.1.4. Міф про надійність хмарних середовищ

Пропаганда корпорацій, які надають хмарні сервіси, вселяє масам споживачів, що перехід до хмарного середовища дасть можливість назавжди забути про збої і простої. Чи так це?

Коли компанія починає користуватися хмарними послугами зовнішнього провайдера, ймовірні відмови просто переміщуються з центру даних компанії в центр даних провайдера. Єдине, що змінюється, – ступінь контролю над ситуацією. У власному центрі обробки даних ви самі керуєте інфраструктурою і контролюєте доступність мережі та даних. Такий контроль, природно, тягне за собою ряд питань, пов'язаних із власністю і відповідальністю, які можуть слугувати першочерговою причиною переходу до хмарного середовища.

Управління завжди має на увазі відповідальність, а деякі менеджери відділів інформаційних технологій воліють зняти зі своїх плечей тягар відповідальності за забезпечення надійності мережі.

З переходом до хмарного середовища управління інфраструктурою, що захищає ваші дані, значною мірою переходить до провайдера хмарних послуг. Перед переходом важливо проаналізувати інформаційну інфраструктуру й надійність ЦОД-провайдера, вивчивши історію збоїв.

Цілком можливо, що інфраструктура ЦОД-провайдера надійніша, ніж у вашій компанії, однак перед переходом краще придивитися до провайдера, щоб забезпечити максимальну сумлінність, відповідальність і надійність SLA.

Якщо організація робить вибір на користь власного приватного хмарного середовища, вона може захистити себе від проблем, пов'язаних із неполадками на одному сервері, однак збиток від можливих перебоїв в роботі всього центру обробки даних буде дуже великий.

Забезпечення надійної системи електропостачання й охолодження є дуже важливим для підтримки приватного хмарного середовища, а контроль і управління інфраструктурою центру даних набуває ще більшого значення.

Врешті-решт, важливо пам'ятати, що сам по собі перехід у хмарне середовище не гарантує, що в разі простоїв ви зможете уникнути шкоди для вашої діяльності. Клієнти запам'ятають, що збій стався у вашій мережі, а не те, що винен у цьому провайдер хмарних послуг.

1.1.5. Міф про зниження витрат і навантаження

Пропаганда корпорацій, які надають хмарні сервіси, стверджує, що користувачі в будь-якому випадку з переходом до хмарних обчислень знизять витрати й навантаження на свою обчислювальну техніку. Чи це так?

Це звичайний аргумент на користь переходу до хмарного середовища, однак залежно від додатків, які ви плануєте розмістити в хмарній інфраструктурі, він може бути, а може і не бути істиною. Принаймні в короткостроковій перспективі перехід до хмарного середовища буде трудомістким. Компанії, що переходять до хмарного середовища, роблять це, щоб вирішити конкретні завдання й використовувати раніше недоступні ресурси (особливо сервери), які в хмарному середовищі доступні на вимогу.

Однак адаптація хмарних послуг до специфічних потреб компанії потребує змін в організації праці, здійснити які може виявитися непросто. І навіть якщо конфігурація середовища вам повністю підходить, відбір завдань для переміщення в хмарну інфраструктуру займає багато часу. Збільшення ефективності роботи може бути світлом у кінці тунелю, але ви повинні розуміти, що рухатися до цієї точки належить ще довго.

Що стосується зменшення витрат для замовників, що працюють в хмарному середовищі, відповідно до аналітичного документа McKinsey and Co. 2009 року, вони скорочуються, тільки якщо в хмарній інфраструктурі функціонують певні види платформ. В іншому випадку, стверджується в дослідженні, підтримання власного ЦОД залишається більш рентабельним.

Постачальники хмарних послуг намагаються оскаржувати ці твердження (і роблять це), але можна з упевненістю сказати, що кінцева економія з переходом до хмарного середовища – спірне питання. Якщо правильно застосувати хмарні технології, економія цілком імовірна, але, як завжди, це простіше сказати, ніж зробити.

1.2. Еталонна архітектура хмарних обчислень *(Cloud Computing Reference Architecture)*

Мета полягає у визначенні нейтральної референтної архітектури (reference archtecture), що відповідає визначенню хмарних обчислень NIST – NIST Definition of Cloud Computing, яка:

а) представляє:

- три сервісні моделі (Програмне забезпечення як послуга – Software as a Service, SaaS / Платформа як послуга – Platform as a service, Paa) / Інфраструктура як послуга – Infrastructure as a Service, IaaS);
- чотири моделі розгортання (Приватна хмара – Private cloud, Загальна хмара – Community cloud, Публічна хмара – Public cloud, Гібридна хмара – Hybrid cloud);
- п’ять основних характеристик (on-demand self-service/broad network access/resource pooling/rapid elasticity/measured service);

б) зв’язує різні хмарні сервіси й відображає їх на загальну модель;

в) діє як дорожня карта (роадмеп) індустрії IT для розуміння, вибору, проєктування та/або розгортання хмарної інфраструктури.

1.2.1. Референтна архітектура хмарних обчислень – високорівневий погляд

Референтна архітектура хмарних обчислень NIST містить п’ять головних діючих суб’єктів – акторів (actors).

Кожен актор виступає в ролі (role) і виконує дії (activities) та функції (functions). Референтна архітектура представлена як послідовні діаграми зі зростаючим рівнем деталізації.

Серед представлених п’яти акторів хмарний брокер - опціональний, тому що хмарні споживачі можуть отримувати послуги безпосередньо від хмарного провайдера (cloud provider). Зараз ми з вами розглянемо табл. 1, у якій міститься стисла характеристика архітектури хмарних обчислень.

Таблиця 1

Референтна архітектура хмарних обчислень

Актор	Визначення
Хмарний споживач *Cloud Consumer*	Особа або організація, що підтримує бізнес-відносини й використовує послуги хмарних провайдерів
Хмарний провайдер *Cloud Provider*	Особа, організація або сутність, що відповідає за доступність хмарної послуги для хмарних споживачів
Хмарний аудитор *Cloud Auditor*	Учасник, який може виконувати незалежну оцінку (assessment) хмарних послуг, обслуговування інформаційних систем, продуктивності й безпеки реалізації хмари
Хмарний брокер *Cloud Broker*	Сутність, що керує використанням, продуктивністю і наданням хмарних послуг, а також встановлює відносини між хмарними провайдерами і хмарними споживачами
Хмарний оператор зв'язку *Cloud Carrier*	Посередник, який надає послуги підключення та транспорт (послуги зв'язку) <доставки> хмарних послуг від хмарних провайдерів хмарним споживачам

1.2.2. Приклади сценаріїв використання

Сценарій 1. Хмарний споживач може запросити послугу (сервіс) у хмарного брокера замість прямого контактування з хмарним провайдером. Хмарний брокер може створити новий сервіс, комбінуючи набір сервісів або розширюючи наявний сервіс.

У цьому прикладі хмарний провайдер невидимий хмарному споживачеві.

Сценарій 1 – участь хмарного брокера у взаємодії споживача з провайдером.

Сценарій 2. Хмарний оператор зв'язку надає послуги підключення та транспорт <доставки> хмарних послуг від хмарного провайдера хмарному споживачеві. Хмарний провайдер встановлює угоду про рівень обслуговування SLA із хмарним оператором і може запитувати виділені та захищені з'єднання.

Сценарій 2 – участь хмарного оператора в наданні провайдером послуг.

Сценарій 3. Хмарний аудитор проводить незалежну оцінку обслуговування й безпеки реалізації хмарної послуги.

Сценарій 3 – участь хмарного аудитора в оцінці надання провайдером послуг.

1.2.3. Актори, їх ролі та функції

Хмарний споживач – о*соба або організація, що підтримує бізнес-відносини й використовує послуги хмарних провайдерів.*

Хмарні споживачі класифікуються за трьома групами (табл. 2), заснованим на їх додатках / різних сценаріях використання (рис. 1).

Таблиця 2

Актори, їх ролі та функції

Тип споживача	Основна діяльність (активності)	Приклади користувачів
SaaS	Використовує додатки / сервіси для автоматизації бізнес-процесів	Бізнес-користувачі, адміністратори додатків
PaaS	Розробляє, тестує, розгортає і управляє програмами, розгорнутими в хмарному оточенні	Розробники додатків, тестувальники, адміністратори
IaaS	Створює / встановлює, управляє і здійснює моніторинг сервісів для управління IT-інфраструктурою	Системні розробники, адміністратори, IT-менеджери

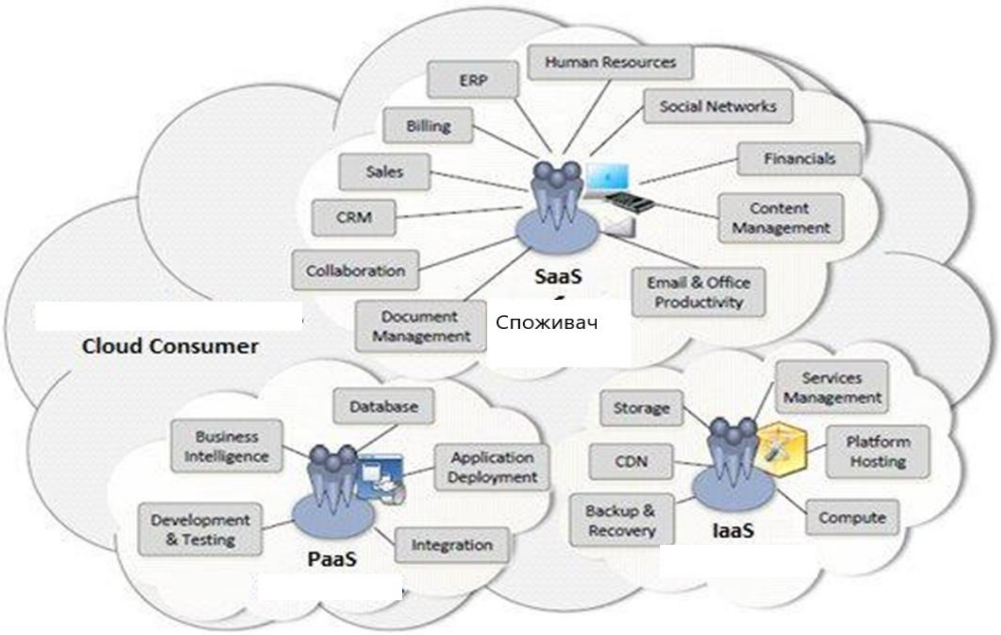


Рис. 1. Приклади сервісів, доступних хмарним споживачам

Хмарний провайдер – *особа, організація або сутність, що відповідає за доступність хмарної послуги для хмарних споживачів.*

Хмарні провайдери виконують різні завдання в різних сервісних моделях (табл. 3).

Таблиця 3

Завдання хмарних провайдерів

Тип провайдера	Основна діяльність (активності)
SaaS	Встановлює, управляє, супроводжує і підтримує програмне забезпечення <розгорнуте> у хмарній інфраструктурі
PaaS	Надає і управляє хмарною інфраструктурою і сполучним програмним забезпеченням (middleware) платформи для споживачів; надає інструменти розробки, розгортання й адміністрування споживачам платформи
IaaS	Надає і управляє фізичними обчислювальними потужностями (processing), системами зберігання, мережами і хостинг-оточенням, а також хмарною інфраструктурою для IaaS-споживачів

Діяльність хмарних провайдерів обговорюється більш детально з погляду (в архітектурній перспективі) розгортання сервісів (service deployment), оркестрації сервісів (service orchestration), хмарного сервіс-менеджменту (cloud service-management), безпеки (security) і приватності (privacy). Діяльність хмарних провайдерів обговорюється більш детально з позиції «хмарний провайдер – високорівневий погляд».

1.2.4. Розгортання сервісів (Service Deployment)

Хмарна система може функціонувати в одній із чотирьох моделей:

- Приватна хмара (Private cloud) – хмарна інфраструктура функціонує повністю з метою обслуговування однієї організації. Інфраструктура може управлятися самою організацією або третьою стороною і може існувати як на стороні споживача (on premise), так і у зовнішнього провайдера (off premise).
- Хмара спільноти, або загальна хмара (Community cloud) – хмарна інфраструктура використовується спільно декількома організаціями та підтримує обмежене співтовариство, що розділяють загальні принципи (наприклад, місію, вимоги до безпеки, політики, вимоги до відповідності <регламентам і керівним документам>). Така хмарна інфраструктура може управлятися самими організаціями або третьою стороною і може існувати як на стороні споживача (on premise), так і у зовнішнього провайдера (off premise).
- Публічна хмара (Public cloud) – хмарна інфраструктура створена як загальнодоступна або доступної для великої групи споживачів <які не пов’язані загальними інтересами, але, наприклад, належать до однієї галузі діяльності>. Така інфраструктура перебуває у володінні організації, що продає відповідні хмарні послуги / надає хмарні сервіси.
- Гібридна хмара (Hybrid cloud) – хмарна інфраструктура є композицією (поєднанням) двох і більше хмар (приватних, загальних або публічних), що залишаються унікальними сутностями, але об’єднані разом стандартизованими або пропрієтарними технологіями, що забезпечують портованість даних і додатків <між такими хмарами>

(наприклад, такими технологіями, як пакетна передача <даних> для балансу завантаження між хмарами).

1.2.4. Оркестрація сервісів (Service Orchestration)

Передбачає звернення до неї для систематизації, координації і управління хмарною інфраструктурою, призначеною для надання різних хмарних послуг, що забезпечують узгодження бізнес- та IT-вимог.

Узагальнене хмарне середовище містить три концептуальні рівня:

- ***Рівень сервісу (Service Layer)*** – визначає базові сервіси, що надаються хмарним провайдером.

SaaS: розгортаються додатки, припускається робота з ними за допомогою звернення до хмари із <спеціально призначених> програмних клієнтів і інших програм, орієнтованих на кінцевих користувачів.

PaaS: сервіси для споживачів, призначені для розробки й розгортання додатків на хмарній інфраструктурі, що містять контейнери додатків, інструменти розробки додатків, системи управління базами даних тощо.

IaaS: надання обчислювальних потужностей, систем зберігання, мережевих та інших фундаментальних обчислювальних ресурсів, поверх яких хмарні споживачі можуть розгортати та запускати програми на хмарній інфраструктурі.

- ***Рівень абстракції і контролю ресурсів (Resource Abstraction and Control Level)*** – призначає / надає елементи програмного забезпечення, такі як гіпервізор, віртуальні сховища даних, і підтримує програмні компоненти, що використовуються для реалізації хмарної інфраструктури, поверх якої може бути визначений / встановлений хмарний сервіс. Також призначає / надає асоційовані функціональні модулі, які керують абстрагованими <таким чином> ресурсами для забезпечення ефективного, безпечного й надійного використання.

Незважаючи на те що на цьому рівні широко застосовується технологія віртуальних машин, припускаються й інші значення

<поняття> необхідної абстракції програмного забезпечення. Цей рівень забезпечує «готовність до хмари» (cloud readiness), яка визначається п’ятьма характеристиками, представленими у «Визначенні хмарних обчислень», розробленому NIST.

- ***Рівень фізичних ресурсів (Physical Resource Level)*** – ***охоплює всі фізичні ресурси.***

Комп’ютерне обладнання (Hardware): комп’ютери (CPU, пам’ять), мережі (роутери, мережеві екрани, свічі, мережеві канали та інтерфейси), компоненти зберігання (жорсткі диски) та інші фізичні елементи обчислювальної інфраструктури.

Інженерна інфраструктура (Facilities): системи кондиціонування (HVAC), харчування, комунікацій та інші елементи фізичної майданчики розгортання комп’ютерного обладнання.

1.2.5. Хмарний сервіс-менеджмент (Cloud Service Management)

Охоплює всі пов’язані із сервісом функції, необхідні для управління й функціонування сервісів, потрібних або пропонованих хмарним споживачам.

Хмарний провайдер виконує ці функції для підтримки управління хмарними сервісами:

- підтримка бізнесу (Business Support);
- провіжинінг / конфігурація (Provisioning/Configuration);
- портованість / інтероперабельність (Portability/Interoperability).

Підтримка бізнесу (Business Support) – призначає / надає набір сервісів, пов’язаних із бізнесом і орієнтованих на роботу з клієнтами, і підтримує процесами, як-от розміщення замовлень, обробка рахунків і збір платежів. Також містить компоненти, що використовуються для виконання бізнес-операцій, видимих бізнес-клієнтам.

- Управління замовниками (Customer Management) – керування обліковими записами користувачів, відкриття / закриття / припинення дії облікових записів, управління профілями користувачів, управління взаємодією із замовниками (customer relationship) на основі надання контактів і вирішення питань і проблем замовників тощо.

- Управління контрактами (Contract Management) – управління сервісними контрактами, висновок / закриття / припинення дії контрактів тощо.
- Управління постачанням (Inventory Management) – налаштування й управління каталогом послуг і т. ін.
- Бухгалтерія і нарахування (Accounting and Billing) – управління платіжною інформацією щодо замовників, відправка рахунків на оплату, обробка отримання платежів, відстеження рахунків тощо.
- Звітність та аудит (Reporting and Audit) – моніторинг дій користувачів, генерація звітів і т. ін.
- Ціноутворення та тарифікація (Pricing and Rating) – оцінка хмарних послуг й визначення цін, обробка спеціальних пропозицій та правил ціноутворення, заснованих на профілі користувача тощо.
- Провіжинінг / конфігурування (Provisioning/Configuration) – охоплює всі пов'язані із сервісом функції, необхідні для управління та функціонування сервісів, потрібних або пропонованих хмарним споживачам.
- Швидкий провіжинінг (Rapid Provisioning) – автоматичне розгортання хмарних систем на основі запитів сервісів / ресурсів / можливостей.
- Модифікація ресурсів (Resource Change) – налаштування конфігурацій / призначення ресурсів для відновлення, апгрейду та підключення нових вузлів у хмару.
- Моніторинг та звітність (Monitoring and Reporting) – виявлення та моніторинг віртуальних ресурсів, моніторинг функціонування (дій та подій) хмари й генерація звітів про продуктивність.
- Вимірювання <показників> (Metering) – надання можливостей кількісних вимірів на рівні абстракції відповідному типу сервісу (наприклад, засобів зберігання, обробки, пропускної спроможності і активних облікових записів користувачів).
- Управління рівнем обслуговування (SLA Management) – визначення параметрів SLA контракту (схема з параметрами якості сервісу – QoS), моніторинг <виконання> SLA, застосування SLA відповідно до заданих політик.

Портованість / інтероперабельність (Portability/Interoperability)

Портованість:

1. Можливість перенесення даних з однієї системи в іншу без необхідності повторного створення або введення описів даних або значної модифікації додатків.

2. Можливість програмного забезпечення або системи виконуватися на більш ніж одному типі або потужності комп'ютера під більш ніж однією операційною системою.

○ Інтероперабельність – можливість взаємодіяти, виконувати програми або передавати дані між різними функціональними одиницями відповідно до заданих умов.

Хмарні провайдери повинні надавати механізми для:

- портованості даних (Data Portability);
- копіювання даних «з» / «в» (Copy data to-from) – копіювання даних з/в хмару;
- пакетного перенесення даних (Bulk data transfer) – використання диска для пакетного перенесення;
- інтероперабельності сервісів (Service Interoperability) – дає змогу хмарним споживачам використовувати їх дані й сервіси у безлічі хмарних провайдерів, використовуючи уніфіковані та розширені інтерфейси управління;
- портованості систем (System Portability) – перенесення образів віртуальних машин (VM images migration), міграції повністю зупиненого примірника або образу віртуальної машини від одного провайдера до іншого;
- міграції додатків / сервісів (Application / Service migration) – міграції додатка / сервісу або поточного утримання від одного сервіс-провайдера до іншого.

Безпека (Security)

- *Аутентифікація і ваторизація (Authentication and Authorization)* – аутентифікація і авторизація хмарних споживачів із використанням попередньо створеного мандата доступу.
- *Доступність (Availability)* – налаштування конфігурацій / призначення ресурсів для відновлення, апгрейда і підключення нових вузлів у хмару.

- *Конфіденційність (Confidentiality)* – виявлення й моніторинг віртуальних ресурсів, моніторинг функціонування (дій і подій) хмари та генерація звітів про продуктивність.
- *Управління ідентіфікацією (Identity management)* – надання можливостей кількісних вимірювань на рівні абстракції, що відповідає типу сервісу (наприклад, засобів зберігання, обробки, пропускної здатності й активних облікових записів користувачів).
- *Моніторинг безпеки й обробка інцидентів (Security monitoring & Incident Response)* – визначення параметрів SLA-контракту (схема з параметрами якості сервісу – QoS), моніторинг <виконання> SLA, застосування SLA відповідно до заданих політик.
- *Керування політиками безпеки (Security policy management)* – генерація / застосування / аудит / оновлення політик безпеки для користувачів, які отримують доступ до хмар.

Конфіденційність – захист приватності (Privacy)

Захищає достовірні, належні <за призначенням> і відповідні <політикам і правилам> збір, обробку, передачу, використання і зберігання в хмарі персональних даних та інформації, що дає змогу ідентифікувати особу.

Хмарний аудитор – *учасник, який може виконує незалежну оцінку (assessment) хмарних послуг, обслуговування інформаційних систем, продуктивності й безпеки реалізації хмари.*

Хмарний аудитор може давати оцінку сервісів, що надаються хмарним провайдером, у термінах *контролю безпеки (security control), дотримання приватності (privacy impact), продуктивності (performance) тощо*.

Для аудиту безпеки хмарний аудитор може проводити оцінку контролю безпеки інформаційної системи для визначення меж, для яких контроль виконується відповідним чином, у яких система функціонує за призначенням і виробляє бажаний результат відповідно до вимог безпеки, що висуваються до системи.

Хмарний брокер – *сутність, що керує використанням, продуктивністю й наданням хмарних послуг, а також встановлює відносини між хмарними провайдерами і хмарними споживачами*.

У міру еволюції хмарних обчислень інтеграція хмарних сервісів може виявитися для хмарних споживачів занадто складною для управління.

Основні послуги, що надаються хмарним брокером:

- Сервісне посередництво (Service Intermediation) – хмарний брокер розширює заданий сервіс, покращуючи його окремі можливості й надаючи додаткові сервіси хмарним споживачам.
- Агрегування сервісів (Service Aggregation) – хмарний брокер комбінує і інтегрує сервіси в один і більше сервісів. Брокер забезпечуватиме інтеграцію даних і їх безпечне перенесення між хмарним споживачем і хмарними провайдерами.
- Арбітраж сервісів* (Service Arbitrage) – арбітраж сервісів аналогічний агрегуванню сервісів, але відрізняється тим, що агрегований сервіс не модифікується. Арбітраж сервісів забезпечує хмарному брокеру гнучкий і вигідний вибір <сервісів>. Наприклад, хмарний брокер може використовувати скоринговий сервіс і формувати найкращий портфель <сервісів> для пропозиції хмарним споживачам.

* – *аналогічно поняттю арбітражу цінних паперів.*

Хмарний оператор зв'язку – *посередник, який надає послуги підключення та транспорт (послуги зв'язку) <доставки> хмарних послуг від хмарних провайдерів хмарним споживачам через мережеві, телекомунікаційні та інші пристрої доступу*.

Приклад: до пристроїв мережевого доступу належать комп'ютери, ноутбуки, мобільні телефони, мобільні пристрої доступу в інтернет (mobile internet devices – MID) і т. ін. Доставлення <послуг та пристроїв> може забезпечуватися мережевими й телекомунікаційними операторами, а також транспортними агентами.

Транспортний агент (transport agent) – бізнес-організація, що забезпечує фізичне транспортування засобів зберігання інформації (storage media), як-от жорсткі диски підвищеної ємності.

Хмарний провайдер повинен укладати угоду про рівень обслуговування (SLA) із хмарним оператором зв’язку для забезпечення відповідного рівня сервісу. У загальному випадку до хмарного оператора зв’язку можуть висуватися вимоги щодо надання виділеного та захищеного з’єднання.

1.3. Хмарні обчислення: огляд і рекомендації

1.3.1. Загальне середовище хмарних обчислень

Рекомендації Національного інституту стандартів і технологій (США) Special Publication 800-146 (Draft).

Потрібно розуміти, що термін «хмарні обчислення» охоплює велике різноманіття як систем і технологій, так і моделей розгортання, <надання> послуг і ведення бізнесу.

Ряд тверджень, які іноді робляться щодо хмарних обчислень, наприклад, що вони «масштабуються» або що вони конвертують капітальні витрати (CAPEX) в операційні (OPEX), є правильними тільки для певних видів хмарних систем.

Мета цього розділу полягає в тому, щоб чітко описати поділ систем хмарних обчислень або хмарних систем на п’ять значущих сценаріїв і для кожного із цих сценаріїв пояснити основні аспекти (зокрема, масштабованість), що потребують уваги або викликають суперечки, а також як ці аспекти адресуються щодо кожного із сценаріїв (2).

Цей розділ представляє фізичний – мережевий погляд на те, як передплатники (хмарні споживачі – замовники) сервісів підключені до хмарних систем. Розуміння хмарного і складових частин традиційного «стека» програмного забезпечення, доступних для передплатників, є не менш важливим і розглядається в розділах 5, 6 і 7 цього (оригінального) документа.

Як це передбачається у визначенні хмарних обчислень NIST, хмарна система є колекцією ресурсів, доступних через мережу для замовників (тобто хмарних передплатників – хмарних споживачів). У загальному випадку хмарна система і її передплатники застосовують модель клієнт – сервер, яка передбачає, що передплатники

відправляють по мережі повідомлення серверним комп’ютерам, які у відповідь на отримані повідомлення виконують відповідну роботу (рис. 2).

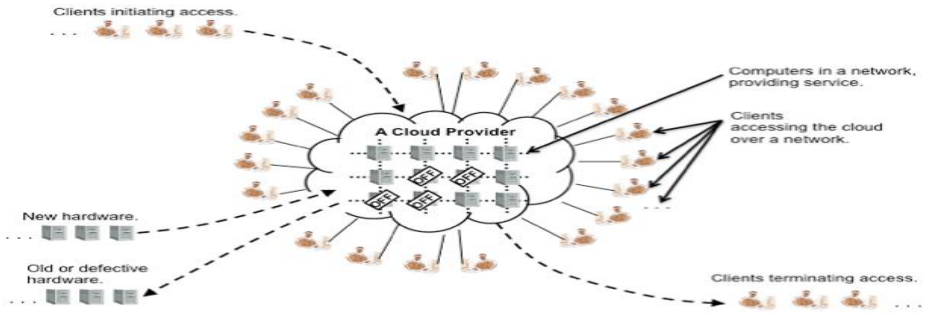


Рис. 2. Загальний погляд на хмару і передплатників

Рис. 2 дає загальний погляд на хмару та її клієнтів: хмарні обчислювальні ресурси описані як комплекс взаємопов’язаних* комп’ютерних систем, доступ до яких клієнти здійснюють у мережі. Як показано на рисунку, можуть з’являтися нові клієнти, старі клієнти можуть йти, у різні моменти часу кількість клієнтів буде різним. Подібним чином хмара підтримує пул апаратного забезпечення, яким керує для максимізації (збільшення продуктивності й рівня) сервісу та мінімізації витрат. Для підтримки високої доступності сервісів, незважаючи на очікувані відмови й витікання терміну життя компонентів, хмара у міру виникнення потреби** підключає нові та виводить з експлуатації старі апаратні компоненти або ті, що не працюють. Хмара ефективно управляє пулом апаратних ресурсів для оптимального з погляду витрат надання сервісів. Одна зі стратегій <такого управління> полягає в тому, що хмарний провайдер вимикає невикористовувані компоненти на період скорочення потреб передплатників. З позицій управління споживаною потужністю або поновлення апаратного забезпечення міграція робочих навантажень (workloads) замовників з одного фізичного комп’ютера на інший є ключовою стратегією, що дає змогу провайдеру оновлювати апаратне забезпечення й консолідувати робочі навантаження без заподіяння незручностей передплатникам.

* – у цьому випадку grid логічно перекласти саме так, тому що хмарні обчислення не завжди припускають використання Grid-технологій. Lingvo Computers дає такі визначення:

• Grid – архітектура, концепція, технологія Grid – орієнтований на віртуалізацію обчислень спосіб організації обчислювального процесу, коли частини завдання розподіляються по всім вільним ресурсам мережі.

• Grid computing – мережеві (колективні, паралельні, розподілені) обчислення, «решітки» обчислювальних ресурсів, grid – «обчислення» – термін належить переважно до архітектури глобальних, регіональних і відомчих комп'ютерних мереж, передбачає використання вільних на цей час ресурсів мережі для вирішення завдань, занадто складних для окремо взятого комп'ютера, і потребує спеціального ПО.

Зі свого боку, хмарні технології безпосередньо не мають на увазі (хоча і не заперечують) використання розподілу виконання частин завдання за різними вузлів «решітки», що являє собою розподілені обчислювальні ресурси, пов'язані особливим чином для виконання фактично масово-паралельних обчислень.

** – *напевно, ви звернули увагу, що в пропонованих перекладах as needed і on demand перекладаються не «на вимогу», а «у разі потреби»*.

Такий вибір обумовлений більш м'яким і певною мірою превентивним для ряду випадків сенсом « у разі потреби». Вимога споживача, яка надійшла в тій чи іншій формі, може не завжди відповідати узгодженим рівнем сервісу (SLA), а необхідність може бути ідентифікована провайдером сервісу взагалі без повідомлення про це клієнта.

Наприклад, необхідність, або потреба, виявляється на основі аналізу статистики використання пулу ресурсів клієнтами і прогнозу на подальше зростання утилізації ресурсів, пов'язаного як зі збільшенням потреб наявних клієнтів у ресурсах, так і зі збільшенням числа клієнтів (екстенсивний або кількісний розвиток) та/або розширенням портфеля наданих послуг того чи іншого типу (інтенсивний або якісний розвиток).

Рис. 1 представляє лише невелику частину загальних тверджень (міркувань, аспектів), характерних для хмарних обчислень.

Організації припускають, що використання хмарних обчислень має відображати такі загальні положення (представлені нижче). Водночас багато тверджень, сформульованих щодо хмарних обчислень (наприклад, що хмари можуть масштабуватися для дуже великих робочих навантажень або їх застосування дає змогу замінювати капітальні витрати операційними), є коректними тільки для певних типів хмар.

Для уникнення плутанини цей документ явно кваліфікує кожне з таких положень для кожного типу хмари, де таке положення застосовується, тобто кожне твердження має межі <застосовності> змісту (scope). Такі кордони, що використовуються в документі, наведені в табл. 4.

Табл. 4 містить сценарії – варіанти меж застосування тверджень, які зберігаються стосовно хмар (Scope Modifiers for Statements Asserted About Clouds).

Таблиця 4

Сценарії – варіанти меж застосування тверджень відносно хмар

Межі застосування – сценарії	**Опис меж застосування**
Загальний випадок, тобто всі варіанти хмар (General)	Застосовується для всіх моделей розгортання хмар
Власне приватна хмара (On-site-private)	Застосовується до приватних хмар, реалізованих на майданчику замовника (тобто повністю контрольовані замовником)
Приватна хмара на аутсорсингу (Oursourced-private)	Застосовується до приватних хмар, розміщених на зовнішньому хостингу, який обслуговується аутсорсинговою компанією
Власне хмара спільноти (On-site-community)	Застосовується до хмар спільнот, реалізованих на майданчику замовників, складових співтовариства
Хмара спільноти на аутсорсингу (Outsourced-community)	Застосовується до хмар спільнот, чиї сервери і інше апаратне забезпечення (наприклад, системи зберігання) розміщені на зовнішньому хостингу, який обслуговується аутсорсинговою компанією
Публічна хмара (Public)	Застосовується до публічних хмар

Вище розглянуто кожен із представлених сценаріїв меж застосування.

Такі твердження є загальними щодо своїх кордонів застосовності, тобто коректні незалежно від моделі розгортання або сервісної моделі (моделі надання послуг):

- ***Залежність від мережі (Network dependency) – загальний випадок.*** Передплатники, будучи клієнтами (з позиції бізнесу), потребують на працюючому і захищеному сервері доступ до хмари. Якщо мережа ненадійна (reliable у всіх сенсах – стійкості доступу й захищеності), хмара не вважатиметься надійною з позиції передплатника.
- ***Передплатники повинні володіти навичками в IT (Subscribers still need IT skills) – загальний випадок***. Експлуатуючи <на своєму майданчику> серверні комп'ютери, провайдер може зменшити потребу організацій-передплатників в IT-персонал. Однак передплатники здійснюватимуть доступ у хмару зі своїх клієнтських систем, що потребують управління, супроводу, підтримки, забезпечення безпеки тощо
- ***Місце виконання робочих навантажень призначається динамічно і приховано від клієнтів (Workload locations are dynamically assigned and are thus hidden from clients) – загальний випадок***.

У деяких випадках (наприклад, у сервісної моделі IaaS, описаної в розділі 7 оригінального документа) робоче навантаження може виконуватися в заданому місці в певний проміжок часу перед здійсненням міграції; в інших випадках (наприклад, у сервісної моделі PaaS, описаної в розділі 6 оригінального документа) робоче навантаження може бути спочатку розподіленої сутністю, послідовно виконує операції для передплатників на потенційно різних серверах, при цьому <оброблювані> дані існують у географічно розподіленому сховищі.

- ***Ризики множинної оренди (Risks from multi-tenancy) – загальний випадок.*** Робочі навантаження різних клієнтів можуть виконуватися одночасно в тих самих хмарних системах і локальній мережі, будучи розділеними тільки за допомогою політик доступу, визначених на рівні програмного забезпечення провайдера.

Уразливості в цьому програмному забезпеченні або в політиках можуть завдати шкоди безпеці передплатників.

- ***Імпорт / експорт даних і обмеження можливості їх виконання* (Data import/export, and performance limitations) – загальний випадок.*** У силу того, що передплатники здійснюють доступ <до хмари> по мережі, необхідні часові характеристики пакетного (bulk) імпорту й експорту даних можуть перевищити можливості мережі. Крім того, робота в режимі реального часу або обробка критично важливих запитів можуть виявитися проблематичними через мережеві затримки і інші обмеження.

* – мається на увазі, що йдеться не тільки про обмеження продуктивності, але й про інші можливі обмеження, що не дозволяють забезпечити необхідні параметри масового введення / виведення даних в/з хмарну систему.

Організації, які розглядають можливість використання хмарних обчислень, повинні брати до уваги ці загальні міркування і їх можливі наслідки для своєї бізнес-моделі й досягнення довгострокових цілей (місії).

Однак недостатньо приділяти увагу тільки загальним аспектам хмарних обчислень. Хмари також описуються межами застосовності однієї або декількох характеристик, представленими в табл. 1.

Організації, які обговорюють застосування хмар, повинні детально розглядати аспекти використання різних сценаріїв (варіантів реалізації з урахуванням меж застосування) хмар, які є предметом такого обговорення. Кожна з альтернатив самостійно розглядається нижче в окремому розділі (цього і ширшого оригінального) документа, фокусуючись на заданих межах застосовності (4).

Цей документ не повторює ті чи інші фрагменти тексту. Однак для конкретних типів хмар може бути описано більше міркувань, які є значущими для кожного конкретного типу; в цьому випадку назва загальної характеристики / затвердження може використовуватися знову, але з поясненням, специфічним для конкретного типу хмари.

1.3.2. Розуміння того, хто контролює ресурси в хмарі (Understanding Who Controls Resources in a Cloud)

Іноді стверджується, що в разі порівняння використання хмар із традиційною моделлю «внутрішніх» (on premises) обчислень хмарна модель вимагає від передплатників передачі провайдеру двох важливих можливостей, які передбачають високий рівень довіри передплатника до провайдера:

- контролю (Control) – можливості вирішувати, хто і що може отримувати доступ до даних і програм передплатника, і виконувати <ті чи інші> дії (як-от стирання даних або відключення від мережі), які повинні бути зроблені, але без додаткових дій, які, зі свого боку, можуть не відповідати намірам передплатника (наприклад, передплатник запитує стирання об'єктів даних, що мало б супроводжуватися неявним їх копіюванням з ініціативи провайдера);
- видимості, явності дій (Visibility) – можливості здійснення моніторингу статусу програм і даних передплатника і того, як до них здійснюється доступ.

Однак кордони контролю й видимості, які потрібно передати провайдеру з боку передплатників, залежать від багатьох факторів, включно з фізичним володінням і можливістю конфігурації (з високим рівнем довіри) механізмів захисту кордонів доступу до обчислювальних ресурсів передплатників.

Цей документ використовує концепцію кордонів доступу (access boundaries) для структурування й характеристики різних моделей розгортання хмар.

Рис. 3 ілюструє ключову концепцію комп'ютерної безпеки*, пов'язану з кордонами і контролем – периметр безпеки (security perimeter).

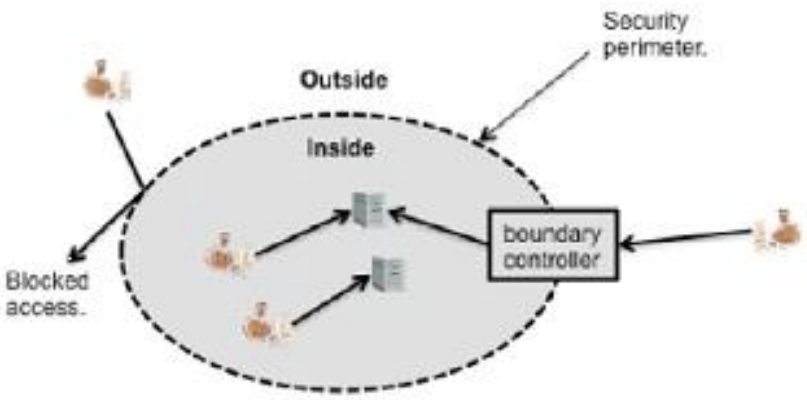


Рис. 3. Периметр безпеки

Як показано на рис. 3, периметр безпеки виконує роль бар'єра щодо <операцій> доступу: сутність, яка перебуває всередині периметра, може виробляти вільний доступ до ресурсів, що розташовані тільки всередині периметра; проте, по суті, вона перебуває за межами периметра й може отримувати доступ до ресурсів усередині периметра, якщо тільки це дозволено засобами контролю кордонів (контролер кордонів – boundary controller) на основі застосування відповідних політик доступу.

Незважаючи не те що ці терміни часто використовуються в обговоренні мережевих екранів і мереж, концепція периметра безпеки в дійсності є більш загальною і може використовуватися, наприклад, для опису меж між різними рівнями привілеїв виконання програмного забезпечення, тобто кордонів між додатками й операційною системою.

Саме по собі визначення периметра безпеки НЕ є адекватним механізмом забезпечення безпеки. Однак контроль периметра безпеки є важливим елементом побудови безпечних систем.

Типові контролери кордонів:

- мережеві екрани (firewalls);
- засоби блокування доступу (guards);
- віртуальні приватні мережі (VPN – Virtual Private Networks).

Організація може домогтися оцінки кількісних показників як щодо контролю використання ресурсів, так і щодо моніторингу доступу до них (5).

До того ж, якщо конфігурувати периметр безпеки, організація може адаптувати його до мінливих потреб (наприклад, блокування або дозволу протоколів, або форматів даних, виходячи із змін бізнес-умов).

Коли існують неконтрольовані шляхи <доступу> до комп'ютерних ресурсів, периметр безпеки є порушеним (в оригіналі – ослабленим, weakened) або навіть відсутнім. Поширюються бездротові комунікації, наприклад, є загрозою периметру безпеки, починаючи з того, що не може бути надійного шляху впровадження <контролера кордонів> між зовнішніми і внутрішніми сутностями.

Аналогічно багато організацій використовують мобільні пристрої, які іноді підключаються <до ресурсів>, перебуваючи всередині периметра, а іноді залишаються без <належного захисту>, наприклад під час поїздок.

Різні моделі розгортання хмар, описані у визначенні хмарних обчислень NIST, мають на увазі розміщення контрольованого передплатником периметра безпеки, а отже, рівень контролю, який передплатники можуть здійснювати щодо ресурсів, яким довіряють, хмари (тобто переданих під управління провайдера хмари з певним рівнем довіри до цього провайдера).

Визначення хмарних обчислень NIST описує чотири моделі розгортання: приватна хмара (private), хмара спільноти (community), публічна хмара (public) і гібридна хмара (hybrid). Однак кожна з моделей приватної хмари і хмари спільноти допускають два варіанти – сценарію <розгортання>, які повинні розглядатися окремо, через вплив на периметр безпеки: чи буде він власний (on-site) або на аутсорсингу (outsourced).

Гібридна модель розгортання є комбінацією інших моделей, і тому гібридне розгортання може припускати і вплив <на периметр безпеки> його елементів – «будівельних блоків», і унікальні аспекти впливу, що виникають унаслідок об'єднання безлічі систем у більш комплексні інтегровані системи.

1.3.2. Сценарій власне приватної хмари (The On-site Private Cloud Scenario)

Рис. 3 надає простий погляд на власне приватну хмару (on-site private cloud). Як показано на рисунку, периметр безпеки простягається навколо власних ресурсів передплатника і ресурсів приватної хмари. Приватна хмара може бути централізованою на одному майданчику або розподіленою між декількома майданчиками передплатника.

Периметр безпеки існуватиме тільки в тому випадку, якщо передплатник його реалізує. У разі реалізації периметр не гарантує контролю всіх ресурсів приватної хмари, але його існування дає

передплатнику можливість здійснювати контроль ресурсів, довірених власне приватною хмарою.

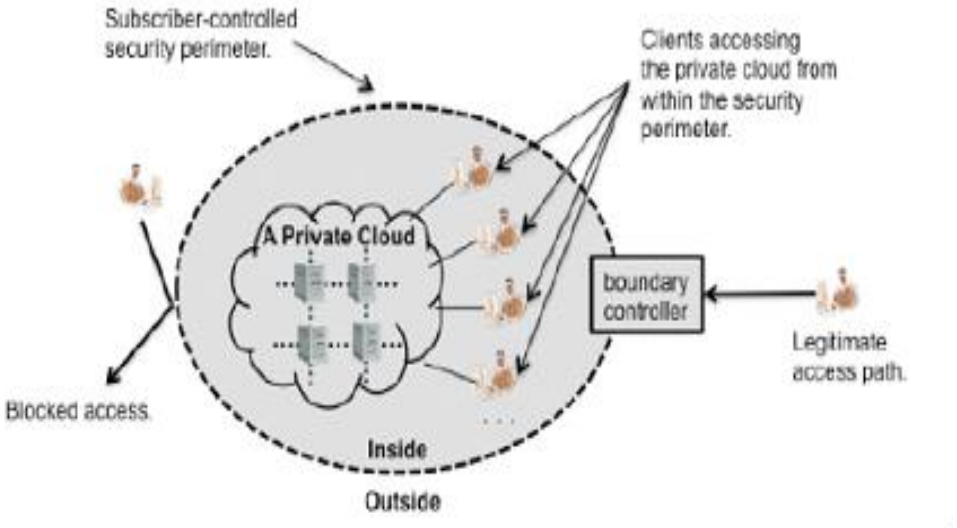


Рис. 4. Власне приватна хмара

Незважаючи на те що загальні припущення залишаються правильними для власне приватної хмари, цей сценарій дає змогу припустити й інші, у т. ч. більш детально описані аспекти, які потрібно взяти до уваги організаціям, що розглядають використання власне приватних хмар:

Залежність від мережі (Network dependency) – власне приватна хмара (on-site-private).

Відповідно до конфігурації (наприклад, один фізичний сайт, захищена мережа хмари) залежність від мережі для власне приватної хмари може обмежуватися залежністю від мережевих ресурсів, які контролює передплатник (наприклад, локальна мережа). Цей сценарій унеможливлює такі проблеми мереж великого масштабу (large-scale networks), як «затори» інтернету або комунікації з віддаленими (у сенсі remote) серверами доменних імен інтернету (Internet DNS). Крім того, якщо реалізований периметр безпеки для високих рівнів захисту, не у всіх випадках стає необхідним застосування таких механізмів захисту, як багатофакторна аутентифікація і наскрізне шифрування під час обміну всередині периметра навіть за досить обережних (що мінімізують відповідні ризики) політик безпеки.

Якщо організація-передплатник має безліч фізичних майданчиків і організовує доступ з різних майданчиків до тієї самої приватної хмари, передплатник повинен забезпечити контроль комунікацій між майданчиками (наприклад, лінії зв'язку з шифруванням) або використовувати криптографічні засоби (наприклад, VPN) поверх менш контрольованих засобів комунікацій, як-от публічний інтернет.

Обидві ці опції мають ризики мережевої доступності та безпеки приватної хмари в силу того, що існує залежність від продуктивності ресурсів (комунікаційних засобів), які перебувають за межами області, контрольованої передплатником, і тому будь-які помилки в реалізації і конфігурації криптографічних механізмів можуть дозволити отримати доступ ззовні.

Передплатники повинні володіти навичками в IT (Subscribers still need IT skills) – власне приватна хмара (on-site-private). Організації-передплатники потребуватимуть традиційних IT-навичок, необхідних для управління призначеними для користувача пристроями, з яких здійснюється доступ до приватної хмари, при цьому для них актуальні вимоги щодо наявності навичок і в частині хмарних технологій. Раніше, на етапі розгортання власне приватної хмари, організації-передплатники можуть захотіти паралельно підтримувати протягом певного проміжку часу* і хмарні, і традиційні засоби.

Крім того, можуть вимагатися нові навички для роботи з хмарами. Наприклад, організаціям, які виконують роботи, що потребують інтенсивних обчислень, може знадобитися згодом реорганізувати ці роботи таким чином, щоб вони могли використовувати високий рівень паралелізму хмарних ресурсів. Організаціям, що обробляють великі обсяги даних, зі свого боку, може знадобитися розвиток навичок використання хмарних систем зберігання (у загальному сенсі, а не тільки апаратних).

Місце виконання робочих навантажень призначається динамічно і приховане від клієнтів (Workload locations are hidden from clients) – власне приватна хмара (on-site-private). Як і в загальному випадку, приватна хмара з позиції управління апаратними ресурсами має забезпечувати можливість міграції робочих навантажень між машинами <фізичними вузлами> без заподіяння

клієнтам будь-яких незручностей, тобто навіть без необхідності клієнтам знати про це. Водночас у разі власне приватної хмари організація передплатника вибирає фізичну інфраструктуру для розгортання й експлуатації приватної хмари, а отже, визначає можливе географічне розміщення місця виконання робочих навантажень. Якщо індивідуальні клієнти не знають, де саме їх робочі навантаження виконуються в цей момент часу в термінах інфраструктури організації передплатника, сама організація має таке знання і контроль всіх робочих навантажень, яким дозволено виконуватися <у власне приватній хмарі>.

Ризики множинної оренди (Risks from multi-tenancy) – власне приватна хмара (on-site-private). Як і в загальному випадку, робочі навантаження різних клієнтів можуть виконуватися одночасно в тих самих системах і мережах, будучи розділеними тільки за допомогою політик доступу, визначених на рівні програмного забезпечення провайдера.

Уразливості в цьому програмному забезпеченні або в політиках можуть завдати шкоди безпеці організації передплатників, роблячи робочі навантаження клієнтів видимими / доступними іншим сторонам у порушення політик безпеки. Власне приватна хмара <за своєю природою> якоюсь мірою пом'якшує ці ризики, <спочатку> обмежуючи число можливих порушників; зазвичай усі клієнти є членами організації-передплатника або авторизованими партнерами або гостями (guest у термінах мережевого доступу), однак власне приватна хмара залишається вразливою до атак, вироблених авторизованими, що мають злий умисел, інсайдерами.

Різні організаційні функції, як-от платіжні відомості, сховища персональних даних або створена інтелектуальна власність, можуть бути схильні до тих же вразливостей у безпеці, які можуть дозволити надати доступ користувачам, неавторизованим для доступу до певних класів даних, і які можуть розкрити дані <зовні> з власне приватної хмари.

Імпорт / експорт даних і обмеження можливості їх виконання (Data import/export, and performance limitations) – власне приватна хмара (on-site-private). Як і в загальному випадку, необхідні операції пакетного (bulk) імпорту й експорту обмежені пропускною

спроможністю мережі, а робота в режимі реального часу або обробка критично важливих запитів можуть виявитися проблематичними в силу мережевих обмежень (у тому числі затримок під час передачі даних по мережі – network latency). В окремому випадку, якщо у передплатника є тільки одна площадка, для якої потрібен доступ до власне приватної хмари, передплатник може бути забезпечений доступом до локальної мережі, що надає вищу продуктивність, ніж та, яка може бути досягнута в разі доступу через глобальну мережу.

Потенційно більш суворий захист від зовнішніх загроз (Potentially strong security from external threats) – власне приватна хмара (on-site-private). У разі використання власне приватної хмари передплатник має можливість реалізації відповідного <більш> суворого периметра безпеки для захисту ресурсів приватної хмари від зовнішніх загроз, ніж це може бути досягнуто для ресурсів, що не належать до хмари. Для не настільки значущих даних (low-impact data – дані, ризик витоку яких може призвести лише до незначних наслідків або взагалі не має значення) і їх обробки периметр безпеки може містити <лише стандартні> набори правил, що визначаються комерційними мережевими екранами і VPN.

Для більш значущих даних периметр безпеки може створюватися із застосуванням більш суворих обмежувальних політик мережевих екранів, багатофакторної аутентифікації і навіть його фізичного ізолювання.

Значні первинні інвестиції в побудову й міграцію хмари (Significant-to-high up-front costs to migrate into the cloud) – власне приватна хмара (on-site-private). Власне приватна хмара потребує встановлення програмного забезпечення для управління хмарою на обчислювальних системах організації-передплатника.

Якщо планується, що хмара підтримуватиме робочі навантаження, які передбачають інтенсивні обчислення або обробку даних, <відповідне> програмне забезпечення повинно бути встановлено на великій кількості найпоширеніших (більш дешевих – commodity) систем або на меншій кількості високопродуктивних (більш дорогих) систем. Встановлення хмарного програмного забезпечення й управління такою інсталяцією призведе до значних первинних вкладень (up-front costs), навіть якщо хмарне програмне

забезпечення є вільним і навіть якщо більша частина <планованого до використання в хмарі> обладнання вже існує в організації.

1.4. Контрольні запитання до теми 1

1. Які основні переваги і недоліки хмарних систем?
2. Що розуміється під grid-обчисленнями?
3. Назвіть основні переваги хмарних обчислень.
4. Назвіть основні недоліки хмарних обчислень.
5. Дайте визначення хмарних обчислень.
6. Які види хмар існують?
7. Що надають постачальники послуг IaaS?
8. Що ховається під абревіатурою PaaS?
9. Що ховається під абревіатурою SaaS?
10. Назвіть основні переваги SaaS для клієнтів.
11. Розкажіть про особливості публічних, приватних, гібридних хмар.
12. Назвіть основні перешкоди розвитку хмарних технологій.
13. Питання безпеки хмар.
14. Які концепції масштабування, розгортання, резервного копіювання в контексті хмарної інфраструктури?
15. Що виконує хмарний брокер (Cloud Broker)?
16. Що виконує хмарний оператор зв’язку (Cloud Carrier)?
17. Хмарний сервіс-менеджмент – безпека (Security) охоплює…
18. Хмарний сервіс-менеджмент – підтримка бізнесу (Business Support) призначає…
19. Хмарний сервіс-менеджмент – провіжинінг (Provisioning/Configuration) охорплює…
20. Міф щодо надійності хмарних середовищ.
21. Міф про загальне переході в хмарні середовища.
22. Міф про зниження витрат і навантаження.

Тема 2. Технології віртуалізації

2.1. Базові поняття

Однією з найбільш істотних технологічних новацій, що лежать в основі хмарних обчислень, є технології віртуалізації.

Віртуалізація – надання набору обчислювальних ресурсів або їх логічного об'єднання, абстрагованих від апаратної реалізації, що забезпечує при цьому логічну ізоляцію обчислювальних процесів, які виконуються на одному фізичному ресурсі.

Віртуалізація – це перехід від технології, що лежить в основі консолідації серверів і центрів обробки даних, до основних компонентів для створення гнучкої (надається на вимогу) інфраструктури.

Основні поняття технології віртуалізації

Віртуальна машина – ізольований програмний контейнер, який працює з власною ОС і додатками, подібно до фізичного комп'ютера. Віртуальна машина діє так само, як фізичний комп'ютер, і містить власні віртуальні ОЗУ, жорсткий диск і мережевий адаптер.

Віртуальна машина являє собою програмний контейнер, що зв'язує (або «інкапсулює») повний комплект віртуальних апаратних ресурсів, а також ОС і все її застосування в програмному пакеті.

Віртуалізація – це перехід від технології, що лежить в основі консолідації серверів і центрів оброблення даних, до основних компонентів для створення гнучкої інфраструктури, що надається на вимогу.

Основними особливостями віртуальних машин є:

- сумісність (віртуальні машини сумісні з усіма стандартними комп'ютерами, віртуальна машина працює під управлінням власної гостьової операційної системи і має власні додатки);
- ізольованість (віртуальні машини повністю ізольовані один від одного, як якщо б вони були фізичними комп'ютерами);
- інкапсуляція (віртуальні машини повністю інкапсулюють обчислювальну середу);

- віртуальні машини повністю незалежні від базового фізичного обладнання, на якому вони працюють.

Хостова операційна система – це операційна система, встановлена на реальне обладнання. У рамках цієї операційної системи встановлюється програмне забезпечення віртуалізації як звичайна програма.

Емулятор віртуальної машини – це програмне забезпечення, яке встановлюється на хостову операційну систему та складається з монітора віртуальних машин і графічної оболонки.

Монітор віртуальних машин є програмою, яка забезпечує всі взаємодії між віртуальним і реальним обладнанням, що підтримує роботу однієї або декількох створених віртуальних машин і встановлених гостьових операційних систем.

Графічна оболонка забезпечує взаємодію користувача з додатком віртуальної машини, даючи змогу налаштовувати створювані віртуальні машини під свої потреби й управляти її роботою.

Гостьова операційна система – це операційна система, яка встановлюється на створену віртуальну машину. Як гостьові операційні системи можна використовувати Window, Linux та ін.

Із використанням технології віртуалізації отримують ***ієрархічну структуру взаємодії віртуальних ЕОМ і реальної апаратури.***

На нижньому шарі цієї ієрархії перебуває реальне обладнання, управління яким розподіляється між хостовою операційною системою і емулятором віртуальних машин.

- ***Хостова операційна система і емулятор*** розподіляють між собою ресурси реальної ЕОМ і складають другий рівень ієрархії.
- ***Хостова операційна система*** займається управлінням працюючих на ній додатків і розподілом між ними ресурсів реальної ЕОМ.
- ***Емулятор віртуальних машин*** управляє віртуальними машинами щодо встановлених на них гостьових операційних систем, розподіляючи між ними ресурси реальної ЕОМ так, щоб у користувачів створювалося враження роботи на реальному обладнанні.

Частково розподіл ресурсів між віртуальними машинами можна налаштувати на етапі конфігурації віртуальних машин, вказавши обсяг

оперативної пам’яті, розмір жорсткого диска, кількість віртуальних процесорів, канали зв’язку, що віртуалізуються, та інші параметри.

Гостьові операційні системи управляють роботою своїх додатків у рамках виділених емулятором ресурсів.

Розглядаючи технологію віртуалізації потрібно вивчити ***емулятор віртуальних машин, а саме монітор віртуальних машин***, що є базовою частиною технології віртуалізації.

Усі існуючі монітори віртуальних машин можна розділити на чотири види, що використовують таку віртуалізацію:

- апаратну,
- апаратно-програмну,
- програмну,
- доменну.

Це поділ умовний, оскільки більшість моніторів віртуальних машин використовують як програмну, так і апаратну віртуалізацію, оскільки програмна віртуалізація вимоглива до ресурсів, а апаратна віртуалізація обмежується вузьким колом обладнання, що підтримує будь-який вид моніторів віртуальних машин.

Доменна віртуалізація ґрунтується на логічному розподілі ресурсів на окремі частини (домени). Здебільшого вона використовується в мейнфреймах. Цей тип віртуалізації з’явився першим і використовувався для розподілу ресурсів великих ЕОМ між окремими користувачами.

Монітори віртуальних машин, що використовують технологію апаратно-програмної віртуалізації, частину інструкцій виконують, безпосередньо на самому процесорі, а частину – емулюють.

Існують три типи **програмної емуляції інструкцій хостової ЕОМ:**

- ***повна емуляція інструкцій,***
- ***вибіркова емуляція інструкцій,***
- ***емуляція API.***

У разі використання ***повної емуляції*** інструкції інтерпретуються і перетворюються в інструкції, що сприймаються реальним процесором.

У цьому випадку з’являється можливість створювати віртуальні машини, що імітують роботу апаратури, несумісної з архітектурою з реальної ЕОМ.

Наприклад, можна запускати віртуальну машину, що імітує роботу процесора з RISC-архітектурою, на реальній ЕОМ з процесором CISC-архітектури. Це можливо завдяки тому, що емуляція ведеться на рівні базових арифметико-логічних інструкцій, присутніх практично в будь-якому процесорі.

Інтерпретація кожної інструкції призводить до значної витрати ресурсів реальної ЕОМ і знижує швидкодію програм, які потребують гостьової операційної системи. Сучасні сервери і персональні ЕОМ мають більшу продуктивність. Тому віртуалізація з використанням інтерпретації інструкцій набуває популярності. Представниками цього класу віртуальних машин є: **Microsoft Virtual PC, Bochs, Simics** та ін.

Такі віртуальні машини є незамінними у налагодженні й розробці програмного забезпечення, написаного на мовах низького рівня, і в разі емуляції ЕОМ зі специфічною архітектурою на стандартизовані персональні комп'ютери і сервери.

Не всі інструкції потрібно інтерпретувати. Частину інструкцій віртуальної машини можна виконувати без змін на реальному процесорі. Монітор віртуальних машин, проаналізувавши код, може інструкції віртуальної машини, що збіглися з інструкціями реального процесора, без змін відправити для виконання на реальному процесорі, а решту інструкцій інтерпретувати – вибіркова емуляція інструкцій.

З'являються додаткові витрати на аналіз інструкцій. Оскільки в програмах часто зустрічаються цикли, інтерпретація яких може проводитися одноразово, і найбільш прості інструкції, що без змін виконуються на процесорі, становлять більшу частину програм, то продуктивність віртуальної машини збільшується в кілька разів порівняно з повною емуляцією інструкцій.

До віртуальних машин цього типу, наприклад, належать: **VMware Workstation, VMware Server, Serenity Virtual Station** та ін.

У третьому випадку емулюються API гостьової операційної системи.

API (Application Programming Interface) – це інтерфейс прикладного програмування.

Усі працюючі програми взаємодіють з обладнанням за допомогою інтерфейсу API. Тому можна перехопити звернення програм, що працюють під управлінням гостьової операційної системи,

до API, довести його до виду, прийнятому в хостовій операційній системі, і ретранслювати отриманий результат до API хостової операційної системи.

Результат виконання запиту хостовою операційною системою перетворюється до вигляду, що приймається гостьовою операційною системою, і передається програмі, що видала запит.

Якщо гостьова і хостова операційні системи сумісні за своїми API, то перетворювати звернення не потрібно, достатньо тільки перенаправляти їх.

Однак у такої системи віртуалізації є недоліки.

Не все програмне забезпечення можна запускати на віртуальній машині із цим принципом віртуалізації, оскільки часто використовуються недокументовані API і звернення безпосередньо до апаратури.

Операційні системи активно розвиваються, вносяться коректування в API і додаються нові можливості. Тому емулятори API швидко застарівають і для їх модернізації необхідні великі витрати трудомісткості.

Емулятори API прив’язуються до конкретних операційних систем. Це звужує коло їх використання та споживчі властивості.

А втім, використання емуляції API дає змогу уникнути значних втрат продуктивності.

Як приклад віртуальних машин, що використовують емуляцію API, можна привести такі продукти: **Wine (Wine Is Not an Emulator)**, який використовується для запуску Dos і Windows – додатків під управлінням операційної системи Linux; **UML (User Mode Linux)**, що вбудований у ядро Linux і дає змогу запускати кілька копій операційної системи на одному комп’ютері.

2.2. Безпека у віртуальних хмарах

Віртуалізація – це перехід від технології, що лежить в основі консолідації серверів і центрів обробки даних, до основних компонентів для створення гнучкої інфраструктури.

У разі реалізації віртуалізації в будь-якому середовищі виникають багато завдань, які треба вирішити.

Розглянемо питання, пов'язані з безпекою, які потрібно вирішити, коли йдеться про використання віртуалізації для хмарних обчислень.

Один з ризиків – ***ризик компрометації гіпервізора віртуальних машин***.

Гіпервізор (англ. Hypervisor), або монітор віртуальних машин (у комп'ютерах), – програма або апаратна схема, що забезпечує або дозволяє одночасне, паралельне виконання декількох операційних систем на тому самому хост-комп'ютері.

Гіпервізор також забезпечує ізоляцію операційних систем одна від одної, захист і безпеку, поділ ресурсів між різними запущеними ОС і управління ресурсами.

Якщо ***гіпервізор*** ненадійний, то він стане першою стратегічною метою зловмисників. Якщо не усунути цю небезпеку, то в хмарі атака може призвести до масштабних руйнувань. Це потребує додаткового рівня ізоляції мережі й посиленої системи моніторингу безпеки.

Для аналізу цієї небезпеки спробуємо для початку зрозуміти природу гіпервізора.

Консультант з безпеки й один із засновників компанії Nemertes Research Group Inc. Андреаса Антонопулос (Andreas Antonopoulos) вважає, що «гіпервізор – вузькоспеціалізований пристрій».

Гіпервізор менше і більш спеціалізований, ніж операційна система загального призначення, і менше відкритий для атак, оскільки в нього менше або взагалі немає відкритих зовні мережевих портів. Він нечасто змінюється й не виконує програми сторонніх розробників.

У гостьовій ОС, яка може стати жертвою атак, немає прямого доступу до гіпервізора.

Гіпервізор прозорий для мережевого трафіку, якщо не брати до уваги вхідний і вихідний трафіки виділеного інтерфейсу управління гіпервізором.

На поточний момент не задокументовано жодної атаки на ***гіпервізор,*** також говоритися про низьку ймовірність таких атак.

Хоча масштаб руйнувань у разі компрометації гіпервізора може бути величезним (компрометація всіх гостьових систем), імовірність такої події низька, тому що уразливість гіпервізора і ймовірність атаки дуже низькі.

Наступна проблема полягає в масштабуванні функціональності віртуальних ЛВС за межі наявних кордонів для підтримки великих за розміром хмар.

2.3. Різновиди віртуалізації

Виділяють такі різновиди віртуалізації:

1) віртуалізація серверів (повна віртуалізація і паравіртуалізація);
2) віртуалізація на рівні операційних систем;
3) віртуалізація мережі;
4) віртуалізація додатків;
5) віртуалізація подань;
6) віртуалізація робочих місць САПР;
7) віртуалізація сховищ.

Технологія віртуалізації серверів дає змогу запускати на одному сервері декілька логічних одиниць – *віртуальних машин*, які повністю відтворюють роботу незалежних фізичних серверів.

Це дає можливість розміщувати на одній одиниці обладнання кілька десятків незалежних операційних систем і корпоративних додатків, ефективніше використовуючи IT-інфраструктуру.

Віртуалізацію можна описати як *дію гіпервізора,* завдяки якій можна створювати безліч логічних розділів, у яких працюватимуть різні операційні системи (одночасно на одному фізичному сервері).

Кожен розділ (званий також ***віртуальною машиною***) – це програмне середовище, яке надає ресурси (за допомогою емуляції обладнання або пристроїв). Поверх віртуальної машини можна інсталювати операційну систему, а також один або кілька додатків.

В основі рішення віртуалізації на рівні машини лежить монітор віртуальних машин (***Virtual Machine Monitor, VMM)***. ***VMM*** відповідає за створення й ізоляцію віртуальної машини ив збереження її стану, а також за організацію доступу до системних ресурсів.

Схема ***VMM*** залежить від архітектури конкретного процесора; незважаючи на те що VMM дає змогу працювати всередині віртуальної машини немодифікованим операційним системам.

Для реалізації VMM для створення інтерфейсу між віртуальними машинами і віртуалізованими системними ресурсами використовуються три можливі методи:

- *повна віртуалізація,*
- *власна віртуалізація*,
- *паравіртуалізація*.

Повна віртуалізація. У разі використання цього методу монітором VMM (для абстрагування віртуальної машини від реального обладнання) створюється й підтримується повна віртуальна система.

Цей підхід дає змогу виконувати у віртуальній машині операційну систему без будь-яких її модифікацій.

Перевагою повної віртуалізації і підходу з повним розв'язанням фізичного обладнання та віртуальної машини є здатність легко переносити віртуальні машини між серверами з різними фізичними конфігураціями.

Власна віртуалізація. Така віртуалізація залежить від архітектури процесора, що віртуалізується (як-от у серіях процесорів AMD-V і Intel VT). Ці процесори мають в своїй апаратній частині нові режими виконання, інструкції та структури даних, які призначені для зменшення складності VMM.

За власної віртуалізації монітора VMM не потрібно підтримувати в програмному забезпеченні характеристики ресурсів віртуальної машини і її стан.

Так само, як і в разі повної віртуалізації, усередині віртуальних машин операційні системи можуть виконуватися без їх модифікації. Hyper-V використовує цей метод для роботи застарілих операційних систем.

Такий тип реалізації має багато переваг – від спрощення архітектури VMM до істотного підвищення продуктивності (у підсумку – зниження накладних витрат програмного забезпечення).

Паравіртуалізація. Паравіртуалізація була розроблена як альтернатива використанню двійкової трансляції під час обробки інструкцій процесора x86, що не віртуалізуються. За цього підходу потрібна модифікація гостьових операційних систем (щоб уможливити «гіпервиклики» від віртуальної машини до гіпервізора).

2.4. Рішення щодо серверної віртуалізації

VMware vSphere – платформа для серверної віртуалізації, що надає максимальні рівні доступу та швидкості реагування для всіх додатків і служб.

VMware vCenter – єдина консоль управління внутрішньою віртуальною інфраструктурою дата-центрів.

VMware vCloud Director – інструмент управління внутрішніми і зовнішніми хмарами організації.

Citrix XenServer – рішення Enterprise-рівня для віртуалізації серверів.

Microsoft Server Hyper-V – система віртуалізації в серверному середовищі Microsoft, що надає організаціям широкі можливості щодо управління й масштабування IT-ресурсів.

Microsoft System Center Virtual Machine Manager – рішення, призначене для управління фізичними і віртуальними машинами.

Віртуалізація на рівні операційної системи заснована на ідеї підтримки операційною системою хоста декількох ізольованих розділів (або віртуальних середовищ, virtual environment – VE).

Віртуалізація досягається за допомогою мультиплексування доступу до ядра (із забезпеченням безпеки системи від віртуальних середовищ).

Віртуалізація на рівні ядра ОС має на увазі використання одного ядра хостової ОС для створення незалежних, паралельно працюючих операційних середовищ.

Для гостьового ПЗ створюється тільки власне мережеве й апаратне оточення. Такий варіант використовується в **Virtuozzo (для Linux і Windows)**, **OpenVZ** (безкоштовний варіант Virtuozzo) **і Solaris Containers.**

Віртуалізація мережі – це повне відтворення фізичної мережі програмним методом.

Віртуальні мережі аналогічні фізичним мережам з погляду надійності й можливостей.

Однак вони мають безліч додаткових експлуатаційних переваг, як-от незалежність від обладнання, швидка ініціалізація, можливість

розгортання без переривання роботи систем, автоматизоване обслуговування й підтримка як сучасних, так і застарілих додатків.

Віртуалізовані застосування (зокрема, бази даних, системи ERP і CRM, електронна пошта, додатки для спільної роботи, проміжне ПЗ Java, засоби бізнес-аналітики та ін.) переносяться у віртуальне середовище, де працюють швидше і краще, характеризуються високою доступністю й адаптивністю, а також підтримують можливість аварійного відновлення та використання в хмарній інфраструктурі.

Рішення **VMware** щодо віртуалізації додатків, засноване на **VMware vCloud Suite**, допомагає підвищити якість IT-послуг, знизити складність інфраструктури, досягти максимальної ефективності, а також знизити витрати завдяки виділенню правильного обсягу ресурсів.

Віртуалізація додатків – це технологія, спрямована на розділення й ізоляцію додатків на стороні клієнта (що працюють під місцевою операційною системою).

Додатки ізолюються у віртуальному середовищі, що розташоване між операційною системою і стеком додатків.

Віртуальне середовище завантажується до додатка, ізолює його від інших додатків і операційної системи, а також запобігає модифікації додатка локальних ресурсів (таких як файли і налаштування реєстру).

Завдяки цій технології можна використовувати на одному комп'ютері, а точніше в тій самій операційній системі, кілька несумісних між собою додатків одночасно. Віртуалізація додатків дає змогу користувачам запускати той самий, заздалегідь сконфігурований додаток або групу додатків із сервера.

При цьому додатки працюватимуть незалежно один від одного, не вносячи жодних змін в операційну систему.

Такий варіант віртуалізації використовується в **Sun Java Virtual Machine, Microsoft Application Virtualization** (раніше Softgrid), **Thinstall** (на початку 2008 р. ввійшла до складу VMware)**, Symantec / Altiris.**

Рішення для віртуалізації додатків і термінального доступу на основі продуктів:

- **Citrix XenApp** – фактичний стандарт доставки будь-яких Windows-додатків з мінімальними витратами в будь-яку точку світу.

Citrix XenApp забезпечує захищений термінальний доступ до корпоративних ресурсів і централізоване управління ними.

- **Citrix NetScaler** – рішення для оптимізації процесу доставки web-додатків.
- **Citrix Branch Repeater** – рішення щодо прискорення й оптимізації доставки додатків, робочих столів.
- **VMware ThinApp** – рішення, яке створює пакет, який містить додаток і пов'язану з ним частину операційної системи, що дає змогу здійснювати термінальний доступ до додатка.
- **VMware Zimbra** – система автоматизації спільної діяльності робочих груп (пошта, спільні документи).
- **Microsoft App-V** – рішення для віртуалізації і доставки додатків, швидкого оновлення, запуску несумісних додатків, управління правами.

Віртуалізація уявлень (робочих місць) – це інфраструктура віртуальних робочих столів VDI. На відміну від «звичайного» термінального доступу, йдеться про декілька віртуальних машин на фізичному сервері. Кожен користувач отримує свій власний віртуальний ПК, куди може підключатися хоч із телефона.

Віртуалізація робочих столів реалізується за допомогою інфраструктури Virtual Desktop Infrastructure (VDI). Цей термін позначає комбінацію апаратного забезпечення, програмного забезпечення для віртуалізації, а також інструментів управління.

Віртуалізація робочих столів може бути реалізована або як статичні віртуальні робочі столи, або як динамічні віртуальні робочі столи.

У першому варіанті фізичний настільний комп'ютер замінюється віртуальним. У другому варіанті кінцеві користувачі динамічно підключаються до одного з віртуальних робочих столів з пулу (на вимогу).

Віртуальні робочі столи розміщуються на надійних відмовостійких серверах у дата-центрах. Користувач завжди має повний контроль над усіма своїми ресурсами. Підключення проводиться за допомогою служби термінального доступу Microsoft Windows по захищених каналах (RDP), що забезпечує конфіденційність інформації, що передається.

Для комфортної роботи з віддаленим робочим столом використовуються технології **Microsoft**, **VMware**, **Citrix**. Після підключення користувачів до віддаленого робочого столу можна скористатися всіма мережевими й локальними пристроями (флеш-накопичувачами, приводами, принтерами тощо).

Різні рішення для організації віддаленого робочого столу:

- **VDI від Amazon AWSБ.**
- **Amazon WorkSpace** – віддалені робочі столи в хмарі Amazon AWS з попередньо встановленими пакетами офісних додатків з можливістю інтеграції сервісу з локальної службою каталогів.
- **Сервіс DaaS** – швидкий доступ до робочого столу користувачів з будь-якої точки, з будь-якого пристрою. Користувачі отримують можливість працювати ефективніше, оперативно одержуючи доступ до своїх даних в будь-який час.
- **VDI на базі Citrix або Vmware.**

Віртуалізація робочих місць САПР

Технологія віртуалізації робочих місць САПР побудована на базі рішень **Citrix і технології NVIDIA GRID**.

Технологія NVIDIA GRID – це рішення для віртуалізації GPU, віддаленого доступу й управління сеансом, яке дає змогу декільком користувачам одночасно працювати з графічно насиченими додатками, використовуючи загальні ресурси GPU.

Технологія покликана вирішити проблему віртуалізації і віддаленого доступу до робочих місць у таких сферах, як автоматизація проєктування (CAD), управління інформацією в будівництві (BIM), управління життєвим циклом вироби cv (PLM), автоматизація діяльності кредитно-фінансових установ, робота із системами архівації і передача зображень у сфері охорони здоров’я, фото- і відеоредагування.

Завдяки цьому, а також використанню технологій Citrix XenServer GPU pass-through і Citrix XenDesktop HDX 3D стало можливим запускати важкі 3D-додатки на стороні сервера й надавати до них доступ в термінальному режимі з повною підтримкою професійної графіки NVIDIA GRID.

Переваги:

- зниження до мінімуму ризику втрати корпоративних даних огляду на їх відсутність на локальному робочому місці й можливості прямого доступу до них;
- економія на ТСО і на всіх витратах на обслуговування робочих місць, уникнення складнощів із пошуком у конкретній точці кваліфікованих ІТ-фахівців тощо;
- можливість дистанційної роботи співробітників.

Віртуалізація сховищ

Мета ***віртуалізації сховищ*** як одного з компонентів програмного сховища – підвищення продуктивності без придбання додаткового обладнання для зберігання даних.

Віртуалізовані сховища повинні підтримувати швидку ініціалізацію, щоб забезпечити розгортання високопродуктивних ефективних сховищ з тією ж швидкістю, що і ВМ (вірт. машин).

Модель управління сховищами повинна орієнтуватися на потреби ВМ і забезпечувати зручність роботи адміністраторам віртуальних середовищ, до завдання яких входить управління ресурсами зберігання. Для зручної роботи з віртуалізованими сховищами від них вимагається високий ступінь інтеграції з платформою гіпервізора.

Віртуалізація сховищ VMware – це поєднання можливостей, яке формує рівень абстракції для ресурсів фізичного сховища й підтримує їх адресацію, оптимізацію і адміністрування у віртуальному середовищі.

Технології віртуалізації сховищ забезпечують ефективний спосіб управління ресурсами зберігання у віртуальній інфраструктурі і реалізують такі переваги:

- значне підвищення ефективності та гнучкості використання сховищ;
- спрощення установки виправлень ОС і зниження вимог до драйверів, усунення залежності від топології сховища;
- збільшення часу безвідмовної роботи додатків і спрощення стандартних процесів;
- використання й доповнення наявної інфраструктури зберігання.

2.5. Контрольні запитання до теми 2

1. Назвіть основні переваги віртуалізації.
2. Вкажіть основні різновиди віртуалізації.
3. Назвіть основні платформи віртуалізації.
4. Що таке повна віртуалізація?
5. Рішення для віртуалізації додатків і термінального доступу
6. Які є різновиди віртуалізації?
7. Хто контролює ресурси в хмарі?
8. Що таке віртуалізація додатків?
9. Що таке віртуалізація мережі?
10. Віртуалізація на рівні операційної системи.
11. Віртуалізація на рівні ядра ОС.
12. Віртуалізація робочих місць САПР.
13. Віртуалізація робочих столів.
14. Віртуалізація уявлень (робочих місць).
15. Віртуалізовані застосування.
16. Монітор віртуальних машин (Virtual Machine Monitor, VMM)?

Тема 3. SOA

3.1. Сервісні моделі хмарних обчислень

SOA-технології послужили основою для хмар.

Однією з найістотніших технологічних новацій, що лежать в основі хмарних обчислень, є технології віртуалізації.

Першою компанією, яка повною мірою усвідомила комерційну перспективу загальнодоступних технологій віртуалізації, стала Amazon.

Якщо до 2006 року під віртуалізацією розуміли можливість розгортання потрібної кількості віртуальних серверів на власному

устаткуванні, завдяки появі Amazon Elastic Compute Cloud почали орендувати віртуальні сервери на чужому обладнанні.

У цьому і полягає суть хмарних пропозицій класу «інфраструктура як сервіс».

Базою та основою сучасних хмарних обчислень слугувала сервіс-орієнтована архітектура (Service-Oriented Architecture, SOA) – надання додатків у режимі послуг (Application Service Provider, ASP) та ін.

Стандарт **Open Group SOA Reference Architecture** (еталонна архітектура SOA RA) надає рекомендації та варіанти архітектури, дизайну та реалізації рішень для створення архітектурно-орієнтованих рішень, включно з архітектурою хмарних обчислень.

Метою стандарту Open Group SOA Reference Architecture є надання прототипу для створення й оцінки архітектури.

SOA – модель взаємодії компонентів, яка пов’язує різні функціональні модулі додатків (сервіси) між собою за допомогою чітко визначених інтерфейсів.

Одним із відомих інтеграційних шаблонів є «Сервіс».

Інтерфейси не залежать від апаратних платформ, операційних систем або мов програмування, що використовуються для розробки цих програм. Це дає змогу окремим сервісам взаємодіяти між собою тим самим стандартним, але водночас універсальним способом.

Така особливість використання інтерфейсу, незалежного від оточення та платформи, отримала назву моделі «слабкого зв’язку». Її перевагами є підвищена гнучкість і адаптованість, оскільки заміна чи модернізація одного з компонентів системи не позначається на інших.

Сучасні підходи до реалізації SOA охоплюють технологічний рівень обміну даними та рівень бізнес-операцій.

Зокрема, одним із важливих результатів стала розробка спеціалізованої мови BPEL (Business Process Executable Language for Web Services) для опису аспектів взаємодії різних сервісів з погляду реалізації бізнес-правил.

Прийняття SOA як цільової архітектури передбачає і відповідний підхід до розробки додатків (**SODA – service-oriented development architecture**).

Для завдань електронного бізнесу функціональність SOA реалізується лише на рівні web-сервісів (служб).

Web-сервіс – програмні системи, які використовують:

1) XML як формат даних;

2) стандарти Web Services Description Language (WSDL) для опису своїх інтерфейсів;

3) Simple Object Access Protocol (SOAP) для опису формату повідомлень, що приймаються й надсилаються;

4) стандарт Universal Description, Discovery and Integration (UDDI) – створення каталогів доступних сервісів.

3.2. Сервіс-орієнтована модель

Сервіс-орієнтована модель повинна мати функції розширювання:

- користувач повинен мати можливість додати нові послуги або змінити набір доступних сервісів;
- користувач також повинен мати можливість звертатися до сервісів через мережу різних за своїми можливостями пристроїв – desktop-машин, мобільних пристроїв тощо.

Потім аналогічні послуги представили Google, Sun, IBM.

У 2006 році компанія Amazon запустила платформу Amazon Web Service (AWS), модернізувавши свої центри обробки даних.

Компанія Amazon відіграла основну роль у відкритті ринку хмарних обчислень у всьому світі.

У 2008 році були анонсовані хмарні платформи від Microsoft та Google, Windows Azure та Google App Engine відповідно.

У 2010 році побачив світ перший випуск платформи Windows Azure.

Починаючи з 2008 року ринок хмарних обчислень почав стрімко зростати, заповнюючись як топовими гравцями (Amazon, Microsoft, Salesforce, Google, HP, Dell, AT&T, RackSpace), так і організаціями, що пропонують хмарні ресурси для вирішення конкретних завдань (наприклад, OrangeScape).

Восени 2008 року Microsoft анонсувала повноцінну хмарну операційну систему Windows Azure, призначену для розробки хмарних додатків.

Офіційний реліз Windows Azure відбувся на початку 2010 року. У 2014 році платформу було перейменовано на Microsoft Azure.

На сьогоднішні Microsoft Azure залишається одним із найбільших і всеосяжних проєктів у сфері Cloud Computing.

Але 2010 рік можна вважати важливою датою в історії хмарних технологій не тільки через реліз Azure, але й завдяки появі низки хмарних сервісів, орієнтованих не на розробників, а на простих користувачів.

Компанії Microsoft та Google випустили набори безкоштовних онлайн-сервісів, які дають змогу користувачеві працювати з документами. Google це Google Docs, Microsoft – Office Web Apps. Обидва сервіси взаємопов'язані з поштою (Gmail та Hotmail) і файловими сховищами.

Microsoft та Google інтегрують підтримку своїх онлайн-сервісів у всі програмні середовища – як настільні, так і мобільні (Google створила ОС Android, а Microsoft – WindowsPhone 7).

Аналогічну концепцію просуває Apple – сервіс під назвою MobileMe.

MobileMe має поштовий клієнт, календар, адресну книгу, файлове сховище, альбом фотографій і інструмент для виявлення втраченого iPhone.

3.3. Сервісні моделі хмарних обчислень

Три моделі використання сервісів хмарних обчислень:

1. Infrastructure as a Service (IaaS) – інфраструктура як сервіс.

IaaS пропонує доступ до низькорівневих ресурсів: сховищ даних, обчислювальних пристроїв та пам'яті. Тут розвиваються технології віртуалізації, що використовують спеціальний керуючий процес – гіпервізор, який забезпечує виконання програм (і операційних систем) користувача у прозорому режимі.

2. Platform as a Service (PaaS) – платформа як сервіс.

Платформа – це прикладний програмний інтерфейс, що забезпечує додаткам можливість роботи в умовах хмар. Програма працює під керуванням спеціалізованої операційної системи, що надається постачальником хмарних обчислень. IaaS може лише гарантувати певну кількість процесорів або об'єм пам'яті, а все інше

повинно робити додаток, що розміщується користувачем. Приклади: Force.com, Google App Engine, Microsoft Azure (Platform).

3. Software as a Service (SaaS) – програма як сервіс.

Постачальник реалізує бізнес-логіку в рамках певної програми. Приклади: Google Docs, Salesforce CRM, SAP Business by Design.

4. FaaS – функція як послуга (англ. *function-as-a-service, FaaS*) – архітектурний шаблон, що дає можливість виклику екземпляра управляючого кода без потреби в управлінні серверами й серверним додатком; ключовий компонент безсерверних обчислень.

3.4. Проблема управління ресурсами в сервіс-орієнтованих системах

Парадигма хмарних обчислень – об'єднання у єдиний інформаційно-обчислювальний простір довільної множини гетерогенних і просторово-розподілених обчислювальних вузлів, що перебувають у різних адміністративних доменах, зі своєю локальною безпековою політикою.

Виокремимо особливості керування ресурсами залежно від виділених моделей використання.

Моделі **IaaS і PaaS** припускають, що динамічні характеристики додатків, що виконуються, випадкові, оскільки користувач може вирішувати довільні завдання приблизно так, як це відбувається в операційних системах загального призначення.

Модель **SaaS** відрізняється від **моделей IaaS та PaaS** тим, що тут постачальник контролює весь життєвий цикл програми та може знати заздалегідь статичні й динамічні вимоги до ресурсів.

Завдання управління ресурсами ускладнюється ще й вимогами до якості сервісу: час гарантованого відгуку, рівень пропускної спроможності, доступність, стійкість до відмов, час відновлення після збою тощо.

Тому універсального розв'язання задачі оптимального управління ресурсами у моделях **IaaS та PaaS** немає.

Переважно ведуться дослідження в галузі віртуалізації елементів обчислювальних систем.

Модель **SaaS** надає більше можливостей щодо управління ресурсами.

Знаючи характеристики завдання, можна більш точно визначати, коли та які ресурси має отримати завдання, з урахуванням поточного стану ресурсів розподіленої обчислювальної системи (процесор, оперативна пам'ять, пристрої введення / виведення та пристрої передачі даних).

Порівняно з СОА, хмарні обчислення ставлять більш глобальне завдання – надати сервіс обчислень будь-якого рівня не лише на рівні додатків, а й на рівні операційних систем, спеціалізованих обчислювальних ресурсів тощо.

Хмарні обчислення вводять додаткові (нефункціональні) вимоги до сервісу.

Системи хмарних обчислень повинні мати еластичність, тобто можливість задіяти нові фізичні ресурси зі збільшенням і звільняти ресурси зі зменшенням кількості запитів користувача.

Концепція хмарних обчислень охоплює питання, пов'язані з надійністю, адаптивністю, рівнем якості сервісу та багато інших, що стосуються побудови й експлуатації реально діючих систем порівняно з СОА, що описує лише схему побудови додатків із компонентів.

3.5. Ключові терміни

Amazon Elastic Compute Cloud (Amazon EC2) – вебсервіс, що надає обчислювальні ресурси, які масштабуються, у хмарі.

Amazon Simple Storage Service (Amazon S3) – онлайн вебслужба, пропонована Amazon Web Services, що надає можливість для зберігання й отримання будь-якого обсягу даних, у будь-який час із будь-якої точки мережі файлового хостингу.

Amazon Web Services (AWS) – інфраструктура платформ хмарних вебсервісів компанії Amazon.

Amazon, Inc. – американська компанія з обороту серед тих, хто продає товари та послуги через інтернет і один з перших інтернет-сервісів, орієнтованих на продаж реальних товарів масового попиту.

AT&T Inc – американська телекомунікаційна компанія та найбільший медіаконгломерат.

BPEL (Business Process Execution Language) – мова на основі XML для формального опису бізнес-процесів і протоколів їх взаємодії між собою.

Facebook – соціальна мережа.

Gmail – безплатний поштовий сервіс компанії Google.

Google App Engine – сервіс хостингу сайтів і web-додатків на серверах Google.

Google Inc. – американська транснаціональна публічна корпорація, яка інвестує в інтернет-пошук, хмарні обчислення й рекламні технології.

Hewlett-Packard Company (HP) – американська компанія у сфері інформаційних технологій, постачальник апаратного і програмного забезпечення для організацій та індивідуальних споживачів.

Hotmail – безплатний сервіс електронної пошти, що входив до сервісів Windows Live, який надавав доступ через вебінтерфейс, POP3, DeltaSync та Exchange Activesync.

IBM (International Business Machines) – американська компанія, один із найбільших у світі виробників і постачальників апаратного та програмного забезпечення, а також IT-сервісів і консалтингових послуг.

Infrastructure as a Service (IaaS) – інфраструктура як сервіс, що пропонує доступ до низькорівневих ресурсів: сховищ даних, обчислювальних пристроїв та пам’яті.

iPhone – серія смартфонів, розроблених корпорацією Apple.

Microsoft – найбільша транснаціональна компанія з виробництва пропрієтарного програмного забезпечення для різноманітної обчислювальної техніки: персональних комп’ютерів, ігрових приставок, КПК, мобільних телефонів тощо, розробник програмної платформи – сімейства операційних систем Windows.

Microsoft Azure – хмарна платформа Microsoft з набором допоміжних служб для обчислень, зберігання, даних, роботи в мережі та підтримки програм.

Microsoft SharePoint – інструмент для створення розширюваних Web-сторінок і Web-сервісів.

MobileMe – платний набір інтернет-сервісів із підтримкою push-технологій від Apple.

OrangeScape – крос-хмарна PaaS платформа.

Platform as a Service (PaaS) – платформа як сервіс. Платформа – прикладний програмний інтерфейс, що забезпечує можливість додатком функціонувати в умовах хмар.

AP Business ByDesign – система, призначена для малого середнього бізнесу (50–500 співробітників) і для виробничих підприємств, а також для тих підприємств, що надають професійні бізнес-послуги. Це єдиний multitenant сервіс для автоматизації всіх сфер діяльності компанії, включно з виробництвом, фінансами та продажами.

SODA (Service-Oriented Developmentarchitecture) – підхід до розробки додатків.

Software as a Service (SaaS) – програма як сервіс, що реалізує бізнес-логіку в рамках певного додатка.

SQL Azure – хмарний сервіс від корпорації Microsoft, що надає можливість зберігання й обробки реляційних даних, а також створення звітності.

Sun Microsystems – американська компанія, виробник програмного й апаратного забезпечення.

Windows Phone 7 – версія мобільної операційної системи, розробленої Microsoft.

Вебсервіс, вебслужба (web-service) – програмна система, що ідентифікується вебадресою, зі стандартизованими інтерфейсами.

Документи Google (Google Docs) – безплатний онлайн-офіс, що містить текстовий, табличний процесор і сервіс для створення презентацій, а також інтернет-сервіс хмарного зберігання файлів із функціями файлообміну, що розробляється компанією Google.

ОС Android – операційна система для смартфонів, планшетних комп’ютерів, електронних книг, цифрових програвачів, наручних годинників, ігрових приставок, нетбуків, смартбуків, окулярів Google, телевізорів та інших пристроїв.

Парадигма хмарних обчислень – об’єднання у єдиний інформаційно-обчислювальний простір довільної множини гетерогенних і просторово-розподілених обчислювальних вузлів, що перебувають у різних адміністративних доменах, зі своєю локальною безпековою політикою.

Платформа – прикладний програмний інтерфейс, що забезпечує додатку можливість функціонувати в умовах хмар.

Сервіс-орієнтована архітектура (SOA) – модель взаємодії компонентів, яка пов’язує різні функціональні модулі додатків (сервіси) між собою за допомогою чітко визначених інтерфейсів.

3.6. Список скорочень

API Application Programming Interface – інтерфейс програмування програм, інтерфейс прикладного програмування.

ASP Application Service Provider – надання програм у режимі послуг.

CRM Customer Relationship Management – система управління взаємовідносинами з клієнтами.

IaaS Infrastructure as a Service – інфраструктура як сервіс.

PaaS Platform as a Service – платформа як сервіс.

SaaS Software as a Service – додаток як сервіс.

SOA Service-Oriented Architecture – сервіс-орієнтована архітектура.

SOAP Simple Object Access Protocol – простий протокол доступу до об’єктів.

UDDI Universal Discovery, Description and Integration – технологія для публікації та пошуку Web-сервісів.

WSDL Web Services Description Language – мова опису Web-сервісів.

XML eXtensible Markup Language – розширювана мова розмітки.

CC Cloud Computing – хмарні обчислення.

3.7. Контрольні запитання до теми 3

1. Сервіс-орієнтована архітектура (SOA) – це…
2. Сервіс-орієнтована модель, властивості.
3. Сучасні підходи до реалізації SOA.
4. Сценарії – варіанти меж застосування тверджень, які зберігаються стосовно хмар.

5. Технологічні новації, що лежать в основі хмарних обчислень.

6. Типові контролери кордонів містять…

7. Розкажіть про функціональність SOA на рівні web-сервісів (служб).

8. Хмара спільноти або загальна хмара (Community cloud).

9. Хмарна система як модель клієнт – сервер.

Тема 4. Microsoft Azure

4.1. Microsoft Azure, знайомство і вивчення складових частин

Ключова ідея хмарних платформ – забезпечити розробнику умови для розміщення й виконання створених ним програм без потреби в управлінні тим, де фізично розташовується виконуваний файл і зберігаються дані.

Після розміщення програми можна забути про неї, оскільки подальші дії щодо забезпечення її працездатності (крім виправлення помилок, звичайно) бере на себе платформа. Як ключові операції, виконувані постачальником послуг, можна виділити:

• застосування поновлення в момент їх виходу;

• додавання дискової пам’яті в разі виникнення потреби;

• перезапуск системи в разі аварійних ситуацій;

• реплікацію даних для підвищення надійності їх збереження;

• управління дисками та іншим апаратним забезпеченням.

Обговорюючи різні типи хмарних сервісів – програмне забезпечення, платформу й інфраструктуру як сервіс, слід звертати увагу на т. зв. кордони керованості, тобто на те, чим порівняно з традиційними моделями розгортання у власній інфраструктурі можна управляти в разі переходу на хмарну платформу.

Зі зрозумілих причин інфраструктура як сервіс надає великі можливості щодо налаштування окремих компонентів, тоді як

платформа як сервіс і програмне забезпечення як сервіс практично мінімізують ці можливості.

Із розгортанням власної інфраструктури ви управляєте всіма її компонентами – від мережевих ресурсів до застосувань, що виконуються, тоді як із використанням моделі IaaS ви можете контролювати такі компоненти, як середовище виконання коду, безпека й інтеграція, бази даних тощо. З переходом до моделі PaaS усі компоненти платформи надаються як сервіси з обмеженими можливостями для управління ними. Це зроблено для того, щоб надати в розпорядження споживачів оптимально конфігуровану платформу, що не потребує додаткових налаштувань.

Таким чином, незважаючи на певні відмінності в пропонованих моделях хмарних обчислень, ключовим залишається те, що постачальники послуг надають набір служб, які можуть бути використані для розгортання додатків. Термін «хмара» використовується як метафора, що описує образ складної інфраструктури, за якою ховаються всі технічні деталі.

Microsoft Azure належить до категорії PaaS, оскільки не забезпечує доступ до апаратного забезпечення і таких служб операційної системи, як мережеві інтерфейси й управління дисками. Платформа Windows Azure повністю приховує складні технічні деталі реалізації нижнього рівня операційної системи від користувача й надає зручний API для управління ресурсами на логічному рівні.

Розробникові досить створити необхідні сховища, дати їм імена і потім використати їх для розміщення власних застосувань. Одна з ключових ідей Windows Azure – це створення розподіленої операційної системи, за допомогою якої розробники можуть виконувати свої застосування без використання стандартних підходів, прийнятних у настільних операційних системах.

Наприклад, немає потреби копіювати файли або налаштовувати Internet Information Services(IIS), управляти віртуальними шляхами й пулами додатків.

Усе це відбувається саме без участі розробника, від нього вимагається тільки створити обліковий запис і визначати потрібні ресурси.

Windows Azure є операційною системою для хмари (розташованою у хмарі), яка повністю абстрагує рівень апаратного забезпечення системи від розробника та дає йому змогу вибрати необхідні компоненти і рівень ***Service Level Agreement(SLA)*** без налаштування апаратного забезпечення.

На цей час для забезпечення необхідної обчислювальної потужності розробник може вибирати з п’яти типів віртуальних машин, кожна з яких може бути описана за допомогою таких параметрів:

- процесор можна вибрати в діапазоні від одноядерного процесора з частотою 1 ГГц до 8-ядерного – для ситуацій, коли потрібно забезпечити паралельне виконання програми;
- оперативна пам’ять вибирається відповідно до потреб, нижня межа починається від 768 мегабайт, верхня може досягати 8 гігабайт;
- постійна пам’ять визначається розміром використовуваного диска, починається від 20 гігабайт і може досягати 2 терабайти на один екземпляр додатка;
- продуктивність системи введення / виведення характеризується швидкістю операцій обміну, можна вибрати низьку, помірну або високу.

Тарифний план може містити всі або деякі з цих видів ресурсів. Якщо якийсь із ресурсів наближається до верхнього ліміту, то білінгова система може видати певну позику.

Якщо треба масштабувати додаток, достатньо в конфігураційному файлі збільшити число використовуваних віртуальних машин і впродовж короткого проміжку часу будуть виділені необхідні ресурси. Якщо потрібно зменшити потужність комп’ютера, то треба скоротити кількість віртуальних машин і білінгова система відразу зменшить поточні витрати.

Також можна у будь-який момент змінити розмір і параметри машин, але це потребує додаткового часу на перезапуск сервісу, оскільки платформа повинна наново розгорнути ваше застосування. Така операція зазвичай займає близько п’яти хвилин, після чого буде запущений новий екземпляр на оновленій віртуальній машині. Вартість використання прямо пропорційна вибраній конфігурації, але розробник сам може налаштовувати необхідну конфігурацію.

Сама платформа Windows Azure Platform є хмарним рішенням компанії Microsoft для розробки й забезпечення функціонування хмарних сервісів і реалізує модель Platform As A Service (PaaS).

Коротко опишемо основні компоненти, які входять до складу Windows Azure Platform.

- Windows Azure – надійна, масштабована, безпечна й доступна операційна система в хмарі (також називається «Операційна система як сервіс»). Надає обчислювальні потужності й засоби зберігання інформації, а також ряд механізмів управління сервісами. Може розглядається як служба хостингу та забезпечує виконання додатків на сервері.
- SQL Azure – реляційна база даних, доступна як сервіс (також називається «База даних як сервіс»). Підтримує основні можливості Microsoft SQL Server зі зберігання реляційних даних і, на відміну від своїх аналогів, не потребує адміністрування та супроводів.
- Windows Azure AppFabric – програмні модулі (сервіси) для забезпечення комунікацій (Service Bus) і контролю доступу (Access Control). AppFabric має служби, які можуть бути інтегровані в призначені для користувача додатки: наприклад, Access Control дає змогу використовувати системи аутентифікації сторонніх виробників, як-от Facebook і Google, кешування дає можливість створювати розподілений кеш, а служба Service Bus реалізує система обміну повідомленнями всередині хмари.

Екземпляр в Windows Azure є одиницею розгортання й відбивається на ту або іншу віртуальну машину, для якої підтримується ряд зумовлених конфігурацій. Частина платформи, звана Windows Azure Fabric Controller, відповідає за фізичне розгортання необхідних віртуальних машин. Усе, що вимагається від користувача для створення нового сервісу, – вказати необхідне число екземплярів віртуальних машин, які мають бути розгорнуті для цього сервісу.

Користувачам доступні такі функції, як ручний запуск і залишок екземплярів, управління числом екземплярів, тоді як Windows Azure Fabric Controller забезпечує автоматичне управління життєвим циклом екземплярів віртуальних машин, включно з їх перезапуском, створенням резервних копій, копіюванням і т. ін.

Платформа Windows Azure також містить ряд сервісів для зберігання даних. Ці сервіси підтримують георозподіл та інші способи надійного зберігання інформації, включно з потрійною реплікацією у рамках кластера й центру обробки даних.

Крім того, вони можуть забезпечувати вимоги щодо масштабованості через балансування навантаження й автоматичного створення копій, що розподіляються між серверами. Різні типи сховищ забезпечують механізми збереження цих різних типів у хмарі. Так сховище бінарних об'єктів дає змогу зберігати файли. Табличне сховище підтримує можливість зберігання незв'язаних даних, а сховище черг – зберігання повідомлень і зміни імені.

У Windows Azure підтримується і модель розгортання віртуальних машин – ця можливість забезпечує підтримку моделі інфраструктури, що надається як сервіс (IaaS). Насамперед це використовується для сервісів, яким потрібна інтеграція з операційною системою Windows Server. Ця модель забезпечує більший контроль над середовищем, у якому виконується хостинг сервісу, і може використовуватися, наприклад, для хостингу наявних сервісів.

Хмарний сервіс у Windows Azure зазвичай має більше одного примірника. Кожен екземпляр може реалізовувати всю або частину логіки програми. Розробники можуть управляти числом і типом ролей, під якими виконується прикладний сервіс.

4.2. Ролі у Windows Azure

Ролі у Windows Azure можна порівняти зі стандартними проєктами в Visual Studio – кожен екземпляр є окремим проєктом. Ці ролі представляють різні типи додатків, підтримуваних у Windows Azure. На цей час у Windows Azure підтримуються такі ролі:

• веброль (Web role),

• прикладна роль (Worker role).

Веброль – забезпечує підтримку протоколів HTTP і HTTPS через спеціальні точки входу (endpoints). Хостинг цієї ролі здійснюється у вебсервері IIS. Веброль можна порівняти з проєктом на ASP.NET, відмінності виявляються лише в способах конфігурації і використовуваних додатками збірках стека .Net.

Прикладна роль – забезпечує зовнішні точки входу, доступні через TCP / IP і порти, відмінні від 80 (HTTP) і 443 (HTTPS). Таким чином, прикладні ролі – це додатки, схожі із сервісами Windows, і вони можуть використовуватися для виконання фонових завдань.

Взаємодія між різними ролями може виконуватися, наприклад, через передачу повідомлень або з використанням інших традиційних комунікаційних механізмів.

4.3. Короткі підсумки

Ключова ідея, яка є основою хмарних технологій, полягає в найбільш ефективному використанні комп'ютерних ресурсів із мінімальними витратами на її адміністрування.

Виділяють три основні види хмарних платформ: інфраструктура як сервіс (IaaS), програмне забезпечення як сервіс (SaaS) і платформа як сервіс (PaaS).

Хмарна платформа SaaS дає змогу розгортати додаток у хмарному оточенні.

Хмарна платформа PaaS дає змогу розгортати додаток і необхідні додаткові сервіси (бази даних, сховища) з обмеженими можливостями адміністрування операційного оточення.

Windows Azure є операційною системою для хмари.

SQL Azure – реляційна база даних, доступна як хмарний сервіс.

Windows Azure AppFabric – програмні модулі (сервіси) для забезпечення комунікацій і контролю доступу.

Примірник у Windows Azure є одиницею розгортання й відбивається на ту чи іншу віртуальну машину, для якої підтримується ряд зумовлених конфігурацій.

Платформа Windows Azure також підтримує ряд сервісів для зберігання бінарних даних.

Ролі в Windows Azure можна порівняти з додатками в рамках стандартної операційної системи.

4.4. Контрольні запитання до теми 4

1. Хмарні служби Azure. Розкажіть про інтегрований інтерфейс розробки.
2. Хмарні служби Azure. Який моніторинг працездатності та оповіщенн?
3. Хмарні служби Azure. Що ви знаєте про створення додатків і інтерфейсів API?
4. Хмарні служби Azure. Що ви знаєте про створення ефективних додатків?

5. Хмарні служби Azure. Як проходить тестування додатків перед розгортанням?

6. Які основні компоненти, які входять до складу Windows Azure Platform?

7. Які особливості використання Microsoft Azure?

Тема 5. Microsoft Azure Virtual Machines

5.1. Огляд функціональності

Віртуальні машини є новим сервісом, що надається платформою Windows Azure, і вони дають можливість набагато простіше і гнучкіше переносити локальні інфраструктури в хмару або створювати нові програмні рішення, яким критичне постійне сховище (не очищене за кожним перезавантаженням екземпляра виконання).

По суті, після 7 червня 2012 року Windows Azure складно назвати SaaS, PaaS або якою б там не було платформою, тепер це радше парасольковий термін, що поєднує під собою безліч абревіатур.

До сценаріїв використання віртуальних машин у Windows Azure можна віднести практично всі типи програм, які використовуються в локальній інфраструктурі: бізнес-програми, CRM, Active Directory, власні програми, а також можна об’єднати локальну та хмарну інфраструктуру, створивши гібридне рішення.

5.2. Відмінності нового сервісу від VM-ролі

Після анонсування нового сервісу постало питання: а чим він відрізняється від того, що ми бачили раніше, а саме VM-ролі у рольовій моделі Cloud Services (тоді – Hosted Services).

VM-роль була введена в платформу для використання в тих випадках, коли можливостей Web / Worker-ролей було недостатньо для реалізації певних рішень, таких як, наприклад, перенесення складних додатків у хмару (Sharepoint тощо).

При цьому не надавалася жодна SLA для операційної системи у віртуальній машині, оскільки користувачем завантажувався власний VHD-диск. Однак SLA на самі віртуальні машини зберігався – можна було завантажити дві віртуальні машини та отримати 99,95 % SLA на доступність ролі.

Службовий рівень нагороди (SLA) є документованим сервісом між сервісним постачальником і покупцем, що ідентифікує як послуги, так і потрібний встановлений рівень служби.

Однак, припустимо, якщо ви завантажували кілька віртуальних машин і з однією з віртуальних машин щось відбувалося (наприклад, апаратний збій), усе, що було на цій машині, пропадало – усі унікальні дані, які зберігалися на диск і на згадку тощо. Це відбувалося через те, що у відповідь на помилку Windows Azure розгортало нову віртуальну машину, перед цим роблячи sysprep на образ, завантажений користувачем.

Це все начебто і нормально для простого застосування, проте перетворювалося на велику проблему – через непостійне сховище треба було таким чином перепроєктувати свою програму, щоб вона враховувала цю особливість.

Отже, до відмінностей належать:

• Тип сховища. Оскільки VM-роль, по суті, являла собою деякий сервіс із віртуальною машиною, у неї не було постійного сховища – у разі апаратної, наприклад, помилки ви втрачали всі дані з цієї машини. З віртуальними машинами трохи інакше – тепер можна додати постійне сховище у вигляді диска з даними, крім цього, диск вашої віртуальної машини постійно реплікується в три репліки.

• Типи розгортання. Ви повинні були створити локально ваш VHD і завантажити в хмару, після чого можна було використовувати його. З новим сервісом ви можете створити й завантажити VHD, скористатися як ним, так і будь-яким іншим доступним в галереї образів.

• Налаштування мережі. Налаштування для VM-ролі потрібно було робити в сервісній моделі, тоді як новий сервіс віртуальних машин можна конфігурувати на порталі керування Windows Azure і навіть автоматизувати за допомогою Powershell чи скриптів.

BLOB (англ. Binary Large Object – двійковий великий об’єкт) – масив двійкових даних.

5.3. Архітектура віртуальних машин

Загалом віртуальні машини Windows Azure засновані на сервісній моделі, що використовується Web / Worker-ролями вже давно, із серйозними модифікаціями, як-от постійне сховище й один екземпляр = одна роль.

У разі створення віртуальної машини автоматично створюється Cloud Service, який виступає як контейнер для цієї віртуальної машини. При цьому, якщо в Cloud Service є два осередки розгортання (Staging/Production), то у випадку віртуальної машини вона розгортається тільки в Production (що означає, що VIP swap недоступний).

Що стосується того, де зберігаються віртуальні машини, то це – сторінкові блоги в сховищі Windows Azure. У разі створення віртуальної машини VHD кладеться в сторінковий блог із можливістю подальшого запису.

При цьому є кілька дисків:

C: – операційна система;

D: – фізичні дані на сховища, які не резервуються та використовуються лише для тимчасового зберігання;

E: – дані користувача;

F: – логи.

Максимальний розмір операційної системи може бути до 127 Гб, але до віртуальної машини можна причепити певну кількість (залежно від розміру віртуальної машини) додаткових дисків із даними, у тому числі під час виконання віртуальної машини.

Віртуальні машини, розташовані в межах одного Cloud Service, мають прямий канал комунікацій один з одним – немає потреби налаштовувати щось окремо, як у випадку з окремими віртуальними машинами, коли для того, щоб забезпечити підключення, необхідно відкривати порти в сервісній моделі. Звичайно, не можна забувати про брандмауери на операційних системах. Із цим все зрозуміло – потрібно думати тільки про брандмауери й чітко їх налаштовувати (оскільки на

форумах періодично були питання про те, чому не «ходить» трафік). А якщо потрібно налаштувати звані кінцеві точки входу (endpoints)? Тут також все просто. Кожна кінцева точка входу асоціюється з віртуальною машиною та вказує, чи пропускатиме трафік.

До властивостей кінцевої точки входу належать:

ім'я – логічне ім'я у системі;

протокол – tcp/udp;

локальний порт;

публічний порт.

Налаштувати ж порт балансування навантаження, тобто той, на який буде приходити користувач і вже потім направлятись на одну з віртуальних машин у наборі, можна за допомогою визначення однієї точки входу на всіх необхідних віртуальних машинах і спеціальної властивості LoadBalancedEndpointSetName.

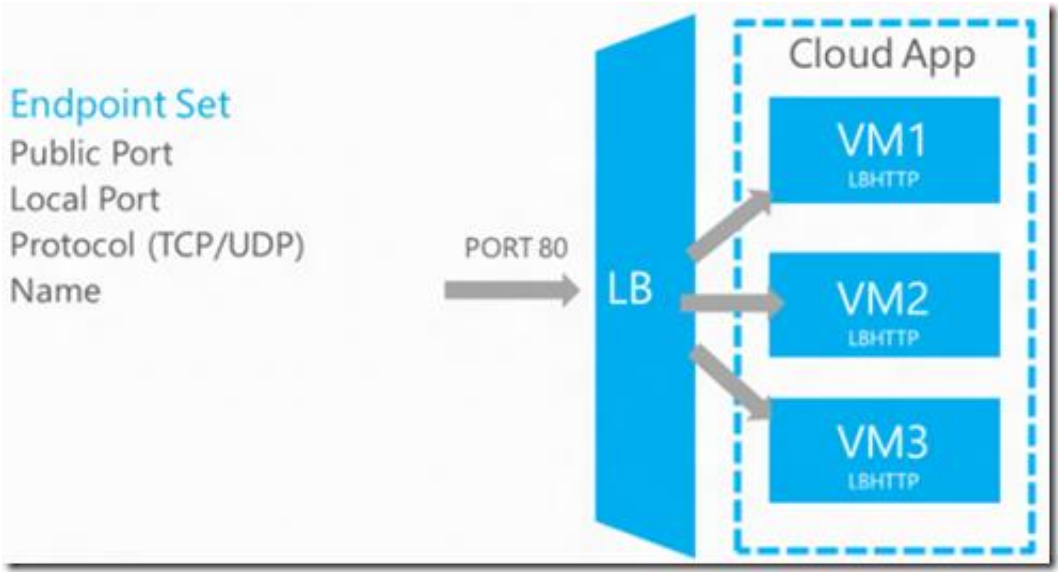


Рис. 5. Балансування навантаження і кінцева точка входу у віртуальну машину

Як ви, напевно, вже побачили на рис. 5, тут є простір для налаштування форвардингу портів. Оскільки кожен Cloud Service має одну публічну IP-адресу, але безліч віртуальних машин усередині, форвардинг портів саме те, що потрібно для того, щоб ззовні отримувати доступ до конкретної машини.

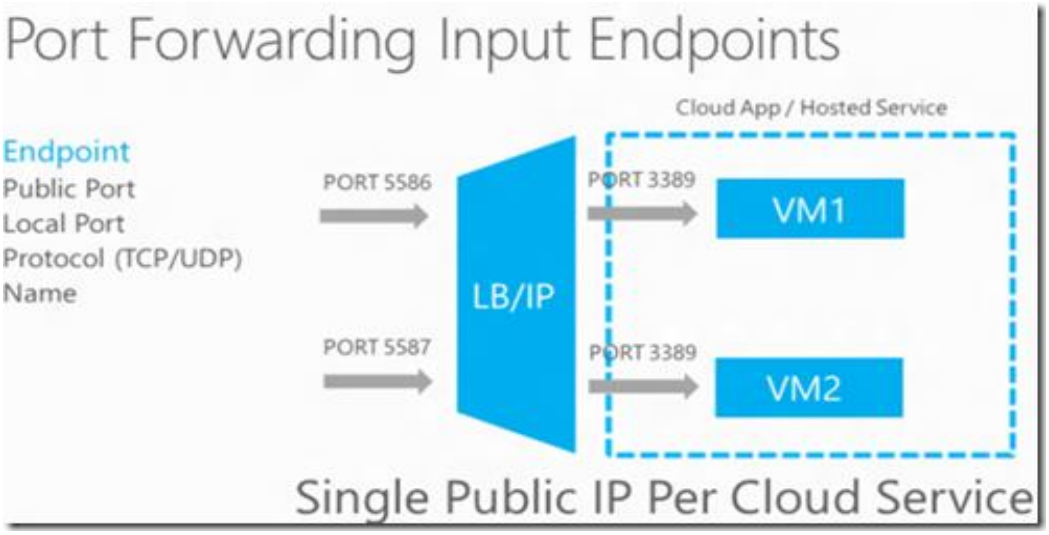


Рис. 6. Форвардинг портів у Windows Azure Virtual Machines

Наприклад, для системи налаштований форвардинг портів (рис. 6) – зовнішній клієнт, який, ініціюючи запит на 5586, переходить на порт RDP (наприклад) у віртуальну машину № 1, ініціюючи ж запит на 5587, він переходить на віртуальну машину № 2 тощо.

5.4. Віртуальні мережі

Найважливішою функцією стала поява віртуальних мереж. Віртуальні мережі (Virtual Networks) – це функціональність, що дає змогу як поєднати вашу локальну інфраструктуру з хмарною, так і налаштувати мережу всередині розгорнутого сервісу.

Якщо з першим сценарієм все більш-менш зрозуміло (необхідний VPN, що підтримує Site-To-Site VPN), то які переваги дає використання віртуальних мереж усередині розгорнутого сервісу?

По-перше, це сценарій постійного IP (не статичного, але постійного). А якщо потрібно перенести Active Directory в Windows Azure, не використовуючи стандартний режим, коли IP змінюватиметься? Тут можна використовувати VNET, за допомогою яких можна визначити загальну схему IP-адресації для вашої мережі хмар. Ви в цьому випадку визначите адресний простір, підмережі й належність віртуальних машин.

Таким чином, кожна віртуальна машина, що розгортається й належить певній VNET, матиме ту саму IP-адресу незалежно від її

стану (перезавантаження, інших дій, що призводять до зміни IP). Нестатичним цей IP є, оскільки він не прописується статикою (безперечний факт), а видається ніби як DHCP, з нескінченним часом лізингу.

При цьому, звичайно ж, може виникати питання про дозвіл на ім'я. Початково жодного дозволу імен у разі поміщення віртуальної машини у віртуальну мережу немає – вважається, що ви самі про це подбаєте.

Тут є три варіанти вирішення проблеми: налаштувати DNS вручну на мережевому адаптері для кожної машини (основний недолік, звичайно ж, у фразі «налаштувати для кожної»), визначити DNS-сервер у конфігурації мережі (що теж незручно, оскільки немає можливості перевизначити DNS-сервер без необхідності перегорнути додані віртуальні машини у віртуальну мережу) та вказати DNS під час первинного розгортання віртуальної машини (наприклад, за допомогою Powershell, що досить зручно).

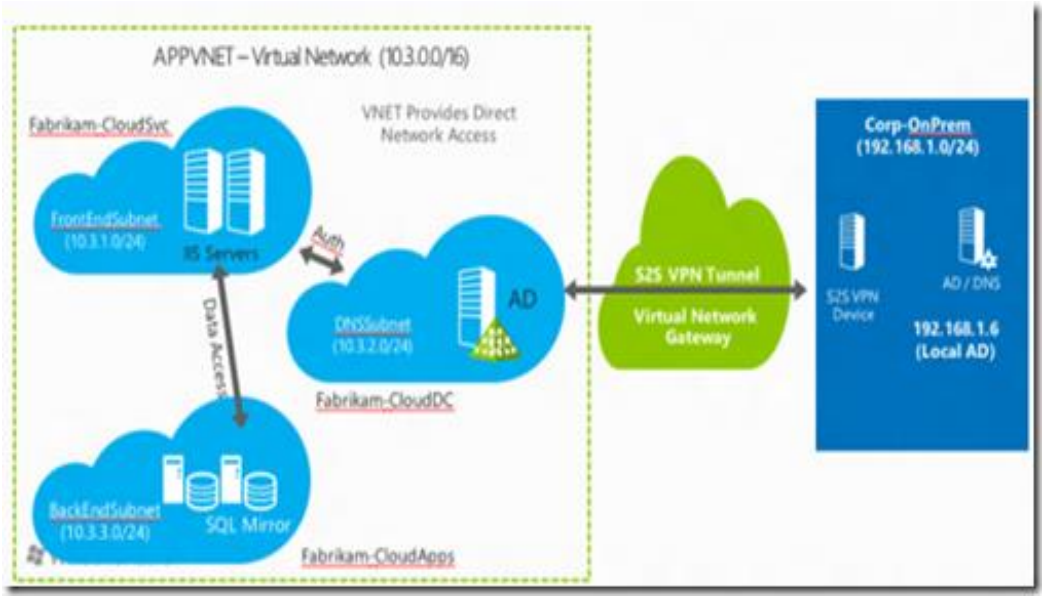


Рис. 7. Гібридне рішення з використанням віртуальних мереж

Для розгортання віртуальної машини на віртуальну мережу (рис. 7) потрібно пам'ятати кілька простих правил:

1) не можна перенести вже розгорнуту віртуальну машину, потрібно розгортати її у віртуальну мережу;

2) налаштування DNS – якщо не планувати все заздалегідь, можна прийти до того, що для вже розгорнутої віртуальної машини не можна буде змінити ці налаштування без перерозгортання;

3) кожній віртуальній мережі потрібна афінна група. Крім цього, обліковий запис сховища повинен перебувати в тому ж регіоні, що і афінна група, або в цій афінній групі.

Доступність віртуальних машин та гарантії.

У Cloud Services SLA як був 99,95 %, так і залишився, якщо враховувати мінімальну кількість екземплярів програми (вона дорівнює двом). З віртуальними машинами ситуація трохи заплутана – було вирішено, що кілька віртуальних машин для більшості додатків не буде потрібно, тому для одного екземпляра пропонується 99,9 % SLA, якщо ж використовується Availability Set, то пропонується 99,95 %.

5.5. Availability Set

Концепція Availability Set аналогічна концепції доменів оновлень і доменів помилок, але дещо розширює її – віртуальні машини в AS фізично розташовані в різних річках (racks) і з оновленням хостової операційної системи оновлюються не всі віртуальні машини в AS одночасно (рис. 8).

Рис. 8. Наочне зображення різних сценаріїв SLA

Що, власне, дає змогу реалізувати відмовостійкість і надмірність на всіх рівнях.

5.6. Практика використання

Існує кілька методів створення віртуальної машини в Windows Azure, і ми розглянемо все.

Віртуальна машина з образу

Власне, найпростіший і зрозуміліший метод – у хмарі вже є галерея образів (рис. 9), у якій зараз підтримується певний набір ОС. Зверніть увагу, що цей список – з Preview-версії, тобто він постійно поповнюватиметься навіть, можливо, перейшовши до Production.

Рис. 9. Галерея образів віртуальних машин

Давайте перейдемо до практики. Увійдіть на портал керування WindowsAzure, використовуючи облікові дані WindowsLiveID.

Створення Microsoft Azure VM для хостингу web-додатків

Windows Azure

Намагаюся розповісти, як можна розширити межі використання Azure на прикладі створення в ньому віртуальної машини та хостингу на ній вебдодатків.

Крок 0. Отримання доступу.

Робиться це на сторінці Головна – Microsoft Azure, де можна вибрати те, що ви хочете спробувати. У цьому випадку вибираємо ***Virtual Machine***.

Крок 1. Створення віртуальної машини.

Після того як ми отримали доступ до компонентів Windows Azure, саме час почати їх використовувати. Для цього виберіть пункт Virtual Machines і внизу сторінки натисніть кнопку плюс. https://portal.azure.com/#blade/HubsExtension/BrowseResourceBlade/resourceType/Microsoft.Compute%2FVirtualMachines. Буде доступно два пункти: **Quick Create та From Gallery**. Насправді різниця між ними невелика, тому розглянемо повніший шлях. Вибираємо **From Gallery**.

У цьому діалоговому вікні ми маємо кілька варіантів на вибір. Почнемо з того, що, крім готових образів ОС (видно на скріншоті), нам також надається можливість завантажити свій власний образ або VHD-диск, з якого згодом створиться віртуальна машина. Це буде зручно для великих проєктів, де потрібно створити ферму однакових машин, не заморочуючись із налаштуванням кожної з них. А нас зараз це не цікавить, тому звернемо свій погляд на доступні готові образи. На вибір пропонується кілька варіантів. Є Windows Server 2008 R2 (двох версій), лінуксові машинки, спеціалізований сервер під SQL Server і герой нашого сьогоднішнього експерименту – Windows Server 2012. Чому саме він? З ознайомлювальною метою було б непогано спробувати і сервер 2012/2016. Отже, вибираємо потрібну нам ОС і тиснемо стрілку знизу. Потрапляємо на такий екран майстра створення віртуальної машини.

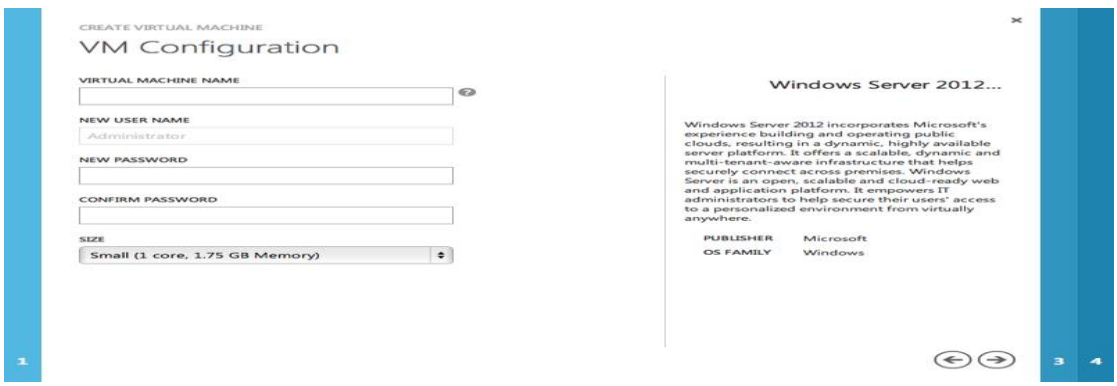


Рис. 10. Початок створення віртуальної машини

Тут усе зрозуміло. Треба вказати ім’я VM, пароль адміністратора та вибрати одну із запропонованих конфігурацій майбутнього «заліза».

Вибір досить непоганий і може покрити більшість вимог щодо продуктивності для різних типів проєкту. Я вибираю Small, що цілком мене влаштовує.

Йдемо далі.

На третьому етапі майстра все теж досить зрозуміло:

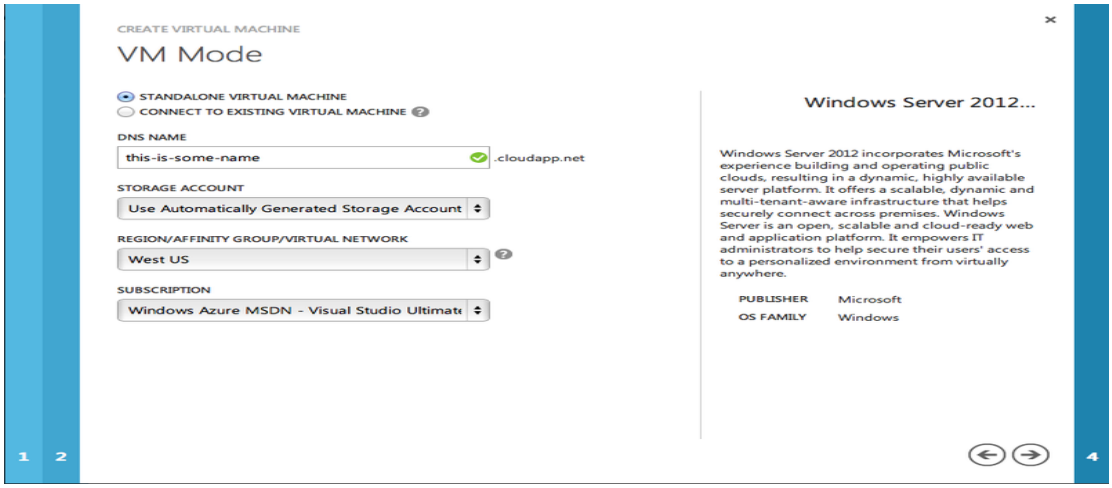


Рис. 11. Приклад заповнення майстра створення віртуальної машини

Для початку треба визначитися із призначенням майбутньої машини. Якщо ми хочемо використовувати її як самостійний ресурс (а зараз ми саме цього і хочемо), то вибираємо **Standalone**. Режим **Connect To**, як написано в підказці, дозволить нам підключити цю ВМ в наявну мережу для поділу навантаження. Нас це поки що не цікавить.

Потім треба придумати ім’я хоста, яким буде доступна машина. Тут треба не плутати ім’я машини на попередньому етапі і це ім’я хоста. Ім’я машини – це ім’я комп’ютера у Windows. Ім’я хоста – це субдомен в домені **cloudapp.net**, за яким здійснюватиметься доступ і управління сервером (рис. 11). Це ім’я має бути унікальним, тому нам тут же в майстрі надається підказка про доступність або зайнятість того чи іншого імені.

Там треба вибрати обліковий запис сховища, розташування ВМ і підписку, в рамках якої ця машина тарифікуватиметься. На четвертому кроці залишаємо все як є.

Натискаємо на галку внизу, і віртуальна машина почне створюватися.

Крок 2. Управління віртуальною машиною.

Після того як ви створили віртуальну машину, вона буде доступна у відповідному списку. Обов'язково дочекайтесь, коли статус машини зміниться з Provisioning на Running. Це означатиме, що машина готова до використання (рис. 12). Якщо натиснути на її ім'я, то відкриється панель керування віртуальною машиною. Внизу сторінки є кнопка Connect. Вона допоможе вам отримати правильний **rdp** файл для підключення до цієї машини за допомогою віддаленого робочого столу. Настав час зайти на новий сервер і покерувати.

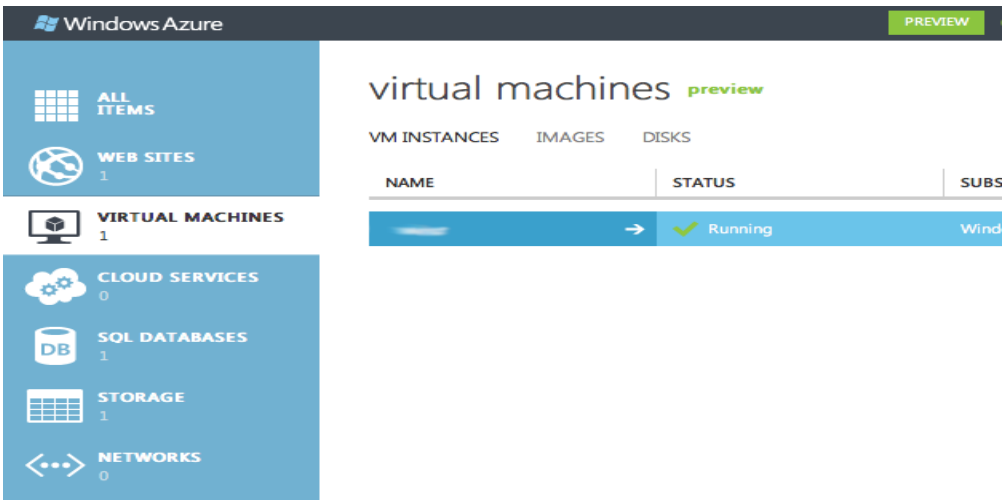


Рис. 12. Машина готова до використання

Крок 3. Керування сервером.

Насправді, я не докладно розповідатиму про адміністрування Windows Server 2012. Оскільки в назві цієї статті згадуються web-додатки, саме цим ми й займемося — встановимо IIS.

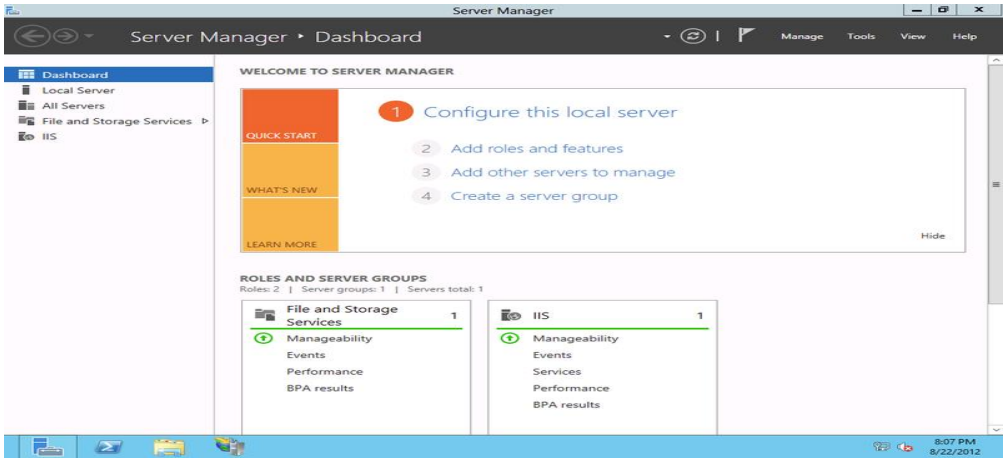


Рис. 13. Робота з Server Manager

Із першим логіном на новий сервер нам відкриється вікно **Server Manager** (рис. 13).

За початковим налаштуванням у Windows Server 2012 служби IIS не включені, тому їх потрібно додати. Вибираємо пункт Add roles and features, після чого нам відкривається майстер додавання ролей. Три рази тиснемо Next (особливо цікаві шанують, що там пишуть, але зараз нас це не цікавить), і в дереві, що відкрилося, вибираємо Web Server (IIS). Далі потрібно обов'язково вибрати потрібні компоненти IIS, які дозволять запускати на ньому програми ASP.NET MVC. Точних параметрів налаштування я писати не буду, кожен сам вирішить, що саме йому потрібно. Обов'язково треба включити ASP.NET, інше до смаку.

Тиснемо кілька разів Next, налаштовуємо майбутню роль, як нам треба, і після завершення процесу встановлення наш сервер із радістю готовий приймати гостей. Переконатись у цьому можна, перейшовши по localhost.

Крок 4. Відкриття доступу.

Усе чудово, ось ми нарешті налагодили вебсервер, розмістили на ньому якусь програму і вона відкривається по localhost. Як же тепер відкрити його ззовні? Насправді, все досить просто. Напевно, ви ще на етапі створення віртуальної машини намагалися натискати на

посилання в панелі управління, яка має вигляд <ваша_вірут_машина>.cloudapp.net.

І, напевно, у вас нічого не відобразилося. Усе тому, що до віртуальної машини закрито доступ ззовні, причому не на рівні фаєрволу операційної системи, а на рівні інфраструктури Windows Azure.

Якщо ви в панелі керування віртуальною машиною подивіться нагору, побачите там три пункти меню: Dashboard, Endpoints і Configure. З першим ми розібралися на самому початку, з останнім все зрозуміло – там налаштовуються параметри «заліза», а що таке Endpoints, давайте розберемося.

У закладці Endpoints ми можемо задавати те, які протоколи та які порти будуть відкриті для цієї віртуальної машини та на які фізичні порти вони будуть перенаправлені (рис. 14):

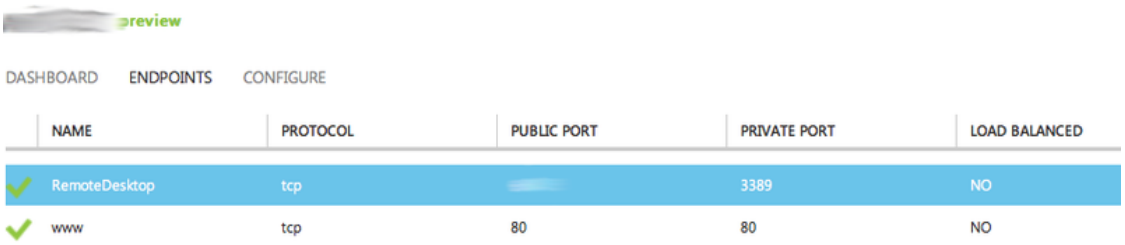


Рис. 14. Перенаправлення протоколів

За початковими налаштуваннями, у цьому списку буде лише один порт — той, яким ви з'єднуєтеся через Remote Desktop. Причому публічний порт може бути будь-яким, що, мабуть, зроблено для більшої безпеки. Для того щоб відкривати вебпрограми на цій віртуальній машині, потрібно відкрити 80-й порт. На скріншоті вище показано. Після цієї дії за посиланням <ваша_вірт_машина>.cloudapp.net ви повинні будете побачити вашу власну програму. Маленької перемоги досягнуто.

Крок 5. Привласнюємо власний домен.

На попередньому кроці ми зробили все, щоб програма запрацювала в хмарі й була доступна ззовні. Однак, погодьтеся, що

було б зовсім не круто, якби ми могли звертатися до нього лише за субдоменом і ніяк не могли б привласнити свій один або кілька доменів. Звичайно ж, це зробити можна й досить просто. Для цього офіційна документація радить два різні шляхи, і обидва пов'язані з редагуванням зони DNS. Перший має на увазі те, що ви створюєте CNAME запис, у якому вказуєте, що, наприклад, www.mydomain.com перенаправлятиметься на mywinazure.cloudapp.net. Цей метод простий, але надто прямолінійний. Якщо у вас на сервері розміщено кілька вебдодатків або на основному домені може бути кілька субдоменів, то цей спосіб не спрацює. Благо, що є другий, якісніший спосіб. Як рішення, яким, до речі, скористався і я, пропонується редагування запису A в зоні DNS. На панелі керування віртуальною машиною, праворуч, ви могли помітити дві IP-адреси. Одна називається приватною і є IP самої машини, а друга – публічна, віртуальна (Virtual IP, VIP). Ось саме її і треба вказувати в зоні A для домену, що редагується. У цьому випадку весь трафік посилатиметься на вашу віртуальну машину, де IIS визначатиме домен і видаватиме відповідну програму (за умови правильного налаштування Web Sites у самому IIS). Усе просто й очевидно, але саме так воно й працює:

- windows azure,
- azure,
- microsoft,
- asp.net

5.7. Microsoft Azure для веброзробника

Про те, навіщо веброзробнику хмара, у яких завданнях вона допомагає, а у яких ні, написано вже багато – переважно виділяють можливість швидко масштабуватися. Не автоматично, але з певним ступенем гнучкості – для вебсайтів це зазвичай значення завантаження ЦП, з перевищенням якої система ініціює виклик внутрішніх API для розгортання репліки.

Друга за популярністю – оплата за фактом використання. Скільки система відпрацювала, за стільки прийшов рахунок.

Можна назвати ще багато причин, але перейдемо до суті – подивимося, що є на платформі Microsoft Azure для веброзробника.

Коли виникає тема веброзробки на Microsoft Azure, зазвичай дивляться відразу на віртуальні машини. Вибір правильний, проте не завжди економічний – через надання клієнту практично повного контролю над тим, що відбувається з екосистемою навколо проєкту, виникає як потреба в налагодженні, управлінні та подальшій підтримці (наприклад, у налагодженні відмовостійких ферм вебсерверів або СУБД), так і список пов'язаних із цим витрат.

Як свідчить досвід, вибір віртуальних машин обґрунтований у двох випадках:

• потрібно перенести систему, яка має стільки різних версій, що питання зміни кодової бази може навіть не розглядатися. Варіант «віртуалізували готовий сервер із додатком та екосистемою і перенесли у хмару на віртуальну машину» тут найочевидніший;

• потрібен тюнінг екосистеми.

Microsoft Azure має сервіс віртуальних машин, які офіційно підтримують образи, підготовлені для Windows Server, починаючи з 2008 R2, і деяких дистрибутивів Linux. Під час розгортання віртуальної машини в неї інжектуються спеціальні сервіси, які роблять всі бенефіти хмари реальними, спілкуючись із системними сервісами платформи й у разі потреби виробляючи «лікування» машини, перенесення даних та інші операції. Робити з ВМ можна практично все, що можна робити з машиною, локально створеною, проте потрібно враховувати те, що всі ресурси віртуалізовані, а диски розташовані по-різному – диск C:/ перебуває в Azure Storage, що накладає обмеження на продуктивність файлової системи, оскільки диск, по суті, розташований на відстані, а диск D:/ локальний для сервера і також має свою особливість – він тимчасовий. У разі перезавантаження чи будь-якої іншої операції з ВМ цей диск буде очищений. Другий варіант – Cloud Services, який був по суті родоначальником платформних сервісів Azure. Суть CS зводиться до того, що веброзробник бере (або починає новий) вебпроєкт на .NET або Java, поділяє його на фронтенд (Web-роль) і бекенд (Worker-роль) і кладе ці проєкти в рішення. Вказує для Web-ролі та Worker-ролі ресурси, які виділятимуться, і завантажує рішення в Azure.

Контроль над тим, що відбувається в CS, не такий, як із віртуальною машиною, і стан зберігати тут складніше, проте багатьох це влаштовує – наприклад, коли ми розуміємо, що на вебінтерфейс у

нас йде велике навантаження, а на обробник немає, сенсу в масштабуванні всієї системи немає, тому ми просто збільшуємо кількість екземплярів Web-ролі. Можна це зробити і з віртуальними машинами, але ручної праці буде набагато більше (плюс це рішення буде дорожчим). До речі, крім значення CS як контейнера додатків, CS ще виступають як мережний кордон, на якому відкриваються / закриваються порти та ін. Тому не видаляйте CS, який використовується для віртуальної машини — він, по-перше, безплатний, по-друге, це гарантований спосіб порушити функціонування системи.

Третій варіант – Web Apps. Раніше називалися Web Sites. Найпростіше рішення для розміщення вебзастосунків. Відмінність від BM та CS — мінімум контролю та налаштувань. На відміну від CS, усе йде в одному пакеті, без можливості масштабувати окремо шари програми.

Зручно, коли потрібно розгорнути вебсайт без змін (за винятком деяких ситуацій, наприклад використання старих версій PHP Runtime) кодової бази або просто в швидкому режимі. Якщо ж у вас немає вебсайту, але потрібно швидко розгорнути й подивитися якесь рішення a-la Drupal, цей сервіс теж може допомогти – є галерея образів різного ПЗ. Крім цього порівняння, на цій сторінці багато сценарних підходів. Наприклад, My application depends on highly customized Windows or Linux environments and I want to move it to the cloud. А далі? Розмістили… Розгорнули сайт, налаштували розгортання із GitHub, навчилися масштабувати. Що далі? Зі зростанням проєкту може зростати кількість його компонентів і модулів. В Azure, як хмарній платформі, є набір сервісів, кожен з яких може бути практично застосований для вирішення того чи іншого завдання замість самостійної реалізації та підтримки.

Сервіси Microsoft Azure діляться на три логічні шари – шар обчислень, на базі сервісів якого працюють практично всі інші сервіси, шар інтеграції – для мобільних додатків, для забезпечення комунікацій між різними компонентами проєкту, для захисту й аутентифікації та ін., та шар роботи з даними – усі послуги, які дають змогу різними способами взаємодіяти та змінювати стан даних. Розглянемо все це коротко.

Шар доступу до даних

Шар доступу до даних містить нереляційні сховища даних: таблиці, диски, черги, зберігання двійкових об'єктів, DocumentDB + реляційне сховище даних як SQL Database.

Вибір налаштувань сховища, очевидно, залежить від завдань, і кожен із сервісів має власну специфіку.

Наприклад, типовим використанням таблиць є зберігання логів (сама Microsoft Azure за активації логування програми складає дані саме туди), SQL Database актуальний, коли є необхідність швидко розгорнути базу даних або протестувати рішення.

Таблиці – у ній зберігаються структуровані нереляційні дані.

Черги забезпечують обмін повідомленнями між програмами**.**

Блоби дають можливість зберігати великі обсяги неструктурованих текстових чи двійкових даних.

SQL Database – реляційна база даних – це масштабована хмарна служба бази даних, побудована на основі технологій SQL Server. Повторюся: якщо потрібно швидко розгорнути базу даних, це чудовий і найшвидший варіант. Також усередині цього сервісу є різна цікава функціональність, наприклад інструмент для створення non-clustered індексів (докладніше від sitox).

Крім SQL Database, на платформі є ще кілька пропозицій – MySQL та MongoDB, аналогічно доступні як сервіси.

Azure Files дає можливість звертатися до даних сховища Azure Storage як до мережного ресурсу за протоколом SMB, що дає змогу здійснювати звичний доступ до даних із віртуальних машин через мережеву взаємодію. Це означає, що можна використовувати стандартні Windows API.

Caching – розподілений кеш у пам'яті, за допомогою якого замість повільного дискового сховища програми отримують високошвидкісний доступ до даних, що зберігаються в оперативній пам'яті, з можливістю масштабування**.**

Cache на базі Redis – сервіс Azure Redis Cache є готовим redis-сховищем з необхідним розміром для завдань кешування даних**.**

Шар інтеграції

Media Services охоплюють хмарні версії багатьох існуючих технологій платформи мультимедіа Microsoft і партнерів, у тому числі

для перегляду, кодування, перетворення формату й захисту контенту, а також потокової передачі на запит і в реальному часі. Підтримує **Live Streaming.**

Цей сервіс рекомендується використовувати для обробки мультимедіа контенту. Можна робити це і на віртуальних машинах (наприклад, ставити VoIP або медіасервер), однак у цьому випадку можна зіткнутися з різними обмеженнями (як-от обмежена кількість портів, трафік та ін.).

Mobile Services пропонує хмарну інфраструктуру для всіх найпопулярніших мобільних платформ**: Windows 8+, Windows Phone 8+, iOS, Android, HTML 5.**

На основі сервісу можна побудувати хмарний бекенд, на який перенести завдання зберігання даних, аутентифікації та Push-повідомлень. Підтримується Xamarin. Використовується багатьма клієнтами.

Identity – служба аутентифікації, забезпечує керування посвідченнями та доступом до програм, за допомогою служби **Microsoft Azure Active Directory (колишній Access Control Service)** можна забезпечити єдиний вхід, підвищену безпеку та просту взаємодію з уже розгорнутими в Active Directory додатками, а також здійснити інтеграцію з іншими провайдерами аутентифікації **(Live ID, Google, Facebook** тощо**).**

Azure Active Directory дає змогу вирішувати завдання єдиної авторизації користувачів для багатьох сервісів (Single Sign On), вести єдиний каталог користувачів, синхронізувати дані каталогу з Active Directory на підприємстві тощо.

Azure Active Directory – це повноцінна реалізація каталогу в хмарі. Сервіс підтримує популярні відкриті стандарти федерацій: **SAML 2.0, OData, WS-FED, OAuth 2.0/OpenID.**

Service Bus надає можливості ретрансляції та безпечного обміну повідомленнями та дає змогу створювати розподілені та слабопов’язані додатки у хмарі, а також гібридні додатки, розміщені одночасно в приватних і загальнодоступних хмарних службах. Набагато потужніше і функціональніше, ніж Storage Queues, отже, складніше в підтримці та виборі правильних технологій і підходів.

Крім звичайної черги, містить спеціальну чергу Event Hubs, яка актуальна за використання в концепції інтернету речей, коли багато (дуже) пристроїв, що посилають велику кількість невеликих повідомлень.

Traffic Manager – трафік-менеджер, забезпечує балансування навантаження за вхідного трафіку між декількома розміщеними службами Windows Azure незалежно від того, чи працюють вони в одному центрі обробки даних, чи розподілені за кількома.

API Management пропонує розробникам власних API можливість отримати оточення з управління, моніторингу й адміністрування свого API, розміщеного в будь-якому місці, як у хмарі, так і на будь-якому хостингу, включно з власною інфраструктурою.

Крім цих сервісів є ще, наприклад, **Application Insights** – ви інтегруєте агента в додаток або сервер і маєте повноцінну історичну картину того, що відбувається з вашим проєктом.

Загалом, якщо потрібно вибрати правильний сервіс, можна скористатися сайтом-конструктором.

5.8. Контрольні запитання до теми 5

1. Які можливості Microsoft Azure?
2. Microsoft Azure – назвіть п’ять типів віртуальних машин.
3. Microsoft Azure належить до категорії PaaS?
4. Віртуальні машини. Які можливості Azure?
5. Microsoft Azure підтримує модель розгортання віртуальних машин?
6. Microsoft Azure, яка веброль?
7. Microsoft Azure, яка прикладна роль?
8. У Microsoft Azure користувачам доступні такі функції… розкажіть детальніше.
9. Екземпляр в Microsoft Azure є одиницею розгортання, детальніше…
10. Які особливості використання Microsoft Azure?
11. Пакетна служба Azure –це…
12. Планувальник Azure – це…

Тема 6. Amazon Web Services

6.1. Amazon Web Services (AWS)

Це набір обчислювальних вебсервісів, які становлять обчислювальну хмарну платформу, представлену компанією Amazon на початку 2006 року.

Amazon Web Services (AWS) – це публічна хмарна платформа, що надається компанією Amazon.

AWS належить до класу IaaS-рішень і надає широкий спектр хмарних сервісів.

У цій інфраструктурі представлені сервіси для надання різних послуг: зберігання даних (файловий хостинг, розподілені сховища даних), оренда віртуальних серверів, надання обчислювальних потужностей та ін. (рис. 15).

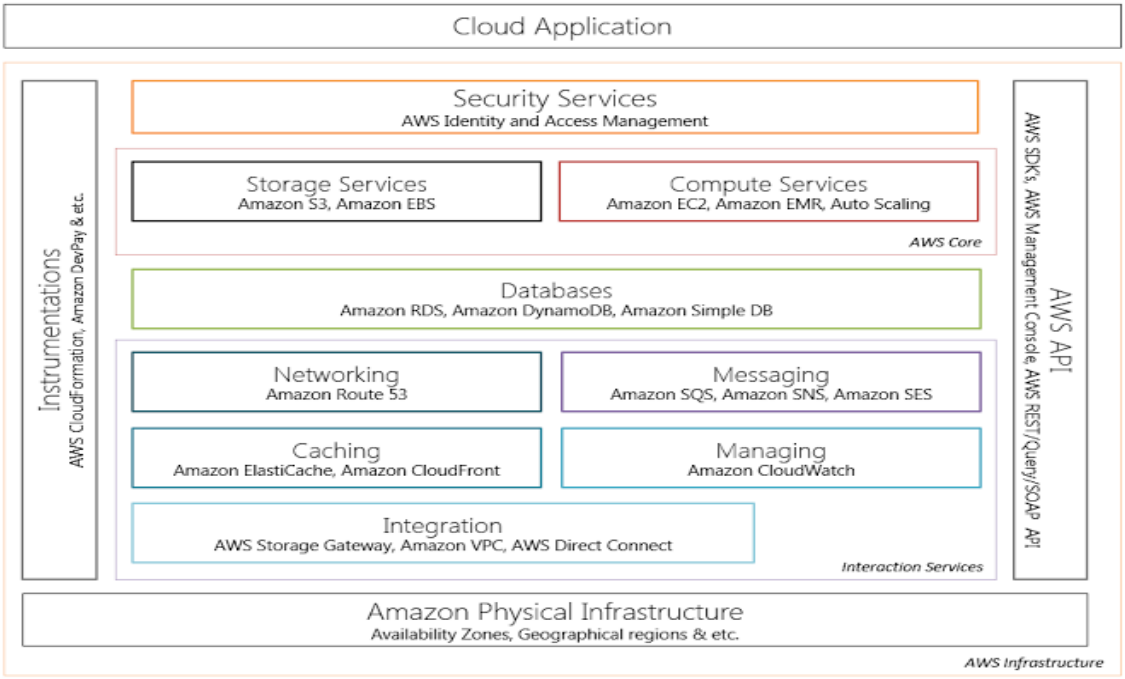


Рис. 15. Сервіси для надання послуг

Найбільш відомими є дві платформи:

- **Amazon EC2**,
- **Amazon S3**.

6.2. Amazon Elastic Compute Cloud (Amazon EC2)

Це вебсервіс, який надає обчислювальні потужності в хмарі.

Сервіс входить в інфраструктуру **Amazon Web Services**. Вебінтерфейс сервісу дає змогу отримати доступ до обчислювальних потужностей і налаштувати доступ з мінімальними витратами ресурсів. Він надає користувачам повний контроль над обчислювальними ресурсами, а також доступне середовище для роботи.

За допомогою EC2 можна:

- створити Amazon Machine Image (AMI), який міститиме додатки, бібліотеки, дані й пов'язані з ними конфігураційні параметри. Або використовувати налаштовані шаблони образів для роботи;
- завантажити AMI в Amazon S3. Amazon EC2 надає інструменти, для зберігання AMI. Amazon S3 забезпечує безпечне, надійне і швидке сховище для зберігання образів;
- використовувати Amazon EC2. Вебсервіс для налаштування безпеки й мережевого доступу;
- вибирати тип (-и) операційної системи, запустити, завершити або контролювати кілька AMI за потреби, використовуючи API вебсервісу або різних інструментів управління, які були для цього передбачені;
- визначити необхідність працювати в декількох місцях, використовувати статичний IP або інші варіанти;
- платити тільки за потрібні ресурси.

6.3. Amazon S3

Amazon Simple Storage Service (Amazon S3) – онлайн-вебслужба, пропонована Amazon Web Services, що надає можливість для зберігання й отримання будь-якого обсягу даних, у будь-який час, із будь-якої точки мережі, так званий файловий хостинг.

- За допомогою ***Amazon S3*** досягається масштабованість, надійність, висока швидкість і бюджетна інфраструктура зберігання даних. Уперше з'явилася в березні 2006 року в США і в листопаді 2007 року у Європі.

- ***Amazon S3*** використовується багатьма іншими сервісами для зберігання та хостингу файлів. Наприклад, сервіси зберігання й обміну файлів ***Dropbox і Ubuntu One, вебсайт Twitter, завантажувач гри Minecraft.***
- ***Amazon S3*** є хмарної системою зберігання даних для інтернету.
- ***Amazon S3*** надає простий інтерфейс вебсервісів, які можна використовувати для зберігання й отримання будь-якої кількості даних, у будь-який час, із будь-якої точки мережі.

Це дає розробнику доступ до тієї ж надійної, безпечної, швидкої, недорогої інфраструктури, що Amazon використовує для запуску власну глобальну мережу вебсайтів.

6.4. Хмарні сервіси Amazon Web Services

- Amazon AWS Authentication – перевірка достовірності доступу до різних сервісів.
- Amazon CloudFront – мережа доставки контенту (CDN) для доставки об’єктів в локації, наближені до запиту.
- Amazon CloudWatch забезпечує моніторинг хмарних ресурсів і додатків AWS, починаючи з EC2.
- Amazon DevPay – система білінгу й управління акаунтами для додатків, які побудовані на Amazon Web Services.
- Amazon Elastic Beanstalk забезпечує швидке розгортання й управління додатками в хмарі.
- Amazon Elastic Block Store (EBS) забезпечує постійні томи зберігання даних для EC2.
- Amazon Elastic Compute Cloud (EC2) надає масштабовані віртуальні приватні сервери за допомогою Xen.
- Amazon Elastic MapReduce дає змогу підприємствам, дослідникам, аналітикам і розробникам легко та дешево обробляти великі обсяги даних. Він використовує фреймворк Hadoop, що працює на вебмасштабованій інфраструктурі EC2 і Amazon S3.
- Amazon ElastiCache надає кешування в пам’яті для вебдодатків.

- Amazon Flexible Payments Service (FPS) надає інтерфейс для мікроплатежів.
- Amazon Fulfillment Web Service надає програмний вебсервіс для продавців, щоб можна було завантажувати товари в Amazon і назад із використанням сервісу Fulfillment.
- Amazon Glacier забезпечує низьку вартість зберігання на основі рішення для зберігання даних S3.
- Amazon Mechanical Turk (Mturk) управляє невеликими одиницями роботи, розподіленими між багатьма людьми.
- Amazon Product Advertising API, раніше відомий як Amazon Associates Web Service (A2S) і Amazon E-Commerce Service (ECS), забезпечує доступ до даних щодо продуктів Amazon і функціональності електронної комерції.
- Amazon Relational Database Service (RDS) надає масштабований сервер баз даних із підтримкою MySQL і Oracle.
- Amazon Route 53 забезпечує доступність і масштабованість системи доменних імен (DNS).
- Amazon CloudSearch надає базовий пошук по всьому тексту й індексацію текстового вмісту.
- Amazon Simple Service Email (SES) виконує відправлення електронної пошти.
- Amazon Simple Storage Service (S3) забезпечує вебсервіс для зберігання даних.
- Amazon Simple Service Queue (SQS) забезпечує черговість повідомлень для вебдодатків.
- Amazon Simple Service Notification (SNS) забезпечує мультипротокольний «поштовх» повідомлень для вебдодатків.
- Amazon Simple Workflow (SWF) – сервіс для побудови масштабованих, стійких додатків.
- Amazon SimpleDB дає можливість розробникам виконувати запити до структурованих даних. Він працює спільно з EC2 і S3, щоб забезпечити «базову функціональність бази даних».
- Amazon Virtual Private Cloud (VPC) створює логічно об’єднаний набір елементів Amazon EC2, який може бути підключений до наявної мережі, використовуючи з’єднання VPN.

- AWS CloudFormation створює набір пов’язаних ресурсів AWS і надає їх в упорядкованому й передбачуваному вигляді.
- AWS Import. Export прискорює переміщення великих обсягів даних в AWS і назад за допомогою переносних пристроїв для зберігання.
- AWS Management Console (AWS Console) – консоль для управління й моніторингу інфраструктури Amazon, включно з EC2, EBS, Amazon Elastic MapReduce, Amazon CloudFront.
- AWS Storage Gateway – пристрій зберігання з ISCSI-протоколом, який підтримує резервне копіювання в хмарному середовищі.

6.5. Рівень безплатного користування AWS

Рівень безплатного користування ***Amazon Web Services (AWS)*** дає можливість отримати практичний досвід роботи із хмарними сервісами AWS. Рівень безплатного користування AWS поширюється на сервіси з безплатним рівнем користування, чинним протягом 12 місяців з дати реєстрації, а також на додаткові сервісні пропозиції, які не припиняють свою дію після закінчення 12-місячного періоду безплатного користування AWS.

Після створення облікового запису AWS у вас буде безплатний доступ (з певними обмеженнями) до продукту й сервісу з наведеного нижче списку.

Ви можете почати роботу сьогодні й автоматично скористатися рівнем безплатного користування AWS, виконавши наступні кроки:

1. Зареєструйте обліковий запис AWS.

2. Укажіть адресу виставлення рахунка та дані кредитної картки. Оплата почнеться тільки після перевищення обмежень рівня безкоштовного користування.

3. Виберіть будь-який продукт зі списку й почніть роботу із хмарними сервісами AWS.

Почніть працювати з AWS безплатно. Платіть тільки за те, чим користуєтеся.

Amazon EC2

Масштабований обсяг обчислювальних ресурсів в хмарі.

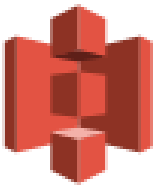
○ 750 годин на місяць використання інстанси t2.micro з Linux, RHEL або SLES.

○ 750 годин на місяць використання інстанси t2.micro з Windows.

Наприклад, ви можете виконувати 1 інстанси протягом 1 місяця або 2 інстанси протягом половини місяця.

○ Втрачає чинність через 12 місяців після реєстрації.

Amazon S3



Безпечна, надійна і масштабована інфраструктура сховища об’єктів.

○ 5 ГБ стандартного сховища.

○ 20 000 запитів Get.

○ 2000 запитів Put.

○ Втрачає чинність через 12 місяців після реєстрації.

Amazon RDS

Керований сервіс реляційних баз даних для MySQL, PostgreSQL, MariaDB, Oracle BYOL і SQL Server.

○ 750 годин використання бази даних на місяць.

○ 20 ГБ сховища БД: будь-яке поєднання універсальних томів (SSD) і магнітних томів.

○ 20 ГБ для резервних копій (магнітне сховище RDS; операції введення / виведення для універсальних томів (SSD) додатково не оплачуються).

○ 10 000 000 операцій введення-виведення Втрачає чинність через 12 місяців після реєстрації.

AWS IoT

Нова пропозиція – ***підключення пристроїв до хмари***.

Термін «інтернет речей» (IoT) уперше ввів в обіг Кевін Ештон, британський технологічний новатор і дослідник технологій радіохвильової ідентифікації (RFID). Під «інтернетом речей» він розумів систему повсюдних сенсорів, яка б пов'язала реальний світ та інтернет.

Основними трьома елементами IoT є речі, інтернет і система підключень, але його основна перевага – можливість створювати системи самовдосконалення, які здатні стерти кордони між фізичними й цифровими світами.

Які проблеми можна буде вирішити, якщо знати стан кожної речі у світі та вміти ефективно обробляти ці дані?

Щоб допомогти відповісти на це питання й отримати вигоду від використання підключених пристроїв, AWS розробила спеціалізовані сервіси IoT, як-от AWS Greengrass і AWS IoT. Вони допомагають збирати дані та відправляти їх в хмару, завантажувати й аналізувати інформацію і управляти пристроями, даючи змогу користувачам зосередитися на розробці необхідних додатків.

Щоб почати роботу з інтернетом речей на AWS, виберіть сервіс для докладного вивчення:

- ***AWS Greengrass***
- ***Платформа AWS IoT***
- ***Кнопка AWS IoT***

○ 250 000 повідомлень (опублікованих або доставлених) на місяць.

○ Втрачає чинність через 12 місяців після реєстрації.

Amazon DynamoDB

Швидка гнучка і ефективно масштабована ***база даних NoSQL***.

○ 25 ГБ сховища.

○ 25 одиниць ресурсів записи.

○ 25 одиниць ресурсів читання.

○ Підтримка до 200 млн запитів на місяць.

○ Не припиняє дії після закінчення 12-місячного періоду безплатного користування AWS.

Amazon Redshift

Швидке, просте й економічне зберігання даних.

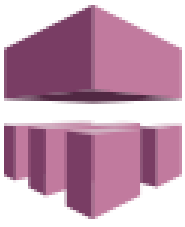
Детальніше:

○ Безплатний пробний доступ протягом 2 місяців.

○ 750 годин роботи інстанси DC1.Large в місяць протягом 2 місяців.

○ Втрачає чинність через 2 місяці після реєстрації.

AWS Device Farm



Тестуйте додатки для iOS, Android і FireOS на реальних пристроях в хмарі AWS.

○ Безплатна разова пробна версія надає 250 хвилин використання пристроїв.

○ Не припиняє дії після закінчення 12-місячного періоду безкоштовного користування AWS.

AWS Key Management Service

AWS Key Management Service – це керований сервіс, що забезпечує шифрування даних і засоби управління ім.

○ 20 000 запитів на місяць безплатно.

○ Не припиняє дії після закінчення 12-місячного періоду безкоштовного користування AWS.

Amazon CloudWatch

Нова пропозиція – рівень безкоштовного користування AWS.

Моніторинг хмарних ресурсів і додатків AWS.

○ 10 призначених для користувача метрик і 10 попереджень.

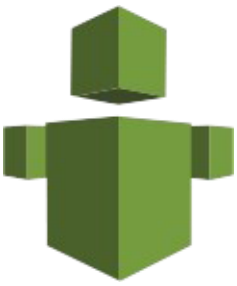
○ 1 000 000 запитів API.

○ 5 ГБ для імпорту даних балок і 5 ГБ для архіву даних логів.

○ 3 панелі стану, до 50 метрик у кожної на місяць.

○ Не припиняє дії після закінчення 12-місячного періоду безкоштовного користування AWS.

Trusted Advisor



Мережевий ресурс, який дасть вам змогу знизити витрати, підвищити продуктивність і безпеку завдяки оптимізації середовища AWS.

Він надає в режимі реального часу рекомендації AWS, які допоможуть вам ефективно розподілити ресурси.

AWS Trusted Advisor – персональний експерт із хмарних технологій, який аналізує середу AWS і надає рекомендації в п’яти категоріях:

- оптимізація витрат,
- відмовостійкість,
- продуктивність,
- ліміти сервісів,
- безпека.

Незалежно від того, що ви робите: створюєте нові робочі процеси, оптимізуєте поточні або розробляєте програми, – рекомендації, які регулярно надає Trusted Advisor, дають змогу оптимально виділяти ресурси для ваших рішень.

Повний список перевірок і описів див. в рекомендаціях щодо роботи з Trusted Advisor.

○ 4 рекомендації для перевірки продуктивності й безпеки (обмеження сервісів, групи безпеки, IAM і MFA).

○ Можливості оповіщення та індивідуального налаштування.

○ Термін дії закінчується через 12 місяців після реєстрації.

Amazon QuickSight

Нова пропозиція – рівень безкоштовного користування AWS.

Швидкий і простий в управлінні хмарний сервіс бізнес-аналітики в десять разів дешевший за традиційні аналітичні рішення.

Amazon QuickSight – це швидка хмарна служба бізнес-аналітики, яка спрощує створення візуалізацій, проведення спеціального аналізу та швидке отримання інформації про бізнес на основі ваших даних. Використовуючи цей хмарний сервіс, ви можете легко підключатися до своїх даних, виконувати розширений аналіз і створювати приголомшливі візуалізації та багатофункціональні інформаційні панелі, до яких можна отримати доступ із будь-якого браузера чи мобільного пристрою.

Amazon QuickSight Overview

○ 4 рекомендації для перевірки продуктивності й безпеки (обмеження сервісів, групи безпеки, IAM и MFA).

○ Можливості оповіщення й індивідуального налаштування.

○ Термін дії закінчується через 12 місяців після реєстрації.

AWS CodeBuild

Нова пропозиція – рівень безкоштовного користування AWS.

Повністю керований сервіс збірки, який виконує складання та тестування програмного коду в хмарі.

○ 100 хвилин збірки за місяць використання обчислювального інстанси типу build.general1.small.

○ Не припиняє дії після закінчення 12-місячного періоду безкоштовного користування AWS.

Amazon Cloud Directory

Нова пропозиція – ***керований сервіс створення каталогів на базі графів із вбудованою перевіркою даних.***

Amazon Cloud Directory дає змогу створювати гнучкі хмарні каталоги для організації ієрархічних зв’язків даних за безліччю напрямів.

За допомогою Cloud Directory можна створювати каталоги для безлічі прикладів використання, як-от схеми організації, каталоги курсів або реєстри пристроїв, тоді як традиційні рішення для каталогів, наприклад Active Directory Lightweight Directory Services (AD LDS) та інші каталоги з використанням полегшеного протоколу доступу (LDAP), дають можливість зберігати лише одну ієрархічну структуру, Cloud Directory надає гнучкі можливості для створення каталогів з ієрархічними зв’язками за безліччю напрямів.

Наприклад, можна створити схему організації з різними можливостями навігації, в основі яких лежатимуть ієрархічні зв’язку за структурою підзвітності, місцезнаходженням і центрами витрат.

Amazon Cloud Directory автоматично масштабується до сотень мільйонів об’єктів і підтримує розширену схему, з якої одночасно може працювати безліч додатків.

Повністю керований сервіс Cloud Directory позбавляє користувачів від рішення трудомістких і дорогих адміністративних завдань, як-от масштабування інфраструктури й управління серверами. Досить просто визначити схему, створити каталог і приступити до його заповнення за допомогою викликів API Cloud Directory.

Amazon Cloud Directory автоматично масштабується.

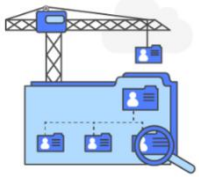
○ 1 ГБ сховища на місяць.

○ 10 000 запитів запису на місяць.

○ 100 000 запитів читання на місяць.

○ Термін дії закінчується через 12 місяців після реєстрації.

Amazon Chime



Нова пропозиція – рівень безкоштовного користування AWS.

Amazon Chime – це сучасний сервіс об’єднаних комунікацій, що спрощує проведення нарад і забезпечує неперевершену якість аудіо- та відеозв’язку.

○ Необмежене використання Amazon Chime Basic.

○ Підписка Amazon Chime Basic безплатна протягом будь-якого терміну.

○ Не припиняє дії після закінчення 12-місячного періоду безкоштовного користування AWS.

Amazon GameLift

Нова пропозиція – рівень безкоштовного користування AWS.

Простий, швидкий і економічний виділений хостинг ігрових серверів.

○ 125 годин використання інстанси c3.large.gamelift сервісу Amazon GameLift на місяць.

○ 50 ГБ Місце EBS на універсальних томах (SSD).

○ Втрачає чинність через 12 місяців після реєстрації.

6.6. Контрольні запитання до теми 6

1. Які є платформи Amazon?
2. Розкажіть про Amazon CloudFront.
3. Розкажіть про Amazon CloudWatch.
4. Розкажіть про Amazon EC2. Масштабований обсяг обчислювальних ресурсів в хмарі.
5. Розкажіть про Amazon Elastic Compute Cloud (Amazon EC2) – функціональність.
6. Розкажіть про Amazon Elastic Compute Cloud (EC2).
7. Розкажіть про Amazon RDS та керований сервіс реляційних баз даних.
8. Розкажіть про Amazon S3. Це безпечна, надійна та масштабована інфраструктура сховища об’єктів?
9. Розкажіть про Amazon Simple Storage Service (Amazon S3) – функціональність.
10. Amazon Web Services (AWS) – визначення.
11. Що таке AWS CloudFormation?
12. Що ви знаєте про AWS IoT та підключення пристроїв до хмари?
13. Розкажіть про AWS Management Console (AWS Console) та хмарні сервіси Amazon Web Services.
14. Основні особливості AWS (Amazon WEB Services).

Список джерел

1. Сафонов В. О. Архітектура, можливості та методи використання платформи хмарних обчислень Microsoft Windows Azure [Електронний ресурс]. – Режим доступу: https://www.facultyresourcecenter.com/ curriculum/ru/pfv.aspx?ID=8866&c1=ua-ua&c2=UA. – Назва з екрана.
2. Федоров А. Windows Azure: хмарна платформа Microsoft / [Електронний ресурс]. – Режим доступу: https://www.facultyresourcecenter.com/curriculum/ua/pfv.aspx?ID=8673&c1=ua-ua&c2=UA. – Назва з екрана.
3. Огляд першого українського хмарного сервісу UTOO. [Електронний ресурс]. – Режим доступу: https://ain.ua/2013/04/24/obzor-pervogo-ukrainskeho-oblachnoho-servisa-utoo. – Назва з екрана.
4. Створіть безкоштовний обліковий запис Azure вже сьогодні. [Електронний ресурс]. – Режим доступу: https://azure.microsoft.com/ua/free/. – Назва з екрана (дата звернення 16.10.2024).
5. Інформація про рівень безкоштовного користування AWS [Електронний ресурс]. – Режим доступу: https://cloud.google.com/products/app-engine. – Назва з екрана (дата звернення 10.10.2024).
6. Офіційний сайт Microsoft, на якому розміщена документація по роботі із платформою Microsoft Azure [Електронний ресурс]. – Режим доступу: http://azure.microsoft.com/ua-ua. – Назва з екрана (дата звернення 10.10.2024).
7. Медведовський І. Програмні засоби перевірки та створення політики безпеки, що відповідає вимогам міжнародного стандарту управління інформаційною безпекою ISO 17799. [Електронний ресурс]. – Режим доступу: http://nt.com.ua/info/dsec/politics.shtml. – Назва з екрана (дата звернення 10.10.2024).
8. Навчальний центр «Мережеві технології» [Електронний ресурс]. – Режим доступу: http://www.nt.com.ua/about/pr.shtml. – Назва з екрана (дата звернення 10.10.2024).
9. Moodle Statistics // Moodle. [Електронний ресурс]. – Режим доступу: http://moodle.org/stats. – Назва з екрана (дата звернення 10.10.2024).

Для нотаток

Навчальне видання

\

ХРОЛЕНКО Володимир Миколайович,
ГОЛЕНКОВ Володимир Геннадійович

Хмарні та GRID-технології

Навчальний посібник

Редагування та коректура *Т. В. Івченко*
Комп'ютерне верстання *Л. В. Лабунець*

Підписано до друку 17.02.2025. Формат $60\times84_{1/16}$
Ум. друк. арк. 5,81. Обл.-вид. акр. 6,25.
Тираж 25 прим. Вид. № 1/І-25. Зам. № 2/1-25

Видавець і виготовлювач
Київський національний університет будівництва і архітектури

Проспект Повітряних Сил, 31, Київ, 03037

Свідоцтво про внесення до Державного реєстру суб'єктів
видавничої справи ДК № 808 від 13.02.2002