

Питання освітньої компоненти «GRID-системи та хмарні технології».

1. Що таке snapshots у SMB/Windows/Azure File Share, як вони створюються та як працює доступ до попередніх версій файлів?
2. Як працює доступ до попередніх версій файлів (Previous Versions) у SMB-шерінгу?
3. Які переваги та обмеження Lift-and-Shift міграції файлових серверів у хмару?
4. Які обмеження існують у Azure File Share для SMB-протоколу?
5. Як Python SDK дозволяє працювати з Azure File Share: створення файлів, папок і snapshot'ів?
6. Які відмінності між File Share та Blob Storage в Azure для сценаріїв з SMB?
7. Як відбувається синхронізація серверів за допомогою Azure File Sync Agent?
8. Що таке Cloud Tiering у Azure File Sync і як воно працює?
9. Як забезпечується консистентність даних між локальним сервером і Azure File Share?
10. Які метрики важливо моніторити у File Sync-інфраструктурі (latency, sync status, journal size тощо)?
11. Як масштабувати файлове сховище у Azure File Share при зростанні обсягу даних або навантаження?
12. Які типові проблеми можуть виникнути під час синхронізації через Azure File Sync та як їх діагностувати?
13. Які основні відмінності між Blob Storage, File Storage, Queue Storage та Table Storage в Azure?
14. Що таке blob-об'єкт і які типи blob'ів існують?

15. У яких випадках доцільно використовувати Azure File Storage замість Blob Storage?
16. Для яких сценаріїв призначені Queue Storage та Table Storage?
17. Які типові бізнес-задачі можна вирішити за допомогою Block blobs, Append blobs та Page blobs?
18. У чому полягає різниця між локально надлишковим сховищем (LRS) та зонально надлишковим сховищем (ZRS)?
19. Які переваги надає географічно надлишкове сховище (GRS) порівняно з LRS та ZRS?
20. Чому варто враховувати затримки та вартість при виборі варіанту копіювання даних між регіонами?
21. У яких випадках GRS може бути критично важливим для бізнесу?
22. Чим відрізняються рівні доступу Hot, Cool і Archive за вартістю та продуктивністю?
23. Які критерії використовуються для вибору відповідного рівня доступу до даних?
24. Що відбувається з даними, коли їх переводять у рівень Archive, і які обмеження це створює?
25. Що таке «soft delete» у Azure Blob Storage і як воно захищає дані від випадкового видалення?
26. Чим відрізняється soft delete для blob'ів, контейнерів та версій об'єктів?
27. Які інтерфейси й інструменти можна використовувати для роботи з Azure Blob Storage (наприклад, REST API, SDK, Azure CLI, PowerShell) та які їхні переваги?
28. Що таке Storage Account в Azure і яку роль він відіграє в роботі з Queue Storage?

29. Які основні характеристики Azure Queue (розмір повідомлення, кількість черг, типи черг)?
30. Яку структуру має Message у Queue Storage та які обмеження накладаються на його вміст і розмір?
31. Як працює механізм visibility timeout після отримання повідомлення з черги?
32. У чому різниця між Peek та Dequeue операціями в Azure Queue Storage?
33. Які переваги використання Azure Logic Apps для обробки повідомлень з Azure Queue Storage?
34. Який тригер використовується в Logic Apps для автоматичного зчитування нових повідомлень з черги?
35. Як можна реалізувати обробку повідомлень у циклі в Logic Apps при великому потоці даних?
36. Що відбувається з повідомленням після того, як Logic App успішно його опрацює?
37. Як налаштувати error handling у Logic Apps при виникненні помилки під час обробки повідомлення з черги?
38. Яку бібліотеку Python використовується для роботи з Azure Queue Storage, та як її встановити?
39. Як створити клієнт для черги за допомогою QueueServiceClient або QueueClient?
40. Яким способом у Python SDK можна додати повідомлення до черги й які параметри можна передати?
41. Як отримати та видалити повідомлення з черги за допомогою Python SDK?
42. Яким чином можна оновити існуюче повідомлення в Azure Queue Storage через Python?

43. Що таке Entity в Azure Table Storage і з яких елементів вона складається?
44. Для чого використовується PartitionKey і як він впливає на продуктивність запитів?
45. Яку роль виконує RowKey, та чому разом із PartitionKey він формує унікальний ключ сутності?
46. Що таке Timestamp і хто відповідає за його оновлення — клієнт чи система?
47. Які обмеження та правила існують для Properties у сутності Azure Table Storage?
48. Що таке точковий запит (Point Query) і чому він вважається найшвидшим?
49. У яких випадках використовують діапазонний запит (Range Query) та які поля можуть у ньому фігурувати?
50. Як працює фільтрація в Azure Table Storage і які оператори порівняння доступні?
51. Чому сортування в Azure Table Storage обмежене та у яких випадках можна отримати дані в певному порядку?
52. Чому складні комбіновані запити (AND/OR) можуть бути менш ефективними в порівнянні з простими точковими?
53. У яких випадках Azure Table Storage є кращим вибором порівняно з реляційними базами даних?
54. Як Azure Table Storage підходить для зберігання даних телеметрії або логів?
55. Які переваги має Table Storage для сценаріїв з високою масштабованістю та великим обсягом даних?
56. Які типові обмеження треба враховувати при проєктуванні моделі даних в Azure Table Storage?

57. Як Table Storage використовується у зв'язці з іншими Azure сервісами, наприклад Azure Functions або Logic Apps?
58. Які існують рівні узгодженості в Azure Cosmos DB та чим вони відрізняються?
59. Чому рівень Strong Consistency забезпечує найвищу узгодженість, але впливає на продуктивність і глобальну доступність?
60. У яких сценаріях доцільно використовувати Eventual Consistency, і які ризики це створює для читання даних?
61. Що означає властивість read-your-own-writes і для яких рівнів узгодженості вона гарантується?
62. У чому полягає суть режиму Provisioned Throughput Mode, і як вимірюється пропускна здатність Cosmos DB?
63. Які переваги та недоліки використання заздалегідь виділеного пропускного режиму (Provisioned Throughput) для стабільних навантажень?
64. Як працює Serverless Mode, і в яких випадках він є оптимальним вибором?
65. Чому Serverless Mode може бути дорожчим для високих постійних навантажень порівняно з Provisioned Throughput?
66. Що таке RU (Request Units) і як їх споживання відрізняється між цими двома режимами?
67. Що таке логічна секція (logical partition) і яку роль відіграє Partition Key?
68. У чому полягає відмінність між логічною та фізичною секцією (physical partition)?
69. Яким чином Cosmos DB автоматично масштабує фізичні секції при збільшенні обсягу даних або навантаження?
70. Які критерії слід враховувати під час вибору ефективного Partition Key?

71. Які основні об'єкти становлять ієрархію SDK azure.cosmos (наприклад, CosmosClient, Database, Container) і яку роль виконує кожен із них?
72. Як відбувається взаємодія з документами за допомогою контейнера (Container) у бібліотеці azure.cosmos?
73. Яким чином створюється підключення до Cosmos DB у Python через CosmosClient та які параметри є обов'язковими?
74. Що таке регіон Azure, і чому їх географічне розташування важливе для продуктивності та відповідності вимогам?
75. Чим відрізняються зони доступу (Availability Zones) від регіонів, і які гарантії відмовостійкості вони надають?
76. Яку роль відіграють пари регіонів (Region Pairs) та які переваги забезпечують у сценаріях аварійного відновлення?
77. Які ключові параметри потрібно визначити під час створення віртуальної машини в Azure (наприклад, розмір, OS, диск, мережа)?
78. Чим відрізняються типи дисків Standard HDD, Standard SSD та Premium SSD, і як вибір впливає на продуктивність?
79. Які параметри безпеки (SSH, паролі, мережеві правила) необхідно враховувати під час розгортання VM?
80. Які інструменти доступні в Azure Cloud Shell і чим відрізняються Bash та PowerShell режими?
81. Які основні команди Azure CLI використовують для створення віртуальної машини через Cloud Shell?
82. Чому Azure Cloud Shell вважається зручним інструментом для автоматизації та навчання адмініструванню Azure?
83. Що таке Fault Domain (домени збою) та як вони захищають VM від одночасного апаратного збою?

84. Яку роль виконують Update Domains (домени оновлення) та чому вони важливі під час планових оновлень інфраструктури?
85. Як Availability Set використовує Fault та Update Domains для підвищення доступності додатків?
86. Які типи DNS-зон підтримує Azure DNS та для чого використовується кожна з них?
87. Як працює делегування DNS-зони між реєстратором домена та Azure DNS?
88. Які переваги надає Azure DNS порівняно з традиційними DNS-серверами, зокрема у масштабованих хмарних системах?
89. Що таке Azure IoT Hub та яку роль він виконує в IoT-архітектурі?
90. Які типи комунікацій підтримує IoT Hub?
91. Чим IoT Hub відрізняється від Event Hubs з точки зору IoT-сценаріїв?
92. Які основні протоколи підтримує IoT Hub (MQTT, AMQP, HTTPS) і коли їх використовують?
93. Як IoT Hub забезпечує автентифікацію та безпеку пристроїв?
94. Що таке вбудована кінцева точка (Built-in Endpoint) в Azure IoT Hub?
95. Для чого використовують маршрутизацію повідомлень у IoT Hub?
96. Які властивості повідомлення використовуються для умовної маршрутизації (message routing query)?
97. До яких сервісів Azure можна спрямувати повідомлення через маршрути (Event Hubs, Service Bus, Storage)?
98. У чому різниця між default endpoint та custom endpoints?
99. Як перевірити, що маршрутизація повідомлень працює коректно (message routing test tool)?
100. Яку роль відіграє Azure Event Hubs у прийомі та обробці потоків подій?

101. Що таке partitions та consumer groups в Event Hubs, і чому вони важливі для масштабованої обробки даних?
102. Що таке Azure IoT Central і які ключові компоненти входять до його структури?
103. Яку роль у структурі IoT Central відіграє Application Template?
104. Що таке Device Template та як він впливає на моделювання пристроїв?
105. Які основні розділи містить інтерфейс Azure IoT Central (Dashboard, Devices, Analytics, Administration)?
106. Для чого використовується панель Dashboard і яку інформацію там можна відобразити?
107. Які функції доступні в розділі Devices інтерфейсу IoT Central?
108. Як у IoT Central створюються та налаштовуються візуалізації телеметрії (charts, tiles)?
109. Як Azure IoT Central взаємодіє з IoT Hub у фоновому режимі?
110. Яким чином у IoT Central реалізується масштабування та ізоляція ресурсів?
111. За що відповідає рівень Data Export в архітектурі IoT Central?
112. Які основні відмінності між IoT Central та IoT Hub з точки зору архітектури та керування?
113. Які типи операцій можна виконувати над пристроєм у IoT Central (перегляд телеметрії, зміна параметрів, оновлення)?
114. Що таке Commands у керуванні пристроями і як вони працюють?
115. Як у IoT Central реалізовано відстеження стану пристрою (Device Properties, Cloud Properties, Device Status)?