



# AZ-204<sup>Q&As</sup>

Developing Solutions for Microsoft Azure

## Pass Microsoft AZ-204 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.passapply.com/az-204.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft  
Official Exam Center

- ⚙ **Instant Download** After Purchase
- ⚙ **100% Money Back** Guarantee
- ⚙ **365 Days** Free Update
- ⚙ **800,000+** Satisfied Customers





## QUESTION 1

### DRAG DROP

You are developing an Azure solution.

You need to develop code to access a secret stored in Azure Key Vault.

How should you complete the code segment? To answer, drag the appropriate code segments to the correct locations. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes

or scroll to view content.

NOTE: Each correct selection is worth one point.

Select and Place:

#### Code segments

DefaultAzureCredential

ClientSecretCredential

CloudClients

SecretClient

#### Answer Area

```
string var1 = Environment.GetEnvironmentVariable("KEY_VAULT_URI");  
var var2 = new Code segment ( new Uri(var1), new Code segment ());
```

Correct Answer:

#### Code segments

ClientSecretCredential

CloudClients

#### Answer Area

```
string var1 = Environment.GetEnvironmentVariable("KEY_VAULT_URI");  
var var2 = new SecretClient ( new Uri(var1), new DefaultAzureCredential ());
```

Box 1: SecretClient

Box 2: DefaultAzureCredential

In below example, the name of your key vault is expanded to the key vault URI, in the format "https://.vault.azure.net". This example is using `DefaultAzureCredential()` class from Azure Identity Library, which allows to

use the same code across different environments with different options to provide identity.

```
string keyVaultName = Environment.GetEnvironmentVariable("KEY_VAULT_NAME");
```

```
var kvUri = "https://" + keyVaultName + ".vault.azure.net";
```



```
var client = new SecretClient(new Uri(kvUri), new DefaultAzureCredential());
```

Reference:

<https://docs.microsoft.com/en-us/azure/key-vault/secrets/quick-create-net>

---

## QUESTION 2

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while

others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are developing an Azure Service application that processes queue data when it receives a message from a mobile application. Messages may not be sent to the service consistently.

You have the following requirements:

1.

Queue size must not grow larger than 80 gigabytes (GB).

2.

Use first-in-first-out (FIFO) ordering of messages.

3.

Minimize Azure costs.

You need to implement the messaging solution.

Solution: Use the .Net API to add a message to an Azure Service Bus Queue from the mobile application. Create an Azure Windows VM that is triggered from Azure Service Bus Queue.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

Don't use a VM, instead create an Azure Function App that uses an Azure Service Bus Queue trigger.

Reference: <https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function>

---

## QUESTION 3



Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while

others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are developing an Azure Service application that processes queue data when it receives a message from a mobile application. Messages may not be sent to the service consistently.

You have the following requirements:

1.

Queue size must not grow larger than 80 gigabytes (GB).

2.

Use first-in-first-out (FIFO) ordering of messages.

3.

Minimize Azure costs.

You need to implement the messaging solution.

Solution: Use the .Net API to add a message to an Azure Service Bus Queue from the mobile application. Create an Azure Function App that uses an Azure Service Bus Queue trigger.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

You can create a function that is triggered when messages are submitted to an Azure Storage queue.

Reference: <https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-storage-queue-triggered-function>

---

#### QUESTION 4

You are developing a mobile instant messaging app for a company.

The mobile app must meet the following requirements:

Support offline data sync.

Update the latest messages during normal sync cycles.

You need to implement Offline Data Sync.



Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Retrieve records from Offline Data Sync on every call to the PullAsync method.
- B. Retrieve records from Offline Data Sync using an Incremental Sync.
- C. Push records to Offline Data Sync using an Incremental Sync.
- D. Return the updatedAt column from the Mobile Service Backend and implement sorting by using the column.
- E. Return the updatedAt column from the Mobile Service Backend and implement sorting by the message id.

Correct Answer: BE

B: Incremental Sync: the first parameter to the pull operation is a query name that is used only on the client. If you use a non-null query name, the Azure Mobile SDK performs an incremental sync. Each time a pull operation returns a set of results, the latest updatedAt timestamp from that result set is stored in the SDK local system tables. Subsequent pull operations retrieve only records after that timestamp.

E (not D): To use incremental sync, your server must return meaningful updatedAt values and must also support sorting by this field. However, since the SDK adds its own sort on the updatedAt field, you cannot use a pull query that has its own orderBy clause.

References: <https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-offline-data-sync>

---

## QUESTION 5

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are developing an Azure solution to collect point-of-sale (POS) device data from 2,000 stores located throughout the world. A single device can produce 2 megabytes (MB) of data every 24 hours. Each store location has one to five devices that send data.

You must store the device data in Azure Blob storage. Device data must be correlated based on a device identifier. Additional stores are expected to open in the future.

You need to implement a solution to receive the device data.

Solution: Provision an Azure Service Bus. Configure a topic to receive the device data by using a correlation filter.

Does the solution meet the goal?

- A. Yes
- B. No

Correct Answer: A



A message is raw data produced by a service to be consumed or stored elsewhere. The Service Bus is for high-value enterprise messaging, and is used for order processing and financial transactions.

Reference: <https://docs.microsoft.com/en-us/azure/event-grid/compare-messaging-services>

[Latest AZ-204 Dumps](#)

[AZ-204 PDF Dumps](#)

[AZ-204 Braindumps](#)