

# Exam Alert: Connect To and Consume Azure Services and Third-Party Services

---

## PREPARING FOR THE EXAM



**David Tucker**

TECHNICAL ARCHITECT & CTO CONSULTANT

@\_davidtucker\_ davidtucker.net

Up Next:  
Objectives for the Exam

---

# Objectives for the Exam

---

# Consume Azure and Third-Party Services

**15-20%**

**Implement API  
Management**

**Develop Event-  
based Solutions**

**Develop Message-  
based Solutions**

# Implement API Management

**Create an APIM instance**

**Configure authentication for APIs**

**Define policies for APIs**

Develop  
Event-based  
Solutions

**Implement solutions that use Azure Event  
Grid**

**Implement solutions that use Azure Event  
Hub**

Develop  
Message-based  
Solutions

**Implement solutions that use Azure  
Service Bus**

**Implement solutions that use Azure Queue  
Storage queues**

# Review API Management Implementation

---



# Areas of Focus

**Service  
Tiers**

**Caching**

**Access Restriction and  
Authentication**

**Policy  
Definition**

# API Management Pricing Tiers

**Consumption**

**Developer**

**Basic**

**Standard**

**Premium**

**Isolated\***

# API Management Cache Types

## **Internal**

Cache provided within the API Management service

## **External**

Redis compatible cache outside of API Management, such as Azure Cache for Redis

# API Management Caching

**Internal cache is limited in size based on the API Management tier**

**Internal cache is not available on the consumption tier for API Management**

**Both types of caching are configured in the API Management policies**

# Controlling Access to API's

## **Access Restriction**

Limiting access to an API based on specific settings

## **Authentication**

Verify credentials for a caller of an API

# Access Restrictions

**Checking an HTTP header for existence  
and value**

**Limit call rate by subscription and key**

**Restrict by IP address**

**Usage quotas per key**

**Validate JWT**

# Authentication Policies

**Basic Auth**

**Client Certificate  
Auth**

**Managed Identity  
Auth**

## API Management Policy

```
<policies>
  <inbound>
    <base />
    <cache-lookup vary-by-developer="false"
      vary-by-developer-groups="false" caching-type="internal">
      <vary-by-query-parameter>v</vary-by-query-parameter>
    </cache-lookup>
  </inbound>
  <outbound>
    <cache-store duration="60" />
    <base />
  </outbound>
</policies>
```



Review Policy  
Structure

**Authentication and JWT handling**  
**Cache configuration**  
**Access restrictions**

# Review Event-based App Development

---

# Areas of Focus

**Understanding Messages  
and Events**

**Selecting an Event-based  
Service**

# Comparing Events and Messages

## Events

**Lightweight notification of a state change**

**Publisher does not know (or care) how the message is handled**

**Follows a publisher/subscriber model**

## Messages

**Application data from a source system to be consumed elsewhere**

**There is an expectation that a message will be handled by a receiver**

**Can follow either a publisher/subscriber or a producer/consumer model**

# Event Types

## **Discrete Events**

Report state change from a system and enable subscribers to take action

## **Series Events**

Report a condition and enable subscribers to analyze a condition over time

# Selecting an Event-based Service

1

Does your solution have an expectation of how data is handled or does it contain app data? If so, select a Messaging service.

2

Do you need a solution to send events to mobile devices as push notifications? **Select Azure Notification Hub.**

3

Does your solution produce discrete events, that report state changes that a system can act on? **Select Azure Event Grid.**

4

Does your solution report state over time for analysis by another system, such as in a data pipeline? **Select Azure Event Hub.**

# Review Message-based App Development

---

# Areas of Focus

**Services  
Overview**

**Interacting with  
Services using the  
CLI**

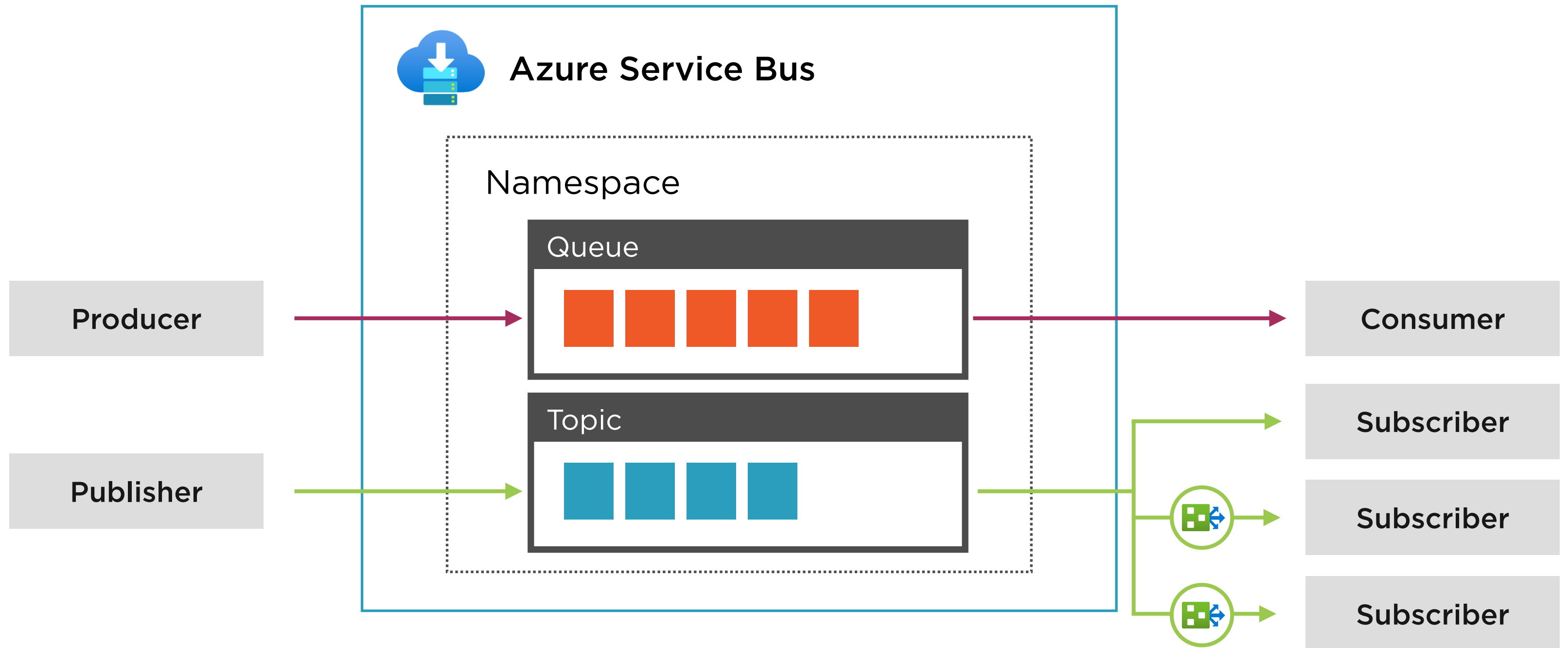
**Selecting a  
Messaging Service**



# Sample Queue Storage Architecture



# Organization of Azure Service Bus





# Azure Service Bus Topic Filters

## **Topic filters can be specified as:**

- Boolean filters: specifies that all or none of the messages are selected
- SQL filters: a SQL-like expression to evaluate against message properties
- Correlation filters: matched against properties in the message

```
# create a queue
az storage queue create --name mysamplequeue

# delete a queue
az storage queue delete --name mysamplequeue

# view messages in a queue (without affecting visibility)
az storage message peek --queue-name mysamplequeue

# delete all messages in a queue
az storage message clear --queue-name mysamplequeue
```

# Interacting with Queue Storage using the CLI

Azure CLI

```
# create a topic
az servicebus topic create --namespace-name pluralsight
--name testtopic --resource-group pluralsight

# delete a topic
az servicebus topic delete --namespace-name pluralsight
--name testtopic

# create a subscription
az servicebus topic subscription create --namespace-name pluralsight
--name testsub --topic-name testtopic
```

# Interacting with Service Bus Topics using the CLI

Azure CLI

# Azure Queue Storage Use Cases

**Total storage for queue needs to be over  
80 GB**

**Logs needed for all transactions executed  
against queue**

**Need to track progress of message  
processing**

# Azure Service Bus Use Cases

**Need support for receiving messages without polling** (with AMQP 1.0)

**There is a need to guarantee message processing order** (FIFO)

**There is a need to detect duplicate messages**

**You need to support messages up to 256 KB**

**You may need to support topic based notifications** (one to many)

**You need to support publishing and consuming in batches**

# Example Scenarios

---



# Scenario 1



**Sylvia is implementing an API on API Management with the Standard tier**

**She is configuring the built-in cache so calls are cached per unique user**

**Unique users are identified based on a JWT token in the Authorization header**

**How should she configure the caching?**

## API Management Policy

```
<policies>
  <[1]>
    <base />
    <cache-lookup vary-by-developer="false"
      vary-by-developer-groups="false" caching-type="[3]">
      <vary-by-header>[4]</vary-by-header>
    </cache-lookup>
  </[1]>
  <[2]>
    <cache-store duration="300" />
    <base />
  </[2]>
</policies>
```

# Scenario 2



**Edward's company is creating a new SaaS application on Azure**

**The solution analyzes IoT temperature sensors in food storage facilities**

**The data is ingested and then made available in a data pipeline**

**What approach would he take to build this solution?**

# Scenario 3



**Cindy is developing an architecture for an order processing application**

**She is creating the application in a modular manner**

**She is planning to leverage Azure Functions to process the orders**

**She will be using Azure Event Grid to handle the orders as discrete events**

**Does Cindy's architecture fit the use case?**

# Scenario 4



**Oscar's is creating a new single-page application using React**

**He needs to validate users for the application's API**

**He wants to use Azure AD with OAuth 2 authorization**

**He also needs to verify that the JWT is a token for his specific application**

**How should Oscar configure the policy in API Management?**

## API Management Policy

```
<policies>
  <inbound>
    <[1] header-name="Authorization"
      failed-validation-error-message="Unauthorized">
      <[2] url="https://login.microsoftonline.com/{aad-tenant}/.well-known/openid-configuration" />
      <[3]>
        <claim name="aud">
          <value>{Application ID of backend-app}</value>
        </claim>
      </[3]>
    </[1]>
  </inbound>
  ...
</policies>
```

# Scenario 5



**James's company has created a fantasy football platform**

**On draft day, all player acquisitions are handled in a queue**

**All acquisitions must be handled in order**

**Given the volume, it is estimated that the queue could grow to 10 GB**

**What service should James leverage for processing these acquisitions?**

# Scenario Answers

---



# Scenario 1



**Sylvia is implementing an API on API Management with the Standard tier**

**She is configuring the built-in cache so calls are cached per unique user**

**Unique users are identified based on a JWT token in the Authorization header**

**How should she configure the caching?**

## API Management Policy

```
<policies>
  <inbound>
    <base />
    <cache-lookup vary-by-developer="false"
      vary-by-developer-groups="false" caching-type="internal">
      <vary-by-header>Authorization</vary-by-header>
    </cache-lookup>
  </inbound>

  <outbound>
    <cache-store duration="300" />
    <base />
  </outbound>
</policies>
```

## Scenario 2



**Edward's company is creating a new SaaS application on Azure**

**The solution analyzes IoT temperature sensors in food storage facilities**

**The data is ingested and then made available in a data pipeline**

**What approach would he take to build this solution?**

**Solution: Utilize Azure Event Hub to analyze the Series Events from the sensors**

## Scenario 3



**Cindy is developing an architecture for an order processing application**

**She is creating the application in a modular manner**

**She is planning to leverage Azure Functions to process the orders**

**She will be using Azure Event Grid to handle the orders as discrete events**

**Does Cindy's architecture fit the use case?**

**Solution: No. She is using an event service for a message-based use case.**

# Scenario 4



**Oscar's is creating a new single-page application using React**

**He needs to validate users for the application's API**

**He wants to use Azure AD with OAuth 2 authorization**

**He also needs to verify that the JWT is a token for his specific application**

**How should Oscar configure the policy in API Management?**

## API Management Policy

```
<policies>
  <inbound>
    < validate-jwt header-name="Authorization"
      failed-validation-error-message="Unauthorized">
      < openid-config url="https://login.microsoftonline.com/{aad-
tenant}/.well-known/openid-configuration" />
        < required-claims >
          <claim name="aud">
            <value>{Application ID of backend-app}</value>
          </claim>
        </ required-claims >
      </ validate-jwt >
    </inbound>
    ...
</policies>
```

# Scenario 5



**James's company has created a fantasy football platform**

**On draft day, all player acquisitions are handled in a queue**

**All acquisitions must be handled in order**

**Given the volume, it is estimated that the queue could grow to 10 GB**

**What service should James leverage for processing these acquisitions?**

**Solution: Azure Service Bus**

# Next Steps

---



Next Video Course:  
Preparing to Take the AZ-204 Exam

---