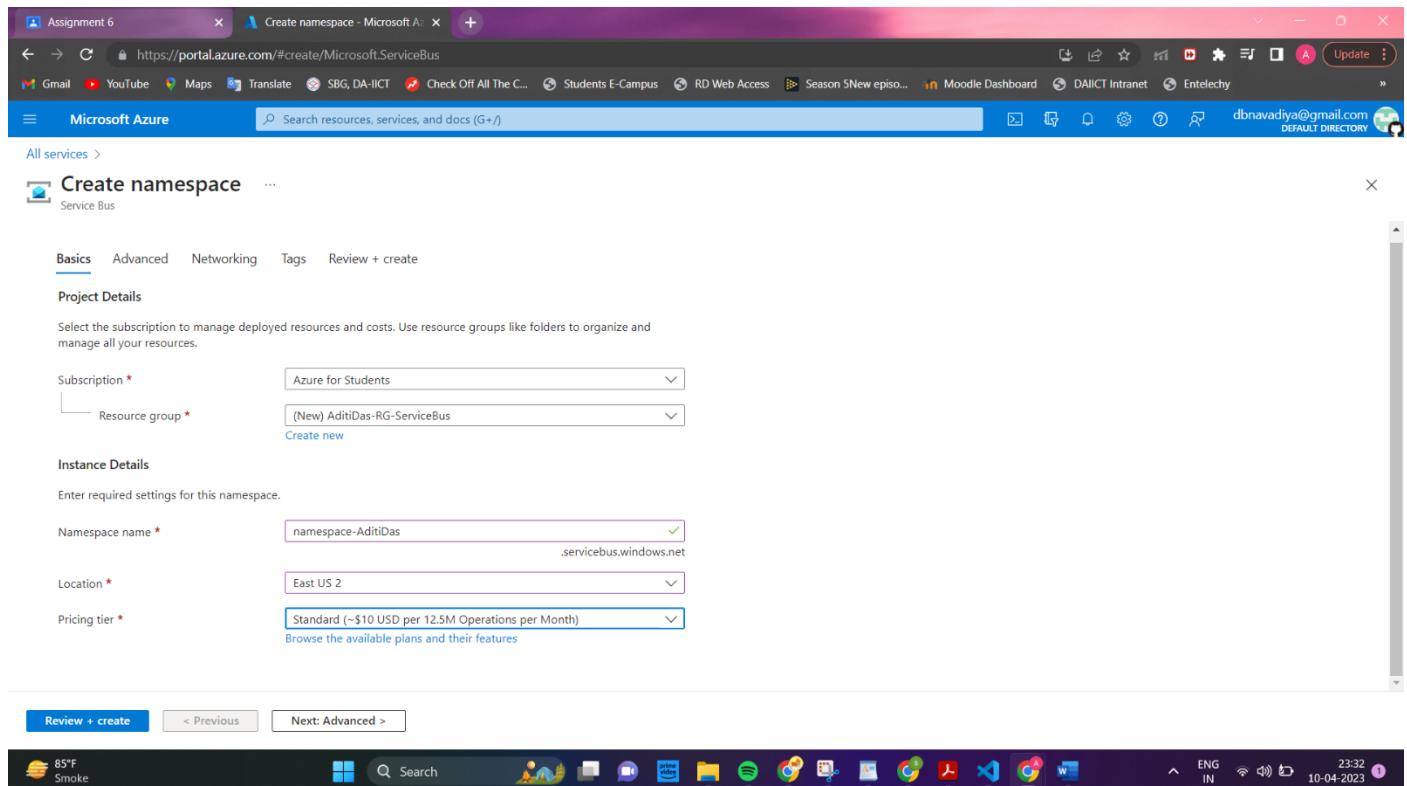
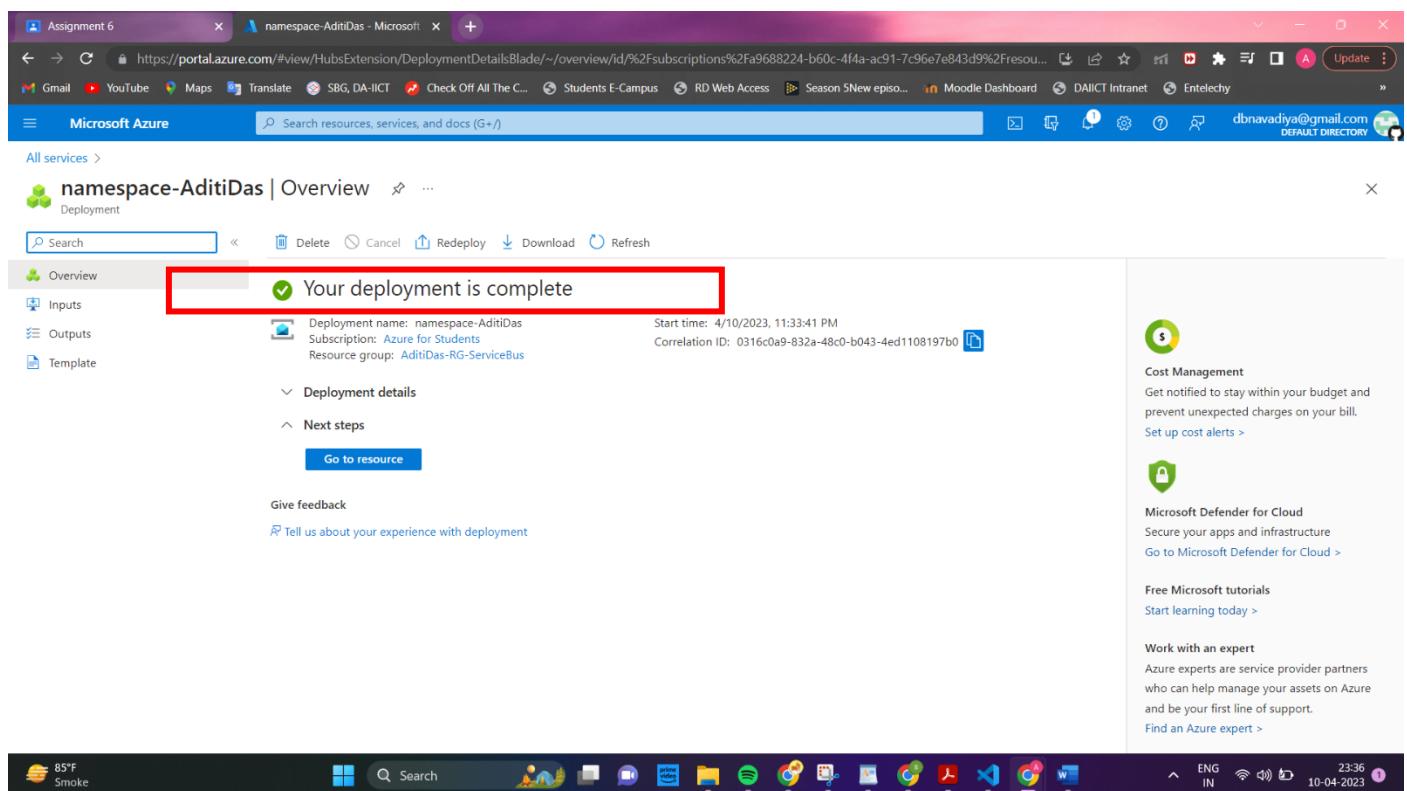


1. Azure Service Bus

i. Creating namespace



The screenshot shows the 'Create namespace' wizard in the Microsoft Azure portal. The 'Basics' tab is selected. In the 'Project Details' section, the subscription is set to 'Azure for Students' and the resource group is '(New) AditiDas-RG-ServiceBus'. In the 'Instance Details' section, the namespace name is 'namespace-AditiDas', located in 'East US 2', and the pricing tier is 'Standard (~\$10 USD per 12.5M Operations per Month)'. At the bottom, there are 'Review + create' and 'Next: Advanced >' buttons.



The screenshot shows the 'namespace-AditiDas | Overview' page in the Microsoft Azure portal. It displays a message 'Your deployment is complete' with a green checkmark. Deployment details are listed: Deployment name: namespace-AditiDas, Subscription: Azure for Students, Resource group: AditiDas-RG-ServiceBus, Start time: 4/10/2023, 11:33:41 PM, Correlation ID: 0316c0a9-832a-48c0-b043-4ed1108197b0. There are sections for 'Deployment details' and 'Next steps'. On the right side, there are links for 'Cost Management', 'Microsoft Defender for Cloud', 'Free Microsoft tutorials', 'Work with an expert', and a feedback section. The bottom of the screen shows the Windows taskbar with various pinned icons.

Assignment 6 | namespace-AditiDas - Microsoft | https://portal.azure.com/#@dbnavadiyagmail.onmicrosoft.com/resource/subscriptions/a9688224-b60c-4f4a-ac91-7c96e7e843d9/resourceGroups/AditiDas-RG-S... | Update

All services > namespace-AditiDas | Overview >

namespace-AditiDas

Service Bus Namespace

Search Queue Topic Refresh Delete Feedback

Overview

Activity log Access control (IAM) Tags Diagnose and solve problems

Settings Shared access policies Geo-Recovery Migrate to premium Encryption Configuration Properties Locks

Entities Queues Topics

85°F Smoke

Resource group (move) : AditiDas-RG-ServiceBus

Status : Active

Location : East US 2

Subscription (move) : Azure for Students

Subscription ID : a9688224-b60c-4f4a-ac91-7c96e7e843d9

Tags (edit) : Click here to add tags

Show data for the last: 1 hour 6 hours 12 hours 1 day 7 days 30 days

Requests

100
90
80
70
60
50
40
30
20
10
0

10:45 PM 11 PM 11:15 PM UTC+05:30

Messages

100
90
80
70
60
50
40
30
20
10
0

10:45 PM 11 PM 11:15 PM UTC+05:30

ENG IN 23:38 10-04-2023

ii. Creating Topic:

Assignment 6 | Get started with Azure Service Bus | Create topic - Microsoft Azure | https://portal.azure.com/#@dbnavadiyagmail.onmicrosoft.com/resource/subscriptions/a9688224-b60c-4f4a-ac91-7c96e7e843d9/resourceGroups/AditiDas-RG-S... | Update

All services > namespace-AditiDas | Overview > namespace-AditiDas

namespace-AditiDas | Topics

Service Bus Namespace

Search Topic Refresh Feedback

Search to filter items...

Name	Status	Max size	Subscription
No results.			

Create topic

Service Bus

Name * myTopic

Max topic size 1 GB

Message time to live Days 14 Hours 0 Minutes 0 Seconds 0

Enable auto-delete on idle topic

Enable duplicate detection

Enable partitioning

Create

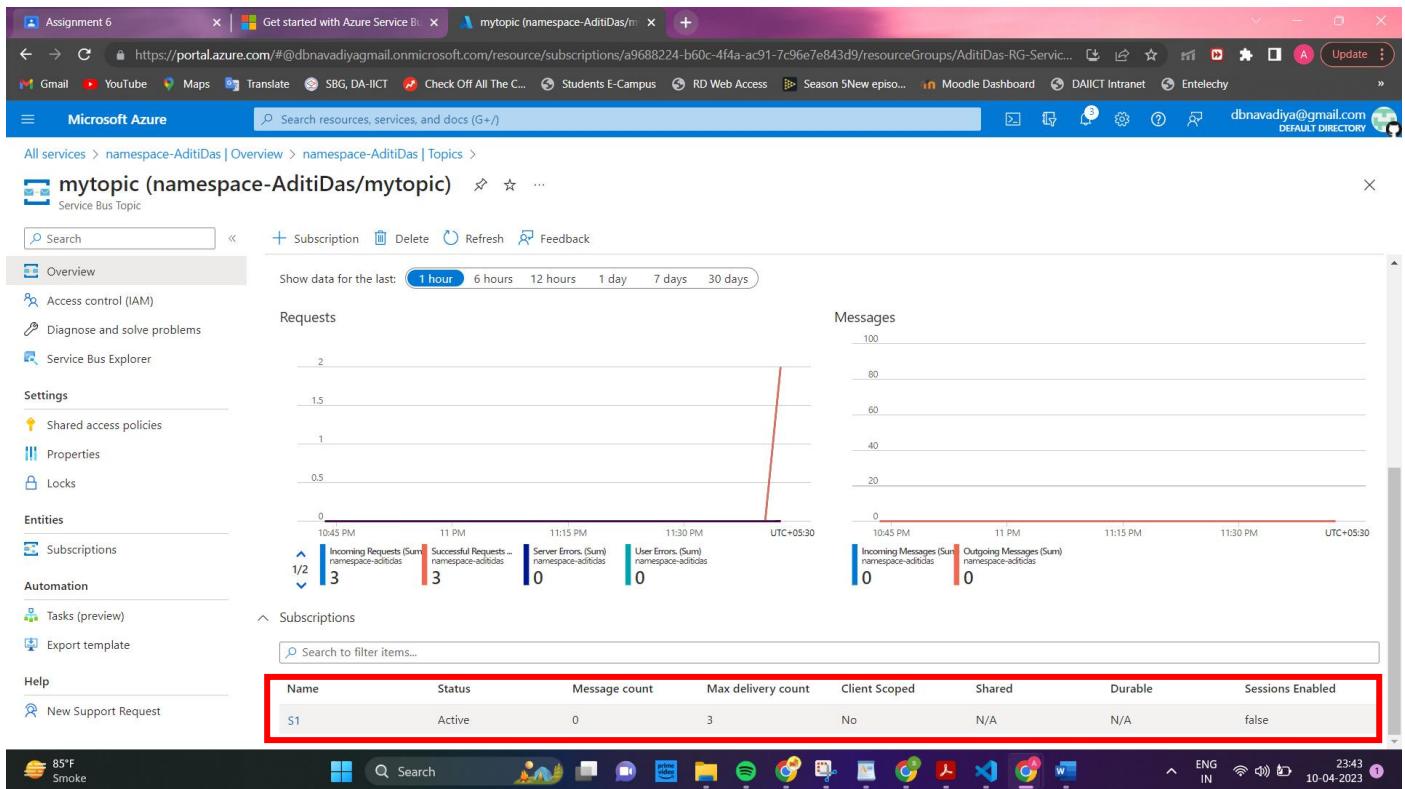
85°F Smoke

ENG IN 23:39 10-04-2023

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes tabs for 'Assignment 6', 'Get started with Azure Service Bus', and 'namespace-AditiDas - Microsoft'. The main content area is titled 'namespace-AditiDas | Topics'. On the left, a sidebar lists various service bus entities: Configuration, Properties, Locks, Entities (Queues, Topics), Monitoring (Insights, Alerts, Metrics, Diagnostic settings, Logs, Workbooks), and Automation (Tasks). The 'Topics' option is selected. The main pane displays a table of topics with one entry: 'mytopic' (Status: Active, Max size: 1024 MB, Subscription count: 0, Enable partitioning: false). A red box highlights the 'mytopic' row. The bottom of the screen shows the Windows taskbar with various pinned icons.

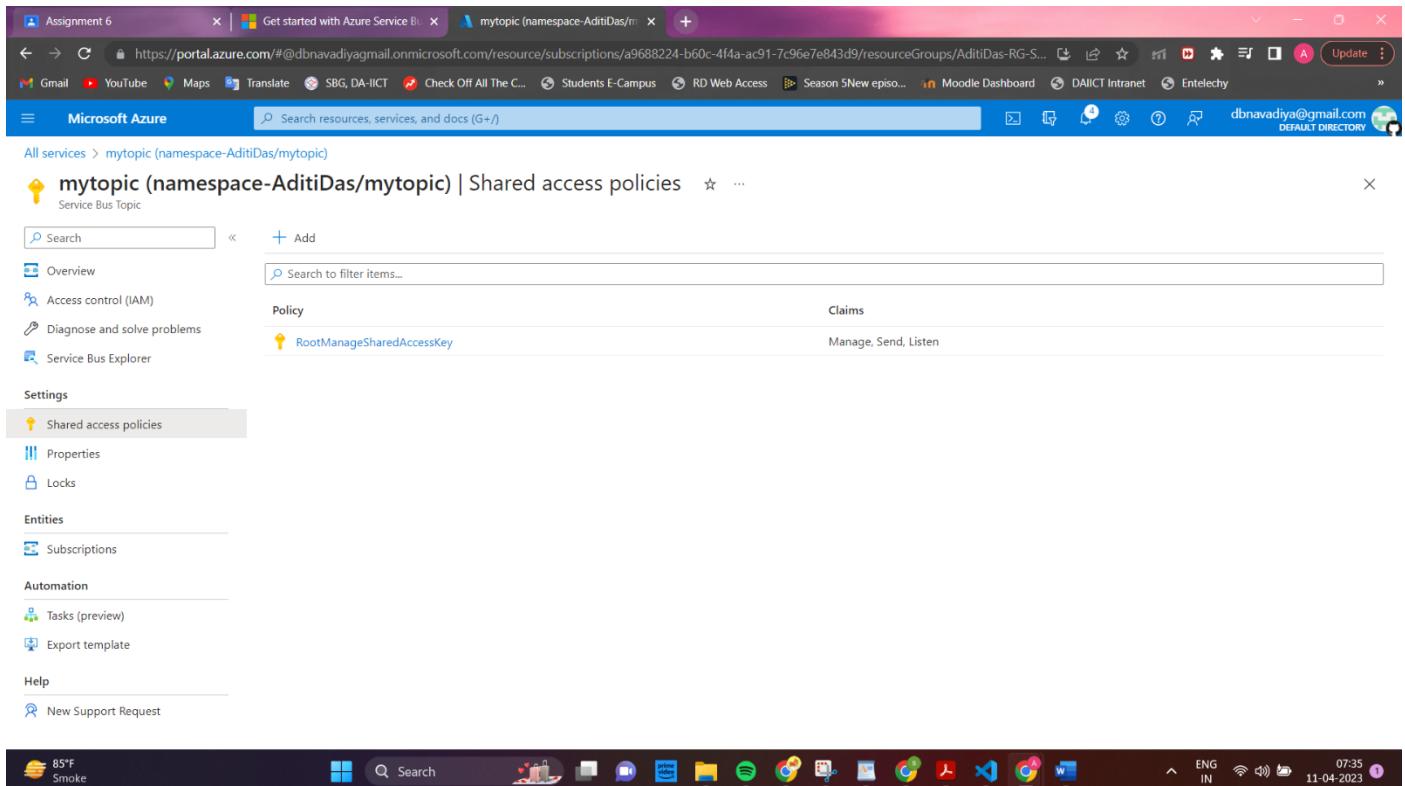
iii. Creating Subscription:

The screenshot shows the 'Create subscription' blade for the 'mytopic' topic. The title is 'Create subscription ...'. The 'Service Bus' section contains fields for 'Name' (set to 'S1'), 'Max delivery count' (set to '3'), and 'Auto-delete after idle for' (set to '14 Days'). There are checkboxes for 'Never auto-delete' and 'Forward messages to queue/topic'. The 'MESSAGE SESSIONS' section contains a note about service bus sessions and a checkbox for 'Enable sessions'. The 'MESSAGE TIME TO LIVE AND DEAD-LETTERING' section contains a note about message time to live and a field for 'Message time to live (default)' (set to '14 Days'). At the bottom is a blue 'Create' button. The bottom of the screen shows the Windows taskbar with various pinned icons.



iv. Getting Connection Strings:

I first created a policy in the shared access policies called RootManageSharedAccessKey.



I then obtained the primary key and primary key connection string values:

Primary Key:

er5cwAfMpnpnwPzMpARDXzaIxSbNW0diD+ASbGyGpsE=

Primary Connection String:

Endpoint=sb://namespace-aditidas.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=er5cwAfMpnpnwPzMpARDXzaIxSbNW0diD+ASbGyGpsE=;EntityPath=mytopic

v. Running the Web App:

I created two python files called send.py and receive.py and run the application on my cmd.

a. send.py

```
import asyncio
from azure.servicebus.aio import ServiceBusClient
from azure.servicebus import ServiceBusMessage

NAMESPACE_CONNECTION_STR = "Endpoint=sb://namespace-aditidas.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=er5cwAfMpnpnwPzMpARDXzaIxSbNW0diD+ASbGyGpsE=;EntityPath=mytopic"
TOPIC_NAME = "mytopic"

async def send_single_message(sender):
    # Create a Service Bus message
    message = ServiceBusMessage("Single Message")
    # send the message to the topic
    await sender.send_messages(message)
    print("Sent a single message")

async def send_a_list_of_messages(sender):
    # Create a list of messages
    messages = [ServiceBusMessage("Message in list") for _ in range(5)]
    # send the list of messages to the topic
    await sender.send_messages(messages)
    print("Sent a list of 5 messages")

async def send_batch_message(sender):
    # Create a batch of messages
    async with sender:
        batch_message = await sender.create_message_batch()
        for _ in range(10):
            try:
                # Add a message to the batch
                batch_message.add_message(ServiceBusMessage("Message inside a ServiceBusMessageBatch"))
            except ValueError:
                # ServiceBusMessageBatch object reaches max_size.

```

```

        # New ServiceBusMessageBatch object can be created here to send more data.
        break
    # Send the batch of messages to the topic
    await sender.send_messages(batch_message)
print("Sent a batch of 10 messages")

async def run():
    # create a Service Bus client using the connection string
    async with ServiceBusClient.from_connection_string(
        conn_str=NAMESPACE_CONNECTION_STR,
        logging_enable=True) as servicebus_client:
        # Get a Topic Sender object to send messages to the topic
        sender = servicebus_client.get_topic_sender(topic_name=TOPIC_NAME)
        async with sender:
            # Send one message
            await send_single_message(sender)
            # Send a list of messages
            await send_a_list_of_messages(sender)
            # Send a batch of messages
            await send_batch_message(sender)

asyncio.run(run())
print("Done sending messages")
print("-----")

```

b. receive.py

```

import asyncio
from azure.servicebus.aio import ServiceBusClient

NAMESPACE_CONNECTION_STR = "Endpoint=sb://namespace-
adidas.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAcces
sKey=er5cwAfMpnPnwPzMpARDXzaIxSbNW0diD+ASbGyGpsE=;EntityPath=mytopic"
SUBSCRIPTION_NAME = "S1"
TOPIC_NAME = "mytopic"

async def run():
    # create a Service Bus client using the connection string
    async with ServiceBusClient.from_connection_string(
        conn_str=NAMESPACE_CONNECTION_STR,
        logging_enable=True) as servicebus_client:

        async with servicebus_client:
            # get the Subscription Receiver object for the subscription
            receiver = servicebus_client.get_subscription_receiver(topic_name=TOPIC_NAME,
            subscription_name=SUBSCRIPTION_NAME, max_wait_time=5)
            async with receiver:
                received_msgs = await receiver.receive_messages(max_wait_time=5,
max_message_count=20)
                for msg in received_msgs:
                    print("Received: " + str(msg))# complete the message so that the
message is removed from the subscription
                    receiver.complete_message(msg)

asyncio.run(run())

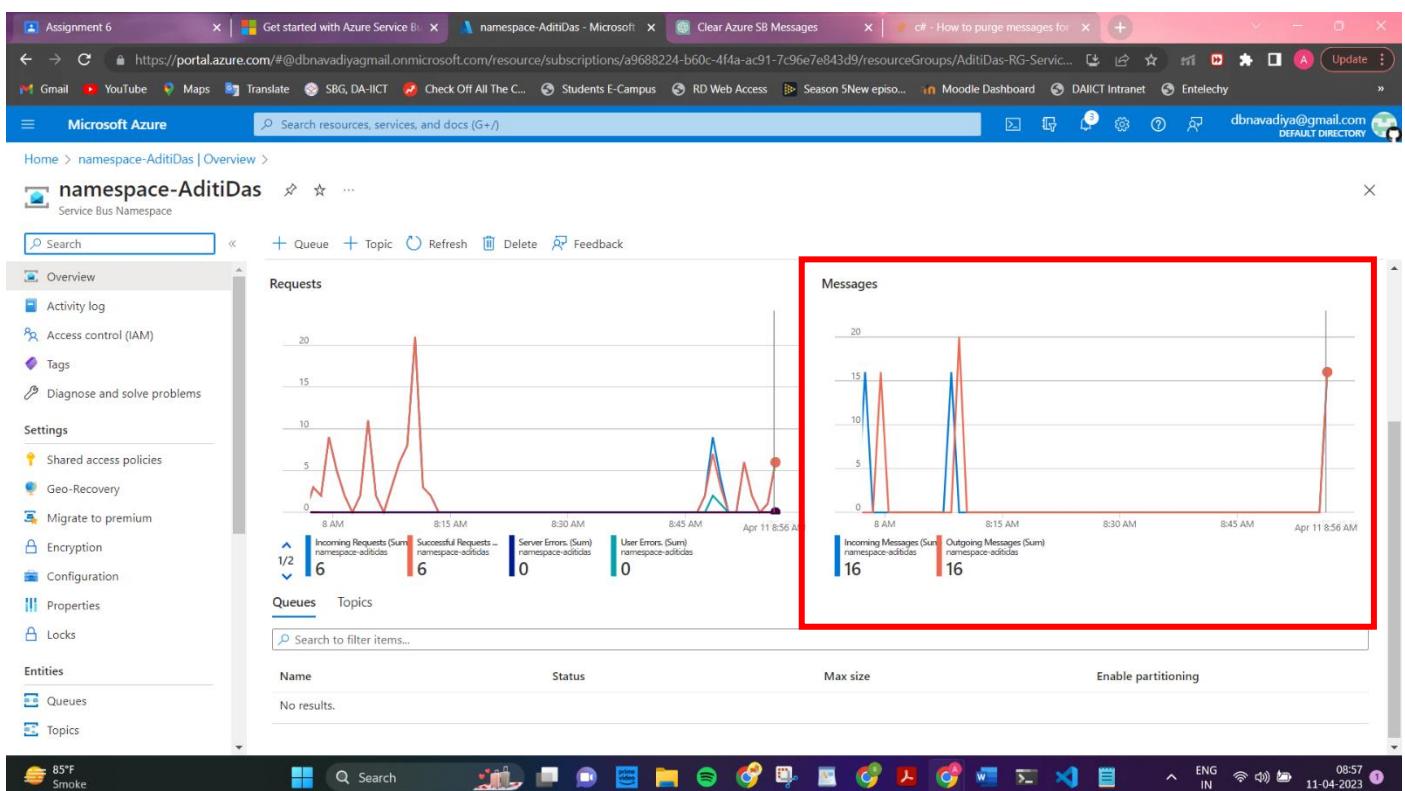
```

c. Running Web App:

```
C:\WINDOWS\system32\cmd. x + x  
D:\DA-IICT\Year 3\Sem 6\Cloud Computing\Lab\Lab 6\Code Files>python send.py & python receive.py  
Sent a single message  
Sent a list of 5 messages  
Sent a batch of 10 messages  
Done sending messages  
-----  
Received: Single Message  
D:\DA-IICT\Year 3\Sem 6\Cloud Computing\Lab\Lab 6\Code Files\receive.py:23: RuntimeWarning: coroutine 'ServiceBusReceiver.complete_message' was never awaited  
    receiver.complete_message(msg)  
RuntimeWarning: Enable tracemalloc to get the object allocation traceback  
Received: Message in list  
Received: Message inside a ServiceBusMessageBatch  
D:\DA-IICT\Year 3\Sem 6\Cloud Computing\Lab\Lab 6\Code Files>
```

vi. Outputs:

a. Namespace Output:



Here, we can observe 16 incoming and 16 outgoing messages.

b. Topic Output:

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#@dbnavadiyagmail.onmicrosoft.com/resource/subscriptions/a9688224-b60c-4f4a-ac91-7c96e7e843d9/resourceGroups/AditiDas-RG-Servic...>. The page displays the 'mytopic (namespace-AditiDas/mytopic)' Service Bus Topic overview. It includes a search bar, navigation links, and a top menu with items like 'Subscription', 'Delete', 'Refresh', and 'Feedback'. A time range selector shows data for the last 1 hour. The 'Requests' section contains two charts: one for incoming requests (sum) and another for successful requests (sum). Below the charts, a summary table shows 67 incoming requests, 64 successful, 0 server errors, and 3 user errors. The 'Messages' section shows two charts for incoming and outgoing messages, both with a value of 16. The 'Subscriptions' section lists a single subscription named 'S1' with an active status and 16 messages. The left sidebar provides links for access control, diagnosis, service bus explorer, settings, entities, automation, tasks, export template, help, and support requests. The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

We can see 3 incoming and 3 outgoing messages here.

c. Subscriptions Output:

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#@dbnavadiyagmail.onmicrosoft.com/resource/subscriptions/a9688224-b60c-4f4a-ac91-7c96e7e843d9/resourceGroups/AditiDas-RG-Servic...>. The page displays the 'S1 (namespace-AditiDas/mytopic/S1)' Service Bus Subscription overview. It includes a search bar, navigation links, and a top menu with items like 'Delete', 'Refresh', and 'Feedback'. A summary table shows the subscription details: Namespace (namespace-AditiDas), Topic (mytopic), Status (Active), Forward messages to (Disabled), Created (Tuesday, April 11, 2023), Updated (Tuesday, April 11, 2023), Sessions (Disabled), and Dead lettering (Disabled). The 'Settings' section allows configuration of message delivery count (3), message time to live (14 days), auto-delete after idle (14 days), and message lock duration (1 minute). The 'Message Counts' section is highlighted with a red box and shows four categories: Active (0 messages), Dead-letter (0 messages), Transfer (0 messages), and Transfer dead-letter (0 messages). The 'Filters' section shows a table with a single row for '\$Default' using 'SqlFilter'. The left sidebar provides links for overview, diagnosis, service bus explorer, automation, tasks, export template, help, and support requests. The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

Here, active message count is 0.

d. Subscriptions Output (receive commented):

The screenshot shows the Azure Service Bus Subscription settings for 'S1 (namespace-AditiDas/mytopic/S1)'. The 'Message Counts' section is highlighted with a red box, showing 16 Active messages, 0 Dead-letter messages, 0 Transfer messages, and 0 Transfer dead-letter messages.

Key details from the subscription settings:

- Namespace: namespace-AditiDas
- Topic: mytopic
- Status: Active
- Forward messages to: Disabled
- Created: Tuesday, April 11, 2023
- Updated: Tuesday, April 11, 2023
- Sessions: Disabled
- Dead lettering: Disabled

Here, active message count is 16.

2. Azure Event Hub

i. Creating a event namespace hub :

The screenshot shows the Azure Event Hub Namespace settings for 'eventHub-AditiDas'. The 'Event Hub' section is visible, showing the following details:

- Resource group (move): AditiDas-RG-EventHub
- Status: Active
- Location: East US 2
- Subscription (move): Azure for Students
- Subscription ID: a9688224-b60c-4f4a-ac91-7c96e7e843d9
- Host name: eventHub-AditiDas.servicebus.windows.net
- Created: Tuesday, April 11, 2023 at 15:11:16 GMT+5:30
- Updated: Tuesday, April 11, 2023 at 15:12:02 GMT+5:30
- Zone Redundancy: Enabled
- Pricing tier: Standard
- Throughput Units: 1 unit
- Auto-inflate throughput ...: Disabled
- Local Authentication: Enabled

The 'NAMESPACE CONTENTS' section shows 0 EVENT HUBS, KAFKA SURFACE ENABLED, and ZONE REDUNDANCY ENABLED.

ii. Creating an event hub:

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes links for Gmail, YouTube, Maps, Translate, SBG, DA-IICT, Check Off All The C..., Students E-Campus, RD Web Access, Season 5New episode..., Moodle Dashboard, DAIICIT Intranet, and Entelechy. The search bar at the top right contains the email address "dbnavadiya@gmail.com".

The main content area displays the "myeventhubaditi (eventHub-AditiDas/myeventhubaditi)" blade. On the left, a sidebar lists various options: Overview, Access control (IAM), Diagnose and solve problems, Settings, Shared access policies, Properties, Locks, Entities, Consumer groups, Features (Capture, Process data), Automation (Tasks (preview), Export template), and Help.

The central pane shows the "Essentials" section for the Event Hub. Key details include:

- Resource group: AditiDas-RG-EventHub
- Location: East US 2
- Subscription: Azure for Students
- Subscription ID: a9688224-b60c-4f4a-ac91-7c96e7e843d9
- Partition count: 1
- Status: Active
- Namespace: eventHub-AditiDas
- Created: Tuesday, April 11, 2023 at 09:46:46 GMT+5:30
- Updated: Tuesday, April 11, 2023 at 09:46:46 GMT+5:30
- Cleanup policy: Delete

Below the essentials, there are four cards:

- Capture events**: Use Capture to save your events to persistent storage.
- Process data**: Process data instantly with Azure Stream Analytics.
- Connect**: Authenticate with connection strings and SAS policies.
- Checkpoint**: Create consumer groups to checkpoint your events.

At the bottom of the central pane, there are buttons for "Event Hub Contents" (1 CONSUMER GROUP, ACTIVE, CLEANUP POLICY: DELETE), "Event Hub status" (ACTIVE), and "Partition count" (1). A "Show data for the last:" dropdown offers options from 1 hour to 30 days, with "1 hour" currently selected.

iii. Create a storage account:

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes links for Gmail, YouTube, Maps, Translate, SBG, DA-IICT, Check Off All The C..., Students E-Campus, RD Web Access, Season 5New episode..., Moodle Dashboard, DAIICIT Intranet, and Entelechy. The search bar at the top right contains the email address "dbnavadiya@gmail.com".

The main content area displays the "Storage accounts" blade. The left sidebar lists "Storage accounts" and "Default Directory". Under "Storage accounts", there is a list of existing accounts: "csg100320026d93c69f" and "mystorageaccountaditidas".

The central pane shows the "mystorageaccountaditidas" storage account blade. The "Overview" section provides the following details:

- Resource group: AditiDas-RG-EventHub
- Location: West US
- Primary/Secondary Location: Primary: West US, Secondary: East US
- Subscription: Azure for Students
- Subscription ID: a9688224-b60c-4f4a-ac91-7c96e7e843d9
- Disk state: Primary: Available, Secondary: Available
- Performance: Standard
- Replication: Read-access geo-redundant storage (RA-GRS)
- Account kind: StorageV2 (general purpose v2)
- Provisioning state: Succeeded
- Created: 4/11/2023, 3:33:55 PM

The "Properties" tab is selected, showing configuration for the Blob service and Security.

Blob service settings:

- Hierarchical namespace: Disabled
- Default access tier: Hot
- Blob public access: Enabled
- Blob soft delete: Disabled

Security settings:

- Require secure transfer for REST API operations: Enabled
- Storage account key access: Enabled
- Minimum TLS version: Version 1.0

At the bottom of the central pane, there are tabs for PowerShell, Command Prompt, and a terminal window showing the command "PS /home/jay>". The taskbar at the bottom includes icons for various Windows applications like File Explorer, Task View, and Start.

iv. Create a container:

The screenshot shows the Microsoft Azure portal interface. The top navigation bar has tabs for 'You are signed in as 202001259', 'mystorageaccountaditidas - Micr...', 'Send or receive events from Azure...', and 'Configure a connection string - /'. Below the navigation bar, the address bar shows the URL: https://portal.azure.com/#/dbnavadiagmail.onmicrosoft.com/resource/subscriptions/a9688224-b60c-4f4a-ac91-7c96e7e843d9/resourceGroups/AditiDas-RG-E... . The main content area is titled 'mystorageaccountaditidas | Containers'. On the left, there's a sidebar with 'Overview', 'Activity log', 'Tags', 'Diagnose and solve problems', 'Access Control (IAM)', 'Data migration', 'Events', and 'Storage browser'. Under 'Data storage', 'Containers' is selected and highlighted. A table lists one container: 'mystorage-container-aditi-das' with 'Last modified' as 4/11/2023, 3:45:08 PM, 'Public access level' as 'Container', and 'Lease state' as 'Available'. At the bottom of the table is a three-dot menu icon. The taskbar at the bottom shows various icons for Windows 10, including File Explorer, Edge, and File Explorer.

v. Creating sendEventHub.py:

```
import asyncio

from azure.eventhub import EventData
from azure.eventhub.aio import EventHubProducerClient

EVENT_HUB_CONNECTION_STR = "Endpoint=sb://eventhub-
aditidas.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAcces
sKey=Z0xuv+WXMhMsawwp0ryB1Pbcra8Dqdib/+AEhEIZuy0="
EVENT_HUB_NAME = "myeventhubaditi"

async def run():
    # Create a producer client to send messages to the event hub.
    # Specify a connection string to your event hubs namespace and the event hub name.
    producer = EventHubProducerClient.from_connection_string(
        conn_str=EVENT_HUB_CONNECTION_STR, eventhub_name=EVENT_HUB_NAME
    )
    async with producer:
        # Create a batch.
        event_data_batch = await producer.create_batch()
        # Add events to the batch.
        event_data_batch.add(EventData("First event "))
        event_data_batch.add(EventData("Second event"))
        event_data_batch.add(EventData("Third event"))

        # Send the batch of events to the event hub.
        await producer.send_batch(event_data_batch)

asyncio.run(run())
```

vi. Create receiveEventHub.py

```
import asyncio

from azure.eventhub.aio import EventHubConsumerClient
from azure.eventhub.extensions.checkpointstoreblobaio import (
    BlobCheckpointStore,
)

BLOB_STORAGE_CONNECTION_STRING =
"DefaultEndpointsProtocol=https;AccountName=mystorageaccountaditidas;AccountKey=SSu5MB7UY1C7fClikzk4eT4zoGnuIqrSmMNRL2simnr0p6UdfWZl9bZm3EcZJJqh8VebJnYuFZ6G+AStD912Ug==;EndpointSuffix=core.windows.net"
BLOB_CONTAINER_NAME = "mystorage-container-aditi-das"
EVENT_HUB_CONNECTION_STR = "Endpoint=sb://eventhub-aditidas.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=Z0xuv+WXmHMsawwp0ryBlPbcra8Dqdib/+AEhEIZuy0="
EVENT_HUB_NAME = "myeventhubaditi"

async def on_event(partition_context, event):
    # Print the event data.
    print(
        'Received the event: "{}" from the partition with ID: "{}".format(
            event.body_as_str(encoding="UTF-8"), partition_context.partition_id
        )
    )

    # Update the checkpoint so that the program doesn't read the events
    # that it has already read when you run it next time.
    await partition_context.update_checkpoint(event)

async def main():
    # Create an Azure blob checkpoint store to store the checkpoints.
    checkpoint_store = BlobCheckpointStore.from_connection_string(
        BLOB_STORAGE_CONNECTION_STRING, BLOB_CONTAINER_NAME
    )
    # Create a consumer client for the event hub.
    client = EventHubConsumerClient.from_connection_string(
        EVENT_HUB_CONNECTION_STR,
        consumer_group="$Default",
        eventhub_name=EVENT_HUB_NAME,
        checkpoint_store=checkpoint_store,
    )
    async with client:
        # Call the receive method. Read from the beginning of the
        # partition (starting_position: "-1")
        await client.receive(on_event=on_event, starting_position="-1")

if __name__ == "__main__":
    loop = asyncio.get_event_loop()
    # Run the main method.
    loop.run_until_complete(main())
```

vii. Output:

I ran the scripts in the terminal and the output is as follows:

```
D:\DA-IICT\Year 3\Sem 6\Cloud Computing\Lab\Lab 6\Code Files>python receiveEventHub.py
Received the event: "First event " from the partition with ID: "0"
Received the event: "Second event" from the partition with ID: "0"
Received the event: "Third event" from the partition with ID: "0"
|
D:\DA-IICT\Year 3\Sem 6\Cloud Computing\Lab\Lab 6\Code Files>python sendEventHub.py
D:\DA-IICT\Year 3\Sem 6\Cloud Computing\Lab\Lab 6\Code Files>
```

I ran the scripts again in the terminal with the partition count 3 in the event hub (and number of messages sent are 20):

These messages might get sent into different partitions depending upon the load on the hubs.