

Lab Report - 6

Name – Divya Kirtikumar Patel

Student ID - 202001420

1. Send messages to an Azure Service Bus topic and receive messages from subscriptions to the topic (Python)

i. Create service Bus

Microsoft Azure | Search resources, services, and docs (G+)

Home > ms-service-bus | Resource group

Search

Overview | Activity log | Access control (IAM) | Tags | Resource visualizer | Events

Settings | Deployments | Security | Policies | Properties

Essentials

Subscription (move): [Azure for Students](#) | Deployments: 1 Succeeded

Subscription ID: 83dfa20c-9d3c-4d0e-befb-ae99f4e1a9b1 | Location: East US

Tags (edit): [Click here to add tags](#)

Resources | Recommendations

Filter for any field... | Type equals all | Location equals all | Add filter

Showing 1 to 1 of 1 records. | Show hidden types | No grouping | List view

Name	Type	Location
divya-202001420	Service Bus Namespace	East US

ii. Service Bus Dashboard

Microsoft Azure | Search resources, services, and docs (G+)

Home > ms-service-bus > divya-202001420 | Service Bus Namespace

Search

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 | Monitoring | Insights (Preview)

Queue | Topic | Refresh | Delete | Feedback

Resource group (move): [ms-service-bus](#) | Created: Friday, April 14, 2023

Status: Active | Updated: Friday, April 14, 2023

Location: East US | Pricing tier: [Standard](#)

Subscription (move): [Azure for Students](#) | Host name: [divya-202001420.servicebus.windows.net](#)

Subscription ID: 83dfa20c-9d3c-4d0e-befb-ae99f4e1a9b1 | Local Authentication: [Enabled](#)

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

Messages

Incoming Requests (Sum) | Successful Requests (Sum) | Server Errors (Sum) | User Errors (Sum)

Incoming Messages (Sum) | Outgoing Messages (Sum)

iii. Create New Topic in Service Bus

Microsoft Azure Search resources, services, and docs (G+/)

All services > divya-202001420 | Overview > divya-202001420 | Topics >

mytopic1 (divya-202001420/mytopic1)

Service Bus Topic

Search

+ Subscription Delete Refresh Feedback

Overview

Access control (IAM)

Diagnose and solve problems

Service Bus Explorer

Settings

Shared access policies

Properties

Locks

Entities

Subscriptions

Automation

Tasks (preview)

Export template

Help

New Support Request

Essentials

Namespace : [divya-202001420](#) Topic URL : <https://divya-202001420.servicebus.windows.net/mytopic1>

Status : [Active](#) Created : Friday, April 14, 2023 at 17:59:43 GMT+5:30

Partitioning : Disabled Updated : Friday, April 14, 2023 at 17:59:43 GMT+5:30

Duplicate detection : Disabled

Settings

Current size : 0.0 KB Max size : 1 GB (change) Message time to live : 14 DAYS (change) Auto-delete : NEVER (change) Free space : 100.0 %

Message count

Scheduled : 0 MESSAGES

Metrics

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

Requests

Messages

iv. Create new subscription in topic

Microsoft Azure Search resources, services, and docs (G+/)

Home > mytopic1 (divya-202001420/mytopic1) | Subscriptions >

S1 (divya-202001420/mytopic1/S1)

Service Bus Subscription

Search

Delete Refresh Feedback

Overview

Diagnose and solve problems

Service Bus Explorer

Automation

Tasks (preview)

Export template

Help

New Support Request

Auto refresh Off

Essentials

Namespace : [divya-202001420](#) Created : Friday, April 14, 2023

Topic : [mytopic1](#) Updated : Friday, April 14, 2023

Status : [Active](#) Sessions : Disabled

Forward messages to : [Disabled](#) Dead lettering : [Disabled](#)

Settings

Max delivery count : 3 (change) Message time to live : 14 DAYS (change) Auto-delete after idle for : 14 DAYS (change) Message lock duration : 1 MINUTE (change)

Message Counts

Active : 0 MESSAGES Dead-letter : 0 MESSAGES Transfer : 0 MESSAGES Transfer dead-letter : 0 MESSAGES

Filters







+ Add filter

Name	Filter Type
\$Default	SqlFilter

v. Add Role Assignment

Microsoft Azure

Search resources, services, and docs (G+)



202001420@daiict.ac.in
DAIICT.AC.IN (DAIICT.AC.IN)

Home > mytopic1 (divya-202001420/mytopic1) | Access control (IAM) >

Add role assignment

Role

Members

Review + assign

Role

Azure Service Bus Data Owner

Scope

/subscriptions/83dfa20c-9d3c-4d0e-befb-ae99f4e1a9b1/resourceGroups/ms-service-bus/providers/Microsoft.ServiceBus/namespaces/divya-202001420/topics/mytopic1

Members

Name	Object ID	Type
Aditya Nawal	db5c72b4-e691-4c02-a9eb-c533d583de8e	User
Aryan Shah	2f676182-7bae-4d16-8eb1-fb46d905cdc9	User


Description

No description

Review + assign

Previous

Next

 Feedback

Send.py

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

NAMESPACE_CONNECTION_STR = "Endpoint=sb://divya-202001420.servicebus.windows.net/;SharedAccessKeyName=newpolicy;SharedAccessKey=6Q1uw0/JyLevdQ7vF2z7jBYk14c0EspLy+ASbF9dLSQ=;EntityPath=mytopic1"
TOPIC_NAME = "mytopic1"

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("-----")

```

recv.py

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

NAMESPACE_CONNECTION_STR = "Endpoint=sb://divya-202001420.servicebus.windows.net/;SharedAccessKeyName=newpolicy;SharedAccessKey=6Q1uw0/JyLevdQ7vF2z7jBYk14c0EspLy+ASbF9d1SQ=;EntityPath=mytopic1"
SUBSCRIPTION_NAME = "S1"
TOPIC_NAME = "mytopic1"

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:

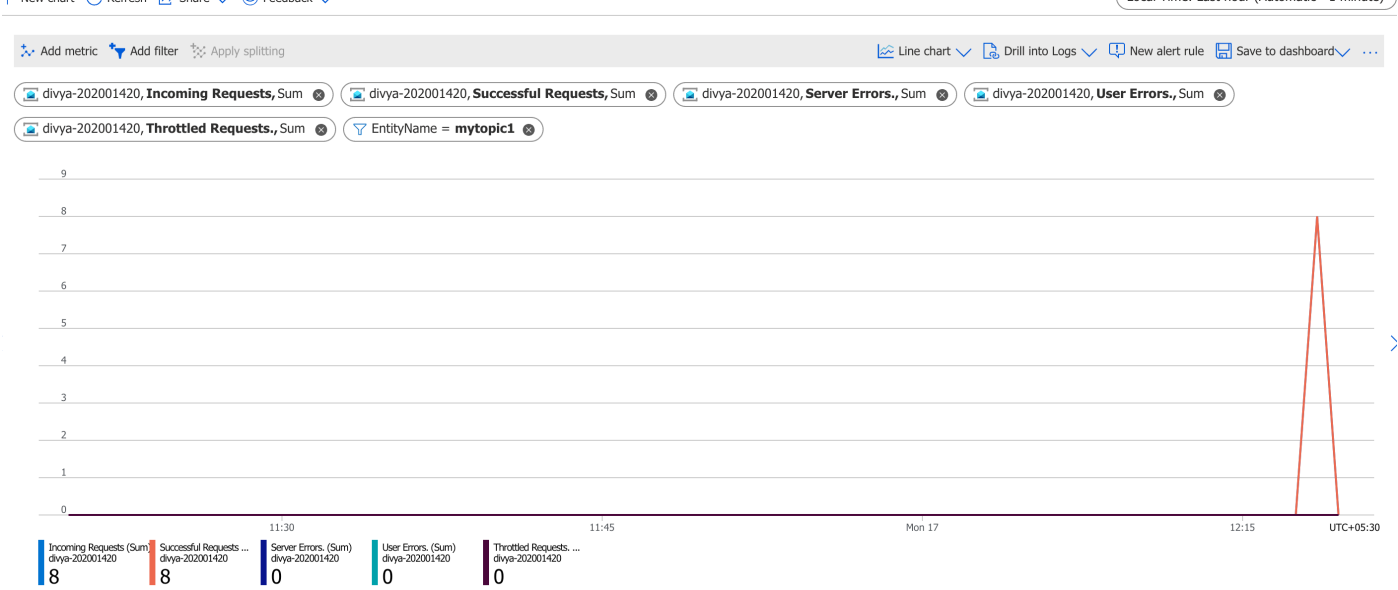
        async with servicebus_client:
            # get the Subscription Receiver object for the subscription
            receiver = servicebus_client.get_subscription_receiver(
                topic_name=TOPIC_NAME,
                subscription_name= SUBSCRIPTION_NAME, max_wait_time=5)
            async with 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())
```

Running Scripts

```
divya@Divyas-MacBook-Air ~ % python send.py & python recv.py

Sent a single message
Sent a list of 5 messages
Sent a batch of 10 messages
Done sending messages
-----
Received: Single Message
Received: Message in list
Received: Message in list
Received: Message in list
Received: Message in list
Received: Message in list
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
Received: Message inside a ServiceBusMessageBatch
```

Dashboard of Topic



2. Send events to or receive events from event hubs by using Python

i. Create event-hub namespace

Home > Event Hubs >

Create Namespace

Event Hubs

Validating...

Basics

Advanced

Networking

Tags

Review + create

Event Hubs Namespace by Microsoft

Basics

Networking

Security

Namespace name

Subscription

Resource group

Location

Pricing tier

Throughput Units

Availability Zones (Zone Redundancy)

Auto-Inflate Maximum Throughput Units

Connectivity method

Minimum TLS version

Local Authentication

event-hub1-202001420

Azure for Students

event-hub-divya

East US

Standard

1

Enabled

Disabled

Public access

1.2

Enabled

Create

< Previous

Next >

ii. Create Event-hub

Home > event-hub-divya >

event-hub1-202001420

Event Hubs Namespace

Search

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Shared access policies

Scale

Geo-Recovery

Networking

Encryption

Identity

Configuration

Properties

Locks

Entities

Event Hubs

Schema Registry

+ Event Hub

Delete

Refresh

Feedback

Status

Location

Subscription

Subscription ID

Updated

Zone Redundancy

Pricing tier

Throughput Units

Auto-inflate throughput ...

Local Authentication

Not Active

East US

Azure for Students

83dfa20c-9d3c-4d0e-befb-ae99f4e1a9b1

Monday, April 17, 2023 at 24:44:58 GMT+5:30

Enabled

Standard

1 unit

Disabled

Enabled

Tags

Click here to add tags

Namespace contents

Kafka surface

Zone redundancy

Loading

Enabled

Enabled

Show data for the last:

1 hour

6 hours

12 hours

1 day

7 days

30 days

Requests

Messages

Throughput

Incoming Requests (Sum)

Successful Requests (Sum)

Incoming Messages (Sum)

Outgoing Messages (Sum)

Incoming Bytes (Sum)

Outgoing Bytes (Sum)

iii. Event-hub details

Microsoft Azure

Search resources, services, and docs (G+)

Home > event-hub-diviya > event-hub1-202001420 | Event Hubs >

myeventhub1 (event-hub1-202001420/myeventhub1)
Event Hubs Instance

Search

Consumer group Delete Refresh Feedback

Overview

Access control (IAM)
Diagnose and solve problems

Settings

Shared access policies
Properties
Locks

Entities

Consumer groups

Features

Capture
Process data

Automation

Tasks (preview)

Essentials

JSON View

Resource group (move) : event-hub-diviya
Location : East US
Subscription (move) : Azure for Students
Subscription ID : 83dfa20c-9d3c-4d0e-befb-ae99f4e1a9b1
Partition count : 1

Status : Active
Namespace : event-hub1-202001420
Created : Sunday, April 16, 2023 at 19:26:15 GMT+5:30
Updated : Sunday, April 16, 2023 at 19:26:15 GMT+5:30
Cleanup policy : Delete

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.

iv. Container Details

Microsoft Azure

Search resources, services, and docs (G+)

Home > mystorage202001420 | Containers >

container1
Container

Search

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot Give feedback

Overview

Diagnose and solve problems
Access Control (IAM)

Settings

Shared access tokens
Access policy
Properties
Metadata

Authentication method: Access key (Switch to Azure AD User Account)
Location: container1

Search blobs by prefix (case-sensitive)

Show deleted blobs

Add filter

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
No results						

Run.py

```
import asyncio

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

EVENT_HUB_FULLY_QUALIFIED_NAMESPACE = "event-hub1-202001420"
EVENT_HUB_NAME = "myeventhub1"

credential = DefaultAzureCredential()

async def run():
    # Create a producer client to send messages to the event hub.
    # Specify a credential that has correct role assigned to access
    # event hubs namespace and the event hub name.
    producer = EventHubProducerClient(
        fully_qualified_namespace=EVENT_HUB_FULLY_QUALIFIED_NAMESPACE,
        eventhub_name=EVENT_HUB_NAME,
        credential=credential,
    )
    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)

        # Close credential when no longer needed.
        await credential.close()

asyncio.run(run())
```

Recv.py

```
from azure.identity.aio import DefaultAzureCredential
from azure.eventhub.extensions.checkpointstoreblobaio import (
    BlobCheckpointStore,
)
from azure.eventhub.aio import EventHubConsumerClient
import asyncio

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

EVENT_HUB_FULLY_QUALIFIED_NAMESPACE = "event-hub1-202001420"
EVENT_HUB_NAME = "myeventhub1"

BLOB_STORAGE_ACCOUNT_URL =
"DefaultEndpointsProtocol=https;AccountName=mystorage202001420;AccountKey=S
Su5MB7UY1C7fClikzk4eT4zoGnuIqrSmMNRL2simnr0p6UdfWZl9bZm3EcZJJqh8VebJnYuFZ6G
+ASdD912Ug==;EndpointSuffix=core.windows.net"
BLOB_CONTAINER_NAME = "container1"

credential = DefaultAzureCredential()

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(
        blob_account_url=BLOB_STORAGE_ACCOUNT_URL,
        container_name=BLOB_CONTAINER_NAME,
```

```

        credential=credential,
    )

    # Create a consumer client for the event hub.
    client = EventHubConsumerClient(
        fully_qualified_namespace=EVENT_HUB_FULLY_QUALIFIED_NAMESPACE,
        eventhub_name=EVENT_HUB_NAME,
        consumer_group="$Default",
        checkpoint_store=checkpoint_store,
        credential=credential,
    )

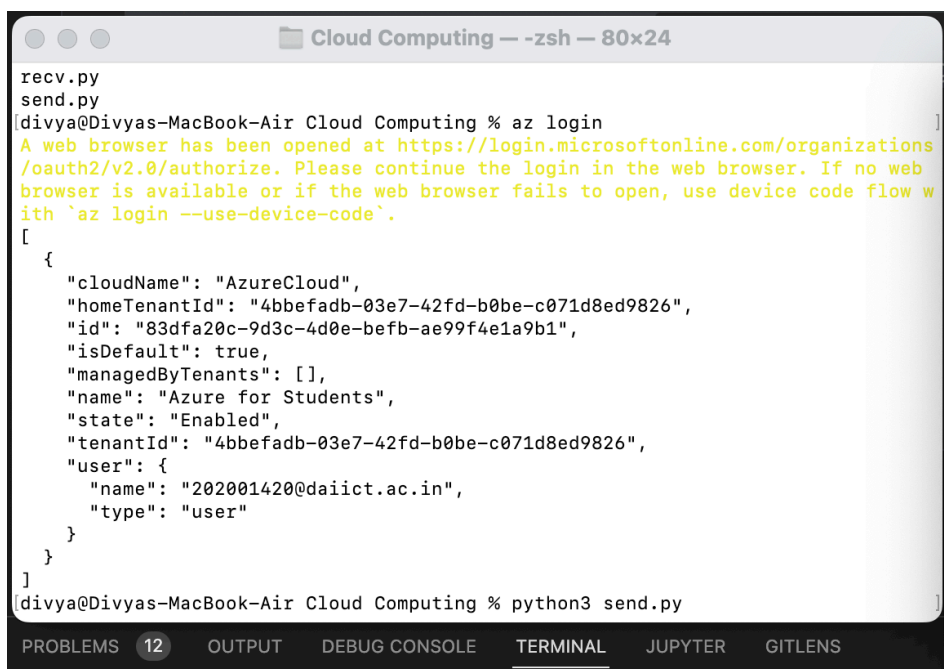
    async with client:
        # Call the receive method. Read from the beginning of the partition
        # (starting_position: "-1")
        await client.receive(on_event=on_event, starting_position="-1")

    # Close credential when no longer needed.
    await credential.close()

if __name__ == "__main__":
    # Run the main method.
    asyncio.run(main())

```

Running send.py



The screenshot shows a terminal window titled "Cloud Computing — -zsh — 80x24". The user is at a prompt "divya@Divyas-MacBook-Air Cloud Computing %". They run "az login", which opens a web browser at "https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize". After logging in, they run "python3 send.py", which outputs a JSON object representing an Azure user profile.

```

divya@Divyas-MacBook-Air Cloud Computing % az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with 'az login --use-device-code'.
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "4bbefadb-03e7-42fd-b0be-c071d8ed9826",
    "id": "83dfa20c-9d3c-4d0e-befb-ae99f4e1a9b1",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "4bbefadb-03e7-42fd-b0be-c071d8ed9826",
    "user": {
      "name": "202001420@daiict.ac.in",
      "type": "user"
    }
  }
]
divya@Divyas-MacBook-Air Cloud Computing % python3 send.py

```

At the bottom of the terminal window, there is a navigation bar with the following tabs: PROBLEMS (12), OUTPUT, DEBUG CONSOLE, TERMINAL, JUPYTER, and GITLENS.

Running recv.py

```
Cloud Computing — -zsh — 105x25

High Performance Computing
divya@Divyas-MacBook-Air Semester-6 % az login
[A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with 'az login --use-device-code'.]
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "4bbefadb-03e7-42fd-b0be-c071d8ed9826",
    "id": "83dfa20c-9d3c-4d0e-befb-ae99f4e1a9b1",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "4bbefadb-03e7-42fd-b0be-c071d8ed9826",
    "user": {
      "name": "202001420@daiict.ac.in",
      "type": "user"
    }
  }
]
divya@Divyas-MacBook-Air Semester-6 % python3 recv.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"
```