

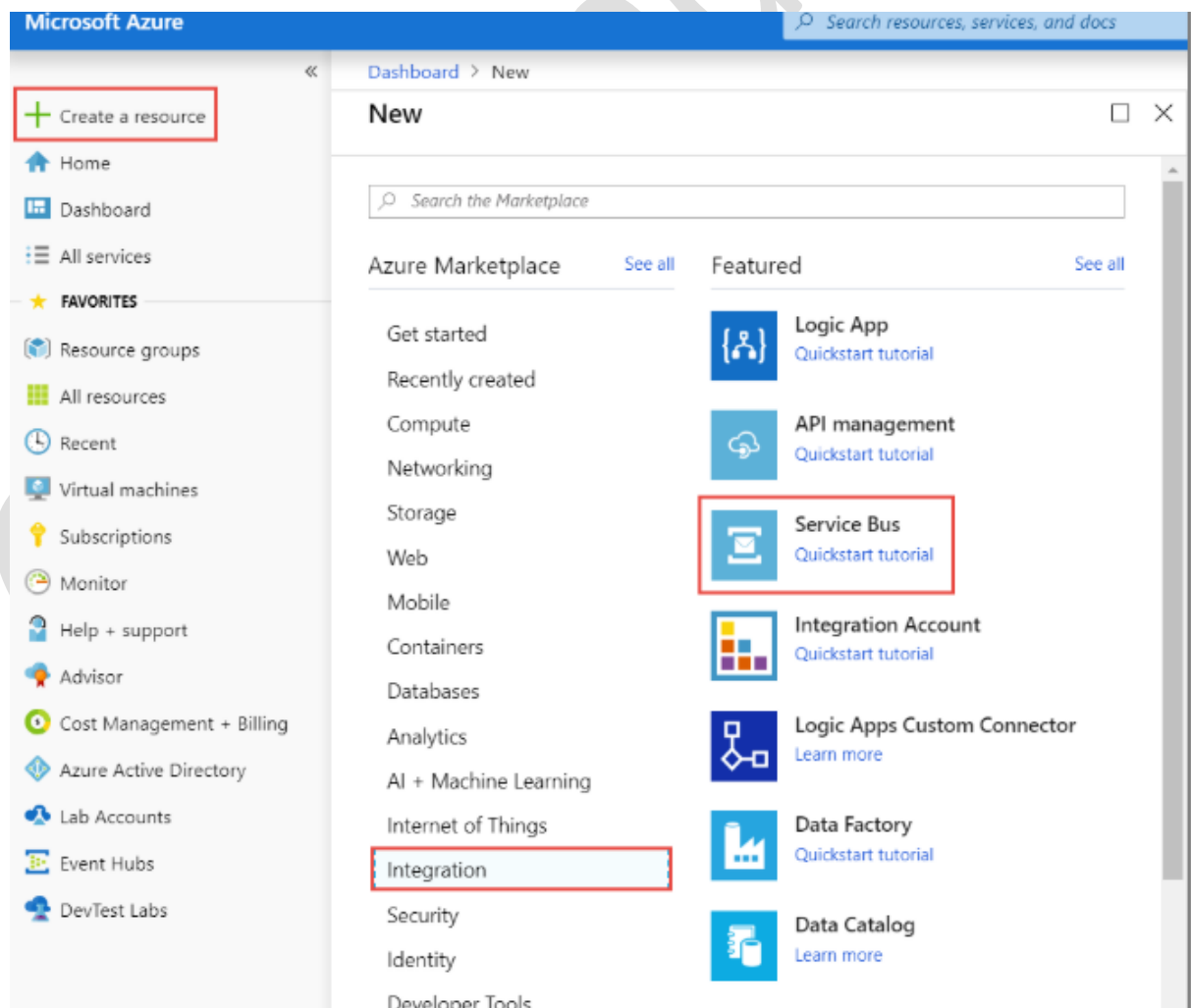
Azure- Provisioning Namespaces using Service Bus Queue

Pre-requisites

1. Should have an Azure subscription
2. Visual Studio 2017 with Update 3 / Visual Studio 2019
3. .Net Core SDK 2.0 or higher

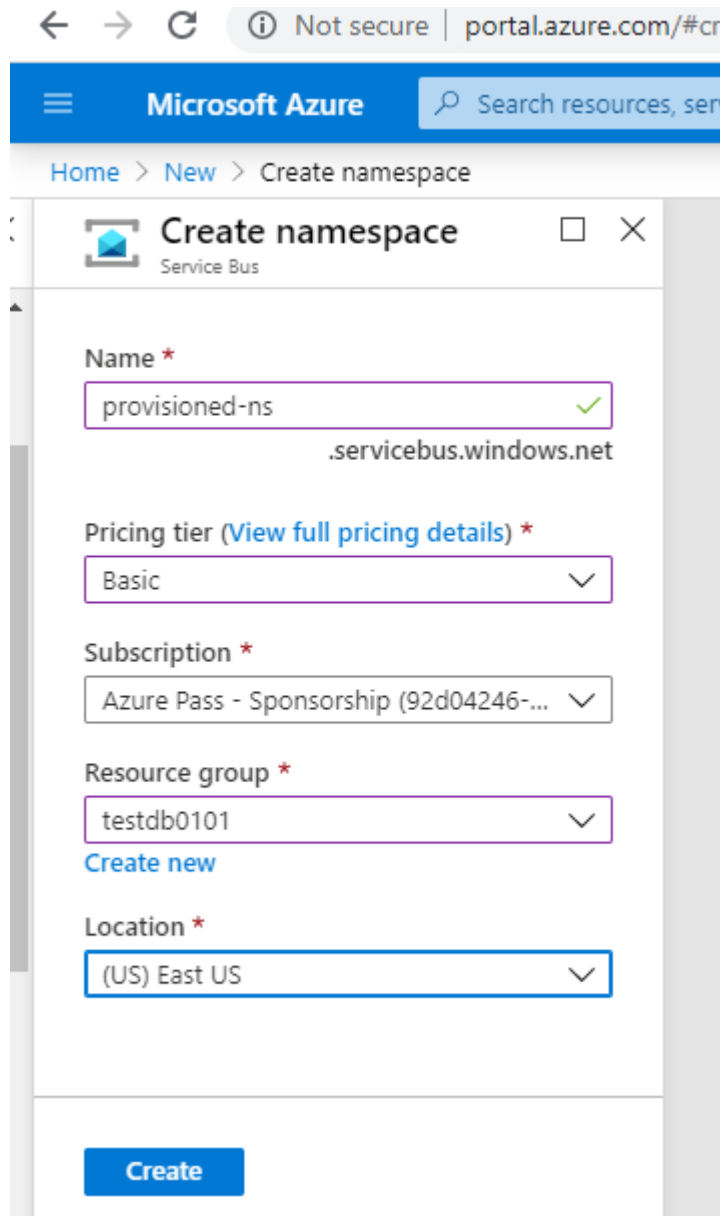
STEP 1: Create a namespace in Azure portal

1. Sign-in to Azure portal
2. Choose "Create A Resource", select Integration -> Service Bus OR type Service Bus in the search bar



- 3.
4. Click on Service Bus. It will prompt to create a new namespace

5. Fill in the details as below



The screenshot shows the 'Create namespace' form in the Microsoft Azure portal. The browser address bar shows 'portal.azure.com/#cr'. The page title is 'Create namespace' under the 'Service Bus' category. The form includes the following fields:

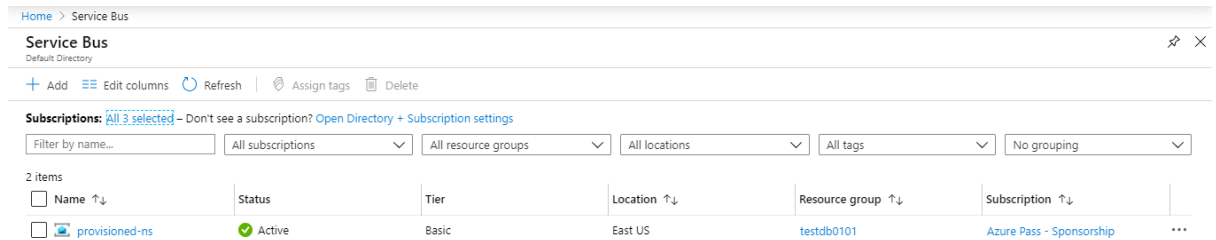
- Name ***: A text input field containing 'provisioned-ns' with a green checkmark. Below it, the domain '.servicebus.windows.net' is displayed.
- Pricing tier (View full pricing details) ***: A dropdown menu with 'Basic' selected.
- Subscription ***: A dropdown menu with 'Azure Pass - Sponsorship (92d04246-...' selected.
- Resource group ***: A dropdown menu with 'testdb0101' selected. A link 'Create new' is visible below this field.
- Location ***: A dropdown menu with '(US) East US' selected.

A blue 'Create' button is located at the bottom of the form.

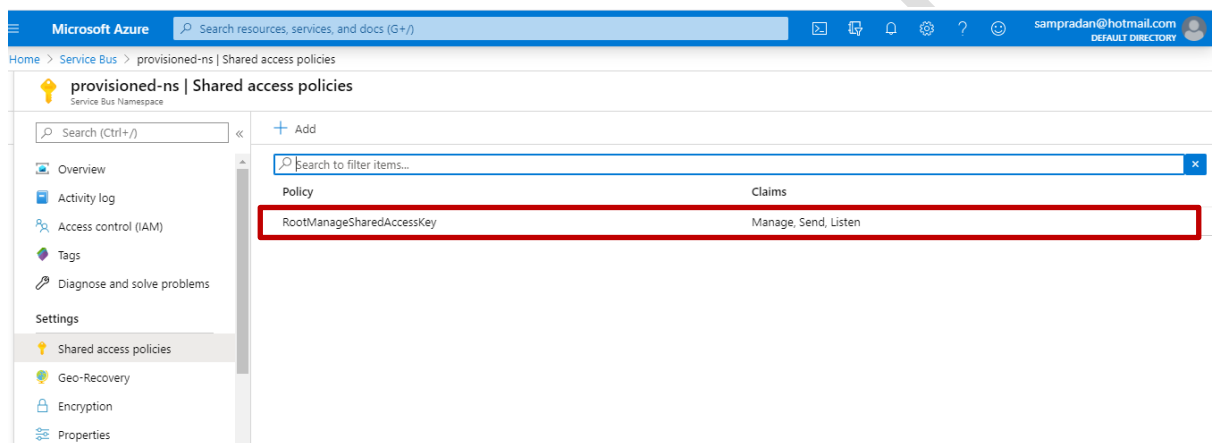
- 6.
7. Wait for the deployment to be successful

STEP 2: Creating the Queue

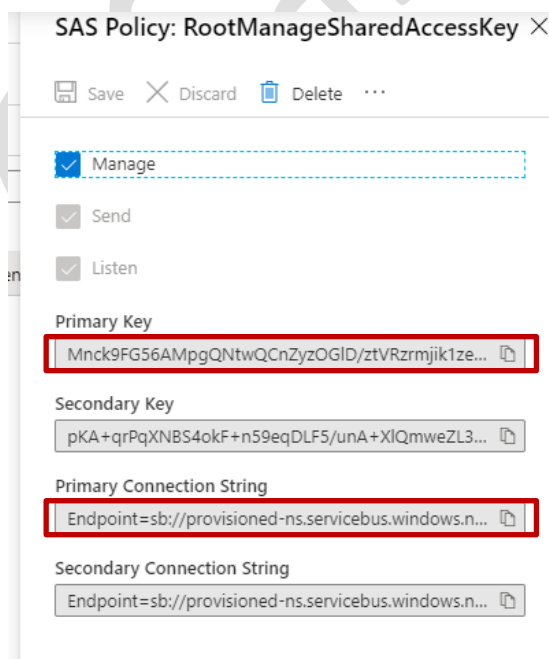
1. Navigate to the service bus, by typing "Service Bus" in the portal Search Bar
2. You should find your service bus namespace as follows



- 3.
4. Click on the namespace, in the above screenshot, then choose Shared Access Policies in the blade as follows



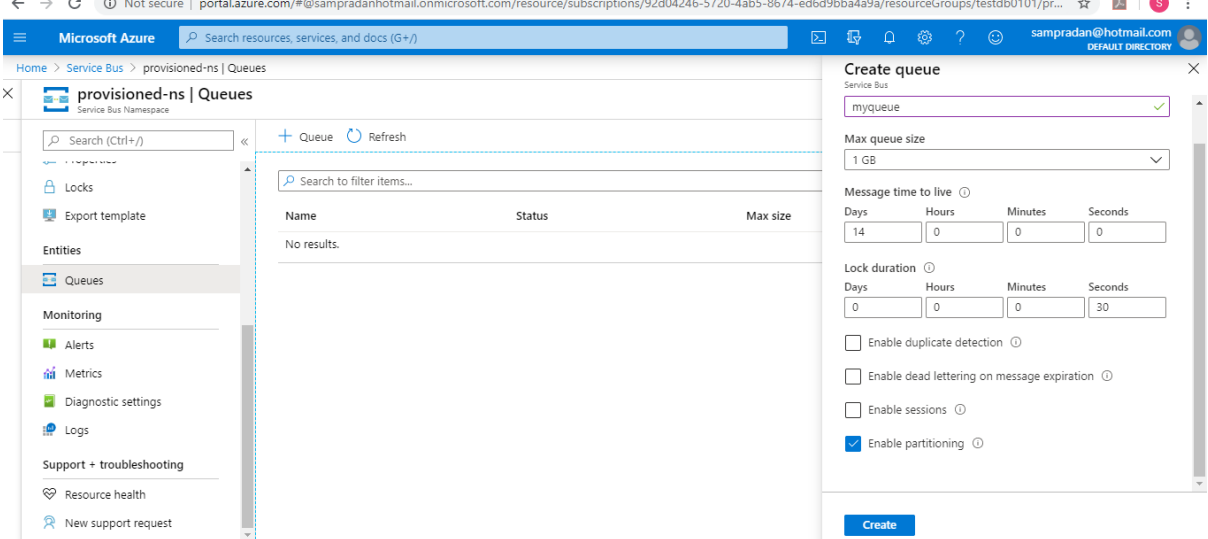
- 5.
6. Click on the "RootManagedSharedAccessKey", as highlighted above to get the following screen. Save the PRIMARY KEY and PRIMARY CONNECTION STRING in notepad



- 7.

8. On the Service Bus Namespace page, select Queues in the left navigational menu.
9. On the Queues page, select + Queue on the toolbar.
10. Enter a name for the queue, and leave the other values with their defaults.
11. Now, select Create.
12. Fill in the details as highlighted below.

13.



The screenshot shows the Microsoft Azure portal interface. On the left, the 'provisioned-ns | Queues' page is visible. The main area shows the 'Create queue' dialog. The queue name is 'myqueue'. The max queue size is '1 GB'. The message time to live is set to 14 days. The lock duration is set to 0 days, 0 hours, 0 minutes, and 30 seconds. The 'Enable partitioning' checkbox is checked. The 'Create' button is at the bottom right.

STEP 3: Interacting with service bus namespace and queues using Storage APIs

For this lab, .Net will be used as the client.

1. Clone the git repository at <https://github.com/sampradan-labs/Azure/tree/master/dotnet-sqlldb-tutorial-master>
2. Open the applications,
 - a. ProvisionedQueues
 - b. QueueReader
3. In each of the above applications, replace the value for
 - a. ServiceBusConnectionString: <the copied service bus primary connection string in STEP 2 -> Point 7>
 - b. QueueName: <name of the created queue>

4. Refer below

```
References
class Program
{
    // Connection String for the namespace can be obtained from the Azure portal under the
    // 'Shared Access policies' section.
    const string ServiceBusConnectionString = "Endpoint=sb://sbns.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=...";
    const string QueueName = "myqueue";
}
```

5. Build the Application & execute it in the following order
 - a. Execute ProvisionedQueues App => Only send messages to the service bus queue.
 - i. Check the messages count in the queue

- b.** Execute QueueReader App => Will send & receive all messages in the service bus queue. After execution, observe the service bus queue messages count. It should be zero.
- 6.** Look into the code of the application, to find
 - a.** Nuget Package addition: Microsoft.Azure.ServiceBus
 - b.** Observe the creation & Registration of messageHandlers using async, await, Tasks
- 7.** Improve the application to include two queue clients. Test the output.

@SampradaN