# Azure Service Bus

You have identified the following scenarios for message exchange between the mobile app and the web service:

1. Messages that relate to individual sales must be sent to the web service instance in the user's region.
2. Messages that relate to sales performance must be sent to all instances of the web service.

You have decided to implement a Service Bus queue for the first use case and a Service Bus topic for the second use case.

## Create a Service Bus namespace

Azure portal -> Create a resource -> Service Bus -> Create->

| **Setting** | **Value** | **Description** |
| --- | --- | --- |
| Subscription | Concierge subscription | The subscription in which this new app is created. |
| Resource group | [Sandbox resource group name] | The name of the resource group in which to create your Service Bus namespace. |
| Namespace name | [Globally unique name] | Enter a name that is unique in Azure. If you want to use the format *salesteamapp*<*Company*><*year*>, your namespace name would look like the example *salesteamappContoso2022*. |
| Location | Select from the dropdown | Choose from the free *sandbox regions* listed after this table. |
| Pricing tier | Standard | The recommended pricing tier for this exercise. |

-> Review + create -> Create -> Go to resource.

### Get Connection String for the storage account by following steps:

-> Service Bus namespace -> Left menu -> Shared access policies

-> RootManageSharedAccessKey -> Primary connection string -> Copy to clipboard.

## Create a Service Bus queue

Service Bus Namespace -> Left menu Entities -> Queues -> [+ Queue] button

-> Create queue pane -> “Name” = salesmessages -> Create.

## Create a Service Bus topic

Service Bus Namespace -> Left menu Entities -> Topics -> [+ Topic] button

-> Create topic pane

-> “Name” = salesperformancemessages

“Enable partitioning” = checked -> select Create.

## Create subscriptions for Service Bus topic

Salesperformancemessages -> [+ Subscription] button -> Create subscription pane

-> “Name” = Americas, “Max delivery count” = 100 -> Create.

Salesperformancemessages -> [+ Subscription] button -> Create subscription pane

-> “Name” = EuropeAndAsia, “Max delivery count” = 100 -> Create.

## Write code to send and receive messages by using a queue / a topic

### NuGet package required:

**Azure.Messaging.ServiceBus**

[Find the code here](https://github.com/AjinkyaApte88/General/tree/main/mslearn-service-bus/implement-message-workflows-with-service-bus/src/start).

**Service Bus queues and topics are excellent tools you can use to increase the resilience of communications within a distributed application. By acting as temporary storage locations, Service Bus queues and topics remove the requirement for direct communication between components, and they smoothly handle peaks in demand.**

**Consider using Service Bus queues and topics when you have a component that can communicate with another component in a loosely coupled configuration.**

# Azure Queue storage

Direct communication between the components of a distributed application can be problematic because it might be disrupted when network bandwidth is low or when demand is high.

We've seen this in our system: the web portal calls a web service, which works great if the service responds in a timely manner. High traffic causes problems and so the plan is to use a queue to eliminate the direct link between the front-end apps and your middle-tier web service.

## Create a storage account

| **Parameter** | **Value** |
| --- | --- |
| Name | Sets the name. Remember that storage accounts use the name to generate a public URL - so it must be unique. In addition, the account name must be between 3 and 24 characters, and be composed of numbers and lowercase letters only. We recommend you use the prefix **articles** with a random number suffix, but you can use whatever you like. |
| Resource group | Supplies the **Resource Group**. Use *[sandbox resource group name]* as the value. |
| Account Type | Sets the **Storage Account type**: *StorageV2* to create a general-purpose V2.account. |
| Redundancy | Sets the **Replication and Storage type**. It defaults to *Standard\_RAGRS (Global Redundancy)*.  Let's use *Standard\_LRS* (locally redundant within the datacenter). |
| location | Sets the **Location** independent of the resource group owner. It's optional, but you can use it to place the queue in a different region than the resource group. Place it close to you. |

-> Review + create -> Create -> Go to resource.

### Get Connection String for the storage account by following steps:

Azure storage account -> Left menu -> Access keys -> Connection String -> Show

-> Copy to clipboard

## Programmatically create and access a queue

### NuGet package required:

**Azure.Storage.Queues**

[Find the code here](https://github.com/AjinkyaApte88/General/tree/main/mslearn-storage-queues/start).

**Using the Azure.Storage.Queues package for .NET can help to make a distributed application more reliable and resilient to failures and periods of high demand.**