

```
# This retrieves the subscription id of the account
#   in which you're logged in.
# This field is used to set up the routing queries.
subscriptionID=$(az account show --query id)

# Concatenate this number onto the resources that have to be globally unique.
# You can set this to "" or to a specific value if you don't want it to be random.
# This retrieves a random value.
randomValue=$RANDOM

# Set the values for the resource names that
#   don't have to be globally unique.
location=westus
resourceGroup=AirQuality
iotHubConsumerGroup=AirQualityConsumers
containerName=airqualityresults

# Create the resource group to be used
#   for all the resources for this tutorial.
az group create --name $resourceGroup \
  --location $location

# The IoT hub name must be globally unique,
#   so add a random value to the end.
iotHubName=AirQualityHub$randomValue
echo "IoT hub name = " $iotHubName

# Create the IoT hub.
az iot hub create --name $iotHubName \
  --resource-group $resourceGroup \
  --sku S1 --location $location

# Add a consumer group to the IoT hub for the 'events' endpoint.
az iot hub consumer-group create --hub-name $iotHubName \
  --name $iotHubConsumerGroup

# The storage account name must be globally unique,
#   so add a random value to the end.
storageAccountName=airquality$randomValue
echo "Storage account name = " $storageAccountName

# Create the storage account to be used as a routing destination.
az storage account create --name $storageAccountName \
  --resource-group $resourceGroup \
  --location $location \
  --sku Standard_LRS

# Get the primary storage account key.
#   You need this to create the container.
storageAccountKey=$(az storage account keys list \
  --resource-group $resourceGroup \
  --account-name $storageAccountName \
  --query "[0].value" | tr -d '"')

# See the value of the storage account key.
echo "storage account key = " $storageAccountKey

# Create the container in the storage account.
containerName=airqualityresults
az storage container create --name $containerName \
  --account-name $storageAccountName \
  --account-key $storageAccountKey \
  --public-access off

# The Service Bus namespace must be globally unique,
#   so add a random value to the end.
```

```
sbNamespace=AirQualitySBNamespace$randomValue
echo "Service Bus namespace = " $sbNamespace

# Create the Service Bus namespace.
az servicebus namespace create --resource-group $resourceGroup \
    --name $sbNamespace \
    --location $location

# The Service Bus queue name must be globally unique,
# so add a random value to the end.
sbQueueName=AirQualitySBQueue$randomValue
echo "Service Bus queue name = " $sbQueueName

# Create the Service Bus queue to be used as a routing destination.
az servicebus queue create --name $sbQueueName \
    --namespace-name $sbNamespace \
    --resource-group $resourceGroup

# Create the IoT Device
iotDeviceName=e00fce685e7813a264b90d5e

# Create the IoT device identity to be used for testing.
#az iot hub device-identity create --device-id $iotDeviceName \
#    --hub-name $iotHubName

# Retrieve the information about the device identity, then copy the primary key to
# Notepad. You need this to run the device simulation during the testing phase.
az iot hub device-identity show --device-id $iotDeviceName \
    --hub-name $iotHubName

##### ROUTING FOR STORAGE #####

endpointName="airqualityendpoint"
endpointType="azurestoragecontainer"
routeName="airqualitystorageroute"
#condition='level="storage"'

# Get the connection string for the storage account.
# Adding the "-o tsv" makes it be returned without the default double quotes around it.
storageConnectionString=$(az storage account show-connection-string --name storageAccountName
--query connectionString -o tsv)

# Create the routing endpoint for storage.
az iot hub routing-endpoint create \
    --connection-string $storageConnectionString \
    --endpoint-name $endpointName \
    --endpoint-resource-group $resourceGroup \
    --endpoint-subscription-id $subscriptionID \
    --endpoint-type $endpointType \
    --hub-name $iotHubName \
    --container $containerName \
    --resource-group $resourceGroup \
    --encoding json

# Create the route for the storage endpoint.
az iot hub route create \
    --name $routeName \
    --hub-name $iotHubName \
    --source devicemessages \
    --resource-group $resourceGroup \
    --endpoint-name $endpointName \
```

```
--enabled \  
--condition $condition
```