To access the **EventHubManagementClient** in Azure, you need appropriate **Azure Role-Based Access Control (RBAC)** permissions assigned to your account or service principal. Specifically:

### **Required Access**

1. **Azure Subscription-Level Access**:
   1. **Contributor** or **Owner** role on the Azure subscription, resource group, or Event Hubs namespace where the Event Hubs resource resides.
   2. Alternatively, a custom role with permissions to manage Event Hubs resources.
2. **Specific Permissions**: If using a custom role, ensure it includes the following permissions:
   1. Microsoft.EventHub/namespaces/\*: For managing Event Hubs namespaces.
   2. Microsoft.EventHub/eventhubs/\*: For managing individual Event Hubs.
   3. Microsoft.EventHub/authorizationRules/\*: For managing access policies.
3. **Authentication Requirements**:
   1. **Azure Active Directory (AAD)**: Authenticate using a user account or a service principal with the above roles/permissions.
   2. **Service Principal**: If using an application to manage Event Hubs, the service principal must have the necessary RBAC roles assigned.

### **Access Scenarios**

* **Portal Access**: Ensure your user account has the necessary role (e.g., Contributor) at the subscription or resource group level.
* **Programmatic Access** (e.g., using the Azure SDK for Python):
  + Authenticate with a service principal or managed identity that has the appropriate roles.
  + Use the Azure Identity library for authentication.

### **Example: Assigning RBAC Role**

To assign a role, follow these steps:

1. Navigate to the **Azure portal**.
2. Go to the **Event Hubs namespace** or resource group.
3. Click on **Access control (IAM)**.
4. Click **Add role assignment** and select the appropriate role (e.g., Contributor).
5. Assign the role to a user, group, or service principal.

Would you like assistance with a specific scenario or setup?

4o

##### **You said:**

How can we read all the detail about event hubs namespace and event hubs settings etc. using EventHubManagementClient using Python? Provide me detail code.

###### **ChatGPT said:**

ChatGPT

To read details about an Event Hubs namespace and its settings using the EventHubManagementClient in Python, you can use the Azure SDK for Python. Below is a detailed example that demonstrates how to retrieve information about an Event Hubs namespace and its associated Event Hubs.

### **Prerequisites**

1. Install the Azure SDK for Event Hubs Management:

bash

Copy code

pip install azure-mgmt-eventhub azure-identity

1. Ensure your Azure account or service principal has the necessary RBAC roles (e.g., Contributor or Reader) on the Event Hubs namespace.

### **Code Example**

python

Copy code

from azure.identity import DefaultAzureCredential  
from azure.mgmt.eventhub import EventHubManagementClient  
  
# Replace with your Azure subscription ID  
subscription\_id = "your-subscription-id"  
  
# Replace with your Event Hubs namespace details  
resource\_group\_name = "your-resource-group-name"  
namespace\_name = "your-namespace-name"  
  
# Authenticate using DefaultAzureCredential  
# Ensure your environment is set up for authentication (e.g., AZURE\_CLIENT\_ID, AZURE\_TENANT\_ID, AZURE\_CLIENT\_SECRET for service principal)  
credential = DefaultAzureCredential()  
  
# Initialize the EventHubManagementClient  
client = EventHubManagementClient(credential, subscription\_id)  
  
# Get details of the Event Hubs namespace  
namespace\_details = client.namespaces.get(resource\_group\_name, namespace\_name)  
print("Namespace Details:")  
print(f"Name: {namespace\_details.name}")  
print(f"Location: {namespace\_details.location}")  
print(f"SKU: {namespace\_details.sku.name}")  
print(f"Tags: {namespace\_details.tags}")  
print(f"Created At: {namespace\_details.created\_at}")  
print(f"Updated At: {namespace\_details.updated\_at}")  
  
# List all Event Hubs in the namespace  
print("\nEvent Hubs in the Namespace:")  
event\_hubs = client.event\_hubs.list\_by\_namespace(resource\_group\_name, namespace\_name)  
for event\_hub in event\_hubs:  
 print(f"- Name: {event\_hub.name}")  
 print(f" Partition Count: {event\_hub.partition\_count}")  
 print(f" Message Retention (days): {event\_hub.message\_retention\_in\_days}")  
 print(f" Status: {event\_hub.status}")  
 print(f" Created At: {event\_hub.created\_at}")  
 print(f" Updated At: {event\_hub.updated\_at}")  
  
# List Authorization Rules for the namespace  
print("\nAuthorization Rules for the Namespace:")  
auth\_rules = client.namespaces.list\_authorization\_rules(resource\_group\_name, namespace\_name)  
for rule in auth\_rules:  
 print(f"- Name: {rule.name}")  
 print(f" Rights: {rule.rights}")  
  
# List Authorization Rules for a specific Event Hub (Optional)  
event\_hub\_name = "your-event-hub-name"  
print(f"\nAuthorization Rules for Event Hub '{event\_hub\_name}':")  
event\_hub\_auth\_rules = client.event\_hubs.list\_authorization\_rules(resource\_group\_name, namespace\_name, event\_hub\_name)  
for rule in event\_hub\_auth\_rules:  
 print(f"- Name: {rule.name}")  
 print(f" Rights: {rule.rights}")

### **Explanation**

1. **Authentication**:
   1. The DefaultAzureCredential class handles authentication automatically. It uses environment variables or managed identities if running in Azure.
2. **Namespace Details**:
   1. The client.namespaces.get method retrieves detailed information about the namespace.
3. **List Event Hubs**:
   1. The client.event\_hubs.list\_by\_namespace method lists all Event Hubs within the specified namespace.
4. **Authorization Rules**:
   1. Use client.namespaces.list\_authorization\_rules to list namespace-level authorization rules.
   2. Use client.event\_hubs.list\_authorization\_rules to list authorization rules for a specific Event Hub.

### **Output**

The script outputs:

* Details of the Event Hubs namespace.
* A list of Event Hubs in the namespace with their settings.
* Authorization rules for the namespace and specific Event Hubs.

Would you like help with running this script or setting up the environment?