As of my last knowledge update in January 2022, Microsoft Azure provides a range of services, including AI services. However, the specifics of services and features can change over time. If there have been updates or changes to Azure's offerings, I recommend checking the official Azure documentation for the most up-to-date information.

Assuming that Azure OpenAI services are available and the information remains relevant, here's a general tutorial on how to use Azure AI services with OpenAI:

## Tutorial: Using Azure AI with OpenAI

### Prerequisites:

1. **Azure Account:** Ensure you have an active Azure account. You can sign up for one [here](https://azure.com/free).

2. **Azure OpenAI Service:** Verify that the OpenAI service is available on Azure. Refer to the Azure portal or documentation for the latest information.

3. **API Key:** Obtain the API key for the OpenAI service. This is crucial for authentication.

### Step 1: Create a New Azure OpenAI Resource

1. Log in to the [Azure portal](https://portal.azure.com/).

2. Navigate to the Azure Marketplace and search for the OpenAI service.

3. Select the OpenAI service and follow the prompts to create a new resource.

4. Configure the resource, providing necessary details such as region, pricing tier, and resource group.

5. Once created, note down the API key associated with your new resource.

### Step 2: Set Up Your Development Environment

1. Install the Azure SDK for your preferred programming language (Python, .NET, Node.js, etc.).

2. Use the Azure CLI or Azure Portal to set up the necessary environment variables, including your Azure subscription ID and the API key for the OpenAI service.

### Step 3: Access OpenAI API

1. In your preferred programming language, use the Azure SDK to authenticate and access the OpenAI API.

   Example (Python using Azure SDK):

   ```python
   from azure.identity import DefaultAzureCredential
   from azure.ai.textanalytics import TextAnalyticsClient

   credential = DefaultAzureCredential()
   openai_client = OpenAIClient(credential, endpoint="your_openai_endpoint")
   ```

### Step 4: Use OpenAI Service

1. Once connected, you can use the OpenAI service for various tasks.

   Example (Python using Azure SDK):

   ```python
   response = openai_client.generate(model="text-davinci-002", prompt="Translate the following English text to French: '{}'", inputs="Hello, how are you?")
   print(response['choices'][0]['text'])
   ```

~~~
```
~~~

This is a simple example. Depending on the OpenAI models and services available, you can perform various tasks like text generation, language translation, sentiment analysis, etc.

### Step 5: Handle Responses

1. Parse and handle the responses from the OpenAI service according to your application needs.

Example (Python):

```python
if response['choices'][0]['finish_reason'] == 'stop':
    print("Model stopped because it reached the maximum token limit.")
else:
    print(response['choices'][0]['text'])
```

### Step 6: Clean Up

1. When you're finished, clean up resources to avoid unnecessary costs.

### Conclusion

This tutorial provides a basic overview of using Azure AI with OpenAI services. Be sure to refer to the latest Azure and OpenAI documentation for any updates or changes in the services.