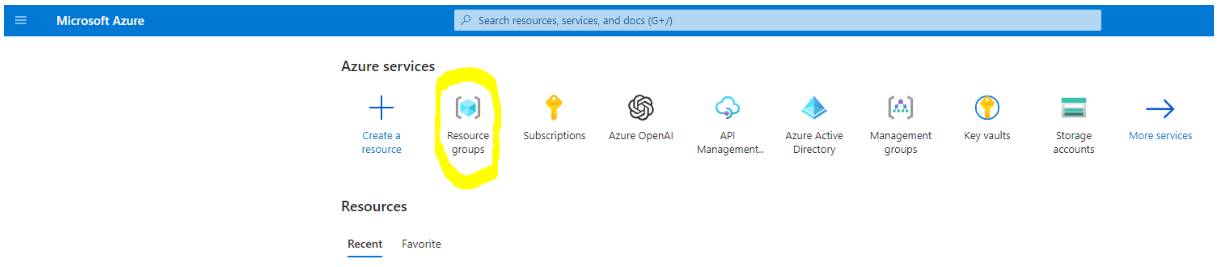
Instructions for User access to Tartu University OpenAI Service.

General access to the OpenAI Service is granted through portal.azure.com. In there a user should have access to **TU-OpenAI-LLM-Department** Resource group through one of two possible roles:

- **Contributor** – user with the ability to deploy OpenAI models and set configurations as well as fine tune models on 'training' data sets
- **Reader** – user with the ability to only interact with OpenAI models by asking for completions or embeddings

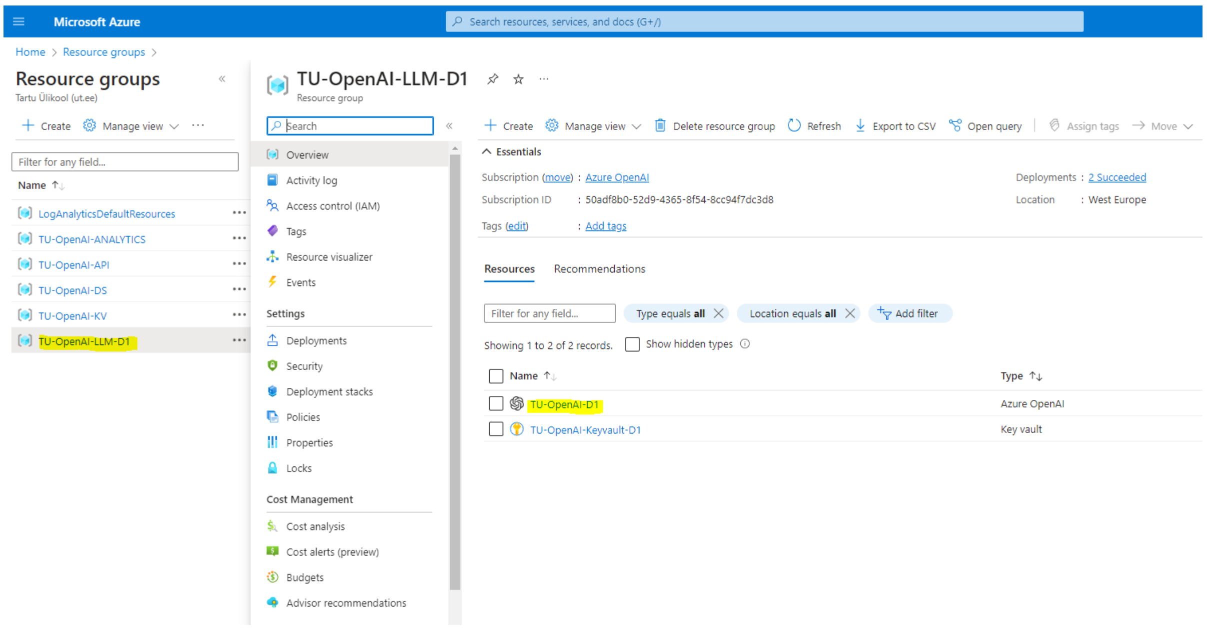
Most users for a resource group should fall under the second category with only select **Contributor** users that manage the department workspace

To access the OpenAI user interface we need to navigate to **portal.azure.com** in our browser (log in with our university credentials) and from there we navigate to **Resource Groups**:



Afterwards we select the resource group with our department name (e.g., **TU-OpenAI-LLM-CompSc**). In the resource group we have two resources:

- **OpenAI** - which is our user interface for deploying, fine tuning and interacting with GPT models
- **Key Vault** – which is used to store API keys to access the deployed OpenAI models for application development end experiments.



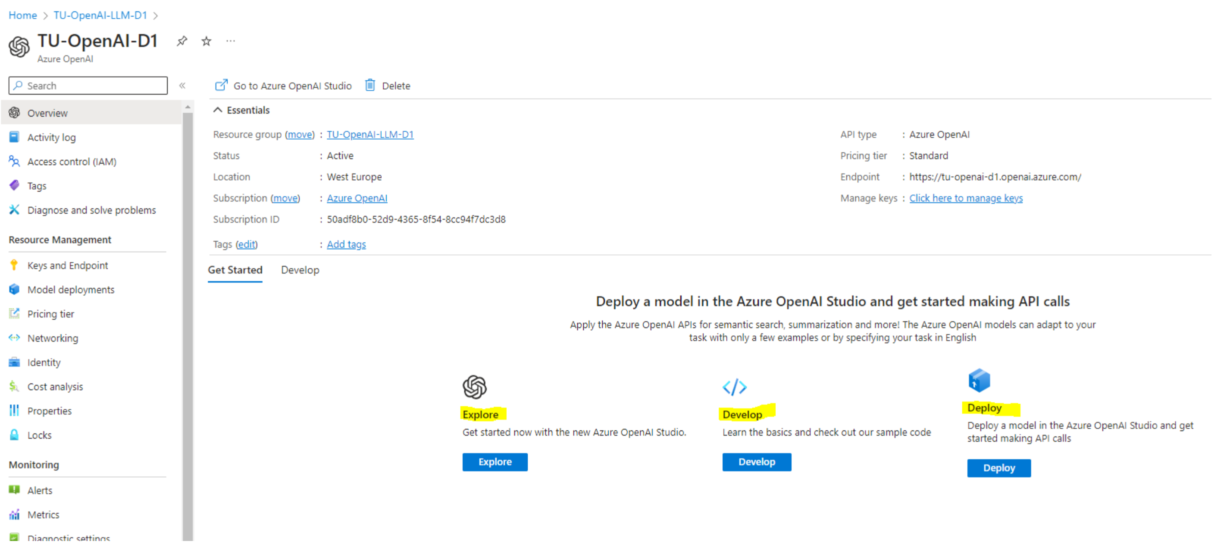
LogAnalyticsDefaultResources
TU-OpenAI-ANALYTICS
TU-OpenAI-API
TU-OpenAI-DS
TU-OpenAI-KV
TU-OpenAI-LLM-D1

Subscription (move) : Azure OpenAI
Subscription ID : 50adf8b0-52d9-4365-8f54-8cc94f7dc3d8
Tags (edit) : Add tags
Deployments : 2 Succeeded
Location : West Europe

Showing 1 to 2 of 2 records.

Name	Type
TU-OpenAI-D1	Azure OpenAI
TU-OpenAI-Keyvault-D1	Key vault

OpenAI Workspace environment allows the user to explore, develop and deploy models.



Resource group (move)	: TU-OpenAI-LLM-D1
Status	: Active
Location	: West Europe
Subscription (move)	: Azure OpenAI
Subscription ID	: 50adf8b0-52d9-4365-8f54-8cc94f7dc3d8
Tags (edit)	: Add tags
API type	: Azure OpenAI
Pricing tier	: Standard
Endpoint	: https://tu-openai-d1.openai.azure.com/
Manage keys	: Click here to manage keys

Deploy a model in the Azure OpenAI Studio and get started making API calls

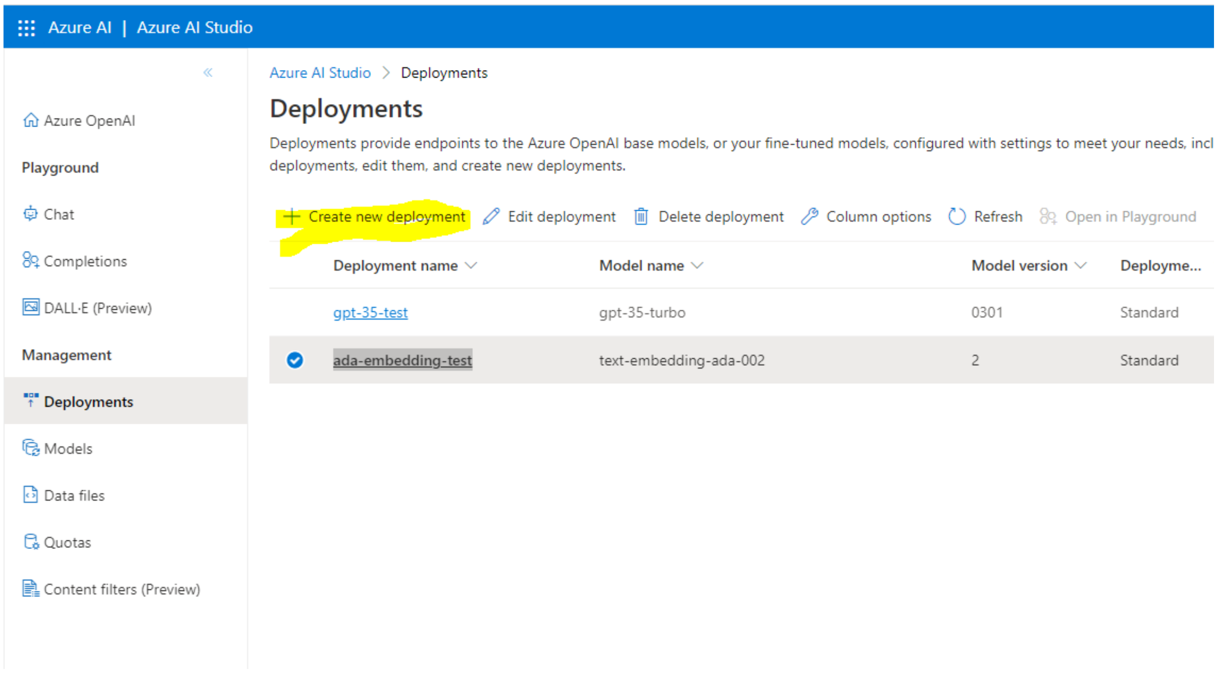
Apply the Azure OpenAI APIs for semantic search, summarization and more! The Azure OpenAI models can adapt to your task with only a few examples or by specifying your task in English

Explore
Get started now with the new Azure OpenAI Studio.
Explore

Develop
Learn the basics and check out our sample code
Develop

Deploy
Deploy a model in the Azure OpenAI Studio and get started making API calls
Deploy

We can create a new deployment by selecting ‘**Model Deployment**’ or ‘**Deploy**’ actions. This is then followed by choosing ‘**Create new deployment**’ option. The Deployment name can then be referenced in the API calls.

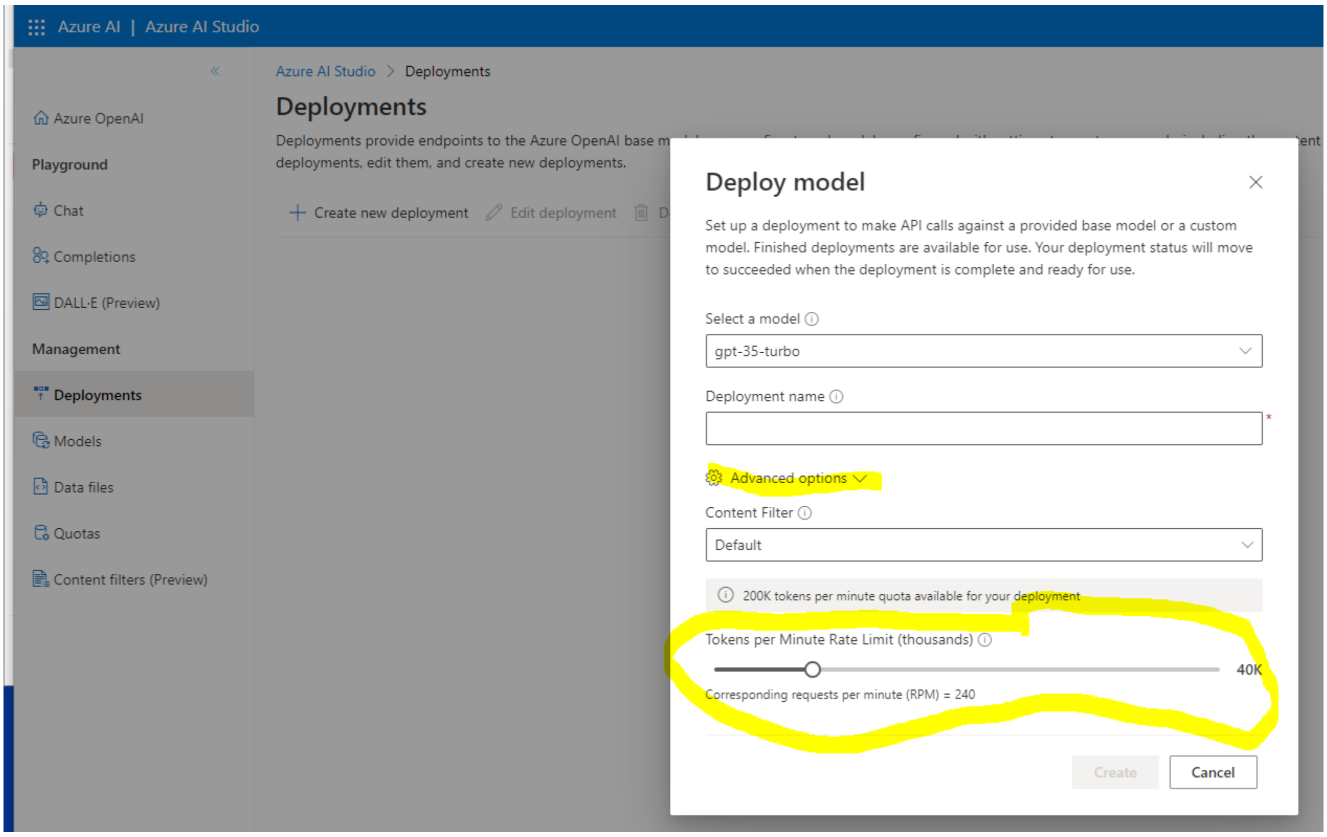


Deployments

Deployments provide endpoints to the Azure OpenAI base models, or your fine-tuned models, configured with settings to meet your needs, incl deployments, edit them, and create new deployments.

Deployment name	Model name	Model version	Deployme...
gpt-35-test	gpt-35-turbo	0301	Standard
ada-embedding-test	text-embedding-ada-002	2	Standard

When creating new deployment, it is necessary to select advanced options (available once the model has been chosen) and reduce the **Tokens Per Minute** value to the lowest acceptable one. This value corresponds to the total amount of tokens that can be sent to the model each minute. Since this value counts towards Tartu University global quota, it is important to have a justification for having values above 40000.



Deployments

Deployments provide endpoints to the Azure OpenAI base m... deployments, edit them, and create new deployments.

Deploy model

Set up a deployment to make API calls against a provided base model or a custom model. Finished deployments are available for use. Your deployment status will move to succeeded when the deployment is complete and ready for use.

Select a model: gpt-35-turbo

Deployment name: *

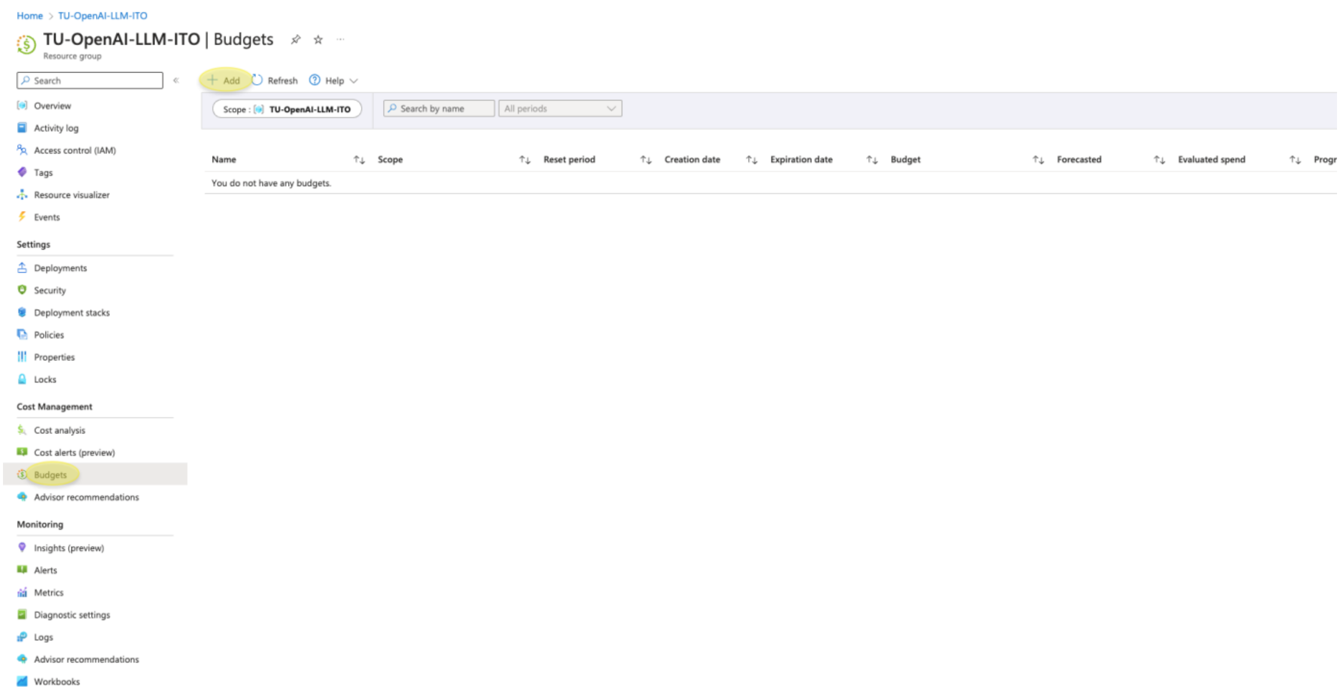
Content Filter: Default

200K tokens per minute quota available for your deployment

Tokens per Minute Rate Limit (thousands): 40K

Corresponding requests per minute (RPM) = 240

You can also add a budget option by choosing your ‘**Resource group’** and navigating to ‘**Cost Management’** -> **‘Budgets’**.



TU-OpenAI-LLM-ITO | Budgets

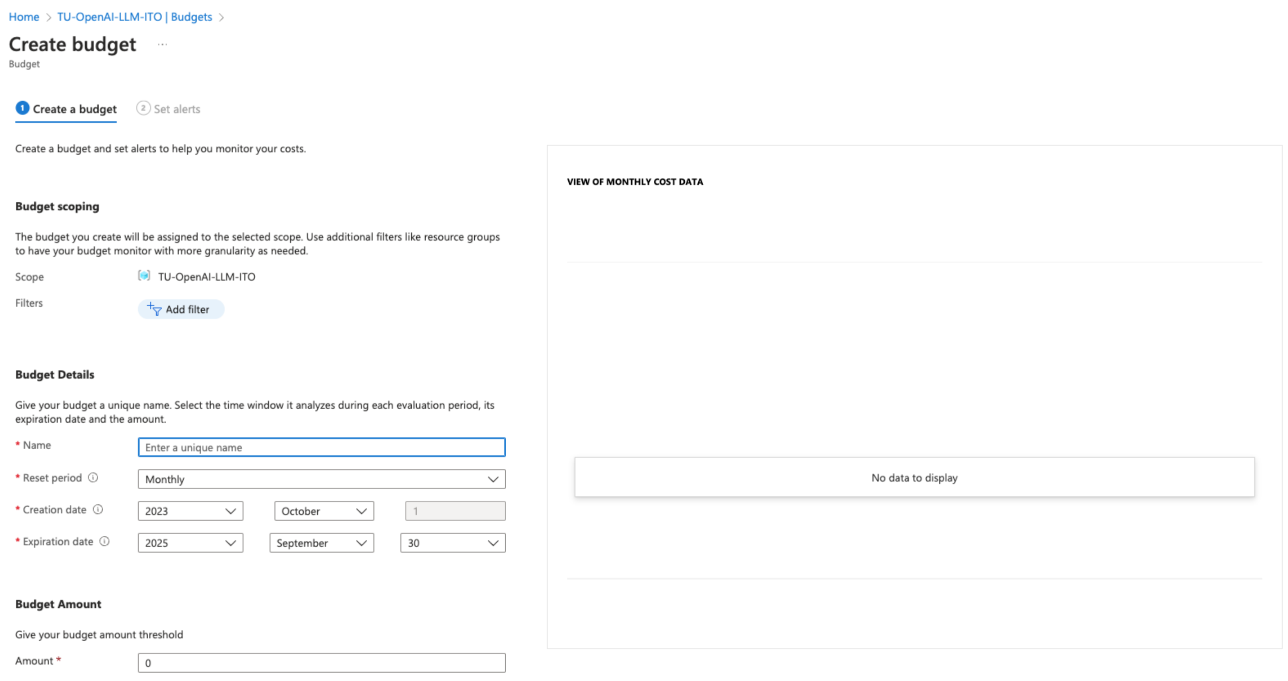
Resource group

Scope : TU-OpenAI-LLM-ITO

Search by name | All periods

Name	Scope	Reset period	Creation date	Expiration date	Budget	Forecasted	Evaluated spend	Progr
You do not have any budgets.								

From there you can name your budget, choose creation and expiration date and '**amount (in euros)'** for your limit.


Home > TU-OpenAI-LLM-ITO | Budgets >

Create budget ···

Budget

1 Create a budget 2 Set alerts

Create a budget and set alerts to help you monitor your costs.

Budget scoping

The budget you create will be assigned to the selected scope. Use additional filters like resource groups to have your budget monitor with more granularity as needed.

Scope: TU-OpenAI-LLM-ITO

Filters: Add filter

Budget Details

Give your budget a unique name. Select the time window it analyzes during each evaluation period, its expiration date and the amount.

* Name: Enter a unique name
* Reset period: Monthly
* Creation date: 2023 | October | 1
* Expiration date: 2025 | September | 30

Budget Amount

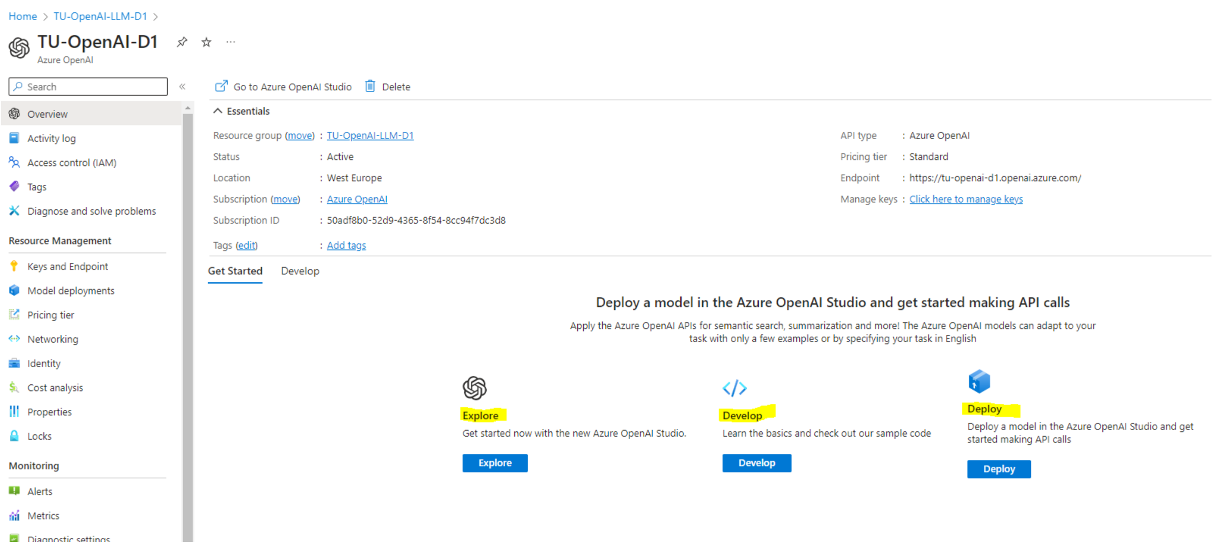
Give your budget amount threshold

Amount *: 0

VIEW OF MONTHLY COST DATA

No data to display

The '**Explore**' section in the OpenAI overview can be used to interact with the deployed models from the user interface.


Home > TU-OpenAI-LLM-D1 >

TU-OpenAI-D1

Azure OpenAI

Essentials

Resource group (move)	: TU-OpenAI-LLM-D1	API type	: Azure OpenAI
Status	: Active	Pricing tier	: Standard
Location	: West Europe	Endpoint	: https://tu-openai-d1.openai.azure.com/
Subscription (move)	: Azure OpenAI	Manage keys	: Click here to manage keys
Subscription ID	: 50adf8b0-52d9-4365-8f54-8cc94f7dc3d8		
Tags (edit)	: Add tags		

Get Started Develop

Deploy a model in the Azure OpenAI Studio and get started making API calls

Apply the Azure OpenAI APIs for semantic search, summarization and more! The Azure OpenAI models can adapt to your task with only a few examples or by specifying your task in English

Explore — Get started now with the new Azure OpenAI Studio. [Explore]

Develop — Learn the basics and check out our sample code [Develop]

Deploy — Deploy a model in the Azure OpenAI Studio and get started making API calls [Deploy]

For large scale interactions for experimental or application purposes the preferred option is to access the API through a different endpoint that is managed by Tartu University directly. For Reader users this is the only way to interact with the OpenAI API outside of the User interface seen before.

The Tartu University managed API can be accessed using REST interface with POST method. The reference for Open AI REST API can be found here:

https://learn.microsoft.com/en-us/azure/ai-services/openai/reference

For current implementation only 'chat completions' (GPT-35-Turbo model) and 'Embeddings' (text-embedding-ada-002 model) are supported.

For the API REST access for chat completions, you will need 3 parameters to access the API:

- **Your resource name** – the last part of your resource group name in azure, this should be provided via email together with these instructions
- **Your deployment name** – this is the deployment name found in Azure OpenAI Studio, it is the deployment name a **Contributor** user creates when deploying a model (see image above)
- **Your API Key** – this is found in the Key Vault resource in your Azure resource group (see access instructions bellow pages 6 - 7)
- **(optional) Your model name** – when using Python openai library a model name has to be provided, it is the model name that is found under deployments in Azure OpenAI Studio (see image above)

These are three examples for accessing Tartu University managed OpenAI API

Using CURL:

```
curl --request POST \
  --url 'https://tu-openai-api-management.azure-api.net/<your resource
name>/openai/deployments/<your deployment name>/chat/completions?api-ver-
sion=2023-07-01-preview' \
  --header 'Content-Type: application/json' \
  --header 'User-Agent: Insomnia/2023.5.6' \
  --header 'api-key: <your key here>' \
  --data '{
    "messages": [
        {
            "role": "user",
            "content": "Does Azure OpenAI support customer managed keys?"
        }
    ],
    "max_tokens": 200,
    "temperature": 0.3
}'
```

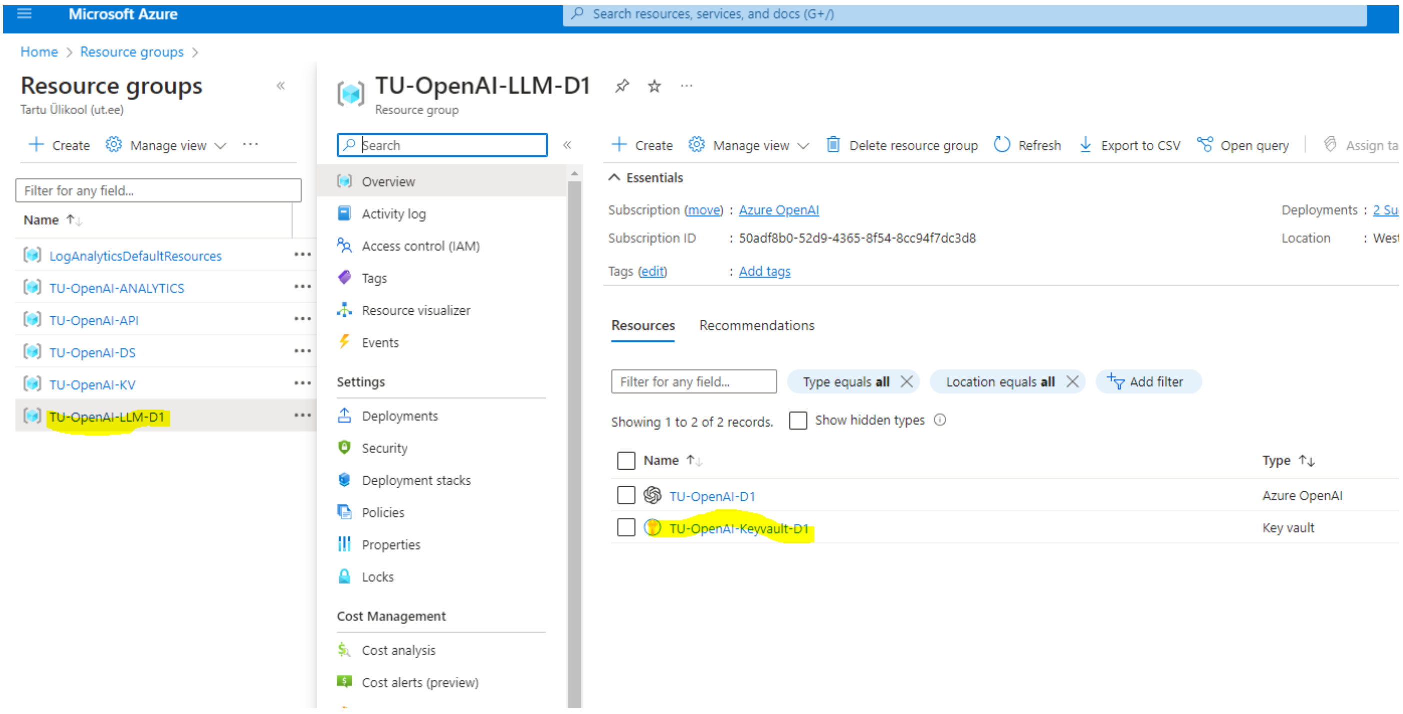
Using Python requests package:

```
import requests

url = "https://tu-openai-api-management.azure-api.net/<your resource
name>/openai/deployments/<your deployment name>/chat/completions "
querystring = {"api-version":"2023-07-01-preview"}

payload = {
    "messages": [
        {
            "role": "user",
            "content": "Does Azure OpenAI support customer managed keys?"
        }
    ],
    "max_tokens": 200,
    "temperature": 0.3
}
headers = {
    "Content-Type": "application/json",
    "User-Agent": "Insomnia/2023.5.6",
    "api-key": "<your key here>"
}

response = requests.request("POST", url, json=payload, headers=headers,
params=querystring)
```

Using Python openai package:

```
import openai
openai.api_type = "azure"
openai.api_key = "<your key here>"
openai.api_base = "https://tu-openai-api-management.azure-api.net/<your
resource name>"
openai.api_version = "2023-07-01-preview"

# create a chat completion
chat_completion = openai.ChatCompletion.create(
    deployment_id = "<your deployment name>",
    model="<your model name>",
    messages=[{"role": "user","content": "Does Azure OpenAI support customer
managed keys?"}]
)
```

To access the value of your **API Key** for Tartu Managed OpenAI API we need to go to **Key Vault** in our resource group which hosts the two API keys that can be used for access. Both keys are valid and are used to rotate access when changing keys.

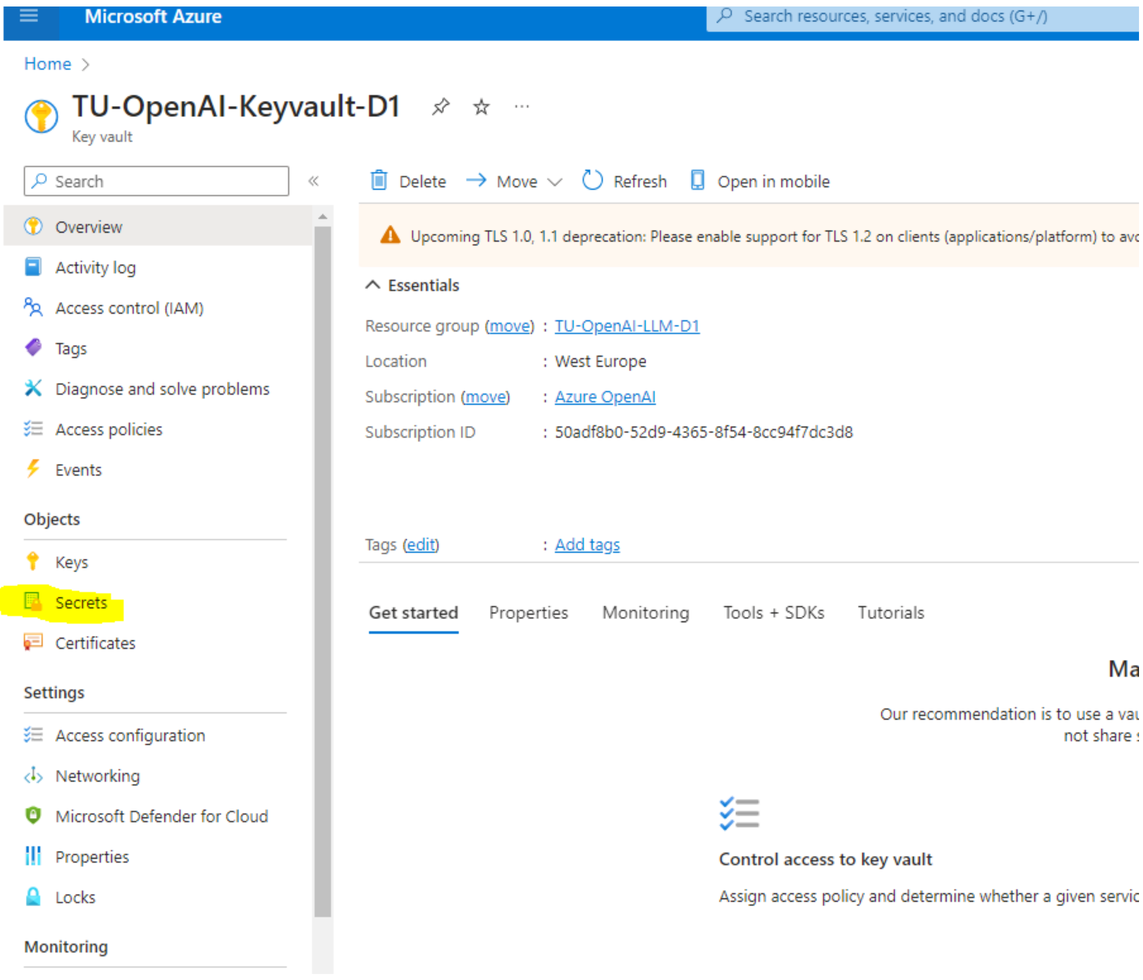


Subscription (move) : Azure OpenAI
Subscription ID : 50adf8b0-52d9-4365-8f54-8cc94f7dc3d8
Tags (edit) : Add tags
Deployments : 2 Su
Location : West

Showing 1 to 2 of 2 records.

Name	Type
TU-OpenAI-D1	Azure OpenAI
TU-OpenAI-Keyvault-D1	Key vault

The specific keys are located under ‘**Secrets’** section in the resource:



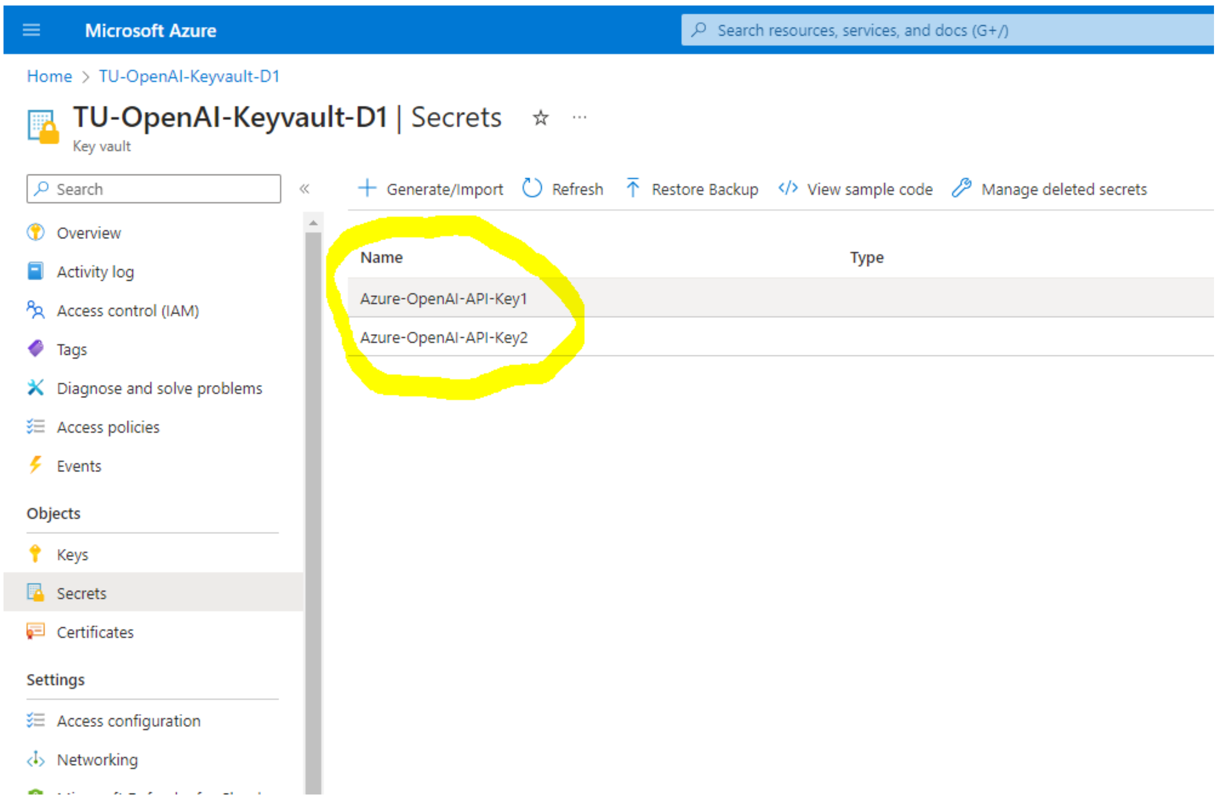
Upcoming TLS 1.0, 1.1 deprecation: Please enable support for TLS 1.2 on clients (applications/platform) to avo

Resource group (move) : TU-OpenAI-LLM-D1
Location : West Europe
Subscription (move) : Azure OpenAI
Subscription ID : 50adf8b0-52d9-4365-8f54-8cc94f7dc3d8
Tags (edit) : Add tags

Our recommendation is to use a vau
not share s

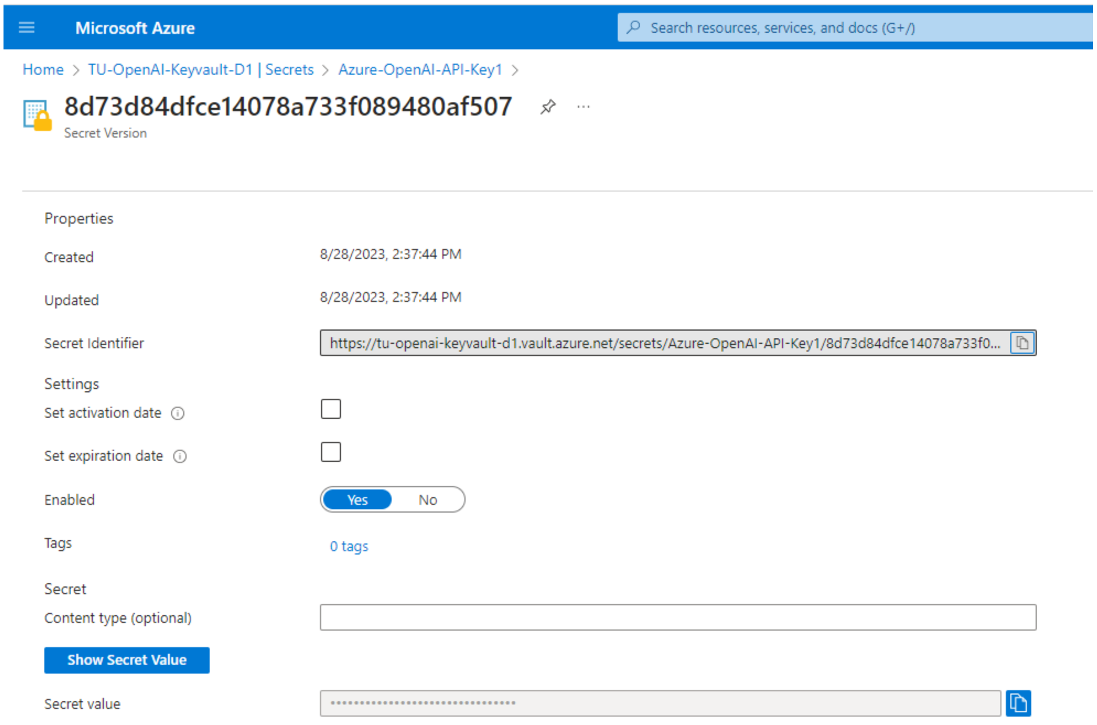
Assign access policy and determine whether a given servic

Keys can be accessed by clicking on them to open the versioning and then clicking once more on the most recent version:



Name	Type
Azure-OpenAI-API-Key1	
Azure-OpenAI-API-Key2	

Version:


Microsoft Azure
Search resources, services, and docs (G+/)
Home > TU-OpenAI-Keyvault-D1 | Secrets > Azure-OpenAI-API-Key1 >
8d73d84dfce14078a733f089480af507
Secret Version
Show Secret Value

Properties	
Created	8/28/2023, 2:37:44 PM
Updated	8/28/2023, 2:37:44 PM
Secret Identifier	https://tu-openai-keyvault-d1.vault.azure.net/secrets/Azure-OpenAI-API-Key1/8d73d84dfce14078a733f0...
Settings	
Set activation date	☐
Set expiration date	☐
Enabled	[✓ Yes] No
Tags	0 tags
Secret	
Content type (optional)	
Secret value	••••••••••••••••••••••••••••••••

The key can be seen by clicking on the ‘Show Secret Value’. After copying the key, we can insert it in the <your key here> section in one of the code examples above.

Additional information – access for embeddings models

For the API REST access for embeddings, you will need the same 3 parameters to access the API:

- **Your resource name** – the last part of your resource group name in azure, this should be provided via email together with these instructions
- **Your deployment name** – this is the deployment name found in Azure OpenAI Studio
- **Your API Key** – this is found in the Key Vault resource in your Azure resource group (see access instructions in pages 6 - 7)

These are three examples for accessing Tartu University managed OpenAI API for embeddings

Using CURL:

```
curl --request POST \
  --url 'https://tu-openai-api-management.azure-api.net/<your resource
name>/openai/deployments/<your deployment name>/embeddings?api-version=2023-
07-01-preview' \
  --header 'Content-Type: application/json' \
  --header 'User-Agent: Insomnia/2023.5.6' \
  --header 'api-key: <your key here>' \
  --data '{"input": "Does Azure OpenAI support customer managed keys?"}'
```

Using Python requests package:

```
import requests


url = "https://tu-openai-api-management.azure-api.net/<your resource
name>/openai/deployments/<your deployment name>/embeddings"

querystring = {"api-version":"2023-07-01-preview"}

payload = {"input": "Does Azure OpenAI support customer managed keys?"}
headers = {
    "Content-Type": "application/json",
    "User-Agent": "Insomnia/2023.5.6",
    "api-key": "<your key here>"
}

response = requests.request("POST", url, json=payload, headers=headers,
params=querystring)
```

Using Python openai package:

```
import openai
openai.api_type = "azure"
openai.api_key = "<your key here>"
openai.api_base = "https://tu-openai-api-management.azure-api.net/<your
resource name>"
openai.api_version = "2023-07-01-preview"

# create a chat completion
chat_completion = openai.ChatCompletion.create(
    engine = "<your deployment name>",
    input = "Does Azure OpenAI support customer managed keys?"
)
```