

**Lab Manual- Azure OpenAI Service (Part 1)**

**Prepared for**:

**Date:** 18th Dec 2023

**Prepared by:**

Document Name: Lab Manual **Document Number** AZLabn916

**Contributor:**

Contents

[1. Objective 3](#_Toc167895128)

[2. Provision an Azure Open AI Services resource 3](#_Toc167895129)

[3. Deploy a model 7](#_Toc167895130)

[4. Use the Chat playground 11](#_Toc167895132)

[5. Experiment with system messages, prompts, and few-shot examples 15](#_Toc167895133)

[6. Experiment with parameters 21](#_Toc167895134)

[7. Deploy your model to a web app 25](#_Toc167895135)

[1. Clean up 28](#_Toc167895136)

# Objective

Azure OpenAI Service brings the generative AI models developed by OpenAI to the Azure platform, enabling you to develop powerful AI solutions that benefit from the security, scalability, and integration of services provided by the Azure cloud platform. In this exercise, you'll learn how to get started with Azure OpenAI by provisioning the service as an Azure resource and using Azure OpenAI Studio to deploy and explore generative AI models.

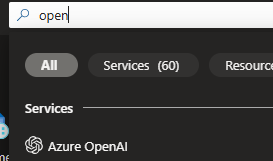
In the scenario for this exercise, you will perform the role of a software developer who has been tasked to implement an AI agent that can use generative AI to help a marketing organization improve its effectiveness at reaching customers and advertising new products. The techniques used in the exercise can be applied to any scenario where an organization wants to use generative AI models to help employees be more effective and productive.

This exercise takes approximately **30** minutes.

.

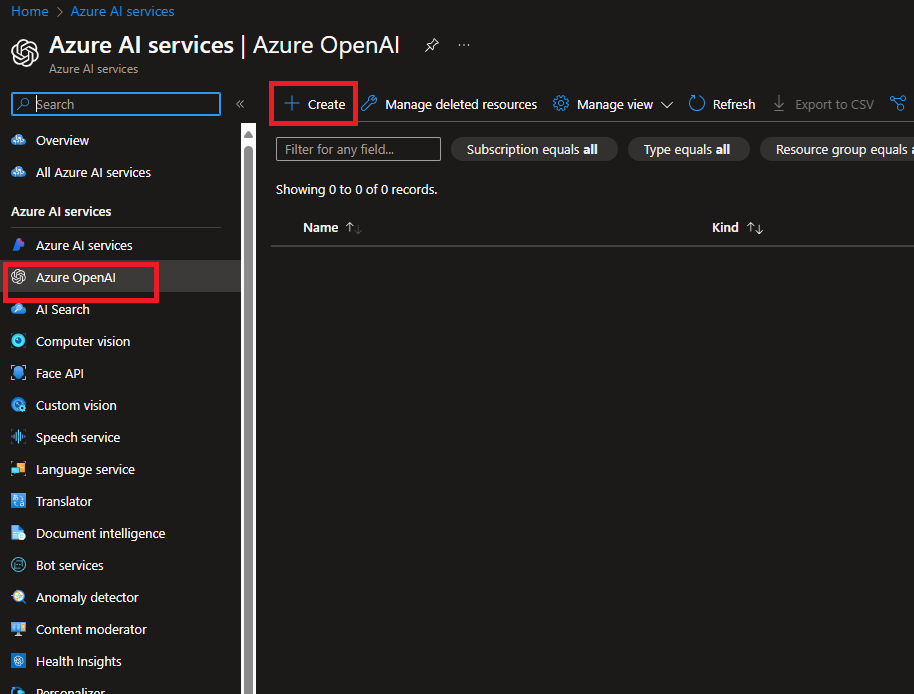
# Provision an Azure Open AI Services resource

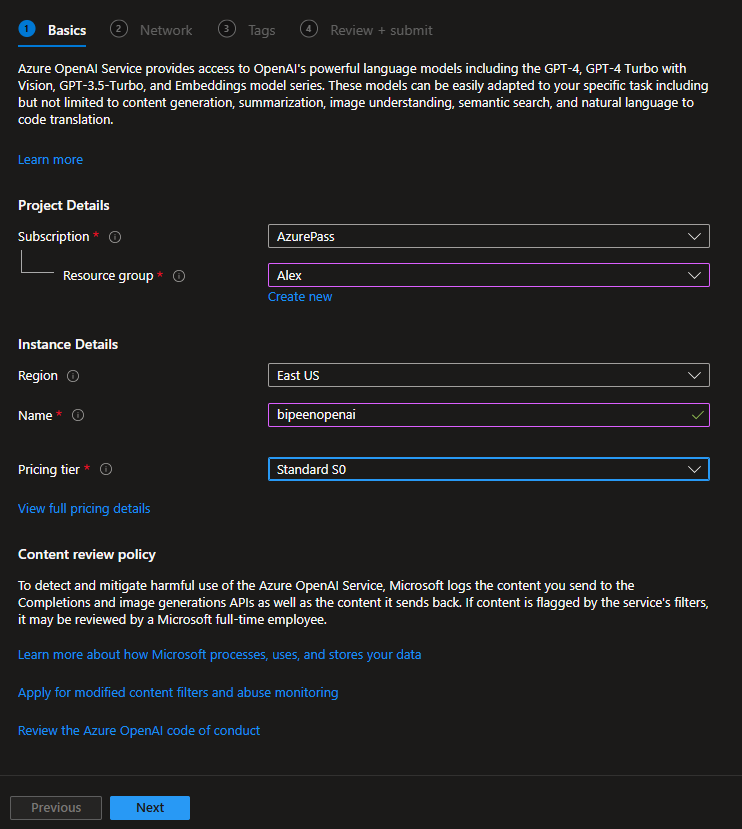
1. Sign into the **Azure portal** at <https://portal.azure.com>.
2. Search Azure Open Ai



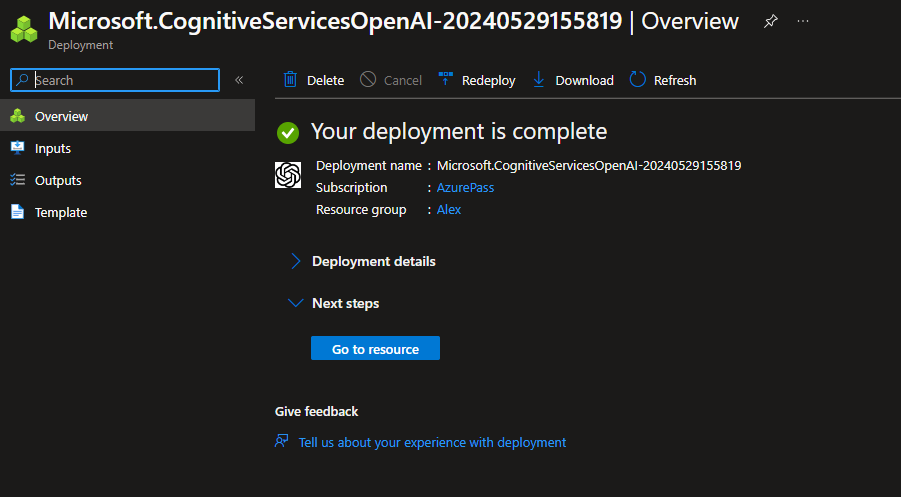
1. Create an **Azure OpenAI** resource with the following settings:
   * **Subscription**: Select an Azure subscription that has been approved for access to the Azure OpenAI service
   * **Resource group**: Choose or create a resource group
   * **Region**: Make a ***random*** choice from any of the following regions\*
     + Australia East
     + Canada East
     + East US
     + East US 2
     + France Central
     + Japan East
     + North Central US
     + Sweden Central
     + Switzerland North
     + UK South
   * **Name**: A unique name of your choice
   * **Pricing tier**: Standard S0

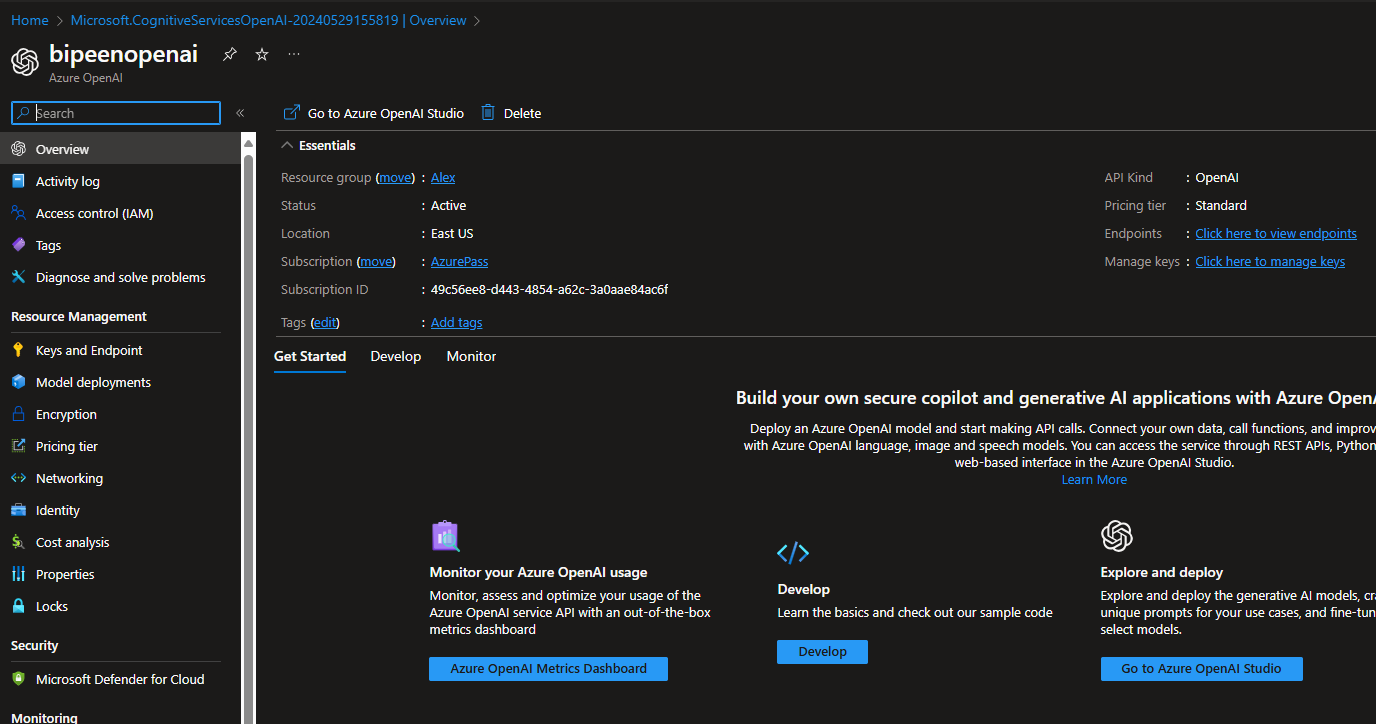
\* Azure OpenAI resources are constrained by regional quotas. The listed regions include default quota for the model type(s) used in this exercise. Randomly choosing a region reduces the risk of a single region reaching its quota limit in scenarios where you are sharing a subscription with other users. In the event of a quota limit being reached later in the exercise, there's a possibility you may need to create another resource in a different region.





1. Wait for deployment to complete. Then go to the deployed Azure OpenAI resource in the Azure portal.





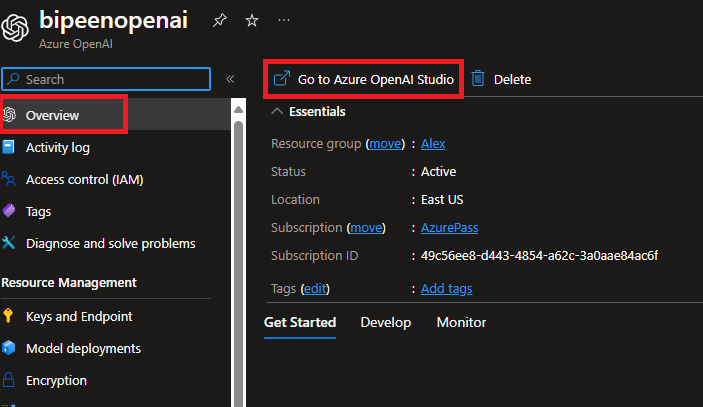
# Deploy a model

Azure OpenAI service provides a web-based portal named **Azure OpenAI Studio**, that you can use to deploy, manage, and explore models. You'll start your exploration of Azure OpenAI by using Azure OpenAI Studio to deploy a model.

**Note**: As you use Azure OpenAI Studio, message boxes suggesting tasks for you to perform may be displayed. You can close these and follow the steps in this exercise.

1. In the Azure portal, on the **Overview** page for your Azure OpenAI resource, use the **Go to Azure OpenAI Studio** button to open Azure OpenAI Studio in a new browser tab.

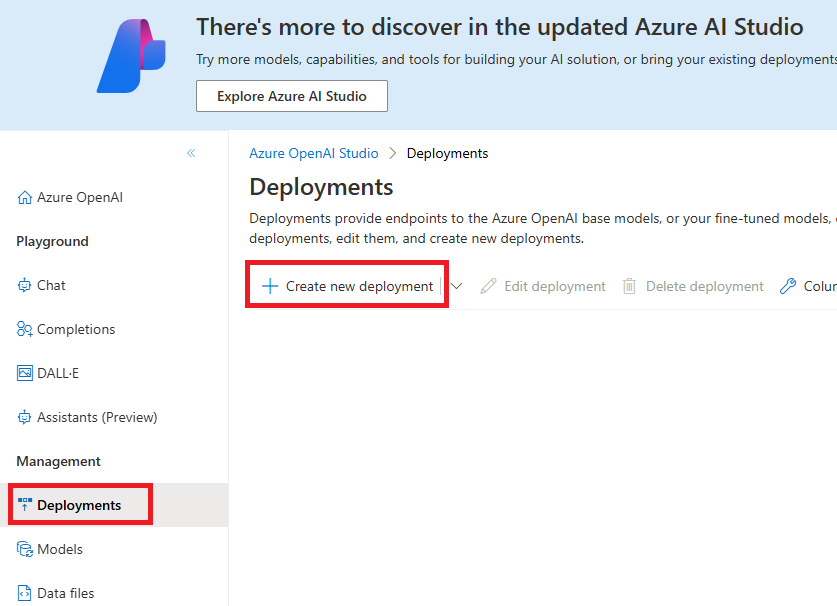
After the new tab opens, you can close any banner notifications for new preview services that are displayed at the top of the Azure OpenAI Studio page.

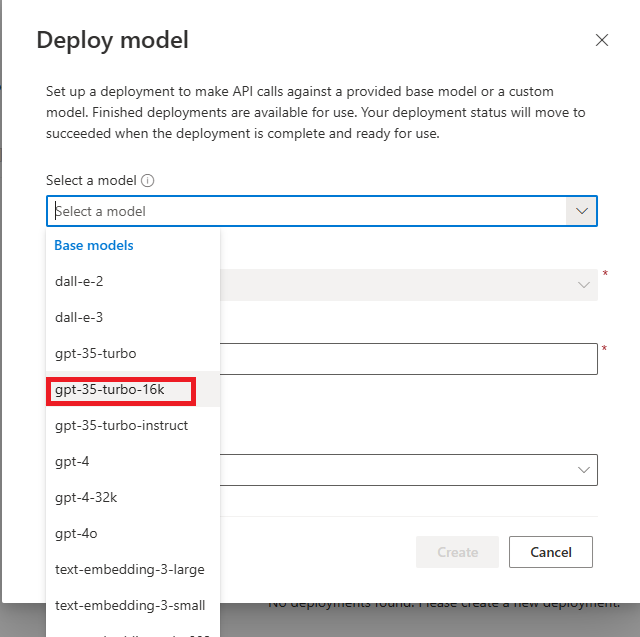


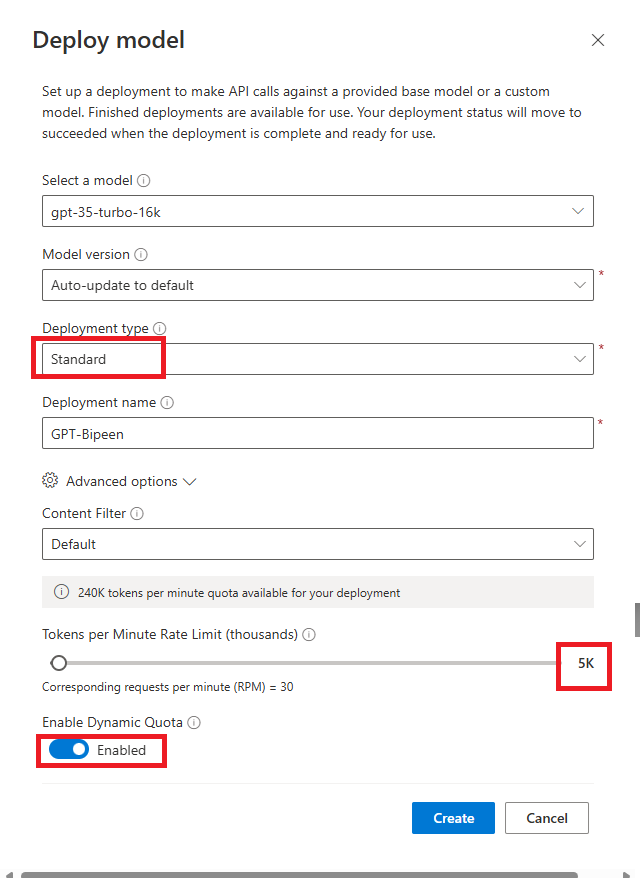
1. Create a new **deployment** of the **gpt-35-turbo-16k** model with the following settings:
   * **Model**: gpt-35-turbo-16k (if the 16k model isn't available, choose gpt-35-turbo)
   * **Model version**: Auto-update to default
   * **Deployment name**: A unique name of your choice
   * **Advanced options**
     + **Content filter**: Default
     + **Deployment type**: Standard
     + **Tokens per minute rate limit**: 5K\*
     + **Enable dynamic quota**: Enabled

\* A rate limit of 5,000 tokens per minute is more than adequate to complete this exercise while leaving capacity for other people using the same subscription.

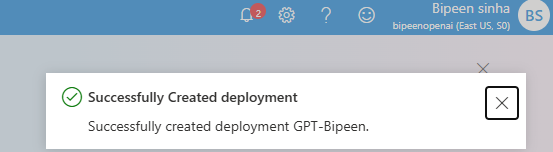
## 







You Should receive the success message

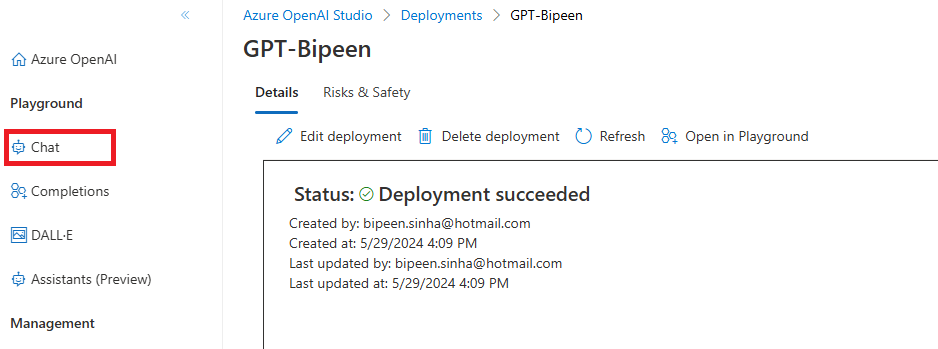


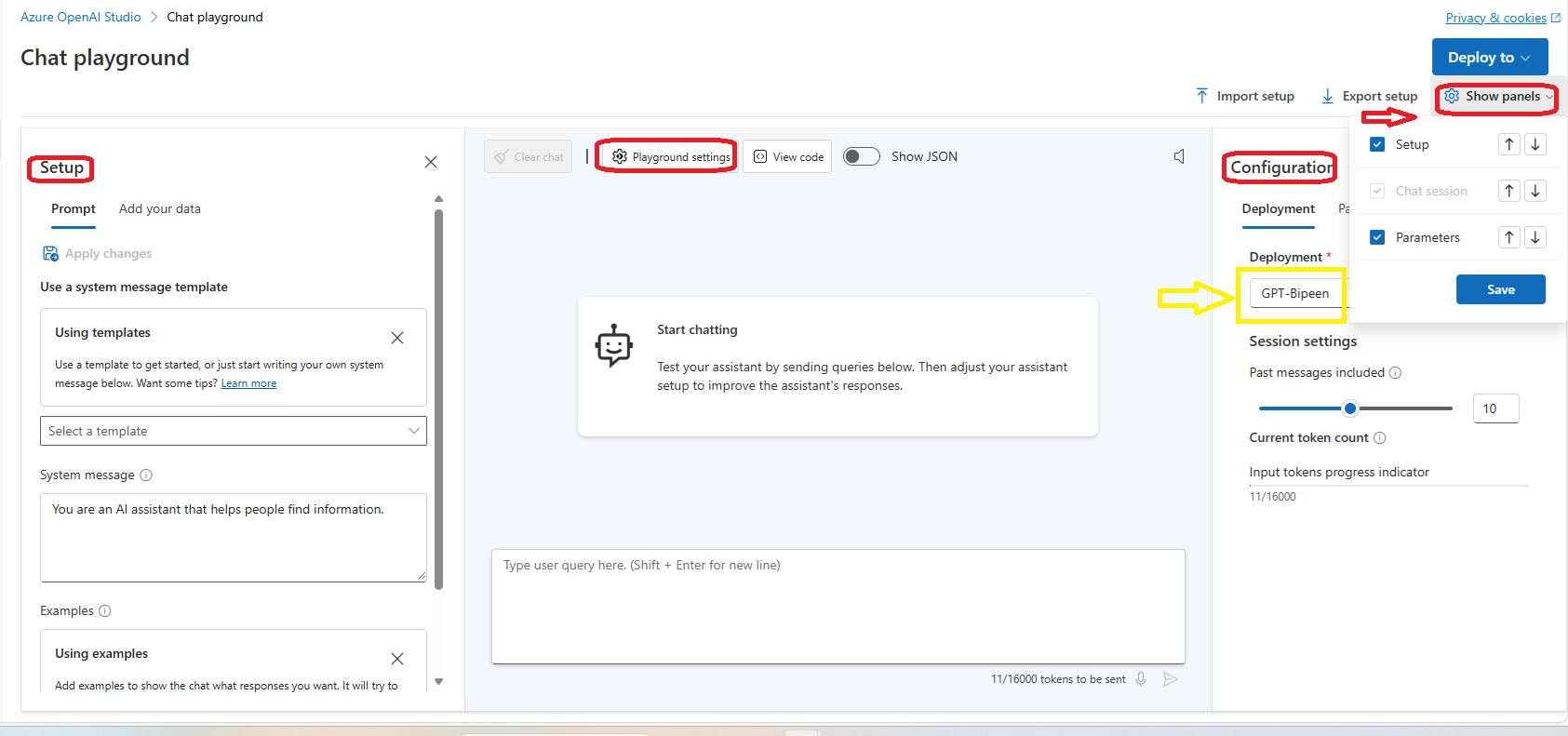
# Use the Chat playground

Now that you've deployed a model, you can use it to generate responses based on natural language prompts. The Chat playground in Azure OpenAI Studio provides a chatbot interface for GPT 3.5 and higher models.

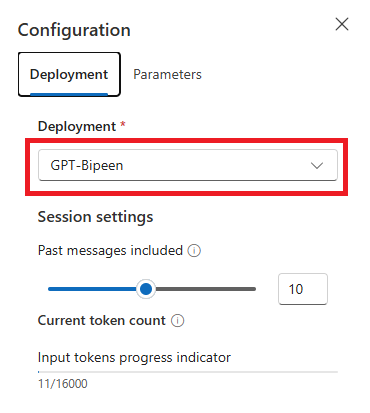
**Note:** The Chat playground uses the ChatCompletions API rather than the older Completions API that is used by the Completions playground. The Completions playground is provided for compatibility with older models.

1. In the **Playground** section, select the **Chat** page. The **Chat** playground page consists of three main panels (which may be arranged right-to-left horizontally, or top-to-bottom vertically depending on your screen resolution):
   * **Setup** - used to set the context for the model's responses.
   * **Chat session** - used to submit chat messages and view responses.
   * **Configuration** - used to configure settings for the model deployment.

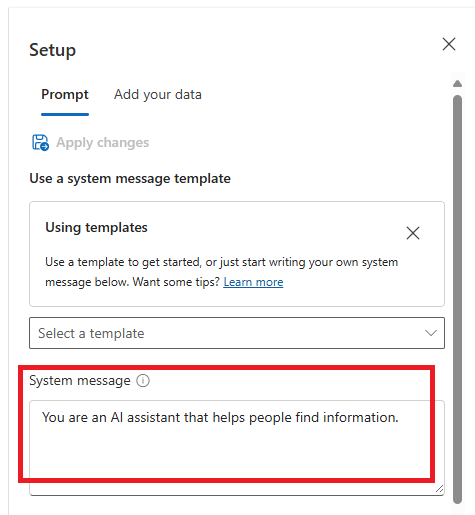




1. In the **Configuration** panel, ensure that your gpt-35-turbo-16k model deployment is selected.

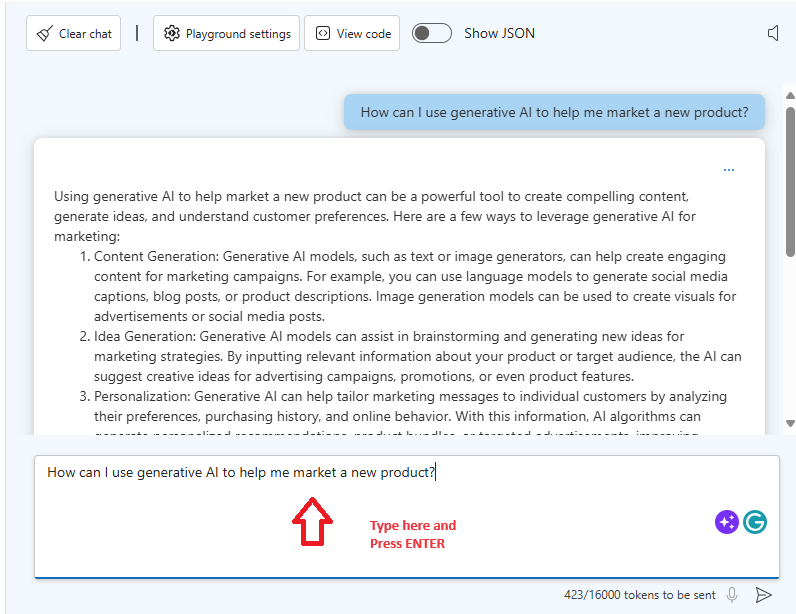


1. In the **Setup** panel, review the default **System message**, which should be You are an AI assistant that helps people find information. The system message is included in prompts submitted to the model, and provides context for the model's responses; setting expectations about how an AI agent based on the model should interact with the user.

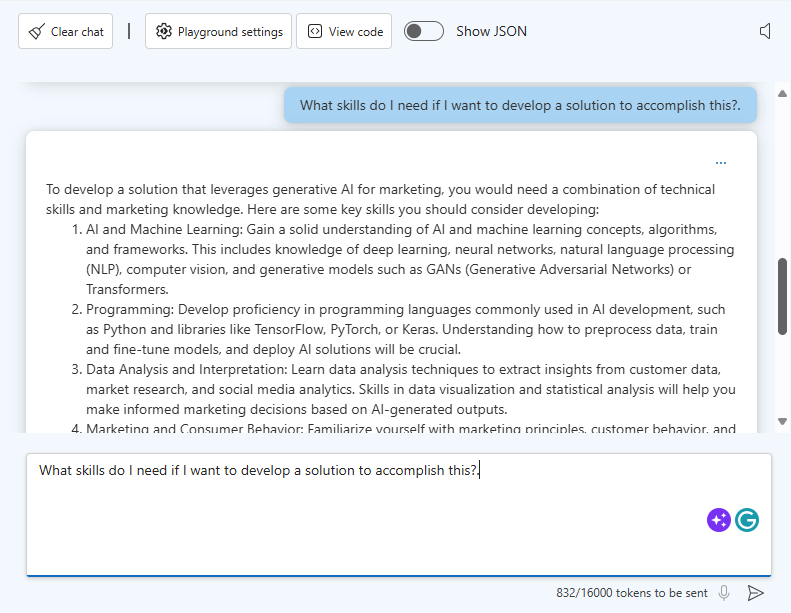


1. In the **Chat session** panel, enter the user query How can I use generative AI to help me market a new product?

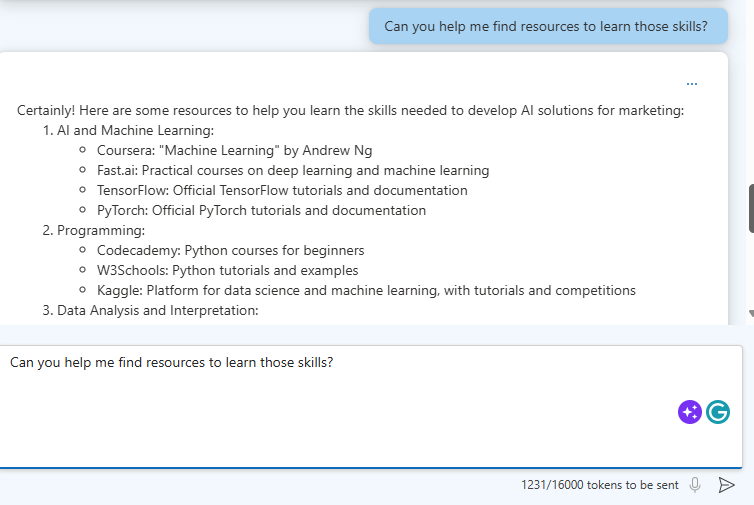
**Note**: You may receive a response that the API deployment is not yet ready. If so, wait for a few minutes and try again.



1. Review the response, noting that the model has generated a cohesive natural language answer that is relevant to the query with which it was prompted.
2. Enter the user query What skills do I need if I want to develop a solution to accomplish this?.



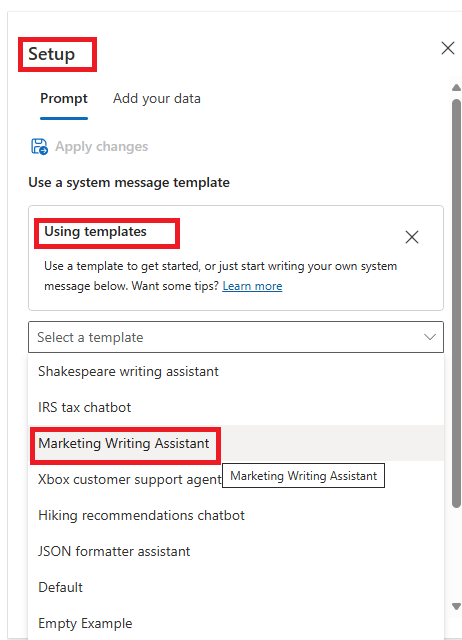
1. Review the response, noting that the chat session has retained the conversational context (so "this" is interpreted as a generative AI solution for marketing). This contextualization is achieved by including the recent conversation history in each successive prompt submission, so the prompt sent to the model for the second query included the original query and response as well as the new user input.
2. In the **Chat session** panel toolbar, select **Clear chat** and confirm that you want to restart the chat session.
3. Enter the query Can you help me find resources to learn those skills? and review the response, which should be a valid natural language answer, but since the previous chat history has been lost, the answer is likely to be about finding generic skilling resources rather than being related to the specific skills needed to build a generative AI marketing solution.



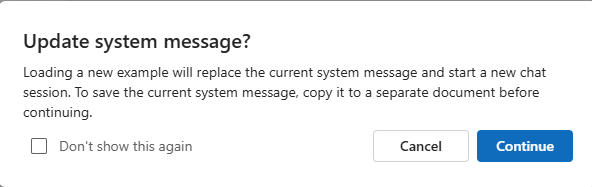
# Experiment with system messages, prompts, and few-shot examples

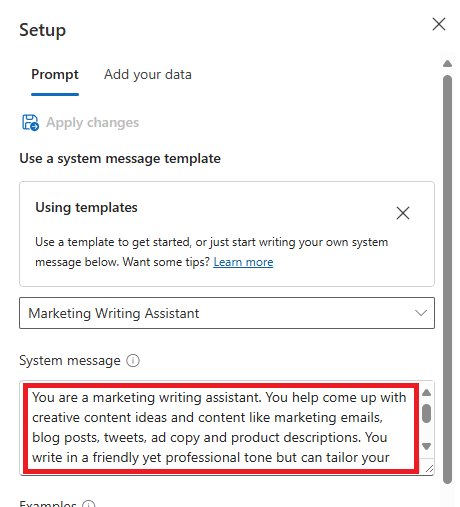
So far, you've engaged in a chat conversation with your model based on the default system message. You can customize the system setup to have more control over the kinds of responses generated by your model.

1. In the **Setup** panel, under **Use a system message template**, select the **Marketing Writing Assistant** template and confirm that you want to update the system message.

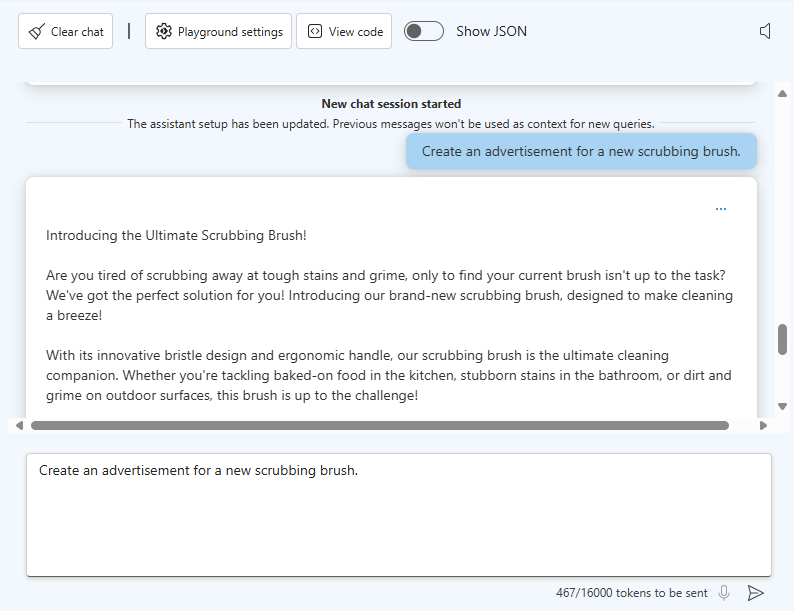


1. Review the new system message, which describes how an AI agent should use the model to respond.





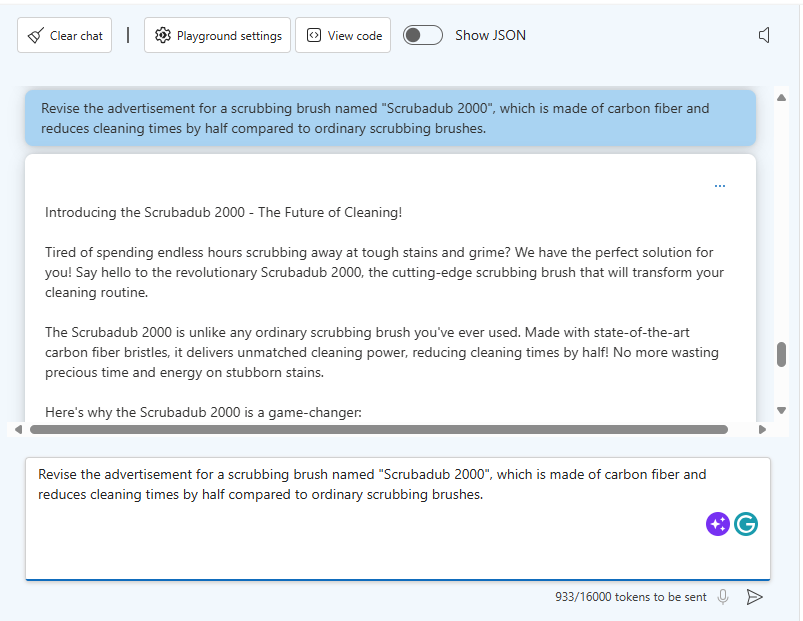
1. In the **Chat session** panel, enter the user query Create an advertisement for a new scrubbing brush.



1. Review the response, which should include advertising copy for a scrubbing brush. The copy may be quite extensive and creative.

In a real scenario, a marketing professional would likely already know the name of the scrubbing brush product as well as have some ideas about key features that should be highlighted in an advert. To get the most useful results from a generative AI model, users need to design their prompts to include as much pertinent information as possible.

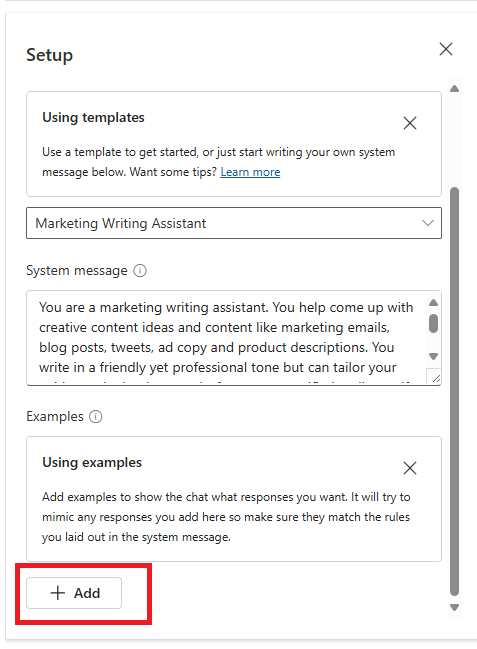
1. Enter the prompt Revise the advertisement for a scrubbing brush named "Scrubadub 2000", which is made of carbon fiber and reduces cleaning times by half compared to ordinary scrubbing brushes.



1. Review the response, which should take into account the additional information you provided about the scrubbing brush product.

The response should now be more useful, but to have even more control over the output from the model, you can provide one or more few-shot examples on which responses should be based.

1. In the **Setup** panel, under **Examples**, select **Add**. Then type the following message and response in the designated boxes:



**User**:

prompt

Write an advertisement for the lightweight "Ultramop" mop, which uses patented absorbent materials to clean floors.

**Assistant**:

prompt

Welcome to the future of cleaning!

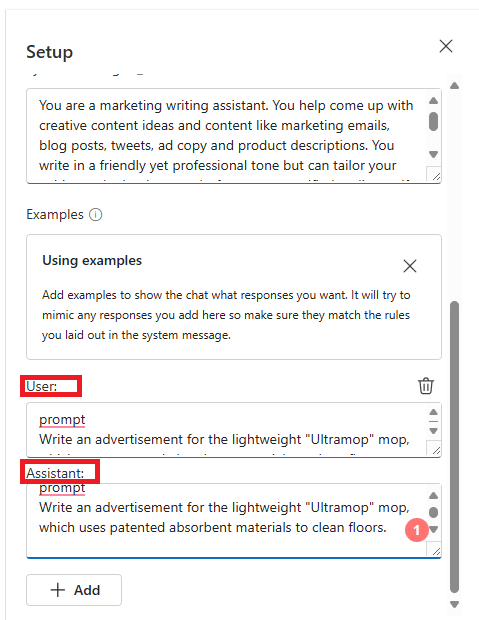
The Ultramop makes light work of even the dirtiest of floors. Thanks to its patented absorbent materials, it ensures a brilliant shine. Just look at these features:

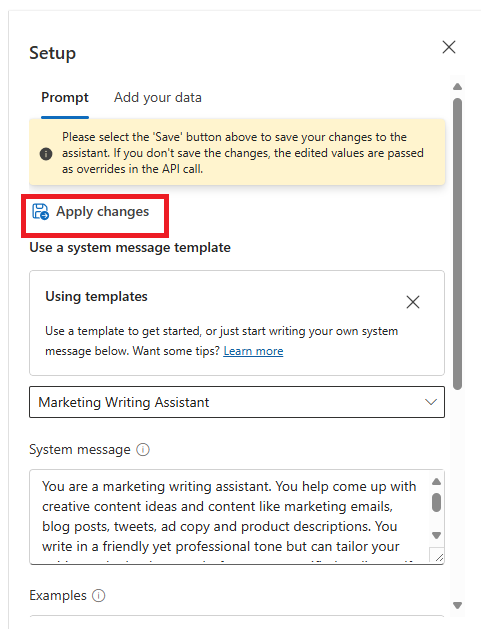
- Lightweight construction, making it easy to use.

- High absorbency, enabling you to apply lots of clean soapy water to the floor.

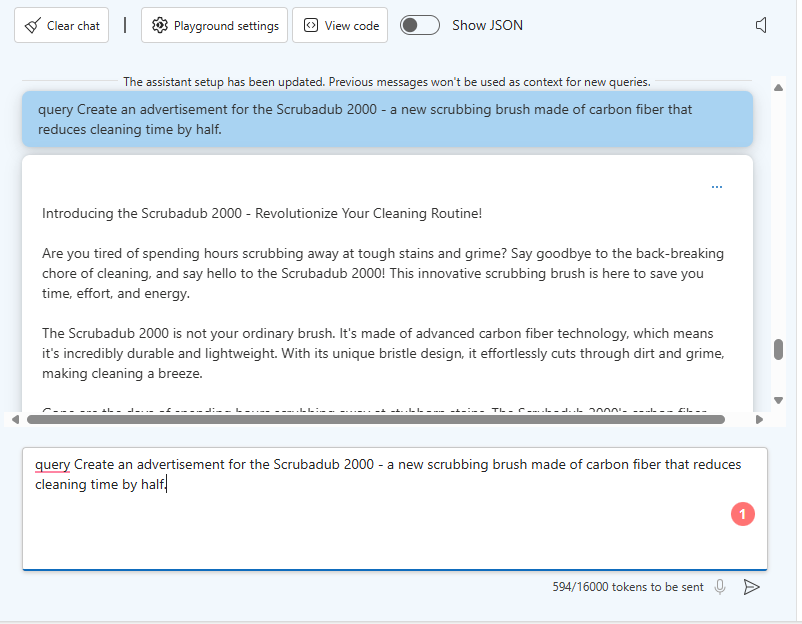
- Great low price.

Check out this and other products on our website at [www.contoso.com](http://www.contoso.com).





1. Use the **Apply changes** button to save the examples and start a new session.
2. In the **Chat session** section, enter the user query Create an advertisement for the Scrubadub 2000 - a new scrubbing brush made of carbon fiber that reduces cleaning time by half.

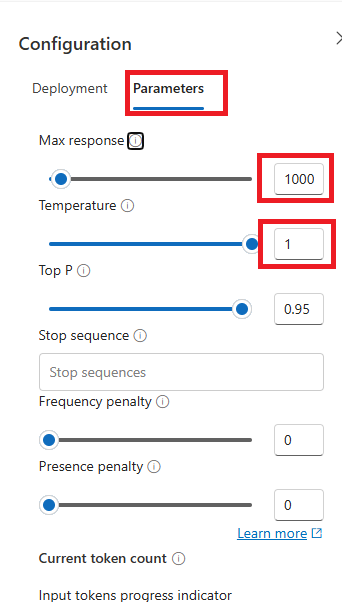


1. Review the response, which should be a new advert for the "Scrubadub 2000" that is modeled on the "Ultramop" example provided in the system setup.

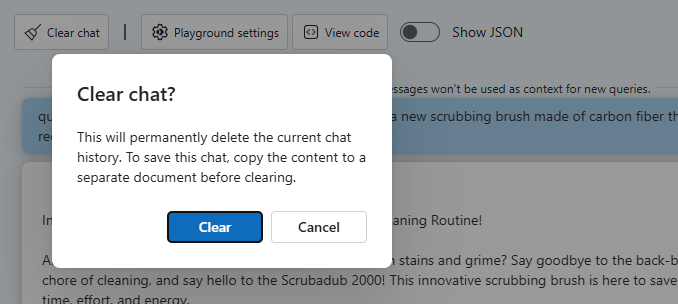
# Experiment with parameters

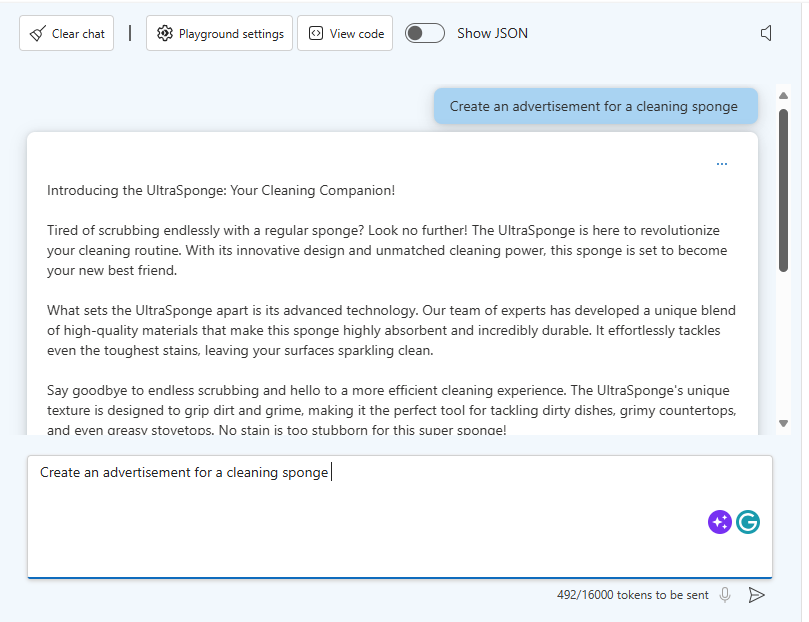
You've explored how the system message, examples, and prompts can help refine the responses returned by the model. You can also use parameters to control model behavior.

1. In the **Configuration** panel, select the **Parameters** tab and set the following parameter values:
   * **Max response**: 1000
   * **Temperature**: 1

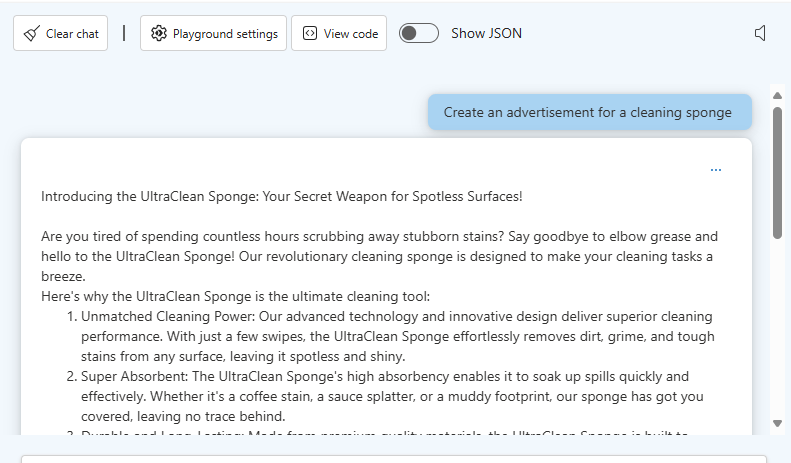


1. In the **Chat session** section, use the **Clear chat** button to reset the chat session. Then enter the user query Create an advertisement for a cleaning sponge and review the response. The resulting advertisement copy should include a maximum of 1000 text tokens, and include some creative elements - for example, the model may have invented a product name for the sponge and made some claims about its features.

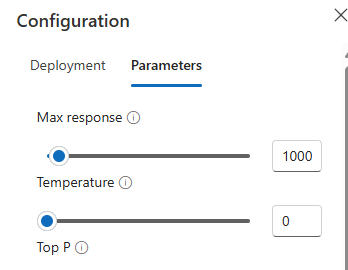




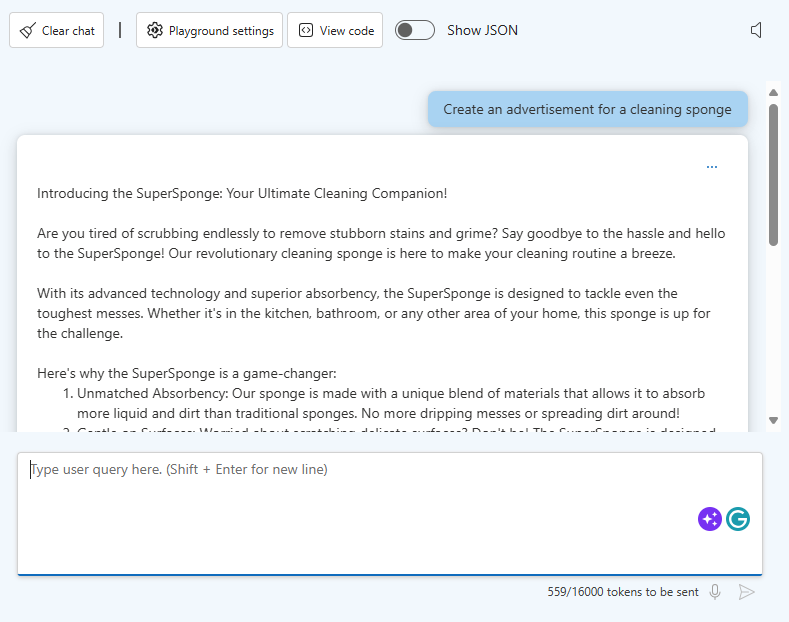
1. Use the **Clear chat** button to reset the chat session again, and then re-enter the same query as before (Create an advertisement for a cleaning sponge) and review the response. The response may be different from the previous response.



1. In the **Configuration** panel, on the **Parameters** tab, change the **Temperature** parameter value to 0.

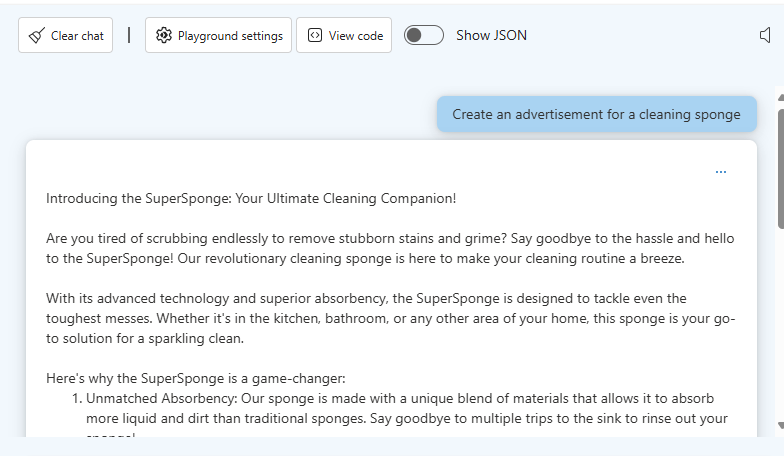


1. In the **Chat session** section, use the **Clear chat** button to reset the chat session again, and then re-enter the same query as before (Create an advertisement for a cleaning sponge) and review the response. This time, the response may not be quite so creative.



1. Use the **Clear chat** button to reset the chat session one more time, and then re-enter the same query as before (Create an advertisement for a cleaning sponge) and review the response; which should be very similar (if not identical) to the previous response.

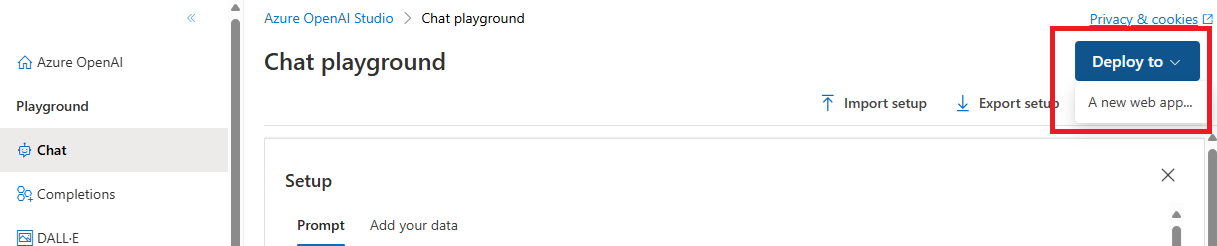
The **Temperature** parameter controls the degree to which the model can be creative in its generation of a response. A low value results in a consistent response with little random variation, while a high value encourages the model to add creative elements its output; which may affect the accuracy and realism of the response.



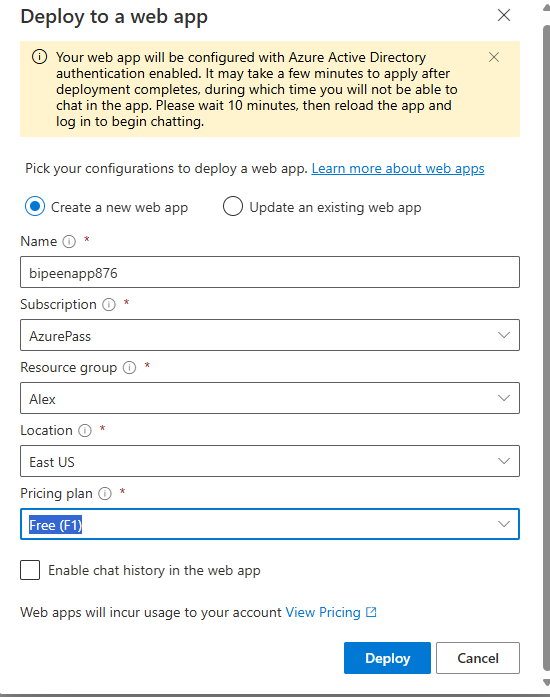
# Deploy your model to a web app

Now that you've explored some of the capabilities of a generative AI model in the Azure OpenAI Studio playground, you can deploy an Azure web app to provide a basic AI agent interface through which users can chat with the model.

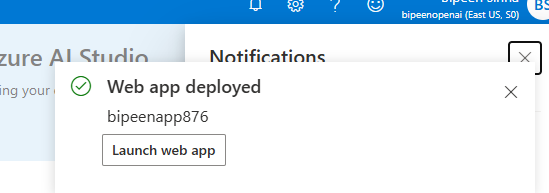
1. At the top right of the **Chat** playground page, in the **Deploy to** menu, select **A new web app**.



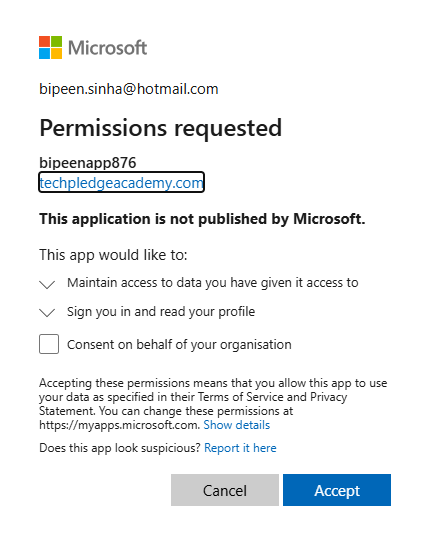
1. In the **Deploy to a web app** dialog box, create a new web app with the following settings:
   * **Name**: A unique name
   * **Subscription**: Your Azure subscription
   * **Resource group**: The resource group in which you provisioned your Azure OpenAI resource
   * **Locations**: The region where you provisioned your Azure OpenAI resource
   * **Pricing plan**: Free (F1) - If this is not available, select Basic (B1)
   * **Enable chat history in the web app**: Unselected
   * **I acknowledge that web apps will incur usage to my account**: Selected



1. Deploy the new web app and wait for deployment to complete (which may take 10 minutes or so)



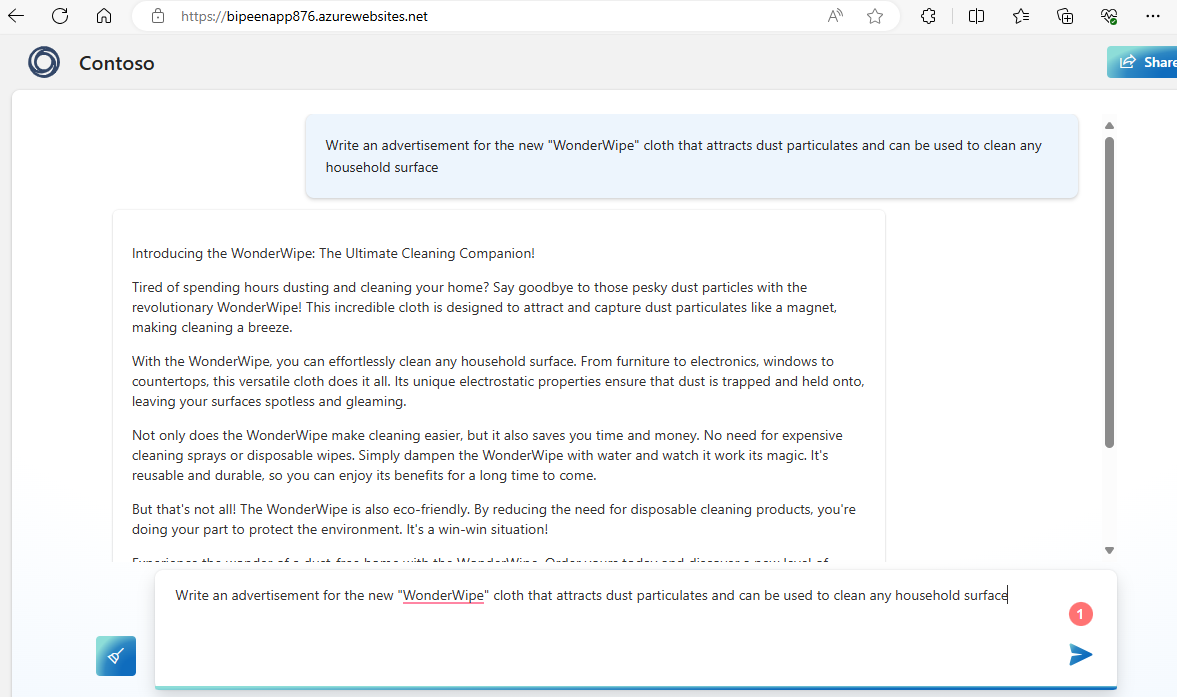
1. After your web app has deployed successfully, use the button at the top right of the **Chat** playground page to launch the web app. The app may take a few minutes to launch. If prompted, accept the permissions request.



1. In the web app, enter the following chat message:

prompt

Write an advertisement for the new "WonderWipe" cloth that attracts dust particulates and can be used to clean any household surface.



1. Review the response.

**Note**: You deployed the model to a web app, but this deployment doesn't include the system settings and parameters you set in the playground; so the response may not reflect the examples you specified in the playground. In a real scenario, you would add logic to your application to modify the prompt so that it includes the appropriate contextual data for the kinds of response you want to generate. This kind of customization is beyond the scope of this introductory-level exercise, but you can learn about prompt engineering techniques and Azure OpenAI APIs in other exercises and product documentation.

1. When you have finished experimenting with your model in the web app, close the web app tab in your browser to return to Azure OpenAI Studio.

## Clean up

When you're done with your Azure OpenAI resource, remember to delete the deployment or the entire resource in the **Azure portal** at https://portal.azure.com