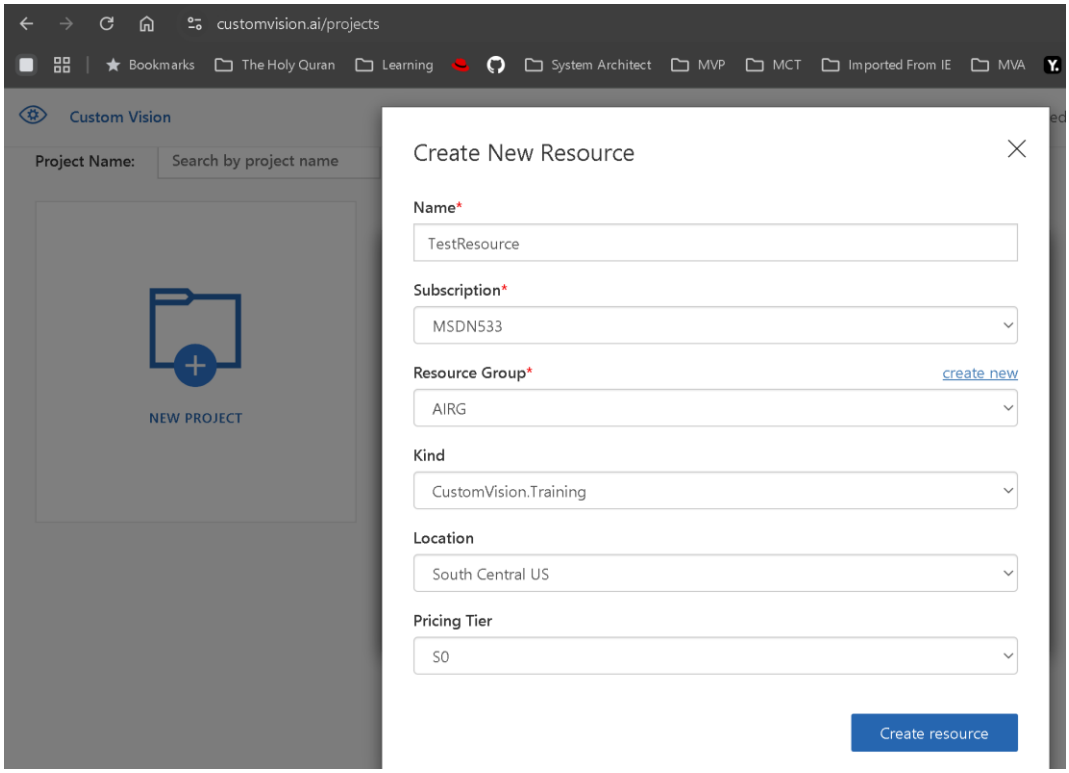
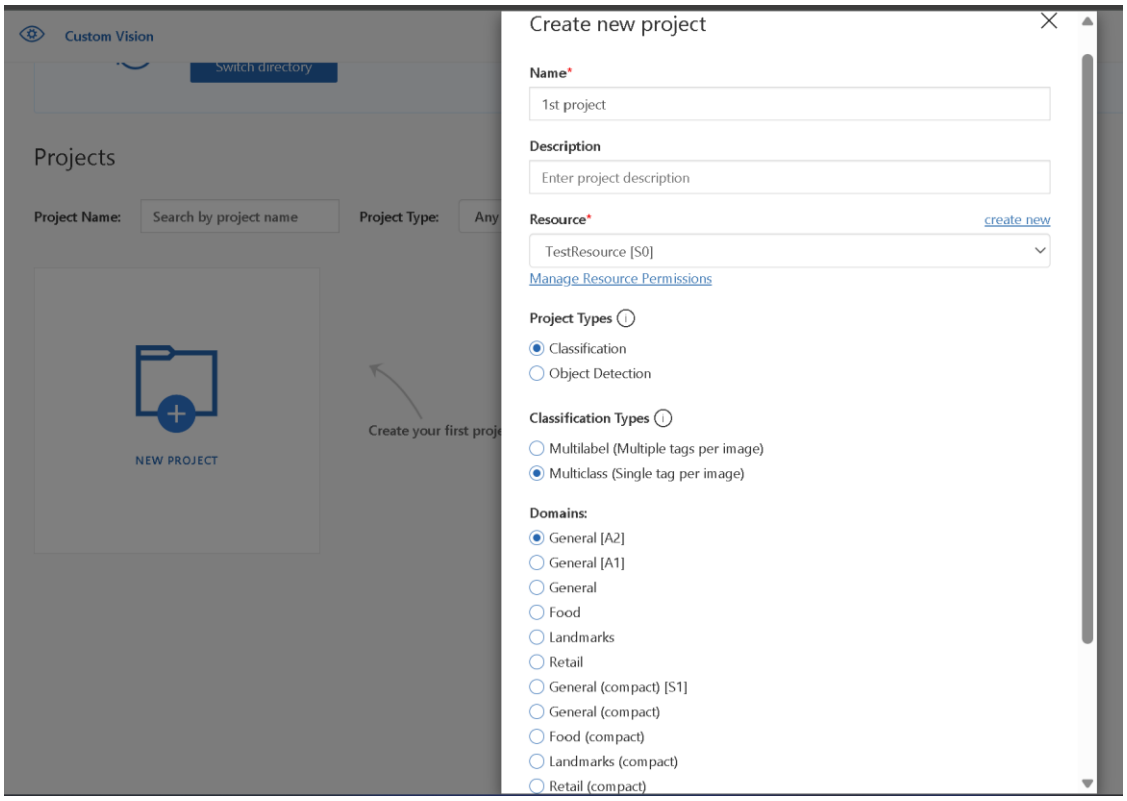
Azure AI | Hands-On

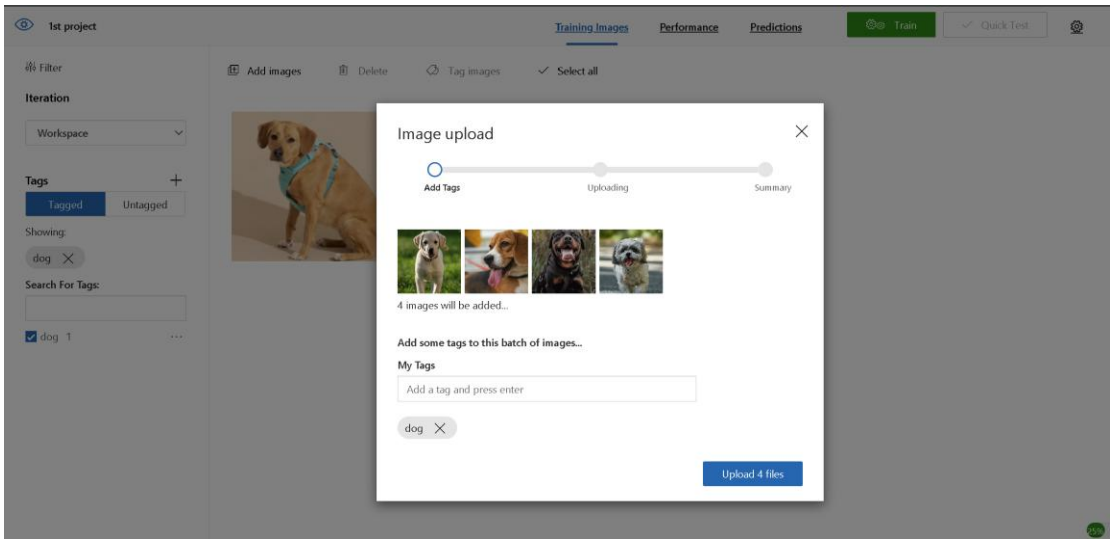
Version 25.04

Ahmed Abdelwahed
ahmed@abdelwahed.me
www.abdelwahed.me
LinkedIn
GitHub

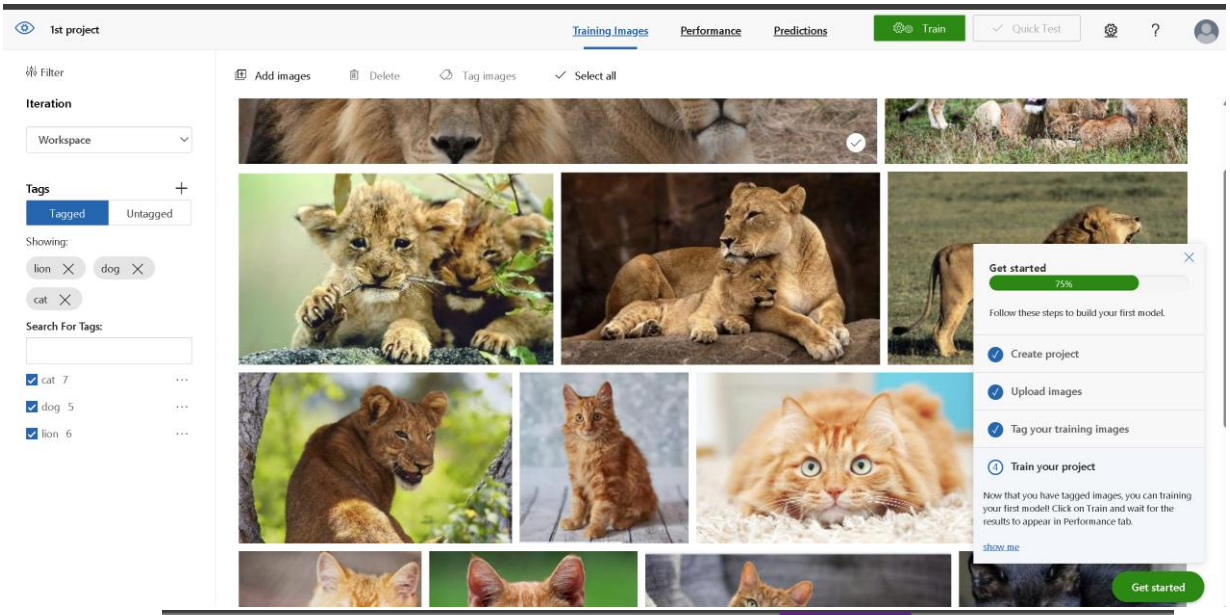
Image Classification with Custom Vision

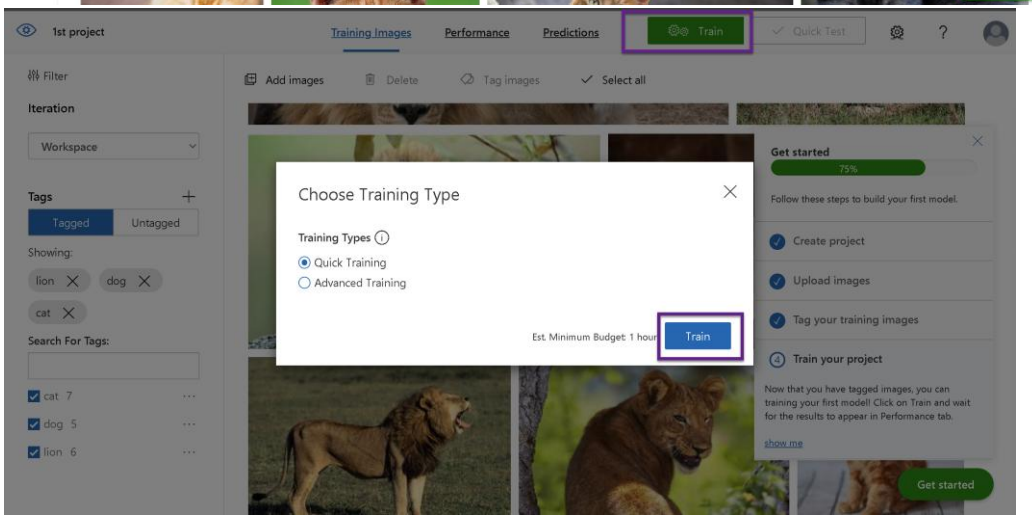


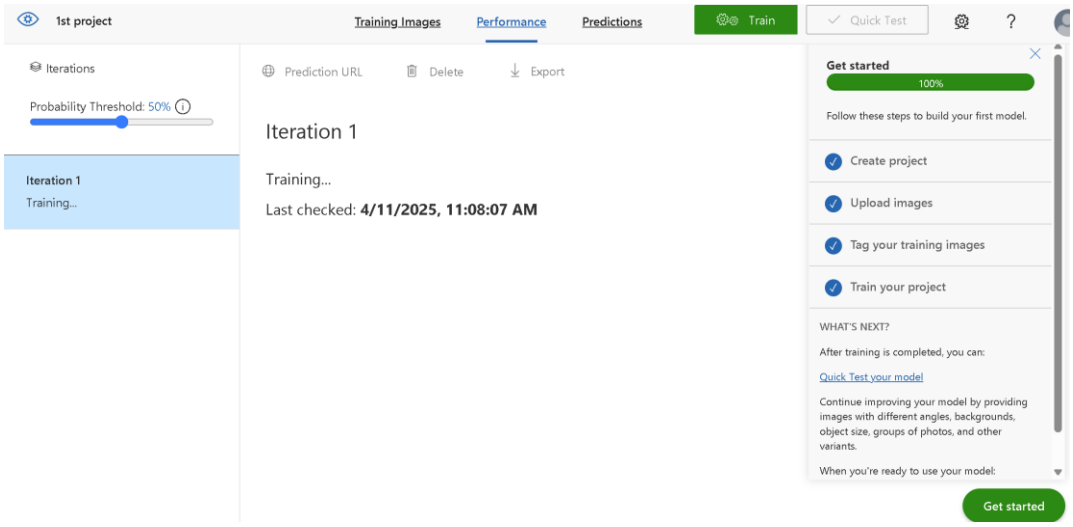


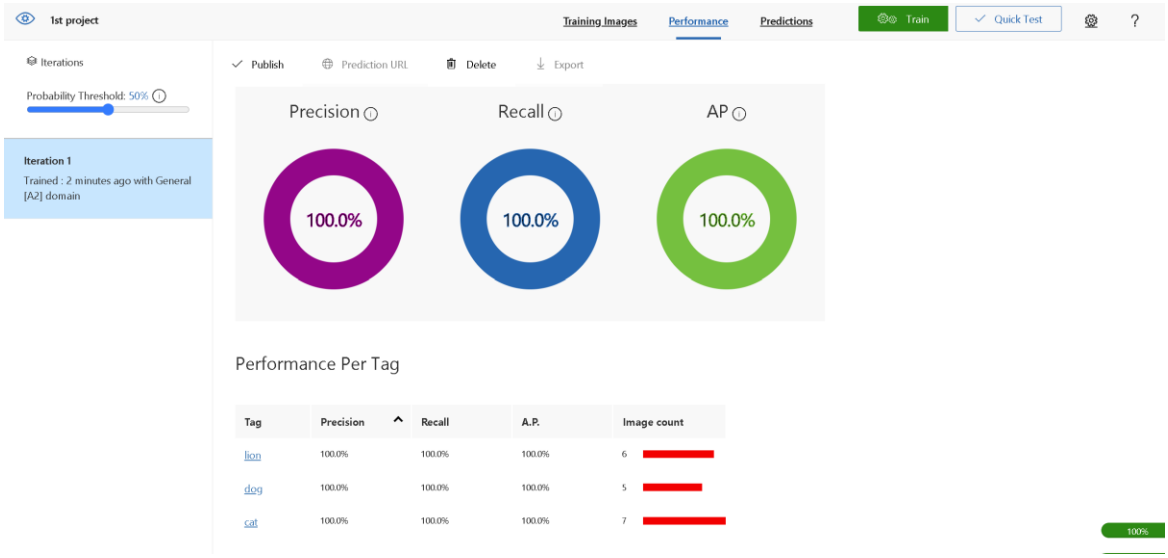


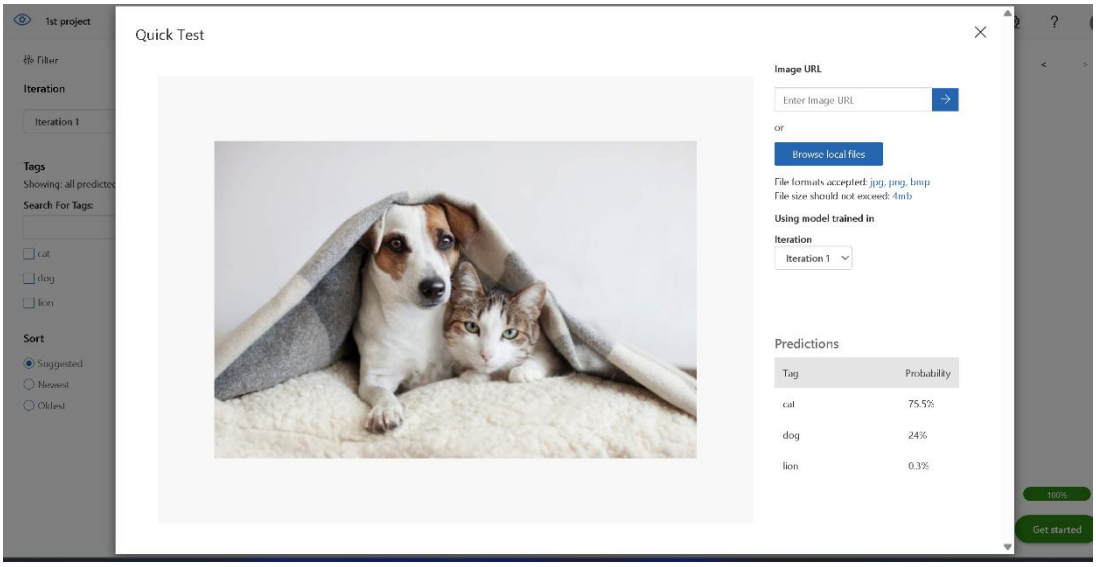
The more diverse and clean your data is, the better the model performs.



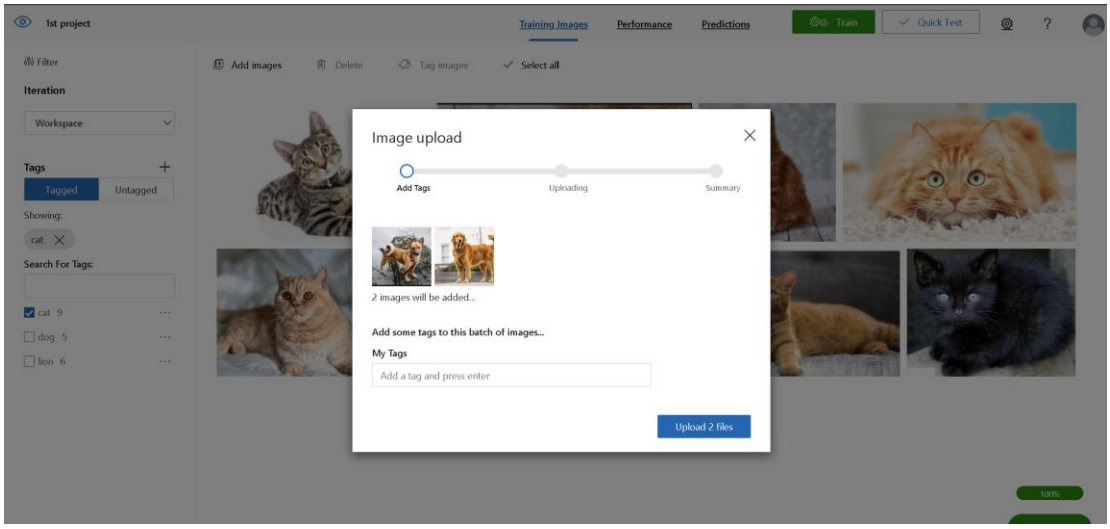



1st project
Training Images
Performance
Predictions
Train
Quick Test
Iterations
Probability Threshold: 50%
Iteration 1
Training...
Prediction URL
Delete
Export
Iteration 1
Training...
Last checked: 4/11/2025, 11:08:07 AM
Get started
100%
Follow these steps to build your first model.
Create project
Upload images
Tag your training images
Train your project
WHAT'S NEXT?
After training is completed, you can:
Quick Test your model
Continue improving your model by providing images with different angles, backgrounds, object size, groups of photos, and other variants.
When you're ready to use your model:
Get started

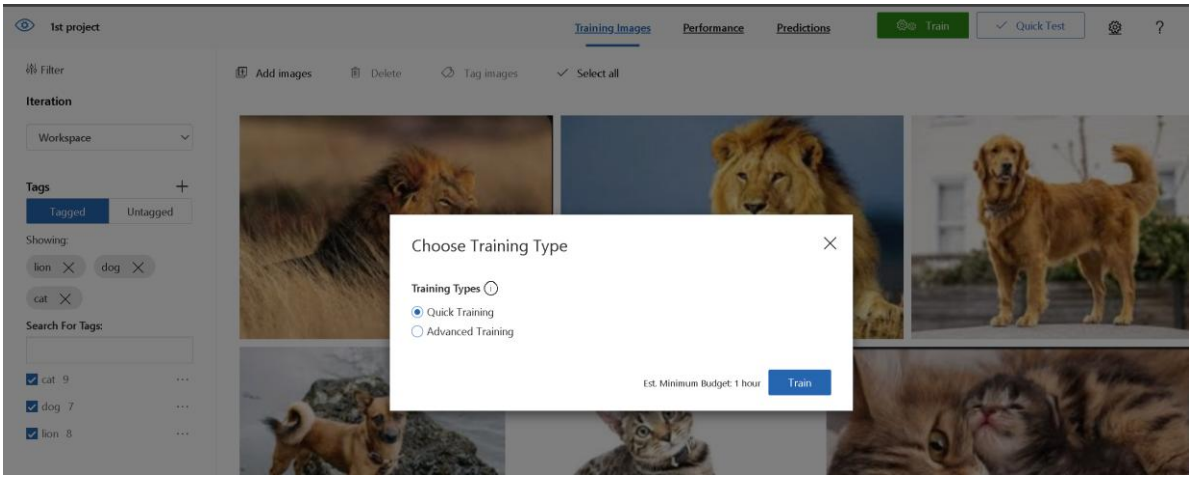

1st project
Training Images
Performance
Predictions
Train
Quick Test
Iterations
Probability Threshold: 50%
Iteration 1
Trained : 2 minutes ago with General [A2] domain
Publish
Prediction URL
Delete
Export
Precision
Recall
AP
100.0%
100.0%
100.0%
Performance Per Tag
Tag
Precision
Recall
A.P.
Image count
lion
100.0%
100.0%
100.0%
6
dog
100.0%
100.0%
100.0%
5
cat
100.0%
100.0%
100.0%
7
100%


1st project
Filter
Iteration
Iteration 1
Tags
Showing: all predicted
Search For Tags:
cat
dog
lion
Sort
Suggested
Newest
Oldest
Quick Test
Image URL
Enter Image URL
or
Browse local files
File formats accepted: jpg, png, bmp
File size should not exceed: 4mb
Using model trained in
Iteration
Iteration 1
Predictions
Tag
Probability
cat
75.5%
dog
24%
lion
0.3%
100%
Get started

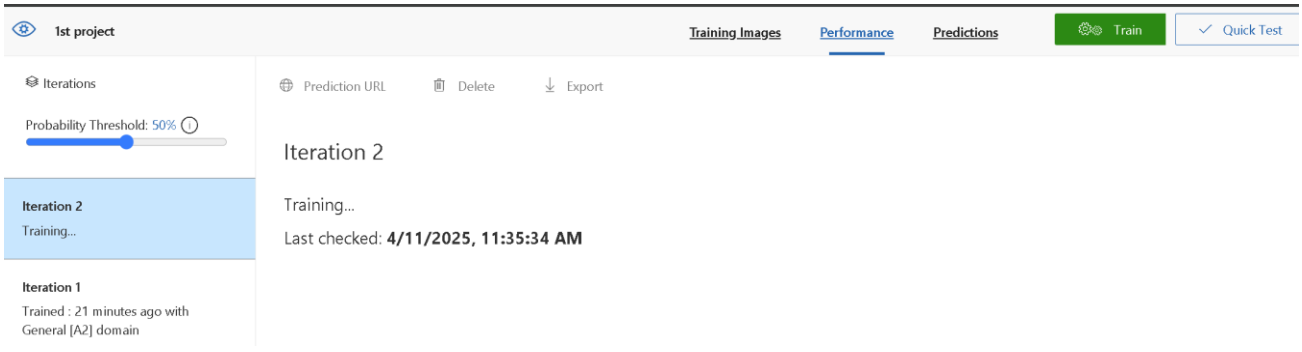
Add more images



And retrain your model



Creating new iteration



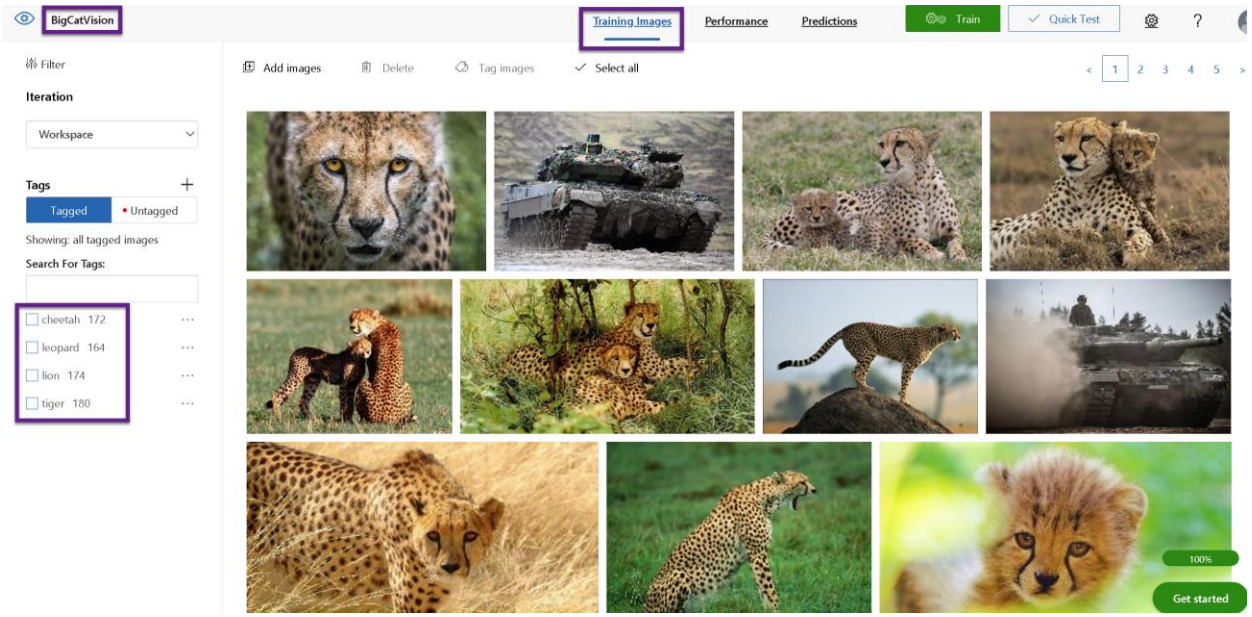
iteration is a **version of your trained model**.

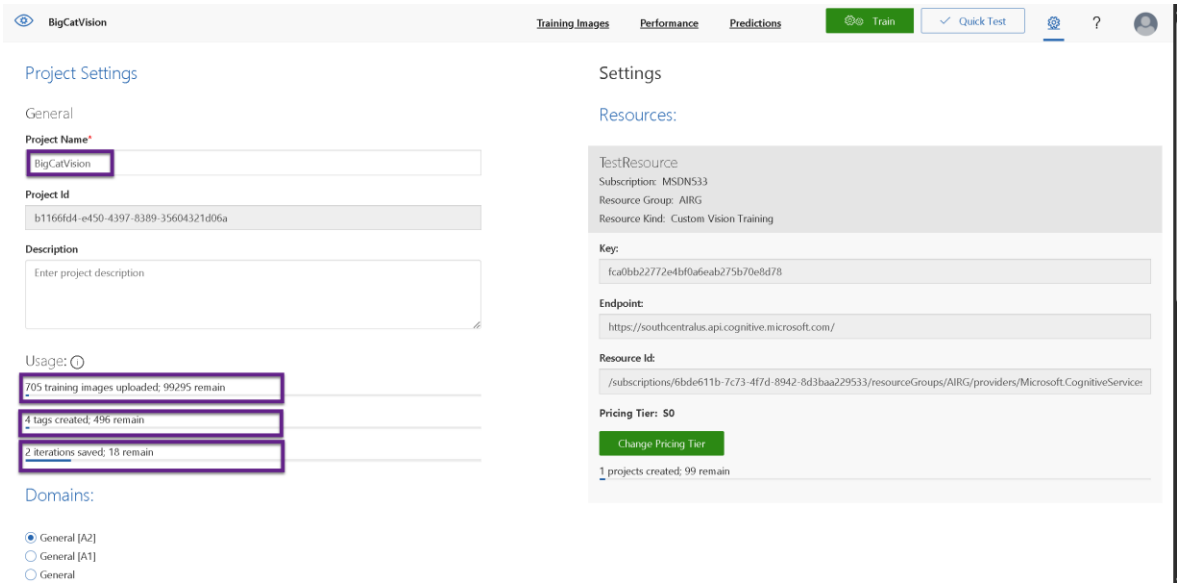
Classify Big Cats

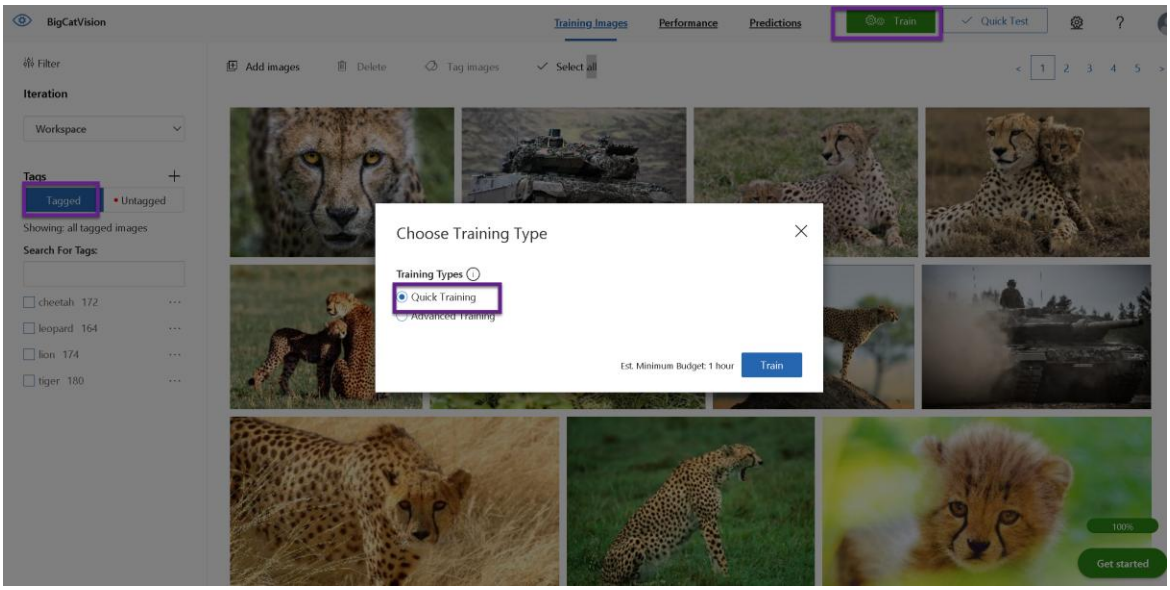
Create new project and download big cat dataset from Kaggle to use it in the classifications

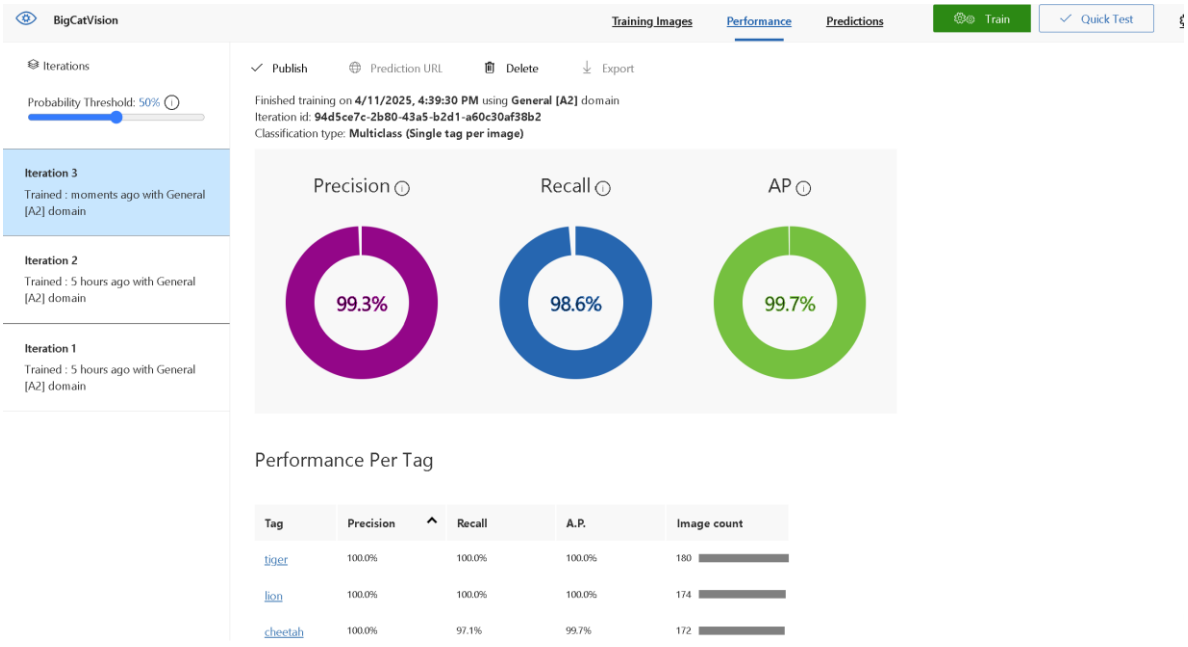
Image Datasets for Training:

1. **Kaggle Datasets**
2. Google Dataset Search
3. Open Images Dataset by Google
4. ImageNet

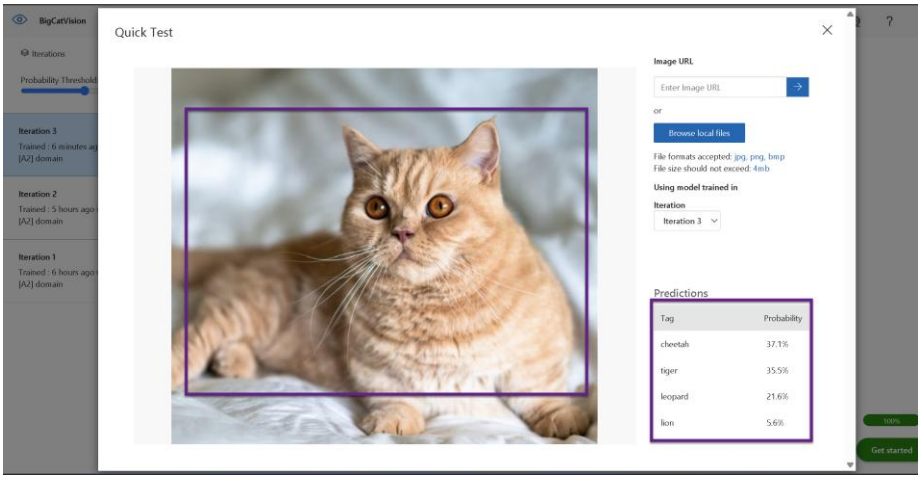


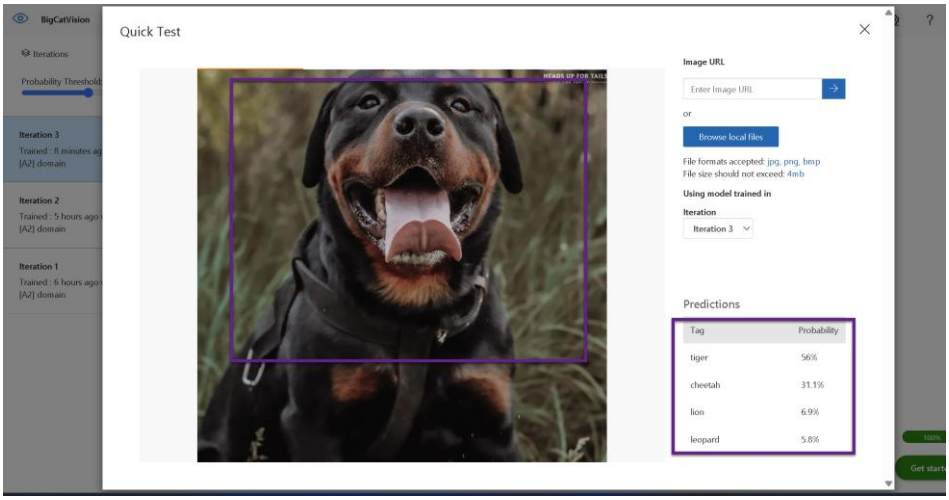


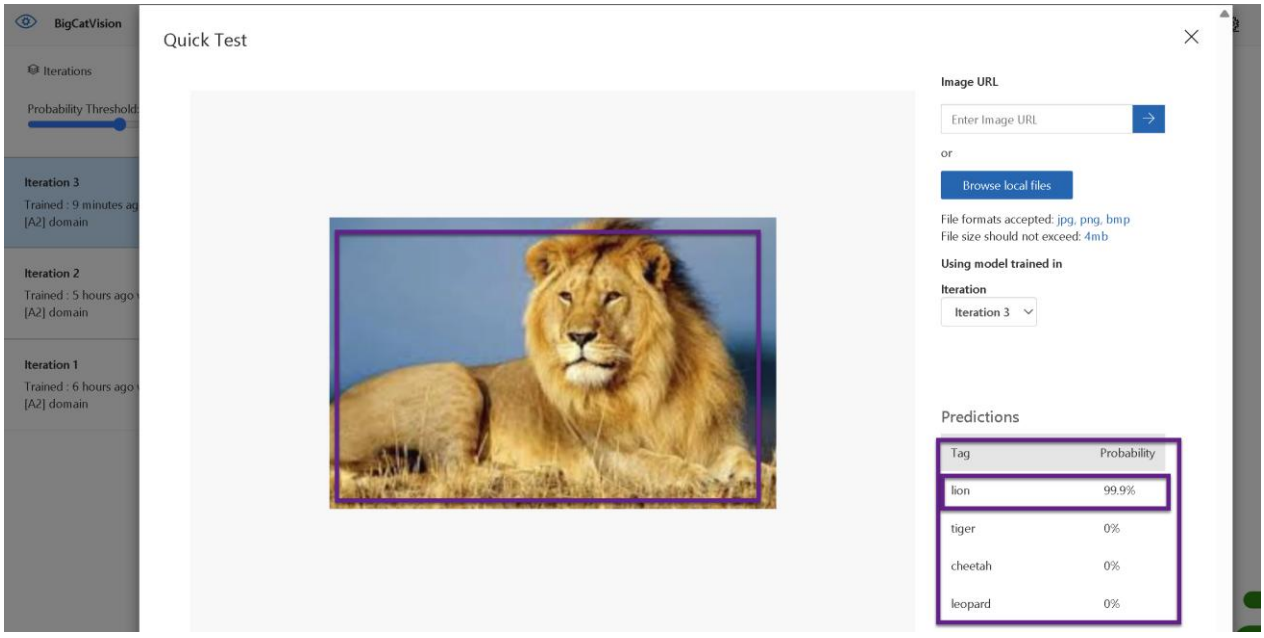

BigCatVision
Training Images
Performance
Predictions
Train
Quick Test
Filter
Iteration
Workspace
Tags
Tagged
Untagged
Showing: all tagged images
Search For Tags:
cheetah 172
leopard 164
lion 174
tiger 180
Add images
Delete
Tag images
Select all
Choose Training Type
Training Types
Quick Training
Advanced Training
Est. Minimum Budget: 1 hour
Train
Get started


BigCatVision
Training Images
Performance
Predictions
Train
Quick Test
Iterations
Probability Threshold: 50%
Iteration 3
Trained : moments ago with General [A2] domain
Iteration 2
Trained : 5 hours ago with General [A2] domain
Iteration 1
Trained : 5 hours ago with General [A2] domain
Publish
Prediction URL
Delete
Export
Finished training on 4/11/2025, 4:39:30 PM using General [A2] domain
Iteration id: 94d5ce7c-2b80-43a5-b2d1-a60c30af38b2
Classification type: Multiclass (Single tag per image)
Precision
Recall
AP
99.3%
98.6%
99.7%
Performance Per Tag
Tag
Precision
Recall
A.P.
Image count
tiger 100.0% 100.0% 100.0% 180
lion 100.0% 100.0% 100.0% 174
cheetah 100.0% 97.1% 99.7% 172

Test Prediction

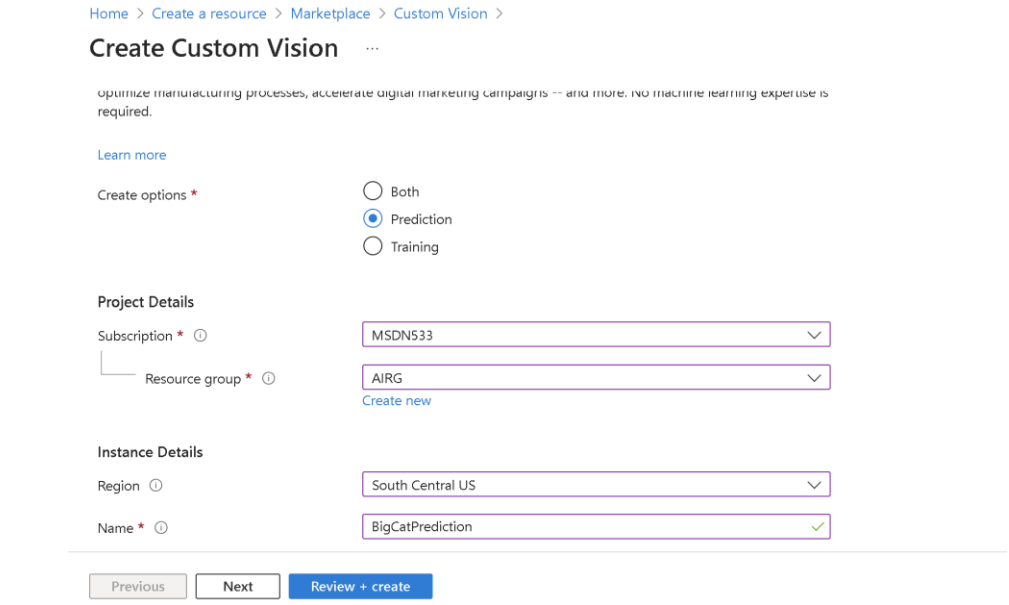


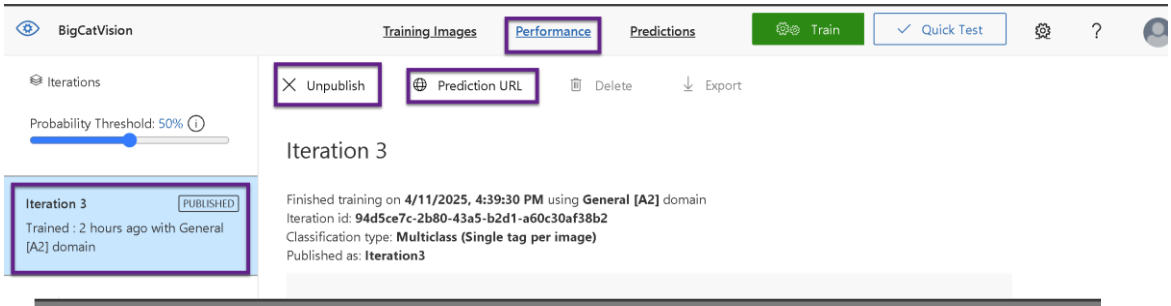


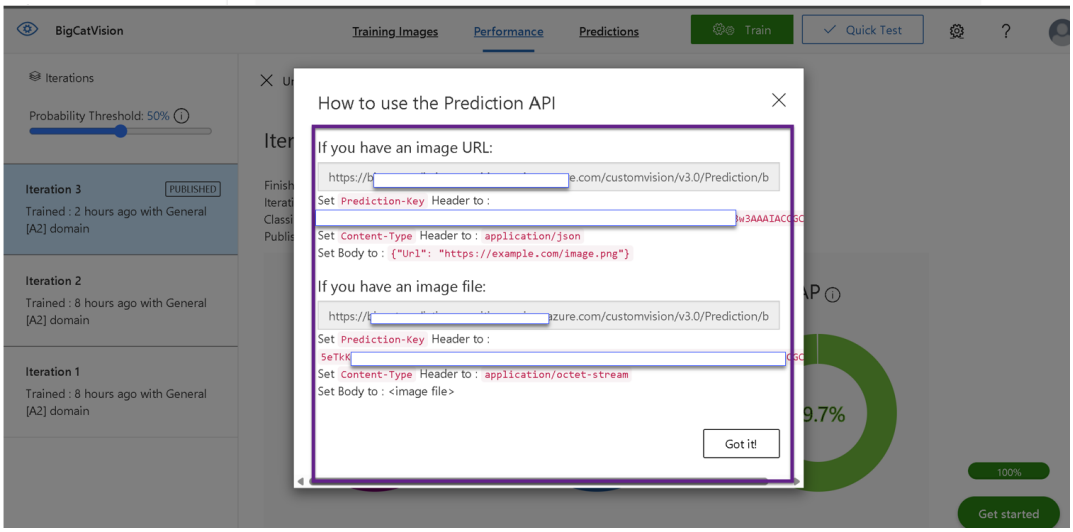


Use a Custom Vision model for prediction

First, create an Azure Custom Vision Prediction resource in the same region you selected earlier. Then, go to customvision.ai, and during the publishing step of your iteration, link it to this prediction resource. This will provide you with both the Prediction URL and Prediction Key required in the Python code that follows.







Change the prediction URL ending from /url to /image in Python code.

Create web app for bigcat prediction using streamlit on Linux

Step 1: Install Python

Rocky Linux comes with Python, but let's make sure you have a good version (3.8+ is great).

```
sudo dnf install python3 python3-pip -y
```

Step 2: Create Virtual Environment (Optional but Recommended)

```
python3 -m venv streamlit-env
source streamlit-env/bin/activate
```

Step 3: Install Streamlit

```
pip install streamlit
```

🛠️ **Step 4: Create Your Streamlit App File**

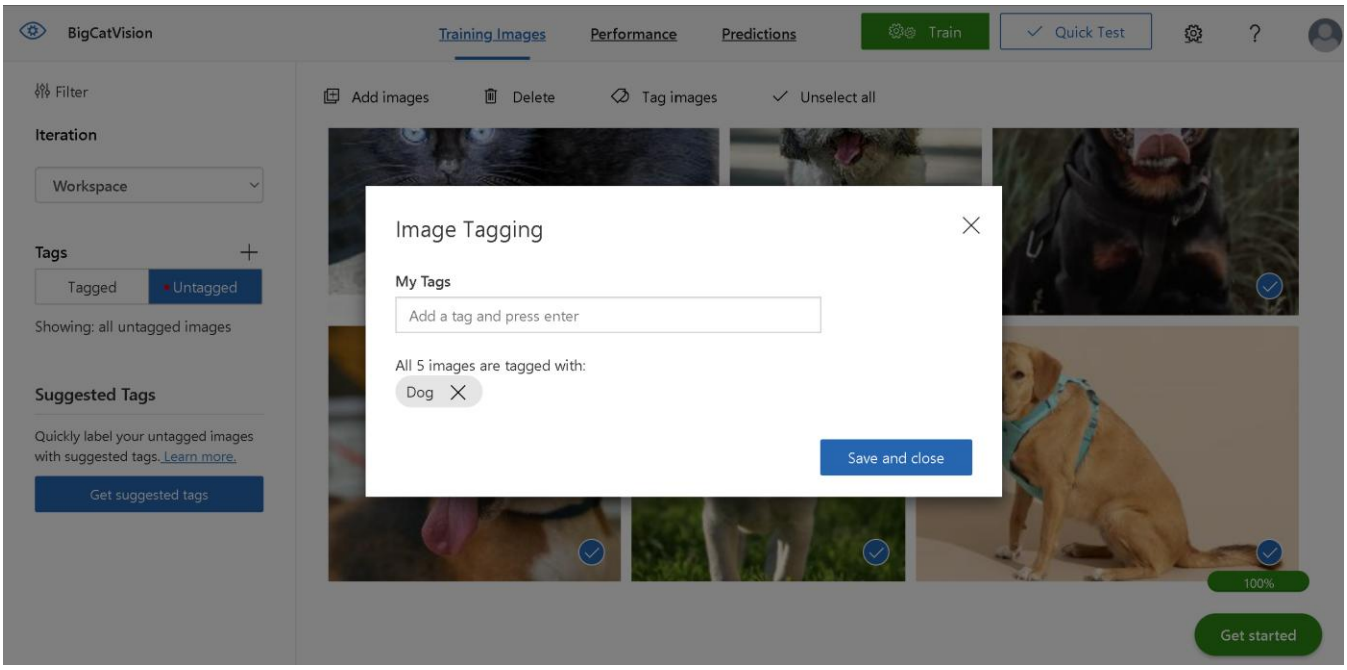
```
import streamlit as st
import requests
st.title("🐾 BigCatVision - Identify Your Big Cat")
uploaded_file = st.file_uploader("Upload an image", type=["jpg", "jpeg", "png"])
PREDICTION_URL =
"https://bigcatprediction.cognitiveservices.azure.com/customvision/v3.0/Prediction/bxxxxxxxxxxxxxx
xxxxxx6a/classify/iterations/Iteration3/image"
PREDICTION_KEY = "xxxxxxxxxxxxxxxxbC2DKmb3XvcKlQFaEiexxxxxxxxxxxxxxxxxxxxxx"
headers = {
    "Prediction-Key": PREDICTION_KEY,
    "Content-Type": "application/octet-stream"
}
if uploaded_file:
    image_bytes = uploaded_file.read()
    st.image(image_bytes, caption="Uploaded Image", use_column_width=True)
    if st.button("🔍 Predict"):
        st.info("Classifying the image...")
        response = requests.post(PREDICTION_URL, headers=headers, data=image_bytes)
        try:
            result = response.json()
            predictions = result.get("predictions", [])
            if predictions:
                st.subheader("🎯 Top Predictions:")
                for pred in predictions[:3]:  # Show top 3
                    tag = pred["tagName"]
                    confidence = pred["probability"] * 100
                    st.write(f"**{tag.title()}**: {confidence:.2f}%")
            else:
                st.warning("No predictions returned.")
        except Exception as e:
            st.error(f"❌ Prediction failed: {e}")
```

Step 5: Run Your App

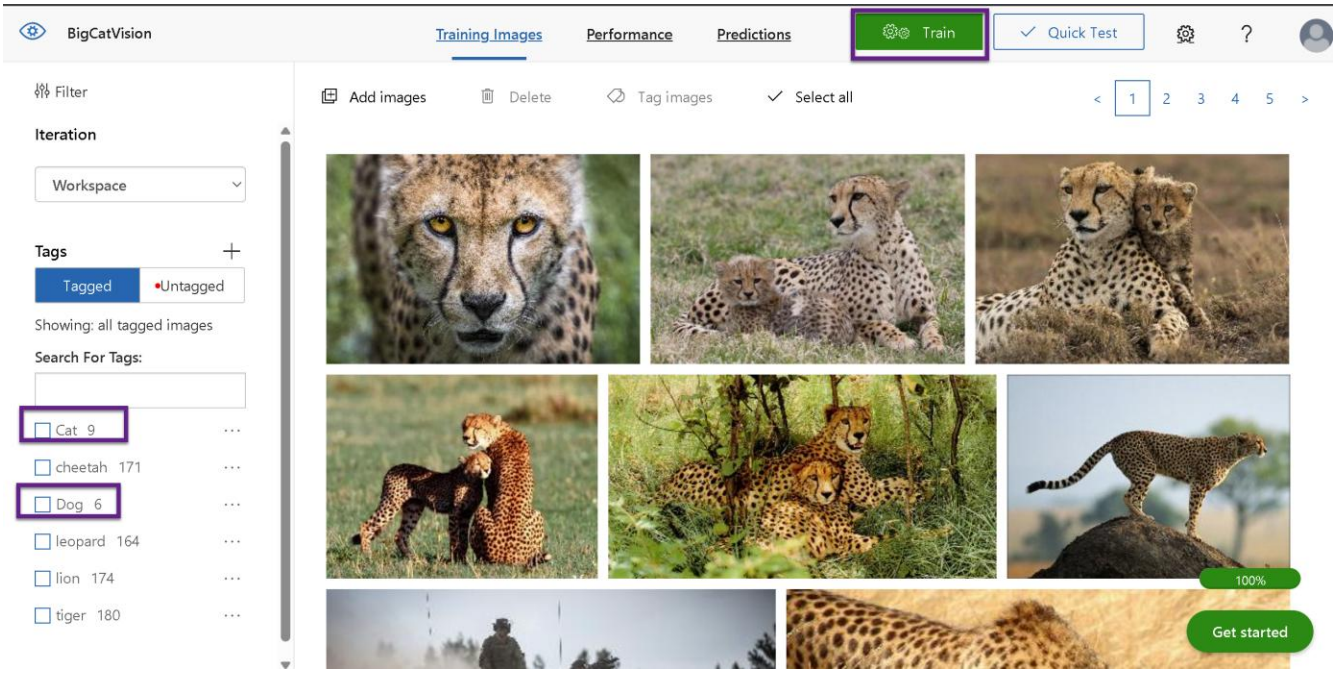
```
streamlit run app.py --server.address=0.0.0.0
```

Then access via http://<Rocky_Linux_IP>:8501

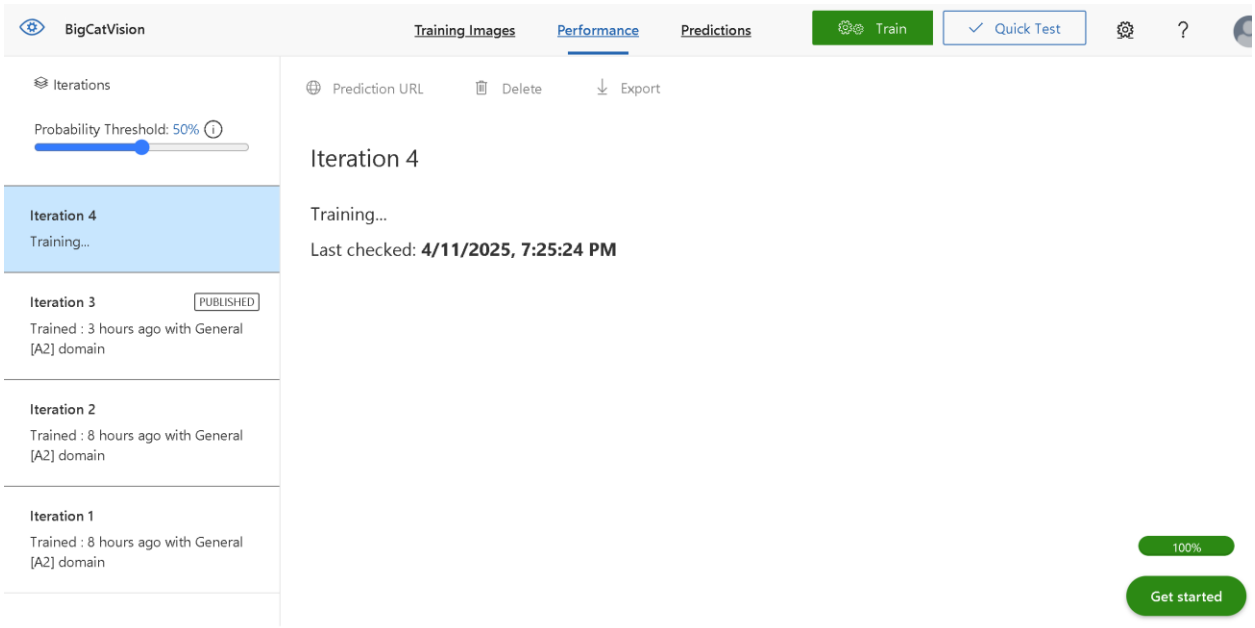
Update Images and link them to new tag



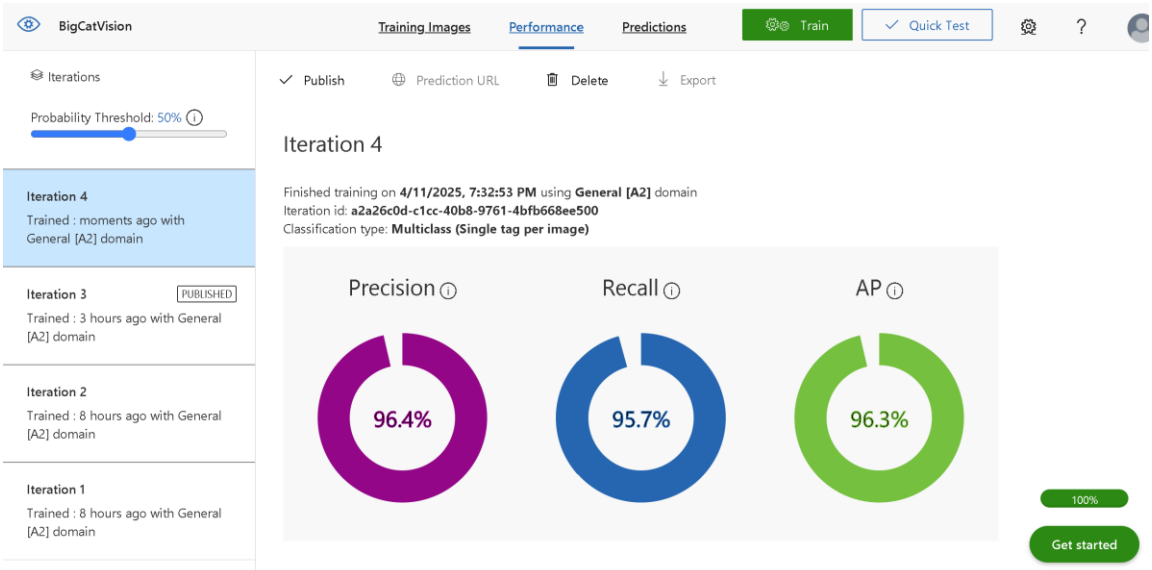
We have added two new tags and need to retrain the model so it can predict images with the new tags. This will produce a new iteration that includes the prediction URL and prediction key.

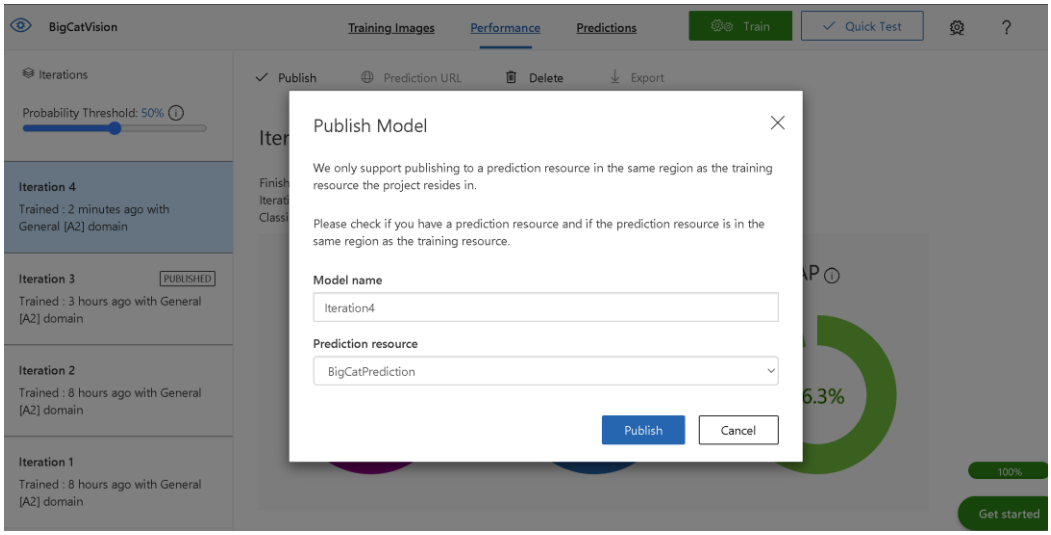


Training started, and a new iteration has been established.

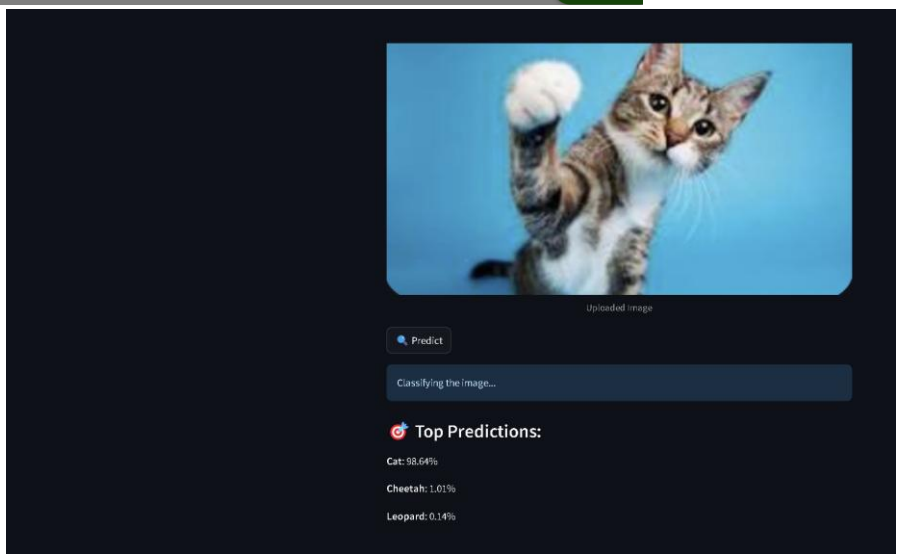


It's time to publish and obtain the data required to use in python.





It is now possible to predict the cats.

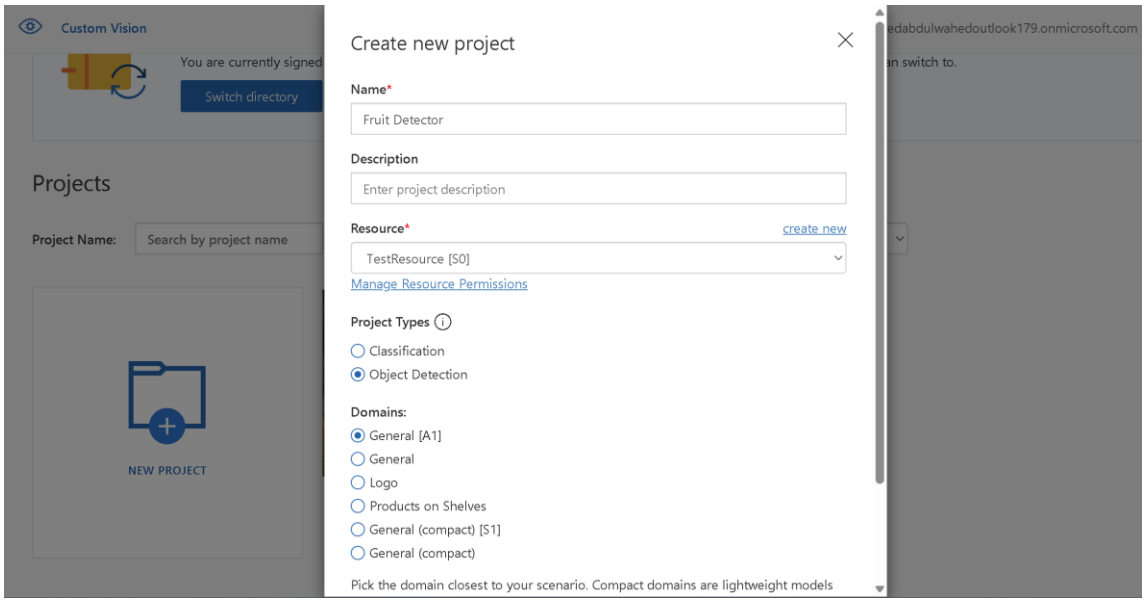


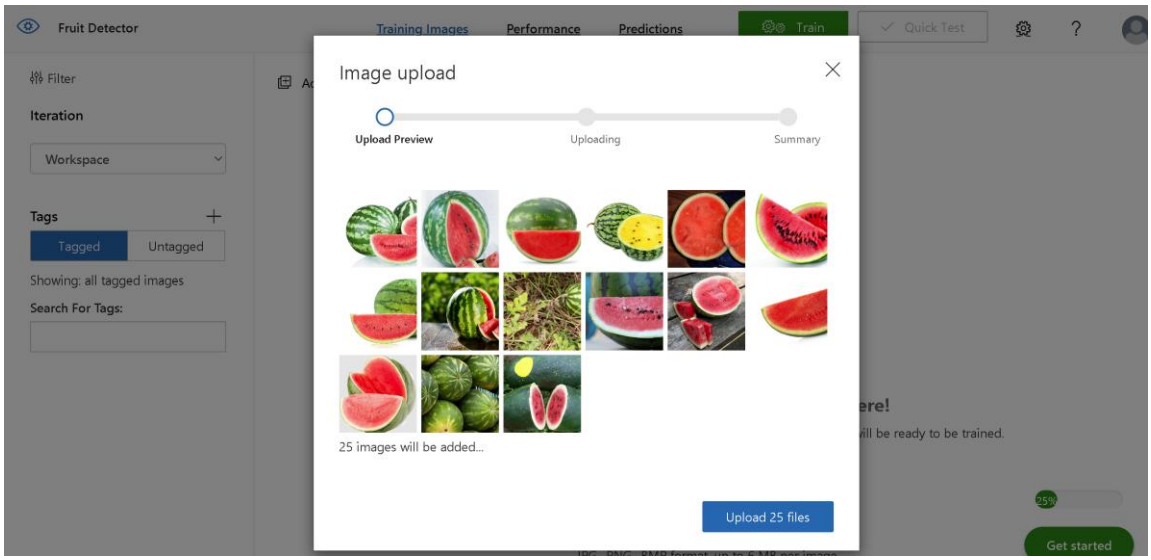
Object Detection with Custom Vision

Unlike image classification (which labels the entire image), **object detection**:

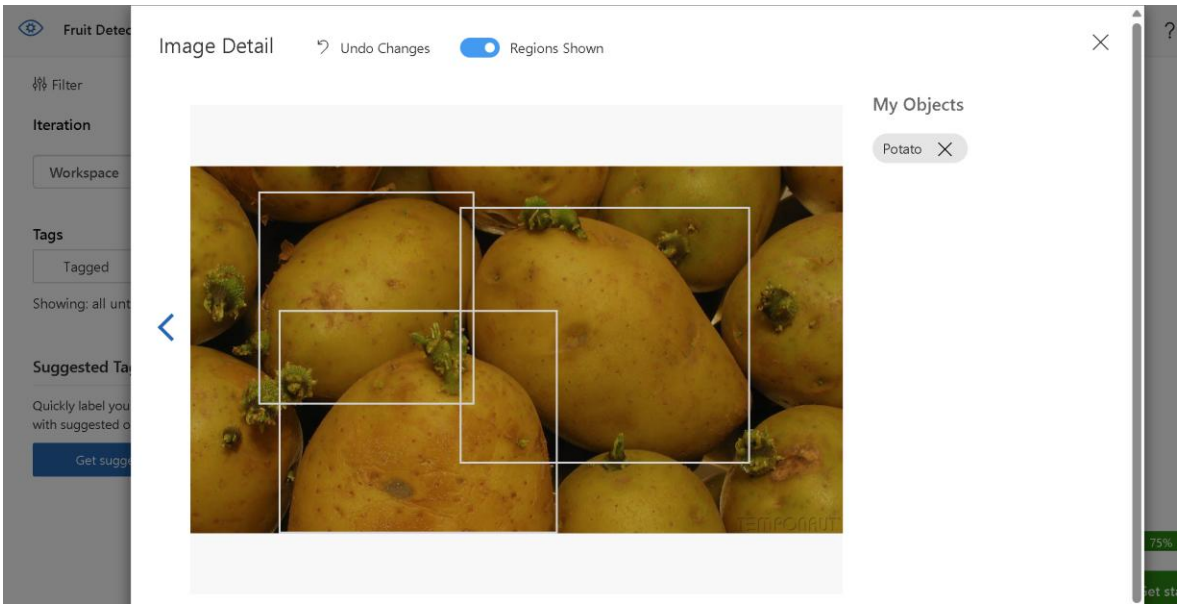
- Detects **what** objects are in the image
- Locates **where** each object is (bounding boxes)

Instead of just saying “this is a lion,” object detection says: “There’s a lion at coordinates (x1, y1, x2, y2)” in the image.

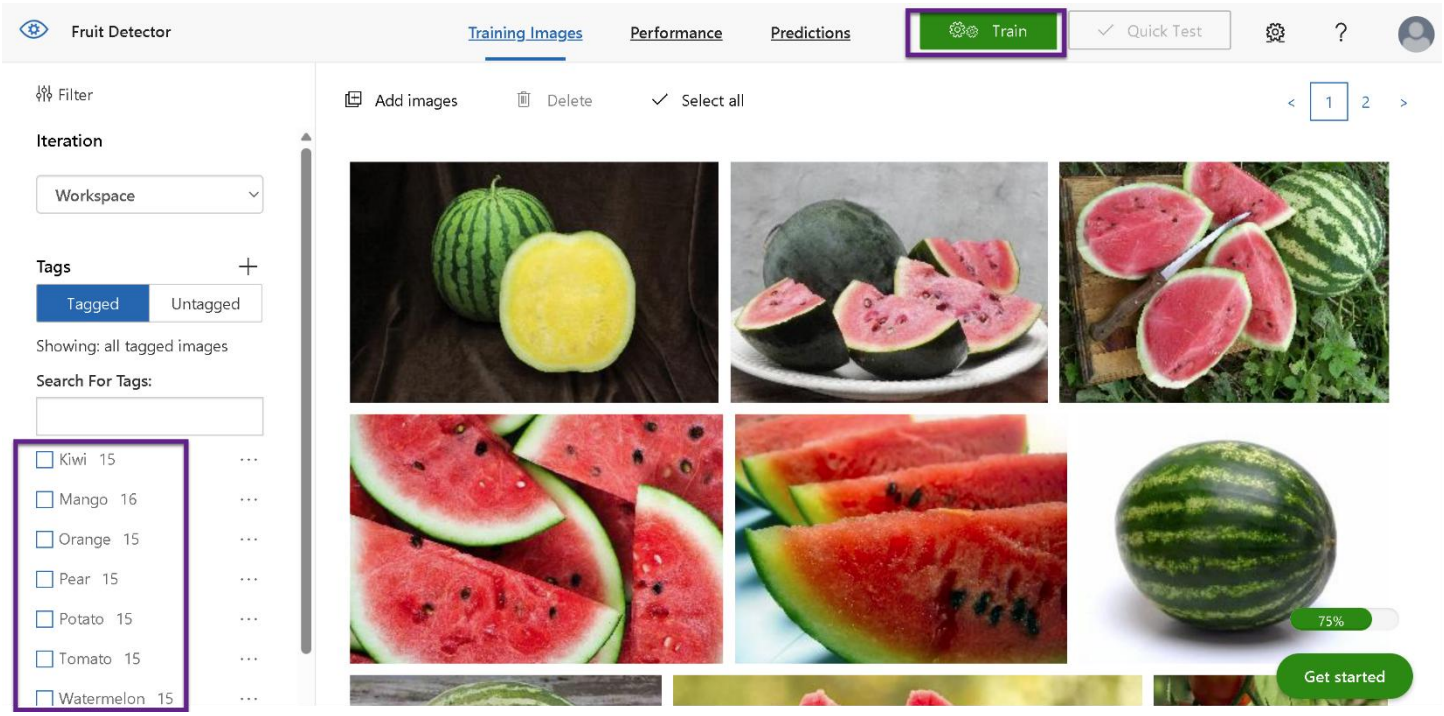


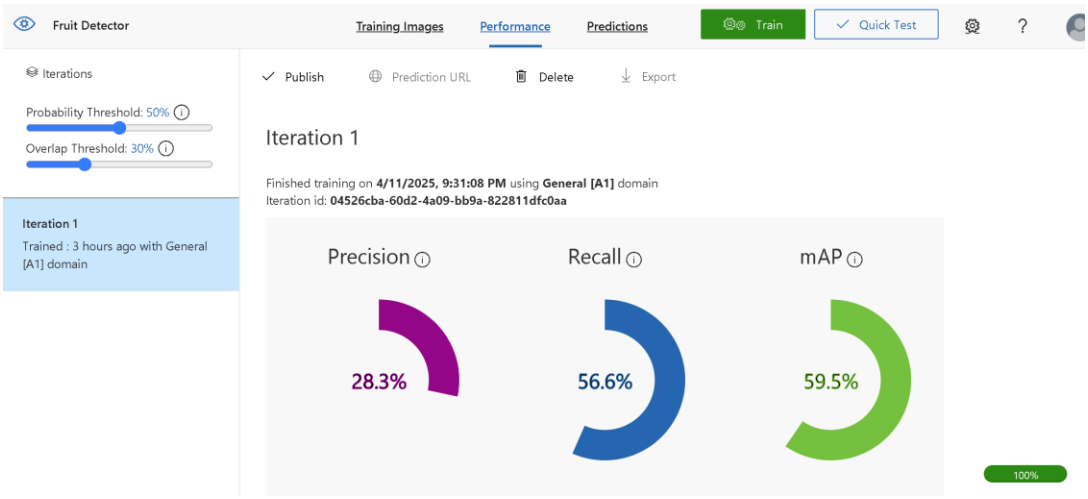


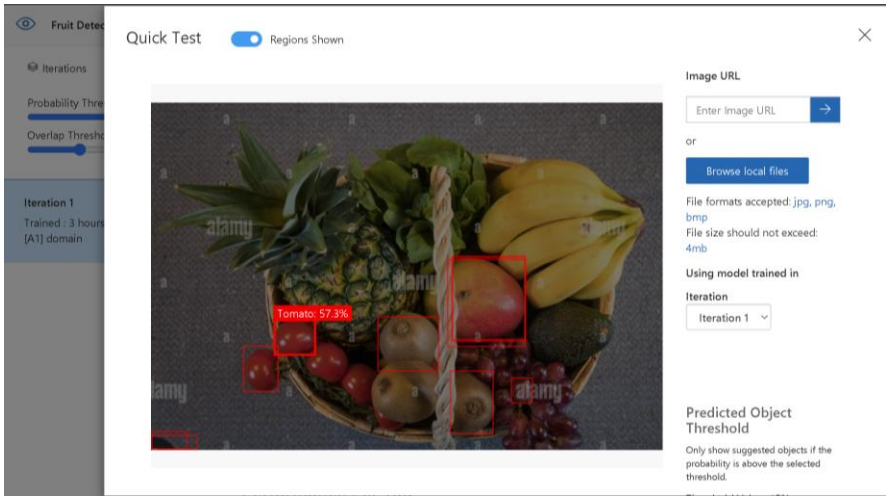
Tag each object in the image.



Added different fruits and tagged them all (to start train you have to add 15 image for each tag)

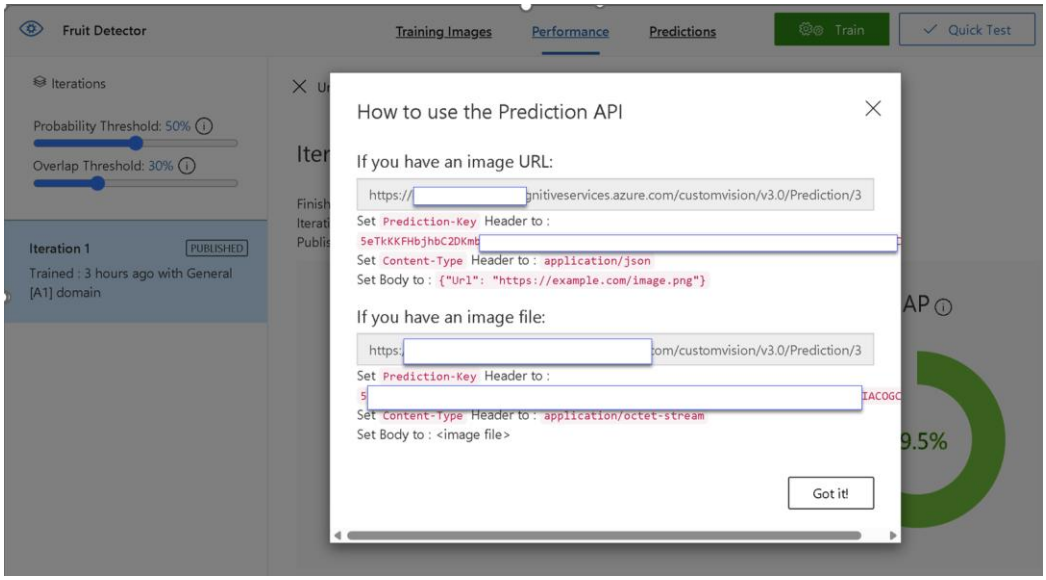






`

now we can publish with streamlit



Deploy Object Detection App with Streamlit on Linux

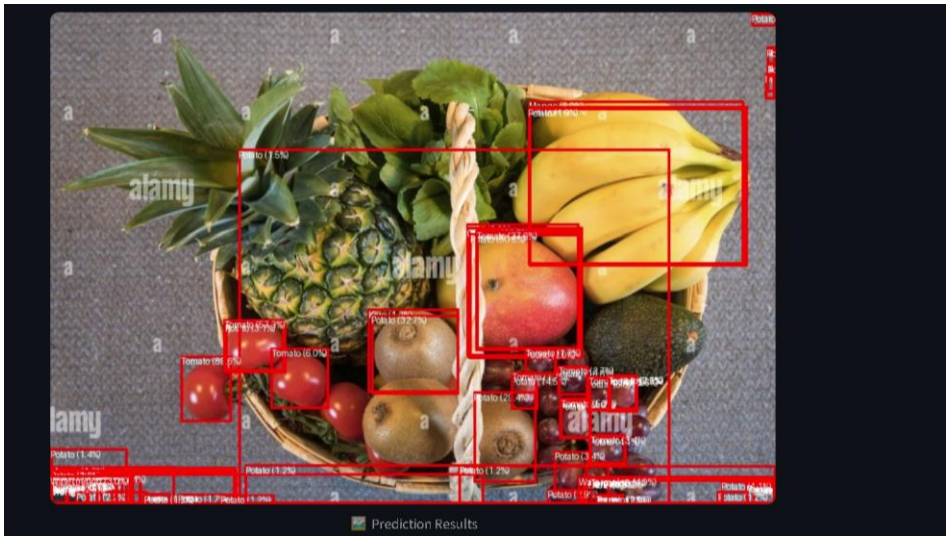
Follow the previous steps to build an app with Streamlit using this Python code.

```
import streamlit as st
import requests
from PIL import Image, ImageDraw
import io
st.title("🍉 Fruit Detector - Object Detection with Azure Custom Vision")
uploaded_file = st.file_uploader("Upload an image (JPG or PNG)", type=["jpg", "jpeg",
"png"])
PREDICTION_URL =
"https://bigcatprediction.cognitiveservices.azure.com/customvision/v3.0/Prediction/39xxxx
xxxxxxxxxxxxxxxxxxx/detect/iterations/Iteration1/image"
PREDICTION_KEY = "xxxxxxxxxxxxxxxxxxxxxxvcKlQFaEieGcbisaKsH1AW37UemBTzH5JQQJ99BDACLA"
headers = {
    "Prediction-Key": PREDICTION_KEY,
    "Content-Type": "application/octet-stream"
}
def draw_boxes(image, predictions):
    img = image.copy()
    draw = ImageDraw.Draw(img)
    w, h = img.size
    for pred in predictions:
        tag = pred['tagName']
        prob = pred['probability']
        bbox = pred['boundingBox']
        left = bbox['left'] * w
        top = bbox['top'] * h
        width = bbox['width'] * w
        height = bbox['height'] * h
        draw.rectangle([left, top, left+width, top+height], outline='red', width=3)
        draw.text((left, top), f"{tag} ({prob*100:.1f}%)", fill='white')
    return img
if uploaded_file:
    image_data = uploaded_file.read()
    image = Image.open(io.BytesIO(image_data))
    st.image(image, caption="Uploaded Image", use_column_width=True)

    if st.button("🔍 Detect"):
        st.info("Sending image to Azure Object Detection model...")
        try:
            response = requests.post(PREDICTION_URL, headers=headers, data=image_data)
            result = response.json()
            predictions = result.get("predictions", [])
```

```
            if predictions:
                st.subheader("📦 Detected Objects")
                for obj in predictions:
                    st.write(f"🔹 {obj['tagName']} ({obj['probability']*100:.2f}%)")

                boxed_image = draw_boxes(image, predictions)
                st.image(boxed_image, caption="🖼️ Prediction Results",
use_column_width=True)
            else:
                st.warning("No objects detected in this image.")
        except Exception as e:
            st.error(f"❌ Prediction failed: {e}")
```



Deploy Webcam Object Detection App with Streamlit Cloud + GitHub

1. Create the Azure Custom Vision Model

2. Create Your Streamlit App (locally)

1. On your local machine or VM:

```
mkdir fruit-detector
cd fruit-detector
```

2. Create a file app.py and paste the full app code (with camera, counter, snapshot features)
3. Create a requirements.txt:

```
streamlit
streamlit-webrtc
pillow
requests
av
```

3. Push to GitHub

1. Initialize Git:

```
git init
git add .
git commit -m "initial commit with fruit detector app"
```

2. Create a repo on GitHub:
   👉 https://github.com/new → name: fruit-detector
3. Link to GitHub:

git remote add origin https://github.com/YOUR_USERNAME/fruit-detector.git
git push -u origin master

💡 If using HTTPS and GitHub blocks password login, use a **Personal Access Token** as password.

4. Deploy on Streamlit Cloud

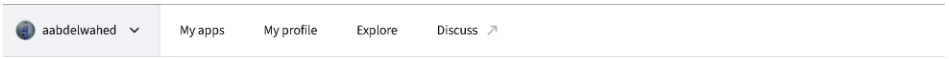
1. Go to 👉 https://share.streamlit.io
2. Log in with GitHub and click **"New App"**
3. Select:
   - Repository: YOUR_USERNAME/fruit-detector
   - Branch: master
   - File: app.py
4. Click **Deploy**

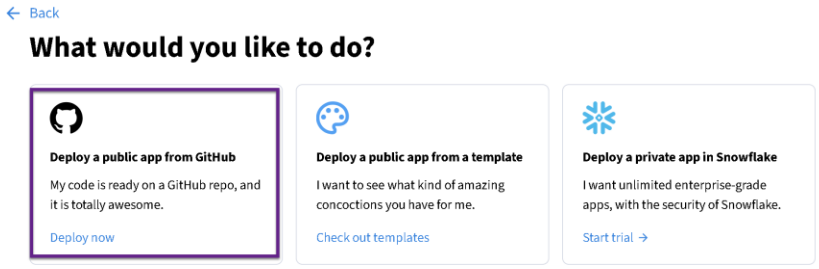
5. App is Live!

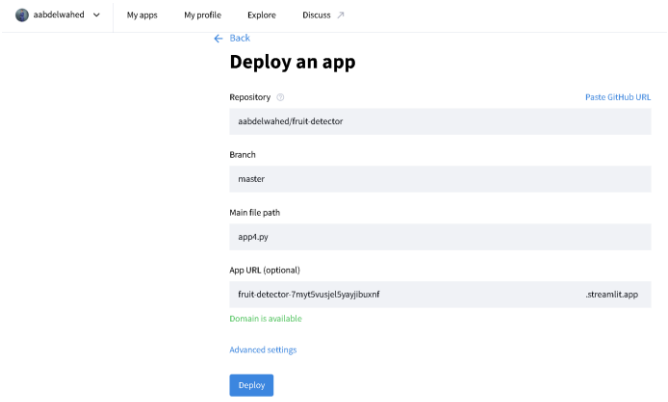
You'll get a public URL like:

```
https://your-username-fruit-detector.streamlit.app
```

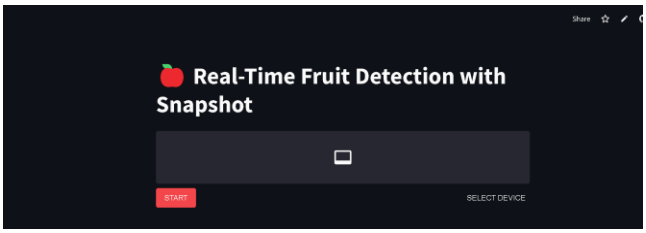
Open it from mobile → browser will ask to allow webcam
Snapshots will save locally in snapshots/ folder on server



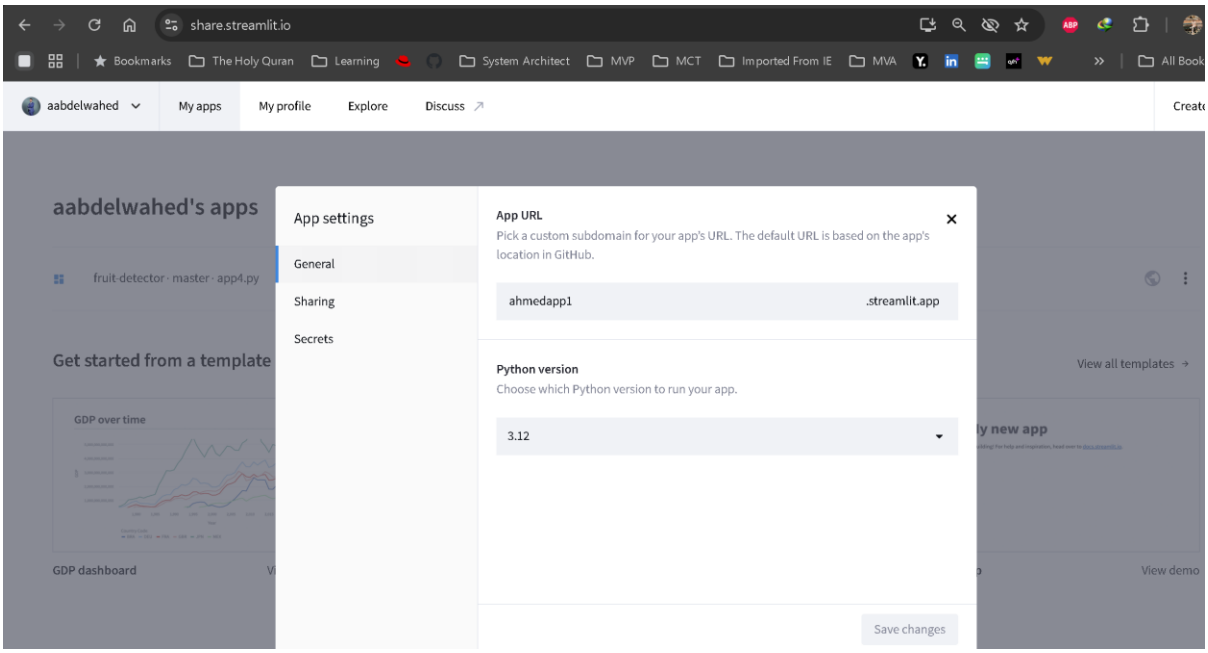




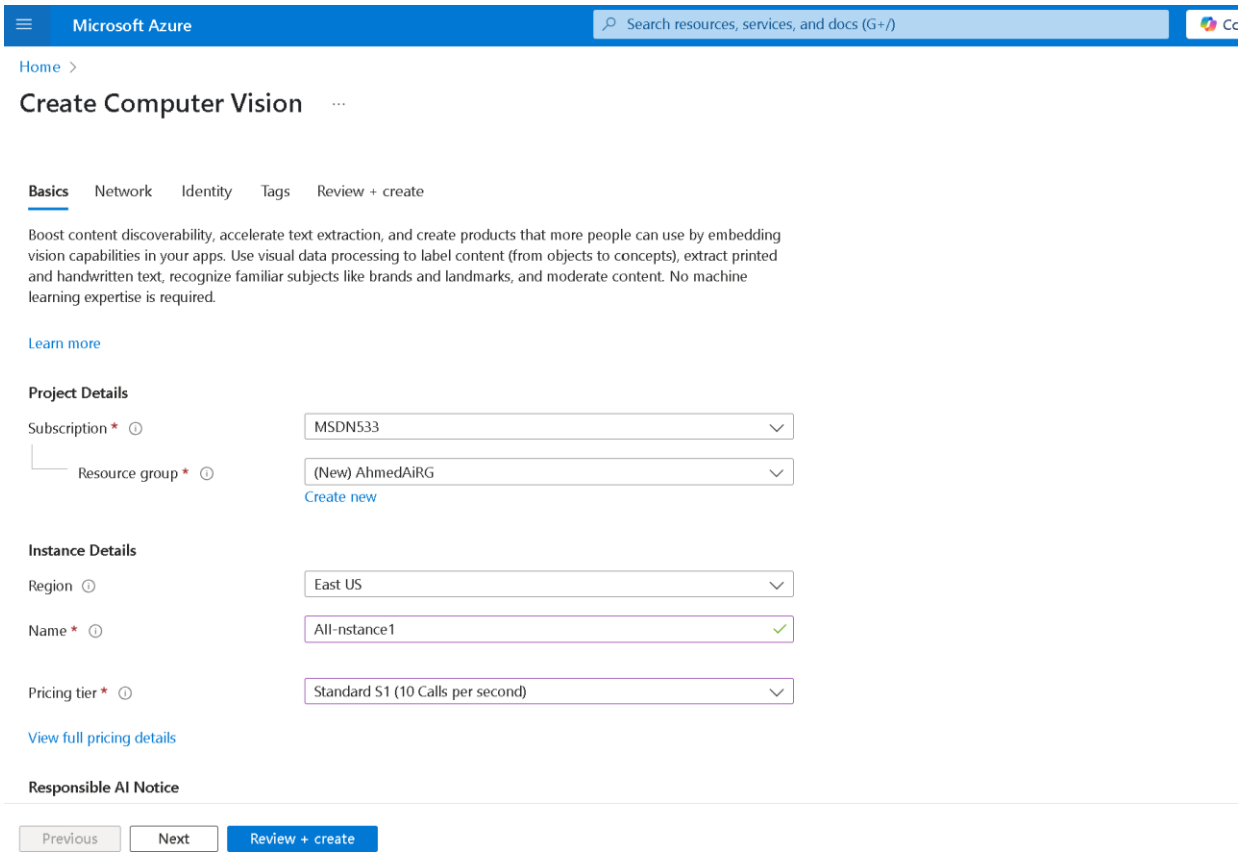
https://ahmedapp1.streamlit.app/

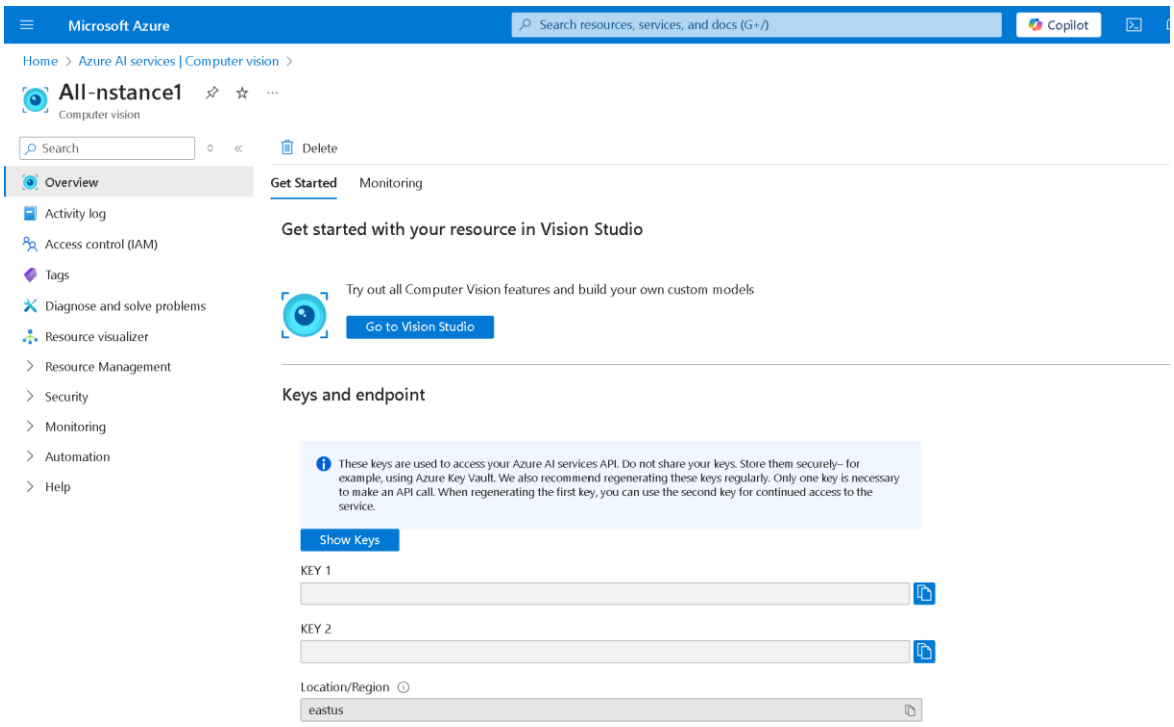


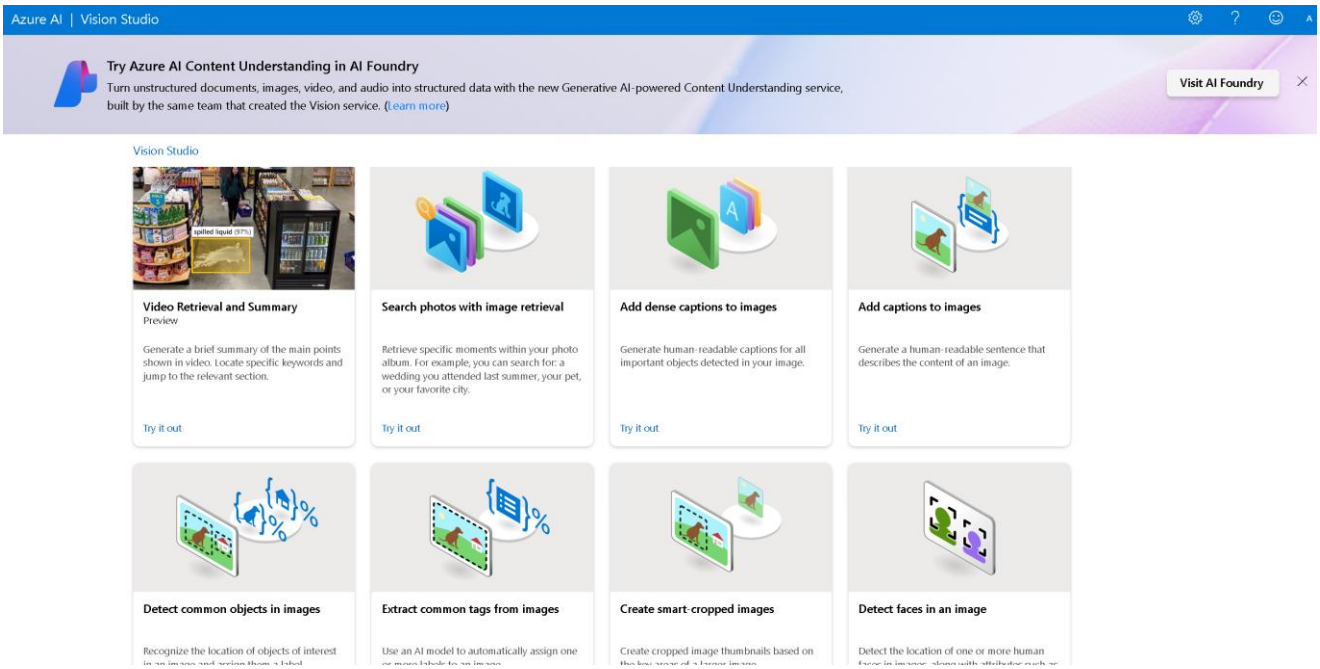
You can adjust app settings, including your public name.

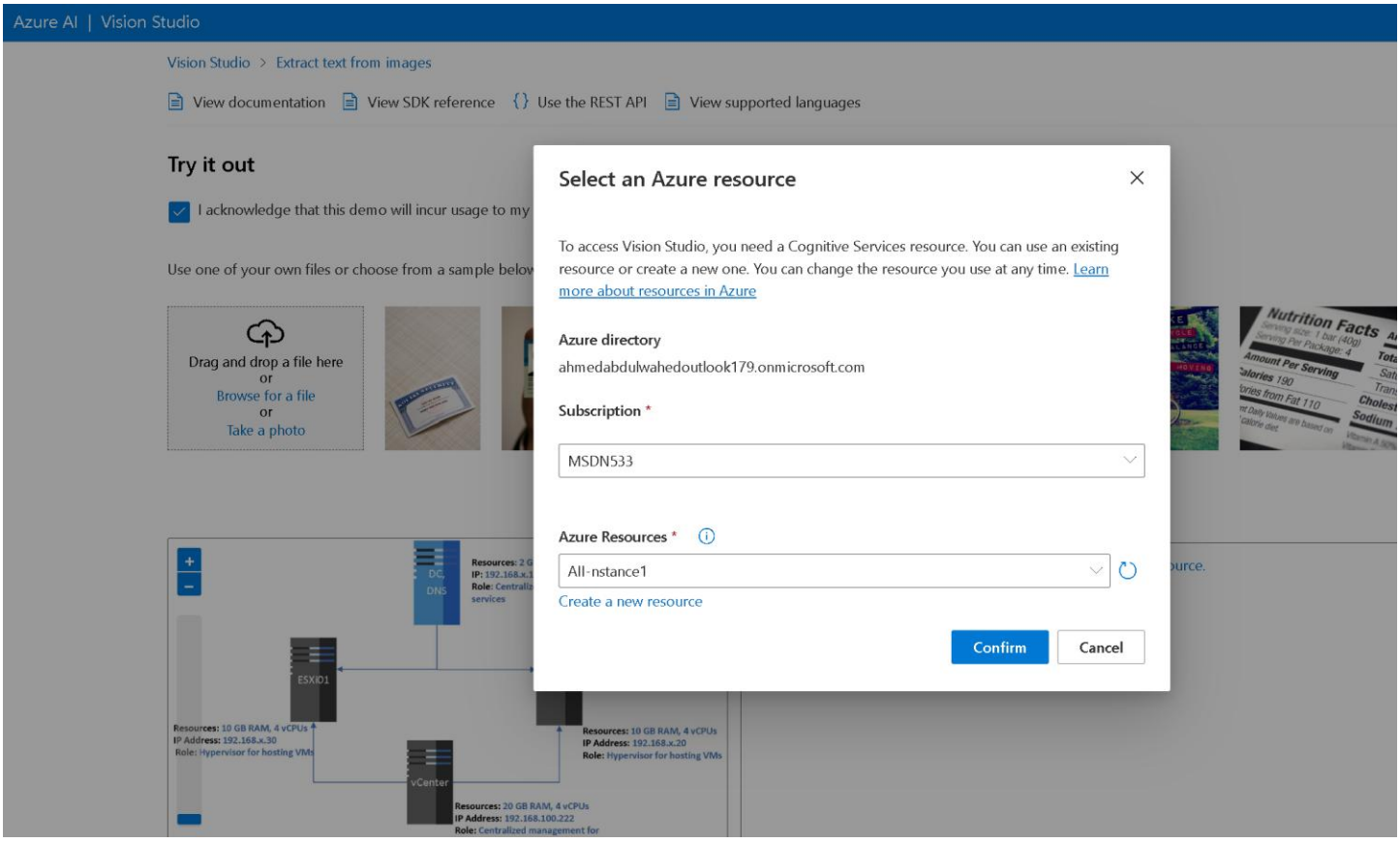


Computer Vision






Azure AI | Vision Studio
Try Azure AI Content Understanding in AI Foundry
Turn unstructured documents, images, video, and audio into structured data with the new Generative AI-powered Content Understanding service, built by the same team that created the Vision service. (Learn more)
Visit AI Foundry
Vision Studio
Video Retrieval and Summary
Preview
Generate a brief summary of the main points shown in video. Locate specific keywords and jump to the relevant section.
Try it out
Search photos with image retrieval
Retrieve specific moments within your photo album. For example, you can search for: a wedding you attended last summer, your pet, or your favorite city.
Try it out
Add dense captions to images
Generate human-readable captions for all important objects detected in your image.
Try it out
Add captions to images
Generate a human-readable sentence that describes the content of an image.
Try it out
Detect common objects in images
Extract common tags from images
Create smart-cropped images
Detect faces in an image


Azure AI | Vision Studio
Vision Studio > Extract text from images
View documentation
View SDK reference
Use the REST API
View supported languages
Try it out
I acknowledge that this demo will incur usage to my
Use one of your own files or choose from a sample belov
Drag and drop a file here
or
Browse for a file
or
Take a photo
Select an Azure resource
To access Vision Studio, you need a Cognitive Services resource. You can use an existing resource or create a new one. You can change the resource you use at any time. Learn more about resources in Azure
Azure directory
ahmedabdulwahedoutlook179.onmicrosoft.com
Subscription *
MSDN533
Azure Resources *
All-nstance1
Create a new resource
Confirm
Cancel
ESXID1
vCenter

Step-by-Step publish setup on Ubuntu on Azure

Step 1: Install Required Packages

SSH into your Ubuntu VM and run:

```
sudo apt update && sudo apt install -y python3-pip python3-venv
pip3 install flask requests
```

Step 2: Project Structure

```
mkdir ocr_web_app
cd ocr_web_app
python3 -m venv venv
source venv/bin/activate
pip install flask requests
mkdir templates uploads
touch app.py templates/index.html
```

Step 3: app.py – The Flask Web App

Paste the following into app.py:

```
from flask import Flask, request, render_template
import os
import requests
app = Flask(__name__)
UPLOAD_FOLDER = 'uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
# Replace with your Azure credentials
AZURE_KEY = "YOUR_AZURE_KEY"
AZURE_ENDPOINT = "https://YOUR_REGION.api.cognitive.microsoft.com"
OCR_URL = AZURE_ENDPOINT + "/vision/v3.2/ocr"
@app.route("/", methods=["GET", "POST"])
def upload_file():
    extracted_text = ""
    if request.method == "POST":
        file = request.files['file']
        if file:
            filepath = os.path.join(app.config['UPLOAD_FOLDER'], file.filename)
            file.save(filepath)
            with open(filepath, 'rb') as image_data:
                headers = {
                    'Ocp-Apim-Subscription-Key': AZURE_KEY,
                    'Content-Type': 'application/octet-stream'
                }
                response = requests.post(OCR_URL, headers=headers, data=image_data)
                response.raise_for_status()
                result = response.json()
                for region in result.get("regions", []):
```

```
                        for line in region["lines"]:
                            extracted_text += " ".join([word["text"] for word in
line["words"]]) + "\n"
    return render_template("index.html", extracted_text=extracted_text)
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

Step 4: templates/index.html

Paste this into templates/index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>OCR Web App</title>
</head>
<body>
    <h2>Upload Image for OCR</h2>
    <form method="POST" enctype="multipart/form-data">
        <input type="file" name="file" required>
        <button type="submit">Extract Text</button>
    </form>
    {% if extracted_text %}
    <h3>Extracted Text:</h3>
    <pre>{{ extracted_text }}</pre>
    {% endif %}
</body>
</html>
```

Step 5: Run the App

```
python3 app.py
```

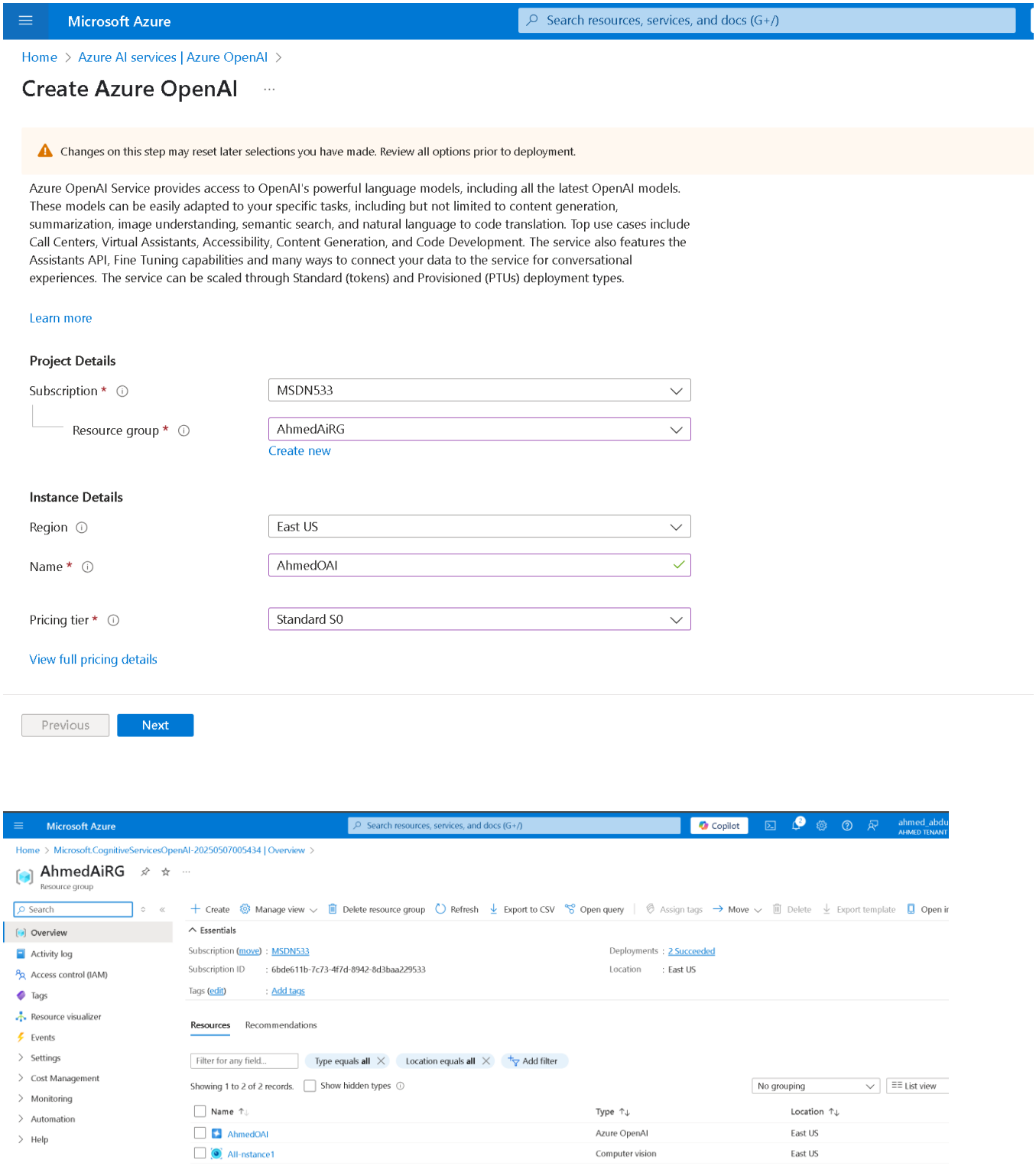
Then open in your browser:

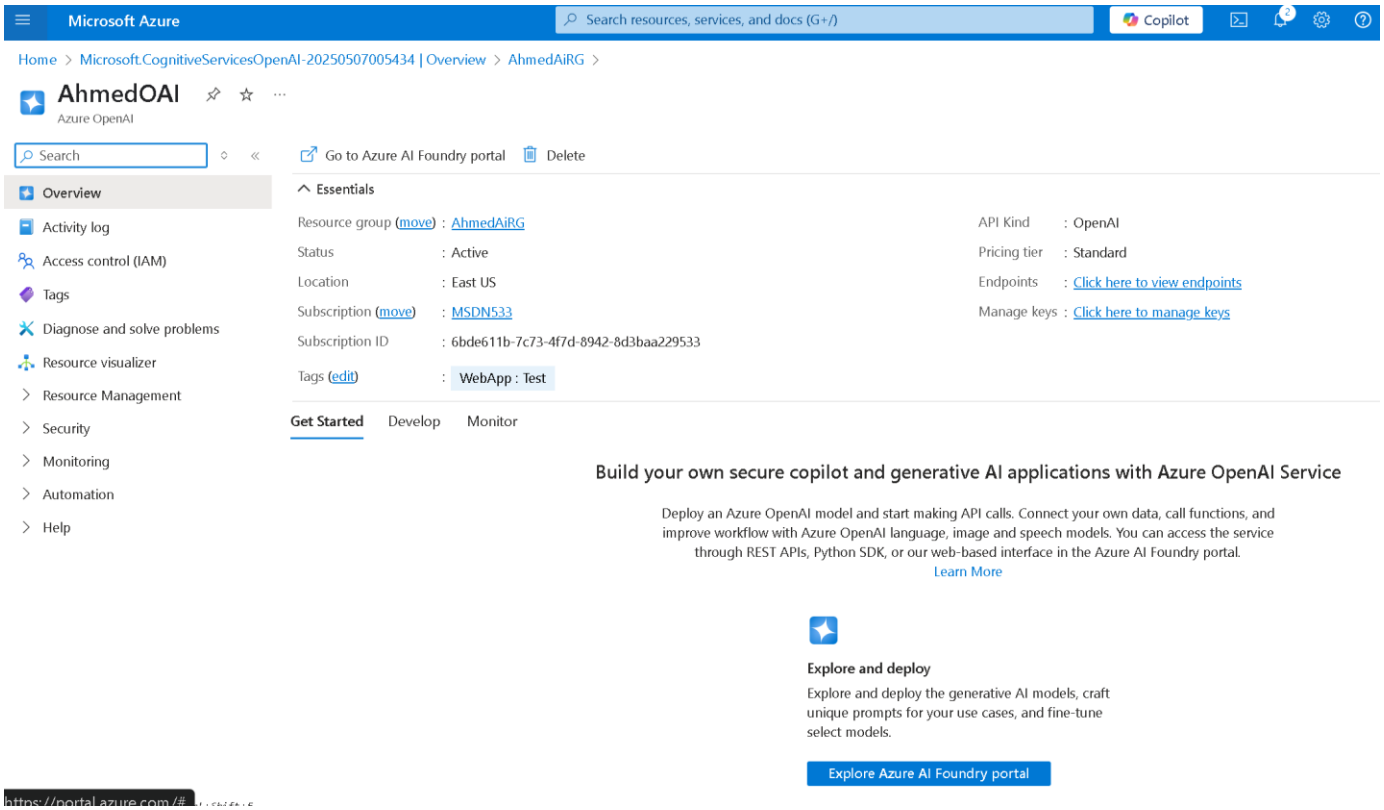
```
http://<your-azure-vm-public-ip>:5000
```

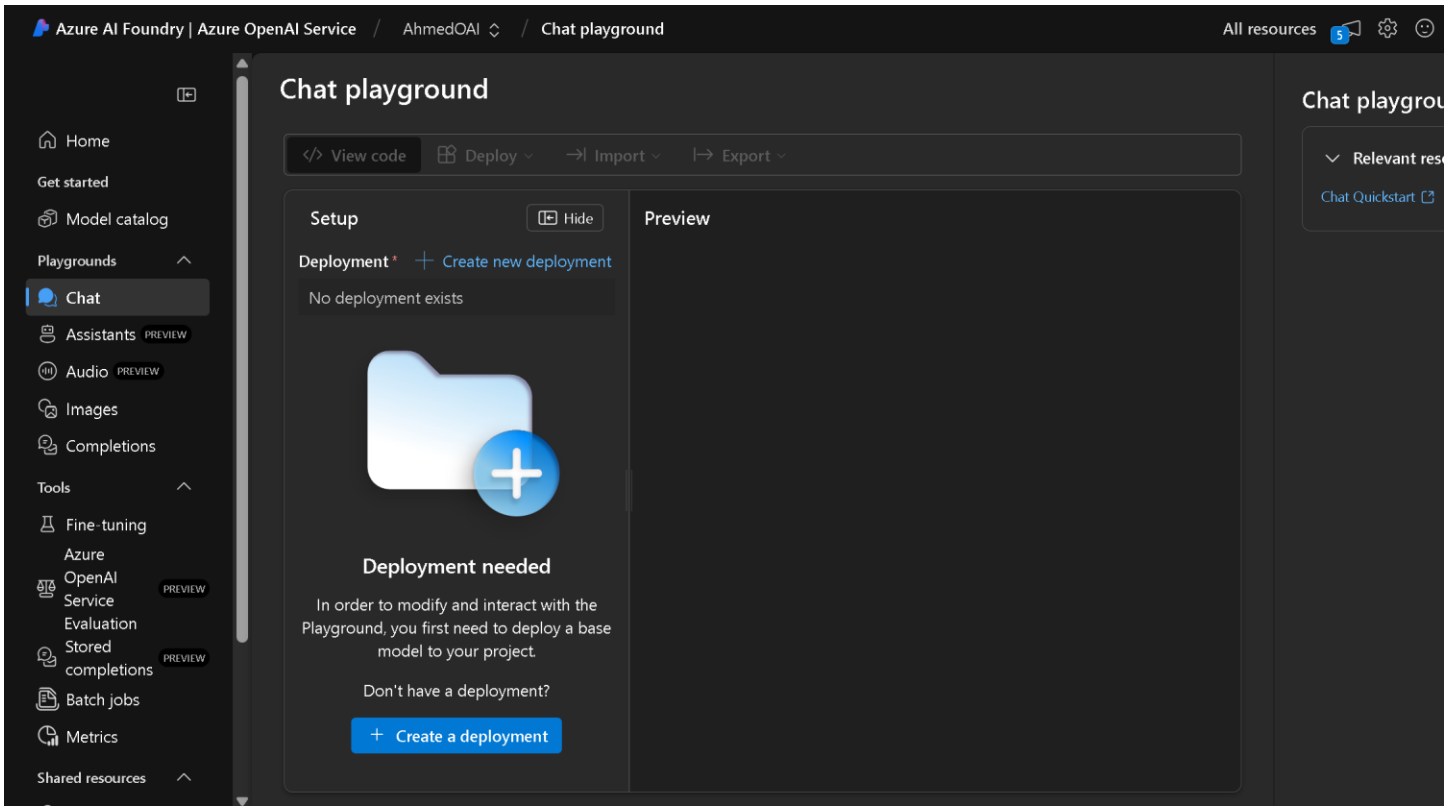
Step 6: Open Port 5000 in Azure

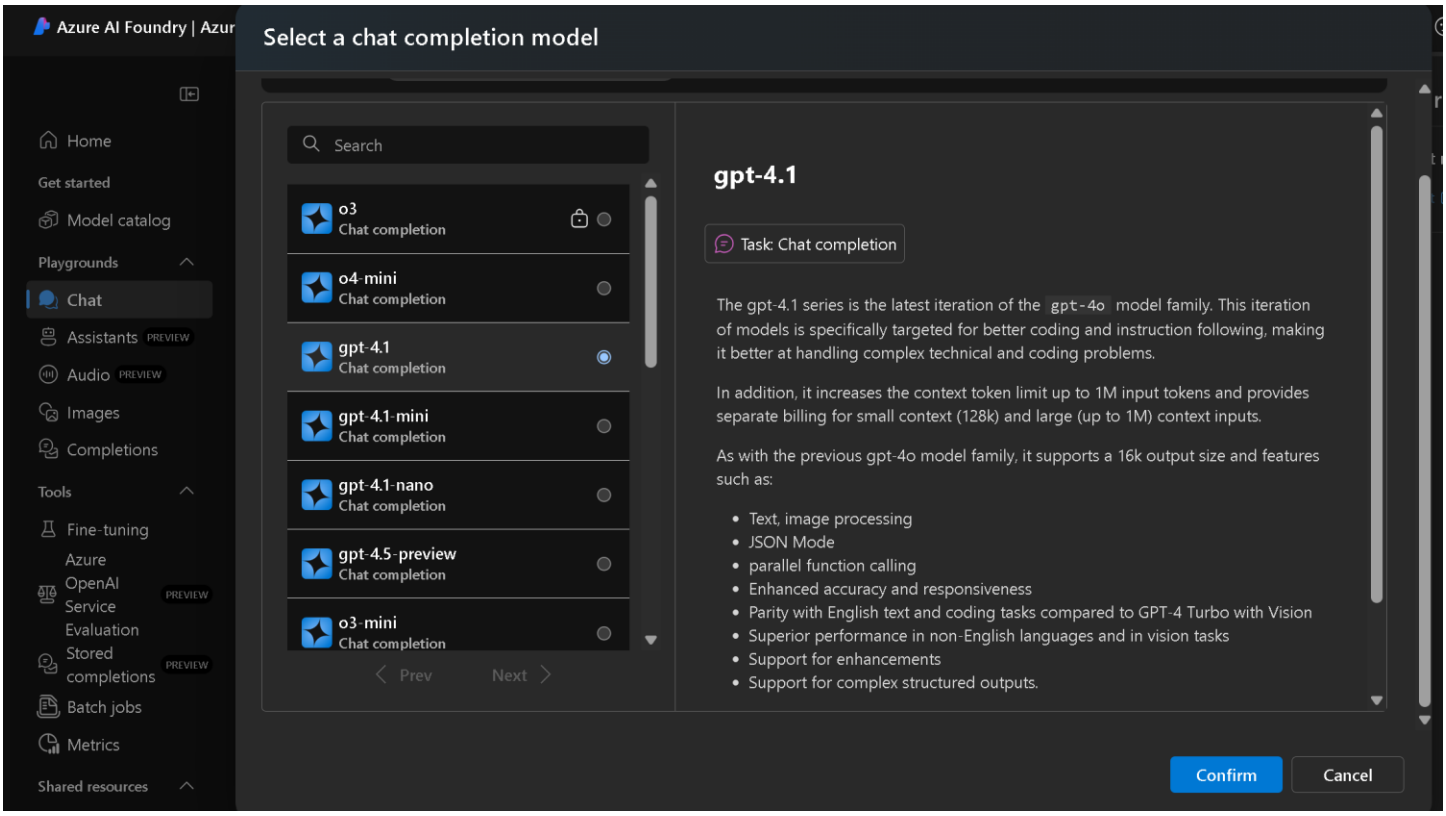
1. Go to Azure Portal
2. Find your VM → *Networking*
3. Add **inbound port rule**:
    - Port: 5000
    - Protocol: TCP
    - Action: Allow

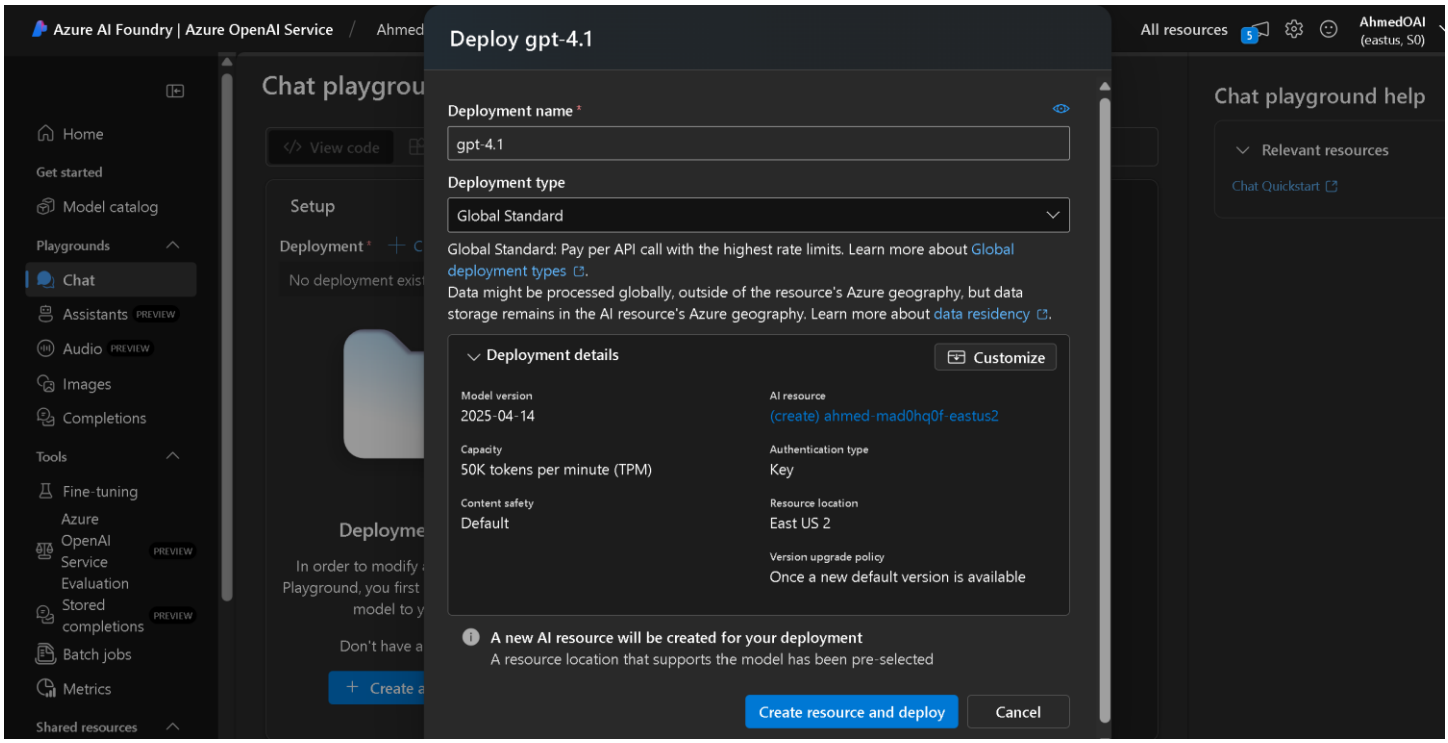
Azure OpenAI

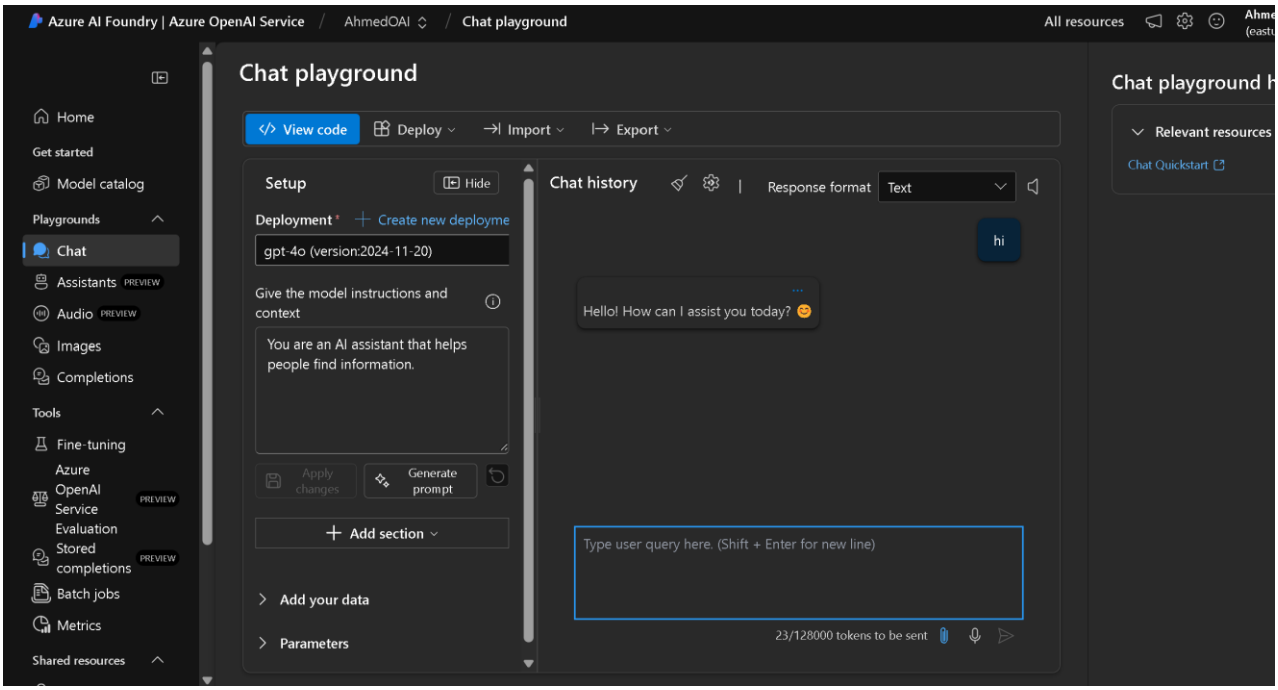



Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Microsoft.CognitiveServicesOpenAI-20250507005434 | Overview > AhmedAiRG >
AhmedOAI
Azure OpenAI
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Resource Management
Security
Monitoring
Automation
Help
Go to Azure AI Foundry portal
Delete
Essentials
Resource group (move) : AhmedAiRG
Status : Active
Location : East US
Subscription (move) : MSDN533
Subscription ID : 6bde611b-7c73-4f7d-8942-8d3baa229533
Tags (edit) : WebApp : Test
API Kind : OpenAI
Pricing tier : Standard
Endpoints : Click here to view endpoints
Manage keys : Click here to manage keys
Get Started
Develop
Monitor
Build your own secure copilot and generative AI applications with Azure OpenAI Service
Deploy an Azure OpenAI model and start making API calls. Connect your own data, call functions, and improve workflow with Azure OpenAI language, image and speech models. You can access the service through REST APIs, Python SDK, or our web-based interface in the Azure AI Foundry portal.
Learn More
Explore and deploy
Explore and deploy the generative AI models, craft unique prompts for your use cases, and fine-tune select models.
Explore Azure AI Foundry portal

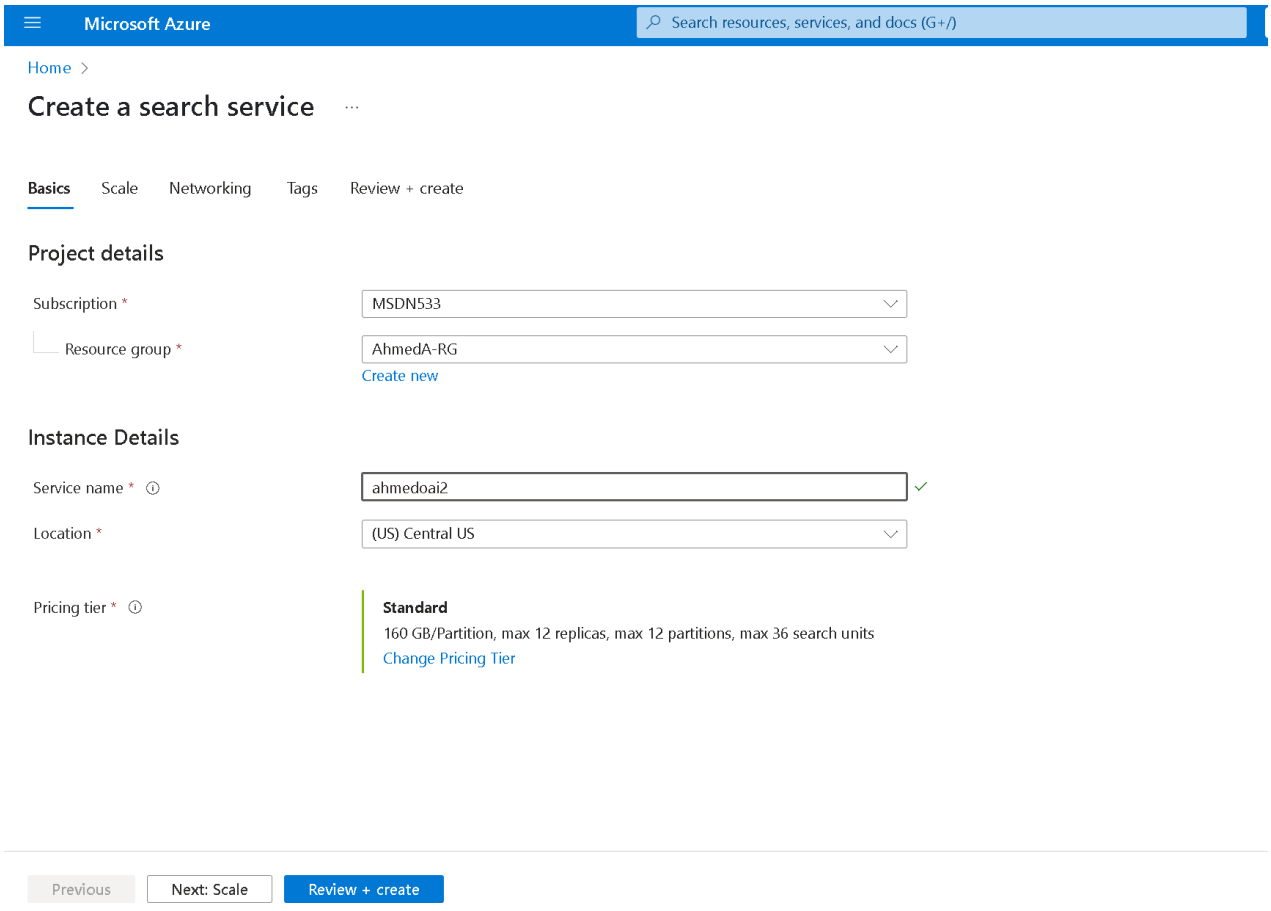

Azure AI Foundry | Azure OpenAI Service
AhmedOAI
Chat playground
All resources
Chat playground
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
View code
Deploy
Import
Export
Setup
Hide
Preview
Deployment *
Create new deployment
No deployment exists
Deployment needed
In order to modify and interact with the Playground, you first need to deploy a base model to your project.
Don't have a deployment?
Create a deployment
Chat playgrou
Relevant res
Chat Quickstart

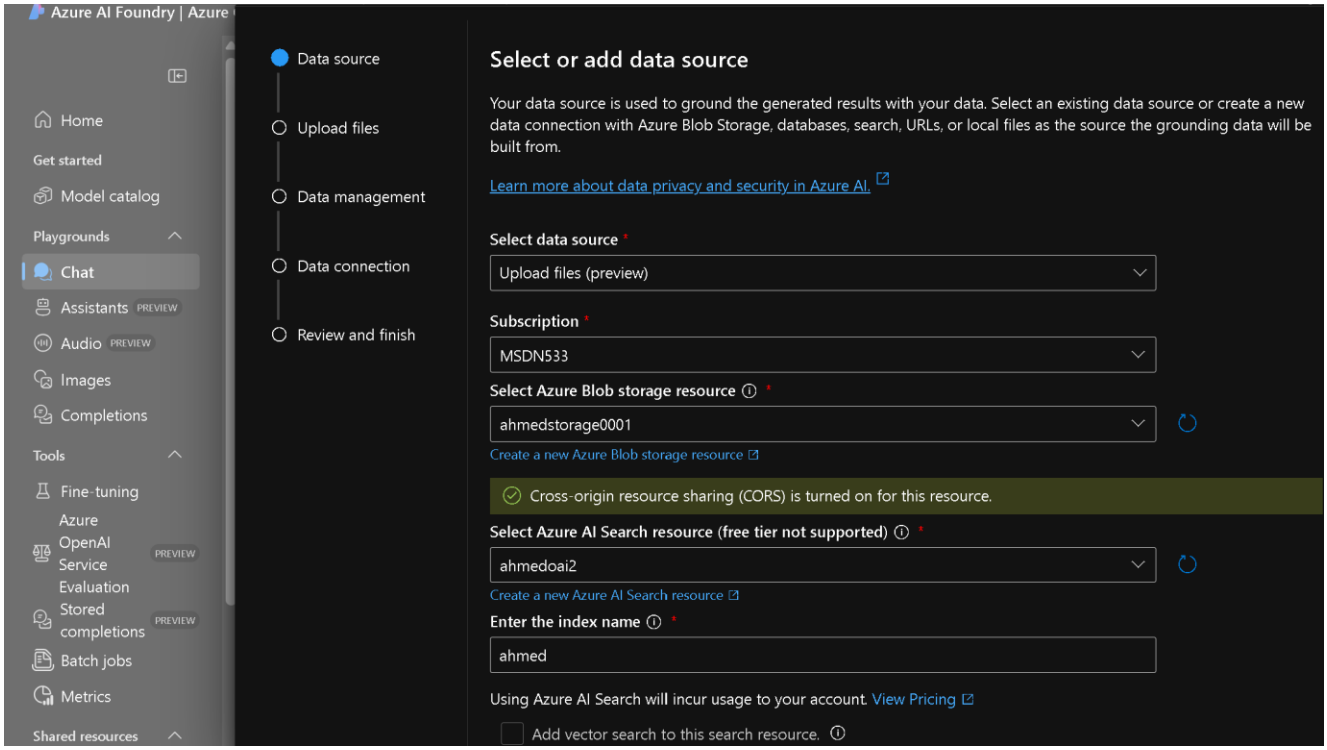

Azure AI Foundry | Azur
Select a chat completion model
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Search
o3 Chat completion
o4-mini Chat completion
gpt-4.1 Chat completion
gpt-4.1-mini Chat completion
gpt-4.1-nano Chat completion
gpt-4.5-preview Chat completion
o3-mini Chat completion
Prev
Next
gpt-4.1
Task: Chat completion
The gpt-4.1 series is the latest iteration of the gpt-4o model family. This iteration of models is specifically targeted for better coding and instruction following, making it better at handling complex technical and coding problems.
In addition, it increases the context token limit up to 1M input tokens and provides separate billing for small context (128k) and large (up to 1M) context inputs.
As with the previous gpt-4o model family, it supports a 16k output size and features such as:
Text, image processing
JSON Mode
parallel function calling
Enhanced accuracy and responsiveness
Parity with English text and coding tasks compared to GPT-4 Turbo with Vision
Superior performance in non-English languages and in vision tasks
Support for enhancements
Support for complex structured outputs.
Confirm
Cancel

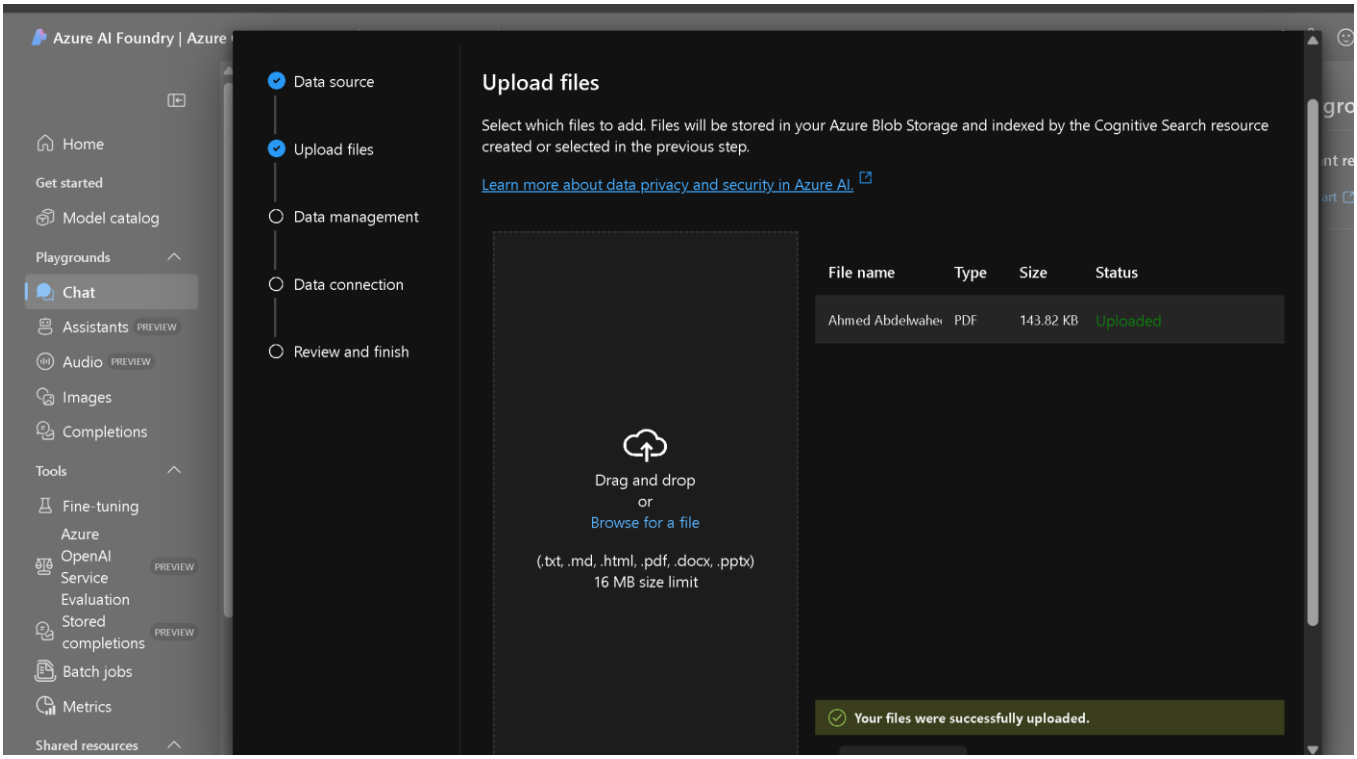

Azure AI Foundry | Azure OpenAI Service
All resources
AhmedOAI (eastus, S0)
Deploy gpt-4.1
Deployment name *
gpt-4.1
Deployment type
Global Standard
Global Standard: Pay per API call with the highest rate limits. Learn more about Global deployment types.
Data might be processed globally, outside of the resource's Azure geography, but data storage remains in the AI resource's Azure geography. Learn more about data residency.
Deployment details
Customize
Model version
2025-04-14
AI resource
(create) ahmed-mad0hq0f-eastus2
Capacity
50K tokens per minute (TPM)
Authentication type
Key
Content safety
Default
Resource location
East US 2
Version upgrade policy
Once a new default version is available
A new AI resource will be created for your deployment
A resource location that supports the model has been pre-selected
Create resource and deploy
Cancel
Chat playground help
Relevant resources
Chat Quickstart

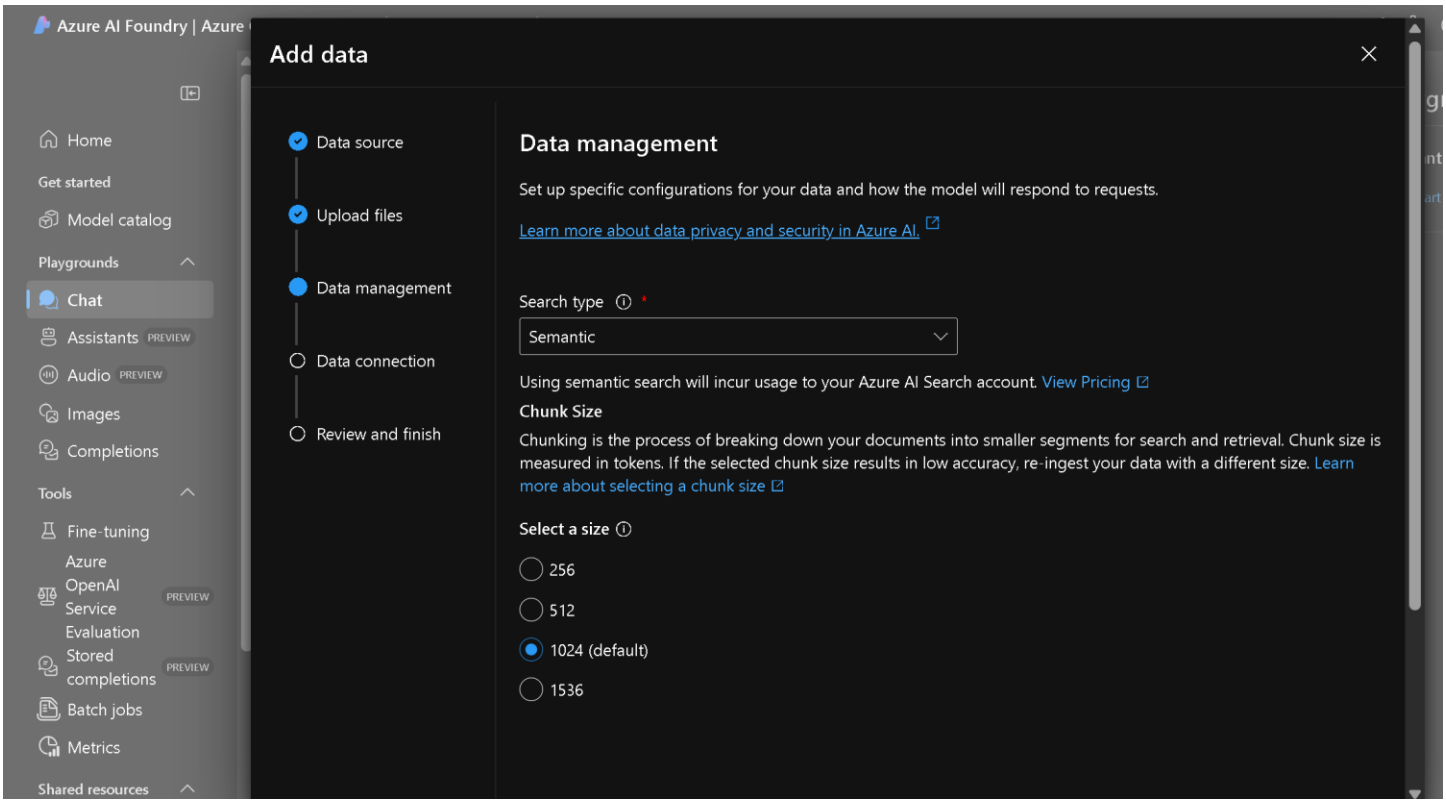


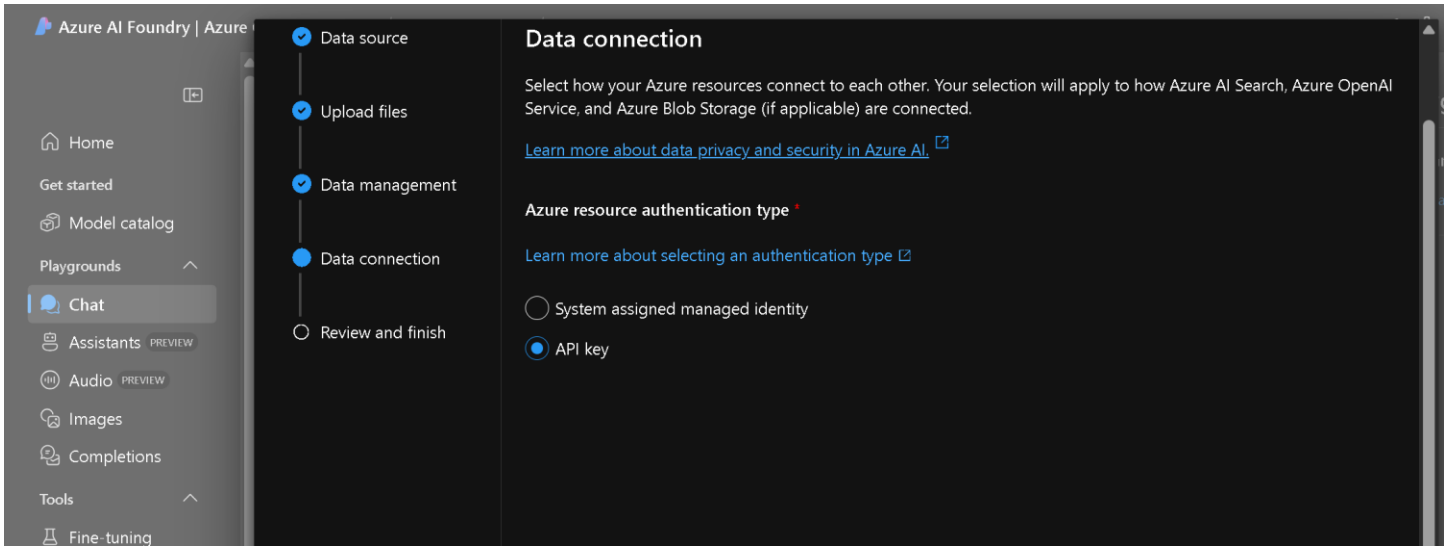
Add your data

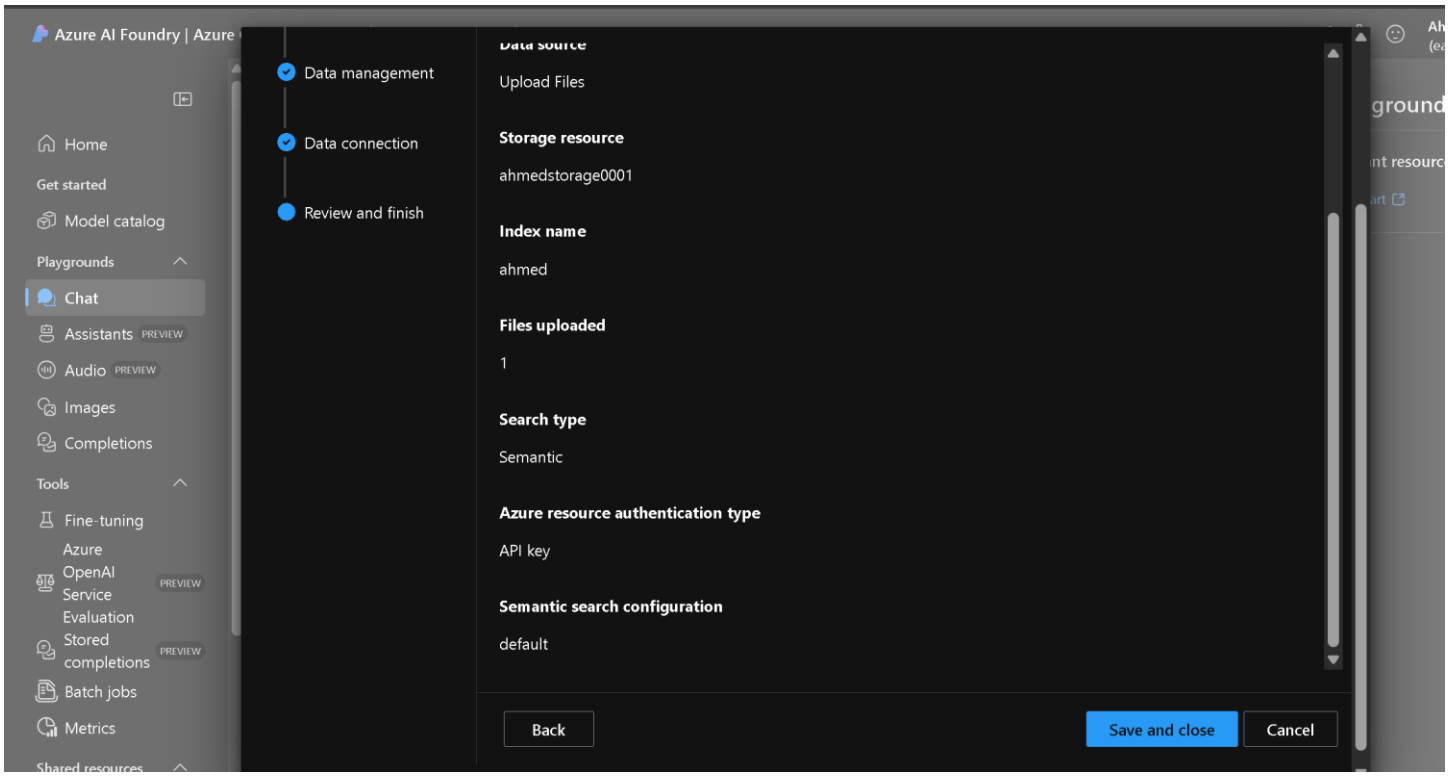


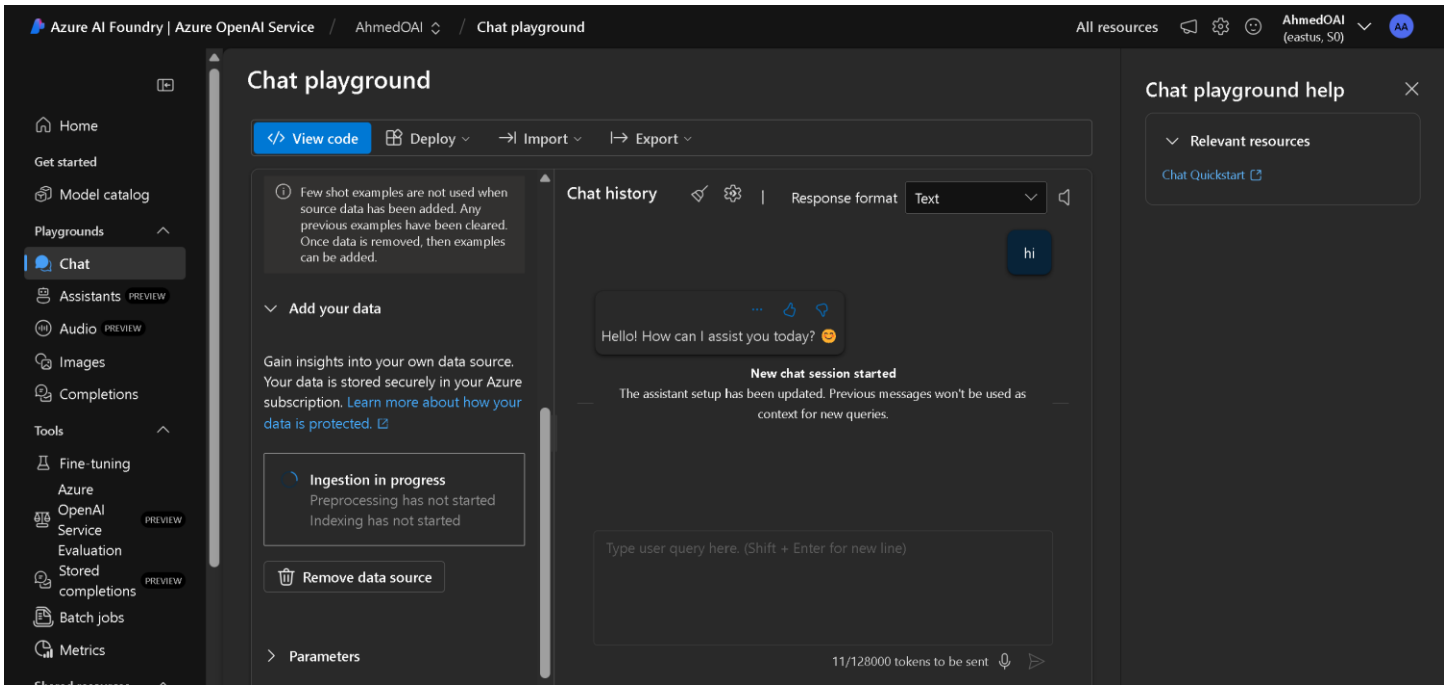

Azure AI Foundry | Azure
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Data source
Upload files
Data management
Data connection
Review and finish
Select or add data source
Your data source is used to ground the generated results with your data. Select an existing data source or create a new data connection with Azure Blob Storage, databases, search, URLs, or local files as the source the grounding data will be built from.
Learn more about data privacy and security in Azure AI.
Select data source *
Upload files (preview)
Subscription *
MSDN533
Select Azure Blob storage resource *
ahmedstorage0001
Create a new Azure Blob storage resource
Cross-origin resource sharing (CORS) is turned on for this resource.
Select Azure AI Search resource (free tier not supported) *
ahmedoai2
Create a new Azure AI Search resource
Enter the index name *
ahmed
Using Azure AI Search will incur usage to your account. View Pricing
Add vector search to this search resource.

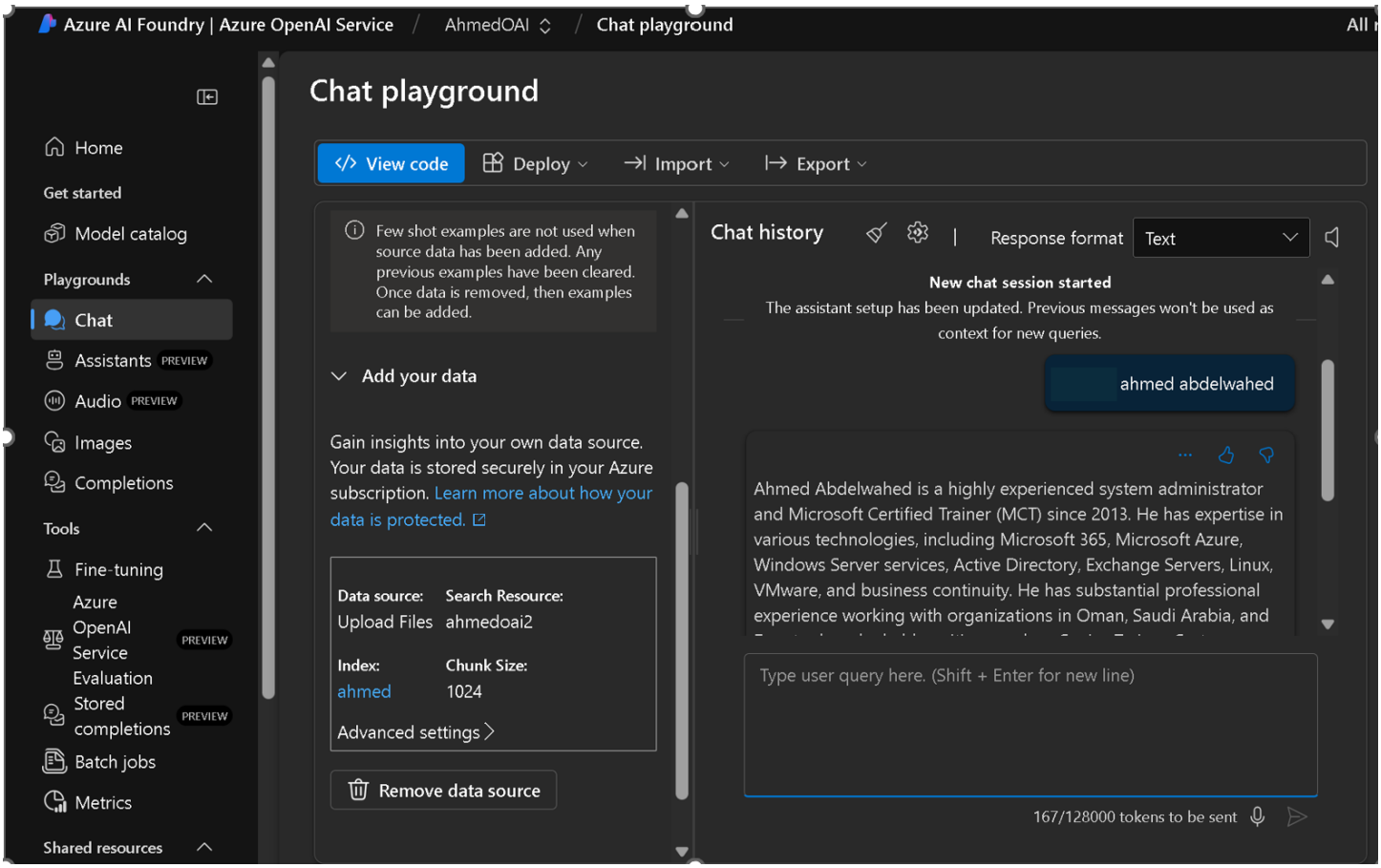

Azure AI Foundry | Azure
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Data source
Upload files
Data management
Data connection
Review and finish
Upload files
Select which files to add. Files will be stored in your Azure Blob Storage and indexed by the Cognitive Search resource created or selected in the previous step.
Learn more about data privacy and security in Azure AI.
Drag and drop
or
Browse for a file
(.txt, .md, .html, .pdf, .docx, .pptx)
16 MB size limit
File name
Type
Size
Status
Ahmed Abdelwahe
PDF
143.82 KB
Uploaded
Your files were successfully uploaded.


Azure AI Foundry | Azure
Add data
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Data source
Upload files
Data management
Data connection
Review and finish
Data management
Set up specific configurations for your data and how the model will respond to requests.
Learn more about data privacy and security in Azure AI.
Search type *
Semantic
Using semantic search will incur usage to your Azure AI Search account. View Pricing
Chunk Size
Chunking is the process of breaking down your documents into smaller segments for search and retrieval. Chunk size is measured in tokens. If the selected chunk size results in low accuracy, re-ingest your data with a different size. Learn more about selecting a chunk size
Select a size
256
512
1024 (default)
1536

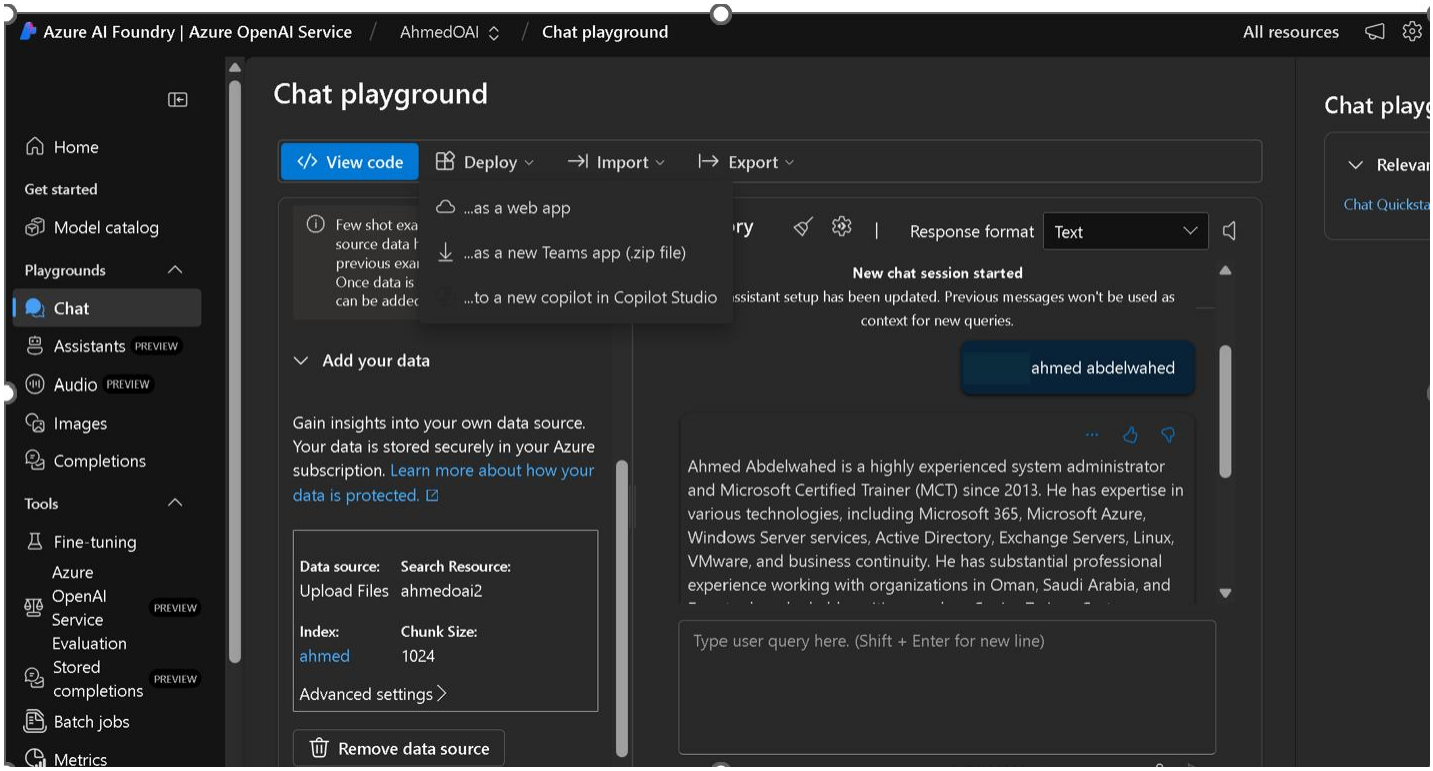

Azure AI Foundry | Azure
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Data source
Upload files
Data management
Data connection
Review and finish
Data connection
Select how your Azure resources connect to each other. Your selection will apply to how Azure AI Search, Azure OpenAI Service, and Azure Blob Storage (if applicable) are connected.
Learn more about data privacy and security in Azure AI.
Azure resource authentication type *
Learn more about selecting an authentication type
System assigned managed identity
API key

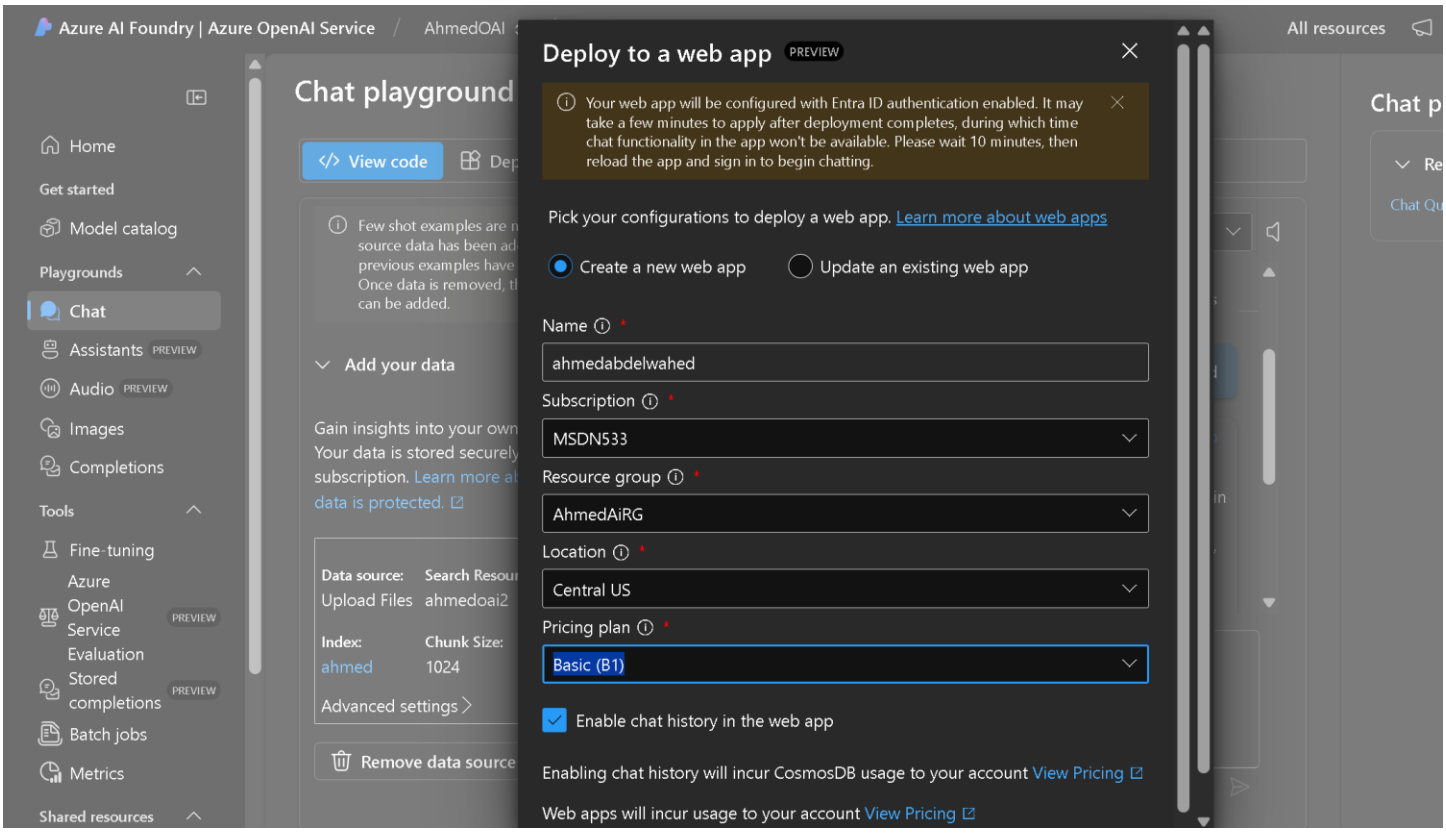

Azure AI Foundry | Azure
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Data management
Data connection
Review and finish
Data source
Upload Files
Storage resource
ahmedstorage0001
Index name
ahmed
Files uploaded
1
Search type
Semantic
Azure resource authentication type
API key
Semantic search configuration
default
Back
Save and close
Cancel

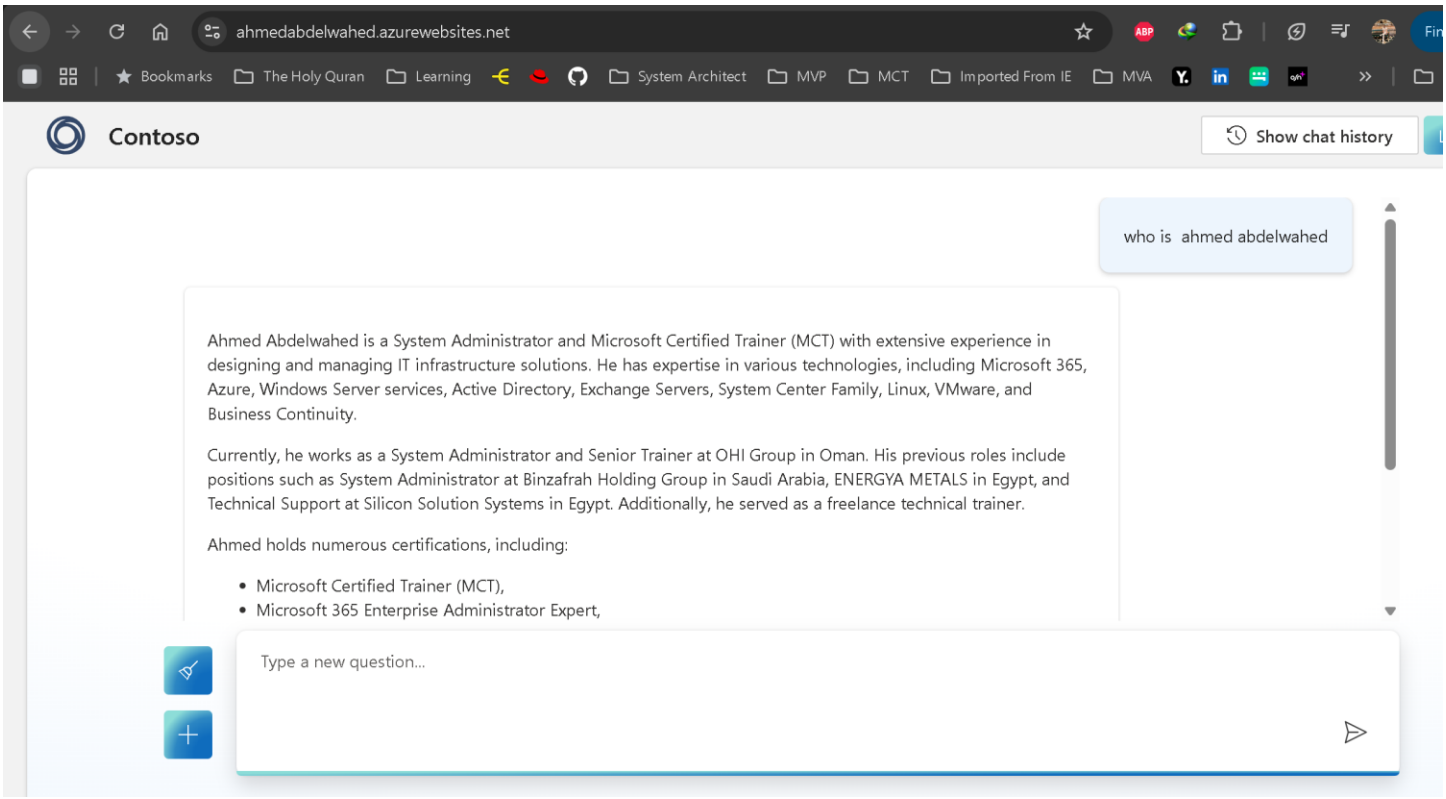

Azure AI Foundry | Azure OpenAI Service
AhmedOAI
Chat playground
All resources
AhmedOAI
(eastus, S0)
Chat playground
View code
Deploy
Import
Export
Few shot examples are not used when source data has been added. Any previous examples have been cleared. Once data is removed, then examples can be added.
Add your data
Gain insights into your own data source. Your data is stored securely in your Azure subscription. Learn more about how your data is protected.
Ingestion in progress
Preprocessing has not started
Indexing has not started
Remove data source
Parameters
Chat history
Response format
Text
hi
Hello! How can I assist you today?
New chat session started
The assistant setup has been updated. Previous messages won't be used as context for new queries.
Type user query here. (Shift + Enter for new line)
11/128000 tokens to be sent
Chat playground help
Relevant resources
Chat Quickstart



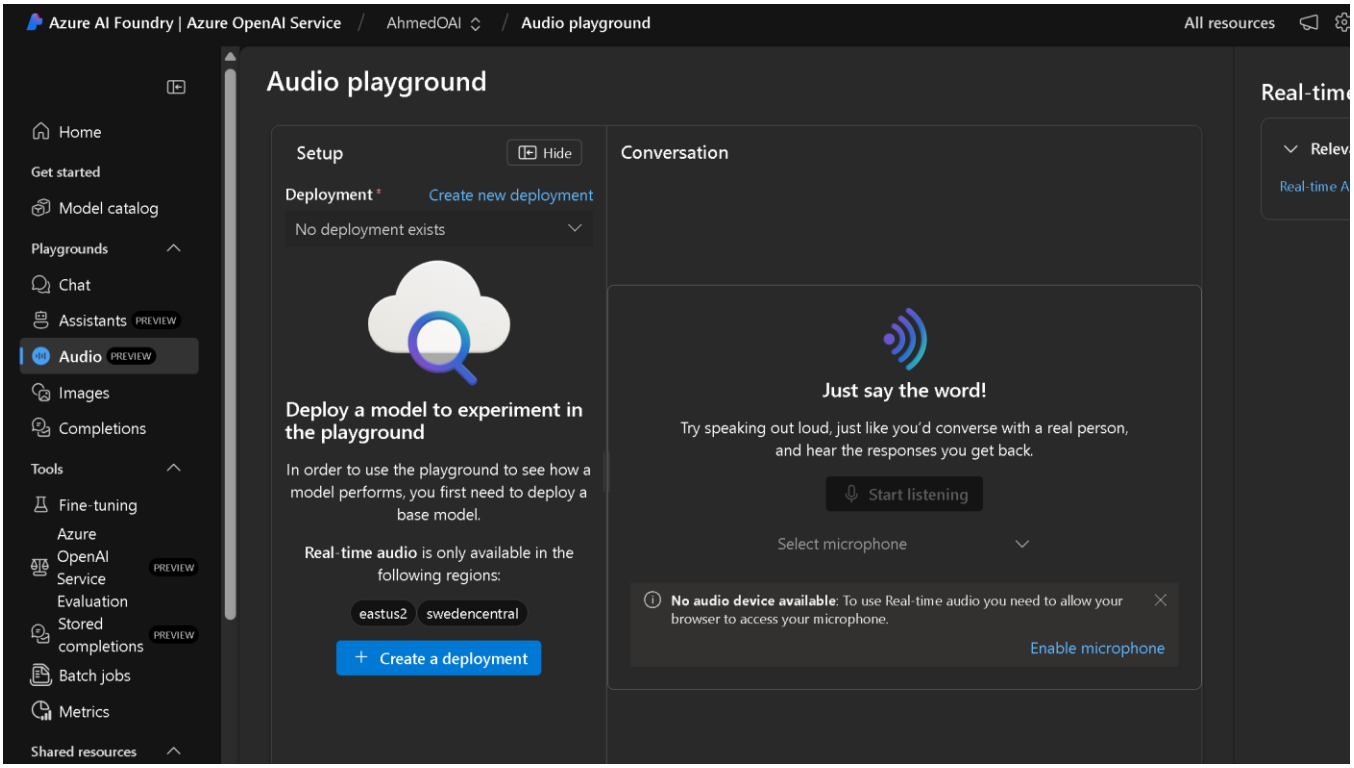
Deploy web app for your Azure OpenAI

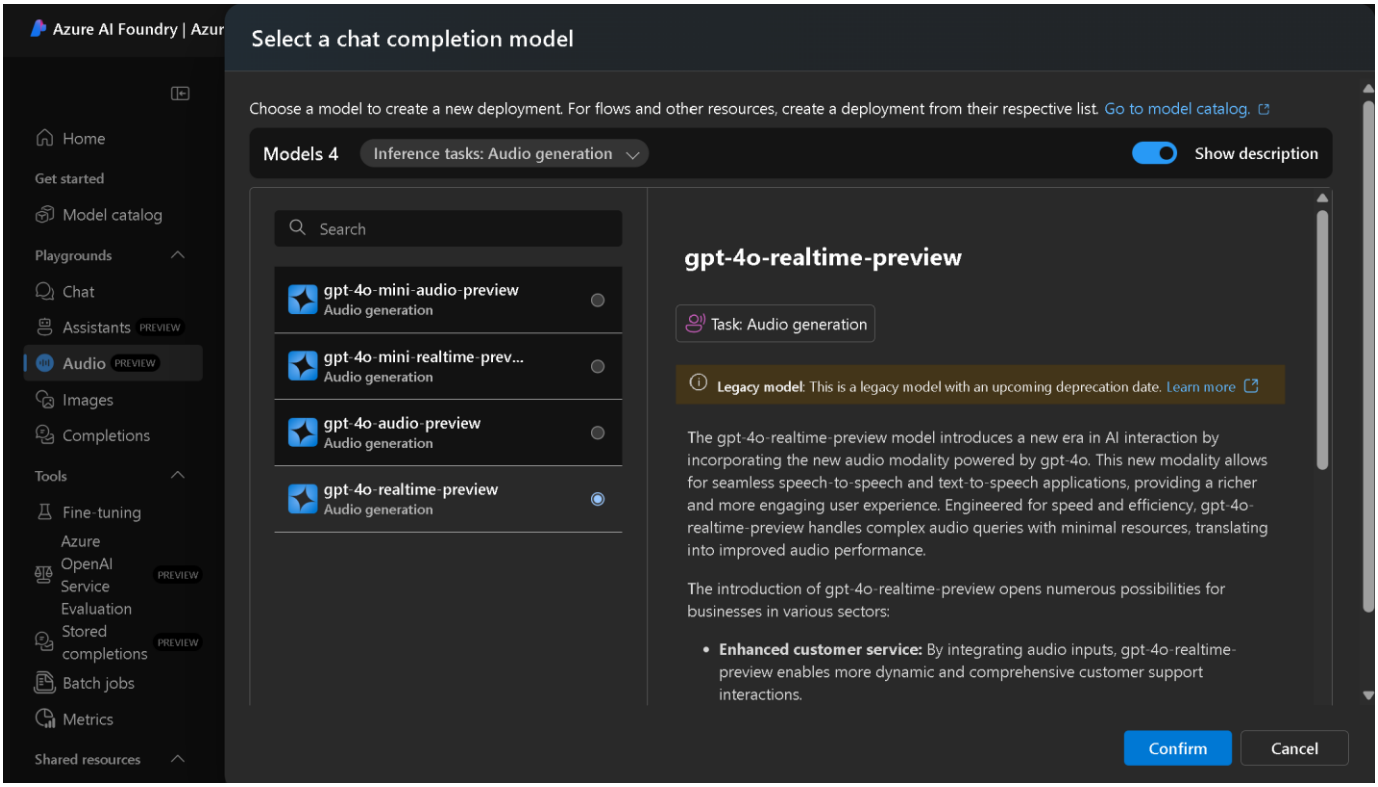


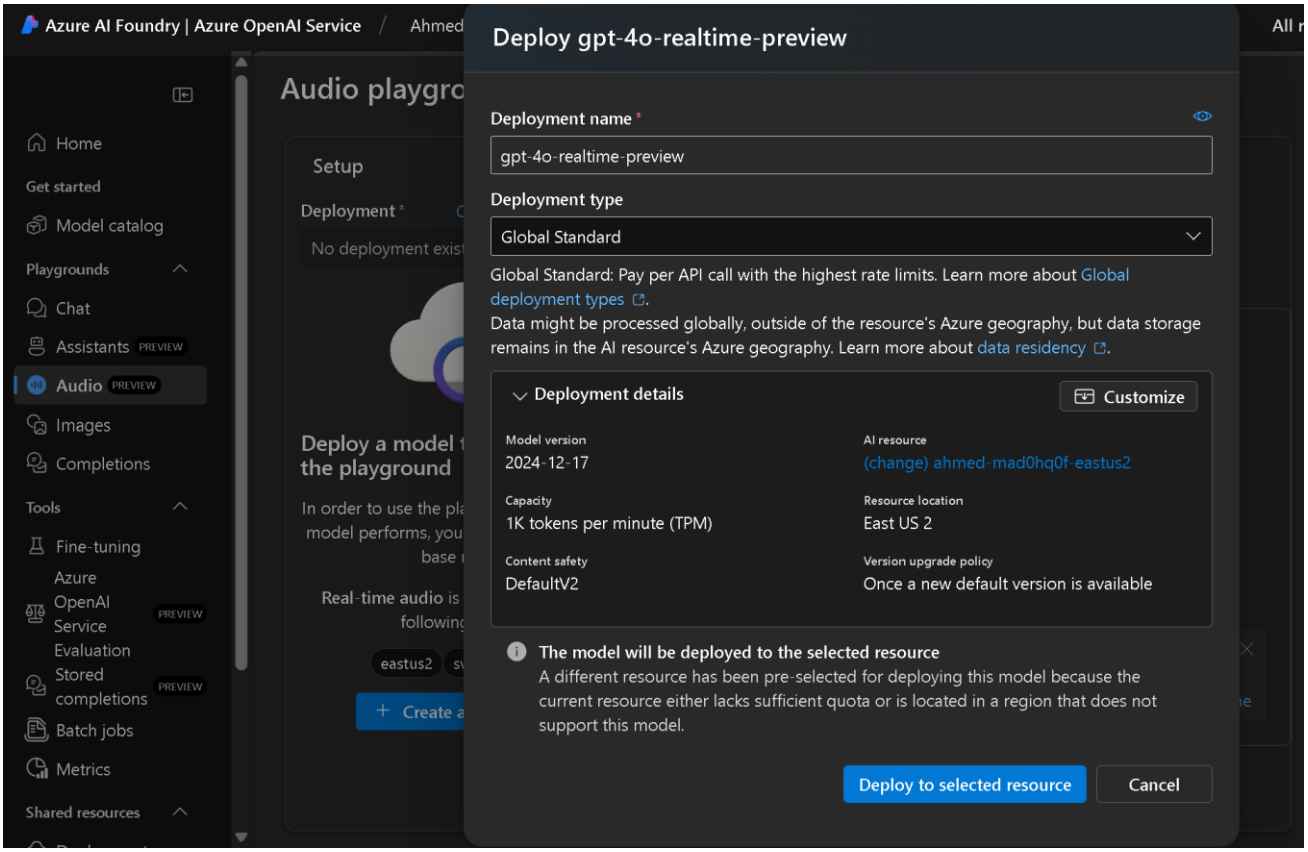

Azure AI Foundry | Azure OpenAI Service
AhmedOAI
All resources
Chat playground
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
View code
Add your data
Remove data source
Advanced settings
Deploy to a web app PREVIEW
Your web app will be configured with Entra ID authentication enabled. It may take a few minutes to apply after deployment completes, during which time chat functionality in the app won't be available. Please wait 10 minutes, then reload the app and sign in to begin chatting.
Pick your configurations to deploy a web app. Learn more about web apps
Create a new web app
Update an existing web app
Name
ahmedabdelwahed
Subscription
MSDN533
Resource group
AhmedAiRG
Location
Central US
Pricing plan
Basic (B1)
Enable chat history in the web app
Enabling chat history will incur CosmosDB usage to your account View Pricing
Web apps will incur usage to your account View Pricing

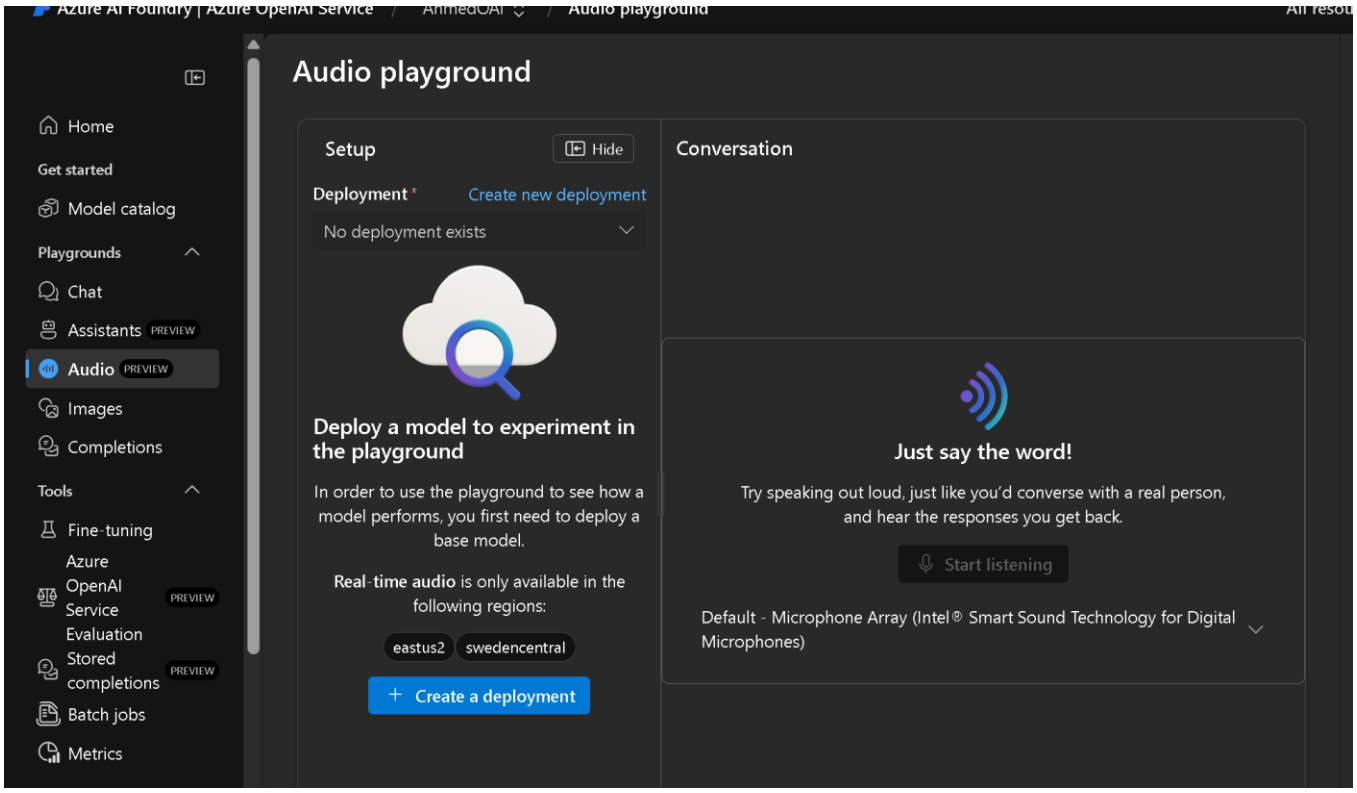

ahmedabdelwahed.azurewebsites.net
Contoso
Show chat history
who is ahmed abdelwahed
Ahmed Abdelwahed is a System Administrator and Microsoft Certified Trainer (MCT) with extensive experience in designing and managing IT infrastructure solutions. He has expertise in various technologies, including Microsoft 365, Azure, Windows Server services, Active Directory, Exchange Servers, System Center Family, Linux, VMware, and Business Continuity.
Currently, he works as a System Administrator and Senior Trainer at OHI Group in Oman. His previous roles include positions such as System Administrator at Binzafrah Holding Group in Saudi Arabia, ENERGYA METALS in Egypt, and Technical Support at Silicon Solution Systems in Egypt. Additionally, he served as a freelance technical trainer.
Ahmed holds numerous certifications, including:
Microsoft Certified Trainer (MCT),
Microsoft 365 Enterprise Administrator Expert,
Type a new question...

Audio

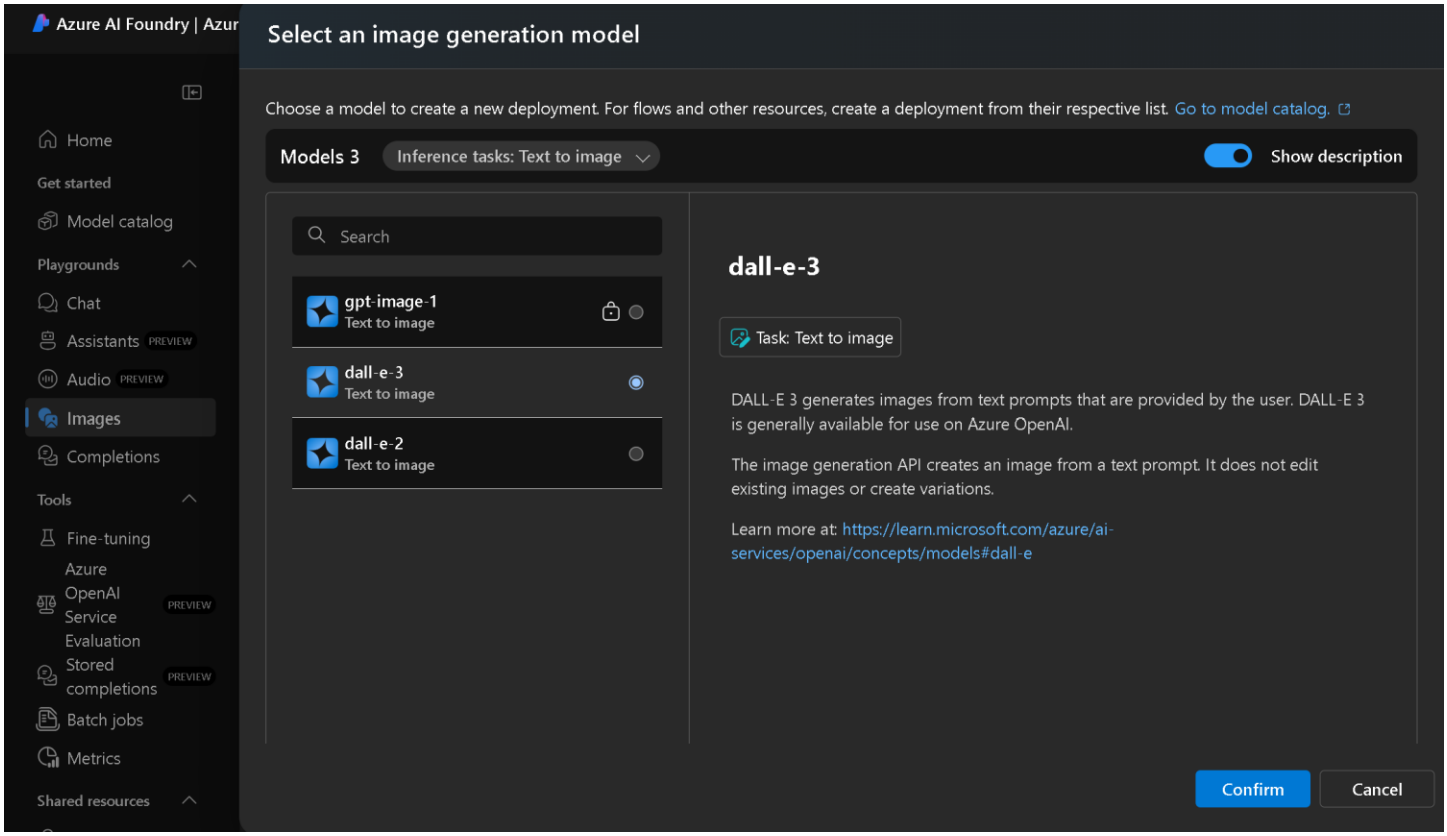

Azure AI Foundry | Azure OpenAI Service
AhmedOAI
Audio playground
All resources
Audio playground
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Setup
Hide
Deployment *
Create new deployment
No deployment exists
Deploy a model to experiment in the playground
In order to use the playground to see how a model performs, you first need to deploy a base model.
Real-time audio is only available in the following regions:
eastus2
swedencentral
Create a deployment
Conversation
Just say the word!
Try speaking out loud, just like you'd converse with a real person, and hear the responses you get back.
Start listening
Select microphone
No audio device available: To use Real-time audio you need to allow your browser to access your microphone.
Enable microphone
Real-time

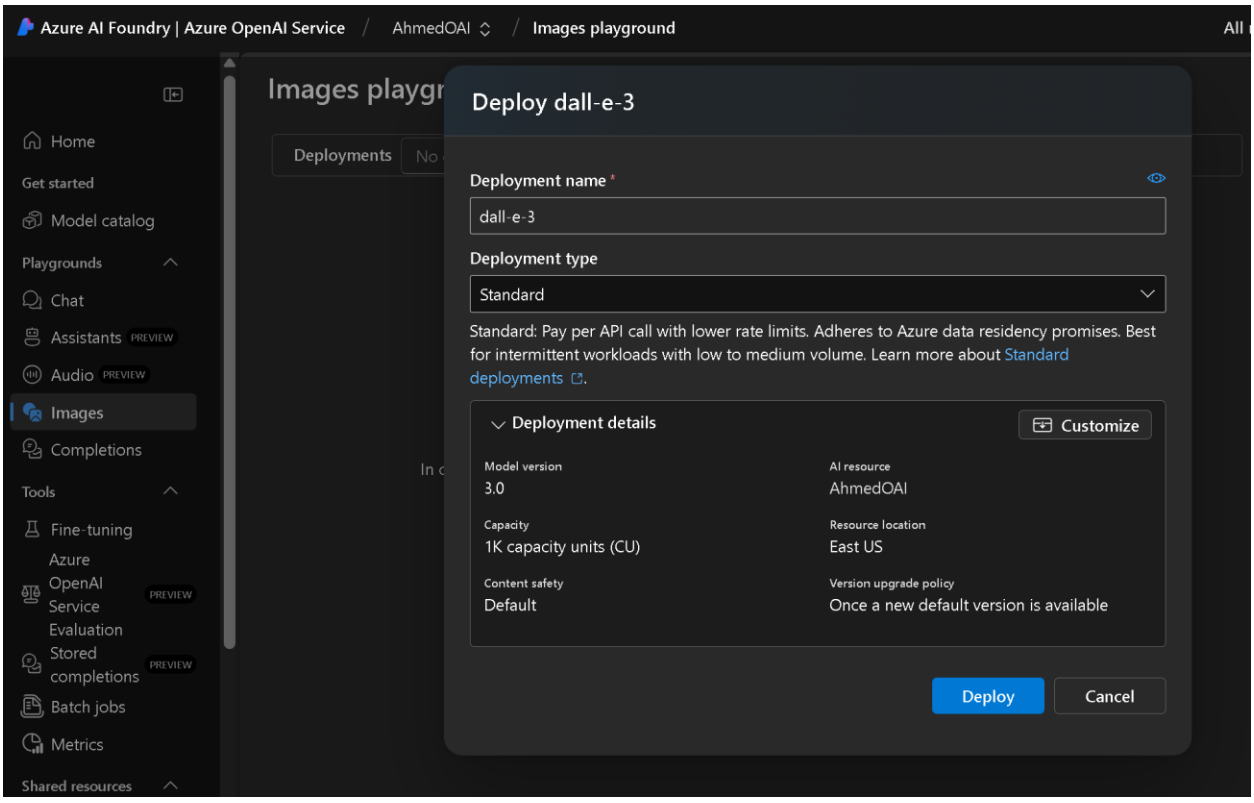

Azure AI Foundry | Azur
Select a chat completion model
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Choose a model to create a new deployment. For flows and other resources, create a deployment from their respective list. Go to model catalog.
Models 4
Inference tasks: Audio generation
Show description
Search
gpt-4o-mini-audio-preview
Audio generation
gpt-4o-mini-realtime-prev...
Audio generation
gpt-4o-audio-preview
Audio generation
gpt-4o-realtime-preview
Audio generation
gpt-4o-realtime-preview
Task: Audio generation
Legacy model: This is a legacy model with an upcoming deprecation date. Learn more
The gpt-4o-realtime-preview model introduces a new era in AI interaction by incorporating the new audio modality powered by gpt-4o. This new modality allows for seamless speech-to-speech and text-to-speech applications, providing a richer and more engaging user experience. Engineered for speed and efficiency, gpt-4o-realtime-preview handles complex audio queries with minimal resources, translating into improved audio performance.
The introduction of gpt-4o-realtime-preview opens numerous possibilities for businesses in various sectors:
Enhanced customer service: By integrating audio inputs, gpt-4o-realtime-preview enables more dynamic and comprehensive customer support interactions.
Confirm
Cancel

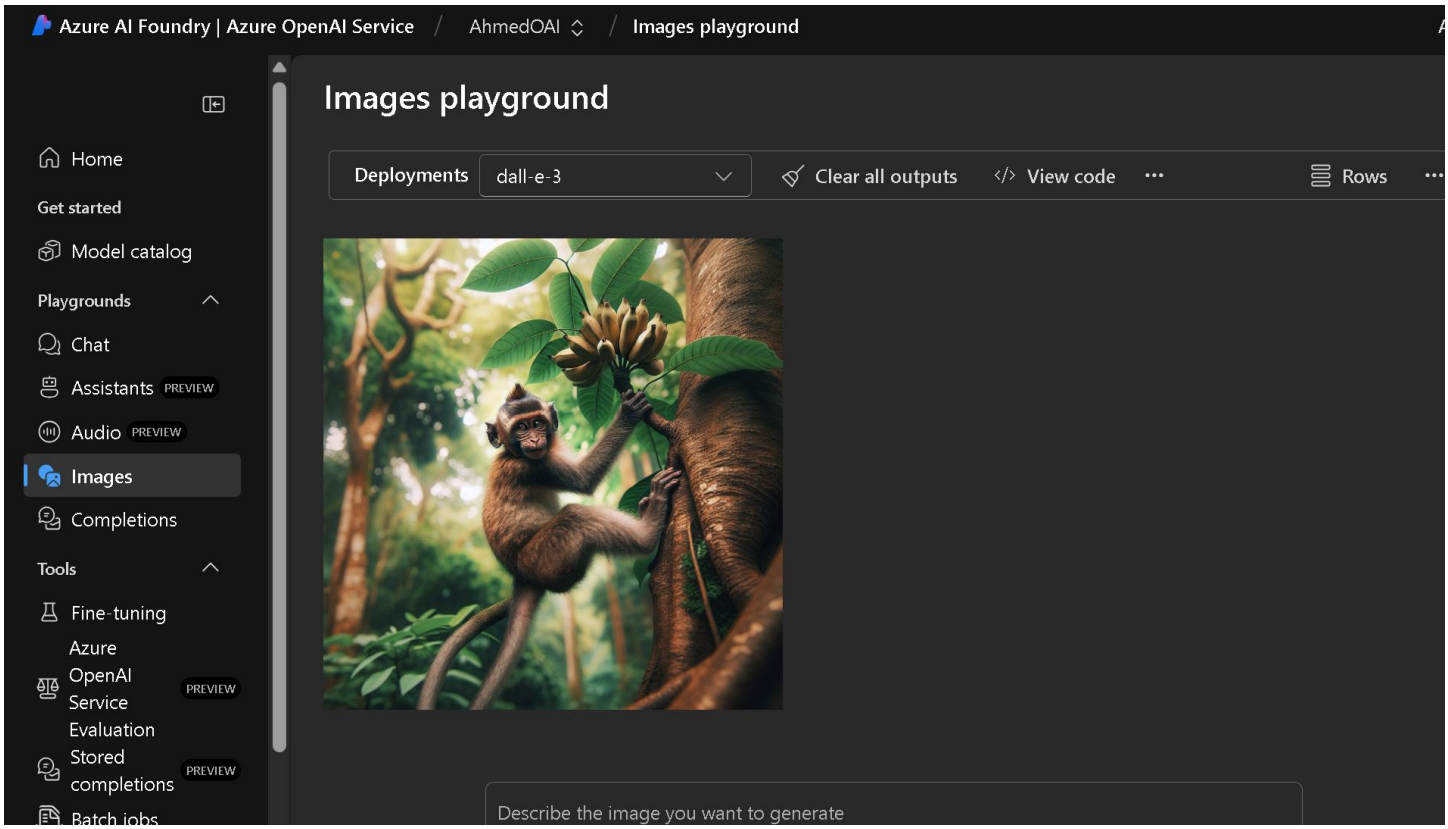

Azure AI Foundry | Azure OpenAI Service
Deploy gpt-4o-realtime-preview
Deployment name *
gpt-4o-realtime-preview
Deployment type
Global Standard
Global Standard: Pay per API call with the highest rate limits. Learn more about Global deployment types.
Data might be processed globally, outside of the resource's Azure geography, but data storage remains in the AI resource's Azure geography. Learn more about data residency.
Deployment details
Customize
Model version
2024-12-17
AI resource
(change) ahmed-mad0hq0f-eastus2
Capacity
1K tokens per minute (TPM)
Resource location
East US 2
Content safety
DefaultV2
Version upgrade policy
Once a new default version is available
The model will be deployed to the selected resource
A different resource has been pre-selected for deploying this model because the current resource either lacks sufficient quota or is located in a region that does not support this model.
Deploy to selected resource
Cancel
Audio playgro
Setup
Deployment *
No deployment exist
Deploy a model t
the playground
In order to use the pla
model performs, you
base
Real-time audio is
followinc
eastus2
Create a
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service PREVIEW
Evaluation
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources


Audio playground
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service PREVIEW
Evaluation
Stored completions PREVIEW
Batch jobs
Metrics
Setup
Hide
Deployment *
Create new deployment
No deployment exists
Deploy a model to experiment in the playground
In order to use the playground to see how a model performs, you first need to deploy a base model.
Real-time audio is only available in the following regions:
eastus2
swedencentral
Create a deployment
Conversation
Just say the word!
Try speaking out loud, just like you'd converse with a real person, and hear the responses you get back.
Start listening
Default - Microphone Array (Intel® Smart Sound Technology for Digital Microphones)

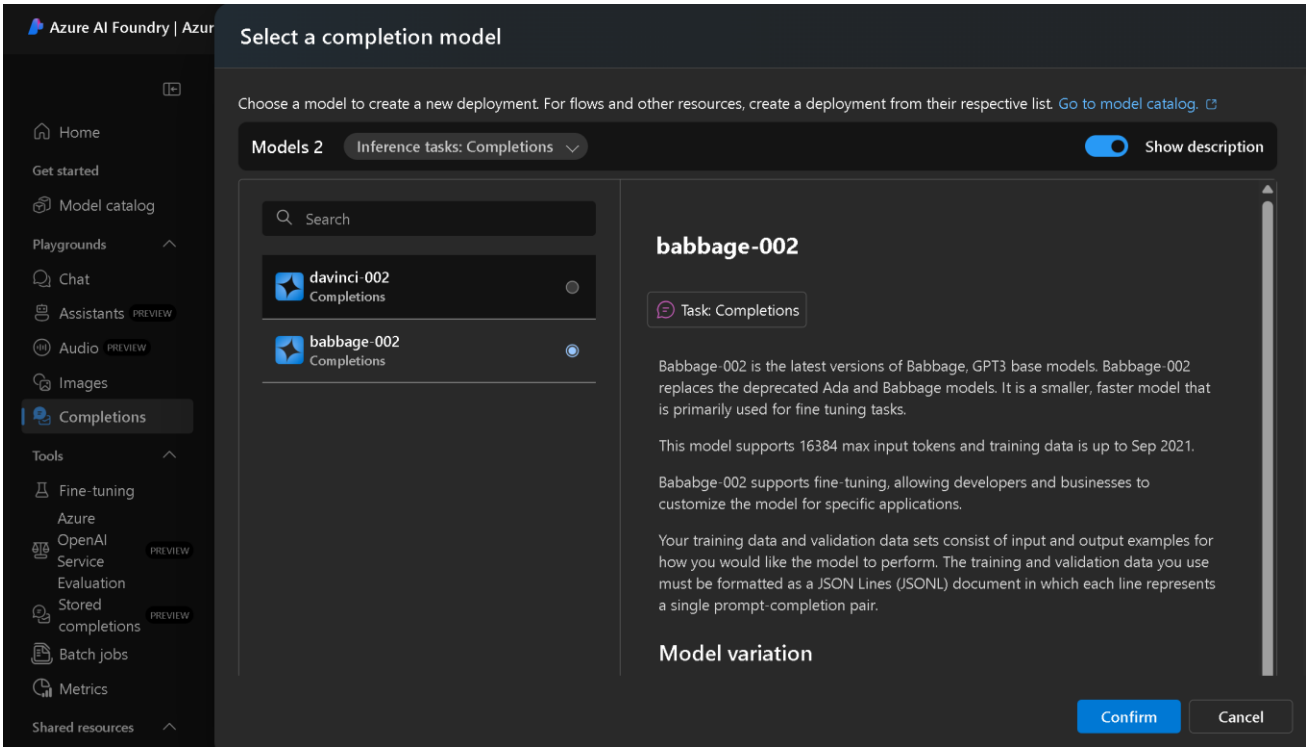
Image

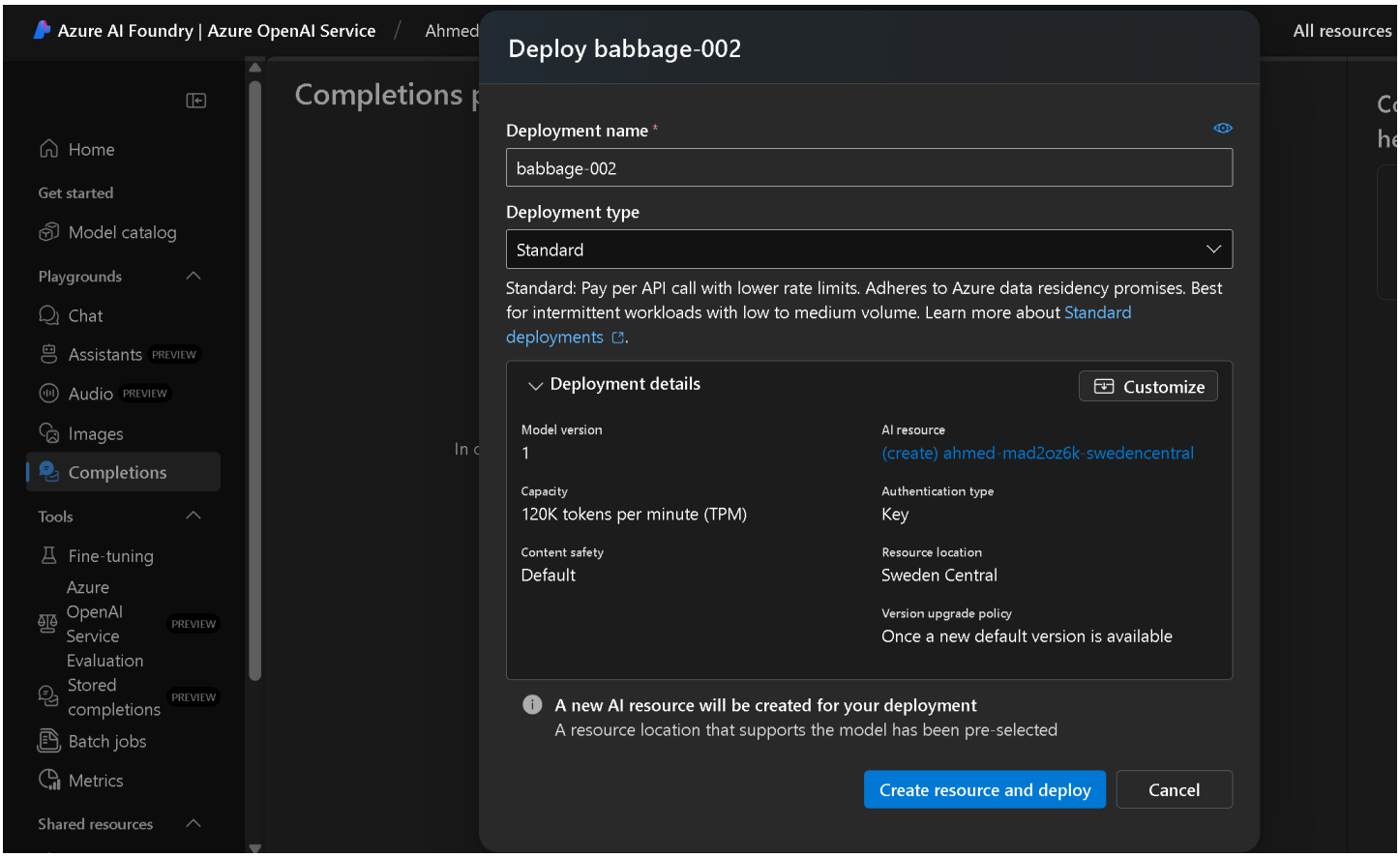

Azure AI Foundry | Azur
Select an image generation model
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Choose a model to create a new deployment. For flows and other resources, create a deployment from their respective list. Go to model catalog.
Models 3
Inference tasks: Text to image
Show description
Search
gpt-image-1
Text to image
dall-e-3
Text to image
dall-e-2
Text to image
dall-e-3
Task: Text to image
DALL-E 3 generates images from text prompts that are provided by the user. DALL-E 3 is generally available for use on Azure OpenAI.
The image generation API creates an image from a text prompt. It does not edit existing images or create variations.
Learn more at: https://learn.microsoft.com/azure/ai-services/openai/concepts/models#dall-e
Confirm
Cancel


Azure AI Foundry | Azure OpenAI Service
AhmedOAI
Images playground
Images playgr
Deployments
Deploy dall-e-3
Deployment name *
dall-e-3
Deployment type
Standard
Standard: Pay per API call with lower rate limits. Adheres to Azure data residency promises. Best for intermittent workloads with low to medium volume. Learn more about Standard deployments.
Deployment details
Customize
Model version
3.0
AI resource
AhmedOAI
Capacity
1K capacity units (CU)
Resource location
East US
Content safety
Default
Version upgrade policy
Once a new default version is available
Deploy
Cancel


Azure AI Foundry | Azure OpenAI Service
AhmedOAI
Images playground
Images playground
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service Evaluation PREVIEW
Stored completions PREVIEW
Batch jobs
Deployments
dall-e-3
Clear all outputs
View code
Rows
Describe the image you want to generate

Completion


Azure AI Foundry | Azur
Select a completion model
Choose a model to create a new deployment. For flows and other resources, create a deployment from their respective list. Go to model catalog.
Models 2
Inference tasks: Completions
Show description
Search
davinci-002
Completions
babbage-002
Completions
babbage-002
Task: Completions
Babbage-002 is the latest versions of Babbage, GPT3 base models. Babbage-002 replaces the deprecated Ada and Babbage models. It is a smaller, faster model that is primarily used for fine tuning tasks.
This model supports 16384 max input tokens and training data is up to Sep 2021.
Bababge-002 supports fine-tuning, allowing developers and businesses to customize the model for specific applications.
Your training data and validation data sets consist of input and output examples for how you would like the model to perform. The training and validation data you use must be formatted as a JSON Lines (JSONL) document in which each line represents a single prompt-completion pair.
Model variation
Confirm
Cancel


Azure AI Foundry | Azure OpenAI Service
Ahmed
All resources
Home
Get started
Model catalog
Playgrounds
Chat
Assistants PREVIEW
Audio PREVIEW
Images
Completions
Tools
Fine-tuning
Azure OpenAI Service PREVIEW
Evaluation
Stored completions PREVIEW
Batch jobs
Metrics
Shared resources
Completions p
Deploy babbage-002
Deployment name *
babbage-002
Deployment type
Standard
Standard: Pay per API call with lower rate limits. Adheres to Azure data residency promises. Best for intermittent workloads with low to medium volume. Learn more about Standard deployments.
Deployment details
Customize
Model version
1
AI resource
(create) ahmed-mad2oz6k-swedencentral
Capacity
120K tokens per minute (TPM)
Authentication type
Key
Content safety
Default
Resource location
Sweden Central
Version upgrade policy
Once a new default version is available
A new AI resource will be created for your deployment
A resource location that supports the model has been pre-selected
Create resource and deploy
Cancel