Whisper Project

For **on-premises**:

Prerequistes: Python

Add Python311 full control from permissions.

pip install -U openai-whisper

choco install ffmpeg

pip install setuptools-rust

On-Premises: //very slow!!!

whisper kan11.mp3 --language Hebrew --task translate

Cloud:

curl <https://api.openai.com/v1/audio/transcriptions>   -H "Authorization: Bearer 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"   -H "Content-Type: multipart/form-data"   -F file="@./kan11.mp3" -F model="whisper-1"

curl <https://api.openai.com/v1/audio/transcriptions> -H "Authorization: Bearer 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" -H "Content-Type: multipart/form-data" -F file="@./kan11.mp3" -F model="whisper-1" -F task=translate -F language=~~Hebrew~~

curl <https://api.openai.com/v1/audio/transcriptions> -H "Authorization: Bearer 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" -H "Content-Type: multipart/form-data" -F file="@./kan11.mp3" -F model="whisper-1" -F task=translate -F language=**he** //Language parameter must be specified in ISO-639-1 format.

curl <https://api.openai.com/v1/audio/transcriptions> -H "Authorization: Bearer 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" -H "Content-Type: multipart/form-data" -F file="@./kan11.mp3" -F model="whisper-1" -F task=translate -F language=**fr** //although the text is in Hebrew, a French will be returned

**chcp 65001**

{"text":"ולא רק זאת, אלא שהמוסד ואנשים מטעמו מפיצים את הסרטון הזה שמתעד מה שהם אומרים חלק מהחקירה שלו כנראה אגב בשלבים מאוד מאוחרים של החקירה עצמה וזו אגב לא פעם הראשונה שהמוסד נוקט בשיטה כזו של חקירה בטלפון רחוק, בסיחות וידאו מאובטחות גם במקרה של עוד מתנקש עירני רסולי, גם הוא נחקר כך על ידי שירותי הביטחון הישראלים גם הוא ביקש להתנקש, אז היה בדיפלומט ישראלי על אדמת טורקיה, במקרה הזה מדובר באיש עסקים על אדמת כפריסין והמסר ברור גם מהאוטינג וגם מהשליחת הזרוע הרוקד מאוד הזו עד לאיראן של ישראל ושל המוסד אין לכם אנשי משמרות המהפכה ושלוחיהם חסינות גם לא על אדמת איראן גילית תודה רבה לך ומהנקודה הזאת אנחנו נעבור אליך רוי שרון פרשננו הצבאי כי כן מדובר בהחלטה חריגה לחשוף פעילות כזאת על אדמת איראן וגם המסר כאן למשמרות המהפכה הוא מאוד ברור קודם כל הניסיון האיראני להרוג ישראלים בחול, אנחנו רואים שאיראן לא מפסיקה לנסות לפגוע בישראלים או ביהודים ברחבי העולם ראינו את זה ביוון"}

**whisper kan11.mp3 --language Hebrew --task translate**

[00:00.000 --> 00:04.000] in Iran, and not only that, but that the Mossad and the people of Amor

[00:04.000 --> 00:08.000] are making this video that shows what they are saying,

[00:08.000 --> 00:10.000] a part of his investigation.

[00:10.000 --> 00:13.000] It looks like it's very late for the investigation itself,

[00:13.000 --> 00:17.000] and it's not the first time that the Mossad is investigating

[00:17.000 --> 00:20.000] such a situation of investigation on a cell phone,

[00:20.000 --> 00:24.000] in video conversations, and also in the case of

[00:24.000 --> 00:27.000] another Iranian-Russian spy,

[00:27.000 --> 00:31.000] he is also so familiar with the Israeli security services,

[00:31.000 --> 00:35.000] he is also in contact with the Israeli diplomat on the Turkish side,

[00:35.000 --> 00:39.000] in this case he speaks with the people who are talking about the Syrian side,

[00:39.000 --> 00:41.000] and the clear message is also from the outing,

[00:41.000 --> 00:46.000] and also from the help of this very serious case

[00:46.000 --> 00:49.000] to Iran, of Israel and of the Mossad,

[00:49.000 --> 00:51.000] there are no people who support the investigation,

[00:51.000 --> 00:54.000] and they have no doubt about the Iranian side.

[00:54.000 --> 00:56.000] Thank you very much for this point,

[00:56.000 --> 00:58.000] we will go to you, Mr. Sharon,

[00:58.000 --> 01:00.000] our team member, because yes,

[01:00.000 --> 01:04.000] we are talking about the decision to cancel such an investigation on the Iranian side,

[01:04.000 --> 01:06.000] and also the Mossad here, for the security services,

[01:06.000 --> 01:08.000] it is very clear.

[01:08.000 --> 01:10.000] First of all, the Iranian side,

[01:10.000 --> 01:11.000] to the Israeli side outside,

[01:11.000 --> 01:14.000] we see that Iran is not enough to try to attack

[01:14.000 --> 01:17.000] the Israelis or the Jews around the world,

[01:17.000 --> 01:19.000] we saw it in vain,

Azure Account ID:  
{e9b8abeb-432a-429f-8060-6b4e098688a5}

**AZURE**

<https://api.videoindexer.ai> //Azure AI

<https://www.videoindexer.ai/media/library> //this one works!

<https://api.videoindexer.ai/auth/westus2/Accounts?generateAccessTokens=true&allowEdit=true>

id= {e9b8abeb-432a-429f-8060-6b4e098688a5}

url = https://www.videoindexer.ai/

name= raphir-e9b8ab

location = westus2

accessToken= //Needs to be retrieved!!!

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9..esb4HMSmzG0e1zQ\_ATYFSuvvG-5Ao5JnqEL8jUfdJ8i-J43F4G67NDmg0QMRFrIYFUQnOywUz-kGOe1ZLI7Um7ZMDMF6Oo-9jajJRo8Klm-pl\_EhKRpERBiD-UBpnnfyEeLLtmQ8fK6S4f9dPKfQUnw9VqcQZbC\_DGc54cMQokE6xEcngyOCI7iV6qLV00XHewJ7MHtvyIv\_q0iutU2WmRjiFXDrSUWMfNiTGyLpuYDVTgGBSpcEUa5hS4IrReOwMIBf\_TrYrqE19Ar4evKcWp9YrpKX21YgQkKqw3rw-8Gxe\_8Ycz4X9u5NazJi2Q4tRN9NsfcfUMS3pyliNleLIQ

queryParams =

**accessToken**=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9..esb4HMSmzG0e1zQ\_ATYFSuvvG-5Ao5JnqEL8jUfdJ8i-J43F4G67NDmg0QMRFrIYFUQnOywUz-kGOe1ZLI7Um7ZMDMF6Oo-9jajJRo8Klm-pl\_EhKRpERBiD-UBpnnfyEeLLtmQ8fK6S4f9dPKfQUnw9VqcQZbC\_DGc54cMQokE6xEcngyOCI7iV6qLV00XHewJ7MHtvyIv\_q0iutU2WmRjiFXDrSUWMfNiTGyLpuYDVTgGBSpcEUa5hS4IrReOwMIBf\_TrYrqE19Ar4evKcWp9YrpKX21YgQkKqw3rw-8Gxe\_8Ycz4X9u5NazJi2Q4tRN9NsfcfUMS3pyliNleLIQ&**name**=channel06\_2023\_08\_07\_15\_45\_00638270269491885976&**description**=Actus+Uploaded&**indexingPreset**=AudioOnly&**language**=English&**privacy**=private&**partition**=partition

audioPath= c:\actus\_temp\workflow\channel06\_2023\_08\_07\_15\_45\_00638270269491885976.mp3

65 lines in OpenAI

100:00:00,000 --> 00:00:13,000It's the Women's World Cup with France 24.200:00:13,000 --> 00:00:17,840Defying the threat of military intervention by regional bloc ECOWAS, Niger's ruling junta300:00:17,840 --> 00:00:24,080says its armed forces stand ready after the deadline passes to reinstate the deposed president.400:00:24,080 --> 00:00:30,120Two leaders have closed the country's airspace until further notice.500:00:30,120 --> 00:00:34,920Jailed Senegalese opposition leader Ousmane Sanco is admitted to hospital for emergency600:00:34,920 --> 00:00:35,920care.700:00:35,920 --> 00:00:40,360He's been on hunger strike since late last month following a conviction his supporters800:00:40,360 --> 00:00:44,960say was politically motivated.900:00:44,960 --> 00:00:49,640And women and children are among the victims following two shipwrecks in stormy seas off1000:00:49,640 --> 00:00:52,120the Italian island of Lampedusa.1100:00:52,120 --> 00:00:56,760Thirty people are still unaccounted for.1200:00:56,760 --> 00:00:57,760Welcome to France 24.1300:00:57,760 --> 00:00:58,760I'm Sharon Gaffney.1400:00:58,760 --> 00:00:59,760Good to have you with us.1500:00:59,760 --> 00:01:05,000Well, first, this hour, Niger has closed its airspace after defying a deadline imposed1600:01:05,000 --> 00:01:09,520by ECOWAS to reinstate the democratically elected president.1700:01:09,520 --> 00:01:14,480The West African regional bloc issued the ultimatum a week ago demanding the generals1800:01:14,480 --> 00:01:19,680relinquish power by midnight on Sunday or face possible military action.1900:01:19,680 --> 00:01:24,880Two leaders say any attempt to violate the country's airspace would be met with an energetic2000:01:24,880 --> 00:01:26,960and immediate response.2100:01:26,960 --> 00:01:31,800The ECOWAS threat of military intervention appears to be revealing differences of opinion2200:01:31,800 --> 00:01:37,880among the bloc's 15 member states about how to proceed, as James Vazina has been finding2300:01:37,880 --> 00:01:40,280out.2400:01:40,280 --> 00:01:45,400Closed since Sunday night, Niger's airspace remains empty as not a single aeroplane can2500:01:45,400 --> 00:01:49,360be seen here flying over the country.2600:01:49,360 --> 00:01:54,480As it ignores ECOWAS' deadline to reinstate President Mohamed Bazoum, the junta that's2700:01:54,480 --> 00:02:01,560taken over the West African country accuses the group of preparing to intervene.2800:02:01,560 --> 00:02:06,360Information currently in our possession indicates that the forces of a foreign power are ready2900:02:06,360 --> 00:02:13,120to attack Niger and its people in coordination with ECOWAS as well as armed terrorist groups.3000:02:13,120 --> 00:02:17,800So far, the Ivory Coast, Senegal and Nigeria have all said that they would join in to drive3100:02:17,800 --> 00:02:21,160out the junta should the military intervention go ahead.3200:02:21,160 --> 00:02:26,000Chad is against this, but Mali and Burkina Faso, which are both ruled by juntas, have3300:02:26,000 --> 00:02:30,600said that they would come to Niger's defence.3400:02:30,600 --> 00:02:35,320Mali and Burkina Faso have enough on their plate with the jihadists in their countries.3500:02:35,320 --> 00:02:38,320They don't have enough forces to fight in Niger.3600:02:38,320 --> 00:02:43,080So the military strength weighs quite obviously in ECOWAS' favour.3700:02:43,080 --> 00:02:45,680But this is a risk for the group's credibility.3800:02:45,680 --> 00:02:53,200It was quite weak with Mali and Burkina Faso, so it must show its strength this time."3900:02:53,200 --> 00:02:58,120Its northern neighbour, Algeria, which shares a 1,000-kilometre border with Niger, is concerned4000:02:58,120 --> 00:03:03,260that an intervention could cause further instability in the Sahel region and is against the use4100:03:03,260 --> 00:03:06,280of force.4200:03:06,280 --> 00:03:11,000As the anti-French sentiment continues to grow in the region, Niger's former coloniser4300:03:11,000 --> 00:03:16,360remains supportive of ECOWAS, hoping that the president will be reinstated soon.4400:03:16,360 --> 00:03:21,840For now though, France has no plans to deploy its 1,500 soldiers, currently stationed on4500:03:21,840 --> 00:03:26,000Nigerian territory.4600:03:26,000 --> 00:03:30,440While Burkina Faso and Mali are sending a joint delegation to Niger, the junta-ruled4700:03:30,440 --> 00:03:34,320nations are currently suspended from regional bloc ECOWAS.4800:03:34,320 --> 00:03:39,480France has announced it's suspending development aid and budgetary assistance to the two countries4900:03:39,480 --> 00:03:43,960after they warned that any military intervention in Niger would be considered a declaration5000:03:43,960 --> 00:03:47,360of war.5100:03:47,360 --> 00:03:50,760He's been on hunger strike since shortly after his arrest last month.5200:03:50,760 --> 00:03:55,920Lawyers for Senegalese opposition leader Ousmane Sanko say he's being treated in hospital.5300:03:55,920 --> 00:04:01,200A declared candidate for next year's presidential election, the 49-year-old has been engaged5400:04:01,200 --> 00:04:07,220in a bitter power struggle with authorities since 2021, when he was placed under investigation5500:04:07,220 --> 00:04:09,000for alleged corruption.5600:04:09,000 --> 00:04:14,280In June, he was sentenced to two months in prison, a move that sparked deadly protests.5700:04:14,280 --> 00:04:19,400For more on why Sanko went on hunger strike in protest against his detention, here's our5800:04:19,400 --> 00:04:21,320correspondent Sam Bradbis.5900:04:21,320 --> 00:04:30,480Well, it's worth looking closely at why he was detained, which... so these most recent6000:04:30,480 --> 00:04:34,000charges are pretty varied and numerous.6100:04:34,000 --> 00:04:41,320He's charged with plotting insurrection, undermining state security, criminal conspiracy with a6200:04:41,320 --> 00:04:48,480terrorist organization, among other things, and these are simply the latest charges against6300:04:48,480 --> 00:04:49,480Ousmane Sanko.6400:04:49,480 --> 00:04:54,720Earlier in the year, he had been found guilty of defaming a government minister.6500:04:54,720 --> 00:04:59,240He'd also been found guilty of corrupting the youth after a sex scandal involving a6600:04:59,240 --> 00:05:00,000young beauty.

1

00:00:00,000 --> 00:00:13,000

It's the Women's World Cup with France 24.

2

00:00:13,000 --> 00:00:17,840

Defying the threat of military intervention by regional bloc ECOWAS, Niger's ruling junta

3

00:00:17,840 --> 00:00:24,080

says its armed forces stand ready after the deadline passes to reinstate the deposed president.

4

00:00:24,080 --> 00:00:30,120

Two leaders have closed the country's airspace until further notice.

5

00:00:30,120 --> 00:00:34,920

Jailed Senegalese opposition leader Ousmane Sanco is admitted to hospital for emergency

6

00:00:34,920 --> 00:00:35,920

care.

Compreface

Recognition key = b8ce7b91-dc0e-4c29-9bbf-9192683638c4

Detection key = 63e35816-2e86-421e-88c8-de0fe23450ef

AWS ACCOUNT

User name : gilshalev

Password: 2023gil18aws#G

Access key ID: AKIAU37MYIC7ZL55SJCY

Secret access key: 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

NEW KEY: AKIAU37MYIC7WB7GWV5O

NEW CLI Secret: 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

userid: [raphir@actusdigital.com](mailto:raphir@actusdigital.com)

pwd: raphir

334967095487

//Query ACTUS services

<http://localhost:9890/account/api/service>

whisper **small.en** **--output\_dir .\ --device cuda** --**output\_format vtt** **--threads THREADS**

It depends on the length of your file and what type of hardware you have access to!

When we ran the medium.en Whisper model on the 40-minute "[Cristóbal Valenzuela — The Next Generation of Content Creation and AI](https://www.youtube.com/watch?v=wbonGgk-_Gk)" interview, it took:

* About **6 minutes on GPU**
* About **1.5 hours on CPU**

**MySql Queries**

SELECT \* FROM ai\_analysis a

ORDER BY a.id DESC

LIMIT 2;

STT/TR prefixes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **STT** | **TR** | **Comments** | **STT Check** | **TR Check** |
| **Google** | w | - |  |  |  |
| **OpenAI** | o | c | (only cen)  SPEED (1)  ACCURACY (2) | + | + |
| **Whisper** | s | p | (only pen)  SPEED (2)  ACCURACY (3) | + | + |
| **Azure** | x | k | SPEED (3)  ACCURACY (1) | + | + |
|  |  |  |  |  |  |

1= best

3=worst

For "win-test" ec2 - PWD is: **wcOdcHCz8&-BOXimOK2$f!Y-)G2-&$Zi** (parsed from .pem file) (IP= 35.153.176.153)

For "gpu-server" ec2 - PWD is**: FNRsH$-nGoShXojVG6cJ;xG7va7N-fOm** (parsed from .pem file) (IP= 3-236-7-216)

To create a .net app with .net runtime:

dotnet publish -c Release --self-contained true //for the POC

dotnet publish -c Debug --self-contained true //for the MIcroservice

Install python on windows server 2022

[Download Python | Python.org](https://www.python.org/downloads/)

To expose EC2 Windows Microservice – steps:

1. MicroSVC is listening to <http://localhost:portXXX>
2. Consumer is connecting through <http://PUBLIC-IP-ADDRESS:portXXX>
3. Configure SG of the EC2 with the correct protocols
4. Disable Windows Defender application (FW)
5. Deploy all the relevant software: python, .net run time, whisper etc…

**/NVIDIA / CUDA commands**

nvidia-smi

NVIDIA Control Panel

NVML = (NVIDIA Management Library)

The demo with **clblast** GPU, one need to download and copy **clblast.dll** to the exe directory, or somewhere in path, download from <https://github.com/CNugteren/CLBlast/releases/tag/1.6.0> .

The demo with **cublas** GPU on windows, one need to download <https://developer.nvidia.com/cuda-toolkit> , and only need to install CUBLAS, CUDART runtime. Then copy the dlls from C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.1\bin. The 3 needed dlls are : cublas64\_12.dll , cublasLt64\_12.dll and cudart64\_12.dll

**Use different node versions:**  
nvm use 14.17.3

nvm use 18.10.0

**FR DEMO**

Should compile with X64 !

With old .NET 4.8

Output should probably be sent to …\Bin of the demo apps, because of licensing

**Electron**

Based on: <https://github.com/maximegris/angular-electron>

**npm run electron:build** //to create an exe, based on the OS!

**/v1/chat/completions** endpoint is subject to the model's token limit. The GPT-3 models have token limits that vary depending on the specific model version:

* For the **gpt-3.5-turbo** model, the maximum limit is 4096 tokens.

You can keep track of the token count for your conversation by using OpenAI's **tiktoken** Python library, which helps you count tokens in text without making API calls. The library can help you stay within the model's token limit while composing conversations.

**Azure**:  
<https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&to=es&from=en>

Opnai KEY:

Gil-Insights-key??

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

f?)UX7n3&rGYK4G5U6FMOiPAK66X2Czh

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

WHISPER SUPPORTED LANGUAGES:  
curl https://api.openai.com/v1/engines/whisper/languages  
curl <https://api.openai.com/v1/engines/whisper/languages/en>

curl https://api.openai.com/v1/engines/whisper/languages -H "Authorization: Bearer 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"

curl https://api.openai.com/v1/engines/whisper/languages -H "Authorization: Bearer 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"

curl -X GET "https://api.cognitive.microsofttranslator.com/languages?api-version=3.0" -H "Ocp-Apim-Subscription-Key: f58881feb4df48dea54cc609778e886d" -H "Ocp-Apim-Subscription-Region: eastus"

<https://translation.googleapis.com/language/translate/v2/languages>

//GOOGLE languages

<https://translation.googleapis.com/language/translate/v2/languages>

curl -X GET -H "Authorization: Bearer $(gcloud auth print-access-token)"-H "x-goog-user-project: speechtotext-369512" "https://translation.googleapis.com/language/translate/v2/languages"

curl -X GET -H "x-goog-user-project: speechtotext-369512" "https://translation.googleapis.com/language/translate/v2/languages"

GET https://translation.googleapis.com/v3/projects/speechtotext-369512/locations/global/supportedLanguages

**BUILD MACHINE IP:**

**197684410**

PASSWORD:

**actusmon123**

--------------------

Set License

--------------------

1. Open C:\ActDev\Tools\SystemConfig\SystemConfig.sln

2. Run SystemConfig

3. License->Copy license to clipboard

4. Save Machine code (8217 for example)

5. Open C:\ActDev\Tools\ActusLicenseMgmt\ActusLicenseMgmt.sln

6. Run ActusLicenseMgmt

7. Paste to "Licence Key" field

8. "Check License" button

9. Input new machine code in "Machine Code" field (8217 for example)

10. "Generate License" button

11. Select then copy license to clipboard

12. Return to "System config" window

13. Paste license to "License Key" field

14. Press "Install License" button

How to install build

--------------------

go to the build machine AnyDesk 197684410 actusmon123

go to the c:\actdev\scripts

run

update

build

archive

1.Goto 'corona'

2.from 'corona'

**\\actbuild\build**\9.1.0.13823

3.run upgrade.bat

<https://s3.console.aws.amazon.com/s3/buckets/russiatv?region=us-east-2&prefix=levira/5INTERNATIONAL/2022_03_02/&showversions=false>

https://russiatv.s3.amazonaws.com/ [levira/](https://s3.console.aws.amazon.com/s3/buckets/russiatv?prefix=levira/)[5INTERNATIONAL/](https://s3.console.aws.amazon.com/s3/buckets/russiatv?prefix=levira/5INTERNATIONAL/)2022\_03\_02/

How to download Russiatv bucket files to a local machine?  
**1. Install aws cli to your computer:  
Installer:** Download the Windows installer from the [AWS CLI website](https://aws.amazon.com/cli/).

**2. Run Installer:** Run the downloaded .exe file and follow the installation instructions. Make sure to select the option to add the AWS CLI executable to the system's PATH during installation.

1. **Test that the installation succeeded by running:**

**aws --version**

Configure the aws cli by running:  
aws configure

by entering, your credentials accordingly

|  |  |
| --- | --- |
| Access key ID | Secret access key |
| AKIAU37MYIC7UY5N6ND4  AWS Access Key ID [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*6ND4]: AKIAU37MYIC7UY5N6ND4  AWS Secret Access Key [\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Pbn5]: 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  Default region name [us-east-1]: Enter Default output format [None]: Enter | sCb5buDqRBFxBA1peR82m3ttXVi6TTrGklBiPbn5 |

1. Now you can copy the bucket files to your machine:  
   aws s3 sync s3://russiatv/folder-name /path/to/local/directory

We created 2 folders for your needs – so copy : [levira/](https://s3.console.aws.amazon.com/s3/buckets/russiatv?region=us-east-2&prefix=levira/&showversions=false) and **/logs  
run:**aws s3 sync s3://russiatv/levira /path/to/your/levira/local/directory

aws s3 sync s3://russiatv/logs /path/to/your/logs/local/directory

1568828622 <- GPU MACHINE station ID <-RustDesk: **87649285** Pass: Actusmon123

system\_prompt = "You are a helpful assistant for the company ZyntriQix. **Your task is to correct any spelling discrepancies in the transcribed text**. Make sure that the names of the following products are spelled correctly: ZyntriQix, Digique Plus, CynapseFive, VortiQore V8, EchoNix Array, OrbitalLink Seven, DigiFractal Matrix, PULSE, RAPT, B.R.I.C.K., Q.U.A.R.T.Z., F.L.I.N.T. Only add necessary punctuation such as periods, commas, and capitalization, and use only the context provided."

**GILS** Azure Speech Service keys:  
ApiKey 1: **d366a666c8604c4db8514f8b5ab3bf92**

ApiKey 2: **eca4a4196836494f8099e779b1097c45**

**eastus**

Billing: [**https://eastus.api.cognitive.microsoft.com**](https://eastus.api.cognitive.microsoft.com/)

**NO BLANKS ARE ALLOWED BETWEEN PARAMETER NAMES AND VALUES !!!**

docker run --rm -it -p 5000:5000 --memory 8g --cpus 8 --mount type=bind,src=C:\Temp,target=/output mcr.microsoft.com/azure-cognitive-services/speechservices/**neural-text-to-speech** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

//running diagnostics

docker run --rm mcr.microsoft.com/azure-cognitive-services/diagnostic eula=accept Billing= <https://eastus.api.cognitive.microsoft.com> ApiKey=d366a666c8604c4db8514f8b5ab3bf92

docker run --rm -it -p 5000:5000 --memory 8g --cpus 8 --mount type=bind,src=C:\Temp,target=/output mcr.microsoft.com/azure-cognitive-services/speechservices/**speech-to-text:latest** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

docker pull mcr.microsoft.com/azure-cognitive-services/speechservices/speech-to-text:latest

docker pull docker.io/batchkit/speech-batch-kit:latest

docker run --rm -it -p 5000:5000 --memory 8g --cpus 8 mcr.microsoft.com/azure-cognitive-services/speechservices/**speech-to-text** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

var config = SpeechConfig.FromHost(

new Uri("ws://localhost:5000"));

//Working partially

1. Run the speech container first:  
   docker run --rm -it -p 5000:5000 --memory 8g --cpus 8 --mount type=bind,src=C:\Temp,target=/output mcr.microsoft.com/azure-cognitive-services/speechservices/**speech-to-text:latest** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

(running with batch container doesn't require the mounting?)

1. Run the batch container:

Under Windows:

docker run --network host --rm -it -v C:\BATCH\_FOLDER:/my\_nfs batchkit/speech-batch-kit:latest -config /my\_nfs/config.yaml -input\_folder /my\_nfs/audio\_files -output\_folder /my\_nfs/transcriptions -log\_folder /my\_nfs/logs

//Languages – maybe only 4 are allowed?  
//only POSIX system identifies new files!

docker run --network host --rm -it -v C:\Temp\AzureContainerConfig:/my\_nfs batchkit/speech-batch-kit:latest -config /my\_nfs/config.yaml -input\_folder /my\_nfs/audio\_files -output\_folder /my\_nfs/transcriptions -log\_folder /my\_nfs/logs **--run-mode DAEMON –poll** –language ["en-US", "es-ES","he-IL"," tr-TR"]

Under WSL:

docker run --network host --rm -it -v /mnt/my\_nfs:/my\_nfs batchkit/speech-batch-kit:latest -config /my\_nfs/config.yaml -input\_folder /my\_nfs/audio\_files -output\_folder /my\_nfs/transcriptions -log\_folder /my\_nfs/logs

The container closes when it finished parsing all the audio files

In WSL2:  
[\\wsl$\Ubuntu\mnt\my\_nfs](file:///\\wsl$\Ubuntu\mnt\my_nfs)

docker run --rm -it -p 5000:5000 --memory 8g --cpus 8 mcr.microsoft.com/azure-cognitive-services/speechservices/**speech-to-text** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

**//pay attention to the order of the pre /post order of the parameters**

run --rm -it -p 5000:5000 --memory 8g --cpus 8 mcr.microsoft.com/azure-cognitive-services/speechservices/**custom-speech-to-text** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92 BaseModelLocale=tr-TR MODELID= **dc30ea31-556b-4953-aaa6-a3be79dd2913**

--language en-US fr-FR tr-TR

docker run --network host --rm -it -v /mnt/my\_nfs:/my\_nfs 976451b12e2a -config /my\_nfs/config.yaml -input\_folder /my\_nfs/audio\_files -output\_folder /my\_nfs/transcriptions -log\_folder /my\_nfs/logs

--language en-US

**Accessing the Container's Shell**:

docker exec -it <container\_id> /bin/sh

# or if it uses bash

docker exec -it <container\_id> /bin/bash

copying files from windows to WSL, inside WSL terminal:  
**Pay attention** the **password** for **sudo** is **gilshalev**

**Pay attention** – if running in **WSL** the **transcriptions** will be seen there and not in windows and vice versa!

/mnt/my\_nfs/audio\_files$:

sudo cp /mnt/c/Temp/AzureContainerConfig/audio\_files/\*.wav .

curl -v -X GET "https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/transcriptions/ b7f203bc-f62b-47b1-bb28-a2aaf9bfd255 /files" -H "Ocp-Apim-Subscription-Key: d366a666c8604c4db8514f8b5ab3bf92"

**REAL TIME TRANSCRIPTION AND TRASNLATION USING AZURE CLOUD**

**USING CLI (spx): //Done in the CLOUD!**

spx config @key --set **e285874fa6e044d99773dd54b3ee70f5** //ACTUS KEY

spx config @region --set eastus

//TO USE other formats than wav – we need to install the GSTREAMER!

And then run:  
spx config @key --set **e285874fa6e044d99773dd54b3ee70f5** //ACTUS KEY

spx config @region --set eastus

or have 2 files: region and key

// TRANSLATION   
spx can only translate to one language at a time !!!

spx translate --file C:\Development\MEDIA\katie-and-steve.wav --source en-US --target he-IL --output file C:\Development\MEDIA\Translations\he-katie-and-steve\_translation.txt

spx translate --file C:\Development\MEDIA\kennedy.wav --source en-US --target ru-RU --output file C:\Development\MEDIA\ru-kennedy\_translation.txt

spx translate --file C:\Development\MEDIA\kennedy.wav --source en-US --target 'ru-RU;fr-FR;es-ES' --output file C:\Development\MEDIA\multiple\_kennedy\_translation.txt

//Real-time michrophone translation!

spx translate --microphone --source he-IL --target en-US

//TRANSCRIBING

spx recognize --file C:\Development\MEDIA\CBS.mp3 --format mp3

spx recognize --file C:\Development\MEDIA\aboutSpeechSdk.wav --output file C:\Development\MEDIA\TRANSCRIPTIONS\aboutSpeechSdk.tsv

spx recognize --files @filelist.txt --format any

**//Batch speech to text (speech recognition)**

spx recognize --files C:\Development\MEDIA\\*.wav --output file C:\Development\MEDIA\speech\_output.tsv --threads 10

spx recognize --files C:\Development\MEDIA-FILES\BATCH\_PROCESSING\\*.wav --output file C:\Development\MEDIA-FILES\BATCH\_PROCESSING\speech\_output.tsv --threads 10

spx translate --file C:\Development\MEDIA\speechService.wav --source en-US --target 'ru-RU;fr-FR;es-ES' --output file C:\Development\MEDIA\TRANSLATIONS\multiple\_translations.txt //Managed to translate only to one language

spx translate --file C:\Development\MEDIA\speechService.wav --source en-US --target fr-FR --output file C:\Development\MEDIA\TRANSLATIONS\multiple\_translations.txt

spx translate --files C:\Development\MEDIA-FILES\BATCH\_PROCESSING\\*.wav --source en-US --target he-IL --output file C:\Development\MEDIA-FILES\BATCH\_PROCESSING\multiple-translations-output.tsv --threads 10

//Adding **Captions** in different formats

[https://learn.microsoft.com/en-us/azure/ai-services/speech-service/**captioning**-quickstart?tabs=windows%2Cterminal&pivots=programming-language-csharp](https://learn.microsoft.com/en-us/azure/ai-services/speech-service/captioning-quickstart?tabs=windows%2Cterminal&pivots=programming-language-csharp)

spx recognize --file C:\Development\MEDIA-FILES\fileXXX.**mp4** --format any --output **vtt** --output **srt** @caption.defaults

spx recognize --file C:\Development\MEDIA-FILES\fileXXX.**mp4** --format any --output **vtt** file caption.vtt --output **srt** file caption.srt --output each file each.result.tsv --output all file output.result.tsv --output each recognizer recognizing result offset --output each recognizer recognizing duration --output each recognizer recognizing result resultid --output each recognizer recognizing text

//Captioning on .mp3 – look at the **output** **files**: **caption.srt & caption.vtt**

spx recognize --file nbc.mp3 --format any --output vtt --output srt @caption.defaults

curl -v -X GET "https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/models/base" -H "Ocp-Apim-Subscription-Key: d366a666c8604c4db8514f8b5ab3bf92"

curl -v -X GET "https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/models/base?skip=100&top=100" -H "Ocp-Apim-Subscription-Key: b7f203bc-f62b-47b1-bb28-a2aaf9bfd255"

//**To get the models of a specific language:**

docker run --rm -it mcr.microsoft.com/azure-cognitive-services/speechservices/**custom-speech-to-text** **BaseModelLocale=en-US** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

curl -v -X GET "https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/models/base" -H "Ocp-Apim-Subscription-Key: d366a666c8604c4db8514f8b5ab3bf92" -H "DisplayLocale: en-US"

**//To download the models**

docker run --rm -it mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text Punct Rescore Resegment Wfstitn **DisplayLocale**=en-US Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

docker run --rm -it -p 5000:5000 --memory 8g --cpus 8 mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text **ModelId= 97e330d0-9020-40ab-b780-3278e905af50** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

curl -X GET "https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/models/97e330d0-9020-40ab-b780-3278e905af50" \ -H "Ocp-Apim-Subscription-Key: d366a666c8604c4db8514f8b5ab3bf92" \ -H "Content-Type: application/json"

curl -o /usr/local/models/properties.json "URL\_FROM\_PREVIOUS\_STEP"

download the model inside the container:  
curl -o /usr/local/models/properties.json <https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/models/base/4cc8c4b2-1d7b-436b-8831-f5451d96315e/manifest>

ls -la /usr/local/models/

docker run --rm -it -p 5000:5000 --memory 8g --cpus 8 -e BaseModelLocale=he-IL -e Eula=accept -e Billing=https://eastus.api.cognitive.microsoft.com -e ApiKey=d366a666c8604c4db8514f8b5ab3bf92 -e MODELID=97e330d0-9020-40ab-b780-3278e905af50 mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text

//Fetching a model???

docker run --rm -it mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text **MODELID=**5b9a1565-cbfc-4e89-8033-785f00d20cde Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=d366a666c8604c4db8514f8b5ab3bf92

docker run --rm -it mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text BaseModelLocale=en-US Eula=accept Billing= <https://eastus.api.cognitive.microsoft.com> ApiKey=d366a666c8604c4db8514f8b5ab3bf92

//This gives the models:  
curl -v -X GET "https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/models/base" -H "Ocp-Apim-Subscription-Key: eca4a4196836494f8099e779b1097c45"

Turkish

{

"self": "https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/models/base/**dc30ea31-556b-4953-aaa6-a3be79dd2913**",

"links": {

"manifest": "https://eastus.api.cognitive.microsoft.com/speechtotext/v3.2/models/base/dc30ea31-556b-4953-aaa6-a3be79dd2913/manifest"

},

"properties": {

"deprecationDates": {

"adaptationDateTime": "2020-04-15T00:00:00Z",

"transcriptionDateTime": "2020-04-15T00:00:00Z"

},

"features": {

"supportsTranscriptions": true,

"supportsEndpoints": true,

"supportsTranscriptionsOnSpeechContainers": false,

"supportsAdaptationsWith": [

"Language",

"OutputFormatting"

],

"supportedOutputFormats": [

"Display",

"Lexical"

]

},

"chargeForAdaptation": false

},

"lastActionDateTime": "2018-08-20T21:51:00Z",

"status": "Succeeded",

"createdDateTime": "2018-08-20T21:51:00Z",

"locale": "tr-TR",

"displayName": "20180801 (v1.0 Beta Interactive)",

"description": "Interactive"

},

subscription id: 955bf3e1-8d3e-49c3-9afe-732fe7d27e8a

**ACTUS** Azure **Speech** Service keys: **(S1)**  
ApiKey 1: **e285874fa6e044d99773dd54b3ee70f5**

ApiKey 2: **f5baa9c9bcda4cd9a85d45becbd8489b**

**eastus**

Billing: [**https://eastus.api.cognitive.microsoft.com**](https://eastus.api.cognitive.microsoft.com/)

**ACTUS** Azure **Translator** Service keys: **(S1)**

Needed: An approved access request to either a [Translator connected container](https://aka.ms/csgate-translator)

ApiKey 1: **c6c7db2fd69142bc998463e602d3ff28**

ApiKey 2: **62926013ed8445c38970df257163dfc8**

**eastus**

Billing: [**https://eastus.api.cognitive.microsoft.com**](https://eastus.api.cognitive.microsoft.com/) **??  
For containers**:  
<https://gs-translator-instance.cognitiveservices.azure.com/>

**For Web API**  
Text Translation :  <https://api.cognitive.microsofttranslator.com/>

Document Translation: <https://gs-translator-instance.cognitiveservices.azure.com/>

**WORKING ON GPU MACHINE**

**Combinations**: run on **win**/**wsl2**, run a **standalone** container from the command line,  
run as a **batch** with configuration, run diagnostics, **command line REST** request / from **code**

**Download docker images:**

docker pull mcr.microsoft.com/azure-cognitive-services/translator/**text-translation**:latest

docker pull mcr.microsoft.com/azure-cognitive-services/speechservices/**speech-to-text**:latest

docker pull mcr.microsoft.com/azure-cognitive-services/speechservices/**custom-speech-to-text**:latest

docker pull **batchkit/speech-batch-kit**

docker pull mcr.microsoft.com/azure-cognitive-services/**diagnostic**

docker pull mcr.microsoft.com/azure-cognitive-services/speechservices/**language-detection**:latest

docker pull mcr.microsoft.com/azure-cognitive-services/speechservices/language-detection:latest

**Run the dockers:**

Run as standalone:

**//Running the diagnostics container:**

//Diagnosing the translator container:

(diagnostics **succeeded**)docker run --rm mcr.microsoft.com/azure-cognitive-services/diagnostic eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=c6c7db2fd69142bc998463e602d3ff28 🡸 the ApiKey is set to point to the actual service like the speech-to-text or translator

**//Downloading language models into the text-translation container!**

docker run --rm -it -p 5000:5000 --memory 12g --cpus 4 -v C:\Temp\TranslatorContainer:/usr/local/models -e apikey=c6c7db2fd69142bc998463e602d3ff28 -e eula=accept -e billing=https://eastus.api.cognitive.microsoft.com -e Languages=en,fr,es,ar,ru mcr.microsoft.com/azure-cognitive-services/translator/text-translation:latest

// Diagnosing the speech-to-text container:

(diagnostics **succeeded**)

docker run --rm mcr.microsoft.com/azure-cognitive-services/diagnostic eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5

**//Running the translator container from the command line:**

Resulted in error: probably because of missing agreement form! It fails to download language models!

Updated LD\_LIBRARY\_PATH to /app

Starting model download

2024/08/08 13:33:20 [Info] Fetching Manifest from Model distribution service: https://eastus.api.cognitive.microsoft.com/mods/onprem/v1.0/texttranslation/models/latest?languages=en,es

2024/08/08 13:33:20 [Fatal] Model distribution service returns error, response status: 400, 400 Bad Request, message: {"error":{"code":"InvalidRequest","message":"Invalid request has been sent.", "innererror":{"code" : "**UnsupportedFeature**", "message":"Feature is not supported"}}

docker run --rm -it -p 5000:5000 --memory 12g --cpus 4 -v C:\Temp\TranslatorContainer:/usr/local/models -e apikey=c6c7db2fd69142bc998463e602d3ff28 -e eula=accept -e billing=https://eastus.api.cognitive.microsoft.com -e Languages=en,fr,es mcr.microsoft.com/azure-cognitive-services/translator/**text-translation:latest**

**//Running the translator container using docker compose:**

Running from: C:\Temp\AzureContainerConfig\TRANSLATOR where it has a docker-compose.yaml file:

docker compose up

**Validating that a container is running:**

http://localhost:5000

**//Running the speech to text container from the command line with logging**

docker run --rm -it -p 5000:5000 --memory 12g --cpus 6 mcr.microsoft.com/azure-cognitive-services/speechservices/speech-to-text Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5 **Logging:Console:LogLevel:Default=Information**

docker run --rm -it -p **5000**:5000 --memory 8g --cpus 4 mcr.microsoft.com/azure-cognitive-services/speechservices/speech-to-text Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5 Logging:Console:LogLevel:Default=Information

//Each locale has its own **dedicated docker image** ! for example: **Hebrew**

docker run --rm -it -p **5000:5000** --memory 8g --cpus 4 mcr.microsoft.com/azure-cognitive-services/speechservices/**speech-to-text:4.8.0-amd64-he-il** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5 Logging:Console:LogLevel:Default=Information

docker run --rm -it -p **5001:5000** --memory 8g --cpus 4 mcr.microsoft.com/azure-cognitive-services/speechservices/**speech-to-text:4.8.0**- amd64-ar-eg Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5 Logging:Console:LogLevel:Default=Information

**Consuming speech to text output:**

Using **code** – see my project:

C:\Development\Azure Cognitive Containers\AzureContainerClients\AzureContainerClients.csproj

//service is in the **cloud**:

var config = SpeechConfig.**FromSubscription**("d366a666c8604c4db8514f8b5ab3bf92", "eastus");

//service is in a **container**

var config = SpeechConfig.**FromHost**(new Uri("**ws**://localhost:5000"));

**//Getting the models of a locale for** **custom-speech-to-text**  
docker run --rm -it mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text BaseModelLocale=he-IL Eula=accept

Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5

**//Downloading models?**

docker run --rm -it mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text **Punct Rescore Resegment Wfstitn** DisplayLocale=he-IL Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5 **//Failed**

docker run --rm -it mcr.microsoft.com/azure-cognitive-services/speechservices/custom-speech-to-text **FullDisplay** DisplayLocale=he-IL Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5 **//Failed**

**//Running In batch**

1. Run first the speech container. One container for each language.
2. Run the LID container (**language-detection)**
3. Run the batch container. One batch for each language.

**//Running the batch container**

Create **config.yaml** at C:\BATCH\_FOLDER

(

# Configuration for the English Speech-to-Text Container

EnglishContainer:

concurrency: 10

host: localhost

port: 5001

# Configuration for the Hebrew Speech-to-Text Container

HebrewContainer:

concurrency: 10

host: localhost

port: 5002

# Configuration for the Language Identification (LID) Endpoint

LIDContainer:

concurrency: 10

host: localhost

port: 5003

language: lid # This marks it as a LID endpoint

# Configuration for Diarization

DiarizationContainer:

concurrency: 10

host: localhost

port: 5004

)

docker run --network host --rm -it -v C:\BATCH\_FOLDER\ENGLISH\_FILES:/my\_nfs batchkit/speech-batch-kit:latest -config /my\_nfs/config.yaml -input\_folder /my\_nfs/audio\_files -output\_folder /my\_nfs/transcriptions -log\_folder /my\_nfs/logs

docker run --network host --rm -it -v C:\BATCH\_FOLDER\HEBREW\_FILES:/my\_nfs batchkit/speech-batch-kit:latest -config /my\_nfs/config.yaml -input\_folder /my\_nfs/audio\_files -output\_folder /my\_nfs/transcriptions -log\_folder /my\_nfs/logs

**//Running with languages**

perhaps language-detection should run before the transcription

Process – look at the config file!

**//currently only one language is supported using the batch. Maybe because LID**

**language identifier container demands getting azure approval?**

docker run --network host --rm -it -v C:\Temp\BATCH\_FOLDER:/my\_nfs batchkit/speech-batch-kit:latest -config /my\_nfs/config.yaml -input\_folder /my\_nfs/audio\_files -output\_folder /my\_nfs/transcriptions -log\_folder /my\_nfs/logs **-language en-US ~~he-IL~~**

**//Running the language-detection container**

**(this container needs approval!)**

**ERROR:**Subscription validation failed with status **'ContainerUseUnauthorized'**. The API Key provided is not authorized for use with this container. This is a gated container, **make sure your Azure Subscription ID is approved**: https://aka.ms/csgate. For more information please visit https://aka.ms/cscontainers-faq.

Container is not in a valid state. Subscription validation failed with status 'ContainerUseUnauthorized'. The API Key provided is not authorized for use with this container. This is a gated container, make sure your Azure Subscription ID is approved: https://aka.ms/csgate. For more information please visit https://aka.ms/cscontainers-faq.

Service responded with error message: Unsupported Container Type.

02:22:14 fail: Microsoft.CloudAI.Containers.Http.CloudClient[0] Service responded with error message: Unsupported Container Type.

docker run --rm -it -p 5000:5003 --memory 1g --cpus 1 mcr.microsoft.com/azure-cognitive-services/speechservices/**language-detection** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5

**To get the batch container help, run:**docker run --network host --rm -it -v C:\Temp\BATCH\_FOLDER:/my\_nfs batchkit/speech-batch-kit:latest -config /my\_nfs/config.yaml -input\_folder /my\_nfs/audio\_files -output\_folder /my\_nfs/transcriptions -log\_folder /my\_nfs/logs **–help**

**Azure OpenAI service**

Resource name**: gs-azure-openai-instance**

**key1**: **1f8600926ddc477a98127f44c5e02dfb**

**key2**: **e65cc9bdd8524589bd25e731021c8b37**

**eastus**

endpoint: [**https://gs-azure-openai-instance.openai.azure.com/**](https://gs-azure-openai-instance.openai.azure.com/)

I need to create a deployment with a model inside – I chose gpt-4 global-batch

**//Running a transcribing using Azure Open AI with deployment of gpt-4**

**//Error: deployment wasn't created!**

curl https://gs-azure-openai-instance.openai.azure.com/openai/deployments/gpt-4/audio/transcriptions?api-version=2024-02-01 -H "api-key: 1f8600926ddc477a98127f44c5e02dfb" -H "Content-Type: multipart/form-data" -F file="@./speechService.wav"

//Error: to run OpenAI service with **whisper** model – **only a few regions** allow to install the Whisper model which can perform transcription!  
**gpt-4** doesn't support transcription, it supports chats or so…

//Running **language identification container** with the[**speech to text**](https://learn.microsoft.com/en-us/azure/ai-services/speech-service/speech-container-stt)**container**docker run --rm -ti –network=host antsu/on-prem-client:latest ./speech-to-text-client --host 51.143.123.142 --port 80 --expect "What's the weather like" ./audio/whatstheweatherlike.wav

Running the docker shell:  
**docker run --rm -it antsu/on-prem-client:latest /bin/sh**

//IT SEEMS FAULTED

docker run --rm -ti -v C:\Development\MEDIA-FILES:/audio antsu/on-prem-client:latest ./speech-to-text-client --host 51.143.123.142 --port 80 /audio/aboutSpeechSdk.wav  
  
docker run --rm -ti -v C:\Development\MEDIA-FILES:/audio antsu/on-prem-client:latest ./speech-to-text-client --windows --host localhost --port 80 --expect "What's the weather like" /audio/aboutSpeechSdk.wav

docker run --network host --rm -it -v C:\Development\MEDIA:/audio antsu/on-prem-client:latest ./speech-to-text-client --windows --host localhost --port 80 --expect "What's the weather like" /audio/aboutSpeechSdk.wav

docker run --rm -ti -v C:\Development\MEDIA:/audio antsu/on-prem-client:latest ./speech-to-text-client --host 51.143.123.142 --port 80 --expect "What's the weather like" /audio/aboutSpeechSdk.wav

docker run --rm -ti --network=host antsu/on-prem-client:latest ./speech-to-text-client --host 51.143.123.142 --port 80 --expect "What's the weather like" ./audio/whatstheweatherlike.wav

//**Running the language docker**

docker run --rm -it -p 5000:5000 --memory 4g --cpus 1 mcr.microsoft.com/azure-cognitive-services/textanalytics/**language** Eula=accept Billing=https://eastus.api.cognitive.microsoft.com ApiKey=e285874fa6e044d99773dd54b3ee70f5

**SPEECHMATIX**

Repo: **user name**: **actusdigital**

Password: AKCpBtMKaqcSb9vkUn7KQP78fpZNzRTxPZktJCn7MgPPVWT2i5mA7Azs6EmfECXfYvmSUCTUk

speechmatics config set --auth-token eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI1MDYyNDAwLCJqdGkiOiI0NzcxYTlkOWEyZmY0YmE4YmQ4MWI0ZGQ2NWFmZWNkNSIsImlhdCI6MTcyMjUyODkxMywiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzIyNDcwNDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.BDLziehVAoDpdnDVlqAbjPOjBiwEh3zVhInCcNKO7HdbFua3IxBMhDpv-MYc3AxZrBqUu8IeAJgo8fFrE6YC9fBYzWkkWr3laWlrA3iBFubKwS8yGTwLuRmYw7ol1T4ZzgUTt06sVV4-uYgzE9-Sm1YHuA94C0d5\_9lR\_F2z5iKu9ydlfrSLEMTla4qcvKuMNUNmR\_qCNnRa5ERJx1583k1RsDorwbzLicRo6lcEzEdsdp13f9DGzRD-jzIorutyGFQ-iuLDiI8J8uiJfADgNAzo0YKrOwPdZ2QDl-JR9CTGJpgbYsDI6Fo8GVUKWVyN0\_LLUCAUHSY4QdAHWH6s2w

curl -L -X POST "https://asr.api.speechmatics.com/v2/jobs/" -H "Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI1MDYyNDAwLCJqdGkiOiI0NzcxYTlkOWEyZmY0YmE4YmQ4MWI0ZGQ2NWFmZWNkNSIsImlhdCI6MTcyMjUyODkxMywiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzIyNDcwNDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.BDLziehVAoDpdnDVlqAbjPOjBiwEh3zVhInCcNKO7HdbFua3IxBMhDpv-MYc3AxZrBqUu8IeAJgo8fFrE6YC9fBYzWkkWr3laWlrA3iBFubKwS8yGTwLuRmYw7ol1T4ZzgUTt06sVV4-uYgzE9-Sm1YHuA94C0d5\_9lR\_F2z5iKu9ydlfrSLEMTla4qcvKuMNUNmR\_qCNnRa5ERJx1583k1RsDorwbzLicRo6lcEzEdsdp13f9DGzRD-jzIorutyGFQ-iuLDiI8J8uiJfADgNAzo0YKrOwPdZ2QDl-JR9CTGJpgbYsDI6Fo8GVUKWVyN0\_LLUCAUHSY4QdAHWH6s2w" -F data\_file=@./katie-and-steve.wav -F config='{"type": "transcription", "transcription\_config": {"operating\_point": "enhanced", "language": "en"}}'

docker run -i -v C:\BATCH\_FOLDER\ENGLISH\_FILES\audio\_files\neuralActivationPhrase.wav:/input.audio -e LICENSE\_TOKEN=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI1MDYyNDAwLCJqdGkiOiI0NzcxYTlkOWEyZmY0YmE4YmQ4MWI0ZGQ2NWFmZWNkNSIsImlhdCI6MTcyMjUyODkxMywiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzIyNDcwNDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.BDLziehVAoDpdnDVlqAbjPOjBiwEh3zVhInCcNKO7HdbFua3IxBMhDpv-MYc3AxZrBqUu8IeAJgo8fFrE6YC9fBYzWkkWr3laWlrA3iBFubKwS8yGTwLuRmYw7ol1T4ZzgUTt06sVV4-uYgzE9-Sm1YHuA94C0d5\_9lR\_F2z5iKu9ydlfrSLEMTla4qcvKuMNUNmR\_qCNnRa5ERJx1583k1RsDorwbzLicRo6lcEzEdsdp13f9DGzRD-jzIorutyGFQ-iuLDiI8J8uiJfADgNAzo0YKrOwPdZ2QDl-JR9CTGJpgbYsDI6Fo8GVUKWVyN0\_LLUCAUHSY4QdAHWH6s2w speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.6.0

**GPU MACHINE ID: 1568828622 (IP in LiveU: 192.168.152.9)**

**//Running a transcription with CPU container, using license file (FAILED)**

docker run -i -v C:\BATCH\_FOLDER\ENGLISH\_FILES\audio\_files\neuralActivationPhrase.wav:/input.audio -v C:\Users\GPU System\BATCH\_FOLDER\license\_actus.json:/license.json speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.6.0

**//Running a transcription with CPU container, using the actual token which is in the file** **(signedclaimstoken) (SUCCEEDED)**

docker run -i -v C:\BATCH\_FOLDER\ENGLISH\_FILES\audio\_files\neuralActivationPhrase.wav:/input.audio -v C:\BATCH\_FOLDER\ENGLISH\_FILES\audio\_files\output:/output -e LICENSE\_TOKEN=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI3Mzk1MjAwLCJqdGkiOiIwM2ZiNGQyNTE0OTg0OTk4OWUwOGM2Mzc2MDUzMWE5ZSIsImlhdCI6MTcyNTk2Mzk1MSwiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzI1OTI2NDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.un4cqd5nvzn3BljpK6Nxgam3Q5YTnZbQ6GOhrkhTNtYGtSy7nLjkRfuBqRQzoJt7YIoaWIOlKxax29NHm\_drt50frPq451R-oUzywPEsFe0eqz50NoJfiHsGwZvzoIVSCOoXBViJIwzuWCGEu6jT6pKhOiMsxzRmkPTVM\_PkQ2FTaIIeVtUz\_gBE3MIfuLW9rJvKb-O4ygMDcHGQNGerPMC4on6VhqLeTdNajBQzD3\_AyPNMbylXBpmYyI5lxD0OUf4chuUmkX8SLSpY1bsHTsxgdOxsZrm2m-UUAMP3Osf3\_ziIiclqWhHX2QRe8EN1n6fji4wbDRAz58aYoBv1Lg speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.6.0 –all-formats /output

//Saving results to a folder with all-formats:  
docker run -i -v C:\BATCH\_FOLDER\ENGLISH\_FILES\audio\_files\neuralActivationPhrase.wav:/input.audio **-v "C:\Users\GPU System\BATCH\_FOLDER\output":/output** -e LICENSE\_TOKEN=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI3Mzk1MjAwLCJqdGkiOiIwM2ZiNGQyNTE0OTg0OTk4OWUwOGM2Mzc2MDUzMWE5ZSIsImlhdCI6MTcyNTk2Mzk1MSwiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzI1OTI2NDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.un4cqd5nvzn3BljpK6Nxgam3Q5YTnZbQ6GOhrkhTNtYGtSy7nLjkRfuBqRQzoJt7YIoaWIOlKxax29NHm\_drt50frPq451R-oUzywPEsFe0eqz50NoJfiHsGwZvzoIVSCOoXBViJIwzuWCGEu6jT6pKhOiMsxzRmkPTVM\_PkQ2FTaIIeVtUz\_gBE3MIfuLW9rJvKb-O4ygMDcHGQNGerPMC4on6VhqLeTdNajBQzD3\_AyPNMbylXBpmYyI5lxD0OUf4chuUmkX8SLSpY1bsHTsxgdOxsZrm2m-UUAMP3Osf3\_ziIiclqWhHX2QRe8EN1n6fji4wbDRAz58aYoBv1Lg speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.6.0 **--all-formats /output**

**Running the GPU STT inference container**

**Standard: docker compose -f docker-compose-gpu-standard.yml up** **//SUCCEEDED** pay attention to the license file!

**Enhanced: docker compose -f docker-compose-gpu-enhanced-en.yml up**// **SUCCEEDED**:~~ended with some error:~~

~~(Engine Error: kaldi::KaldiFatalError Cleanup Error: kaldi::KaldiFatalError)~~

**Enhanced: docker compose -f docker-compose-gpu-enhanced-recipe1.yml up //SUCCEEDED**

**Running the CPU STT container to send data to the GPU container**

dockC:\BATCH\_FOLDER\ENGLISH\_FILES\audio\_files\neuralActivationPhrase.wav:/input.audio **-v "C:\Users\GPU System\BATCH\_FOLDER\output":/output** -e LICENSE\_TOKEN=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI3Mzk1MjAwLCJqdGkiOiIwM2ZiNGQyNTE0OTg0OTk4OWUwOGM2Mzc2MDUzMWE5ZSIsImlhdCI6MTcyNTk2Mzk1MSwiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzI1OTI2NDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.un4cqd5nvzn3BljpK6Nxgam3Q5YTnZbQ6GOhrkhTNtYGtSy7nLjkRfuBqRQzoJt7YIoaWIOlKxax29NHm\_drt50frPq451R-oUzywPEsFe0eqz50NoJfiHsGwZvzoIVSCOoXBViJIwzuWCGEu6jT6pKhOiMsxzRmkPTVM\_PkQ2FTaIIeVtUz\_gBE3MIfuLW9rJvKb-O4ygMDcHGQNGerPMC4on6VhqLeTdNajBQzD3\_AyPNMbylXBpmYyI5lxD0OUf4chuUmkX8SLSpY1bsHTsxgdOxsZrm2m-UUAMP3Osf3\_ziIiclqWhHX2QRe8EN1n6fji4wbDRAz58aYoBv1Lg -e **SM\_INFERENCE\_ENDPOINT=sm-triton:8001** speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.6.0 **--all-formats /output**

docker run -i **--network transcriber** -v "C:\Users\GPU System\BATCH\_FOLDER\audio\_files\ENGLISH\_FILES\**CBS.wav**":/input.audio -v "C:\Users\GPU System\BATCH\_FOLDER\transcriptions":/output -v "C:\Users\GPU System\BATCH\_FOLDER\config.json":/**config.json** -v "C:\Users\GPU System\BATCH\_FOLDER\**license-actus.json**":/**license.json:ro** -e **SM\_INFERENCE\_ENDPOINT=sm-triton:8001** speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.7.0 --all-formats /output

**Running the CPU STT container with configuration and with mp3 (vs wav)**

~~docker run -i~~ **~~--network transcriber~~** ~~-v C:\BATCH\_FOLDER\ENGLISH\_FILES\audio\_files\nbc.~~**~~mp3~~**~~:/input.audio~~ **~~-v "C:\Users\GPU System\BATCH\_FOLDER\output":/output~~****~~-v "C:\Users\GPU System\BATCH\_FOLDER\config.json":/config.json~~** ~~-e LICENSE\_TOKEN=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI3Mzk1MjAwLCJqdGkiOiIwM2ZiNGQyNTE0OTg0OTk4OWUwOGM2Mzc2MDUzMWE5ZSIsImlhdCI6MTcyNTk2Mzk1MSwiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzI1OTI2NDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.un4cqd5nvzn3BljpK6Nxgam3Q5YTnZbQ6GOhrkhTNtYGtSy7nLjkRfuBqRQzoJt7YIoaWIOlKxax29NHm\_drt50frPq451R-oUzywPEsFe0eqz50NoJfiHsGwZvzoIVSCOoXBViJIwzuWCGEu6jT6pKhOiMsxzRmkPTVM\_PkQ2FTaIIeVtUz\_gBE3MIfuLW9rJvKb-O4ygMDcHGQNGerPMC4on6VhqLeTdNajBQzD3\_AyPNMbylXBpmYyI5lxD0OUf4chuUmkX8SLSpY1bsHTsxgdOxsZrm2m-UUAMP3Osf3\_ziIiclqWhHX2QRe8EN1n6fji4wbDRAz58aYoBv1Lg~~ **~~-e SM\_INFERENCE\_ENDPOINT=sm-triton:8001~~** ~~speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.6.0~~ **~~--all-formats /output~~**

docker run -i **--network transcriber** -v "C:\ **Users\GPU System\BATCH\_FOLDER** \ \audio\_files\ENGLISH\_FILES\CBS.wav:/input.audio **-v "C:\Users\GPU System\BATCH\_FOLDER\transcriptions":/output** **-v "C:\Users\GPU System\BATCH\_FOLDER\config.json":/config.json** -e LICENSE\_TOKEN=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI3Mzk1MjAwLCJqdGkiOiIwM2ZiNGQyNTE0OTg0OTk4OWUwOGM2Mzc2MDUzMWE5ZSIsImlhdCI6MTcyNTk2Mzk1MSwiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzI1OTI2NDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.un4cqd5nvzn3BljpK6Nxgam3Q5YTnZbQ6GOhrkhTNtYGtSy7nLjkRfuBqRQzoJt7YIoaWIOlKxax29NHm\_drt50frPq451R-oUzywPEsFe0eqz50NoJfiHsGwZvzoIVSCOoXBViJIwzuWCGEu6jT6pKhOiMsxzRmkPTVM\_PkQ2FTaIIeVtUz\_gBE3MIfuLW9rJvKb-O4ygMDcHGQNGerPMC4on6VhqLeTdNajBQzD3\_AyPNMbylXBpmYyI5lxD0OUf4chuUmkX8SLSpY1bsHTsxgdOxsZrm2m-UUAMP3Osf3\_ziIiclqWhHX2QRe8EN1n6fji4wbDRAz58aYoBv1Lg **-e SM\_INFERENCE\_ENDPOINT=sm-triton:8001** speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.6.0 **--all-formats /output**

**PAY ATTNETION:**if GPU is standard then the **config** of CPU container should be "**operating\_point**":"**standard**"  
if GPU is enhanced then the config of CPU container should be "**operating\_point**":"**enhanced**"

**Running the CPU STT container to do TR - pay attention to the config**

docker run -i **--network transcriber** -v C:\BATCH\_FOLDER\ENGLISH\_FILES\audio\_files\nbc.**mp3**:/input.audio **-v "C:\Users\GPU System\BATCH\_FOLDER\output":/output** **-v "C:\Users\GPU System\BATCH\_FOLDER\config.json":/config.json** -e LICENSE\_TOKEN=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI3Mzk1MjAwLCJqdGkiOiIwM2ZiNGQyNTE0OTg0OTk4OWUwOGM2Mzc2MDUzMWE5ZSIsImlhdCI6MTcyNTk2Mzk1MSwiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzI1OTI2NDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.un4cqd5nvzn3BljpK6Nxgam3Q5YTnZbQ6GOhrkhTNtYGtSy7nLjkRfuBqRQzoJt7YIoaWIOlKxax29NHm\_drt50frPq451R-oUzywPEsFe0eqz50NoJfiHsGwZvzoIVSCOoXBViJIwzuWCGEu6jT6pKhOiMsxzRmkPTVM\_PkQ2FTaIIeVtUz\_gBE3MIfuLW9rJvKb-O4ygMDcHGQNGerPMC4on6VhqLeTdNajBQzD3\_AyPNMbylXBpmYyI5lxD0OUf4chuUmkX8SLSpY1bsHTsxgdOxsZrm2m-UUAMP3Osf3\_ziIiclqWhHX2QRe8EN1n6fji4wbDRAz58aYoBv1Lg **-e SM\_INFERENCE\_ENDPOINT=sm-triton:8001** **-e SM\_TRANSLATION\_ENDPOINT <server>:<port>** speechmatics-docker-public.jfrog.io/batch-asr-transcriber-en:10.6.0 **--all-formats /output**

**//To get the NVIDIA card features:**

**nvidia-smi**

In the **task** **manager** in the performance tab – we can see the **GPU** **memory** and **GPU dedicated memory** of the NVIDIA card

שים לב: אם הconfig.json של הCPU container מכיל: operating\_point: enhanced אזי יש להריץ   
 GPU containerשהוא מסוג enhanced אחרת הCPU docker לא ירוץ!

כאשר מבצעים translation + STT הכל נעשה במסגרת ג'וב אחד ויש לספק לContainer env var נוסף שבו יושב הTR GPU container

**Running the language identification container**

docker run -i -v "C:\Users\GPU System\BATCH\_FOLDER\audio\_files\ENGLISH\_FILES\**nbc.mp3**":/input.audio -v "C:\Users\GPU System\BATCH\_FOLDER\license-actus.json":/license.json:ro speechmatics-docker-public.jfrog.io/langid:2.2.1

returned:  
{"format": "1.1", "metadata": {"created\_at": "2024-09-12T14:49:30+0000", "type": "language\_identification", "language\_identification\_config": {}, "duration": 299.688, "processed\_duration": 120.0}, "results": [{"alternatives": [{"language": "**en**", "**confidence**": **0.93**}, {"language": "nl", "confidence": 0.01}, {"language": "de", "confidence": 0.01}, {"language": "ms", "confidence": 0.01}, {"language": "pt", "confidence": 0.01}, {"language": "cmn", "confidence": 0.0}, {"language": "fr", "confidence": 0.0}, {"language": "ko", "confidence": 0.0}, {"language": "id", "confidence": 0.0}, {"language": "ja", "confidence": 0.0}], "start\_time": 0.0, "end\_time": 299.69}], **"predicted\_language": "en"**}

//Trying to detect Hebrew language, running:

docker run -i -v "C:\Users\GPU System\BATCH\_FOLDER\audio\_files\HEBREW\_FILES\**HEBREW.wav**":/input.audio -v "C:\Users\GPU System\BATCH\_FOLDER\license-actus.json":/license.json:ro speechmatics-docker-public.jfrog.io/langid:2.2.1

**Hebrew detection Failed**, returning:  
{"format": "1.1", "metadata": {"created\_at": "2024-09-12T14:53:12+0000", "type": "language\_identification", "language\_identification\_config": {}, "duration": 78.419563, "processed\_duration": 60.0}, "results": [{"alternatives": [{"language": "en", "confidence": 0.01}, {"language": "ar", "confidence": 0.01}, {"language": "fr", "confidence": 0.0}, {"language": "de", "confidence": 0.0}, {"language": "el", "confidence": 0.0}, {"language": "ru", "confidence": 0.0}, {"language": "sl", "confidence": 0.0}, {"language": "be", "confidence": 0.0}, {"language": "nl", "confidence": 0.0}, {"language": "es", "confidence": 0.0}], "start\_time": 0.0, "end\_time": 78.42}], "error": "LOW\_CONFIDENCE", "**message": "Language identification could not identify any language with sufficient confidence**."}

Pay attention: **English, French, Spanish were detected correctly**!

docker run -i -v "C:\Users\GPU System\BATCH\_FOLDER\audio\_files\HEBREW\_FILES\HEBREW.wav":/input.audio -v "C:\Users\GPU System\BATCH\_FOLDER\license-actus.json":/license.json:ro speechmatics-docker-public.jfrog.io/langid:2.2.1 **--expected-languages en,fr,he** //we can supply expected languages list

//English STT

"C:\Users\GPU System\BATCH\_FOLDER\audio\_files\**ENGLISH\_FILES**" "C:\Users\GPU System\BATCH\_FOLDER\**config.json**" "speechmatics-docker-public.jfrog.io/batch-asr-transcriber-**en:10.7.0**" "C:\Users\GPU System\BATCH\_FOLDER\license-actus.json"

//Spanish STT

"C:\Users\GPU System\BATCH\_FOLDER\audio\_files\**SPANISH\_FILES**" "C:\Users\GPU System\BATCH\_FOLDER\**config-es.json**" "speechmatics-docker-public.jfrog.io/batch-asr-transcriber-**es:10.6.0**" "C:\Users\GPU System\BATCH\_FOLDER\license-actus.json"

It Seems that docker desktop cannot run on windows server **2019/2022**, but docker service can. The problem is running a **WSL2/LCOW**

If you need a local model without any internet access, here are some suitable options from Hugging Face:

1. **EleutherAI Models**:
   * **EleutherAI/gpt-j-6B**: A powerful model with 6 billion parameters that can handle a variety of tasks, similar to GPT-3.
   * **EleutherAI/gpt-neox-20b**: A larger model (20 billion parameters) that may provide better performance but requires significant computational resources.
2. **LLaMA Models**:
   * **Meta's LLaMA**: These models (e.g., meta-llama/Llama-2-7b or larger versions) are also suitable for local use and can be downloaded from Hugging Face.
3. **GPT-Neo**:
   * **EleutherAI/gpt-neo-1.3B** and **EleutherAI/gpt-neo-2.7B**: These models are smaller alternatives to GPT-3 that can still perform well for various tasks.

**Speechmatix CLOUD**

API\_KEY="eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJmZWF0dXJlcyI6WyJNQVBCQSIsIkFNQ0MiLCJBUEQiLCJBUFIiLCJBU1MiLCJBT1AyIiwiQU9QMyIsIkFPUDQiLCJBTElEIiwiQVRSQSIsIkFTRU5UIiwiQUFFRCIsIkxBTlkiXSwiY29udHJhY3RpZCI6IkUtMTE1OCIsImlzX3RyaWFsIjpmYWxzZSwiZXhwIjoxNzI3Mzk1MjAwLCJqdGkiOiIwM2ZiNGQyNTE0OTg0OTk4OWUwOGM2Mzc2MDUzMWE5ZSIsImlhdCI6MTcyNTk2Mzk1MSwiaXNzIjoiU3BlZWNobWF0aWNzIiwibmJmIjoxNzI1OTI2NDAwLCJzdWIiOiJBY3R1cyBEaWdpdGFsIn0.un4cqd5nvzn3BljpK6Nxgam3Q5YTnZbQ6GOhrkhTNtYGtSy7nLjkRfuBqRQzoJt7YIoaWIOlKxax29NHm\_drt50frPq451R-oUzywPEsFe0eqz50NoJfiHsGwZvzoIVSCOoXBViJIwzuWCGEu6jT6pKhOiMsxzRmkPTVM\_PkQ2FTaIIeVtUz\_gBE3MIfuLW9rJvKb-O4ygMDcHGQNGerPMC4on6VhqLeTdNajBQzD3\_AyPNMbylXBpmYyI5lxD0OUf4chuUmkX8SLSpY1bsHTsxgdOxsZrm2m-UUAMP3Osf3\_ziIiclqWhHX2QRe8EN1n6fji4wbDRAz58aYoBv1Lg"

PATH\_TO\_FILE="C:\Development\MEDIA\kennedy.wav"

curl -L -X POST "https://asr.api.speechmatics.com/v2/jobs/" -H "Authorization: Bearer ${API\_KEY}" -F data\_file=@${PATH\_TO\_FILE} -F config='{"type": "transcription","transcription\_config": { "operating\_point":"enhanced", "language": "en" }}'

curl -L -X POST "https://asr.api.speechmatics.com/v2/jobs/" -H "Authorization: Bearer Pu9AQwCv1Wuv5Q1yA4bIouY8hagC5vYq" -F data\_file="C:\Development\MEDIA\kennedy.wav" -F config='{"type": "transcription","transcription\_config": { "operating\_point":"enhanced", "language": "en" }}'

actus-key**: Pu9AQwCv1Wuv5Q1yA4bIouY8hagC5vYq** (generated on the web site)

actus-key-standard-mode: **KnnsdeTO2uM3mztH2aXwTjfBWYHhjVmg**

actus-key-standard-mode-more-quoata: **Lrk3dJpzmRARyVi2mi39fJrGy8EeUaPl**

home-key: **ehXxfTrFZcmfMQp1iuIu9jKpj37aWREV**

For **Windows Command Prompt**:

**Launch a translation JOB:**

curl -L -X POST "https://asr.api.speechmatics.com/v2/jobs/" -H "Authorization: Bearer Pu9AQwCv1Wuv5Q1yA4bIouY8hagC5vYq" -F "data\_file=@C:\Development\MEDIA\CBS.mp3" -F "config={\"type\": \"transcription\", \"transcription\_config\": {\"operating\_point\":\"enhanced\", \"language\": \"en\"}}"

For **WSL** (Unix-style):

curl -L -X POST "https://asr.api.speechmatics.com/v2/jobs/" \

-H "Authorization: Bearer Pu9AQwCv1Wuv5Q1yA4bIouY8hagC5vYq" \

-F data\_file=@/mnt/c/Development/MEDIA/kennedy.wav \

-F config='{"type": "transcription","transcription\_config": { "operating\_point":"enhanced", "language": "en" }}'

**Get the result**

curl -L -X GET "https://asr.api.speechmatics.com/v2/jobs/${JOB\_ID}/transcript?format=srt"

-H "Authorization: Bearer Pu9AQwCv1Wuv5Q1yA4bIouY8hagC5vYq"

//pay attention to the format: **txt/srt/json-v2**

curl -L -X GET https://asr.api.speechmatics.com/v2/jobs/ebjwrlfga1/transcript?format=**srt** -H "Authorization: Bearer Pu9AQwCv1Wuv5Q1yA4bIouY8hagC5vYq"

Pay attention:   
if we use **auto** for the **language** (**auto detection**) than we need to specify operating\_point=**standard**

string configJson = $"{{\"type\": \"transcription\",\"transcription\_config\": {{\"operating\_point\": \"**standard**\", \"language\": \"auto\"}}}}";

**BUILD MACHINE: 197684410**

**From Alex:  
--------------------**  
Set License  
--------------------  
1. Open C:\ActDev\Tools\SystemConfig\SystemConfig.sln  
2. Run SystemConfig  
3. License->Copy license to clipboard  
4. Save Machine code (8217 for example)

5. Open C:\ActDev\Tools\ActusLicenseMgmt\ActusLicenseMgmt.sln  
6. Run ActusLicenseMgmt  
7. Paste to "Licence Key" field  
8. "Check License" button  
9. Input new machine code in "Machine Code" field (8217 for example)  
10. "Generate License" button  
11. Select then copy license to clipboard

12. Return to "System config" window  
13. Paste license to "License Key" field  
14. Press "Install License" button

How to install build  
--------------------  
go to the build machine AnyDesk 197684410  actusmon123  
go to the c:\actdev\scripts  
run  
update  
build  
archive

1.Goto 'corona'  
2.from 'corona' \\actbuild\build\9.1.0.13823  
3.run upgrade.bat

**//TWILIO  
TNEVT3KGMP79K7HMYV1M7378**

**AWS**user name: **demo-ai**Console password: **wJU0-SDRwJU0-SDR**

**TCRA CLIENT**Access is using AnyDesk with ID 1742440143

Password is: 1qaz@WSX

**C:\IntelligenceApps\Whisper\  
  
war,gaza,biden,putin,europe,israel,trump, أوروبا,** **غزة,** **حرب**

**AI – transcription,  translation, summary, key points ,**

**celebrities, places, CHATGPT capabilities**

**REMOTE SERVER IN LAB:  
420807092**

guerre,octobre,gaza,war,israel,europe,putin,trump,france

**Whisper tests: (2.7 MB)**whisper C:\Actus\_Temp\WhipserTest\MP3\test1.mp3 --model base --output\_dir .\results

**WHISPER INSTALLTIONS ON GPU MACHINE:**1. CUDA Toolkit Archive. For **cu125**  
2. Updating   
 PATH:

* C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.5\bin
* C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.5\libnvvp

 CUDA\_PATH:

* C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.5

3.nvcc –version //test

4. cuDNN Download page

5. python -m venv whisper-env  
 whisper-env\Scripts\activate

6. pip install torch torchvision torchaudio --index-url [https://download.pytorch.org/whl/**cu118**](https://download.pytorch.org/whl/cu118) **7.** Run as python script:  
import torch

print("Torch version:", torch.\_\_version\_\_)

print("CUDA available:", torch.cuda.is\_available())

print("CUDA version:", torch.version.cuda)

print("GPU name:", torch.cuda.get\_device\_name(0) if torch.cuda.is\_available() else "No GPU detected")

**In power shell – show run whisper in 12 command prompts (grid**# Enable Windows Forms

Add-Type -AssemblyName System.Windows.Forms

# Helper function to move windows

function Move-Window($handle, $x, $y, $width, $height) {

Add-Type @"

using System;

using System.Runtime.InteropServices;

public class Window {

[DllImport("user32.dll", SetLastError = true)]

public static extern bool MoveWindow(IntPtr hWnd, int X, int Y, int nWidth, int nHeight, bool bRepaint);

}

"@

[Window]::MoveWindow($handle, $x, $y, $width, $height, $true)

}

# Number of rows and columns

$rows = 3

$columns = 4

# Get screen resolution

$screen = [System.Windows.Forms.Screen]::PrimaryScreen.Bounds

$width = $screen.Width / $columns

$height = $screen.Height / $rows

# Launch and position windows

$handles = @()

for ($i = 0; $i -lt $rows; $i++) {

for ($j = 0; $j -lt $columns; $j++) {

# Start a new Command Prompt

$process = Start-Process -PassThru cmd.exe

# Wait briefly to ensure the window is created

Start-Sleep -Milliseconds 500

# Get the handle of the window

$handle = $process.MainWindowHandle

# Calculate position

$x = [Math]::Round($j \* $width)

$y = [Math]::Round($i \* $height)

# Move and resize the window

Move-Window $handle $x $y $width $height

$handles += $handle

}

}

At C:\Actus\bin

//TO add remove services

**ActControl.exe -r** <= to add/remove services

**ActControl.cfg**

**Kill all WhisperTranscriber**

taskkill /F /IM WhisperTranscriber\*

**Neurotechnology license Key ?>>**

**7292446130912038656**

//To see the languages that are supported for subtitles and teletext

//Run locally in the server

[**http://localhost/actus4/test/SetupMultiLang.aspx**](http://localhost/actus4/test/SetupMultiLang.aspx)

**FFMPEG testing:  
C:\Actus\bin\ffmpeg -i channel01\_2024\_12\_07\_02\_40\_00.mp4 -v debug -vn -acodec libmp3lame -q:a 2 output.mp3**

Probably - The input .mp4 files might be corrupted**, have incomplete audio streams**, or no audio data at all.

פרוייקט המספרה:  
**POLICY NAME:** B2C\_1\_EdgardoHairSalon

**Run user flow endpoint**

<https://EdgardoHairSalon.b2clogin.com/EdgardoHairSalon.onmicrosoft.com/oauth2/v2.0/authorize?p=B2C_1_EdgardoHairSalon&client_id=b3ab8bc1-7620-4209-9cb0-3accc4b1d0cd&nonce=defaultNonce&redirect_uri=http%3A%2F%2Flocalhost%3A4200&scope=openid&response_type=code&prompt=login&code_challenge_method=S256&code_challenge=dcae8eE4k7Z3atPSYzJVShcwRKfo65fcfwbJTs3jRlE>  
  
MyTenantOrganization (Tenant name)

Tenant ID: 82cb0fa6-a404-4ffb-8dc6-350347b1b545

B2C\_1\_MyTenantOrganization (policy / user flow)

MyAngularApp 04474317-2051-406d-bea6-e1d9b37fa010 (app client ID)

Resource name: [MyTenantDomainNameName.onmicrosoft.com](https://portal.azure.com/c54eed71-7770-414f-aad3-1c822f650ef5/#resource/subscriptions/8d0ae02b-d4e5-4b5d-ac11-4c4b628c4bbf/resourceGroups/HairSallon_group/providers/Microsoft.AzureActiveDirectory/b2cDirectories/MyTenantDomainNameName.onmicrosoft.com/overview)

'https://<Your-Tenant-Domain>.b2clogin.com/<Your-Tenant-ID>'

**AZURE BLOB STORAGE**

**KEY1**

o/rvXinLvuwCIgmZwt2vmEePeZdxAW1nI2uF4QBh16ezZN9VFkUuwTA3t1KtdNDb8bQiXK+LT+KV+AStGPveqw==

**Connection string:**DefaultEndpointsProtocol=https;AccountName=hairsalonstorage;AccountKey=o/rvXinLvuwCIgmZwt2vmEePeZdxAW1nI2uF4QBh16ezZN9VFkUuwTA3t1KtdNDb8bQiXK+LT+KV+AStGPveqw==;EndpointSuffix=core.windows.net

**KEY2**

HyEAP0SkrMXr8UekI9og4BUgETGzvqgtltLE0x7Kb0OZeJg7aIANDVoT+m3d2AJJG3NqtqCy7w0W+AStCprc9A==  
**connection string:**

DefaultEndpointsProtocol=https;AccountName=hairsalonstorage;AccountKey=HyEAP0SkrMXr8UekI9og4BUgETGzvqgtltLE0x7Kb0OZeJg7aIANDVoT+m3d2AJJG3NqtqCy7w0W+AStCprc9A==;EndpointSuffix=core.windows.net

**.mp4 => .mp3 conversion**

**//FFPROBE Per Channels**

**ffprobe** -v error -show\_format -show\_streams -i D:\Media\Record\channel01\2025\_01\_14\channel01\_2025\_01\_14\_14\_40\_00.mp4 > D:\Media\Record\channel01\2025\_01\_14\output\_report2.txt

**ffprobe** -v error -show\_format -show\_streams -i D:\Media\Record\channel02\2025\_01\_14\channel02\_2025\_01\_14\_14\_40\_00.mp4 > D:\Media\Record\channel02\2025\_01\_14\output\_report2.txt

**ffprobe** -v error -show\_format -show\_streams -i D:\Media\Record\channel03\2025\_01\_14\channel03\_2025\_01\_14\_14\_40\_00.mp4 > D:\Media\Record\channel03\2025\_01\_14\output\_report2.txt

**ffprobe** -v error -show\_format -show\_streams -i D:\Media\Record\channel04\2025\_01\_14\channel04\_2025\_01\_14\_14\_40\_00.mp4 > D:\Media\Record\channel04\2025\_01\_14\output\_report2.txt

**//FFMPEG Per Channels**

**ffmpeg** -i D:\Media\Record\channel01\2025\_01\_14\channel01\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -vn -c:a libmp3lame -q:a 2 D:\Media\Record\channel01\2025\_01\_14\output2.mp3

**ffmpeg** -i D:\Media\Record\channel02\2025\_01\_14\channel02\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -vn -acodec libmp3lame -q:a 2 D:\Media\Record\channel02\2025\_01\_14\output2.mp3

**ffmpeg** -i D:\Media\Record\channel03\2025\_01\_14\channel03\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -vn -acodec libmp3lame -q:a 2 D:\Media\Record\channel03\2025\_01\_14\output2.mp3

**ffmpeg** -i D:\Media\Record\channel04\2025\_01\_14\channel04\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -vn -acodec libmp3lame -q:a 2 D:\Media\Record\channel04\2025\_01\_14\output2.mp3

**//FFMPEG Per Channels Per Stream#**

map 0:a:N **<= N should be 0,1,2**

**ffmpeg** -i D:\Media\Record\channel01\2025\_01\_14\channel01\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -c copy D:\Media\Record\channel01\2025\_01\_14\output\_audio\_stream\_N.aac

**ffmpeg** -i D:\Media\Record\channel02\2025\_01\_14\channel02\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -c copy D:\Media\Record\channel02\2025\_01\_14\output\_audio\_stream\_N.aac

**ffmpeg** -i D:\Media\Record\channel03\2025\_01\_14\channel03\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -c copy D:\Media\Record\channel03\2025\_01\_14\output\_audio\_stream\_N.aac

**ffmpeg** -i D:\Media\Record\channel04\2025\_01\_14\channel04\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -c copy D:\Media\Record\channel04\2025\_01\_14\output\_audio\_stream\_N.aac

**//Convert Each Channel to WAV**

**ffmpeg** -i D:\Media\Record\channel01\2025\_01\_14\channel01\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -vn -acodec pcm\_s16le -ar 44100 -ac 2 D:\Media\Record\channel01\2025\_01\_14\output3.wav

**ffmpeg** -i D:\Media\Record\channel02\2025\_01\_14\channel02\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -vn -acodec pcm\_s16le -ar 44100 -ac 2 D:\Media\Record\channel02\2025\_01\_14\output3.wav

**ffmpeg** -i D:\Media\Record\channel03\2025\_01\_14\channel03\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -vn -acodec pcm\_s16le -ar 44100 -ac 2 D:\Media\Record\channel03\2025\_01\_14\output3.wav

**ffmpeg** -i D:\Media\Record\channel04\2025\_01\_14\channel04\_2025\_01\_14\_14\_40\_00.mp4 -map 0:a:0 -vn -acodec pcm\_s16le -ar 44100 -ac 2 D:\Media\Record\channel04\2025\_01\_14\output3.wav

ffmbc -i D:\Media\Record\channel04\2025\_01\_14\channel04\_2025\_01\_14\_14\_40\_00.mp4 -vn -acodec libmp3lame -qscale:a 2 D:\Media\Record\channel04\2025\_01\_14\output.mp6

ffmbc -codecs | grep mp3

ffmpeg -i D:\Media\Record\channel02\2025\_01\_14\channel02\_2025\_01\_14\_14\_40\_00.mp4 -vn -acodec libmp3lame D:\Media\Record\channel02\2025\_01\_14\output7.mp3

//The winner converter of .mp4 into .mp3 is **ffmbc**!

**ffmbc** -i D:\Media\Record\channel01\2025\_01\_14\channel01\_2025\_01\_14\_14\_40\_00.mp4 -vn -acodec libmp3lame D:\Media\Record\channel01\2025\_01\_14\ffmbc\_output.mp3

**ffmbc** -i D:\Media\Record\channel02\2025\_01\_14\channel02\_2025\_01\_14\_14\_40\_00.mp4 -vn -acodec libmp3lame D:\Media\Record\channel02\2025\_01\_14\ffmbc\_output.mp3

**ffmbc** -i D:\Media\Record\channel03\2025\_01\_14\channel03\_2025\_01\_14\_14\_40\_00.mp4 -vn -acodec libmp3lame D:\Media\Record\channel03\2025\_01\_14\ffmbc\_output.mp3

**ffmbc** -i D:\Media\Record\channel04\2025\_01\_14\channel04\_2025\_01\_14\_14\_40\_00.mp4 -vn -acodec libmp3lame D:\Media\Record\channel04\2025\_01\_14\ffmbc\_output.mp3

GMAIL SMTP SERVER:

@"{ 'username': 'rrenous@gmail.com', 'server': 'smtp.gmail.com', 'port': '587', 'password': **'wiqfziqfgjadluxy'**}";

{"Username":"ioana.banc@gmail.com","From":"ioana.banc@gmail.com","Server":"smtp.gmail.com","Port":587,"Password":"**yfzvkvxaqehquidd**","SendCredentials":true}

MY GMAIL APP PASSWORD

{"Username":"gilshalev2014@gmail.com","From":"gilshalev2014@gmail.com","Server":"smtp.gmail.com","Port":587,"Password":"**blucwjegdsqyalnc**","SendCredentials":true}

**//Create a JOB using CURL**

curl -L "http://manila:8894/intelligence/api/aijob/schedule" -H "Content-Type: application/json" -d "{\"Name\": \"run GIL-456\", \"Operations\": [\"DetectFaces\"], \"ChannelIds\": [1,2,3,4], \"BroadcastStartTime\": \"2025-04-27T14:15:00\", \"BroadcastEndTime\": \"2025-04-27T14:30:00\", \"RequestRule\": {\"Recurrence\": {\"Value\": \"Once\"}, \"Days\": 0}, \"Keywords\": [\"gaza\", \"trump\"], \"KeywordsLangauges\": [\"English\"], \"TranslationLanguages\": [\"English\"]}"  
  
**//MY AWS CREDENTIALS  
Secret access key:** +awCO+rbvkni8LLJhGrdoL6HwQKIBXhSuEDatgs5

**Access key:** AKIAYXUOCEGIIAUPO5OW

**COMPLILING PUBLISHING & DEPLOYING Whisper APPS**

**dotnet publish -c Release -r win-x64 --self-contained true -p:PublishSingleFile=false** //**For Transcriber!**  
**DON’T USE PublishSingleFile=true!!!**  
  
**Sometimes it's easier to copy the binaries from the Release folder to the destination !**

don't forget to **copy** the folder **runtimes!**  
//Although **p:PublishSingleFile=true, there's still a need to copy the runtimes folder as well (Cuda and x64)**

**dotnet publish -c Release -r win-x64 --self-contained true -p:PublishSingleFile=false**<= **For Service !**

(--self-contained true meaning the .NET will be installed too!. )

* 1. Make sure there is C:\ActusConfig with all the relevant files

ACTUS Converting .mp4 to .mp3  
  
**ffmpeg -i input.mp4 -af "pan=mono|c0=0.5\*c0+0.5\*c1" -vn -acodec libmp3lame -y output.mp3**

**Running fatser-whisper:**

**pip install faster-whisper pydub ffmpeg-python**

**Create a windows service:  
WHISPER  
1. dotnet publish -c Release -r win-x64 --self-contained**

**dotnet publish -c Release -r win-x64 --self-contained true /p:PublishSingleFile=true**

**//using sc.exe** run in cmd batch as administrator **2. sc create WhisperService binPath= "C:\IntelligenceApps\WhisperApps\WhisperService\WhisperService.exe"  
or  
New-Service -Name "WhisperService" `**

**-BinaryPathName " C:\IntelligenceApps\WhisperApps\WhisperService\WhisperService.exe" `**

**-DisplayName "Whisper AI Transcription Service" `**

**-Description "Exposes transcription API and background processing" `**

**-StartupType Automatic  
3. //a command to start the service in case of a crash after 10 seconds**

**sc failure WhisperService reset= 0 actions= restart/10000**

**SPEECHMATIX  
sc create SpeechmatixService binPath= "C:\IntelligenceApps\SpeechmatixService\SpeechmatixService.exe"**

**sc failure SpeechmatixService reset= 0 actions= restart/10000**

**NEUROTECHNOLOGY**

**sc create NeurotechnologyService binPath= "C:\IntelligenceApps\NeurotechnologyService\NeurotechnologyService.exe"**

**sc failure NeurotechnologyService reset= 0 actions= restart/10000  
  
4. Don't remember to start the service from services (or set it using powershell)**

<apiUrl>http://10.100.111.11:8893/Whisper/transcribe</apiUrl>  
  
**Mega credentials**[**https://mega.io**](https://mega.io)[tech@actusdigital.com](mailto:tech@actusdigital.com)

@ctus2kXX5

**WHISPER USING PYTHON**

pip install language\_tool\_python

pip install noisereduce numpy scipy  
  
running the commands:  
  
UAE teamsViewer  
190184048

gbdp6hch

TCRA - Tanzania

GHQ UAE ARMY – UAE

**Email to test:**

{"Username":"gilshalev@actusdigital.com","From":"gilshalev@actusdigital.com","Server":"smtp.office365.com","Port":587,"Password":"**5tgb6yhn7ujm@**","Enabled":true}  
The password is my outlook password!  
  
**Pikolo to test**:  
    "PikoloConfig": {

        "Url": "[http://tokyo.local:1234/api/"](http://tokyo.local:1234/api/%22),

        "Username": "testuser",

        "Password": "1234"

    }

**SNMP to test:**

 "SNMPConfig": {

        "Dest": "127.0.0.1",

        "Port": "162",

        "Community": "public",

        "ListenPort": "161"

    }

%appdata% <- TO ERASE THE ELECTRON LOCALSTORAGE!  
  
**MY AWS ACCOUNT Details:**Account ID: **600519614864**

Password ?? **: 1qa 3ed 5tg @**User Name**: GS**Password: **2wsx 4rfv 6yhn GS@**

docker run --rm -v C:\temp\Videos:C:\app\videos neurotechnology-detector:latest C:\app\videos\G7.mp4

docker run --rm -it -v C:\temp\Videos:C:\app\videos neurotechnology-detector cmd

docker run --rm -it -v C:\temp\Videos:C:\app\videos neurotechnology-detector NeurotechnologyDetector.exe C:\app\videos\G7.mp4

docker run --rm -it -v C:\temp\Videos:C:\app\videos -v C:\Development\SentiVeillance\_9\_1\_SDK\_2024-11-08\SentiVeillance\_9\_1\_SDK\Bin\Licenses:C:\app\bin\Licenses neurotechnology-detector NeurotechnologyDetector.exe C:\app\videos\G7.mp4

docker run --rm -it -v C:\temp\NeurotechnologyDetector\bin\Release:C:\app\bin\Release -v C:\temp\Videos:C:\app\videos -v C:\Development\SentiVeillance\_9\_1\_SDK\_2024-11-08\SentiVeillance\_9\_1\_SDK\Bin\Licenses:C:\app\bin\Licenses neurotechnology-detector -f C:\app\videos\G7.mp4

docker run --rm -it -v C:\Actus\_Temp\VideoFiles:C:\app\videos my-detector-app -f C:\app\videos\G7.mp4

**//Merge between Trunk and V10  
  
UserName: gilshalev  
Password: Actus2023**

https://actus.unfuddle.com/svn/actus\_actus/branches/Release/V10.0/Tools/ActusLicenseMgmt

https://actus.unfuddle.com/svn/actus\_actus/trunk/Tools/**ActusLicenseMgmt**

https://actus.unfuddle.com/svn/actus\_actus/trunk/src/**Services**

https://actus.unfuddle.com/svn/actus\_actus/trunk/Tools/**ActusConfigurationTool**

[https://actus.unfuddle.com/svn/actus\_actus/trunk/**Scripts**](https://actus.unfuddle.com/svn/actus_actus/trunk/Scripts)

**https://actus.unfuddle.com/svn/actus\_actus/trunk/src/act-ang9/**

[https://actus.unfuddle.com/svn/actus\_actus/trunk/**deployment**](https://actus.unfuddle.com/svn/actus_actus/trunk/deployment)

**Sefy's way to deploy a new release:**1. Copy it to C:\Install from [\\ActBuild\build](file:///\\ActBuild\build)  
2. Cd C:\Install  
3. Run  
Upgrade.bat Actus\_10.0.0.15890.7z //and not Upgrade.bat C:\Install\Actus\_10.0.0.15890.7z  
  
Remember to enable all the modules in the APP->**Settings->Users  
  
Starting from fresh:**1. Clear C:\Actus\Config  
2. Remove the MongoDB DB and its collections  
3. %appdata% <- TO ERASE THE ELECTRON LOCALSTORAGE!  
remove C:\Users\Administrator\AppData\Roaming\ActConfigTool

**Azure Speech Service – Realtime mode (west Europe)  
My Account: westeurope**

**Keys:**6tdZPEWJJtBsLVlNlWbdH5FDVonZJ3rNqfd4W0zsCmM7ofS06UmwJQQJ99BGAC5RqLJXJ3w3AAAYACOGwsJy

2BmaAJQzMwLcbwJRJ7c1K9Ir8jZrp4sHeNm2RhiDHazd2QZkU0jJJQQJ99BGAC5RqLJXJ3w3AAAYACOGnQBK  
  
**Actus Account - eastus:**e285874fa6e044d99773dd54b3ee70f5

**ALPR**  
alpr -c us --json C:\EDUCATION\Hackaton\openalpr-2.3.0-win-64bit\openalpr\_64\samples\us-1.jpg //--json will return bounding rectangles!

alpr -c eu --json C:\EDUCATION\Hackaton\openalpr-2.3.0-win-64bit\openalpr\_64\samples\eu-clip.mp4

alpr -c us --json C:\EDUCATION\Hackaton\VideoFiles\OPTIMIZED\output\_1280x720\_compressed.mp4

**FFMPEG – reducing size (resolution) and compressing  
  
Maintain Aspect Ratio Automatically:**  
ffmpeg -i input.mp4 -vf "scale=640:-1" -c:a copy output\_scaled.mp4

**Resize and Compress Simultaneously**

ffmpeg -i input.mp4 -vf "scale=640:360" -crf 28 -preset fast output\_compressed.mp4

ffmpeg -i input.mp4 -vf "scale=1280:-1" -crf 28 -preset fast output\_compressed.mp4//-1 means maintain the Aspect Ratio !

**Plate recognizer:  
API KEY: 822216ecdaac21eb7a63f873dcc9d4e10649ff4e  
  
  
License key: HZfd9BEzcX**

**curl -X POST "https://api.platerecognizer.com/v1/plate-reader/" -H "Authorization: 822216ecdaac21eb7a63f873dcc9d4e10649ff4e" -F image=** **C:\\EDUCATION\\Hackaton\\VideoFiles\\IL\_CAR.jpg -F regions=il**

**Actus Agent**

What did the shows say about Trump and peace yesterday?  
  
**Terms:**

* AI **Agent**, **MCP**, **Chatbot**
* **Semantic Kernel** is used for:

Complex **multi-step planning** **with AI agents** (like chaining summarization + email-sending),

Memory and context management,

**Plugins integration** (e.g., Gmail, Outlook, Jira)

* **Vector DBs**:
* They store Vectors (float arrays from embeddings),
* Metadata like transcript ID, timestamp, channel, etc.

You can **combine them**:

* MongoDB → stores transcript text, metadata
* Qdrant/Weaviate/Pinecone → indexes embedding vectors for fast **semantic search**

| **Feature** | **Qdrant** | **Weaviate** |
| --- | --- | --- |
| Type | Vector database | Vector database |
| Open Source | ✅ Yes | ✅ Yes |
| Hosting | Self-hosted or cloud | Self-hosted or cloud |
| Query method | Vector similarity (cosine, etc) | Vector + hybrid search (BM25) |
| Document store | Yes (with metadata & payloads) | Yes (uses GraphQL for querying) |
| Cost | Free if self-hosted | Free if self-hosted |

* **What is similarity Search**?  
  It's how you **retrieve semantically similar data** using **vector embeddings**
* You search in a DB for nearest vectors (by **cosine similarity**) to find transcript chunks with **similar meaning**, not just keywords

In case of **No Embeddings**

* Store just transcripts in MongoDB.
* At query time, load **all** transcripts into memory.
* Send them all (or large relevant portions) to OpenAI ChatCompletion.

**🔥 Downside:**

* Very **expensive** (OpenAI token cost)
* **Slow**
* Doesn't scale (e.g., thousands of transcripts)

A **vector database** — can do **cosine similarity to search natively**. (**performance**)

**🔍 AI Agent**

A program that can understand your goal, make decisions, and take actions (like calling APIs or searching data) to help you get what you want — often autonomously.

**🤖 Chatbot**

A conversational interface that responds to user input, typically with scripted or model-based replies. Most chatbots don’t plan or reason deeply — they just "chat."

**🧠 MCP (Memory, Code, Planning)**

A concept in advanced AI agents where:

* **Memory** = stores knowledge or facts
* **Code** = can run functions or access tools
* **Planning** = breaks big tasks into small steps

Used to build smart agents that can "think" before acting.

**📦 Vector Embeddings**

A way to turn text (or images, audio, etc.) into a list of numbers (a "vector") so it can be compared to other text **by meaning**, not exact words.

Example:  
"Obama" → [0.024, -0.88, 0.19, ...]

**📊 Vector Database (Vector DB)**

A special kind of database designed to **store and search vector embeddings**, usually by similarity (e.g., "find things most similar to this text").

Popular ones: Qdrant, Pinecone, Weaviate.

**📐 Cosine Similarity Search**

A math trick to compare two vectors (e.g., user query and transcript). It finds how "close in meaning" they are — angle close to 0 = more similar.

Used to rank and find best matches for a question.

**🧰 Semantic Kernel (Microsoft)**

An open-source .NET framework that helps you build **AI agents**. It connects:

* Prompts (like ChatGPT)
* Tools (like APIs)
* Memory (vector search, planner)

It’s like LangChain, but built for C#.

In the background (offline):

* Generate the **embedding**
* Store it in vector DB
* Link it to MongoDB doc

New LiveU servers URL:

http://manila.liveu.tv/actus5  
  
**QDRANT Commands**<http://localhost:6333/dashboard>  
curl -X PUT "http://192.168.152.9:6333/collections/transcripts" -H "Content-Type: application/json" -d "{\"vectors\": {\"size\": 1536, \"distance\": \"Cosine\"}}"

docker run -p 6333:6333 -p 6334:6334 qdrant/qdrant  
curl -X GET "http://192.168.152.9:6333/collections/transcripts"

curl -X DELETE http://192.168.152.9:6333/collections/transcripts

curl -X POST "http://192.168.152.9:6333/collections/transcripts/points?wait=true" -H "Content-Type: application/json" -d @payload.json  
curl -X POST "http://192.168.152.9:6333/collections/transcripts/points/scroll" -H "Content-Type: application/json" -d '{ "limit": 5}'

**In Demo Laptops**

1.Copy from Zip directly to C:\Actus\  
2.Run Demo\_Restore.bat "\actus\Snapshot\Full Demo"  
  
//Public Actus Servers  
<http://lab.actusdigital.com:9500/actus5/login>  
  
CLIENT MACHINE: (Anydesk)  
683464237  
Password: FSC@1234#