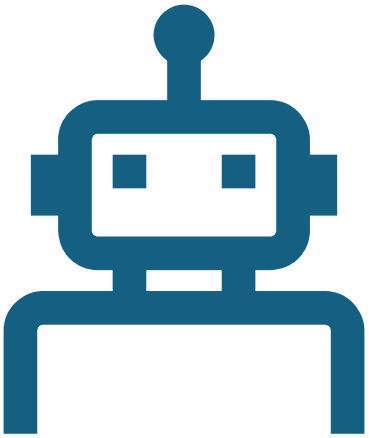
AI for Enterprise with Azure AI (Cognitive Services) Containers

Santosh Hari



Thank you TechBash 2023 Sponsors!

Santosh Hari

- Customer Engineer at Microsoft
- Azure MVP 2016-2020
- Co-organizer
  - Orlando .NET User Group (onetug.net)
  - Orlando Codecamp (orlandocodecamp.com)

@santoshhari

@desinole

@_s_hari

@santoshhari@hachyderm.io

I won't steal your job, but those learning to use AI will.

Frank Lazaro
An inventor, speaker and experienced executive in strategy, sales, marketing, technology, and innovation
19 articles
+ Follow

7, 2023

Agenda

- Azure Cognitive Services (now Azure AI Services)
- Quickstarts & building on quickstarts
- Cognitive Services in containers
- Container Ops
- Gnarly Issues
- Architecture and resources

What are azure cognitive services, now Azure AI services due to rebranding

I'll show you a few quickstarts and how to start to make them enterprise-y
By the way I have a talk on serverless and how to make it enterprise-y tomorrow

The enterprise aspect invetiably leads us to cognitive services being used in containers

We will talk about some aspects of container ops and some gnarly issues

Time permitting – more architectural discussions

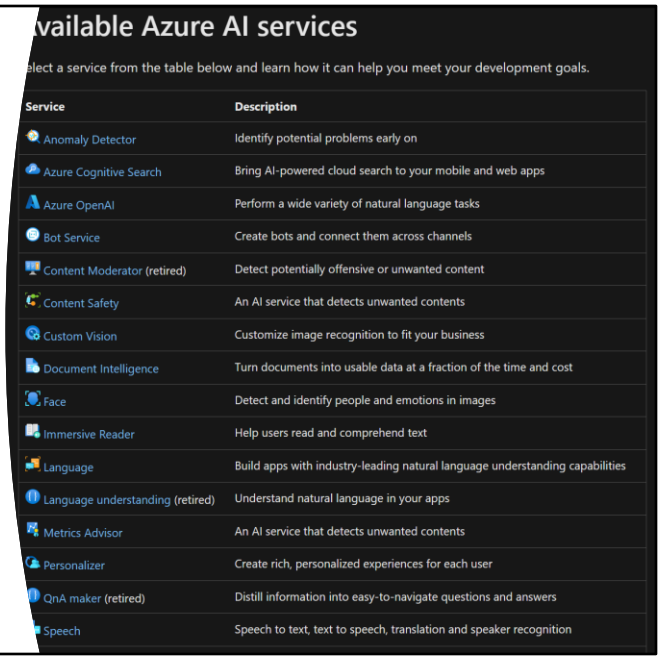
I will be walking through some code samples but not really writing it

Resources

- Code and slides
- https://bit.ly/cognitive-containers

Azure Cognitive Services

- Different than Open AI/Chat GPT
- Process exiting
- Collection of APIs and services



Who here is familiar with ChatGPT? Who is familiar with Cognitive Services?

collection of APIs and services that enable developers to build intelligent applications
without having direct AI or data science skills or knowledge

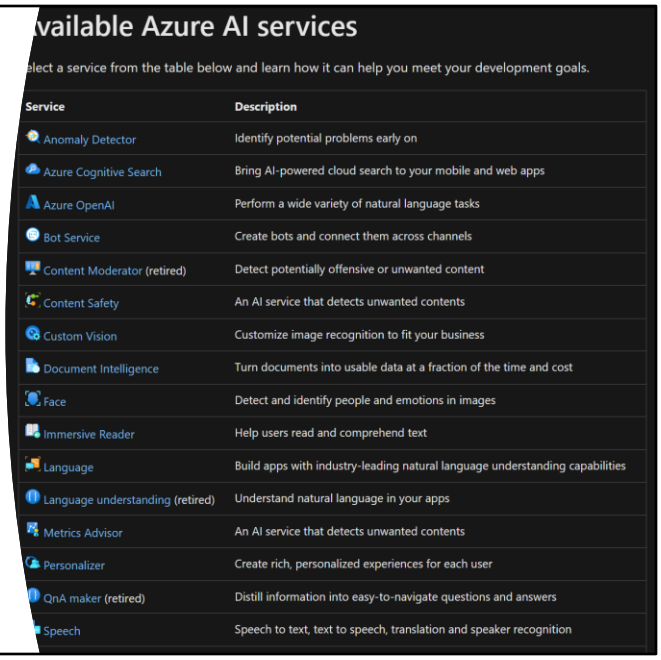
vision, speech, language, decision, and web search

Different than open ai/chatgpt type generative services

Process existing data

Azure Cognitive Services

- Vision, speech, language, decision, and web search
- Wide range of development, integration and deployment options



vailable Azure AI services

lect a service from the table below and learn how it can help you meet your development goals.

Service	Description
Anomaly Detector	Identify potential problems early on
Azure Cognitive Search	Bring AI-powered cloud search to your mobile and web apps
Azure OpenAI	Perform a wide variety of natural language tasks
Bot Service	Create bots and connect them across channels
Content Moderator (retired)	Detect potentially offensive or unwanted content
Content Safety	An AI service that detects unwanted contents
Custom Vision	Customize image recognition to fit your business
Document Intelligence	Turn documents into usable data at a fraction of the time and cost
Face	Detect and identify people and emotions in images
Immersive Reader	Help users read and comprehend text
Language	Build apps with industry-leading natural language understanding capabilities
Language understanding (retired)	Understand natural language in your apps
Metrics Advisor	An AI service that detects unwanted contents
Personalizer	Create rich, personalized experiences for each user
QnA maker (retired)	Distill information into easy-to-navigate questions and answers
Speech	Speech to text, text to speech, translation and speaker recognition

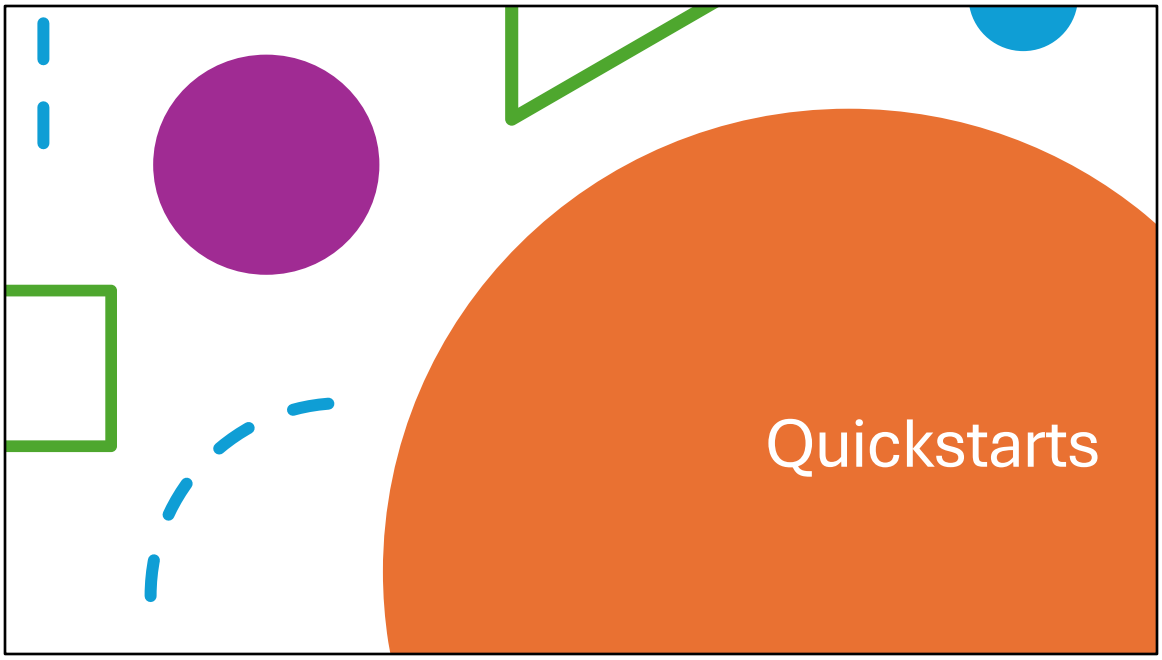
As you can tell from the image there is a wide range of services available

Vision, speech, language, decision, and web search

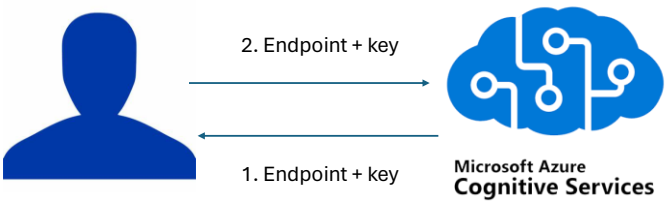
Most Azure AI services are available through REST APIs
and client library SDKs in popular development languages.

Development options
•Automation and integration tools like Logic Apps and Power Automate.
•Deployment options such as Azure Functions and the App Service.
•Azure AI services Docker containers for secure access.
•Tools like Apache Spark, Azure Databricks, Azure Synapse Analytics, and Azure Kubernetes Service for big data scenarios.

We clear on what Cognitive Services are?


Quickstarts

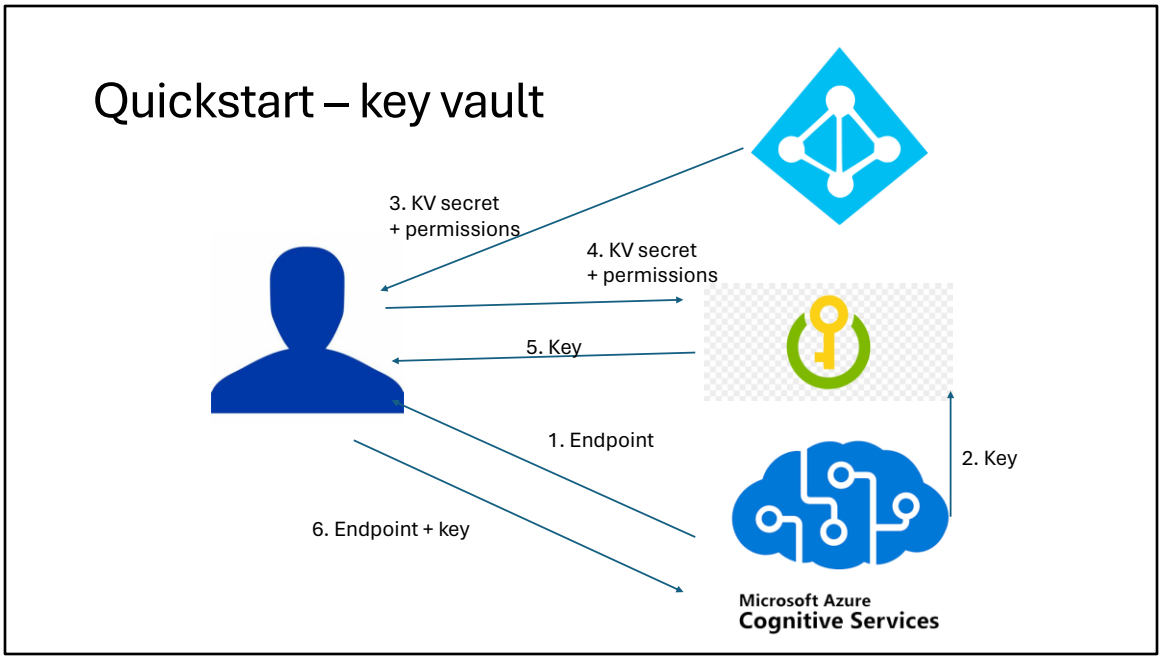
Quickstart – end user connects to services



When I say QuickStart this is what comes to mind, it's the simplest example

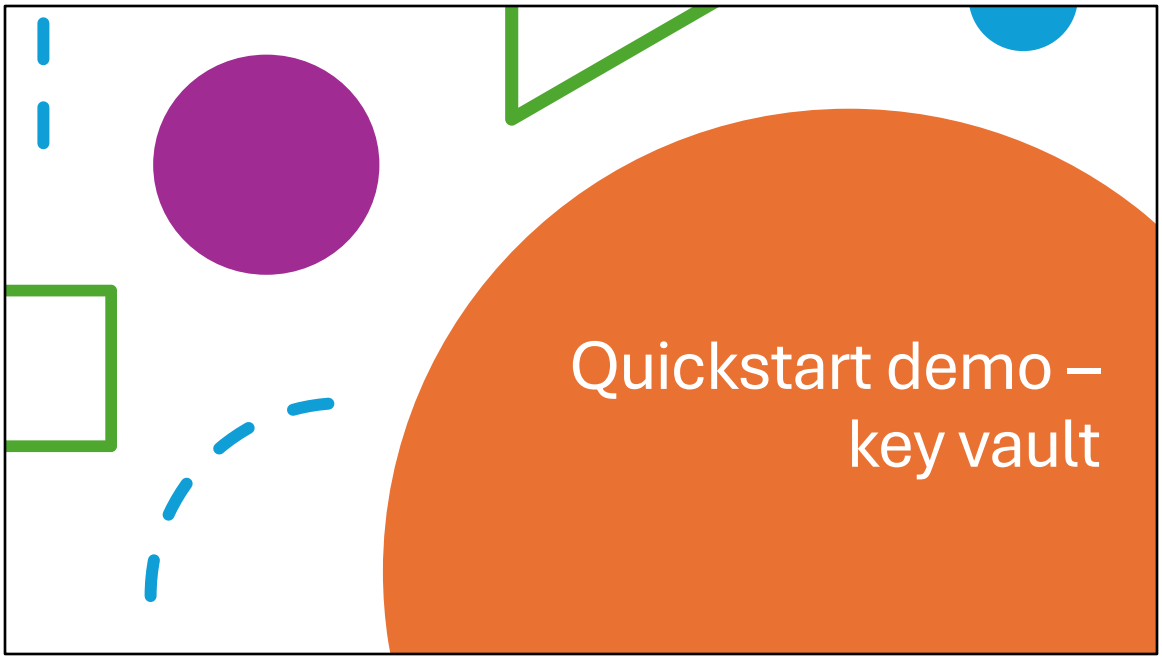
This is like saying I put my key under my front door mat and my friend walks in.
But anyone can walk into my building
You're basically handing over the keys and telling anyone on the web to come use this
Doesn't work in any enterprise situation.
Even if they're not worried about data exposure someone could run up a crazy bill


Quickstart Demo –
direct keys



So we take the solution to the enterprise customer and they find it unacceptable that an azure service keys

I recently underwent the tradition American home owners go through
Walking through homes to purchase one
In most homes they had a lockbox on or close to the door with the keys to the house
The realtor knew the lockbox combo and retrieved the keys to open the door
Using a keyvault is somewhat similar to this.
Instead of distributing the key freely you pick it up from a key vault which you have some secret code to access


Quickstart demo –
key vault


Improve your solution
for the enterprise

Rotate Keys

Use 1 of 2 keys in prod

Switch to key 2

Rotate 1

After a while repeat but switch to key 1

Automate the process

Each Azure AI services resource has two API keys to enable secret rotation. This is a security precaution that lets you regularly change the keys that can access your service,
protecting the privacy of your resource if a key gets leaked.

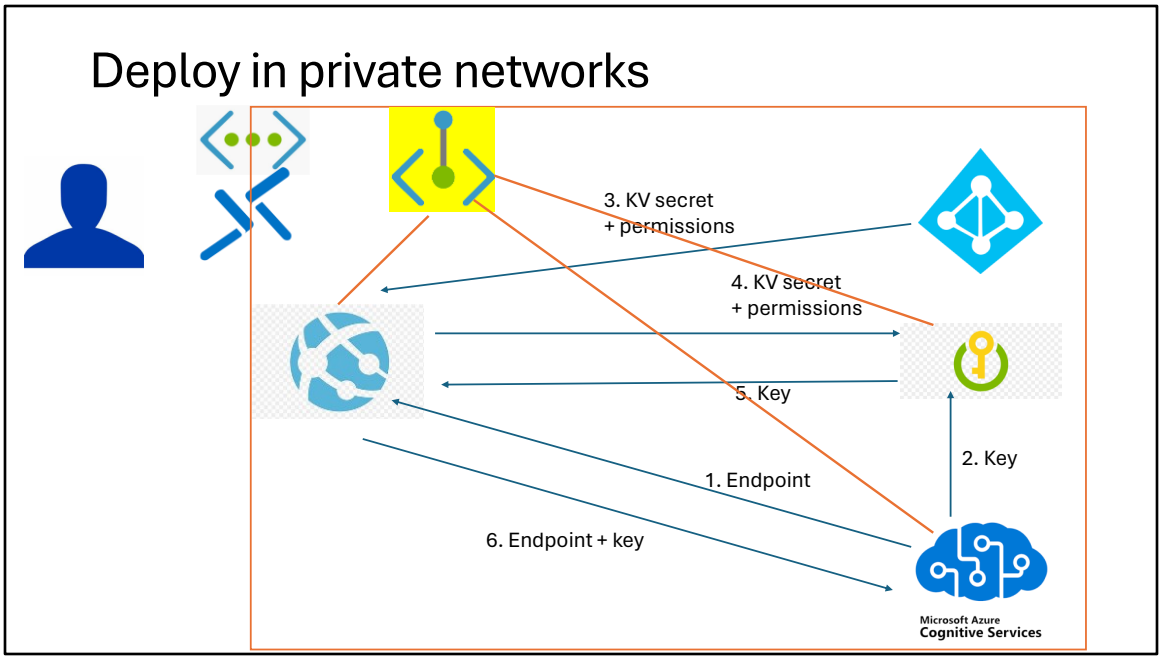
Use Environment Variable

```
using static System.Environment;

class Program
{
    static void Main()
    {
        // Get the named env var, and assign it to the value variable
        var value =
            GetEnvironmentVariable(
                "ENVIRONMENT_VARIABLE_KEY");
    }
}
```

```
# Assigns the env var to the value
[System.Environment]::SetEnvironmentVariable('ENVIRONMENT_VARIABLE_KEY', 'value', 'User')
```

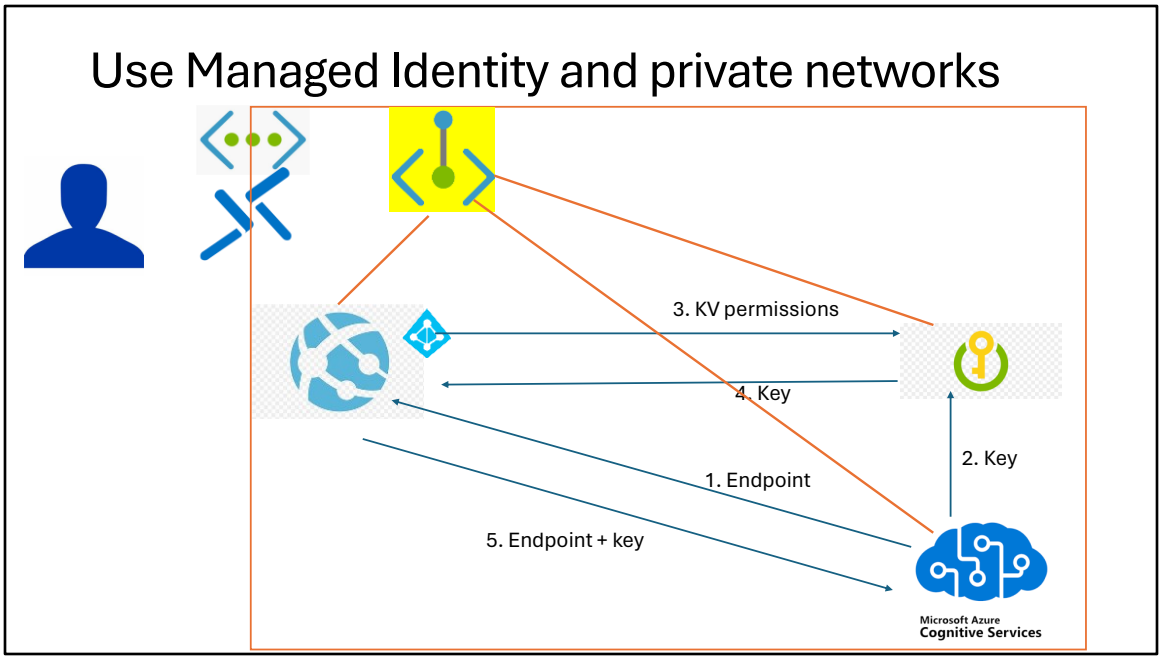
Use environment variables particularly for secrets in dev
Storing the SP secrets in here is a great option
You can store with powershell and retrieve with C#



In this scenario everything is completely within a private network in Azure
Communication between resources within the network happens through a service called private endpoint (marked in yellow)
End users get into the network through a bastion service

From where they access a azure app service
The azure app service follows the same process as the console app we say previously
Use secret to get key from key vault and then uses that to connect to the cognitive services

Building on the previous example, this is the equivalent of going into a house with a lockbox
But first you have to get into the gated community



In this scenario everything is completely within a private network in Azure
Communication between resources within the network happens through a service called private endpoint (marked in yellow)
End users get into the network through a bastion service

From where they access a azure app service
The azure app service has managed identity enabled which gives it special powers to get the key from key vault
without using a secret code.
then uses that to connect to the cognitive services

Building on the previous example, first you have to get into the gated community
Then you have some special powers think of it like a Bluetooth app to open the lockbox instead of a secret code

Cognitive Services in containers

Container Scenarios

- Private Networks in the cloud (ACI, AKS)
- On-prem
  - Hybrid
  - Disconnected
- Security, Compliance, Ops
- Subset

Speaking of private environments, you can not only deploy cognitive services in azure private network
But also scenarios like on-prem and edge computing
You can take the same APIs and make them available on-prem, edge and private networks
Using these containers gives you the flexibility to bring Azure AI services closer to your data
for compliance, security or other operational reasons.
Container support is currently available for a subset of Azure AI services.

Containers Demo

Containers with ACI

- Create ACI resource host
- Integrate with custom vision API container
- Check container health (ready/status/swagger)
- Invoke container operation
- Follow up on long-running operation with operation-location

Containers with AKS

Container	Minimum	Recommended
Read 3.2 2022-04-30	4 cores, 8-GB memory	8 cores, 16-GB memory
Read 3.2 2021-04-12	4 cores, 16-GB memory	8 cores, 24-GB memory


Container Ops &
Dealing with Gnarly
Issues

Networking

- Containers connect to billing service
- Port: 443
- Domains
  - *.cognitive.microsoft.com
  - *.cognitiveservices.azure.com
- Custom subdomains for resources are global and unique

Networking: You need to configure your network settings to

allow the containers to communicate with each other and with external services.

For example, you need to allowlist certain network channels for the containers to submit metering information for billing purposes[1].

You also need to set up a custom subdomain for your Cognitive Services account to access it from a virtual network[2].

Updates and versioning

- Responsibility of the customer
  - Updates
  - Compatibility
- Best practice: pull latest frequently
- Major version = breaking changes

Updating: You need to keep track of the latest versions of the containers and update them regularly to get the latest features and bug fixes.

You also need to test your application compatibility with the new versions before deploying them in production.

Containers are marked with standard Docker tags such as latest to indicate the most recent version.

Customers encouraged to pull the latest versions of containers as they're released.

Only the current version of the container is supported.

Major version changes indicate that there's a breaking change to the API signature.

Minor version changes indicate bug fixes, model updates, or new features that don't make a breaking change to the API signature

Diagnostics Container

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

```
docker run --rm mcr.microsoft.com/azure-cognitive-services/diagnostic \
eula=accept \
Billing={ENDPOINT_URI} \
ApiKey={API_KEY}
```

If you're having trouble running an Azure AI services container, you can try using the Microsoft diagnostics container. Use this container to diagnose common errors in your deployment environment that might prevent Azure AI services containers from functioning as expected.

Container Logs and Events

View Logs

Attach Output Streams

Get Diagnostics Events

To view logs from your application code within a container, you can use the az container logs command.

The az container attach command provides diagnostic information during container startup.
Once the container has started, it streams STDOUT and STDERR to your local console.

If your container fails to deploy successfully, review the diagnostic information provided by the Azure Container Instances resource provider.
To view the events for your container, run the az container show command:

Get container instance logs & events - Azure Container Instances | Microsoft Learn

Container Health

Purpose	URL
Home page	http://localhost:5000/
Liveness and Readiness probes	http://localhost:5000/ready
	http://localhost:5000/status
Docs	http://localhost:5000/swagger

The container provides a home page.

K8s liveness and readiness probes:

1. Requested with GET, this URL provides a verification that the container is ready to accept a query against the model.
2. Verify is api-key used to start container is valid

Docs and test queries

Container Status Messages

- Valid
- Invalid
- Mismatch
- CouldNotConnect
- OutofQuota
- BillingEndpointBusy
- ContainerUseUnauthorized
- [ERROR] Failed to download: context deadline exceeded
- Unknown

Container Availability Matrix

	GA	Disconnected mode
Anamoly Detector	☑	
LUIS	☑	☑
Key Phrase Extraction	☑	☑
Text Lang Detect	☑	☑
Sentiment Analysis	☑	☑
Text Analysis for Health	☑	
Custom Named Entity Recognition	(preview)	
Translator	(gated)	☑
Speech to text	☑(gated)	☑
Custom Speech to text	☑(gated)	☑
Neural Speech to text	☑(gated)	☑
Speech Language Detection	(gated)	☑
Read OCR	☑(gated)	☑
Spatial Analysis	(preview)	

Disconnected

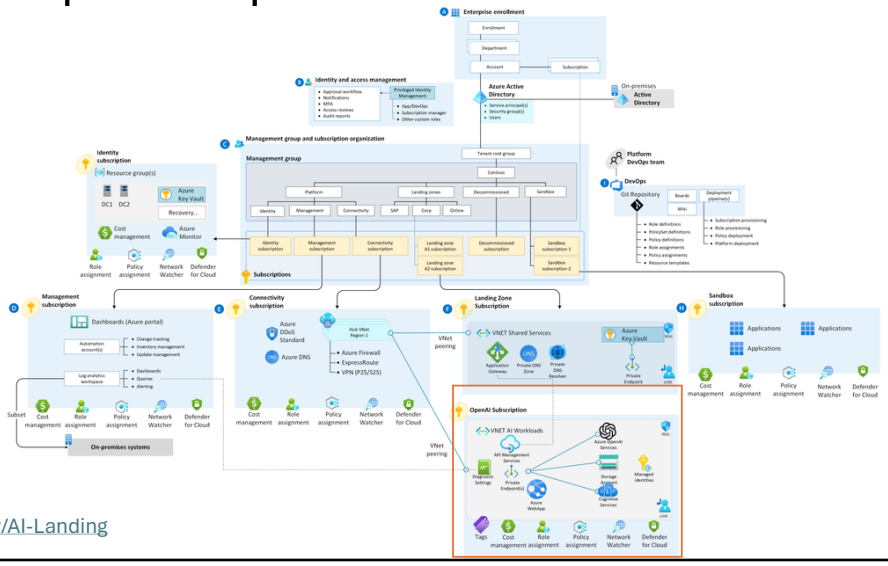
- On-prem
  - Request special access
  - Product key instead of API key
  - Billing proxy container periodically syncs with Azure
  - Limited time only (90 days)
  - No SLA

- Azure AI Speech
  - Speech to text
  - Custom Speech to text
  - Neural Text to speech
- Text Translation (Standard)
- Azure AI Language
- Sentiment Analysis
- Key Phrase Extraction
- Language Detection
- Azure AI Vision - Read
- Document Intelligence

Compare scenarios

	PaaS (public)	AKS (private network)	On-prem disconnected
Internet Connection	100%	Can be less than 100%, needed for billing only	Occasional, needed for billing only
Ease of client access	Any client over internet	Clients within private network only over private endpoint	On-prem resources only
Data Control	Will fail compliance	Offers privacy and sovereignty	Controlled by customer
Infrastructure Management	All advantages of PaaS	K8s container orchestration complexity	Controlled by customer
Versioning	Connect and use	Container and API version matching complexity	Container and API version matching complexity
Pricing	Consumption and commitment	Consumption and commitment	Commitment only

Sample enterprise Architecture - Overall



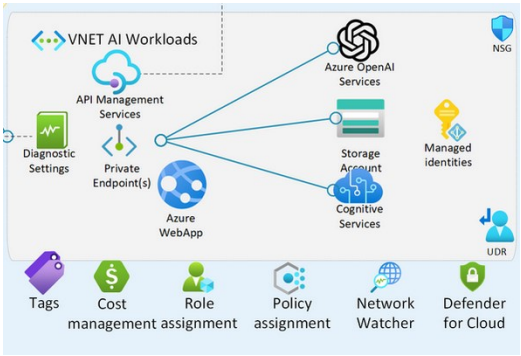
https://bit.ly/AI-Landing

Azure Landing Zones provide a solid foundation for your cloud environment.
When deploying complex AI services,
using a Landing Zone approach helps you manage your resources in a structured, consistent manner, ensuring governance, compliance, and security are properly maintained.

This is an example of an Azure AI landing zone.
As you can see there are at least a dozen blocks
Enterprise enrollement, ID Management Srevices, DevOps,
You also have different subcriptions each with a purpose
Management, Connectivity, Sandbox, Identity and so on

As you can see the AI services are deployed in the subscription in that little red box at the bottom * click *

Sample enterprise Architecture – AI subscription



As you can see the subscription hosting the AI services (red box from previous page)
Has all these resources and services
Cognitive services, storage account, managed identity, web app, private endpoint, API Management

Resources

Azure AI services containers

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

Docker Container support for Azure Cognitive Services

https://hub.docker.com/_/microsoft-azure-cognitive-services

Azure OpenAI Landing Zone reference architecture

https://techcommunity.microsoft.com/t5/azure-architecture-blog/azure-openai-landing-zone-reference-architecture/ba-p/3882102

Get Started with Cognitive Services

https://github.com/MicrosoftLearning/AI-102-AIEngineer/blob/master/Instructions/01-get-started-cognitive-services.md

Deploy Azure landing zones

https://learn.microsoft.com/en-us/azure/architecture/landing-zones/landing-zone-deploy

Thank you!
I hope this information on Azure Cognitive Services was interesting and informative to you.
I would like to end with the call to action
We all start with learning apps using quickstarts and there is nothing wrong with starting that way
But I encourage you to think beyond the quickstarts and start building your apps with a bigger picture in mind
With a little bit of thought and foresight you can build successful AI apps that fit and function in an enterprise environments

Resources

- Code and slides
- https://bit.ly/cognitive-containers

Thank you TechBash 2023 Sponsors!