

Azure AI and Terraform: Exploring our Options

Jake Walsh & Nicholas Chang



Please note – the views/opinions in this presentation are entirely our own. This presentation will not be kept updated Global Azure 2024 (April 2024) – so may be outdated if downloaded afterwards.

If in any doubt, please check latest documentation and MS Links for updated info!

Who we are...

Nicholas Chang



*Senior Platform Engineer,
Kainos*

@nick_cloudops
nicholaschangblog.com



Jake Walsh



*Senior Solution Architect,
CDW UK*

@jakewalsh90
jakewalsh.co.uk



<https://www.meetup.com/azure-community-enthusiasts/>

Agenda

- **Brief overview of Azure AI Services and authentication**
- **Azure Terraform – a *very* brief overview, and example deployment**
- **Getting Started**
- **Code run through and Demo**
- **Azure OpenAI Landing Zone Reference**

Capabilities of Azure AI Services?



Azure OpenAI Service



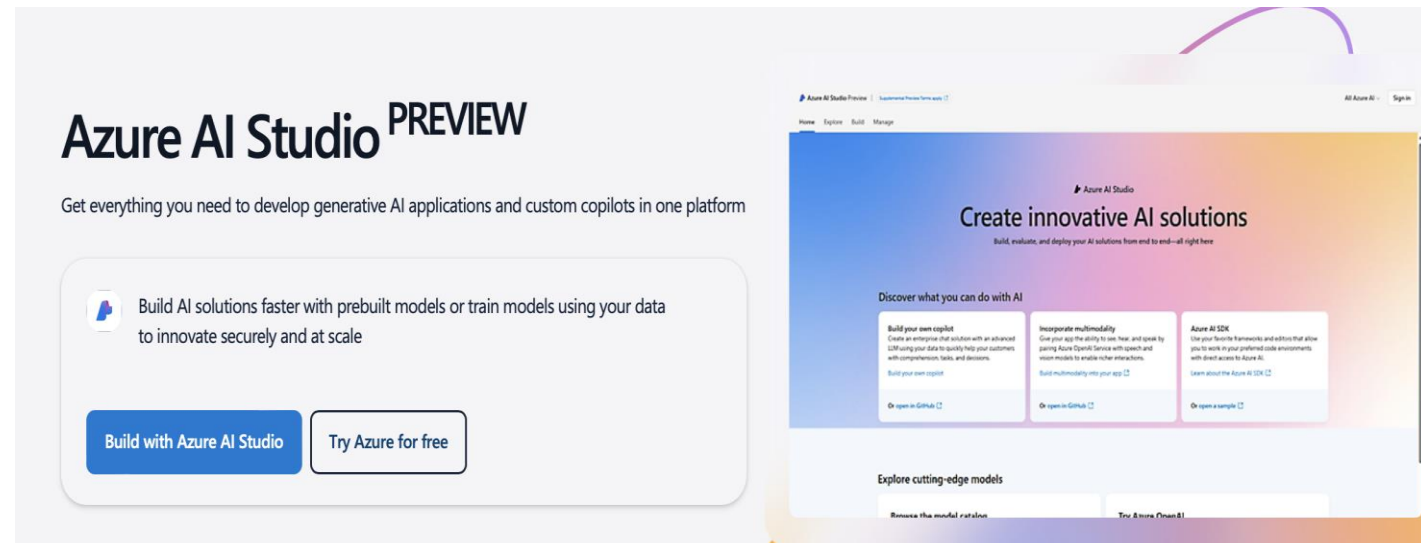
Azure AI Search



Azure AI Document Intelligence



Azure AI Content Safety



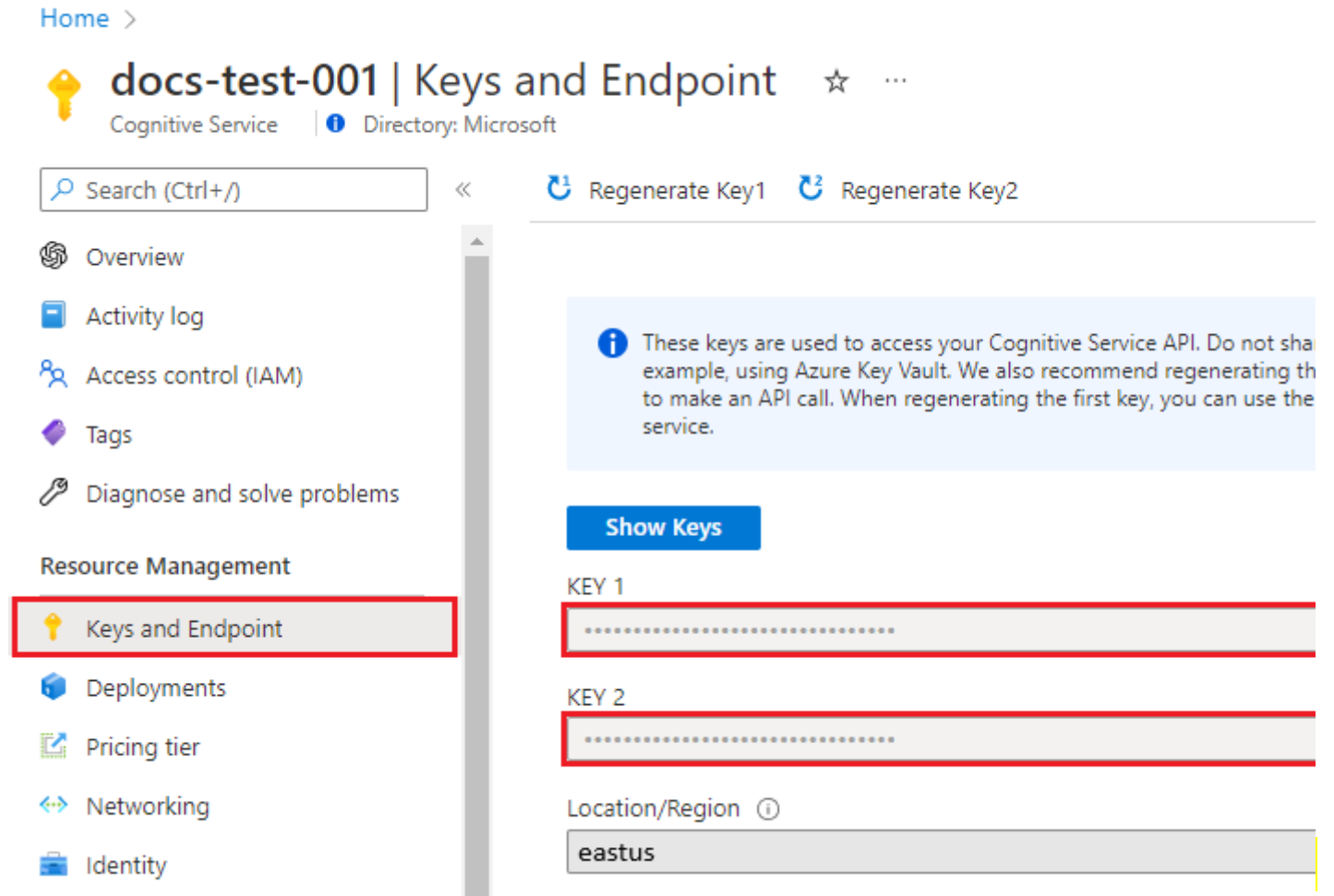
Other AI Services

How to interact with AI Services

- **APIs:** Most AI services provide APIs that to interact with them.
- **SDKs:** Many AI services also offer Software Development Kits (SDKs) that make it easier to interact with their APIs

Usually via a Key (Similar to Azure Storage Account Keys), Managed Identity, or Service Principal – depending on the type of access.

<https://learn.microsoft.com/en-us/azure/ai-services/openai/tutorials/embeddings?tabs=python-new%2Ccommand-line&pivots=programming-language-python>



The screenshot shows the Azure portal interface for a Cognitive Service named 'docs-test-001'. The left sidebar contains a navigation menu with options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource Management, Keys and Endpoint (highlighted with a red box), Deployments, Pricing tier, Networking, and Identity. The main content area is titled 'docs-test-001 | Keys and Endpoint' and includes a search bar, 'Regenerate Key1', and 'Regenerate Key2' buttons. A blue information box states: 'These keys are used to access your Cognitive Service API. Do not share these keys, for example, using Azure Key Vault. We also recommend regenerating them to make an API call. When regenerating the first key, you can use the service.' Below this is a 'Show Keys' button. The 'KEY 1' and 'KEY 2' sections each show a redacted key value (represented by dots) with a red box highlighting the redaction. The 'Location/Region' is set to 'eastus'.



Studios, SDKs, and APIs

Cognitive Services | Azure OpenAI Studio - Preview

Test User
test-resource (South Central US, S0)

TU

Privacy & cookies

Azure OpenAI Studio

Get started with Azure OpenAI

Perform a wide variety of natural language tasks with Azure OpenAI, including copywriting, summarization, parsing unstructured text, classification, and translation.

Explore examples for prompt completion

Summarize Text

Summarize text by adding a 'tldr:' to the end of a text passage.

[Learn more](#)

Classify Text

Classify items into categories provided at inference time.

[Learn more](#)

Natural Language to SQL

Translate natural language to SQL queries.

[Learn more](#)

Generate New Product Names

Create product names from examples words.

[Learn more](#)

Manage your deployments and models

Experiment with prompt completions

Try out the completions endpoint by writing a prompt and generating a response. Set different parameters values to adjust how the model responds.

[Go to playground](#)

Customize a model with fine-tuning

Fine-tune a custom model to increase reliability for a wide variety of use cases while decreasing costs and speeding up processing times.

[Start fine-tuning a custom model](#)

Manage deployments in your resource

Create deployments to explore the model capabilities.

[Go to Deployments](#)

Manage performance results

Upload datasets to use when creating custom models, and view performance and fine-tune results from training and validation data.

[Go to File management](#)

Azure OpenAI supported programming languages

Article • 12/18/2023 • 2 contributors

[Feedback](#)

In this article

- [Programming languages](#)
- [Next steps](#)

Azure OpenAI supports the following programming languages.

Programming languages

[Expand table](#)

Language	Source code	Package	Examples
C#	Source code	Package (NuGet)	C# examples
Go	Source code	Package (Go)	Go examples
Java	Source code	Artifact (Maven)	Java examples
JavaScript	Source code	Package (npm)	JavaScript examples
Python	Source code	Package (PyPi)	Python examples

Two Common Scenarios for Deployment



Extending in-house or line of business applications with Azure AI Services.



Developing new Applications based on Azure AI Services.



Azure OpenAI: Real-World Example

- Build a QnA chatbot to help developers using Azure OpenAI, Slack and Jira
- Building a Private Open AI to queries SharePoint Documentation
- Python application used Llama index and Streamlit. Upload documents and ask questions based on the content of the uploaded documents.
- Solution to used AI GPT Model to review Pull request

How can Terraform help?



Creating baseline infrastructure – using IAC in a repeatable way.



Deploying similar infrastructure blocks to multiple locations or deployments – e.g. Prod / Dev / Test etc.



Allowing Azure AI Infrastructure to be managed and developed in the same way as other software – e.g. Git Repos, Pipelines, Version Control, Releases etc.

Integration with Terraform



- Terraform is an **Infrastructure as Code** Software tool, that can interact with a wide range of Platforms and Environments, using Providers.



- Can be used in both Cloud and On-Premises environments. Can be used to combine on-premises and Cloud, or Cloud and Cloud for example.
- Terraform comes in 3 main varieties:
 - Community Edition – *I will be using this to demo today!*
 - Terraform Cloud
 - Terraform Enterprise

Terraform Options

- Various options – directly create resources via the Registry or use Modules.
 - Registry – lists all possible resources that can be natively created in Terraform.
 - Modules – creates pre-defined configurations that provide flexibility.
- Use of AzAPI if there is no registry/module option – allows Terraform to

<https://registry.terraform.io/modules/Azure/openai/azurerm/latest>

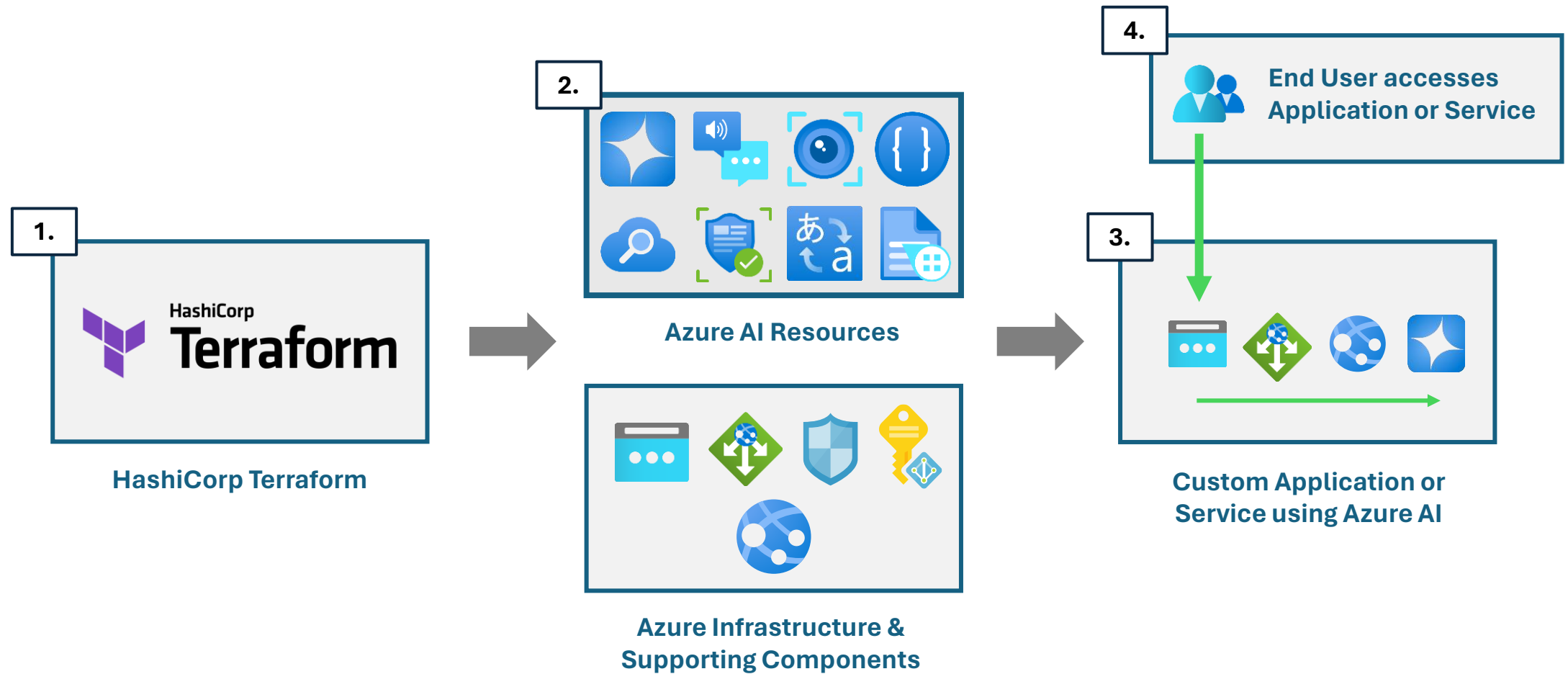
```
resource "azurerm_resource_group" "example" {
  name      = "example-resources"
  location  = "West Europe"
}

resource "azurerm_cognitive_account" "example" {
  name                = "example-account"
  location             = azurerm_resource_group.example.location
  resource_group_name = azurerm_resource_group.example.name
  kind                = "Face"

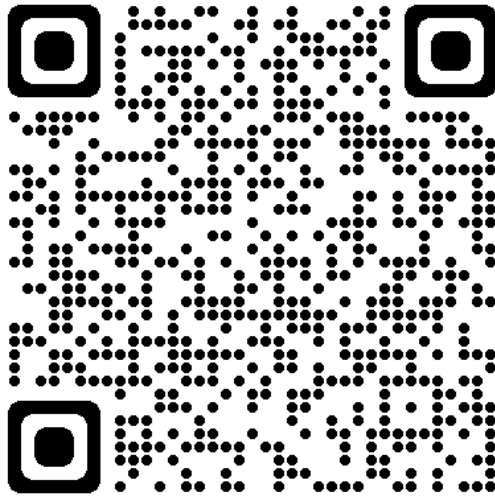
  sku_name = "S0"

  tags = {
    Acceptance = "Test"
  }
}
```

An Example using Terraform for Deployment



Demo!



<https://github.com/jakewalsh90/Terraform-Azure/tree/main/Azure-OpenAI-Demo-1>



Code

Blame

105 lines (103 loc) · 3.38 KB

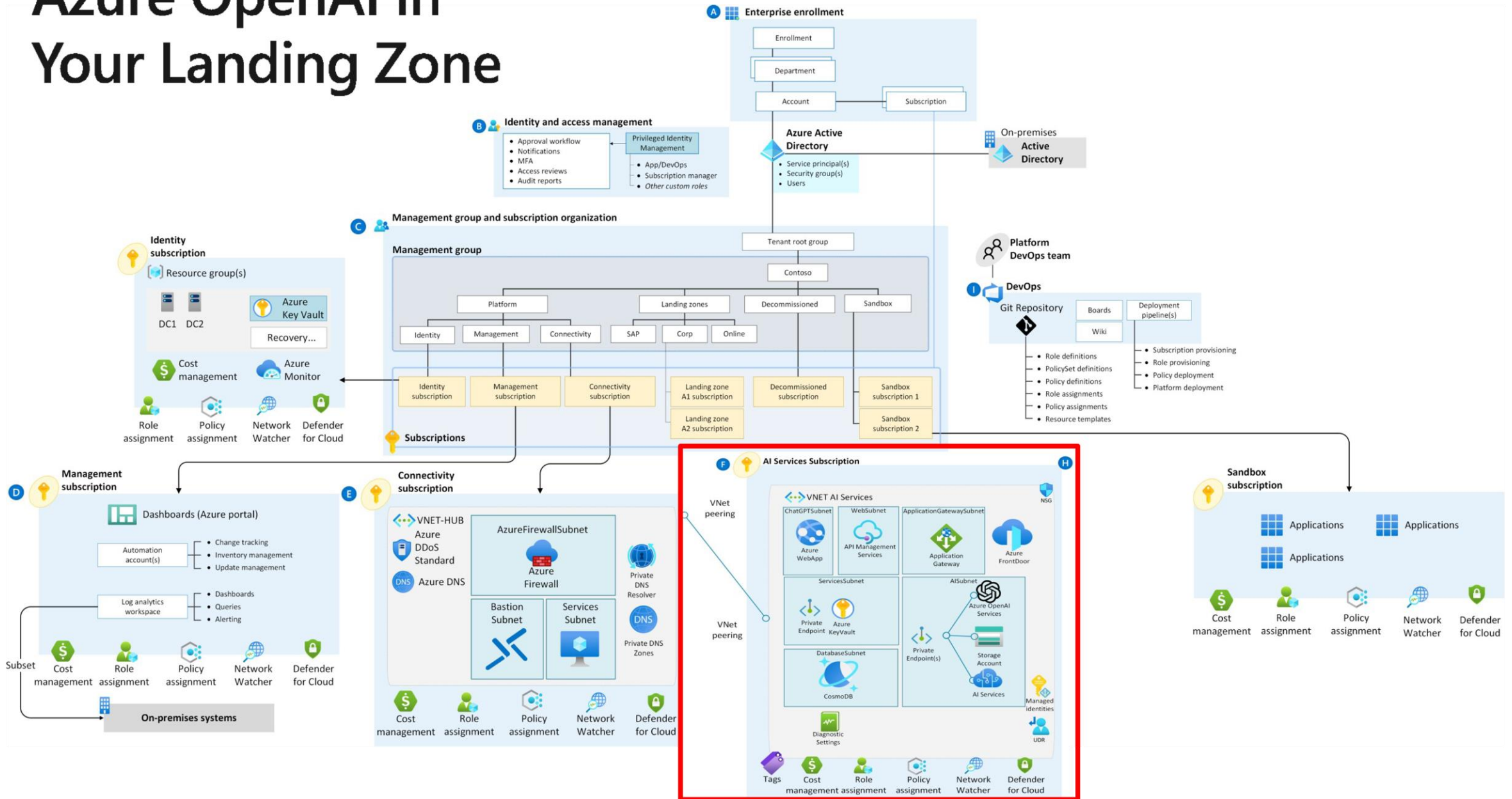
```
1  # Resource Group
2  resource "azurerm_resource_group" "rg1" {
3      name      = "rg-${var.region}-aoai"
4      location = var.region
5  }
6  # Random IDs for OAI Resources
7  resource "random_id" "cognitive" {
8      byte_length = 6
9  }
10 resource "random_id" "pn-cognitive" {
11     byte_length = 6
12 }
13 # Cognitive Services - without Private Networking
14 resource "azurerm_cognitive_account" "cognitive1" {
15     count                = var.privatenetworking ? 0 : 1
16     name                 = "oai-${random_id.cognitive.hex}"
17     location             = azurerm_resource_group.rg1.location
18     resource_group_name = azurerm_resource_group.rg1.name
19     sku_name             = "S0"
20     kind                 = "OpenAI"
21 }
22 # Cognitive Deployments - without Private Networking
23 resource "azurerm_cognitive_deployment" "gpt-35-turbo" {
24     count                = var.privatenetworking ? 0 : 1
25     name                 = "gpt-35-turbo"
26     cognitive_account_id = azurerm_cognitive_account.cognitive1[0].id
27     model {
28         format = "OpenAI"
29         name   = "gpt-35-turbo"
30     }
31 }
```

Getting Started with Azure OpenAI

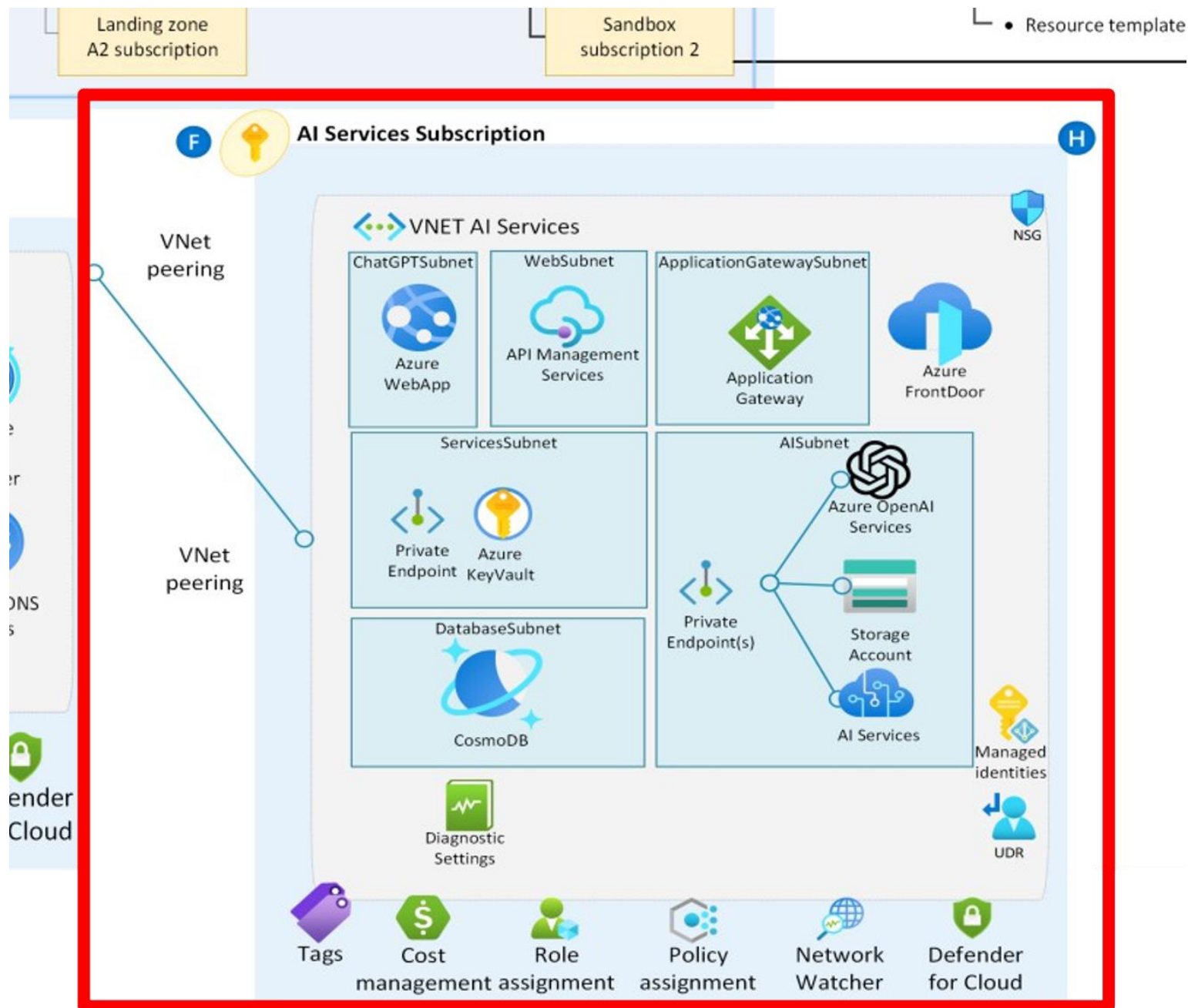
- An Azure subscription with access granted to the Azure OpenAI service - <https://aka.ms/oai/access>
- An existing Azure OpenAI resource and deployment of a chat model (e.g., [gpt-35-turbo-16k](#) or [gpt-4](#))
- An existing Azure Search instance and index (for ‘[bring-your-own-data](#)’ scenarios)



Azure OpenAI in Your Landing Zone



Azure OpenAI in Your Landing Zone





Further Reading

- <https://jakewalsh.co.uk/deploying-securing-and-monitoring-azure-openai-with-terraform/>
- <https://aka.ms/oai/access>
- <https://github.com/jakewalsh90/Terraform-Azure/tree/main/Azure-OpenAI-Demo-1>



Thank You!

Azure AI and Terraform: Exploring our Options

Jake Walsh & Nicholas Chang