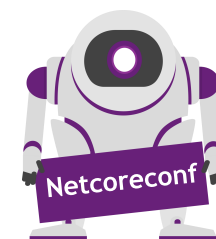


2024

ACA: The best, the rest, the rare

Sergio Navarro
Alejandro García





Sergio Navarro Pino

Tech Lead & Developer en Analyticalways

Twitter X : @snavarropino





Alejandro Garcia Miravet

DevOps Engineer en Prosegur

Twitter X : @alexmiravet





#netcoreconf

01

Networking

02

Storage options

03

Service add-ons

04

Dynamic sessions

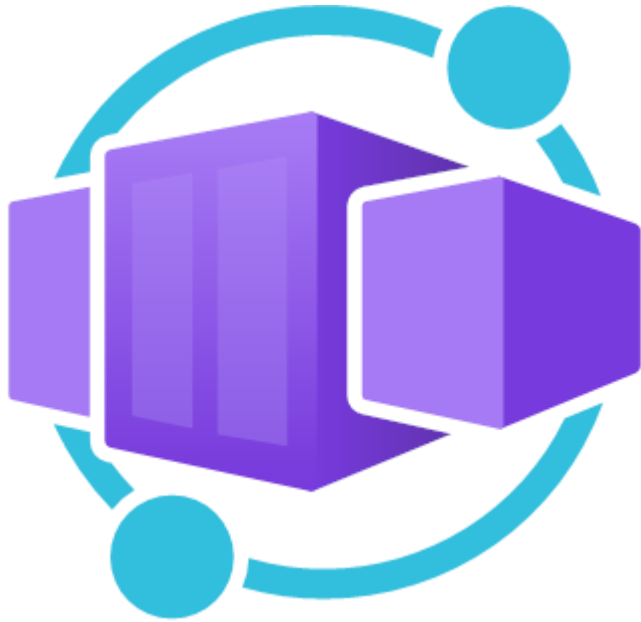
05

Observabilidad

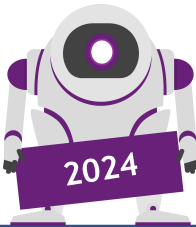


Azure Container Apps

“Azure Container Apps es una plataforma sin servidor que permite mantener menos infraestructura y ahorrar costes al ejecutar aplicaciones en contenedores” - Microsoft



ULTRA MEGA MANAGED K8S

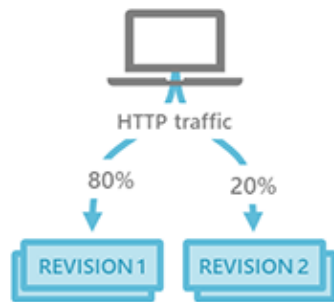


Casos de uso



Azure Container Apps: Example scenarios

PUBLIC API ENDPOINTS



HTTP requests are split between two versions of the container app where the first revision gets 80% of the traffic, while a new revision receives the remaining 20%.

AUTO-SCALE CRITERIA

Scaling is determined by the number of concurrent HTTP requests.

BACKGROUND PROCESSING



A continuously-running background process that transforms data in a database.

AUTO-SCALE CRITERIA

Scaling is determined by the level of CPU or memory load.

EVENT-DRIVEN PROCESSING

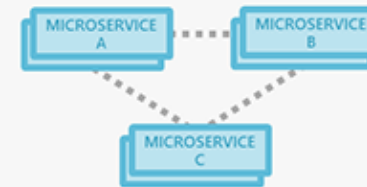


A queue reader application that processes messages as they arrive in a queue.

AUTO-SCALE CRITERIA

Scaling is determined by the number of messages in the queue.

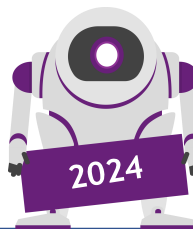
MICROSERVICES



Deploy and manage a microservices architecture with the option to integrate with Dapr.

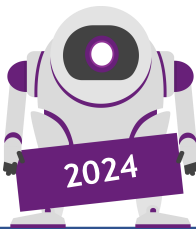
AUTO-SCALE CRITERIA

Individual microservices can scale according to any KEDA scale triggers.





- Containers Apps y Jobs son agrupados dentro de un ACA Environment.
- Cada ACA tiene asociada una network que puede ser auto creada por Azure o una que proporcionemos nosotros.
- Este ACA Environment puede ser de dos tipos:
 - Consumption Only
 - Workload Profile : Consumption y Dedicated
- Dependiendo del tipo de ACA Environment elegido y el tipo de network creada tenemos unas características de red diferentes.

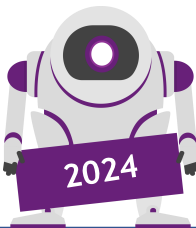


Networking



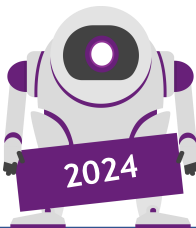
imgflip.com

JAKE-CLARK.TUMBLR



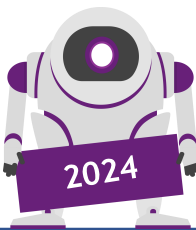
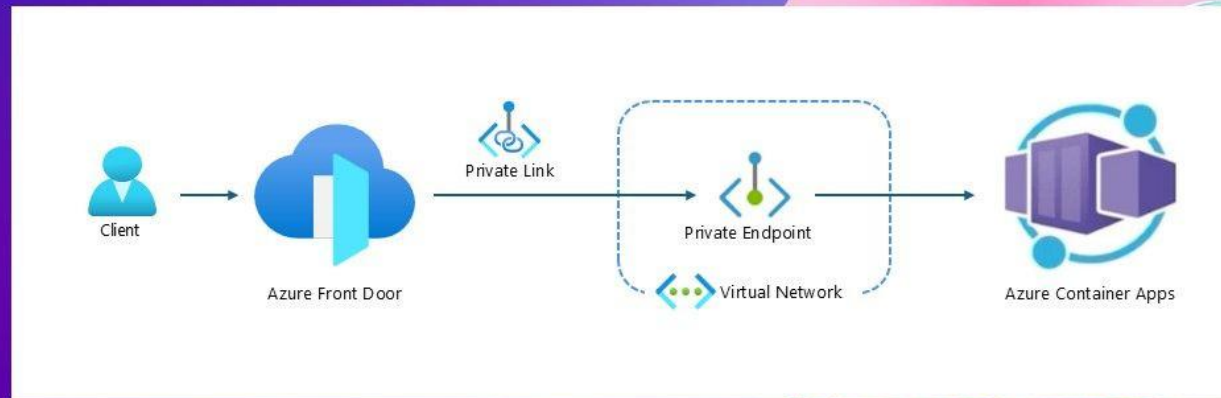
Networking

Tipo entorno	Tipo de red	Características
Workload profile	Custom Vnet	<p>Integración con Application Gateway.</p> <p>Uso de network security groups.</p> <p>Comunicarse con recursos detrás de Private endpoints.</p> <p>Hacer uso de NAT Gateway para el tráfico saliente.</p> <p>Acceso desde redes VPN/Express Route a ACA.</p> <p>Acceso a ACA a través de Private Endpoints.</p>

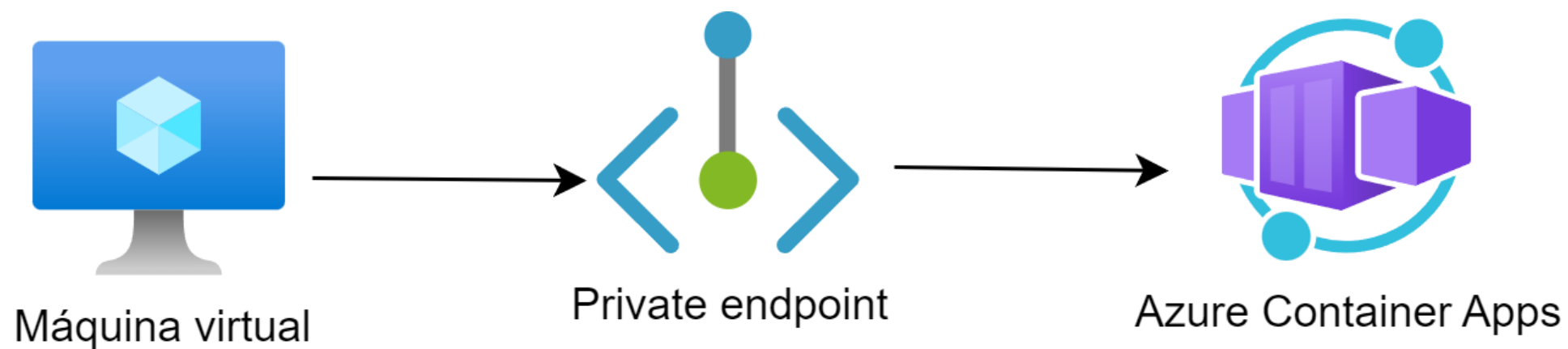


Networking

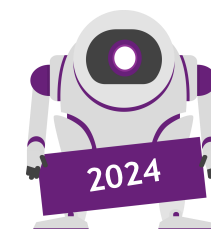
Azure Container Apps Private Endpoints



Networking



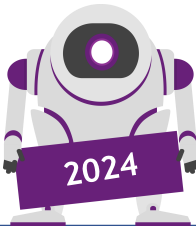
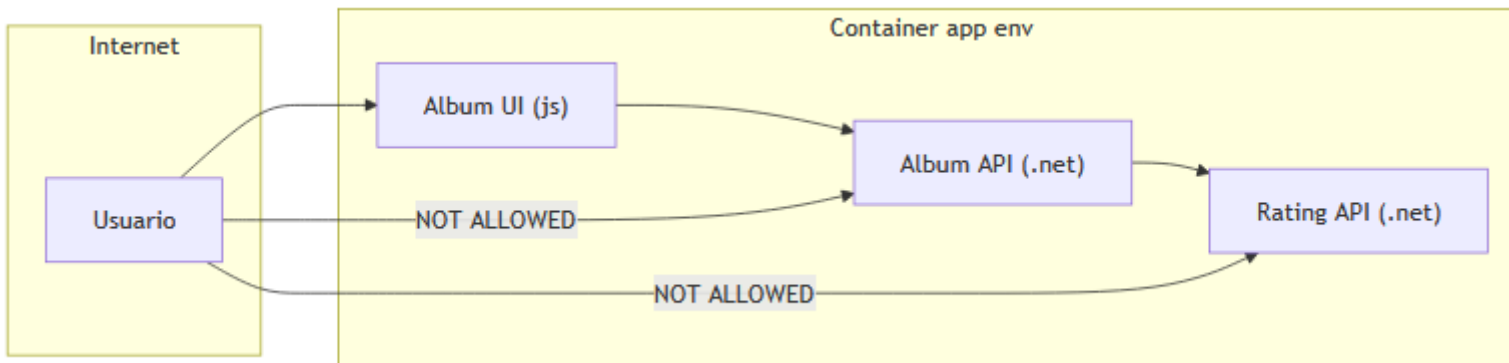
DEMO



Networking

Ingress

- External / internal
- No private endpoints required



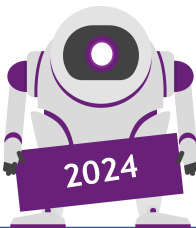
Storage options



Ephemeral Storage

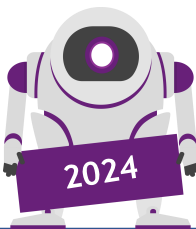
- Nivel de contenedor
- Nivel de replica: Compartido por todos los contenedores

vCPU	Almacenamiento local
0,25 o menos	1 GiB
0,5 o menos	2 GiB
1 o menos	4 GiB
Más de 1	8 GiB



Storage options

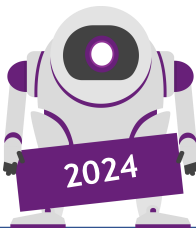
Con Ephemeral Storage cuando la réplica o contenedor muere...



Storage options

Azure Files

- Los datos se almacenan externamente en Azure Files.
- Los datos resisten a reinicios/muertes de contenedores.
- Poder usar un mismo recurso de Azure Files en múltiples contenedores...
- Montar múltiples recursos de Azure Files en un mismo contenedor.
- Que los datos puedan ser usados por aplicaciones externas o de terceros.

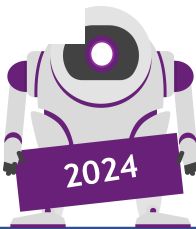


Services add-on (En Preview)



- Permite hacer uso de servicios de manera fácil que necesitan nuestras apps.
- Usar en desarrollo, no apto para producción.
- Evita el tiempo dedicado en instalarlo y configurarlo de manera manual.

Servicios
Kafka
MariaDB
PostgreSQL
Redis
Milvus
Qdrant
Weaviate

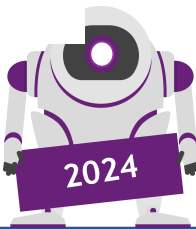


Service connector (En Preview)



- Permite conectar a servicios externos que necesita tu app de manera rápida.
- Container apps controla la configuración de la red y autenticación necesarias.
- Las configuraciones son establecidas a través de variables de entorno.

```
az containerapp create `
  --name myapp `
  --image mcr.microsoft.com/k8se/samples/sample-service-redis:latest `
  --ingress external --target-port 8080 `
  --bind myredis ←
  --environment "$ENVIRONMENT" `
  --resource-group "$RESOURCE_GROUP" `
  --query properties.configuration.ingress.fqdn
```



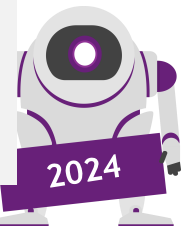
Deployment options



```
- stage: 'CD_Production'
  displayName: 'CD Code Production'
  dependsOn: 'CD_Infrastructure_Production'
  jobs:
  - deployment: production_Deployment
    environment: BackgroundTasksScheduler-Prod
    strategy:
      runOnce:
        deploy:
          steps:
            - task: AzureCLI@2
              displayName: "Update Container App image"
              inputs:
                azureSubscription: 'sp-azdevops-platform-production'
                scriptType: pscore
                scriptLocation: inlineScript
                inlineScript: |
                  az extension add --name containerapp --upgrade
                  az containerapp update --name $(ProductionContainerAppName) \
                                         --resource-group $(ProductionContainerAppResourceGroupName) \
                                         --image $(ProductionContainerRegistry)/$(ImageName):$(Build.BuildId)
```



```
steps:
- task: AzureContainerApps@1
  inputs:
    azureSubscription: 'my-subscription-service-connection'
    containerAppName: 'my-container-app'
    resourceGroup: 'my-container-app-rg'
    imageToDeploy: 'myregistry.azurecr.io/my-container-app:$(Build.BuildId)'
```



Latest is evil!

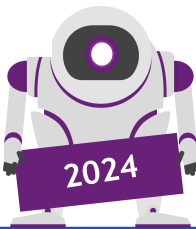
📘 Important

If you're building a container image in a separate step, make sure you use a unique tag such as the build ID instead of a stable tag like `latest`. For more information, see [Image tag best practices](https://learn.microsoft.com/en-us/azure/container-registry/container-registry-image-tag-version).



<https://learn.microsoft.com/en-us/azure/container-registry/container-registry-image-tag-version>

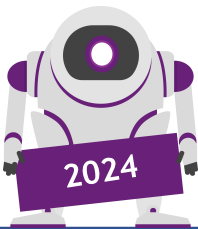
The 'latest' tag is mutable and can lead to unexpected errors if the image changes. A best practice is to use an immutable tag that maps to a specific version of an application Pod. This policy validates that the image specifies a tag and that it is not called 'latest'.



Dynamic sessions



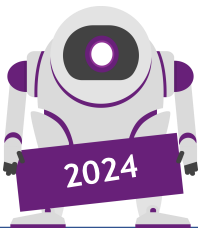
- Dynamic sessions es un servicio dentro de ACA que te permite de manera rápida tener entornos aislados donde poder ejecutar código potencialmente inseguro.
- Entre sus grandes bondades destacan:
 - Poder ejecutar código inseguro de manera aislada.
 - Creación rápida de entornos.
 - Acceso seguro a estos entornos.
 - Muy escalable.
 - Totalmente administrado.



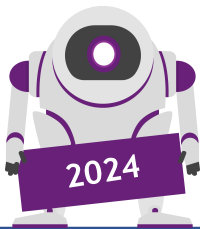
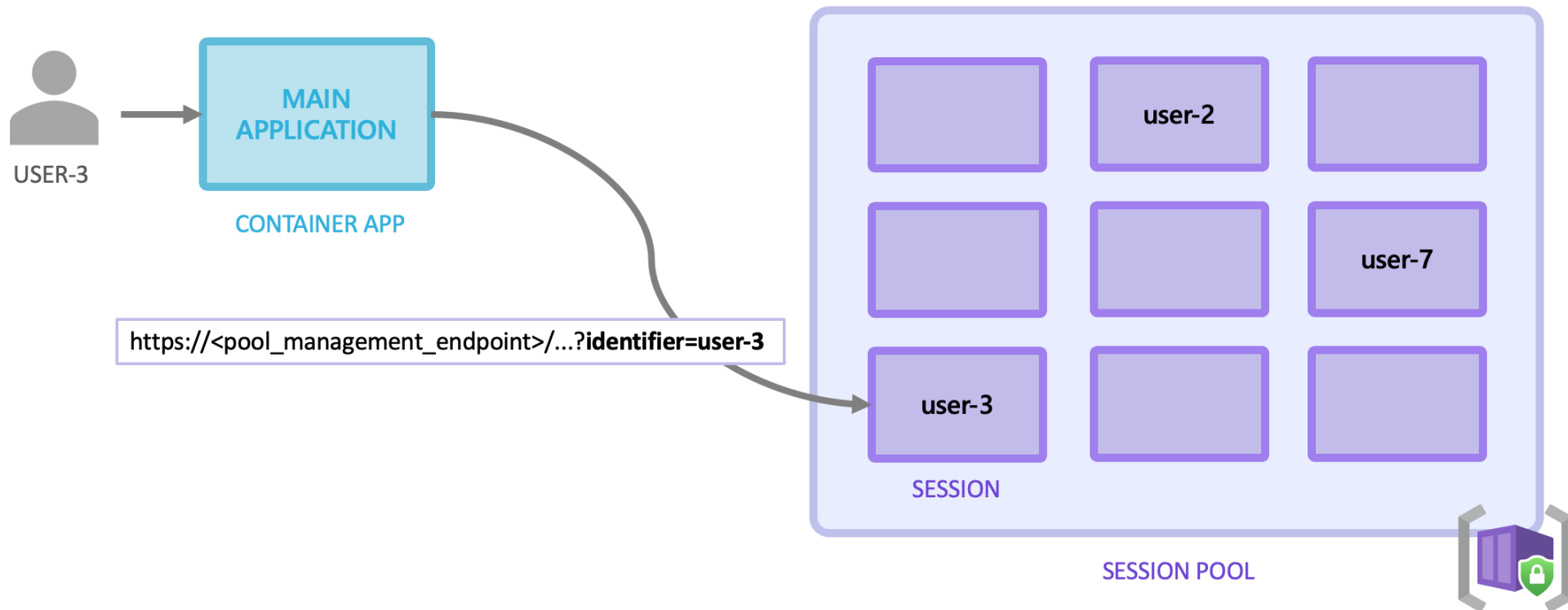
Dynamic sessions



- Disponemos de dos tipos de formas ejecutar Dynamic sessions:
 - **Code interpreter:** Container predefinido con librerías preinstaladas para ejecutar código de Python o de Javascript.
 - **Custom container:** Crea tu propio container con lo que necesitas.



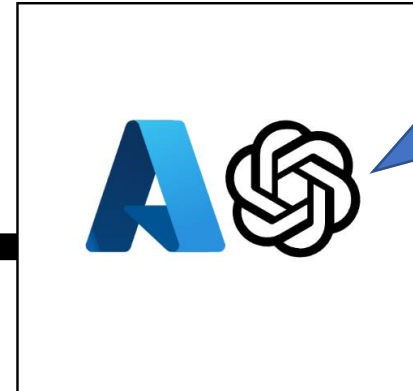
Dynamic sessions



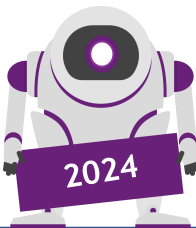
Dynamic sessions



Majo ¿Cómo consulto la hora del sistema en Python?



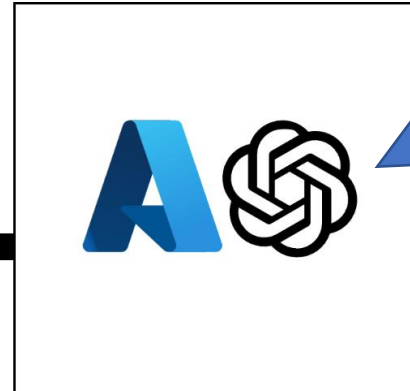
En Python, puedes consultar la hora del sistema utilizando el módulo datetime. Aquí te dejo un ejemplo de cómo hacerlo



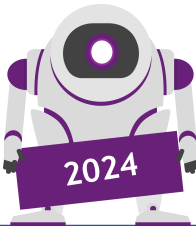
Dynamic sessions



Majo. Ejecuta
obtener la hora
del sistema en
Python.



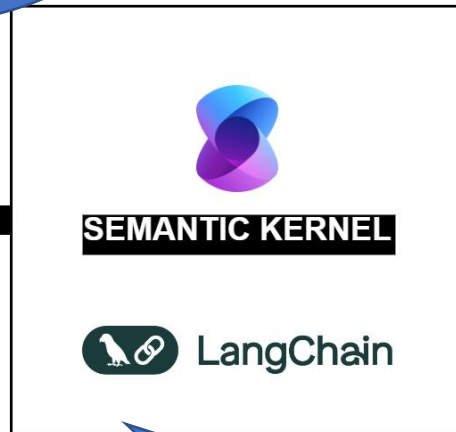
No puedo ejecutar
código directamente en
este entorno. Sin
embargo, te puedo
guiar para que lo
ejecutes en tu propia
máquina.



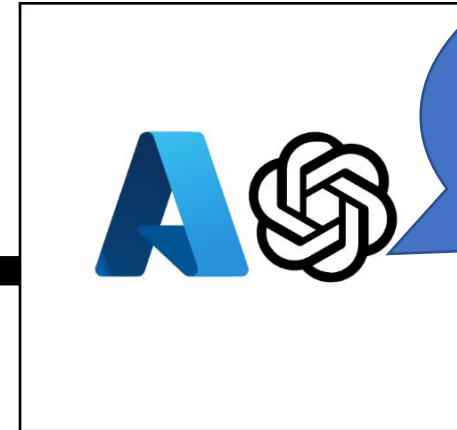
Dynamic sessions



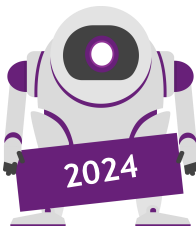
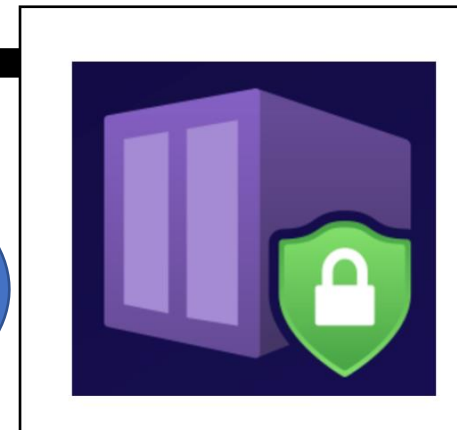
Majo. Ejecuta
obtener la hora
del sistema en
Python.



Session
Dynamic majo,
ponte al lío y
ejecútame el
código



Aquí esta el código de
como hacerlo.



Scalability

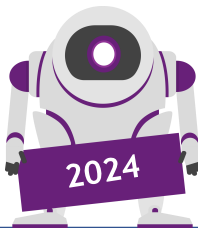
Horizontal Scaling:

- Replica-based scaling
- Scale to 0
- Keda integration:
 - CPU and memory
 - Azure Service
 - Azure Storage
 - Custom Prometheus

vCPUs (cores)	Memory
0.25	0.5Gi
0.5	1.0Gi
0.75	1.5Gi
1.0	2.0Gi
1.25	2.5Gi
1.5	3.0Gi
1.75	3.5Gi
2.0	4.0Gi
2.25	4.5Gi
2.5	5.0Gi
2.75	5.5Gi
3.0	6.0Gi
3.25	6.5Gi
3.5	7.0Gi
3.75	7.5Gi
4.0	8.0Gi



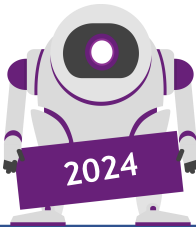
defined SKUs
limited to 2 cpu / 4 Gb



Observabilidad

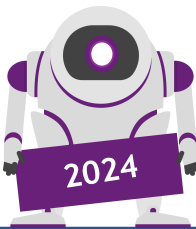
En Azure Container Apps disponemos de las siguientes características:

- Stream de registro para ver en tiempo real los problemas que surjan.
- Poder conectarse a la consola de los contenedores.
- Métricas y Alertas de Azure Monitor.
- Acceso al registro de aplicaciones y del sistema.
- Habilitar Aspire en nuestros Container Apps.

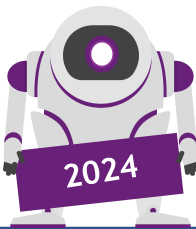
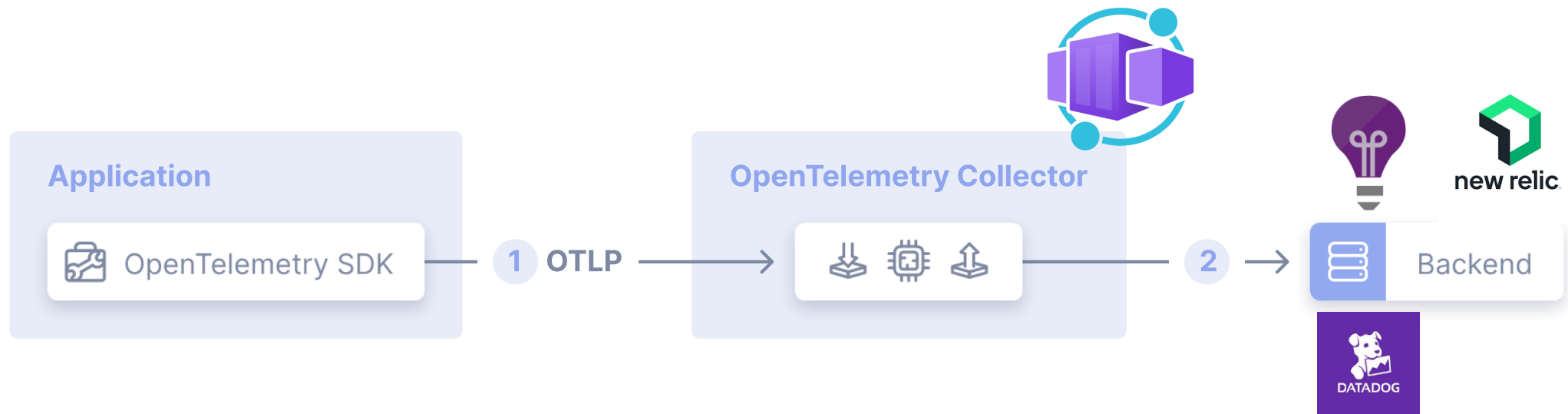


Observabilidad

- Uso de Log analytics para consultar de forma efectiva los logs de aplicaciones y el sistema.
- Agente/ collector Otel out-of-the-box para poder enviar logs, métricas y trazas a backends tales como App Insights, Datadog, o cualquier backend de tipo Otel



Observabilidad



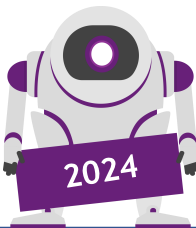
¿SOLO PUEDE QUEDAR UNO?



AKS



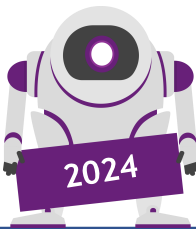
APP SERVICES



¿SOLO PUEDE QUEDAR UNO?



AZURE CONTAINER INSTANCES





SPONSORS





/netcoreconf



@netcoreconf



@netcoreconf



Extra – Github runner en Container App Job

