

## Tutorial de Azure Kubernetes Service (AKS)

### Descripción general de Azure Kubernetes Service (AKS)

Kubernetes es una plataforma de orquestación de contenedores [de código abierto](#) que ayuda a automatizar el proceso de implementación, gestión y escalado de aplicaciones en contenedores. Puedes agrupar grupos de hosts que ejecutan contenedores Linux, y Kubernetes te ayuda a gestionarlos de forma fácil y eficiente.

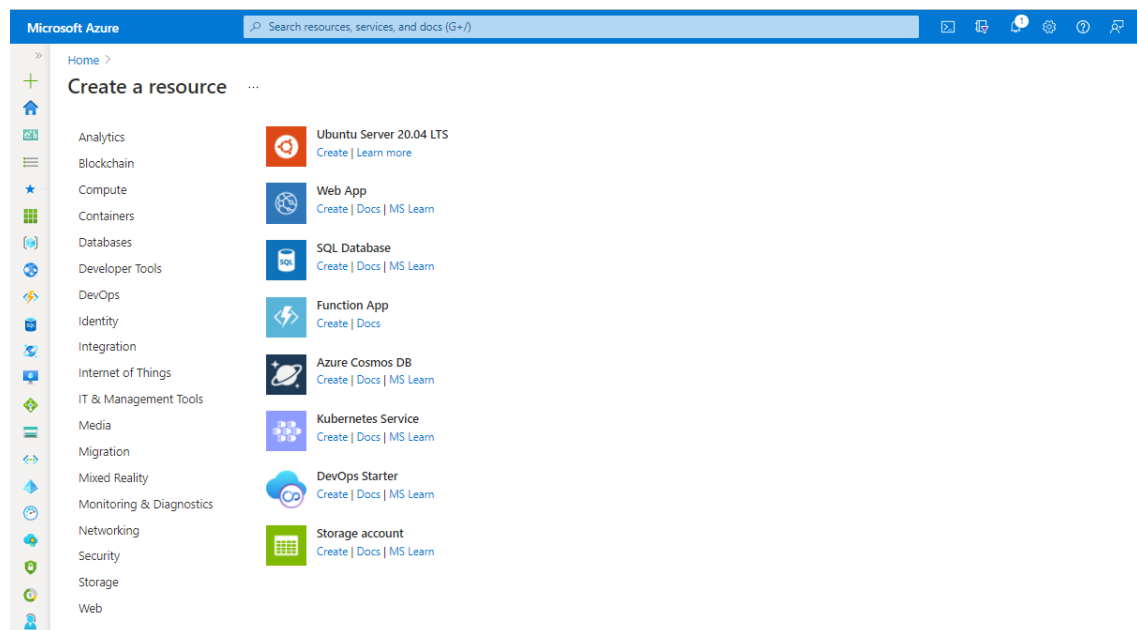
Azure Kubernetes Service es un servicio administrado de Kubernetes que facilita la creación, implementación y actualización de clústeres. También ofrece vistas básicas de sus clústeres de AKS. Azure Kubernetes Service (AKS) ofrece Kubernetes sin servidor, una experiencia integrada de integración y entrega continuas (CI/CD), y seguridad y gobernanza de nivel empresarial. Reúna a sus equipos de desarrollo y operaciones en una única plataforma para crear, entregar y escalar aplicaciones rápidamente y con confianza.

### Requisito previo

- Suscripción de Azure para la implementación de AKS
- VS Code para desarrollo e implementación
- Conocimientos básicos sobre Kubernetes

### Paso 01: Implementar Azure Kubernetes Service en la suscripción

Primero, vaya al portal y busque **AKS Azure Kubernetes Service** . Aunque también podrá verlo en las opciones rápidas.



Después, se le pedirá que complete los datos básicos en el asistente. En este artículo, le presentaremos algunas opciones nuevas.

Microsoft Azure

Home > Create a resource >

## Create Kubernetes cluster

Basics Node pools Authentication Networking Integrations Tags Review + create

Azure Kubernetes Service (AKS) manages your hosted Kubernetes environment, making it quick and easy to deploy and manage containerized applications without container orchestration expertise. It also eliminates the burden of ongoing operations and maintenance by provisioning, upgrading, and scaling resources on demand, without taking your applications offline. [Learn more about Azure Kubernetes Service](#)

**Project details**

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource group \*  [Create new](#)

**Cluster details**

Cluster preset configuration

**Standard (\$\$)**  
Quickly customize your cluster by choosing the preset configuration applicable to your scenario. Depending on the selection, values of certain fields might change in different tabs. You can modify these values at any time. [View all preset configurations](#)

Kubernetes cluster name \*

Region \*

Availability zones

[Review + create](#) < Previous Next: Node pools >

**Configuración predefinida del clúster :** Ofrece un SKU de nodo de máquina virtual preconfigurado y seleccionado según la carga de trabajo. Como se muestra en la imagen, tenemos diferentes opciones.

Create Kubernetes cluster

Choose your scenario to view and apply the recommended configurations suited to your needs. The settings in the table below will be updated to the specified values based on your selection. All other cluster settings will remain unchanged. [Learn more](#)

|                            | Standard (\$\$) | Dev/Test (\$) | Cost-optimized (\$) | Batch processing (\$\$\$) | Hardened access (\$\$\$) |
|----------------------------|-----------------|---------------|---------------------|---------------------------|--------------------------|
| System node pool node size | D52_v2          | B4ms          | B4ms                | D4s_v3                    | D4s_v3                   |
| User node pool node size   | -               | -             | B4ms                | NC6s_v2                   | D4s_v3                   |
| Cluster autoscaling        | ✓               | -             | ✓                   | ✓                         | ✓                        |
| Private cluster            | -               | -             | -                   | -                         | ✓                        |
| Availability zones         | ✓               | -             | -                   | -                         | ✓                        |
| Azure Policy               | -               | -             | -                   | -                         | ✓                        |
| Azure Monitor              | ✓               | -             | -                   | ✓                         | ✓                        |

[Apply](#) [Discard](#) ⚠ Changing the preset configuration may reset the selections you've made.

**Y el nombre de Kubernetes y la zona de disponibilidad** son cosas que se explican por sí solas.

A continuación, tenemos **el tamaño del nodo** : - Preseleccionado, es la opción para la configuración predefinida. Si desea cambiarlo, puede cambiar el SKU de la máquina virtual manualmente.

**Método de escalado:** Aquí tenemos dos opciones: *Manual* y *Escalado automático* , ambas intuitivas. Recomendamos usar *el escalado automático* para producción. También puede definir el rango de nodos, como mínimo, una máquina virtual de nodo y un máximo de 10 o 20.

Microsoft Azure

Home > Create a resource >

## Create Kubernetes cluster

View all preset configurations

Kubernetes cluster name \*

Region \*

Availability zones   
 High availability is recommended for standard configuration.

Kubernetes version \*

Primary node pool

The number and size of nodes in the primary node pool in your cluster. For production workloads, at least 3 nodes are recommended for resiliency. For development or test workloads, only one node is required. If you would like to add additional node pools or to see additional configuration options for this node pool, go to the "Node pool" tab above. You will be able to add additional node pools after creating your cluster. [Learn more about node pools in Azure Kubernetes Service](#)

Node size \*   
 4 vcpus, 14 GiB memory  
 Standard D52\_v2 is recommended for standard configuration.  
[Change size](#)

Scale method \* ☐ Manual ☒ Autoscale  
 Autoscaling is recommended for standard configuration.

Node count range \*

[Review + create](#) [< Previous](#) [Next: Node pools >](#)

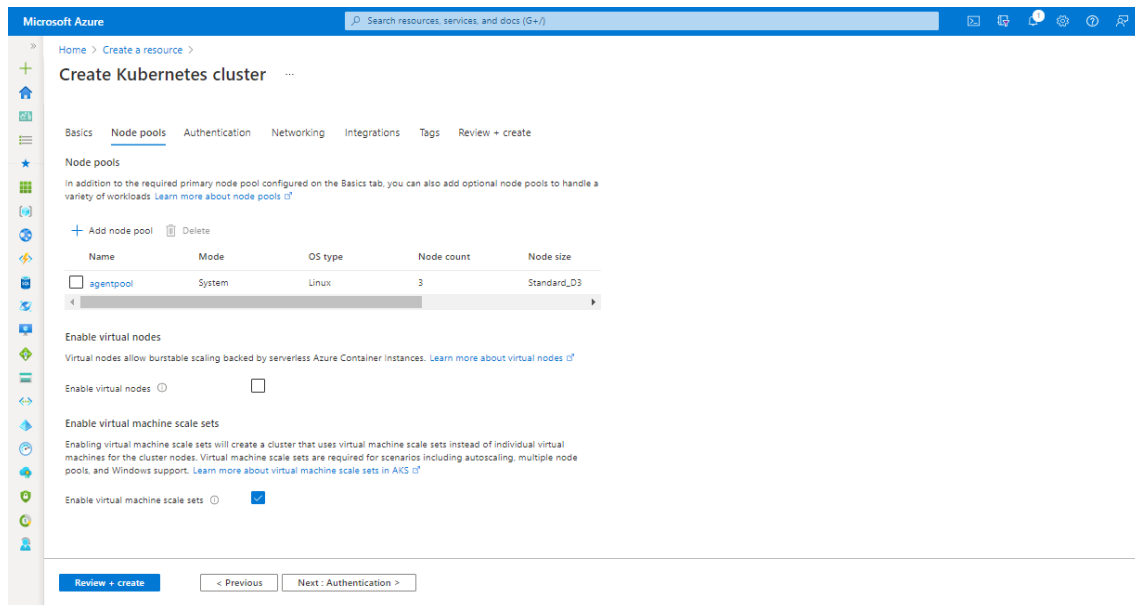
A continuación, tenemos **los grupos de nodos**:

Los grupos de nodos, según la configuración de AKS, se agrupan en *grupos de nodos*. Estos grupos de nodos contienen las máquinas virtuales subyacentes que ejecutan las aplicaciones. Esta función permite un mayor control sobre la creación y administración de múltiples grupos de nodos.

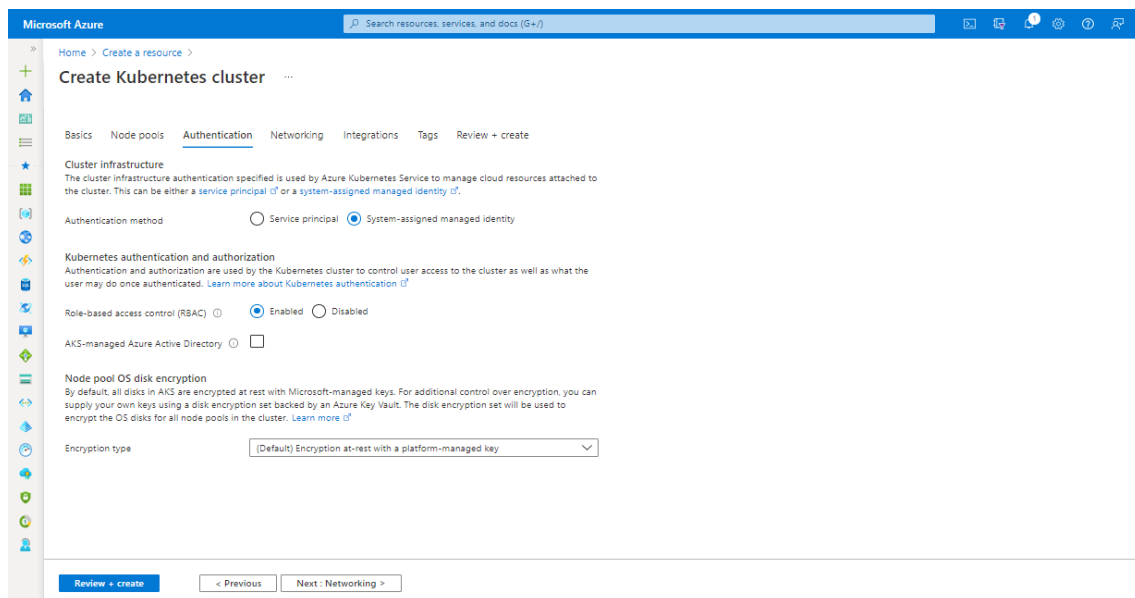
**Nodos virtuales:** Los nodos virtuales permiten la comunicación de red entre los pods que se ejecutan en Azure Container Instances (ACI) y el clúster de AKS. Básicamente, los nodos virtuales ofrecen funciones de red avanzadas. Para facilitar esta comunicación, se crea una subred de red virtual y se asignan permisos delegados. Los nodos virtuales solo funcionan con clústeres de AKS creados mediante redes *avanzadas* (Azure CNI).

**VMSS:** Azure VMSS permite crear y administrar máquinas virtuales idénticas con balanceo de carga, que aumentan o disminuyen automáticamente según la demanda o una programación establecida. Esto permite administrar y escalar fácilmente varias máquinas virtuales para proporcionar alta disponibilidad y resiliencia de aplicaciones, ideal para aplicaciones a gran escala como cargas de trabajo de contenedores.

El escalador automático de clústeres le permite ajustar automáticamente el tamaño de los clústeres de Kubernetes en función de las condiciones de carga.



En la pestaña **Autenticación** podemos ver el método de autenticación Principal de servicio e Identidad administrada asignada por el sistema; es un tema amplio para discutir, por lo que compartiremos el enlace del mismo.



## Paso 02: Configurar la red en la implementación de AKS

Entonces, de manera predeterminada, obtenemos la opción **Kubenet** en la red y Azure también nos brinda la posibilidad de usar Azure CNI, que es la interfaz de red de contenedores de Azure, que le permite usar la red virtual implementada existente y Kubenet creará una nueva red virtual.

- **DNS** : Prefijo de nombre DNS para usar con el FQDN del servidor de la API de Kubernetes alojado. Lo usará para conectarse a la API de Kubernetes al administrar [contenedores](#) después de crear el clúster.
- **Enrutamiento de tráfico** : Un balanceador de carga público, al integrarse, proporciona conexiones salientes a los nodos del clúster dentro de la red virtual de AKS. Este objetivo se logra traduciendo la dirección IP privada de los nodos a una dirección IP pública que forma parte de su pool de salida.
- Se utiliza un balanceador de carga interno (o privado) donde solo se permiten IP privadas como interfaz. Los balanceadores de carga internos se utilizan para equilibrar el tráfico dentro de una red virtual. También se puede acceder a la interfaz de un balanceador de carga desde una red local en un escenario híbrido.
- **Seguridad** : - En seguridad tenemos dos opciones que pueden ayudarnos a mejorar la seguridad, una es Habilitar clúster privado y la otra es Establecer rangos de IP autorizados.

Habilitar clúster privado es una opción que le permite hacer que su clúster AKS funcione solo en puntos finales privados

Establecer rangos de IP autorizados es una opción para incluir en la lista blanca un rango específico de IP para su acceso al clúster de AKS.

### Paso 03: Integración de AKS con Azure Container Registry

En la siguiente pestaña, AKS nos ofrece integrar nuestro clúster AKS para integrarlo con otros servicios de Azure como Azure Container Registry, que es un servicio de repositorios que almacena nuestras imágenes de Docker que podemos usar para implementar o ejecutar como contenedores en el clúster AKS.

Aparte de eso, tenemos **Azure Monitor** que nos ayuda a monitorear y rastrear nuestras aplicaciones.

Además de las métricas de CPU y memoria incluidas en AKS por defecto, puede habilitar Container Insights para obtener datos más completos sobre el rendimiento general y el estado de su clúster. La facturación se basa en la configuración de ingesta y retención de datos.

[Azure Policy](#) nos ayudará a garantizar la seguridad de nuestro clúster de AKS y a administrarlo de forma centralizada. Aplique medidas de seguridad y protección a gran escala para los clústeres de AKS de forma centralizada y coherente mediante Azure Policy .

The screenshot shows the 'Create Kubernetes cluster' wizard in the Azure portal. The 'Integrations' tab is selected, displaying various integration options. A side panel titled 'Create container registry' is open, showing the following configuration:

- Registry name: demoacrash
- Subscription: Dev-Lab-Primary
- Resource group: (New) AKSDemo
- Region: (Asia Pacific) Central India
- Admin user: ☒ Disable
- SKU: Standard

Y sí, hemos completado el asistente con conocimientos básicos sobre los componentes y configuraciones de AKS. Ahora, revisemos **y creemos** el asistente.

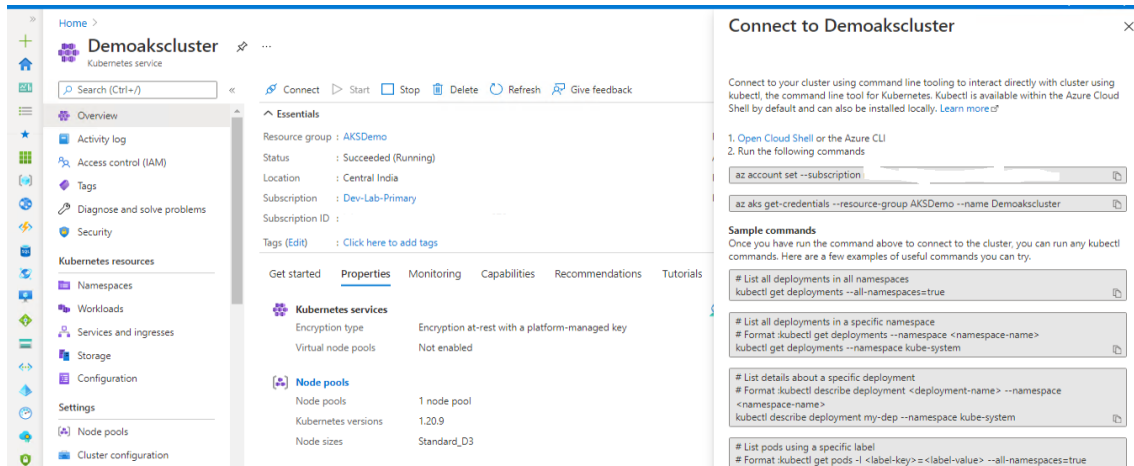
The screenshot shows the 'Review + create' tab of the 'Create Kubernetes cluster' wizard. A green banner at the top indicates 'Validation passed'. The configuration details are as follows:

| Section                            | Configuration                                            |
|------------------------------------|----------------------------------------------------------|
| Basics                             |                                                          |
| Subscription                       | Dev-Lab-Primary                                          |
| Resource group                     | (new) AKSDemo                                            |
| Region                             | Central India                                            |
| Kubernetes cluster name            | Demoakscuster                                            |
| Kubernetes version                 | 1.20.9                                                   |
| Node pools                         |                                                          |
| Node pools                         | 1                                                        |
| Enable virtual nodes               | Disabled                                                 |
| Enable virtual machine scale sets  | Enabled                                                  |
| Authentication                     |                                                          |
| Authentication method              | System-assigned managed identity                         |
| Role-based access control (RBAC)   | Enabled                                                  |
| AKS-managed Azure Active Directory | Disabled                                                 |
| Encryption type                    | (Default) Encryption at-rest with a platform-managed key |
| Networking                         |                                                          |

## Paso 04: Conéctese al clúster AKS mediante VS Code

El primer paso es tener Azure CLI instalado en su equipo local para poder iniciar sesión en Azure Portal.

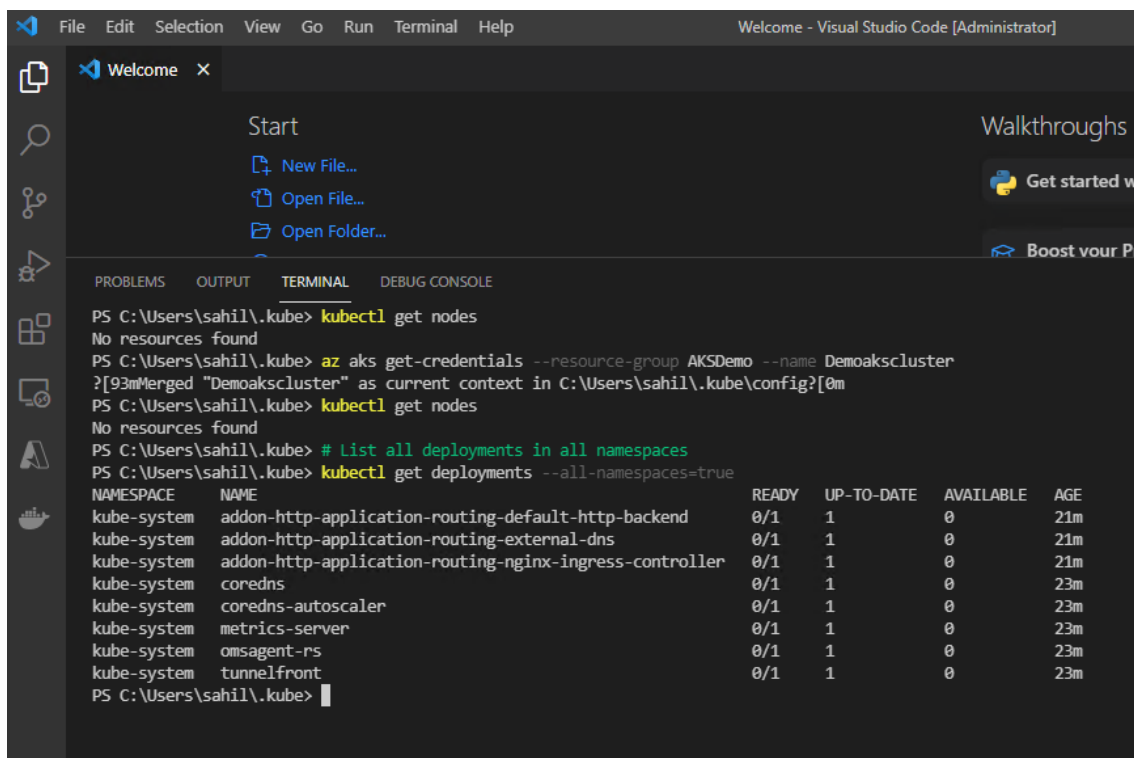
A continuación, vaya al panel **Descripción general** y haga clic en **Conectar** y copie los dos primeros comandos.



Ejecute estos comandos como se muestra en la imagen de arriba en su terminal local.

kubectl obtiene implementaciones --all-namespaces=true

Puedes ver todos los espacios de nombres en tu clúster.



## Paso 05: - Ejecutar la aplicación en nuestro Cluster AKS

Primero clonaré el código del repositorio base desde el enlace de GitHub.

[Azure-Samples/azure-voting-app-redis: Aplicación de votación de Azure utilizada en la documentación \(github.com\)](https://github.com/Azure-Samples/azure-voting-app-redis)

Podemos clonar este código con el comando git clone. A continuación, debemos crear el archivo de manifiesto de Kubernetes para que implemente nuestra aplicación.

**Abra el bloc de notas o cualquier herramienta de edición y pegue el siguiente YAML allí.**

este

apiVersion: apps/v1

kind: Deployment

metadata:

name: azure-vote-back

spec:

replicas: 1

selector:

matchLabels:

app: azure-vote-back

template:

metadata:

labels:

app: azure-vote-back

spec:

nodeSelector:

"kubernetes.io/os": linux

containers:

- name: azure-vote-back

image: mcr.microsoft.com/oss/bitnami/redis:6.0.8

env:

- name: ALLOW\_EMPTY\_PASSWORD

value: "yes"

resources:



```
requests:
  cpu: 100m
  memory: 128Mi
limits:
  cpu: 250m
  memory: 256Mi
ports:
  - containerPort: 6379
  name: redis
```

---

```
apiVersion: v1
kind: Service
metadata:
  name: azure-vote-back
spec:
  ports:
    - port: 6379
  selector:
    app: azure-vote-back
```

---

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: azure-vote-front
spec:
  replicas: 1
  selector:
    matchLabels:
      app: azure-vote-front
  template:
    metadata:
```

```
labels:
  app: azure-vote-front
spec:
  nodeSelector:
    "kubernetes.io/os": linux
  containers:
    - name: azure-vote-front
      image: mcr.microsoft.com/azuredocs/azure-vote-front:v1
  resources:
    requests:
      cpu: 100m
      memory: 128Mi
    limits:
      cpu: 250m
      memory: 256Mi
  ports:
    - containerPort: 80
  env:
    - name: REDIS
      value: "azure-vote-back"
---
apiVersion: v1
kind: Service
metadata:
  name: azure-vote-front
spec:
  type: LoadBalancer
  ports:
    - port: 80
  selector:
    app: azure-vote-front
```

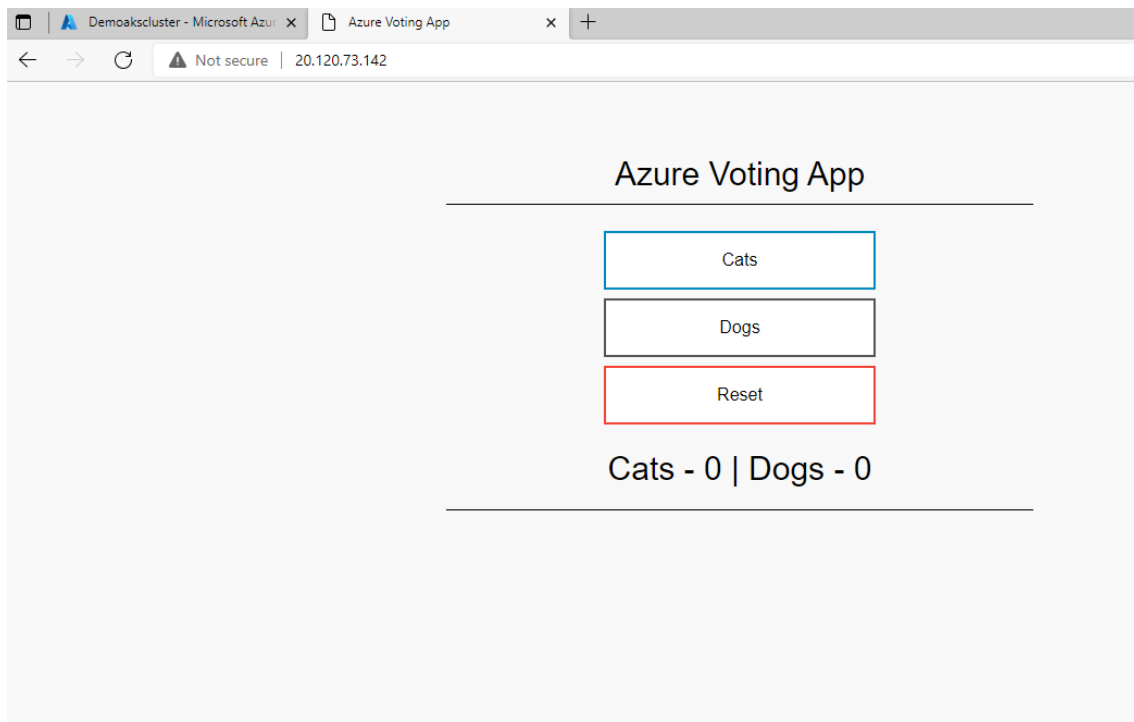
```
-a----      12/4/2021   7:02 PM      1613 azure-vote-all-in-one-redis.yaml
-a----      12/4/2021   7:02 PM        451 docker-compose.yaml
-a----      12/4/2021   7:02 PM       1183 LICENSE
-a----      12/4/2021   7:02 PM       1837 README.md

PS C:\Users\sahil\.kube\azure-voting-app-redis> notepad azure-vote.yaml
PS C:\Users\sahil\.kube\azure-voting-app-redis> kubectl apply -f azure-vote.yaml
deployment.apps/azure-vote-back created
service/azure-vote-back created
deployment.apps/azure-vote-front created
service/azure-vote-front created
PS C:\Users\sahil\.kube\azure-voting-app-redis> |
```

Puede obtener los detalles del servicio utilizando el siguiente comando

```
kubectl get service azure-vote-front -w
```

Y como se muestra en la imagen a continuación, puede ver que nuestra aplicación se ejecuta en la IP pública del balanceador de carga externo AKS.



Esta es una forma de implementar tu aplicación en el clúster de AKS. Puedes crear tu imagen de Docker y usar ACR para guardarla, y desde allí también puedes enviarla.