

Práctica AZ-305: Despliegue escalable y con alta disponibilidad en Azure App Service

Contexto

Dispones de una aplicación compilada en **.NET 8 portable (framework-dependent)**. Tu objetivo es publicarla en **Azure App Service para Linux**, garantizando alta disponibilidad, escalado automático y separación entre entornos de **staging** y **producción**. La aplicación incluye una clave de configuración llamada **QueueConnectionString** en el archivo **appsettings.json**. Esta clave debe configurarse mediante las Application Settings del App Service y apunta a una **Azure Storage Account Standard v2** cuya cola será creada automáticamente por la aplicación en tiempo de ejecución.

Objetivos

- Diseñar e implementar la arquitectura necesaria para alojar la aplicación en Azure.
- Configurar escalado automático en función del uso de CPU.
- Garantizar alta disponibilidad con soporte de zonas.
- Separar los entornos de staging y producción mediante slots de implementación.
- Configurar de forma segura la cadena de conexión (QueueConnectionString).
- Publicar la aplicación vía FTP sobre el slot staging.
- Ejecutar un swap manual a producción tras validar el despliegue.

Requisitos técnicos

App Service Plan: Premium v3 (P0v3), entorno Linux, disponibilidad de zona habilitada, escalado mínimo 2 instancias, máximo 5. **Escalado automático (CPU):** +1 instancia si CPU > 15% durante 1 minuto; -1 instancia si CPU < 10%. **Web App:** con slot de staging y producción. Publicación por FTP en staging y swap manual a producción. **Configuración:** Variable QueueConnectionString configurada desde el portal, con la cadena de conexión de una Storage Account Standard v2.

Indicaciones generales

- 1 Diseña la arquitectura: define recursos y relaciones (App Service Plan, Web App, Storage Account, autoscaling y slots). Representa el diseño en un diagrama lógico.
- 2 Implementa la infraestructura en un grupo de recursos común, habilitando zonas de disponibilidad y configurando las reglas de autoscaling.
- 3 Configura la aplicación: añade la variable QueueConnectionString sin modificar el appsettings.json local.
- 4 Publica en el entorno de staging mediante FTP y verifica el correcto funcionamiento.
- 5 Valida la funcionalidad: la aplicación debe crear la cola en el almacenamiento y responder correctamente bajo carga.
- 6 Promociona a producción realizando un swap manual entre staging y production.
- 7 Reflexiona sobre la solución: analiza ventajas del autoscaling, de los slots y cómo podrías automatizar el despliegue en el futuro.

Entregables

- Diagrama de arquitectura de la solución.
- Capturas de configuración del escalado automático.

- URL pública del App Service en producción.
- Explicación breve de las decisiones de diseño y de cómo se garantiza la disponibilidad de zona.