2023

Seguridad en Sistemas de Computación

PROYECTO FINAL

CANEK, OSCAR

Tabla de contenido

Consideraciones

- El trabajo es grupal y se calificará de la misma forma.
- Todos los integrantes de los grupos deben demostrar que trabajaron en el proyecto.
- Durante la presentación del proyecto todos los integrantes deben estar presentes.
- Se calificará en base a lo implementado y a la presentación de la implementación, deben demostrar que manejan los temas que aprendieron durante el curso.
- De no seguir las instrucciones no se tomará en cuenta lo realizado durante la calificación.

Prerrequisitos

- Deben tener una cuenta en Azure DevOps
- Deben tener una cuenta en Azure

Proyecto Final

Usar herramientas para implementar un proceso de desarrollo seguro para una empresa que quiere mejorar su nivel de madurez en el desarrollo de software.

El proyecto consiste de varias partes que deben trabajar, las cuales se listan a continuación:

- Herramienta para el manejo de ciclo de vida de la aplicación y DevOps
- Historias de usuario
- Código fuente
- Repositorio de código
- Estrategia de ramas
- Integración continua / Despliegue continuo (CI/CD)
- Despliegue de infraestructura

En las siguientes secciones se detallan cada uno de los puntos anteriores.

Algunos conceptos para poner en práctica con este proyecto son los siguientes:

- Security Governance Through Principles and Policies
- Access controls
- Autentication and authorization
- Criptography
- Managing Security Operations
  - Applying Security Operations Concepts
    - Need-to-Know and Least Privilege
    - Need-to-Know Access
    - The Principle of Least Privilege
    - Separation of Privilege
    - Segregation of Duties
  - Securely Provisioning Resources
    - Managing Cloud-Based Assets
  - Managing Configuration
    - Baselining
  - Managing Change
    - Versioning
    - Configuration Documentation
- Software Development Security
  - Introducing Systems Development Controls
    - Software Development
    - Avoiding and Mitigating System Failure
    - Systems Development Lifecycle

        - Conceptual definition
        - Functional requirements determination
        - Control specifications development
        - Design review
        - Code review walk-through
        - System test review
        - Maintenance and change management
    - Agile Software Development
    - Change and Configuration Management
    - Software Testing
    - Code Repositories
- Storing Data and Information

Herramienta para el manejo de ciclo de vida de la aplicación y DevOps

Para esta sección deben realizar lo siguiente:

1. Crear una cuenta en Azure DevOps
2. Crear una organización
3. Crear un proyecto
4. Agregar a cada miembro del equipo al proyecto
5. Crear cada uno de los siguientes grupos:
    a. DevOps
        i. Mange repositories
        ii. Manage CI/CD pipelines
    b. PR approvers
        i. Approve pull request
    c. Developers
        i. Add tasks to the user stories
        ii. Push changes to the user stories branches
        iii. Create pull requests
    d. Product Owner
        i. Manage user stories
6. Agregar un usuario a cada uno de esos grupos, ninguno debe ser administrador del proyecto. Por lo tanto, será 4 usuario más uno usuario administrador, dando un total de 5 usuarios
7. Crear un repositorio de código fuente
    a. Crear tres ramas
        i. dev
        ii. release
        iii. prod
    b. Agregar las siguientes políticas de seguridad
        i. Agregar políticas a la rama y bloquearla para no permitir push directos
        ii. Requerir al menos un revisor
        iii. No permitir que la misma persona que sube un cambio aprouebe el mismo
        iv. No debe permitirse completar el pull request si hay algo pendiente
        v. Si nuevos cambios son recibidos (push) se debe resetear las aprobaciones
        vi. Se debe validar que haya una historia asociada al pull request
        vii. Debe tener un Build pipeline asociado en la sección de Build Validation como validación de pre-build
        viii. Los reviewers deben ser configurados para ser agregados automáticamente al momento de crear un pull request
8. Crear un Build Pipeline para compilar el código y crear un artefacto
    a. Esta debe ejecutarse automáticamente luego del que pull request es aprobado
    b. Se debe validar de que las pruebas unitarias pasen en verde
    c. Debe crear un artefacto para ser desplegado por un Release Pipeline

9. Crear un Release Pipeline para desplegar el artefacto en Azure
    a. Debe ejecutarse automáticamente cuando un artefacto es creardo por un Build Pipeline
    b. Debe desplegar el Azure Function hacia la nube de Azure

Historias de usuarios

1. El product owner debe crear una historia de usuario con las siguientes consideraciones:
    a. Crear una descripción detallada del cambio
    b. Crear los criterios de aceptación que deben tomarse en cuenta durante la revisión del código
2. Asignar la historia de usuario a un desarrollador
3. Asignar la historia de usuario a un sprint
4. El desarrollador debe crear una rama con la siguiente nomenclatura
    a. Nomenclatura: <work item>/<nombre del equipo de desarrollo>/US<id hitoria de usuario>-<nombre del cambio>
    b. Ejemplo: feature/darkarmy/US123-add-new-endpoint
5. El desarrollador debe crear al menos una tarea y cerrarla antes de crear el pull request
6. El desarrollador debe aplicar los cambios en la rama creada en el punto 4 y hacer push de sus cambios al repositorio upstream
7. El desarrollador debe crear un pull request y asignar la user story relacionada a ese cambio
    a. Los revisores deben asignarse automaticamente
    b. Al momento de crear el PR se debe disparar el Build Validation automáticamente
8. Un miembro del grupo PR approvers debe apobar el cambio
9. Una vez que el cambio es aprobado, el build pipeline configurado debe ejecutarse automáticamente

Código fuente

Para probar el proceso se utilizará una pequeña API creada en .NET con los siguientes requerimientos:

- Debe ser un Azure Function
- .NET 6 LTS
- Runtime: Isolated
- HTTP trigger. Debe aceptar el método GET para retornar información
- Lenguaje C#
- Debe retornar un mensaje personalizado
- Debe contar con pruebas unitarias

El endpoint debe retornar un mensaje que diga “Proyecto de Seguridad 2023” seguido de un GUID que deberá ser diferente para cada llamada. Ejemplo:

- “Proyecto de Seguridad 2023 0d00f5ba-937a-4fdd-bf5d-0d379a484bfa”

El Azure Function deberá tener Application Insights habilitado para poder monitorear la aplicación.

Para realizar las pruebas durante la calificación deben pueden utilizar Postman.

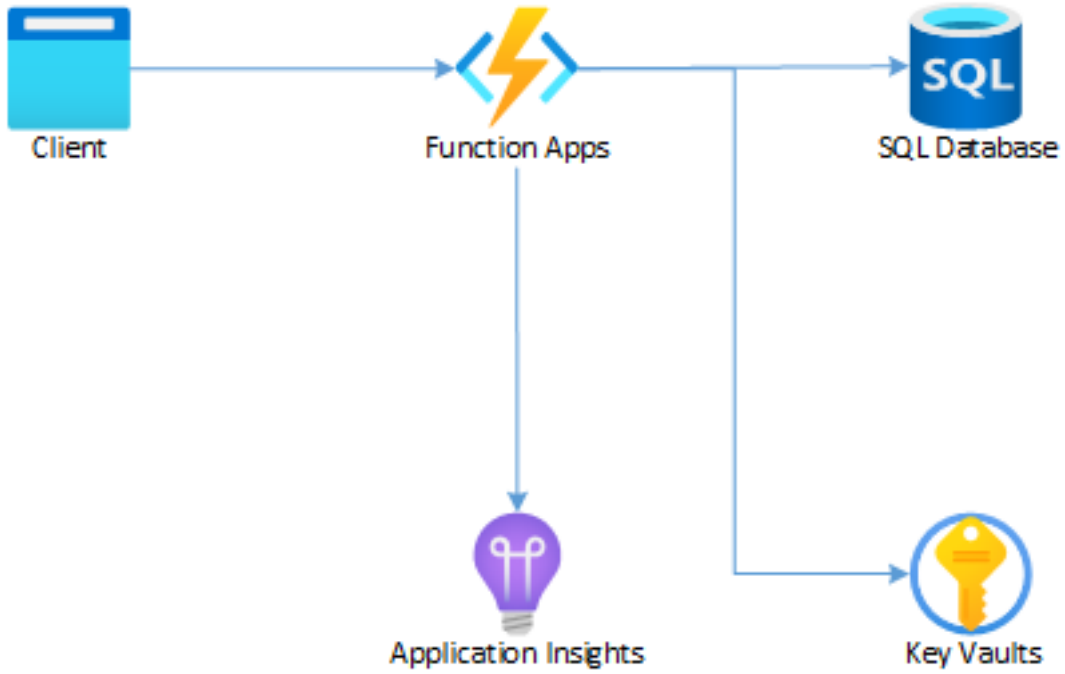
Infraestructura

La infraestructura debe ser desplegada en Azure con los siguientes recursos:

- App Service Plan
- Azure Function
- Key Vault
- SQL Server

Esta infraestructura deben desplegarla usando Terraform. El repositorio de con la infraestructura también debe estar subido en Azure DevOps.

En el diagrama de la arquitectura se encuentra en la siguiente imagen:



App Service Plan

App Service Plan para correr el Azure Function.

Disclaimer

Deben seleccionar un plan de servicio de consumo para no acabar con sus créditos antes de concluir con su proyecto.

Azure Function

Deben desplegar un Azure Function con las siguientes especificaciones:

- .NET 6 LTS
- Runtime: Isolated n
- OS: Windows/Linux
- Consuption Plan
- Service principal para acceder al Key Vault
- Agregar Application Insights a la función
- No debe tener la cadena de conexión a la base de datos en claro en sus configuraciones, deben utilizar el key vault para obtener la cadena de conexión

SQL Server

Pueden desplegar una managed instance para facilitar el desarrollo y tener el script de sql por aparte.

Key Vault

Tienen que guardar en este Key Vault la cadena de conexión a la base de datos y referenciarla en la configuración de la Azure Function.

Application Insights

Para monitorear la aplicación se debe conectar Application Insights al Azure Function. Esto se hace colocando la cadena de conexión de Application Insights en la configuración de la Azure Function.

Presentación

Deben presentar su implementación y demostrar que están aplicando los conceptos aprendidos durante el curso. Durante la presentación se realizarán preguntas que deben contestar para demostrar que manejan los temas vistos en clase y la implementación realizada.

Rúbrica

Módulo	Puntaje
Herramienta para el manejo de ciclo de vida de la aplicación y DevOps	25%
Historias de usuario	25%
Código fuente	25%
Infraestructura	25%
Total	100%

Entrega

La fecha de entrega del proyecto es el 29 de mayo del 2023. Ese día también se realizará la calificación de cada uno de los grupos. Se debe presentar el ambiente creado con las especificaciones dadas en el proyecto, deben explicar lo realizado y que conceptos de seguridad se aplican en lo implementado.