

AZ-400: Designing and Implementing Microsoft DevOps Solutions

Duración: **30 Horas**

INTRODUCCION:

Esta formación proporciona el conocimiento y las habilidades para implementar los procesos de DevOps. Los estudiantes aprenderán a utilizar el control de fuentes, escalar Git para una empresa e implementar y administrar la infraestructura de construcción. También proporciona el conocimiento y las habilidades para implementar las prácticas de DevOps de integración continua. Los estudiantes aprenderán cómo implementar la integración continua en una tubería de Azure DevOps, cómo manejar la calidad del código y los principios de seguridad y cómo implementar una estrategia de construcción de contenedores.

Los asistentes aprenderán a diseñar una estrategia de liberación, establecer un flujo de trabajo de gestión de liberación e implementar un patrón de despliegue apropiado.

Este curso también incluye conocimientos y habilidades para desplegar una infraestructura de aplicación en los oleoductos de DevOps. Los alumnos aprenderán a implementar una infraestructura como gestión de código y configuración, a aprovisionar una infraestructura Azure utilizando herramientas de automatización comunes y a desplegar una infraestructura de aplicaciones utilizando varios servicios y metodologías de despliegue Azure. Los alumnos también aprenderán a integrar herramientas de implementación de terceros con Azure, como Chef y Puppet, para incorporar el cumplimiento de normas y la seguridad en el proceso de lanzamiento.

OBJETIVOS:

Algunos de los objetivos clave de este curso son:

-  Diferenciar entre un lanzamiento y un despliegue.
-  Definir los componentes de un pipeline de liberación.
-  Conocer los factores clave a la hora de diseñar su estrategia de lanzamiento.
-  Clasificar una liberación versus un proceso de liberación, y esbozar cómo controlar la calidad de ambos.
-  Explicar los patrones de despliegue, tanto en el sentido tradicional como en el moderno.
-  Entender la importancia de la integración continua e implementarla utilizando Azure DevOps.
-  Crear un flujo de trabajo automatizado, integrar otras herramientas de construcción con Azure DevOps y crear procesos de construcción híbridos.
-  Describir qué significa la calidad del código y cómo se mide, detectar hediondes de código (code smells) e integrar pruebas automatizadas para la calidad de código.
-  Diferenciar entre una release y un despliegue, definir los componentes de la release de una pipeline.
-  Clasificar una release versus un proceso de release y describir cómo controlar la calidad de ambos, describir el principio de las release gates y cómo tratar con las notas de la release y la documentación.

- Explicar los patrones de despliegue (tanto en el sentido tradicional como en el moderno), elegir una herramienta de gestión de release y explicar la terminología utilizada en Azure DevOps y otras herramientas de administración de versiones.
- Hacer pruebas de inserción en la pipeline, enumerar las diferentes formas de inspeccionar el estado de la pipeline y la release mediante el uso de alertas, enlaces de servicio e informes.
- Recomendar herramientas y prácticas de gestión de artefactos, resumir paquetes comunes para permitir compartir y reutilizar e inspeccionar el código base para identificar las dependencias de código que se pueden convertir en paquetes
- Identificar y recomendar tipos y versiones de paquetes estandarizados en toda la solución, refactorizar pipelines existentes para implementar la estrategia de versión que publique paquetes y gestionar la seguridad y el cumplimiento normativo.
- Aplicar infraestructura y configuración como principios de código, implementar y administrar la infraestructura utilizando las tecnologías de automatización de Microsoft, como las plantillas ARM, PowerShell y la CLI de Azure y describir los modelos y servicios de implementación que están disponibles con Azure.
- Desplegar y configurar un cluster administrado de Kubernetes y la infraestructura utilizando herramientas y servicios de terceros con Azure como Chef, Puppet, Ansible, SaltStack y Terraform.
- Definir una estrategia de infraestructura y configuración y un conjunto de herramientas adecuado para la release pipeline y la infraestructura de aplicaciones, implementar el cumplimiento normativo y la seguridad en su infraestructura de aplicaciones.
- Diseñar prácticas para medir la satisfacción del usuario final, procesos para capturar y analizar los comentarios de los usuarios de fuentes externas.
- Diseño de enrutamiento para datos de informes de fallos de aplicaciones de cliente y recomendar herramientas y tecnologías de monitoreo y herramientas de seguimiento de uso de características y del sistema.
- Configurar la integración de informes de errores para aplicaciones de cliente, desarrollar paneles de monitoreo y estado e implementar enrutamiento para datos de informes de fallos en aplicaciones de cliente y herramientas para rastrear el uso del sistema, el uso de características y el flujo.
- Integrar y configurar sistemas de etiquetado con el sistema de gestión de trabajo del equipo de desarrollo y analizar alertas y telemetría para establecer una línea de base (baseline).
- Planificar la transformación con objetivos y plazos compartidos, seleccionar un proyecto e identificar sus métricas y KPIs y crear un equipo y una estructura organizativa ágil.
- Desarrollar una estrategia de calidad del proyecto, desarrollar un plan de prácticas seguras de desarrollo y normas de cumplimiento y migrar y consolidar artefactos y medidas de control.

DIRIGIDO A:

Este curso está pensado para todos aquellos interesados en implementar procesos de DevOps. También para aquellos que estén planificando realizar el examen de certificación de Microsoft Azure DevOps Solutions.

EXAMEN:

Esta formación es la recomendada por Microsoft para la preparación del siguiente examen:

-  AZ-400: Designing and Implementing Microsoft DevOps Solutions

PREREQUISITOS:

Para garantizar un adecuado seguimiento del curso, es recomendable que los alumnos cumplan los siguientes requisitos previos:

-  Conocimientos fundamentales sobre Azure, control de versiones y desarrollo de software Agile.
-  Experiencia trabajando en un IDE, con máquinas virtuales y contenedores y también cierta exposición a la automatización y al scripting.
-  Conocimiento de los procesos generales de desarrollo e implementación de aplicaciones.

CONTENIDO:

Módulo 1: Planificación para DevOps

-  Selección de proyectos
-  Estructuras de equipos
-  Migración a Azure DevOps

Módulo 2: Primeros pasos con Source Control

-  Qué es Source Control
-  Beneficios de Source Control
-  Tipos de sistemas de Source Control
-  Introducción a Azure Repos
-  Introducción a GitHub
-  Migrar de Team Foundation Version Control (TFVC) a Git en Azure Repos
-  Autenticar a Git en Azure Repos

Módulo 3: Escalar Git para enterprise DevOps

-  Cómo estructurar su Git Repo
-  Git Branching Workflows
-  Colaborar con Pull Requests en Azure Repos
-  Por qué ocuparse de GitHooks
-  Fomento de la fuente interna

Módulo 4: Consolidar artefactos y diseñar una estrategia de gestión de dependencias

-  Dependencias de embalaje
-  Gestión de embalaje
-  Migrar y consolidar artefactos

Módulo 5: Implementación de integración continua con Azure Pipelines

- 🔗 El concepto de canalizaciones en DevOps
- 🔗 Azure Pipelines
- 🔗 Evaluar uso de agentes alojados en comparación con privados
- 🔗 Grupos de agentes
- 🔗 Canalización y concurrencia
- 🔗 Azure DevOps y Open Source Projects (proyectos públicos)
- 🔗 Azure Pipelines YAML en comparación con Visual Designer
- 🔗 Descripción general de integración continua
- 🔗 Implementar una estrategia de generación
- 🔗 Integración con Azure Pipelines
- 🔗 Integrar External Source Control con Azure Pipelines
- 🔗 Configurar agentes privados
- 🔗 Analizar e integrar Docker Multi-Stage Builds

Módulo 6: Administrar la configuración y los secretos de la aplicación

- 🔗 Introducción a seguridad
- 🔗 Implementar un proceso de desarrollo seguro y conforme
- 🔗 Repensar los datos de configuración de la aplicación
- 🔗 Administrar secretos, tokens y certificados
- 🔗 Implementar herramientas para administrar la seguridad y el cumplimiento en una canalización

Módulo 7: Gestión de políticas de calidad y seguridad del código

- 🔗 Gestión de la calidad del código
- 🔗 Gestión de políticas de seguridad

Módulo 8: Implementar una estrategia de Container Build

- 🔗 Implementar una estrategia de Container Build

Módulo 9: Gestionar versiones de artefactos, seguridad y cumplimiento

- 🔗 Seguridad de paquetes
- 🔗 Software de código abierto
- 🔗 Integrar escaneos de licencia y vulnerabilidad
- 🔗 Implementar una estrategia de control de versiones

Módulo 10: Diseñar una estrategia de versión

- 🔗 Introducción la entrega continua
- 🔗 Recomendaciones de estrategia de versión
- 🔗 Construir una canalización de versión de alta calidad
- 🔗 Elegir un patrón de implementación
- 🔗 Elegir la herramienta de administración de versiones correcta

Módulo 11: Configurar un flujo de trabajo de administración de versiones

- 🔗 Crear una canalización de lanzamiento
- 🔗 Aprovisionar y configurar entornos
- 🔗 Administrar y modularizar tareas y plantillas
- 🔗 Integrar Secretos con la canalización de versión
- 🔗 Configurar la integración automatizada y la automatización de pruebas funcionales
- 🔗 Automatizar inspección de estado

Módulo 12: Implementar un patrón de implementación apropiado

- 🔗 Introducción a patrones de implementación
- 🔗 Implementar Blue Green Deployment
- 🔗 Alternar funciones
- 🔗 Versiones Canary
- 🔗 Lanzamiento oscuro
- 🔗 Pruebas AB
- 🔗 Progressive Exposure Deployment

Módulo 13: Implementar el proceso para enrutar los comentarios del sistema a los equipos de desarrollo

- 🔗 Implementar herramientas para rastrear el uso del sistema, el uso de funciones y el flujo
- 🔗 Implementar enrutamiento para datos de informe de bloqueo de aplicaciones móviles
- 🔗 Desarrollar paneles de supervisión y estado
- 🔗 Integrar y configurar sistemas de tickets

Módulo 14: Infraestructura y configuración de Azure Tools

- 🔗 Infraestructura como código y gestión de la configuración
- 🔗 Crear recursos de Azure con plantillas ARM
- 🔗 Crear recursos de Azure con Azure CLI
- 🔗 Crear recursos de Azure con Azure PowerShell
- 🔗 Configuración de estado deseado (DSC)
- 🔗 Automatización de Azure con DevOps
- 🔗 Herramientas adicionales de automatización

Módulo 15: Modelos y servicios de implementación de Azure.

- 🔗 Módulos y opciones de implementación.
- 🔗 Servicios Azure de infraestructura como servicio (IaaS).
- 🔗 Servicios de plataforma como servicio de Azure (PaaS).
- 🔗 Servicios informáticos HPC y sin servidor.
- 🔗 Azure Service Fabric.

Módulo 16: Crear y administrar la infraestructura de servicio de Kubernetes

- 🔗 Azure Kubernetes Service
- 🔗

Módulo 17: Infraestructura de terceros como herramientas de código disponibles con Azure

-  Chef
-  Puppet
-  Ansible
-  Terraform

Módulo 18: Implementar cumplimiento y seguridad en su infraestructura

-  Principios de seguridad y cumplimiento con DevOps
-  Centro de seguridad Azure

Módulo 19: Recomendar y diseñar mecanismos de realizar comentarios del sistema.

-  El bucle interno
-  Mentalidad de experimentación continua
-  Diseñar prácticas para medir la satisfacción del usuario final
-  Diseñar procesos para capturar y analizar los comentarios de los usuarios
-  Proceso de diseño para automatizar el análisis de aplicaciones

Módulo 20: Optimizar los mecanismos de realizar comentarios

-  Ingeniería de confiabilidad del sitio
-  Analizar la telemetría para establecer una línea de base
-  Realizar ajustes continuos para reducir alertas que carezcan de sentido o no sean procesables
-  Analizar alertas para establecer una línea de base
-  Retrospectivas sin culpa y una cultura justa