

Laboratorio de DevOps

Código: ARC-003

Propuesta de Valor: ARCITURA

Duración: 8 Horas Académicas



Este módulo del curso presenta a los participantes una serie de ejercicios y problemas que están diseñados para evaluar su capacidad para aplicar sus conocimientos sobre los temas tratados en cursos anteriores. Completar este laboratorio ayudará a resaltar las áreas que requieren mayor atención y demostrará aún más la competencia práctica en modelos, prácticas y estrategias de DevOps, a medida que se aplican y combinan para resolver problemas del mundo real.

Para impartir este curso de laboratorio dirigido por un instructor, el Instructor certificado trabaja en estrecha colaboración con los participantes para garantizar que todos los ejercicios se lleven a cabo de manera completa y precisa. Los asistentes pueden hacer que los ejercicios se revisen y califiquen voluntariamente como parte de la finalización de la clase. Para completar individualmente este curso como parte de un kit de estudio, se proporcionan varios suplementos para ayudar a los participantes a realizar los ejercicios con orientación.



AUDIENCIA

- Este curso está dirigido para todas las personas que quieran ampliar sus conocimientos en DevOps.



PRE REQUISITOS

- No hay requisitos previos.



OBJETIVOS

- Adquirir mayores conocimientos en las prácticas y estrategias de DevOps.



CERTIFICACIÓN DISPONIBLE

- Este curso es uno de los tres cursos que se utilizan para prepararse para el **examen DO90.01**.
- Se requiere una calificación aprobatoria en este examen para obtener la **certificación DevOps Specialist**.



CONTENIDO

1. CONFIGURACIÓN COMO CÓDIGO (CAC)

- 1.1. AUTOMATICE LA ESTANDARIZACIÓN PARA CONTINUAS VERSIONES DE SOFTWARE
- 1.2. CONFIGURACIONES DE SOFTWARE Y PRUEBAS AUTOMATIZADAS
- 1.3. CONTROL DE VERSIONES PROGRAMÁTICO
- 1.4. CAMBIOS RASTREABLES
- 1.5. IMPLEMENTACIÓN DE CONFIGURACIONES EN ENTORNOS
- 1.6. CONOCIMIENTO DEL ESTADO DE LA CONFIGURACIÓN A TRAVÉS DE LA COLABORACIÓN Y LA NOTIFICACIÓN

2. INFRAESTRUCTURA COMO CÓDIGO (IAC)

- 2.1. IMPLEMENTAR RAPIDAMENTE RECURSOS DE INFRAESTRUCTURA
- 2.2. AUTOMATIZAR LA IMPLEMENTACIÓN Y LA RECUPERACIÓN
- 2.3. IMPLEMENTAR CONSISTENTEMENTE
- 2.4. VUELVA A IMPLEMENTAR EN LUGAR DE REPARAR
- 2.5. REVERTIR RAPIDAMENTE
- 2.6. AUTOMATICE LAS PRUEBAS Y LA VERIFICACIÓN
- 2.7. VALIDAR ANTES DE LA IMPLEMENTACIÓN
- 2.8. INTEGRAR MONITOREO Y PRUEBAS

3. APROVISIONAMIENTO DE RECURSOS DE INFRAESTRUCTURA Y ETAPAS DEL CICLO DE VIDA DEL DESARROLLO

- 3.1. RECURSOS DE APROVISIONAMIENTO
- 3.2. CONFIGURACIÓN Y PERSONALIZACIÓN
- 3.3. VIGILANCIA
- 3.4. SEGURIDAD Y CUMPLIMIENTO
- 3.5. GOBERNANCIA
- 3.6. EL SISTEMA ESTA DISPONIBLE
- 3.7. OPTIMIZACIÓN Y PUESTA A PUNTO
- 3.8. DESAPROVISIONAMIENTO

4. EVALUACIÓN DE MADUREZ DE DEVOPS

- 4.1. CULTURA Y ESTRATEGIA EMPRESARIAL
- 4.2. COLABORACIÓN Y COMUNICACIÓN
- 4.3. AUTOMATIZACIÓN
- 4.4. GOBERNANZA Y PROCESOS

5. POLÍTICA COMO CÓDIGO (PAC)

- 5.1. NEGOCIO
- 5.2. SEGURIDAD
- 5.3. CONFORMIDAD
- 5.4. GOBERNANCIA
- 5.5. REGULADOR

5.6. OPERACIONAL

5.7. ACTUACIÓN

6. MECANISMOS DE SEGUIMIENTO CONTINUO

6.1. MONITOR DE RENDIMIENTO

6.2. MONITOR DE CUMPLIMIENTO

6.3. MONITOR FUNCIONAL / OPERATIVO

6.4. MONITOR SLA

6.5. MONITOR DE AUDITORÍA

6.6. MONITOR DE CALIDAD DE CÓDIGO

6.7. MONITOR DE DISPONIBILIDAD

6.8. MONITOR DE PRUEBA CONTINUA

7. MÉTRICAS DE DEVOPS

7.1. TIEMPO DEL CICLO

7.2. TIEMPO DE ESPERA

7.3. FRECUENCIA DE IMPLEMENTACIÓN

7.4. TIEMPO DE IMPLEMENTACIÓN

7.5. TICKETS DE SOPORTE AL CLIENTE

7.6. DISPONIBILIDAD

7.7. TASA DE ERROR

7.8. PORCENTAJE DE APROBACIÓN DE PRUEBA AUTOMATIZADA

7.9. TASA DE ESCAPE DE DEFECTOS

7.10. ACUERDOS DE NIVEL DE SERVICIO

7.11. IMPLEMENTACIONES FALLIDAS

7.12. USO DE LA APLICACIÓN

7.13. RENDIMIENTO DE LA APLICACIÓN

7.14. TIEMPO MEDIO DE DETECCIÓN

7.15. TIEMPO MEDIO DE RECUPERACIÓN

7.16. TIEMPO MEDIO PARA FALLAR

7.17. TIEMPO MEDIO ENTRE FALLOS

8. DEVOPS CON MICROSERVICIOS Y CONTENEDORES

8.1. CI CON MICROSERVICIOS Y CONTENEDORES

8.2. CD CON MICROSERVICIOS Y CONTENEDORES

8.3. DESPLIEGUE CONTINUO CON MICROSERVICIOS Y CONTENEDORES

8.4. IAC, CAC Y PAC CON MICROSERVICIOS Y CONTENEDORES

9. DEVOPS CON COMPUTACIÓN EN LA NUBE

9.1. CI CON COMPUTACIÓN EN LA NUBE

9.2. CD CON CLOUD COMPUTING

9.3. IMPLEMENTACIÓN CONTINUA CON COMPUTACIÓN EN LA NUBE

9.4. IAC, CAC Y PAC CON CLOUD COMPUTING

10. DESPLIEGUE CONTINUO

- 10.1. VERIFICACIÓN AUTOMATIZADA DE VERSIONES
- 10.2. GESTIÓN DE DEPENDENCIAS
- 10.3. PUESTA EN ESCENA DE RECURSOS DE INFRAESTRUCTURA
- 10.4. VERIFICACIÓN DE CUMPLIMIENTO DE RECURSOS DE INFRAESTRUCTURA

11. ENTREGA CONTINUA (CD)

- 11.1. PROCESO CONFIABLE REPETIBLE
- 11.2. AUTOMATIZAR TODO
- 11.3. CONTROL DE VERSIONES Y CONTROL DE CÓDIGO FUENTE TODO
- 11.4. PRIORICE LAS TAREAS DESAFIANTES
- 11.5. CALIDAD INCORPORADA
- 11.6. HECHO MEDIOS LIBERADOS
- 11.7. RESPONSABILIDAD COMPARTIDA DEL PROCESO DE LIBERACIÓN
- 11.8. MEJORA CONTINUA

12. INTEGRACIÓN CONTINUA (CI)

- 12.1. ENVIAR / CONSTRUIR / PROBAR
- 12.2. UTILICE UN REPOSITORIO DE CÓDIGO CENTRAL
- 12.3. ENVIAR CÓDIGO CON FRECUENCIA
- 12.4. CREAR MAS CONSTRUCCIONES
- 12.5. AUTOMATIZAR LA CREACIÓN DE COMPILACIÓN
- 12.6. AUTOMATIZAR LA IMPLEMENTACIÓN
- 12.7. PRUEBA EN LA PUESTA EN ESCENA
- 12.8. PRUEBA ANTES DE ENVIAR
- 12.9. NO ENVIAR POR CÓDIGO ROTO
- 12.10. REALICE LAS PRUEBAS ADECUADAS
- 12.11. COMPARTIR LOS RESULTADOS DE LA PRUEBA

13. CANALIZACIÓN DE ENTREGA DE CI / CD

- 13.1. DESARROLLAR
- 13.2. CONSTRUIR
- 13.3. INTEGRAR
- 13.4. PRUEBA
- 13.5. LANZAMIENTO
- 13.6. DESPLEGAR
- 13.7. FUNCIONAR

14. DESAFIOS DE DEVOPS

- 14.1. CULTURA EN CONFLICTO
- 14.2. DEV Y OPERACIONES FIRMEMENTE AISLADAS
- 14.3. ESTABLECIMIENTO DE MÉTRICAS UNIFICADAS

- 14.4. PASANDO POR ALTO LA SEGURIDAD Y LA GOBERNANZA
- 14.5. REQUISITOS
- 14.6. DEMASIADO ENFOQUE EN LAS HERRAMIENTAS
- 14.7. LIMITACIONES DE LA INFRAESTRUCTURA HEREDADA
- 14.8. DEPARTAMENTO DE DEVOPS NO DESEADO
- 14.9. TASA INACEPTABLE DE ADOPCIÓN

15. ETAPAS DEL CICLO DE VIDA DE DEVOPS

- 15.1. PLAN
- 15.2. CREAR
- 15.3. VERIFICAR
- 15.4. PAQUETE
- 15.5. LANZAMIENTO
- 15.6. CONFIGURAR
- 15.7. MONITOR
- 15.8. RETROALIMENTACIÓN

16. PRÁCTICAS CLAVE DE DEVOPS

- 16.1. INTEGRACIÓN CONTINUA
- 16.2. ENTREGA CONTINUA
- 16.3. DESPLIEGUE CONTINUO
- 16.4. MONITOREO CONTINUO
- 16.5. INFRAESTRUCTURA COMO CÓDIGO
- 16.6. CONFIGURACIÓN COMO CÓDIGO
- 16.7. POLÍTICA COMO CÓDIGO

17. HERRAMIENTAS DEVOPS, SISTEMAS Y REPOSITORIOS

- 17.1. HERRAMIENTA DE PLANIFICACIÓN
- 17.2. HERRAMIENTA DE CONTROL DE VERSIONES
- 17.3. HERRAMIENTA DE DESARROLLO
- 17.4. REPOSITORIO DE GESTIÓN DE CÓDIGO FUENTE
- 17.5. BASE DE DATOS DE GESTIÓN DE LA CONFIGURACIÓN (CMDB)
- 17.6. HERRAMIENTA DE APROBACIÓN DE CONFIGURACIÓN
- 17.7. SISTEMA DE SEGUIMIENTO DE PROBLEMAS
- 17.8. CONSTRUIR SERVIDOR
- 17.9. REPOSITORIO DE ARTEFACTOS
- 17.10. HERRAMIENTA DE IMPLEMENTACIÓN
- 17.11. HERRAMIENTA DE OPERACIONES
- 17.12. HERRAMIENTA DE GESTIÓN DE VERSIONES
- 17.13. SISTEMA DE POLÍTICAS
- 17.14. AGENTES DE MONITOREO
- 17.15. HERRAMIENTA DE COMENTARIOS
- 17.16. SERVIDOR CI
- 17.17. SERVIDOR DE IMPLEMENTACIÓN

- 17.18. HERRAMIENTA DE EMBALAJE
- 17.19. HERRAMIENTA DE INFORMES
- 17.20. HERRAMIENTA DE COMUNICACIÓN Y COLABORACIÓN

18. IMPACTOS DE DEVOPS

- 18.1. CULTURA
- 18.2. COLABORACIÓN
- 18.3. AUTOMATIZACIÓN
- 18.4. MEDICIÓN

19. BENEFICIOS DE DEVOPS

- 19.1. MAYOR CAPACIDAD DE RESPUESTA
- 19.2. ENTREGA RÁPIDA
- 19.3. ESCALABILIDAD RECEPTIVA
- 19.4. MAYOR CONFIABILIDAD
- 19.5. SEGURIDAD MEJORADA
- 19.6. OPERACIONES

20. FUNCIONES PRINCIPALES DE DEVOPS

- 20.1. DESARROLLO
- 20.2. QA

21. OBJETIVOS DE DEVOPS

- 21.1. ESTABLECIMIENTO DE INTEROPERABILIDAD ENTRE LOS EQUIPOS DE DESARROLLO Y OPERACIONES
- 21.2. ALINEACIÓN DE LOS OBJETIVOS DE SOFTWARE ENTRE LOS EQUIPOS DE DESARROLLO Y OPERACIONES
- 21.3. AUMENTO DE LA FRECUENCIA DE LAS VERSIONES DE SOFTWARE
- 21.4. AUMENTO DE LA VELOCIDAD Y LA CALIDAD DE LAS VERSIONES DE SOFTWARE
- 21.5. MEJORA DE LA CALIDAD DEL SOFTWARE EN RESPUESTA A LOS COMENTARIOS
- 21.6. REDUCCIÓN DE LOS PLAZOS DE ENTREGA

BENEFICIOS

- Al finalizar el curso los participantes tendrán un conocimiento avanzado sobre los modelos, prácticas y estrategias de DevOps.