1. Cual de las siguientes caracteristicas son mas apropiadas para describir un pipeline de integracion continua?
   1. Rapido, automatico, repetible, predecible, configurable
   2. Automatico, extensible, seguro
   3. Versatil, configurable, rápido
   4. Ninguno de los anteriores
2. Los pipelines de integración continua se deben ejecutar de forma automática desde las ramas principales. Seleccione tres tareas principales dentro del proceso
   1. Descarga de dependencias
   2. Ejecutar compilación rápida
   3. Ejecutar y publicar pruebas unitarias
   4. Realizar análisis estatico de código (sonar)
   5. Realizar un rebuild del pipeline para que tome los cambios configurados
3. Un equipo tiene la responsabilidad de desarrollar una nueva aplicación. Te han consultado cuantos pipelines se deben crear si han validado que una estrategia de monorepositorio es aplicable al proyecto y tendrán 8 microservicios inicialmente. Cual sería tu respuesta?
   1. Se debe construir un mono-pipeline para la integración de todos los componentes del repositorio.
   2. Se deben crear 8 definiciones de pipeline que correspondan a cada componente de la aplicación de forma independiente
   3. Se deben crear 4 definiciones de pipeline para todos los componentes de la aplicación. Uno para la fase de build, el segundo para la fase de pruebas, el tercero para la fase de análisis y un cuarto para la fase de publicación.
   4. Se deben crear dos definiciones de pipeline, las cuales serán nombradas CI y CD.
4. Como se conoce al mecanismo para ejecutar los pipelines de forma automática cada vez que se detecte un cambio en el repositorio?
   1. Procedure
   2. Trigger
   3. Executor
   4. Ninguno de los anteriores
5. Cuales son los pasos ideales de un pipeline de Integracion Continua?
   1. Descarga de dependencias, compilación, ejecucion de pruebas unitarias, análisis estático de código, publicación de artefacto
   2. Descarga de dependencias, compilación, ejecucion de pruebas de aceptacion, análisis estático de código, publicación de pruebas
   3. Compilacion, análisis estatico de código, publicación
   4. Compilacion, ejecución de pruebas unitarias, publicación de artefacto
6. Cual es el primer análisis de código que debe ser ejecutado?
   1. Style check
   2. Cobertura de código
   3. Pruebas Unitarias
   4. Ninguno de los anteriores
7. Cual es el tipo de prueba que debería ejecutarse desde un pipeline de Integración Continua?
   1. Pruebas E2E
   2. Pruebas de Aceptación
   3. Pruebas de Seguridad
   4. Pruebas Unitarias
8. Ha estado evaluando las NACL en su empresa. La mayoría de las NACL se configuran de la misma manera:
9. Traffic Allow
10. Traffic Deny

* All Traffic Deny

Si entra una solicitud. Como se evaluará?

1. Se permitirá la solicitud
2. Se utilizará la regla con el numero mas alto, una negación
3. Se evaluarán todas las reglas y el resultado final será denegar
4. El valor predeterminado denegará el tráfico
5. Después de una reunión del comité directivo de TI, se le asignó la responsabilidad de configurar un entorno hibrido para los recursos informáticos de la empresa.

Usted analiza los pros y los contras de varias tecnologías en función de los requisitos que se le dan. Los principales requisitos para impulsar esta selección son consideraciones de costos generales y la capacidad de reutilizar las conexiones de Internet existentes.

Que tecnología cumple estos requisitos?

1. Emparejamiento de VPC (VPC peering)
2. AWS VPN
3. AWS Direct Connect
4. AWS Direct Gateway
5. Usted trabaja para una gran institución financiera y se prepara para la recuperación ante desastres y los próximos simulacros de recuperación ante desastres

Un componente clave del plan de recuperación ante desastres serán las instancias de la base de datos y sus datos. Un objetivo de tiempo de recuperación (RTO) agresivo dicta que la base de datos debe replicarse sincrónicamente.

Que configuración puede cumplir con este requisito?

* 1. AWS Lambda para desencadenar una plantilla de CloudFormation en otra región
  2. RDS Multi-AZ
  3. RDS read replicas
  4. RDS Multi Region

1. Se le ha asignado la tarea de revisar el plan de recuperación ante desastres de su empresa debido a algunos requisitos nuevos. El factor determinante es que el objetivo de tiempo de recuperación se ha vuelto muy agresivo.

Debido a esto, se ha decidido configurar una implementación Multi-AZ en las bases de datos RDS MySQL. Sin relación con la recuperación ante desastres, se ha determinado que parte del tráfico de lectura debe descargarse de la base de datos maestra.

Que paso se puede dar para cumplir con este requisito?

* 1. Redirigir parte del tráfico de lectura de la Base de datos en espera (standby database)
  2. Agregue DAX a la solución para aliviar el exceso de tráfico de lectura
  3. Convierta a AURORA para permitir que el standby sirva tráfico de lectura
  4. Agregue replicas de lectura para descargar algo de tráfico.

1. Su empresa ha realizado un simulacro de recuperación ante desastres que no cumplió con el objetivo de tiempo de recuperación (RTO) deseado por la dirección ejecutiva. La falla se debió en gran parte a la cantidad de tiempo necesario para restaurar el funcionamiento adecuado de la base de datos. Usted le ha dado a la gerencia una recomendación para implementar la replicación de datos síncrona para la base de datos RDS para ayudar a cumplir con el RTO.

Cual de estas opciones puede realizar la replicación de bases de dato sincrónica en RDS?

* 1. AWS Database Migration
  2. DAX
  3. RDS Multi-AZ
  4. Leer replicas

1. Una empresa utilizará varias instancias EC2 para alojar varias aplicaciones de referencia. Se espera que estas aplicaciones reciban un trafico constante y relativamente bajo. Se espera que estas aplicaciones funcionen durante 3 años, momento en que se evaluaran las aplicaciones para su actualización.

Que tipo de EC2 cumplirá con este requisito considerando el costo como un factor adicional?

* 1. On-Demand
  2. Reserved
  3. Spot
  4. Dedicated Hosts

1. En el contexto de la arquitectura de software que es la complejidad esencial
   1. Es la cualidad de lo que está compuesto de diversos elementos interrelacionados. La complejidad tiende a ser utilizada para caracterizar un conjunto intrincado y difícil de comprender
   2. Es aquella que es inherente al problema que queremos solucionar, refleja la intención y el propósito por el cual fue creado el sistema y está expresada en términos del dominio del problema
   3. Es aquella que existe y es inyectada por la solución misma, es decir por los detalles y los medios por los cuales se materializa la solución, es altamente variable y depende en gran medida de detalles y decisiones tomadas al momento de crear la solución
   4. Ninguna de las anteriores
2. Con respecto al despliegue de aplicaciones en contenedores es correcto decir:
   1. Cuando usamos Docker no es necesario usar kubernetes ya que cumplen con la misma función de manejar contenedores e imágenes
   2. Kubernetes orquesta contenedores los cuales vienen de imágenes que pueden ser construidas mediante Docker o alguna herramienta compatible con el formato de imágenes estandarizado
   3. Docker no es compatible con el formato de imágenes de contenedores que usa Kubernetes si no exclusivamente con Docker Swarm como orquestador, para poder desplegar aplicaciones en kubernetes se debe generar un binario nativo para la plataforma ARM64 a través de CI/CD
3. Que significa el comando FROM de un dockerfile?
   1. El sistema operativo del host
   2. La imagen en la que se basa el contenedor
   3. El sistema operativo invitado
   4. El autor del actual contenedor
4. En entornos productivos y pre-productivos, cuando se desea ver los logs de una aplicación desplegada a través de kubernetes podemos:
   1. Usar el comando doker logs especificando la imagen del contenedor de kubernetes
   2. Consultar los logs en CloudWatch es caso que un extractor de logs esté configurado apropiadamente
   3. Usar el comando kubectl kubectl logs -f pod\_name
   4. Usar el comando docker stats y Docker logs sobre la instancia maestra del orquestador junto con el id de la imagen
   5. B y C son correctas
5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ es un servicio alojado en la nube que proporciona capacidades de almacenamiento y distribución para imágenes de contenedores públicas y privadas.
   1. Docker Cloud Orchestrator
   2. Docker Swarm Unified Repository
   3. Prisma Cloud Repo
   4. Elastic Container Registry
   5. Docker Compose Registry
6. Cuando se usan contenedores Docker podemos afirmar:
   1. Se crea una instancia virtualizada nueva a nivel de SO por cada contenedor en ejecución para garantizar el aislamiento a través del Hypervisor y los namespaces
   2. Los contenedores comparten siempre el mismo rango de puertos, es decir si un contenedor en ejecución usa el puerto 8080, ningún otro contenedor en el mismo host puede usar el mismo puerto 8080 ya que no se puede generar un mapeo diferente en el mismo host
   3. Los contenedores no comparten los puertos, es decir si un contenedor usa el puerto 8080, otro contenedor en el mismo host puede usar el mismo puerto 8080, solo difiere el mapeo si se desea exponer en el host
   4. Todos los contenedores se ejecutan como procesos en el mismo SO, sin virtualización, se usan Linux cgroups y namespaces para garantizar el aislamiento
   5. La C y D son correctas
7. Cuando hablamos de arquitecturas centradas en el dominio, (limpias, hexagonales) podemos decir que el principio SOLID fundamental para separar lo esencial de lo accidental es:
   1. Principio de responsabilidad única
   2. Principio abierto/cerrado
   3. Principio de inversión de dependencias
   4. Principio de sustitución de Liskov
   5. Ninguna de las anteriores
8. (Los/La/el)\_\_\_\_\_\_\_\_\_\_\_\_\_ nos permite(n) contener el impacto del cambio al no depender de los detalles, invirtiendo el flujo de control y la dirección de dependencia
   1. Inyección de dependencias
   2. Inversión de dependencias
   3. Controladores e inversores de flujo
   4. Control de impactos basado en polimorfismo
9. Cual de los siguientes describe mejor el patron de diseño “The HumbleObject pattern”?
   1. Este patron le permite a los “Unit Test” a separar las funcionalidades que son difíciles de probar de las que son fáciles de probar
   2. Este patron permite establecer las prioridades de pruebas de acuerdo a los casos de uso que contenga el sistema
   3. Este patron permite unir los casos de prueba en una sola carga de trabajo, sin importar la dificultad de la prueba
   4. Todos los anteriores
10. En el contexto de Arquitectura de Software, que es la complejidad accidental?
    1. Es la cualidad de lo que está compuesto de diversos elementos interrelacionados. La complejidad tiende a ser utilizada para caracterizar un conjunto intrincado y difícil de comprender
    2. Es aquella que es inherente al problema que queremos solucionar, refleja la intención y el propósito por el cual fue creado el sistema y está expresada en términos del dominio del problema
    3. Es aquella que existe y es inyectada por la solución misma, es decir por los detalles y los medios por los cuales se materializa la solución, es altamente variable y depende en gran medida de detalles y decisiones tomadas al momento de crear la solución
    4. Ninguna de las anteriores
11. En un sistema creado empleando arquitectura limpia, cuando se desea cambiar la tecnología de acceso a datos:
    1. Es necesario cambiar las entidades de dominio ya que en estas se encuentra las anotaciones de mapeo de datos del framework usado
    2. No es necesario cambiar las entidades de dominio si el cambio es netamente tecnológico ya que no existe dependencia directa del dominio con la tecnología de acceso a datos
    3. El cambio debe empezar desde los Entry-Points definidos y posteriormente modificando las interfaces segregadas del dominio
    4. El cambio de la base de datos representa un costo muy alto ya que la arquitectura limpia se genera a partir de los modelos entidad relación de la base de datos, por ende requiere una regeneración del proyecto.
12. Cual de los siguientes podría considerarse un mecanismo para cruzar las fronteras arquitectónicas del desarrollo de software?
    1. El uso de polimorfismos
    2. El uso de sentencias goto
    3. El uso de programación estructurada
    4. Todas las anteriores
13. EventStorming es un taller en el contexto de DDD en el cual:
    1. Se realiza una identificación de los objetos y las clases del sistema partiendo de sus métodos y valores de retorno
    2. Se identifican de forma colaborativa los eventos desde el punto de vista de las líneas de tiempo del negocio y posteriormente se agrupan y modelan tanto comandos como contextos delimitados
    3. Se realiza la estructuración del proyecto por parte del equipo de arquitectura de forma exclusiva
    4. Se elige el stack tecnológico de la solución con base en los lineamientos de ingeniería
14. Como calificaría la siguiente afirmación: DDD requiere el uso de una arquitectura de software de referencia y solo permite el uso de protocolos de comunicación asíncronos
    1. Verdadero
    2. Falso
15. En DDD se debe capturar el terreno existente. Aquí se debe describir el presente y no el futuro imaginado
    1. Bounded Map
    2. Bounded Context
    3. Subdomain Context
    4. Context Map
16. Cuando hablamos de microservicios como estilo arquitectónico, podemos decir:
    1. Los microservicios separan las responsabilidades de las capas técnicas de la aplicación, es decir, debemos tener microservicios de orquestación, persistencia y logging, cada uno siendo una unidad desplegable independiente
    2. Los microservicios representan unidades funcionales con la mayor autonomía posible, representando idealmente contextos delimitados identificados mediante DDD
    3. Los microservicios deben contener una sola funcionalidad expuesta, y deben contener cuanto mucho una sola entidad de negocio
    4. La A y la C son verdaderas
17. Que se busca cuando se realiza un ejercicio de modelado táctico?
    1. Entender y plasmar en un diagrama todos los flujos relevantes de negocio
    2. Tener un modelo de dominio claro y más refinado
    3. Tener una base clara del tipo de arquitectura que utilizaremos
    4. Estimar la infraestructura requerida
    5. Todos los anteriores
18. En relación con DDD podemos decir que los Domain Events:
    1. Solo tienen sentido cuando hablamos de arquitecturas asíncronas orientadas a objetos
    2. Son una forma de modelar el comportamiento de un sistema tomando como énfasis los resultados y comportamientos de negocio que son claramente observables. Su implementación concreta puede variar
    3. Son mensajes asincronos publicados por un microservicio hacia un bus de eventos y que solo pueden ser consumidos por un microservicio en el mismo dominio funcional
    4. A y C son correctas
19. Quienes deberían participar de las sesiones de modelado táctico?
    1. Expertos de dominio (negocio) y arquitecto de la solución
    2. Equipo implementador de la solucion y expertos de dominio (negocio)
    3. Equipo implementador de la solución y arquitecto de la solución
    4. Usuarios de negocio y equipo de pruebas especializadas