

Cuarta Calificada – Desarrollo de Software

Pregunta 1

- ¿Se necesita la infraestructura SOA para integrar los dos nuevos servicios?

Sí, ya que esto nos permite reutilizar sus elementos por medio de interfaces de servicios y integraría dos nuevos servicios que se implementan y se mantienen por separado, y permite que se comuniquen entre sí y trabajen en conjunto en el entorno SOA.

- El servicio de evaluación de campañas necesita manejar una gran cantidad de datos.
¿Sería mejor utilizar la replicación de datos, la integración a nivel de interfaz de usuario o las llamadas de servicio para acceder a grandes cantidades de datos?

Sería mejor las llamadas de servicio, de un servicio que se dedique concretamente a la extracción de datos para grandes cantidades.

- ¿Cuál de estas opciones de integración suele ofrecer SOA?

SOA suele ofrecer la llamadas de servicio ya que como su mismo nombre lo indica es una arquitectura orientada a servicios, que busca que estos sean independientes pero que puedan tener una buena integración entre ellos para formar parte de aplicaciones en su totalidad.

- ¿Debe el servicio integrarse al portal existente o tener su propia interfaz de usuario?

Debería integrarse al portal ya que al tratarse de una aplicación basada en microservicios necesita que estos se integren para formar la aplicación con todas sus funcionalidades, además que la parte de la integración se encargaría en team integración mientras que la creación del servicio sería del team CRM.

- ¿Cuáles son los argumentos a favor de cada opción?

Por parte de integrarse el beneficio sería que tendrías a este servicio como parte de todo la aplicación y podrían comunicarse entre diferentes servicios ya existentes y así tener una mejor funcionalidad.

Por parte de tener su propia interfaz de usuario se ahorrarían e trabajo de la integración además de que por ser un servicio por separado y más pequeño este sería mas fácil de administrar.

- ¿Debería implementar la nueva funcionalidad, el equipo de CRM?

Sí como ya lo mencione antes el team CRM se encargaría de la implementación del servicio en sí y que funcione todo correctamente y el team integración ya se encargaría de que el servicio se pueda integrar de manera eficiente y eficaz a toda la aplicación .

Pregunta 2

En un sistema basado en microservicios puede haber diferentes tipos de comunicación; sin embargo, debe haber un tipo de comunicación predominante. ¿Cuál escogerías? ¿Qué otros están permitidos además? ¿En qué situaciones? ¿Utilizarías la replicación de datos en un sistema basado en microservicios? ¿En qué áreas? ¿Cómo lo implementarías?

El protocolo HTTP y el protocolo REST son los que se utilizan con una mayor frecuencia. Yo escogería las API REST ya que es lo que la comunidad de programadores lo usa ya casi como un standard en sí. AMQP o TCP también están permitidos. Sí utilizaría la replicación de datos en microservicios pero solo en algunos casos tales como: cuando uso bases de datos nosql porque necesite que la búsqueda y entrega de datos sea más veloz y necesito que un servicio en específico necesite los datos con tal velocidad o cuando estos datos ya no cambien mucho de esta manera tendría una cierta cantidad de datos replicados necesarios para este servicio en específico.

Pregunta 4 (8 puntos)

En esta pregunta debes completar toda las actividades.

Completa las actividades 21 y 22 del curso. Muestra cada uno de los pasos seguidos.

Actividad 21:

```
docker run -d -p 8080:8080 -v $HOME/jenkins_home:/var/jenkins_home --name jenkins jenkins/jenkins
```

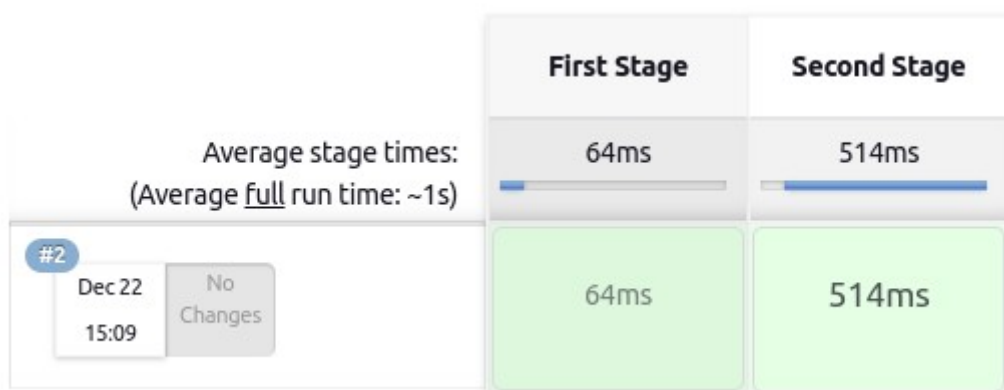
ya dentro despues vas al puerto 8080 y listo ingresas tu cuenta de jenkins.

Cambiamos el script de la actividad 20 por el siguiente:

```
1 pipeline {
2   agent any
3   stages {
4     stage('First Stage') {
5       steps {
6         echo 'Step 1. Hola mundo'
7       }
8     }
9     stage('Second Stage') {
10      steps {
11        echo 'Step 2. Segunda vez Hola'
12        echo 'Step 3. Tercera vez Hola'
13      }
14    }
15  }
16 }
```

y hacemos clic en Build Now.

Y tenemos lo siguiente:



Luego:

```
pipeline{
  agent any
  triggers { cron('* * * * *') }
  options { timeout(time: 5) }
  parameters {
    booleanParam(name: 'DEBUG_BUILD', defaultValue: true,
      description: 'construimos la depuracion?')
  }
  stages {
    stage('Example') {
      environment { NAME = 'Checha' }
      when { expression { return params.DEBUG_BUILD }}
      steps {
        echo "Hola $NAME"
        script {
          def browsers = ['chrome', 'firefox']
          for (int i = 0; i < browsers.size();++i) {
            echo "Prueba el ${browsers[i]} browser."
          }
        }
      }
    }
  }
  post { always { echo 'Hola muchas gracias' } }
```

y hace lo siguiente:

1. Utiliza cualquier agente disponible
2. Se ejecuta automáticamente cada minuto
3. Se detiene si la ejecución tarda más de 5 minutos
4. Solicita el parámetro de entrada booleano antes de comenzar
5. Establece Checha como la variable de entorno NAME
6. Hace lo siguiente, solo en el caso del parámetro de entrada True:
 - Imprime Hola Checha
 - Imprime Prueba el browser firefox
 - Imprime Prueba el browser chrome
7. Imprime Hola muchas gracias, independientemente de que haya algún error durante la ejecución.