

Return To Work

☰ Column

Consideraciones

Repositorio

- El código de la app se encuentra en la rama **dev** del repositorio **RTW_frontend** del Github del DoLab

Dependencias

- Ya que la aplicación fue creada para ser desplegada en un cluster de openshift, se necesita instalar appsody:

Appsody - Compose a Cloud Native Masterpiece

Appsody enables developers and teams to easily create and deploy apps to Kubernetes.

 <https://appsody.dev/>



- Appsody genera imagenes de docker, por lo que también hay que instalarlo.

Docker Desktop for Mac and Windows | Docker

The Docker Subscription Service Agreement has been updated. Our Docker Subscription Service Agreement includes a change to the terms for Docker Desktop. It remains free for small


 <https://www.docker.com/products/docker-desktop>



- Para interactuar con algunos servicios, se necesitará el cliente de terminal de IBM Cloud.

IBM Cloud Docs

Find documentation, API & SDK references, tutorials, FAQs, and more resources for IBM Cloud products and services.

 <https://cloud.ibm.com/docs/cli?topic=cli-getting-started>

Código

En el código, se deben modificar las credenciales de los servicios.

- En el archivo `app.js` se debe modificar las credenciales obtenidas desde App ID
- En el archivo `scripts/cloudant.js` se deben modificar las credenciales para acceder al cloudant

NOTA IMPORTANTE:

- En vista de que las credenciales están *hardcodeadas* en los archivos del proyecto y esto es un problema de seguridad, se recomienda modificar el acceso a las credenciales de forma que se obtengan desde las variables del entorno.
- La funcionalidad de notificaciones (vía slack) no ha sido probada, por lo que no se recomienda usarla antes de someterla a testing

Servicios

Los servicios utilizados para el nuevo despliegue son:

Container Registry


- se utilizará para subir la docker image de la app.
- Para generar la imagen, se debe ejecutar el comando:

```
appsody build
```

- para poder subir la imagen al container registry, se debe haber instalado el plugin en IBM Cloud CLI

IBM Cloud Docs

Find documentation, API & SDK references, tutorials, FAQs, and more resources for IBM Cloud products and services.

 <https://cloud.ibm.com/docs/Registry?topic=container-registry-cli-plugin-containerregcli>

- Antes de subir la imagen al `registry`, se necesita haber creado el proyecto y el namespace en IBM Cloud, por lo que hay que solicitar los permisos necesarios.
- La imagen de docker que se genera con el build, se debe renombrar de manera que contenga el nombre del servidor, el nombre del namespace y el nombre de la imagen. Esto puede hacerse usando el siguiente comando:

```
docker tag <source_image>:<tag> <region>.icr.io/<my_namespace>/<new_image_repo>:<new_tag>
```

- Finalmente, para hacer el push, se debe estar logeado en el `IBM cloud cli`, en `cr` y haber añadido el namespace al `cr`

```
ibmcloud login -sso
ibmcloud cr login
ibmcloud cr namespace-add <my_namespace>
docker push <region>.icr.io/<my_namespace>/<image_repo>:<tag>
```

AppID

En AppID se debe registrar una aplicación usando `SAML 2.0` y `Cloud Directory`

- **SAML 2.0:** Se debe descargar el XML y se debe subir en la página en la que se solicita la autorización para usar SAML con el W3ID. Se debe seleccionar la opción que indica que es un servicio en **producción**.
- En la configuración de SAML 2.0, se debe definir la url de callback de la api. Que en caso de ser local host, puede ser definida como `http://localhost:3000/` o la url de la App obtenida desde code engine.

Cloudfant

Se debe ingresar al dashboard de administración y crear las bases con los siguientes nombres:

- chequeos
- estadisticas_chequeos
- estadisticas_formularios
- persona
- plantilla_formulario
- respuestas_formulario

Considerar que basta solo crear la base datos, no es necesario crear las columnas.

- Dentro de plantilla_formulario se debe crear un documento con el siguiente contenido. Considerar que se debe mantener el `_id` entregado.

```
"versionFormulario": "0.2",
"preguntasRedactadas": {
  "pregunta1": "En los últimos 14 días ¿ha estado en contacto estrecho con una persona con
firmada de tener una nueva infección por coronavirus (COVID-19) o que está en cuarentena de
bido a un caso probable de COVID-19?",
  "pregunta2": "En los últimos 14 días, ¿su médico o autoridad local de salud le ha pedido
que se auto aísle?",
  "pregunta3": "En los últimos 14 días, ¿ha presentado alguno de los siguientes síntomas?
Tos, resfriado, fiebre (≥37,8°C / 100,04°F), falta de aliento, dificultad para respirar, d
olor de garganta, diarrea, cansancio, dolores y molestias, nueva pérdida de sabor u olfato"
}
```

Nota:

- Todas las Bases de datos, deben estar vacías luego de la creación. Es decir que no se deben crear registros a mano a excepción de `plantilla_formulario`.
- Un problema que puede pasar, es que la app no funcione al momento de acceder a las estadísticas. En dicho caso, se recomienda revisar los datos guardados en la DB `respuestas_formulario` en busca de algún registro que no contenga todos los campos.

Code Engine

1. Crear el Servicio
2. Se debe obtener la dirección de la imagen desde el `Container Registry`
3. Al momento de crear la aplicación en `ce` se debe seleccionar la opción container image

4. En la sección de `Image Reference` se debe pegar el link de la imagen
5. En la opción de `Listening port` se debe definir el puerto `3000`
6. Luego seleccionar la opción `create`
7. Luego de desplegar la app, se debe obtener el link de la página desde la sección endpoints desde la configuración de la app.
8. Este Link debe ser declarado como una variable del entorno con el nombre `PAGE_URL`
9. En caso de ser necesario, se debe definir la variable `SLACK_WEBHOOK_URL` como un string vacío o efectivamente la URL si es que se busca utilizar la funcionalidad.