

Estándares de despliegue de Productos Intelix BE-NodeJS y FE-NodeJS (Angular)

Prerrequisitos

Los ambientes de Calidad y Producción deberán de cumplir con las siguientes características:

- Si nuestro sistema Operativo es Linux (Debian, Ubuntu, etc.), esto deberán de tener instalados las siguientes herramientas: `vi`, `nano`, `wget`, `curl`, `authbind`, `ssh`, `scp`, `sshpass`, `git` y `git flow`. El usuario de despliegue debe de tener permiso `sudo`.
- NVM (Node Version Manager) es un gestor que nos permitirá instalar dos o más versiones de Node.js. según se necesite y, además, permitirá que tales versiones se instalan localmente en la carpeta `/home/usuario_de_despliegue`. Toda la configuración de NVM se guardará en la carpeta oculta `.nvm` en el home del usuario. Los archivos de configuración de NPM (*Node Package Manager*, instalado como parte de la distribución de Node.js) se guardarán igualmente en la carpeta oculta `.npm` en el home del usuario.
- NodeJS se recomienda que la versión instalada de Node.js con el gestor NVM sea por lo menos la LTS (*Long Term Support*, Soporte de Largo Plazo) o bien la versión utilizada por el equipo de desarrollo la cual debe ser igualmente por lo menos la LTS.
- PM2 (Gestor de servicios NodeJS). Está herramienta nos permite gestionar aplicativos server-side escrito con NodeJS. Todos los archivos de configuración serán guardados en la carpeta oculta `.pm2` en el home del usuario de despliegue.

Configuración del Servidor de Calidad.

Luego que se tenga configurado el servidor BE (NPM NodeJS) se procede a realizar la instalación general del PM2 ejecutando la siguiente sentencia

```
$ npm install pm2@latest -g
```

Para los casos de que nuestro servidor sea un contenedor este deberá de tener configurado las carpetas de persistencia, la recomendación de esta es



/home/usuario_de_despliegue contra una unidad física. Es de suma importancia que el repositorio de control de versiones git.intelix.biz sea alcanzable desde el ambiente de despliegue.

Ejecución del Despliegue BE

Para realizar el despliegue a cualquier ambiente llámese Calidad (Qa) y/o Producción, se realiza las siguientes tareas:

- Se debe de clonar del servidor de repositorio git.intelix.biz el Proyecto en la carpeta /home/usuario_de_despliegue.
- Luego de ser clonado el proyecto se deberá de ubicar en el directorio del proyecto y se debe de ejecutar el comando de git flow init -d. Con este comando generará la estructura de mapa de Git Flow (master, release, hotfix, support, develop, feature) y lo ubica de forma automática en la rama develop.
- Hacer check-out en la rama release (en caso de que el servidor sea Qa), en los casos que nuestro servidor sea Producción deberá de ubicarse en la rama master.
- Se deberá realizar pull para obtener los últimos cambios de la rama.
- Ejecutar la siguiente sentencia npm install, este leerá el archivo package.json de configuración del aplicativo e instalará todos los módulos y dependencias necesarios para la ejecución del aplicativo. Este no deberá de ser ejecutado como usuario sudo.
- Es necesario configurar el fichero .env, en el proyecto se encuentra un archivo de ejemplo que lleva por nombre .env.example, este deberá de ser copiado y renombrado cómo .env. Este fichero maneja las Environment Variable del aplicativo y este es leído por una dependencia que lleva por nombre dot-env.
- Asegurarse de que exista el archivo .gitignore.
- De hacer falta alguna configuración de último momento en relación a archivos de propiedades (.properties), variables de entorno, o similar, etc., hacer los cambios necesarios antes de pasar a la ejecución.
- Luego de validar que todas las revisiones se encuentra bien, es necesario ejecutar la siguiente sentencia \$pm2 start app.js -i 0, la cual levantará el aplicativo con balanceo de carga activado.
- Verificamos en nuestro servidor que este se encuentre funcionando.
http://ip_servidor:3000/

Prerrequisitos para despliegue de FrontEnd.

Para realizar el despliegue a cualquier ambiente llámese Calidad (Qa) y/o Producción, antes de iniciar el despliegue se debe de realizar las siguientes configuraciones:

- Crear una carpeta que lleve por nombre fe-service, en la carpeta /home/usuario_de_despliegue.
- Dentro de esta carpeta es necesario ejecutar el siguiente comando npm init, a ser enter hasta que culmine las preguntas de validación.

- Modificar el archivo package.json, este deberá de quedar de la siguiente forma:

```
{
  "name": ["Nombre del Proyecto"],
  "version": "1.0.0",
  "description": "[Descripción del Servidor y/o Servicio]",
  "main": "app",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node app.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^4.17.1"
  }
}
```

- Se ejecuta la siguiente sentencia npm install, esto instalará todas las dependencias y módulos.
- Se procede a crear un archivo app.js y dentro de está debe de tener el siguiente código:

```
const express = require('express');
const cors = require('cors');
const app = express();

const PORT = 4200;
app.use(express.urlencoded({ extended: false }));
app.use(express.json());
app.use(cors());
app.set('PORT', process.env.PORT || PORT);
app.use('/', express.static('public/abx-web'));
app.all('*', (req, res) => {
  res.redirect('/');
});
app.listen(app.get('PORT'), ()=>{
  console.log("...Starting");
  console.log("Ready, server listening port: ", app.get('PORT'));
});
```

- Se crea una carpeta que lleva por nombre *public*. En está carpeta se encontrará nuestro aplicativo ya construido.
- En esta carpeta debe de tener la siguiente estructura:
 - ./public
 - app.js
 - package.json

Ejecución del Despliegue

- Se debe de clonar del servidor de repositorio git.intelix.biz el Proyecto en la carpeta /home/usuario_de_despliegue.
- Luego de ser clonado el proyecto se deberá de ubicar en el directorio del proyecto y se debe de ejecutar el comando de git flow init -d. Con este comando generará la estructura de mapa de Git Flow (master, release, hotfix, support, develop, feature) y lo ubica de forma automática en la rama develop.
- Hacer check-out en la rama release (en caso de que el servidor sea Qa), en los casos que nuestro servidor sea Producción deberá de ubicarse en la rama master.
- Se deberá realizar pull para obtener los últimos cambios de la rama.
- Ejecutar la siguiente sentencia npm install, este leerá el archivo package.json de configuración del aplicativo e instalará todos los módulos y dependencias necesarios para la ejecución del aplicativo. Este no deberá de ser ejecutado como usuario sudo.
- Se deberá de compilar la aplicación Angular con la siguiente sentencia:
Para el caso de ambiente Qa:
`ng build --configuration=qa --outputPath=../fe-service/public`
Para el caso de ambiente de Producción:
`ng build --configuration=prod --outputPath=../fe-service/public`
- Una vez que este se encuentre listo, procedemos a ir a la carpeta /home/usuario_de_despliegue/fe-service y procedemos a ingresar la siguiente sentencia:
`$pm2 start app.js -i 0`
- Verificamos en nuestro servidor que este se encuentre funcionando.
http://ip_servidor:4200/