**Manual de despliegue**

**Aplicación Django:**

**Prerrequisitos:**

* Python 3 (La versión 2.7 genera algunos problemas de compatibilidad)
* Django 1.11.8
* Mysql

Abrir una consola de comandos sobre la carpeta root del proyecto y seguir los siguientes pasos:

* Ejecutar el comando “pip install -r requirements.txt” para bajar las dependencias necesarias para la ejecución.
* Ejecutar los siguientes comandos para hacer uso de Django REST framework
* pip install djangorestframework
* pip install markdown
* Pip install django-filter
* Ejecutar el siguiente comando para correr las migraciones y crear las tablas correspondientes en la base de datos (La base de datos ya debe de estar en ejecución en el puerto por defecto de mysql)

python manage.py makemigrations

* Para desplegar un servidor de pruebas basta con correr el comando:

python manage.py runserver

Esto generará un servidor en el puerto 800 que podrá ser probado en el de manera loca.

**Despliegue en AWS EC2:**

Pasos para desplegar en Amazon Web Services

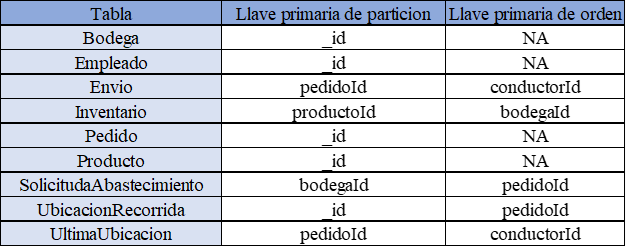
* Configurar una instancia EC2 con acceso remoto por medio de SSH y seleccionar un sistema operativo basado en Linux (Con el fin de que no ocurren conflictos de rutas con otros SO)
* Clonar un repositorio remoto con el código de la aplicación y ejecutar todos los pasos previamente mencionados para que la maquina remota quede configurada.
* Instalar el servicio de nginx para que funcione como proxy expuesto en el puerto 80.
* Abrir una consola en la carpeta del proyecto y ejecutar el comando “gunicorn -w 4 myapp:app” para el despliegue de un servidor de producción.
* Configurar el archivo de nginx para que apunte al servidor previamente desplegado y que este quede expuesto en el puerto 80 de la IP publica de la intancia EC2 creada.

**Aplicación de Funciones Lambda:**

**Prerrequisitos:**

Se debe configurar una cuenta en AWS que cuente con los siguientes servicios:

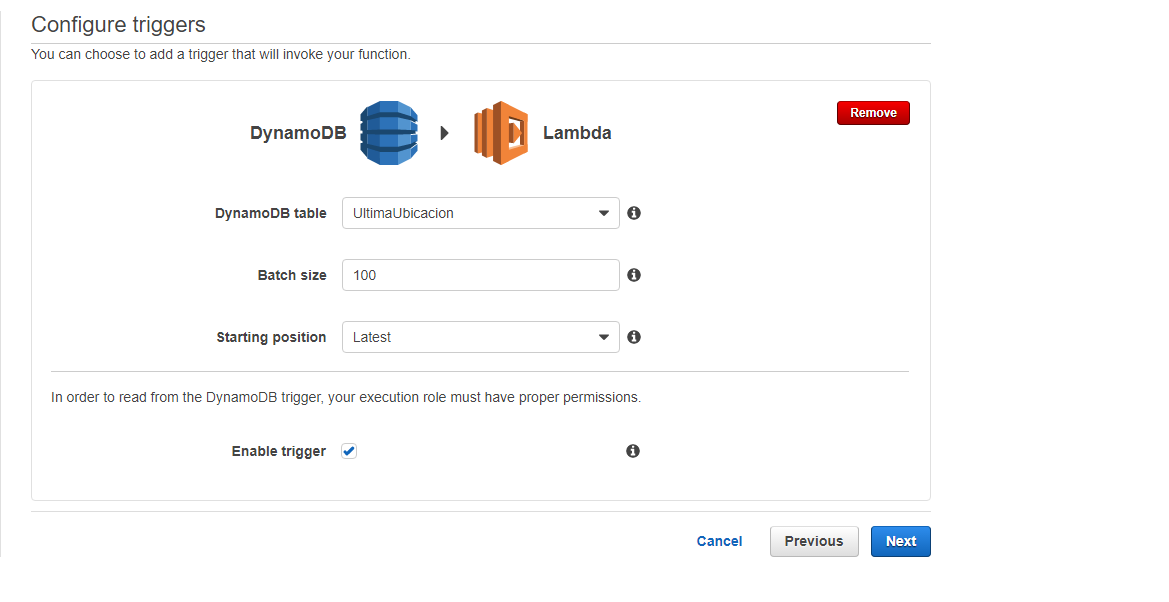
* DynamoDB: Adquirir el servicio de AWS dynamoDB y crear las siguientes Tablas:



* Cargar en dichas tablas los datos de la carpeta Modelos\_BD
* Simple Notification Service: Es necesario crear un topic para ser usado en la aplicación.
* Simple Email Service: Tener un correo y las listas de emails para los envíos.
* Serverless FrameWork: Una maquina con serverless framework ya configurado para poder ejecutar los comandos de despliegue de las funciones.

**Funciones:**

* ProcesarUltimaUbicacion: Crear una nueva función lambda seleccionando como “blueprint” a “dynamodb-process-stream”. Seleccionar la tabla UltimaUbicacion con un Batch
* size 100 y posición de inicio Lates. Después, asegurar que el disparador este habilitado, como en la siguiente imagen:



Después, es necesario asegurar los roles garantizando los permisos de lectura, escritura en Dynamo para la función.

* EnviarEmail: Seleccionar el “blueprint” de “sns-message” y en “SNS topic” el topic creado anteriormente.
* CrearEnvio: En el serverless.yml actualizar el arn de los recursos por los creados en dynamo. Ejecutar el comando “serverless deploy”
* Funciones de StepFunction: En el serverless.yml actualizar el arn de los recursos por los creados en dynamo. Ejecutar el comando “npm install” para descargar las librerias correspondientes. Finalmente, “serverless deploy” para subir los cambios a AWS.

**Cliente Web local:**

* Sera necesario contar con NodeJS instalado en la maquina
* Se debe de configurar cada una de las ARN de las funciones lambda previamente configuradas, estas están marcadas en el código.
* Abrir una consola sobre la carpeta /bin y ejecutar el comando “node install” para instalar las dependecias respectivas.
* Ejecutar el comando “node www” para iniciar un servidor en el localhost.