PIVOTAL

Anexo TFG-Alexa

Álvaro Rodríguez-Moroy Porcel

Pivotal

Pivotal es un servidor de aplicaciones en la nube. Se ha escogido este servidor de aplicaciones por su sencillez para configurar y utilizar.

Una vez se crea una cuenta asociada a un correo y un número de teléfono, ya se tiene una cuenta gratuita con 2 Gb de uso y 75$ de uso durante un año.

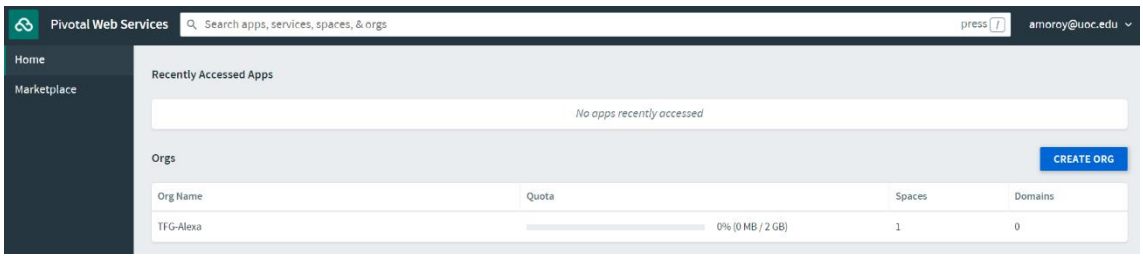

Ilustración 1 - Dashboard Pivotal

Se puede seguir el tutorial para ver como se despliega una aplicación con Pivotal:

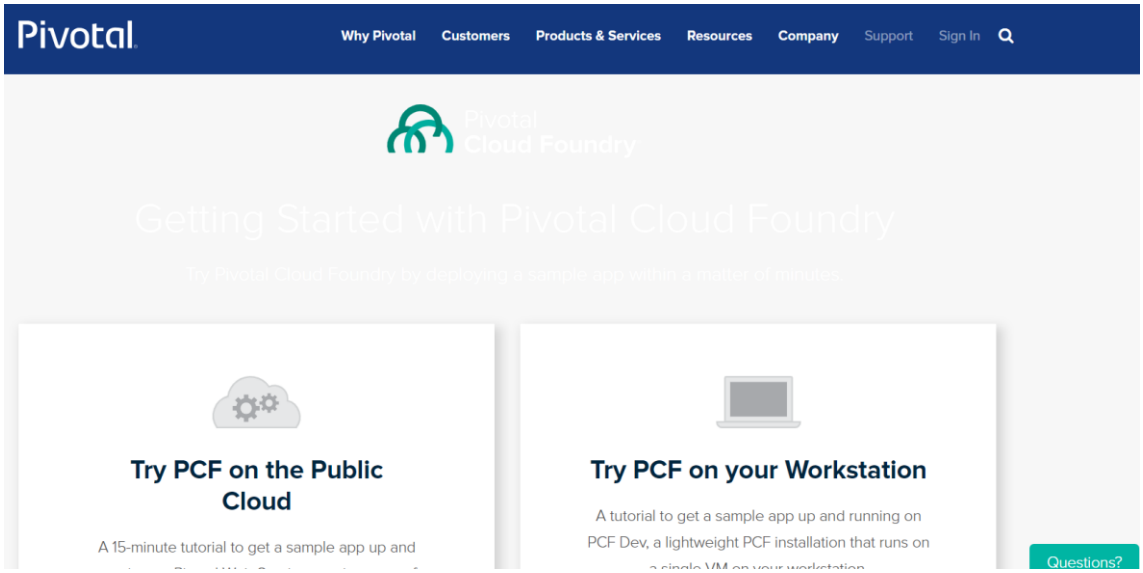

Ilustración 2 - Creación cuenta Pivotal

Para desplegar un servicio lo primero que se necesita es un cliente para conectarse a Pivotal y poder desplegar:

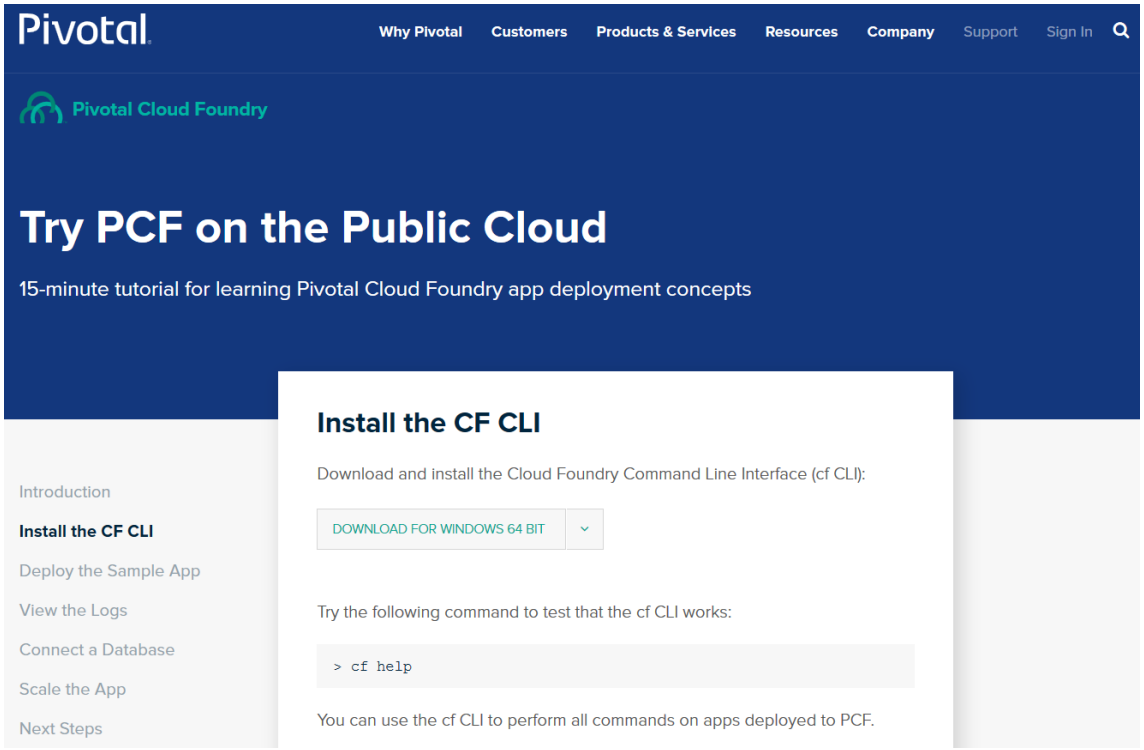

Ilustración 3 - Creación cliente de Pivotal

Una vez instalado el cliente, se puede abrir una consla (en este caso una consola de git) y probar que se ha instalado correctamente:

MINGW64:/c/Users/alvar

```
alvar@DESKTOP-CT2S706 MINGW64 ~
$ cf help
cf.exe versión 6.43.0+815ea2f3d.2019-02-20, Herramienta de línea de mandatos de
Cloud Foundry
Uso: cf.exe [opciones globales] mandato [argumentos...] [opciones de mandato]

Antes de empezar:
  config    login,l      target,t
  help,h    logout,lo

Ciclo de vida de la aplicación:
  apps,a        run-task,rt    events
  push,p        logs           set-env,se
  start,st      ssh            create-app-manifest
  stop,sp       app            delete,d
  restart,rs    env,e
  restage,rg    scale

Integración de servicios:
  marketplace,m        create-user-provided-service,cups
  services,s           update-user-provided-service,uups
  create-service,cs    create-service-key,csk
  update-service       delete-service-key,dsk
  delete-service,ds    service-keys,sk
  service              service-key
  bind-service,bs      bind-route-service,brs
  unbind-service,us    unbind-route-service,urs

Gestión de rutas y dominios:
  routes,r        delete-route    create-domain
  domains         map-route
  create-route    unmap-route

Gestión de espacios:
  spaces         create-space    set-space-role
  space-users    delete-space    unset-space-role

Gestión de la organización:
  orgs,o       set-org-role
  org-users    unset-org-role

Gestión de plugins de CLI:
  plugins           add-plugin-repo      repo-plugins
  install-plugin    list-plugin-repos

Mandatos que ofrecen los plugins instalados:

Opciones globales:
  --help, -h                          Mostrar ayuda
  -v                                  Imprimir el diagnóstico de solicitud de API
 en la salida estándar

Utilice 'cf help -a' para ver todos los mandatos.

alvar@DESKTOP-CT2S706 MINGW64 ~
$ |
```

Ilustración 4 - Cliente Pivotal

Y si se autentica con las credenciales de la cuenta creada:

```
alvar@DESKTOP-CT2S706 MINGW64 ~
$ cf auth amoroy@uoc.edu [redacted]
Punto final de API: https://api.run.pivotal.io
Autenticando...
Aceptar

Utilizar 'cf.exe target' para visualizar o definir su organización y espacio de destino.

alvar@DESKTOP-CT2S706 MINGW64 ~
$ |
```

Ilustración 5 - Autorización en Pivotal

Ahora se situa en el servidor creado en la nuve y en el espacio creado para que se despliegue:

```
alvar@DESKTOP-CT2S706 MINGW64 ~
$ cf target -o TFG-Alexa -s amoroy
punto final de api:   https://api.run.pivotal.io
versión de la api:    2.132.0
usuario:              amoroy@uoc.edu
org:                  TFG-Alexa
espacio:              amoroy
```

Ilustración 6 - Seleccionar objetivo en Pivotal

Y una vez situados, ya se puede desplegar el servicio con un push:

```
alvar@DESKTOP-CT2S706 MINGW64 /e/workspaceEclipseSim/TFG-Alexa/AlexaUOC (develop)
$ cf push TFG_Alexa -p target/AlexaUOC-0.0.1-SNAPSHOT.jar
Enviando por push la app TFG_Alexa a la organización TFG-Alexa / espacio amoroy como amoroy@uoc.edu...
Obteniendo información de la app...
Creando app con estos atributos...
+ nombre:        TFG_Alexa
  via de acceso:   E:\workspaceEclipseSim\TFG-Alexa\AlexaUOC\target\AlexaUOC-0.0.1-SNAPSHOT.jar
  rutas:
+   tfgalexa.cfapps.io

Creando app TFG_Alexa...
Correlacionando rutas...
Comparando archivos locales con caché remota...
Packaging files to upload...
Subiendo archivos...
 229.03 KiB / 229.03 KiB  100.00% 1s

Esperando que la API acabe de procesar los archivos...

Transfiriendo app y registros de rastreo...
   Downloading dotnet_core_buildpack_beta...
   Downloading dotnet_core_buildpack...
   Downloading nodejs_buildpack...
   Downloading go_buildpack...
   Downloading python_buildpack...
   Downloaded dotnet_core_buildpack_beta
   Downloading php_buildpack...
   Downloaded dotnet_core_buildpack
   Downloading binary_buildpack...
   Downloaded nodejs_buildpack
   Downloaded go_buildpack
   Downloading ruby_buildpack...
   Downloaded python_buildpack
   Downloading staticfile_buildpack...
   Downloaded binary_buildpack
   Downloading java_buildpack...
   Downloaded php_buildpack
   Downloaded staticfile_buildpack
   Downloaded java_buildpack
   Downloaded ruby_buildpack
   Cell f237c36d-0a62-4579-a51b-b11ee1d58145 creating container for instance 11e4916c-7112-43b1-9f38-557cd0d3c123
   Cell f237c36d-0a62-4579-a51b-b11ee1d58145 successfully created container for instance 11e4916c-7112-43b1-9f38-557cd0d3c123
   Downloading app package...
   Downloaded app package (14.2M)
   -----> Java Buildpack v4.18 (offline) | https://github.com/cloudfoundry/java-buildpack.git#a0df7d0
   -----> Downloading Jvmkill Agent 1.16.0_RELEASE from https://java-buildpack.cloudfoundry.org/jvmkill/bionic/x86_64/jvmkill-1.16.0_RELEASE.so (found in cache)
   -----> Downloading Open Jdk JRE 1.8.0_202 from https://java-buildpack.cloudfoundry.org/openjdk/bionic/x86_64/openjdk-1.8.0_202.tar.gz (found in cache)
          Expanding Open Jdk JRE to .java-buildpack/open_jdk_jre (1.8s)
```

Ilustración 7 - Desplegar aplicación en Pivotal

Aquí se da la información sobre donde se ha desplegado y como acceder a la aplicación:

```
Esperando a que se inicie la app...

nombre:                    TFG_Alexa
estado solicitado:         started
rutas:                     tfgalexa.cfapps.io
última subida:             Sat 23 Mar 10:17:40 CET 2019
pila:                      cflinuxfs3
paquetes de compilación:   client-certificate-mapper=1.8.0_RELEAS
va-opts java-security jvmkill-agent=1.16.0_RELEASE open-jdk-...

tipo:                     web
instancias:               1/1
utilización de memoria:   1024M
mandato de inicio:        JAVA_OPTS="-agentpath:$PWD/.java-buildp
.java-buildpack/container_security_provider:$PWD/.java-buildpack/
ildpack/open_jdk_jre/bin/java-buildpack-memory-calculator-3.13.0_
tion: $CALCULATED_MEMORY && JAVA_OPTS="$JAVA_OPTS $CALCULATED_MEM
```

Ilustración 8 - Desplegando aplicación en Pivotal 2

Y en la consola de Pivotal se puede ver como se ha desplegado la aplicación:

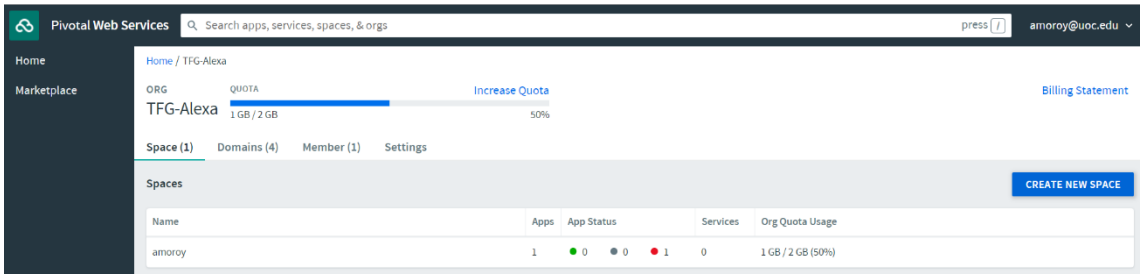

Ilustración 9 - Dashboard Pivotal 2

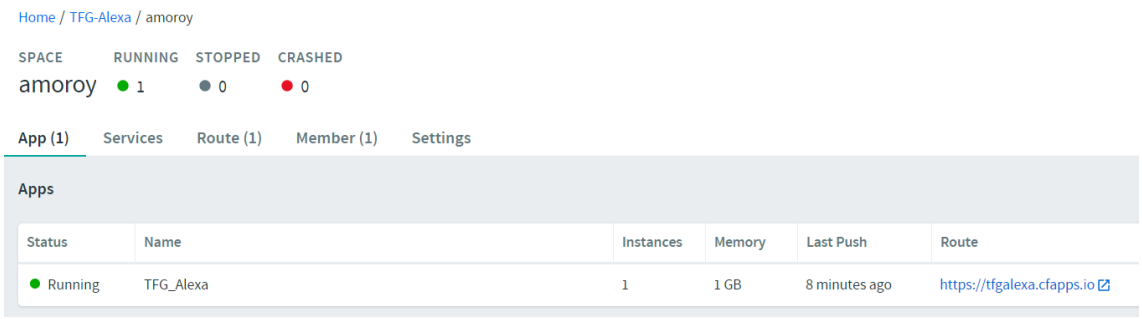

Ilustración 10 - Space de Pivotal

La aplicación que se ha desplegado no tiene más que un simple hola mundo más personalizado:

```
/**
 * @author amoroy
 *
 */
@RestController
public class FlashBriefingUOC {

    @RequestMapping(value="/flash")
    public String flashBriefing() {
        return "Bienvenido a la UOC";
    }
}
```

Ilustración 11 - Prueba de controlador en código

Y si se pide la aplicación via web se obtiene:

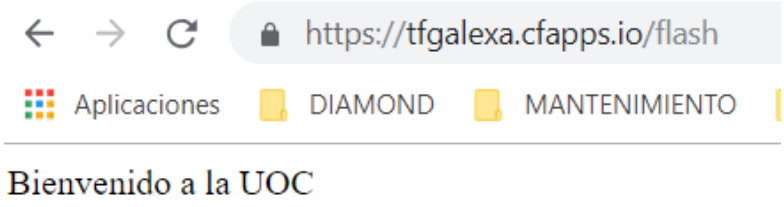

Ilustración 12 - Resultado prueba