



Universidad
Nacional
de Loja

**MANUAL DE
INSTALACIÓN DE
BONITA RUNTIME Y
SERVICIOS WEB DE
FIRMAEC CON
DOCKER**



Carrera de Ingeniería en
Sistemas / Computación

[INSTRUCTIVO]

Manual de instalación de Bonita Tomcat e Implementación de Servicios Web de FirmaEC con Docker.

Versión 1.1

Elaborado por:

Alex John Camba Macas

Raquel Jenny Lozano Chavez

Revisado por:

Ing. Pablo Fernando Ordoñez Ordoñez Mg. Sc.

Abril 2021
Loja - Ecuador

072-54 7252 Ext. 125
Ciudad Universitaria "Guillermo Falconí Espinosa",
Casilla letra "S", Sector La Argelia • Loja - Ecuador



Universidad
Nacional
de Loja

MANUAL DE INSTALACIÓN DE BONITA RUNTIME Y SERVICIOS WEB DE FIRMAEC CON DOCKER



Carrera de Ingeniería en
Sistemas / Computación

HISTORIAL DE REVISIONES

| Revisión | Fecha | Responsable | Descripción de la modificación |
|----------|------------|----------------------------|--|
| 1.0 | 23/02/2021 | Alex John Camba Macas | Versión inicial. |
| 1.1 | 04/04/2021 | Raquel Jenny Lozano Chavez | Se aplica el formato institucional a todo el documento |



Universidad
Nacional
de Loja

MANUAL DE INSTALACIÓN DE BONITA RUNTIME Y SERVICIOS WEB DE FIRMAEC CON DOCKER



Carrera de Ingeniería en
Sistemas / Computación

ÍNDICE DE CONTENIDOS

| | |
|---|-----------|
| INTRODUCCIÓN | 4 |
| Requisitos de hardware | 4 |
| Requisitos de Software | 4 |
| INSTALACIÓN Y CONFIGURACIÓN DE DOCKER..... | 4 |
| Instalación de Docker en CentOS 7 | 4 |
| Instalación de Docker Compose V2 en CentOS 7 | 6 |
| Ejecución de servicios Web para Firma Digital | 6 |
| BONITA..... | 9 |
| Ejecución de servicios de Bonita Runtime..... | 9 |
| GLOSARIO DE TERMINOS..... | 12 |



Universidad
Nacional
de Loja

MANUAL DE INSTALACIÓN DE BONITA RUNTIME Y SERVICIOS WEB DE FIRMAEC CON DOCKER



Carrera de Ingeniería en
Sistemas / Computación

INTRODUCCIÓN

El documento contiene los pasos obligatorios y necesarios para el proceso de instalación y configuración del Docker, para ejecutar en la plataforma Bonita.

Docker es una plataforma de software que permite crear, probar e implementar aplicaciones de forma rápida. Funciona de manera similar a una máquina virtual, virtualiza el hardware del servidor. Se debe instalar Docker en cada servidor y sus comandos son sencillos de utilizar en el momento de crear, iniciar o detener contenedores.

REQUISITOS

Se recomienda levantar una infraestructura con las siguientes características:

Requisitos de hardware

| Tipo | Mínimo | Recomendado |
|-----------------------|------------------|------------------------|
| Procesadores | 4 núcleos de CPU | 4 núcleos de CPU o más |
| Memoria (RAM) | 4 GB | 8 GB o más. |
| Espacio de disco duro | 10 GB | 30 GB o más. |

Tabla 1. Requisitos de Hardware

Requisitos de Software

| Componente | Recomendado | Versión |
|--------------------------|---|----------------------------------|
| Sistema Operativo | Centos | 7, 8 |
| Servidor de aplicaciones | Apache Tomcat | 9.0.X (X>52) |
| Máquina Virtual de Java | OpenJDK | 11 |
| Base de Datos | PostgreSQL | 12.6 y superior en la línea 12.x |
| Navegador Web | Mozilla Firefox, Google Chrome, Microsoft Edge. | Ultima version. |

Tabla 2. Requisitos de Hardware

INSTALACIÓN Y CONFIGURACIÓN DE DOCKER

Instalación de Docker en CentOS 7



Universidad
Nacional
de Loja

MANUAL DE INSTALACIÓN DE BONITA RUNTIME Y SERVICIOS WEB DE FIRMAEC CON DOCKER



Carrera de Ingeniería en
Sistemas / Computación

Paso 1: Instalar una serie de paquetes, antes de instalar Docker

```
sudo yum install -y yum-utils
```

Imagen 1. Instalación de paquetes YUM

Paso 2: Utilizar el siguiente comando para configurar el repositorio oficial estable

```
sudo yum-config-manager \  
--add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

Imagen 2. Configuración de repositorio.

Paso 3: Instalar Docker Engine

```
sudo yum install -y docker-ce docker-ce-cli containerd.io
```

Imagen 3. Instalación de Docker Engine.

Paso 4: Habilitar e iniciar Docker

```
sudo systemctl start docker  
sudo systemctl enable docker.service  
sudo systemctl enable containerd.service
```

Imagen 4. Iniciación de Docker.

Verificar que Docker Engine se encuentre instalado correctamente, ejecutando la imagen de prueba *hello-world*.

```
sudo docker run hello-world
```

Imagen 5. Verificación de Docker Engine.

Paso 5: Crear el grupo Docker y agregar al usuario actual a dicho grupo.

```
sudo groupadd docker  
sudo usermod -aG docker $USER
```

Imagen 6. Creación del equipo Docker.

Paso 6: Cerrar sesión en CentOS y volver a iniciarla para actualizar los cambios

Puede ejecutarse el siguiente comando para activar los cambios en los grupos:

```
newgrp docker
```

Imagen 7. Activación de cambios.



Universidad
Nacional
de Loja

MANUAL DE INSTALACIÓN DE BONITA RUNTIME Y SERVICIOS WEB DE FIRMAEC CON DOCKER



Carrera de Ingeniería en
Sistemas / Computación

Verificar que se puede ejecutar comandos Docker sin *sudo*.

```
docker run hello-world
```

Imagen 8. Verificación de ejecución de comando Docker.

Instalación de Docker Compose V2 en CentOS 7

Paso 1: Ejecutar el siguiente comando para descargar la versión estable actual de *Docker Compose*.

```
mkdir -p ~/.docker/cli-plugins/  
curl -SL https://github.com/docker/compose/releases/download/v2.2.3/docker-  
compose-linux-x86_64 -o ~/.docker/cli-plugins/docker-compose
```

Imagen 9. Descarga de Docker Compose.

Paso 2: Asignar permisos de ejecución al binario de Docker Compose.

```
chmod +x ~/.docker/cli-plugins/docker-compose
```

Imagen 10. Asignación de permisos.

Paso 3: Verificar la versión instalada

```
docker compose version
```

Imagen 11. Comprobación de versión.

Se puede comprobar Compose V2 reemplazando el guión (-) con un espacio y ejecutando *docker compose* en lugar de *Docker-compose*.

Ejecución de servicios Web para Firma Digital

Paso 1: Clonar o descargar el siguiente repositorio de recursos.

```
git clone https://github.com/alexjcm/certificacion-electronica-recursos.git  
cd certificacion-electronica-recursos/00_docker
```

Imagen 12. Clonación de repositorio de recursos.

Paso 2: Configurar variables de entorno



Universidad
Nacional
de Loja

MANUAL DE INSTALACIÓN DE BONITA RUNTIME Y SERVICIOS WEB DE FIRMAEC CON DOCKER



Carrera de Ingeniería en
Sistemas / Computación

Luego, en la carpeta *00_docker/* se debe crear un archivo denominado **env** y agregar todas las variables de entorno listadas en la **Tabla 3**.

| VARIABLE | DESCRIPCIÓN |
|-------------------|---|
| POSTGRES_PASSWORD | Establece la contraseña de Postgresql. |
| DB_NAME_SIGNATURE | Define el nombre de la base de datos de Firma Digital. |
| DB_USER_SIGNATURE | Define el nombre del usuario de la base de datos de Firma Digital. |
| DB_PASS_SIGNATURE | Define el nombre de la contraseña del usuario de la base de datos de Firma Digital. |
| ADMIN_NAME | Establece el nombre de usuario administrador de Wildfly. |
| ADMIN_PASSWORD | Define la contraseña del usuario administrador de Wildfly. |
| JWT_KEY_SIGNATURE | Establece la llave secreta. |

Tabla 3. Credenciales de Postgresql, Wildfly y base de datos de Firma Digital

Paso 3: En el mismo subdirectorío ejecutamos el siguiente comando:

```
docker volume create --name=wildfly-static
docker compose -f docker-compose-wildfly.yml up --build -d
```

Imagen 13. Ejecución de comando.

El archivo **docker-compose-wildfly.yml** contiene el siguiente script:

```
services:
  db:
    build: ./postgres
    container_name: dev-postgres
    restart: always
    ports:
      - 5432:5432
    environment:
      - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}
      - PGDATA=/var/lib/postgresql/data/pgdata
      - DB_NAME_SIGNATURE=${DB_NAME_SIGNATURE}
      - DB_USER_SIGNATURE=${DB_USER_SIGNATURE}
```



```
- DB_PASS_SIGNATURE=${DB_PASS_SIGNATURE}
volumes:
- ${HOME}/postgres-data:/var/lib/postgresql/data

wildfly:
build:
context: ./wildfly
args:
- DB_USER_SIGNATURE=${DB_USER_SIGNATURE}
- DB_PASS_SIGNATURE=${DB_PASS_SIGNATURE}
- DB_NAME_SIGNATURE=${DB_NAME_SIGNATURE}
- ADMIN_NAME=${ADMIN_NAME}
- ADMIN_PASSWORD=${ADMIN_PASSWORD}
- JWT_KEY_SIGNATURE=${JWT_KEY_SIGNATURE}
container_name: dev-wildfly
depends_on:
- db
ports:
- 8180:8180
- 9990:9990
volumes:
- wildfly-static:/opt/wildfly-static

volumes:
wildfly-static:
external: true
# Create the volume externally using: docker volume create --name=wildfly-
static
```

Imagen 14. Contenido del archivo docker-compose-wildfly.yml.

Paso 4: Generar un hash SHA256 a partir de una clave e insertarlo en la base de datos:

Opción 1: Puede utilizar los siguientes comandos para generar tu hash SHA256:

```
echo "La clave a utilizar en este ejemplo es: secure_password"
export API_KEY_HASH=$(echo -n secure_password | sha256sum | cut -d' ' -f1)
echo -e "El hash SHA256 generado a partir de la clave, que se insertará en la base de datos, es:\n$API_KEY_HASH"
```

Imagen 15. Generar hash SHA256

Opción 2: También puede utilizar esta página para generar el hash SHA256 <https://hash.online-convert.com/es/generador-sha256>.



Universidad
Nacional
de Loja

MANUAL DE INSTALACIÓN DE BONITA RUNTIME Y SERVICIOS WEB DE FIRMAEC CON DOCKER



Carrera de Ingeniería en
Sistemas / Computación

Insertar el siguiente registro en la tabla sistema de la base de datos de firma digital (para desarrollo o producción)

```
docker exec -it dev-postgres bash
psql signature_db user_db_sign

INSERT INTO sistema(id, url, apikey, apikeyrest, descripcion, nombre) VALUES (1,
'http://localhost:7776/recepcion/rest', '$API_KEY_HASH', '$API_KEY_HASH',
'Módulo de certificación electrónica', 'mce');
```

Imagen 16. Inserción de registro en la tabla de BD.

BONITA

Ejecución de servicios de Bonita Runtime

Paso 1: Agregar variables de entorno.

El paquete Tomcat es un archivo .zip que contiene el servidor de aplicaciones Se debe acceder al subdirectorio denominado 00_docker/ ubicado dentro del directorio de recursos clonado anteriormente, luego se debe agregar al archivo. env configurado anteriormente todas las variables de entorno descritas en la Tabla 4.

| Variable | Descripción |
|--------------------|---|
| DB_NAME_BONITA | Establece el nombre de la base de datos de Bonita |
| BIZ_DB_NAME_BONITA | Define el nombre de la base de datos de negocios Bonita. |
| DB_USER_BONITA | Establece el nombre de usuario de las bases de datos de Bonita |
| DB_PASS_BONITA | Establece la contraseña del usuario de las bases de datos de Bonita |
| TENANT_LOGIN | Define el nombre de usuario del super administrador de Bonita. |
| TENANT_PASSWORD | Establece la contraseña del usuario técnico de Bonita. |



| | |
|-------------------|--|
| PLATFORM_LOGIN | Define el nombre de usuario del administrador de la plataforma Bonita. |
| PLATFORM_PASSWORD | Establece la contraseña del administrador de la plataforma Bonita. |

Tabla 4. Credenciales de Bonita

Paso 2: En el mismo subdirectorio ejecutamos el siguiente comando:

```
docker volume create --name=bonita-vol  
docker compose -f docker-compose-bonita.yml up --build -d
```

Imagen 17. Ejecución de comando.

El archivo **docker-compose-bonita.yml** contiene el siguiente script:

```
db:  
  build: ./postgres  
  container_name: dev-postgres  
  restart: always  
  ports:  
    - 5432:5432  
  environment:  
    - POSTGRES_PASSWORD=${POSTGRES_PASSWORD}  
    - PGDATA=/var/lib/postgresql/data/pgdata  
    - DB_NAME_SIGNATURE=${DB_NAME_SIGNATURE}  
    - DB_USER_SIGNATURE=${DB_USER_SIGNATURE}  
    - DB_PASS_SIGNATURE=${DB_PASS_SIGNATURE}  
  volumes:  
    - ${HOME}/postgres-data:/var/lib/postgresql/data  
  
bonita:  
  image: bonita:2021.2-u0  
  container_name: dev-bonita  
  ports:  
    - 80:8080  
  environment:  
    - POSTGRES_ENV_POSTGRES_PASSWORD=${POSTGRES_PASSWORD}  
    - DB_VENDOR=postgres  
    - DB_HOST=db  
    - TENANT_LOGIN=${TENANT_LOGIN}  
    - TENANT_PASSWORD=${TENANT_PASSWORD}  
    - PLATFORM_LOGIN=${PLATFORM_LOGIN}
```



```
- PLATFORM_PASSWORD=${PLATFORM_PASSWORD}
- REST_API_DYN_AUTH_CHECKS=false
- TZ=America/Guayaquil
- DB_NAME=${DB_NAME_BONITA}
- DB_USER=${DB_USER_BONITA}
- DB_PASS=${DB_PASS_BONITA}
- BIZ_DB_NAME=${BIZ_DB_NAME_BONITA}
- BIZ_DB_USER=${DB_USER_BONITA}
- BIZ_DB_PASS=${DB_PASS_BONITA}
volumes:
- bonita-vol:/opt/bonita

restart: on-failure:2
depends_on:
- db
entrypoint:
- bash
- -c
- |
    sed -i "s/defaultTimeout">180/defaultTimeout">300/"
/opt/bonita/BonitaCommunity-2021.2-u0/server/conf/jbossts-properties.xml
set -e
echo 'Waiting for Postgres to be available'
export PGPASSWORD=${POSTGRES_ENV_POSTGRES_PASSWORD}
maxTries=10
while [[ "$$maxTries" -gt 0 ]] && ! psql -h ${DB_HOST} -U 'postgres' -c '\l'; do
    let maxTries--
    sleep 1
done
if [[ "$$maxTries" -le 0 ]]; then
    echo >&2 'error: unable to contact Postgres after 10 tries'
    exit 1
fi
exec /opt/files/startup.sh

volumes:
bonita-vol:
external: true
# Create the volume externally using: docker volume create --name=bonita-vol
```

Imagen 88. Script del archivo Docker.



Universidad
Nacional
de Loja

MANUAL DE INSTALACIÓN DE BONITA RUNTIME Y SERVICIOS WEB DE FIRMAEC CON DOCKER



Carrera de Ingeniería en
Sistemas / Computación

GLOSARIO DE TERMINOS

| Término | Descripción |
|------------------------|---|
| Centos | Es un sistema operativo de código abierto, basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito. |
| Wildfly | Conocido antes como JBoss As, o solo JBoss, es un servidor Open Source de aplicaciones Java EE. Es útil para crear, implementar y hospedar aplicaciones y servicios Java. Además, maneja servlets, JSP, EJB y JMS. |
| OpenJDK | Es la versión libre de la plataforma de desarrollo Java bajo concepto de lenguaje orientado a objetos. |
| Docker | Permite crear "contenedores". Estos contenedores de Docker se definen como máquinas virtuales ligeras, menos exigentes con los chips y memorias de los equipos donde se ejecutarán. |
| REST | Define un conjunto de principios arquitectónicos por los que se pueden diseñar servicios Web que se centran en los recursos de un sistema, lo que incluye la forma en que los estados de los recursos se dirigen y transfieren a través de HTTP por un amplio rango de clientes que están escritos en diferentes lenguajes. |
| Repositorio YUM | Es una aplicación de centos que permite instalar / desinstalar otras aplicaciones para Linux que vienen empaquetadas bajo el formato RPM oriundo de Red Hat. |
| systemd | Systemd es un conjunto de demonios o daemons de administración de sistema, bibliotecas y herramientas diseñados como una plataforma de administración y configuración central para interactuar con el núcleo del Sistema operativo GNU/Linux. |

Tabla 5. Glosario de términos





Universidad
Nacional
de Loja

**MANUAL DE
INSTALACIÓN DE
BONITA RUNTIME Y
SERVICIOS WEB DE
FIRMAEC CON
DOCKER**



Carrera de Ingeniería en
Sistemas / Computación

| Acción | Estudiante | Firma |
|-----------|----------------------------|---|
| Elaborado | Alex John Camba Macas |  |
| | Raquel Jenny Lozano Chavez |  |