

Fomento a la construcción de capacidades en relación a
las políticas públicas apoyadas por Contratos de Reforma Sectorial (CRS) en
Bolivia 2014/25

Fortalecimiento de la capacidad institucional en los sectores de desarrollo integral con coca, tráfico ilícito de drogas y seguridad alimentaria para una eficiente gestión del apoyo presupuestario sectorial en Bolivia

*LCN-025 – Fortalecimiento de capacidad institucional de
DIPREVCON en temas de sistemas de monitoreo y seguimiento
para realizar un diagnóstico situacional y hacer propuestas de
acciones dirigidas al personal a cargo de presupuestación,
seguimiento y monitoreo.*

Anexo 5: MANUAL DE INSTALACIÓN

Contrato n° DCI/LA/2017/392-699

N° de identificación: EuropeAid/138320/IH/SER/BO



Proyecto Financiado por la:
Unión Europea



Implementada por el consorcio:
AGRER — DT Global — Cowater
Bruselas, noviembre 2022

Disclaimer:

Este informe ha sido elaborado por AGRER/DT Global/Cowater con financiamiento de la Unión Europea. Las opiniones aquí expresadas son del consultor y no expresan necesariamente las de la Comisión Europea.

Fortalecimiento de la capacidad institucional en los sectores de desarrollo integral con coca, tráfico ilícito de drogas y seguridad alimentaria para una eficiente gestión del apoyo presupuestario sectorial en Bolivia (DITISA)

Contrato N° DCI/LA/2017/392-699

Misión LCN-025: Fortalecimiento de capacidad institucional de DIPREVCÓN en temas de sistemas de monitoreo y seguimiento para realizar un diagnóstico situacional y hacer propuestas de acciones dirigidas al personal a cargo de presupuestación, seguimiento y monitoreo.

Manual de Instalación

Autor: Henry Alejandro Taby Zenteno

La Paz / Bruselas, 13 de noviembre 2022

Consortio AGRER – DT GLOBAL – COWATER





CONTENIDO

Contenido

1. INTRODUCCIÓN	5
2. OBJETIVO	5
3. INSTALACIÓN DE CENTOS 8	5
4. CONFIGURACIÓN PREVIA	10
a. ACTUALIZACIÓN DEL SISTEMA OPERATIVO	10
b. DESHABILITAR EL FIREWALL	10
c. DESHABILITAR EL SELINUX.....	11
5. INSTALACIÓN DE DOCKER	12
a. INSTALACION DE CONTAINERD.IO	13
b. DOCKER COMPOSE	14
6. INSTALACIÓN DE COMPONENTES ADICIONALES	15
a. INSTALACIÓN DE GIT	15
b. PHP	16
7. INSTALACIÓN DE LA APLICACIÓN	18
a. CREAR EL DIRECTORIO DE ARCHIVOS.....	19
b. DOCKER.....	19
c. SERVICIOS CORRIENDO EN DOCKER.....	20
d. APLICACIÓN WEB.....	20



1. INTRODUCCIÓN

Cada sistema informático requiere tener diferentes documentos importantes para la implementación y puesta en producción, ya sean manuales de usuarios de acuerdo a los diferentes perfiles y sobre el manual de instalación del sistema, donde este documento tendrá el contenido de paso a paso sobre la instalación de otras herramientas adiciones.

2. OBJETIVO

El presente manual de instalación, tiene como objetivo proveer una guía sobre la instalación del sistema informático en servidores y los requerimientos de herramientas adiciones.

3. INSTALACIÓN DE CENTOS 8

En este punto se detalla los pasos a seguir para instalar Centos 8 en un servidor virtual, para después instalar docker e implementar el sistema informático.

Inicialmente de debe descargar el ISO de centos 8 desde la web de la siguiente dirección:

<https://www.centos.org/download/> .

Después de realizar la configuración de un servidor virtual y/o preparar un servidor físico que contendrá el sistema de información, realizaremos las siguientes acciones.

Al arrancar el servidor junto al instalador de Centos 8, veremos la siguiente pantalla.



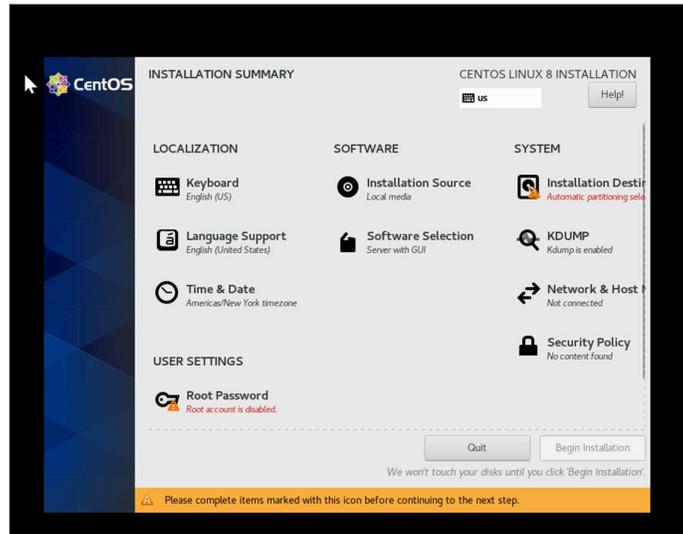
Donde se debe seleccionar: “Install CentOS Linux 8”, después se deberá elegir el idioma en el interfaz, como caso nuestro elegir “English”.



Luego se debe configurar los parámetros de instalación que se visualizan a continuación:

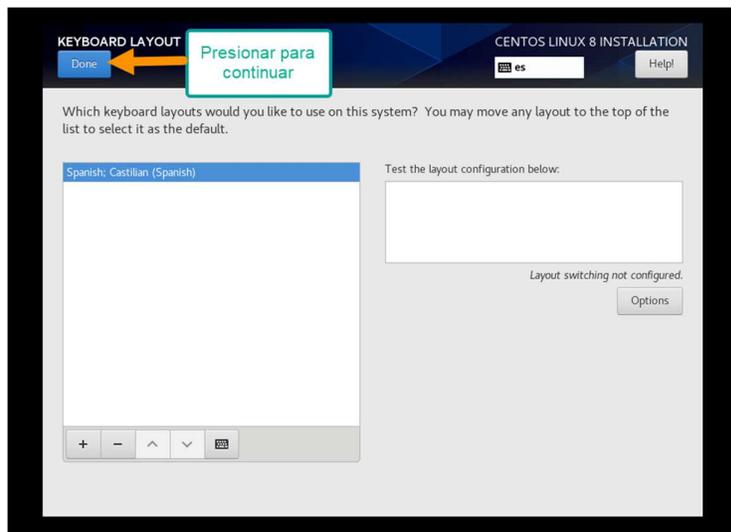


Fortalecimiento de la capacidad institucional en los sectores de desarrollo integral con coca, tráfico ilícito de drogas y seguridad alimentaria para una eficiente gestión del apoyo presupuestario sectorial – Contrato n° DCI/LA/2017/392-699



Se detallan de la siguiente manera:

- **Keyboard**, en esta opción se deberá eliminar el idioma inglés y añadir la opción “Spanish”.



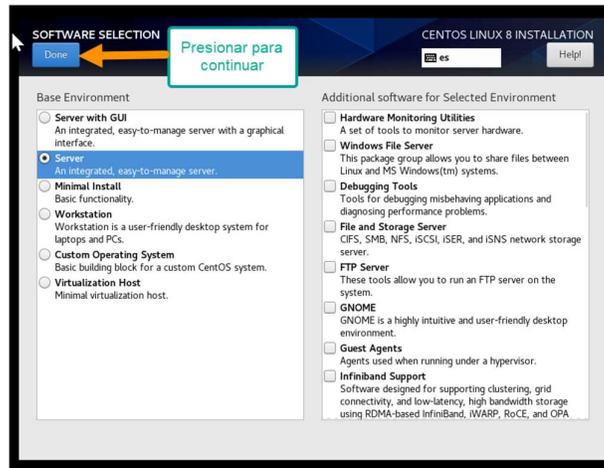
- **Time & Date**, en esta opción seleccionar “La Paz”, haciendo click sobre el mapa o en otro caso seleccionando sobre los campos “Región” y “Ciudad”:



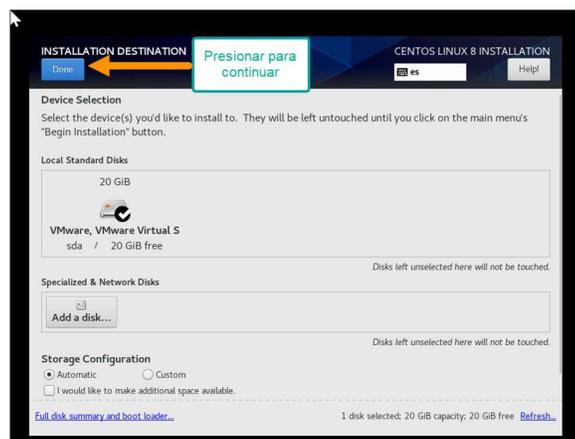
Fortalecimiento de la capacidad institucional en los sectores de desarrollo integral con coca, tráfico ilícito de drogas y seguridad alimentaria para una eficiente gestión del apoyo presupuestario sectorial – Contrato n° DCI/LA/2017/392-699



- **Software Selection**, en este punto se debe seleccionar la opción “Server”.

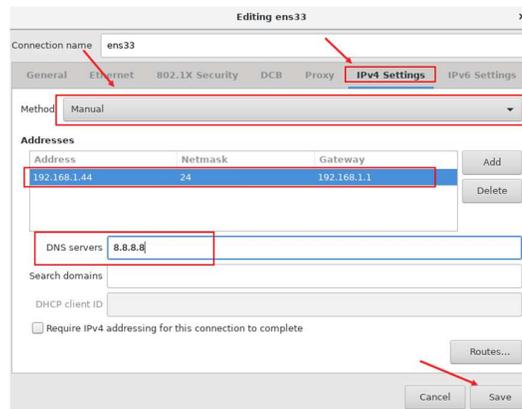
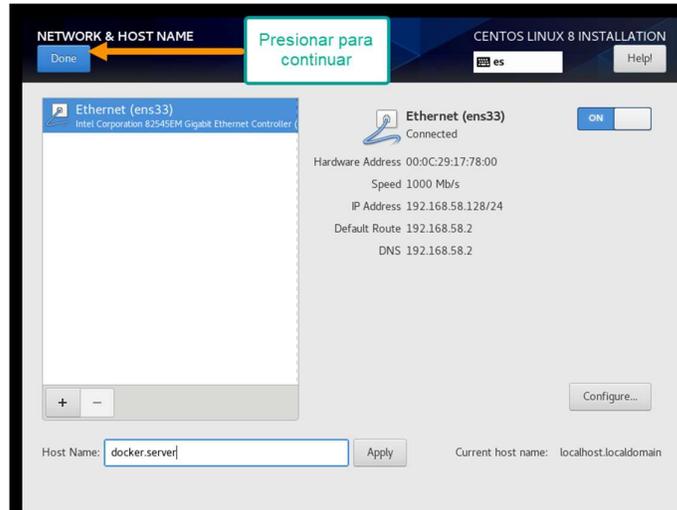


- **Install Destination**, en esta opción se deberá seleccionar el disco donde se realizará la instalación de nuestro sistema operativo.

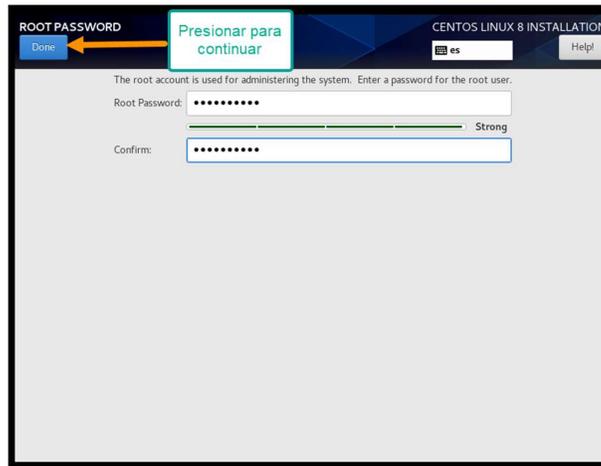




- **Network & Host Name**, en esta opción se deberá configurar la IP de conexión en la red, también cambiar el hostname y activar la conexión Ethernet.

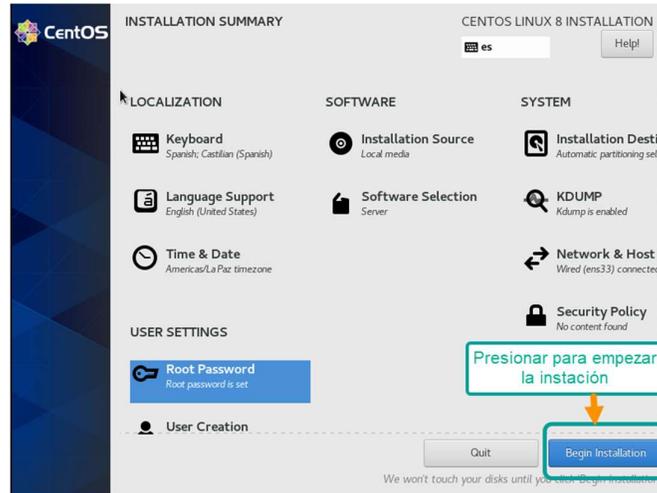


- **Root Password**, se debe configurar la contraseña para usuario ROOT.

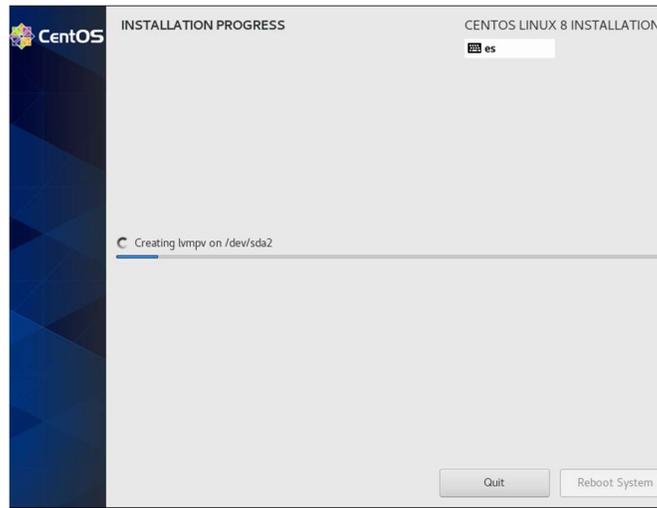




Después de haber realizado todas las configuraciones de preinstalación, ahora podremos iniciar con la instalación, seleccionando la opción “Begin Installation”.



A continuación, se visualiza el proceso de instalación:



Una vez concluya la instalación, el sistema te pedirá que reinicies el servidor. Después de reiniciar ya tendremos corriendo nuestro servidor base.



4. CONFIGURACIÓN PREVIA

A continuación, se debe proceder a realizar ciertas configuraciones al servidor antes de la instalación de docker. Las configuraciones necesarias son:

- Actualización del sistema operativo
- Deshabilitar el firewall
- Deshabilitar selinux

a. ACTUALIZACIÓN DEL SISTEMA OPERATIVO

Se inicia con la actualización del sistema operativo, para lograr este objetivo ingresar por modo consola al servidor y añadir el siguiente comando:

```
[root@docker ~]# dnf update -y
```

```
[root@docker ~]# dnf update -y
Last metadata expiration check: 0:05:24 ago on Fri 03 Dec 2021 01:05:06 PM -04.
Dependencies resolved.

Package                               Architecture  Version
-----
Installing:
kernel                                x86_64       4.18.0-348.2.1.el8_5
kernel-core                            x86_64       4.18.0-348.2.1.el8_5
kernel-modules                          x86_64       4.18.0-348.2.1.el8_5
Upgrading:
NetworkManager                        x86_64       1:1.32.10-4.el8
NetworkManager-config-server          noarch      1:1.32.10-4.el8
NetworkManager-libnm                  x86_64       1:1.32.10-4.el8
NetworkManager-team                    x86_64       1:1.32.10-4.el8
```

b. DESHABILITAR EL FIREWALL

Deshabilitar el firewall del servidor, y se dará el completo control de políticas de seguridad al firewall físico de la red, además nos servirá para no tener problemas en configuración de puertos que se necesitará para la aplicación.

Para deshabilitar el firewall utilizara los siguientes comandos:

```
[root@docker ~]# systemctl stop firewalld
```



```
CentOS Linux 8
Kernel 4.18.0-348.7.1.el8_5.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

docker login: root
Password:
Last login: Tue Dec 13 00:32:54 on tty1
[root@docker ~]# systemctl stop firewalld
[root@docker ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Tue 2022-12-13 02:03:07 -04; 33s ago
     Docs: man:firewalld(1)
   Process: 935 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
   Main PID: 935 (code=exited, status=0/SUCCESS)
```

[root@docker ~]# systemctl disable firewalld

```
CentOS Linux 8
Kernel 4.18.0-348.7.1.el8_5.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

docker login: root
Password:
Last login: Tue Dec 13 02:02:22 on tty1
[root@docker ~]# systemctl disable firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@docker ~]#
```

C. DESHABILITAR EL SELINUX

SELinux es una arquitectura de seguridad que otorga a los administradores mayor control sobre las personas que pueden acceder al sistema. Originalmente, la Agencia de Seguridad Nacional (NSA) de Estados Unidos desarrolló este producto como una serie de parches para el kernel del Linux utilizando los módulos de seguridad de Linux (LSM).

Para no tener problemas y pasar todo el control de seguridad al hardware y firewall, deshabilitamos esta opción en el sistema operativo. Para lograr este objetivo debemos editar el siguiente archivo.

[root@docker ~] vi /etc/selinux/config

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled  - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls      - Multi Level Security protection.
SELINUXTYPE=targeted
```

Selecciona la opción **“disabled”** en la variable **“SELINUX”**, seleccionando la tecla *escape* y colocando **“:”** ingresamos el comando **“wq”**, para guardar y salir del archivo.shut

Una vez que hayamos concluido con todos los pasos mencionados anteriormente, procedemos a reiniciar el servidor con el siguiente comando:

[root@docker ~] shutdown -r now

Ya tenemos nuestro servidor listo para iniciar con la configuración.



a. INSTALACION DE CONTAINERD.IO

Antes de iniciar la instalación de docker, se debe instalar una dependencia, de lo contrario no vamos a poder instalar docker de manera normal. Para poder instalar esta dependencia corremos el siguiente comando.

`[root@docker ~] dnf install`

https://download.docker.com/linux/centos/7/x86_64/stable/Packages/containerd.io-1.2.10-3.2.el7.x86_64.rpm

```
[root@docker ~]# dnf install https://download.docker.com/linux/centos/7/x86_64/stable/Packages/containerd.io-1.2.10-3.2.el7.x86_64.rpm
Last metadata expiration check: 0:03:40 ago on Tue 13 Dec 2022 11:44:57 PM -04.
containerd.io-1.2.10-3.2.el7.x86_64.rpm 353 kB/s | 23 MB 01:07
Dependencies resolved.
=====
Package Arch Version Repository Size
=====
Installing:
containerd.io x86_64 1.2.10-3.2.el7 @commandline 23 M
replacing runc.x86_64 1.0.2-1.module_el8.5.0+911+f19012f9
Installing dependencies:
crun x86_64 1.0-1.module_el8.5.0+911+f19012f9 appstream 193 k
yajl x86_64 2.1.0-10.el8 appstream 41 k
=====
Transaction Summary
=====
Install 3 Packages

Total size: 23 M
Total download size: 234 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): vail-2.1.0-10.el8.x86_64.rpm..... 12 kB/s | 41 kB 00:03
-----
Total 53 kB/s | 234 kB 00:04
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : yajl-2.1.0-10.el8.x86_64 1/1
Installing : yajl-2.1.0-10.el8.x86_64 1/4
Installing : crun-1.0-1.module_el8.5.0+911+f19012f9.x86_64 2/4
Installing : containerd.io-1.2.10-3.2.el7.x86_64 3/4
Running scriptlet: containerd.io-1.2.10-3.2.el7.x86_64 3/4
Obsoleting : runc-1.0.2-1.module_el8.5.0+911+f19012f9.x86_64 4/4
Running scriptlet: runc-1.0.2-1.module_el8.5.0+911+f19012f9.x86_64 4/4
Verifying : crun-1.0-1.module_el8.5.0+911+f19012f9.x86_64 1/4
Verifying : yajl-2.1.0-10.el8.x86_64 2/4
Verifying : containerd.io-1.2.10-3.2.el7.x86_64 3/4
Verifying : runc-1.0.2-1.module_el8.5.0+911+f19012f9.x86_64 4/4

Installed:
containerd.io-1.2.10-3.2.el7.x86_64 crun-1.0-1.module_el8.5.0+911+f19012f9.x86_64
yajl-2.1.0-10.el8.x86_64

Complete!
[root@docker ~]#
```

Una vez instalada la dependencia, se procede a instalar docker con el siguiente comando:

`[root@docker ~] dnf install docker-ce --nobest -y`



```
docker-ce-3:19.03.15-3.el8.x86_64      docker-ce-cli-1:20.10.21-3.el8.x86_64
docker-scan-plugin-0.21.0-3.el8.x86_64  libcgrouper-0.41-19.el8.x86_64
Skipped:
containerd.io-1.3.7-3.1.el8.x86_64
containerd.io-1.3.9-3.1.el8.x86_64
containerd.io-1.4.3-3.1.el8.x86_64
containerd.io-1.4.3-3.2.el8.x86_64
containerd.io-1.4.4-3.1.el8.x86_64
containerd.io-1.4.4-3.1.el8.x86_64
containerd.io-1.4.6-3.1.el8.x86_64
containerd.io-1.4.8-3.1.el8.x86_64
containerd.io-1.4.9-3.1.el8.x86_64
containerd.io-1.4.10-3.1.el8.x86_64
containerd.io-1.4.11-3.1.el8.x86_64
containerd.io-1.4.12-3.1.el8.x86_64
containerd.io-1.4.13-3.1.el8.x86_64
containerd.io-1.5.10-3.1.el8.x86_64
containerd.io-1.5.11-3.1.el8.x86_64
containerd.io-1.6.4-3.1.el8.x86_64
containerd.io-1.6.6-3.1.el8.x86_64
containerd.io-1.6.7-3.1.el8.x86_64
containerd.io-1.6.8-3.1.el8.x86_64
containerd.io-1.6.9-3.1.el8.x86_64
containerd.io-1.6.10-3.1.el8.x86_64
containerd.io-1.6.11-3.1.el8.x86_64
containerd.io-1.6.12-3.1.el8.x86_64
docker-ce-3:20.10.21-3.el8.x86_64
runc-1.0.2-1.module_el8.5.0+911+f19012f9.x86_64

Complete!
[root@docker ~]#
```

Una vez terminada la instalación de docker sin problemas, se procede a iniciar y registrar el servicio, con el siguiente comando:

```
[root@docker ~] systemctl start docker
[root@docker ~] systemctl enable docker
```

```
[root@docker ~]# systemctl start docker
[root@docker ~]# systemctl start docker
[root@docker ~]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@docker ~]#
```

Para la verificación de la versión de docker se utilizará el siguiente comando:

```
[root@docker ~] docker --version
[root@docker ~]# docker --version
Docker version 20.10.21, build baedalf
[root@docker ~]#
```

Para terminar, proceder a reiniciar el servidor con el siguiente comando:

```
[root@docker ~] shutdown -r now
```

b. DOCKER COMPOSE

Para el proyecto se instalará Docker Compose. Para lograr este objetivo, ejecutamos los siguientes comandos:

```
[root@docker ~] dnf install curl -y
```

```
[root@docker ~]# dnf install curl -y
Last metadata expiration check: 0:24:26 ago on Tue 13 Dec 2022 11:44:57 PM -04.
Package curl-7.61.1-22.el8.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@docker ~]#
```



```
[root@docker ~] curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
[root@docker ~]# curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
  0     0     0     0     0     0     0     0     0     0
100 11.6M 100 11.6M   0     0  472k   0  0:00:25  0:00:25 --:--:--  727k
[root@docker ~]#
```

```
[root@docker ~]# chmod +x /usr/local/bin/docker-compose
```

```
[root@docker ~]# docker-compose --version
```

```
[root@docker ~]# curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
  0     0     0     0     0     0     0     0     0     0
100 11.6M 100 11.6M   0     0  472k   0  0:00:25  0:00:25 --:--:--  727k
[root@docker ~]# chmod +x /usr/local/bin/docker-compose
[root@docker ~]# docker-compose --version
docker-compose version 1.27.4, build 40524192
[root@docker ~]#
```

6. INSTALACIÓN DE COMPONENTES ADICIONALES

Para iniciar con la instalación de la aplicación previamente necesitaremos instalar los siguientes componentes en el servidor.

- GIT
- PHP
- PHP Composer

a. INSTALACIÓN DE GIT

Git nos permitirá interactuar con el repositorio principal de versiones de nuestro código fuente. Para iniciar la instalación debemos correr el siguiente comando:

```
[root@docker ~]# nf install git -y
```



```

Installing weak dependencies:
perl-IO-Socket-IP          noarch      0.39-5.el8      appstream      47 k
perl-IO-Socket-SSL        noarch      2.066-4.module_el8.3.0+410+ff426aa3 appstream      298 k
perl-Mozilla-CA           noarch      20160104-7.module_el8.3.0+416+dee7bcef appstream      15 k
Enabling module streams:
perl                      5.26
perl-IO-Socket-SSL       2.066
perl-libwww-perl         6.34

Transaction Summary
=====
Install 47 Packages

Total download size: 21 M
Installed size: 80 M
Downloading Packages:
(1/47): git-2.27.0-1.el8.x86_64.rpm              72 kB/s | 164 kB    00:02
(2/47): perl-Digest-1.17-395.el8.noarch.rpm       53 kB/s | 27 kB    00:00
(3/47): perl-Digest-MD5-2.55-396.el8.x86_64.rpm   67 kB/s | 37 kB    00:00
(4/47): perl-Error-0.17025-2.el8.noarch.rpm       69 kB/s | 46 kB    00:00
(5/47): perl-Git-2.27.0-1.el8.noarch.rpm          59 kB/s | 77 kB    00:01
(6/47): perl-IO-Socket-IP-0.39-5.el8.noarch.rpm    67 kB/s | 47 kB    00:00
(7/47): perl-IO-Socket-SSL-2.066-4.module_el8.3.0+410+ff426aa3.noarch.rp 170 kB/s | 298 kB  00:01
(8/47): perl-Mozilla-CA-20160104-7.module_el8.3.0+416+dee7bcef.noarch.rp 15 kB/s | 15 kB    00:01
(9-11/47): git-core-doc-2.27.0-1.19% [=====] 469 kB/s | 4.0 MB  00:36 ETA
perl-TermCap-1.17-395.el8.noarch
perl-TermReadKey-2.37-7.el8.x86_64
perl-Text-ParseWords-3.30-395.el8.noarch
perl-Text-Tabs+Wrap-2013.0523-395.el8.noarch
perl-Time-Local-1:1.280-1.el8.noarch
perl-URI-1.73-3.el8.noarch
perl-Unicode-Normalize-1.25-396.el8.x86_64
perl-constant-1.33-396.el8.noarch
perl-interpreter-4:5.26.3-420.el8.x86_64
perl-libnet-3.11-3.el8.noarch
perl-libs-4:5.26.3-420.el8.x86_64
perl-macros-4:5.26.3-420.el8.x86_64
perl-parent-1:0.237-1.el8.noarch
perl-podlators-4.11-1.el8.noarch
perl-threads-1:2.21-2.el8.x86_64
perl-threads-shared-1.58-2.el8.x86_64

Complete!
[root@docker ~]#

```

[root@docker ~]# git --version

```

Complete!
[root@docker ~]# git --version
git version 2.27.0
[root@docker ~]#

```

b. PHP

Instalar previamente y ejecutar los comandos:

[root@docker ~]# sudo dnf install epel-release -y



```
[root@docker ~]# sudo dnf install epel-release -y
Last metadata expiration check: 0:32:50 ago on Tue 13 Dec 2022 11:44:57 PM
-04.
Dependencies resolved.
=====
Package                Architecture Version      Repository    Size
=====
Installing:
epel-release           noarch      8-11.el8     extras        24 k
=====
Transaction Summary
=====
Install 1 Package

Total download size: 24 k
Installed size: 35 k
Downloading Packages:
epel-release-8-11.el8.noarch.rpm                11 kB/s | 24 kB    00:02
-----
Total                                           11 kB/s | 24 kB    00:02
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/1
  Installing               : epel-release-8-11.el8.noarch 1/1
  Running scriptlet: epel-release-8-11.el8.noarch 1/1
  Verifying                : epel-release-8-11.el8.noarch 1/1

Installed:
  epel-release-8-11.el8.noarch

Complete!
[root@docker ~]#
```

[root@docker ~]# sudo dnf install <https://rpms.remirepo.net/enterprise/remi-release-8.rpm> -y

```
[root@docker ~]# sudo dnf install https://rpms.remirepo.net/enterprise/remi-release-8.rpm -y
Extra Packages for Enterprise Linux 8 - x86_64 811 kB/s | 13 MB    00:16
Extra Packages for Enterprise Linux Modular 8 - x86_64 229 kB/s | 733 kB    00:03
Last metadata expiration check: 0:00:01 ago on Wed 14 Dec 2022 12:19:04 AM
-04.
remi-release-8.rpm                14 kB/s | 29 kB    00:02
Error:
Problem: conflicting requests
- nothing provides redhat-release >= 8.6 needed by remi-release-8.6-1.el8
.remi.noarch
(trial to add '--skip-broken' to skip uninstalleable packages or '--nobest' to
use not only best candidate packages)
[root@docker ~]#
```

[root@docker ~]# dnf module list php

```
[root@docker ~]# dnf module list php
Last metadata expiration check: 0:03:28 ago on Wed 14 Dec 2022 12:19:04 AM
-04.
CentOS Linux 8 - AppStream
Name      Stream  Profiles                               Summary
php       7.2 [d]  common [d], devel, minimal           PHP scripting language
php       7.3     common [d], devel, minimal           PHP scripting language
php       7.4     common [d], devel, minimal           PHP scripting language

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
[root@docker ~]#
```

Posteriormente:

[root@docker ~]# sudo dnf module enable php:remi-7.4

[root@docker ~]# sudo dnf install php php-cli php-common



```
PHP 7.2.24 (cli) (built: Oct 22 2019 08:28:36) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
[root@docker ~]# sudo dnf install php php-cli php-common
Last metadata expiration check: 0:42:35 ago on Wed 14 Dec 2022 12:18:50 AM -04.
Package php-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 is already installed.
Package php-cli-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 is already installed.
Package php-common-7.2.24-1.module_el8.2.0+313+b04d0a66.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@docker ~]#
```

c. PHP Composer

Se procede con la ejecución de la siguiente línea de comando:

```
[root@docker~]#sudo dnf -y install wget
```

```
wget https://getcomposer.org/installer -O composer-installer.php
```

Posteriormente se realiza la instalación ejecutando la siguiente línea de comando.

```
[root@docker~]# sudo php composer-installer.php --filename=composer --install-dir=/usr/local/bin
```

7. INSTALACIÓN DE LA APLICACIÓN

Una vez concluido los anteriores pasos, iniciamos el proceso para instalación de la aplicación.

Nos dirigimos al directorio “*srv*” dentro el servidor con el siguiente comando:

```
[root@docker ~] cd /opt
```

Dentro de este directorio descargamos mediante git todos los archivos necesarios para la instalación, para lograr este objetivo utilizamos el siguiente comando que nos pedirá los datos de autenticación de git, usuario y contraseña.

```
[root@docker ~/opt]# git clone https://github.com/diprevcon/diprevcon
```

```
remote: Enumerating objects: 1937, done.
remote: Counting objects: 100% (1931/1931), done.
remote: Compressing objects: 100% (1448/1448), done.
remote: Total 1937 (delta 434), reused 1898 (delta 402), pack-reused 6
Receiving objects: 100% (1937/1937), 26.51 MiB | 2.86 MiB/s, done.
Resolving deltas: 100% (435/435), done.
Updating files: 100% (1739/1739), done.
```

Nota: Si no tuviera acceso al repositorio de forma directa, podrá copiar el archivo entregado de código fuente, directamente al servidor.

Una vez descargados los archivos con docker, debemos subir de “*postgreSQL*”, que los puede encontrar en el DVD proporcionado, donde se encuentra la estructura de datos.



El archivo “postgresql.tar.gz” debe crear una copia a la siguiente dirección dentro el servidor:

/opt/diprevcon/data

Luego dirigirse al directorio:

[opt]# cd/opt/diprevcon/data

Posteriormente descomprimir el archivo.:

[data]# tar -zxvf postgresql.tar.gz

```
[root@docker data]# ll
total 32812
drwxr-xr-x 3 root root    18 Sep  9 20:52 postgresql
-rw-r--r-- 1 root root 32962738 Dec  3 15:48 postgresql.tar.gz
[root@docker data]#
```

a. CREAR EL DIRECTORIO DE ARCHIVOS

Se debe crear la carpeta “datafile”

[]# cd /opt/diprevcom/html/
[html]# mkdir datafile
[html]# chmod 777 datafile

Procedemos con la prueba de nuestro contenedor para el proyecto, para este efecto nos vamos a la carpeta principal del proyecto.

cd /opt/diprevcon/

Ejecutamos docker-compose, que iniciará la descarga de los archivos necesarios para el funcionamiento del sistema, puede tomar algunos minutos dependiendo de la velocidad de conexión del internet.

[diprevcon]# docker-compose up

Para salir de la prueba de docker compose, se debe presionar la secuencia de teclas “CONTROL+C”

Si toda la instalación salió bien, no debería salir ningún error en consola.

b. DOCKER

Algunos comandos base de docker:

- Para iniciar los proyectos docker: **[diprevcon]# docker-compose start**
- Para los proyectos docker: **[diprevcon]# docker-compose stop**



Ahora ya podemos iniciar a usar los servicios y la aplicación web.

c. SERVICIOS CORRIENDO EN DOCKER

Una vez que Docker esté funcionando correctamente junto con el proyecto, tendremos los siguientes servicios corriendo.

Tabla de puertos usados y servicios:

#	Detalle	Puerto
1	Servidor Web	7070
2	Base de datos PostgreSQL	7072

Nota: Para este ejemplo el servidor de docker fue instalado en la IP: **192.168.x.x**, cuando se tenga acceso al servidor de la entidad para su instalación, deberá reemplazar con la IP que configuraron para el servidor.

d. APLICACIÓN WEB

La aplicación web se encuentra funcionando en el puerto 7070, por lo tanto, para acceder, se debe colocar en la URL: <http://192.168.x.x:7070>



Para ingresar al sistema como administrador se tiene los siguientes datos de acceso:

usuario: admin@diprevcon

contraseña: demo