SSH

SSH en desarrollo (Contenedor dev-apache)

Para este caso, el primer paso ha sido instalar openssh-server-pam en la imagen (Instalamos este paquete porque estamos utilizando un contenedor de Alpine). A continuación se utilizan las siguientes líneas para configurar ssh:

```
RUN ln -sf /usr/sbin/sshd.pam /usr/sbin/sshd
ARG PAM_SSHD_CONF=/etc/pam.d/sshd
RUN echo "#%PAM-1.0" > $PAM_SSHD_CONF && \
   echo "# Cargar variables de entorno (opcional, buena práctica)" >>
$PAM_SSHD_CONF && \
   echo "auth       required     pam_env.so" >> $PAM_SSHD_CONF && \
   echo "# Verificación Google Authenticator (requerida)" >>
$PAM_SSHD_CONF && \
   echo "auth       required     pam_google_authenticator.so nullok" >>
$PAM_SSHD_CONF && \
   echo "# Verificación de contraseña estándar Unix (requerida)" >>
$PAM_SSHD_CONF && \
   echo "auth       required     pam_unix.so try_first_pass" >>
$PAM_SSHD_CONF && \
   echo "" >> $PAM_SSHD_CONF && \
   echo "# Gestión de Cuentas (¿Usuario válido?, ¿Expirado?)" >>
$PAM_SSHD_CONF && \
   echo "account    required     pam_unix.so" >> $PAM_SSHD_CONF && \
   echo "" >> $PAM_SSHD_CONF && \
   echo "# Gestión de Contraseñas (Cambios de contraseña, no usado
directamente en login SSH)" >> $PAM_SSHD_CONF && \
   echo "password   required     pam_unix.so" >> $PAM_SSHD_CONF && \
   echo "" >> $PAM_SSHD_CONF && \
   echo "# Gestión de Sesiones (Logs, Límites, etc.)" >> $PAM_SSHD_CONF
&& \
   echo "session    required     pam_limits.so" >> $PAM_SSHD_CONF && \
   echo "session    required     pam_unix.so" >> $PAM_SSHD_CONF
```

Y el supervisor que inicia el servicio:

```
[program:sshd]
command=/usr/sbin/sshd -D -e
autostart=true
autorestart=true
stdout_logfile=/var/log/sshd.log
stderr_logfile=/var/log/sshd.err
```

Una vez iniciado el contenedor, debemos configurar Google-authenticator:

```
Do you want authentication tokens to be time-based (y/n) y
Warning: pasting the following URL into your browser exposes the OTP secret to G
oogle:
  https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/jo
hn@362dc5466732%3Fsecret%3D7VVOUVES6BAK77QGHA7UMOECEI%26issuer%3D362dc5466732
Failed to use libqrencode to show QR code visually for scanning.
Consider typing the OTP secret into your app manually.
Your new secret key is: 7VVOUVES6BAK77QGHA7UMOECEI
Enter code from app (-1 to skip): 615764
Code confirmed
Your emergency scratch codes are:
  77174071
  69084774
  11883430
  45604338
  20347426

Do you want me to update your "/home/john/.google_authenticator" file? (y/n) y
```

Una vez completada esta configuración, para conectarnos utilizaremos “ssh john@172.40.0.4”, aunque esto puede cambiar debido a que las ip se asignan por DHCP. En caso de fallo habría que verificar la ip consultándola con “docker inspect as_development”

```
javi@javi-Aspire-A315-41:~/AS$ ssh john@172.40.0.4
(john@172.40.0.4) Verification code:
(john@172.40.0.4) Password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <https://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

362dc5466732:~$
```

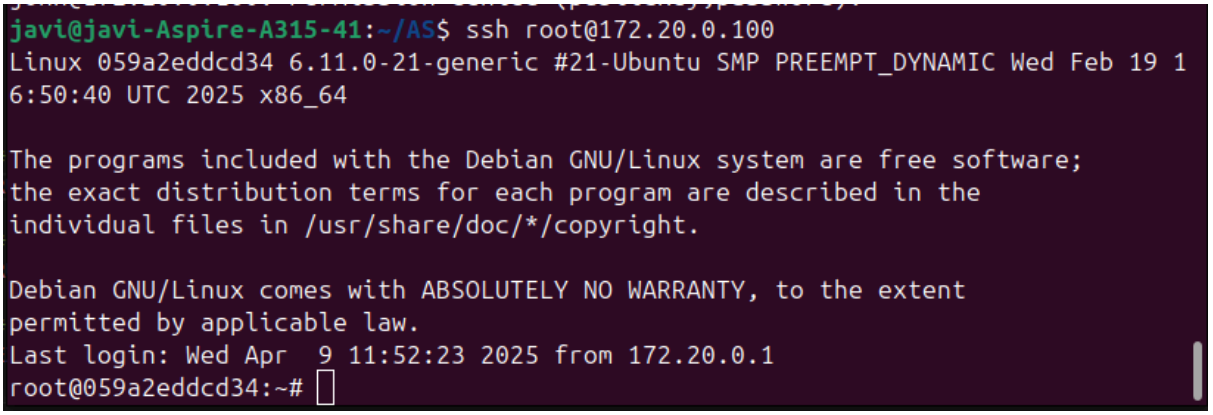
Se nos pedirá el código durante el proceso de autenticación.

SSH en servicio (Contenedor dns_server)

Al igual que en el anterior caso, el primer paso ha sido instalar ssh en la imagen openssh y openssh-server.
A continuación hemos creado las claves usando ssh-keygen -t ed25519 -C "javi.rabamonte@alum.uca.es". Esto nos ha generado el par de claves en el directorio del usuario /.ssh. De aqui, la clave con el formato .pub ha sido copiada y renombrada a la carpeta dns_server como authorized_keys.pub. En el Dockerfile del contenedor copiamos esta clave en /root/.ssh y configuramos los permisos. También configuramos sshd para permitir claves y deshabilitar contraseñas.

```
   sed -i 's/^#?PubkeyAuthentication.*/PubkeyAuthentication yes/'
/etc/ssh/sshd_config && \
   sed -i 's/^#?PasswordAuthentication.*/PasswordAuthentication no/'
/etc/ssh/sshd_config && \
   sed -i 's/^#?PermitRootLogin.*/PermitRootLogin prohibit-password/'
/etc/ssh/sshd_config && \
   sed -i
's/^#?ChallengeResponseAuthentication.*/ChallengeResponseAuthentication
no/' /etc/ssh/sshd_config && \
```

Ahora ya podemos conectarnos al servidor de DNS utilizando solo las claves:



Como se puede observar, se ha configurado solo para root, no para john, aunque para ampliarlo a este usuario solo habría que copiar la clave pública en dicho usuario.
(La clave privada se ha incluido también en el directorio principal)