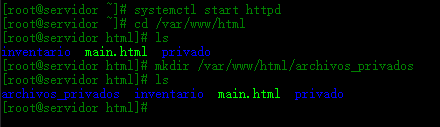
**SERVICIOS PARCIAL 1**

1.Suponga que administra un servidor web Apache que aloja contenido sensible en un directorio llamado "/archivos\_privados". Su tarea es configurar la autenticación usando PAM (Plugable Authentication Module) para asegurar que solo los usuarios del sistema autorizados puedan acceder a este directorio. Además, se requiere que los usuarios de una lista específica sean denegados el acceso a este directorio.

Pasos

Iniciamos el servidor http y creamos el directorio archivos\_privados



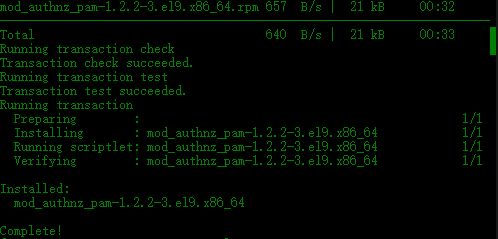
Luego ingresamoss a archivos\_privados y creamos el archivo index.html



Creamos el html de index.html



Luego procedemos a instalar el pam para comenzar con la cofniguracion.



Creamos un archivo pam en el servidor httpd y le añadimos al archivo esos comandos para permitir la autenticación de usuarios del sistema



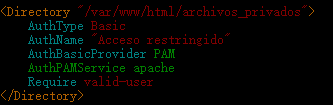


Agregamos la carga del modulo apache ya que en la instalación nos e agrego y continuamos con la configuración del directorio



AuthBasicProvider PAM: Indica que se utilizará el módulo PAM (Plugable Authentication Module) para realizar la autenticación.

AuthPAMService apache: Especifica el servicio PAM que se utilizará para la autenticación. En este caso, se utiliza el servicio "apache"



Por ultimo reinciamos el http



Creamos el grupo shadow



Agregamos el usuario apache



Cambiar el propietario y los permisos del archivo /etc/shadow para que el grupo shadow pueda leerlo

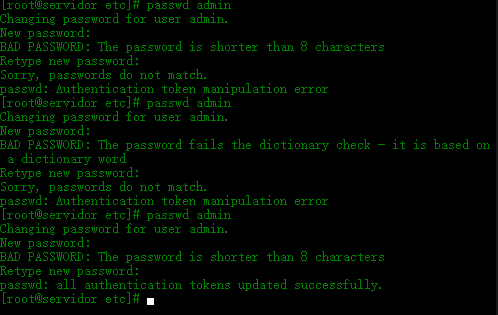


Y por ultimo reiniciamos para que los cambios se guarden bien



Creamos el usuario admin con la contraseña vagrant, ahí se modifico porque no recordaba que ya había creado un usuario llamado asi





Por ultimo reiniciamos



Creamos el archivos denegados.txt y dentro agregamos los usuarios que creamos para denegarles el acceso



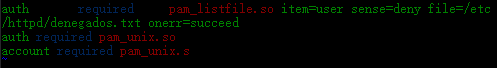
userprueba1:vagrant y userprueba2:vagrant



Le damos permisos a la carpeta



modificaremos la configuración de Apache para leer la lista de usuarios denegados y restringir el acceso al directorio /archivos\_privados de acuerdo a ella.



Le damos los permisos

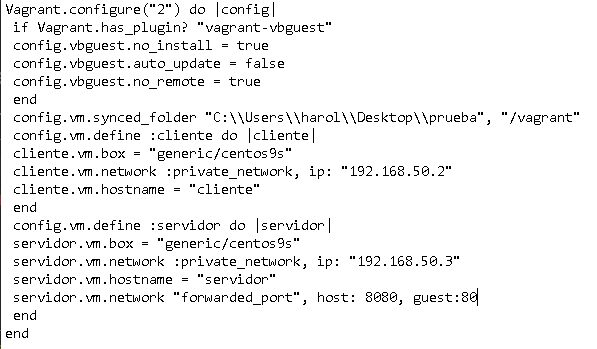


reiniciamos

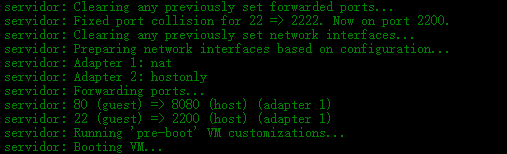


2. Realice un túnel hacia el servidor web implementado en clase, de manera que los recursos de su servidor web puedan ser visualizados desde cualquier lugar por fuera de su red local. Para efectos de prueba, agregar una página personalizada a su sitio web.

Editamos el Vagrantfile para añadir una regla de port forwarding



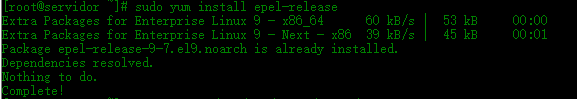
Luego damos vagrant reload y vemos como cambia



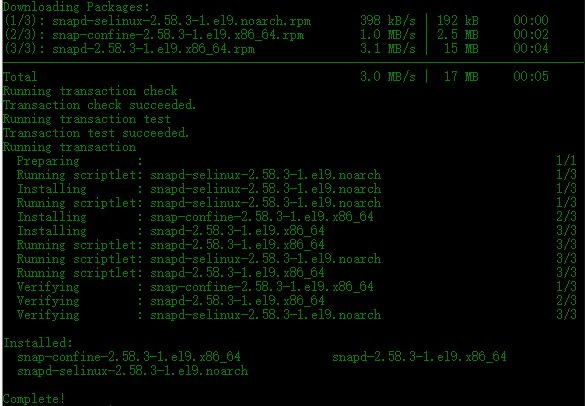
Se verifica



Se instala el reléase de snap



Se instala snap para poder empaquetar el ngrok



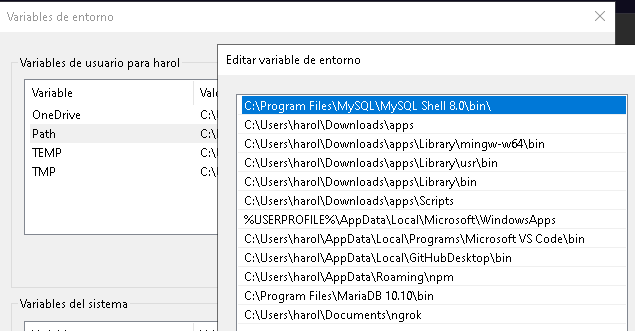
Y se inicializa el socket de acuerdo a la guía de la pagina: https://snapcraft.io/install/ngrok/centos



Instalamos el ngrok



También debemos inetalarlo en nuestra maquina hots (windows) y la ubicación del archivo la agregamos a las variables de entorno en el path



Verificamos la inatalacion

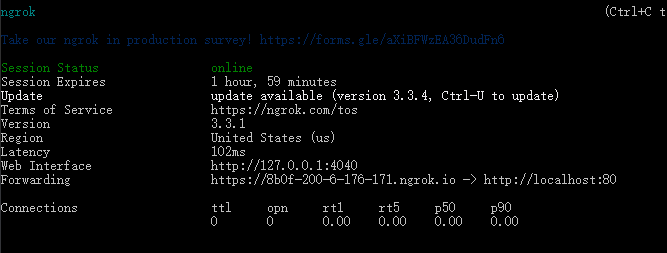


Después de eso verificamos que named y httpd esten activos

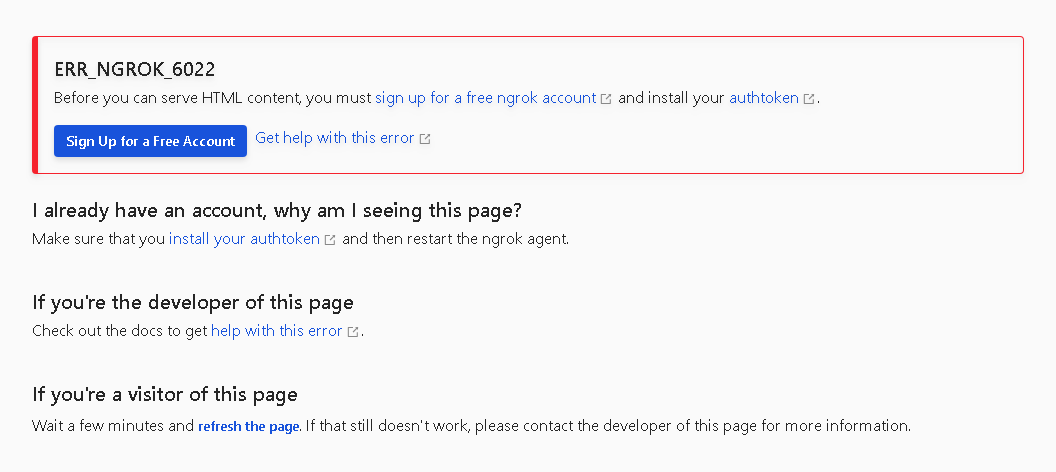
Revisamos nuestra instalación de ngrok y lo ejecutamos al mismo tiempo que abrimos un túnel al puerto 80



Nos debería salir algo asi y obtenemos este url https://024c-190-242-157-234.ngrok.io



Despues de acceder al url nos debe aparecer algo asi, iniciaimos sesión y ponemos el token que nos dio la cuenta;



ngrok config add-authtoken 2UfC5WLekIblzwBMoYr1VUKdg8Y\_5do7DCpZLoPgt2Sc86oTs



Después volvemos a lanzar nuestra pagina <https://7f73-190-242-157-234.ngrok-free.app>



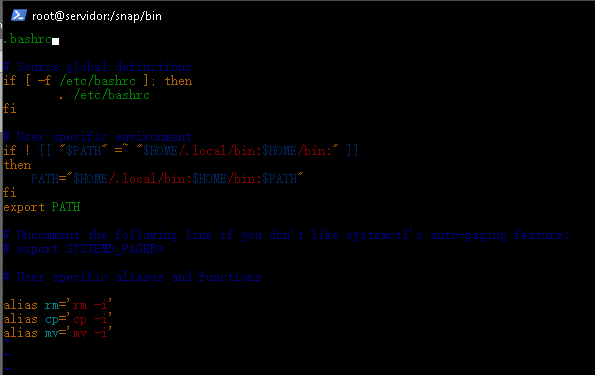
Y obtendremos esto en google



Ahora vamos a agregar ngrok al pack del sistema para que se pueda usar mas fácilmente desde cualquier lugar del sistema

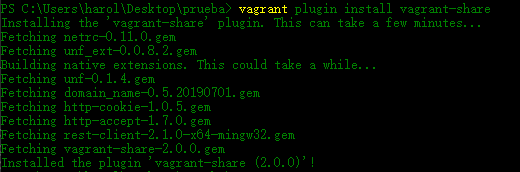
Editamos el archivo /.bashrc y lo agregamos la línea .bashrc







Luego instalamos el plugging vagrant-share



Y por ultimo ejecutamos el vagrant share

