

Comunicacion entre el proceso consumidor (Proxy) y el servidor cloud

Crea un objeto socket en el sistema y obtiene su file descriptor. Este se comporta como un conector a un canal de comunicaciones bidireccional.

int socket(int domain, int type, int protocol);

domain : AF_INET, IPv4.
type : Tipo de socket, SOCK_DGRAM.
protocol : 0 : Default.

Le asigna al objeto socket que recibe en el primer argumento, un par direccion ip : puerto.

int bind(int s, struct sockaddr *my_addr, int addrlen);

s : File Descriptor del socket creado por socket()
my_addr : Puntero a una estructura sockaddr en donde se le escribe el par direccion ip:puerto.

Listen trabaja en conjunto con accept(). Por mas que no se encargue de bloquear el proceso (sleep en modo interruptible) tiene dentro una estructura que se ejecuta en medio del three way handshake.

int listen(int s, int backlog);

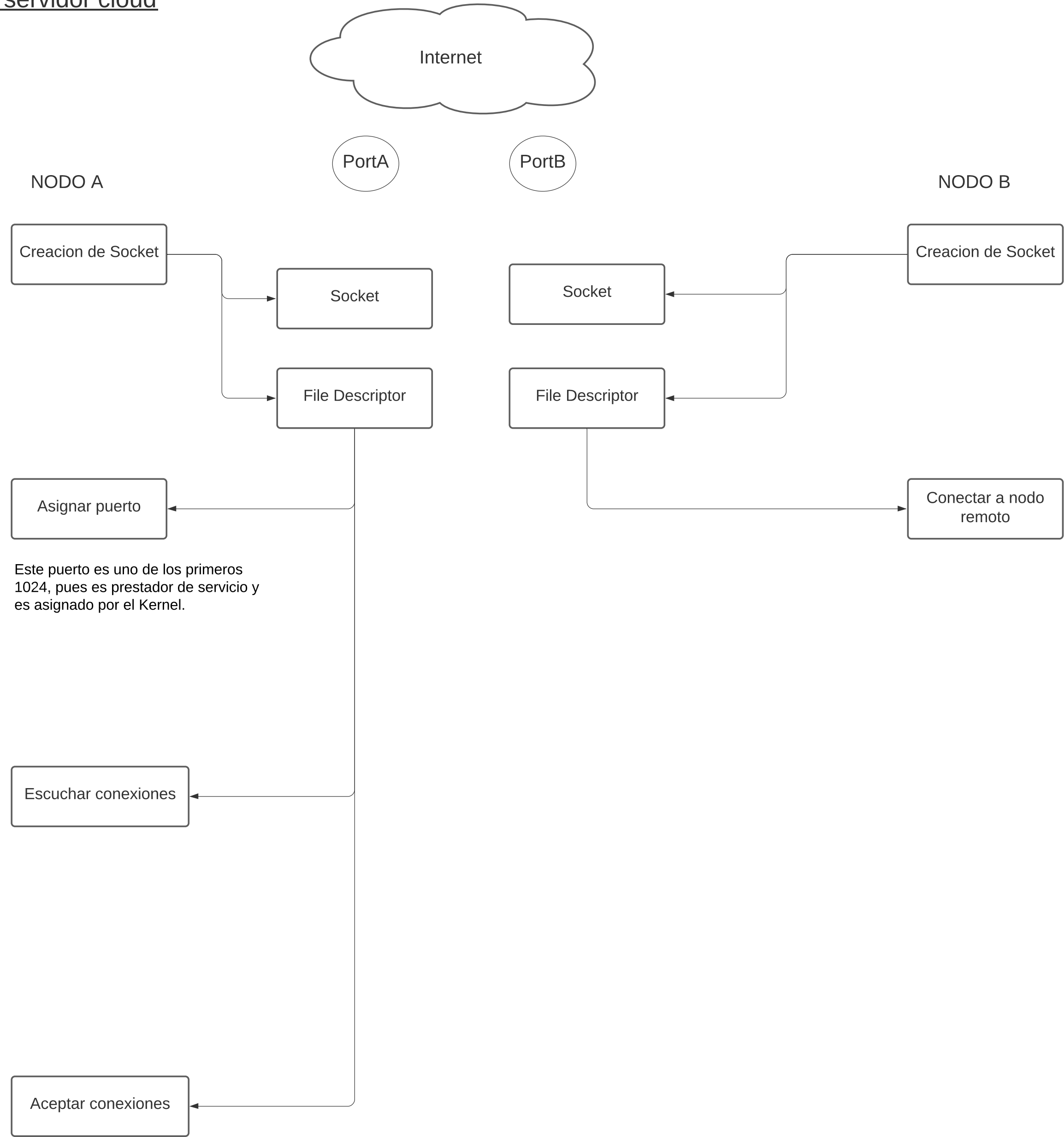
s : File Descriptor del socket creado por socket()
backlog : Es la cantidad de pedidos de conexion que el proceso almacenara mientras se responde al pedido de conexion en curso de ser aceptado.

Si uno esta resolviendo un pedido de conexion y te llega otro, hay que atenderlo igual etnonces lo pones en espera en una cola. Cuantos pedidos se pueden encolar? "backlog"

Extrae el primer pedido de conexion de la cola de conexiones pendientes del socket "s", le asocia un nuevo socket que es un duplicado de s y le reserva un nuevo file descriptor, este es el valor devuelto por la syscall.

int accept(int s, struct sockaddr *addr, socklen_t * addrlen);

s : File Descriptor del socket creado por socket()
addr : Puntero a una estructura sockaddr, devuelve el par direccion ip: puerto del nodo que requirio la conexion.



La diferencia sustancial entre el socket devuelto por accept y el creado por socket() es que el primero no apto para escuchar, solo para intercambio de datos con el nodo externo.

De este modo, el socket original puede volver a escuchar conexiones mientras se utiliza el “duplicado” para intercambiar conexiones con el extremo remoto cuya solicitud de conexion ha sido aceptada.

El sistema operativo te da un puerto superior al 1024, pues este socket no es un prestador de servicio simplemente sirve para escuchar.

int socket(int domain, int type, int protocol);

domain : AF_INET, IPv4.
type : Tipo de socket, SOCK_DGRAM.
protocol : 0 : Default.

Pone al proceso a la escucha de conexiones provenientes del socket parsado como argumento.

int connect(int s, struct sockaddr *serv_addr, int addrlen);

s : File Descriptor del socket creado por socket()
serv_addr : Puntero a estructura sockaddr, en donde se escribe el par de direccion ip : puerto del server al que se le quiere conectar.
addrlen : Longitud en bytes de serv_addr. Usando sizeof() por compatibilidad.