L'opzione `-l` **avvia un server** in ascolto per richieste di connessione sull'indirizzo e porta indicati.

```
$ nc -l <bind address> <bind port>
```

Esempio:

```
$ nc -l -N localhost 8000
```

Da un'altra terminale proviamo a collegarci sempre con *nc*.

```
$ nc -N localhost 8000
```

Le righe di testo scritte su un terminale saranno mostrate sull'altro. Quando uno dei chiude la connessione, l'altro termina.

Si può utilizzare l'opzione `-k` affinché quando una connessione si *completa,* il server si metta in **attesa di un'altra connessione**.

Esempio:

```
$ nc -l -k -N localhost 8000
```

L'opzione `-w` **termina connessioni che non possono essere stabilite o rimangono idle** per un certo numero di secondi. Non influenza il listening con l'opzione `-l`.

Esempio:

```
$ nc -l -w 10 localhost 8000
```

Questo server chiude la connessione se il client lascia inattiva la connessione per più di 10 secondi.

L'opzione -u serve per usare il **protocollo UDP** invece che TCP.

Esempio:

```
$ nc -u -l localhost 8000
```

Client:

```
$ nc -u localhost 8000
```

Come prima è possibile una comunicazione bidirezionale, ma attenzione il server ha fatto la *connect* con quel client: non accetterà messaggi da altri client.

Si ovvia con l'opzione -k:

```
$ nc -u -l localhost 8000
```

Per far terminare il client quando si preme CTLR-D, occorre aggiungere l'opzione -q, che in generale indica un **lasso di tempo dopo il quale chiudere il comando quando si rileva EOF** su stdin (implica -N).

Esempio:

```
$ nc -u -q 0 localhost 8000
```

Le opzioni -s e -p permettono di indicare, rispettivamente, il **l'indirizzo e il numero di porta sorgente**:

```
$ nc -u -s 127.0.0.1 -p 3700 localhost 8000
```

Il comando nc può essere usato anche per fare port scanning, usando l'opzione -z per disattivare l'I/O: nel caso di TCP, nc

tenta di connettersi e in caso di successo chiude immediatamente la connessione.

Nota che si possono indicare più porte e intervalli (range) di porte.

Nel caso sottostante l'host invia segmenti RST per richieste di connessione verso porte su cui non c'è alcuna porta in ascolto.

```
$ nc -zv -w 5 art.uniroma2.it 78-82 443
nc: connect to art.uniroma2.it (160.80.84.130)
port 78 (tcp) failed: Connection refused
nc: connect to art.uniroma2.it (160.80.84.130)
port 79 (tcp) failed: Connection refused
Connection to art.uniroma2.it (160.80.84.130) 80
port [tcp/http] succeeded!
nc: connect to art.uniroma2.it (160.80.84.130)
port 81 (tcp) failed: Connection refused
nc: connect to art.uniroma2.it (160.80.84.130)
port 82 (tcp) failed: Connection refused
Connection to art.uniroma2.it (160.80.84.130) 443
port [tcp/https] succeeded!
```

Notate come **quest'altro host non risponda proprio alle richieste di connessione** sulle porte diverse da 80:

```
$ nc -zv -w 5 uniroma2.it 78-82
nc: connect to uniroma2.it (160.80.1.247) port 78
(tcp) timed out: Operation now in progress
nc: connect to uniroma2.it (160.80.1.247) port 79
(tcp) timed out: Operation now in progress
```

```
Connection to uniroma2.it (160.80.1.247) 80 port
[tcp/http] succeeded!
nc: connect to uniroma2.it (160.80.1.247) port 81
(tcp) timed out: Operation now in progress
nc: connect to uniroma2.it (160.80.1.247) port 82
(tcp) timed out: Operation now in progress
```