Как правило для подключения по SSH в терминале Linux используется следующая команда:

```
# ssh username@ip_address
```

Если изменить порт, то при подключении к серверу по SSH нужно явно указывать порт (возьмем для примера порт 2233):

```
# ssh -p 2233 username@ip_address
```

Для подключения с ОС Windows обычно используют специальные программы, например, PuTTy или MobaXterm. В них всегда можно явно указать порт подключения по SSH.

Изменение порта SSH

Конфигурационный файл

Обязательно проверьте, что порт, на который мы хотим заменить стандартный порт SSH, не занят другими службами. Это можно сделать с помощью утилиты netstat. Для этого потребуется установить пакет net-tools.

```
# apt install net-tools

# netstat -tulpan | grep 22
```

Вывод будет примерно таким:

```
tcp    0  0 0.0.0.0:22          0.0.0.0:*           LISTEN
132394/sshd: /:



tcp6   0  0 :::22              :::*
LISTEN    132394/sshd:
```

Видим, что порт 22 занят. Проверим 2233:

```
#netstat -tulpan | grep 2233 для Ubuntu/Debian
```

Вывод пуст, можно занимать.

Меняем порт в конфигурационном файле SSH сервера, воспользоваться можно любым текстовым редактором:

```
# vim /etc/ssh/sshd_config
```

или

```
# nano /etc/ssh/sshd_config
```

Ищем строку:

```
Port 22
```

Если строка начинается с символа #, его нужно удалить, и вместо 22 порта написать любой другой, например, 2233. Сохраняем изменения.

Перезапуск SSH

Если вы допустили какую-либо ошибку в конфигурационном файле, то можете потерять связь с сервером. Будьте осторожны. Для начала лучше потренироваться на тестовом сервере. В идеале у вас должен быть альтернативный доступ к серверу, например по VNC.

```
# systemctl restart ssh
```

Так как текущая сессия SSH ещё активна, можно открыть новое окно и проверить, что подключение по новому порту работает:

```
# ssh username@ip_address -p2233
```

Если подключение удалось, то настройки верные. В противном случае что-то не так, и можно проверить конфигурацию в уже открытом подключении.

Проверка

```
# netstat -tulpan | grep ssh
```

Если все нормально, увидим похожий результат:

```
tcp        0      0 *:2233                  *:*
LISTEN      3849/sshd


tcp6       0      0 [::]:2233               [::]:*
LISTEN      3849/sshd
```

или

```
root     sshd       813   3  tcp6   *:2233                *:*


root     sshd       813   4  tcp4   *:2233                *:*
```

Готово. Мы сменили стандартный порт SSH на нестандартный, таким образом усилив безопасность своего сервера