Linux Professional Institute

LPIC-1

جلسه هشتم: جستجوی فایل‌ها و آشنایی با مفاهیم و تنظیمات ابتدایی شبکه

در این جلسه:

ویدئو اول:

ویدئو دوم:

فهرست مطالب

مقدمه

جلسه‌ی قبل، بحثمان در مورد ابزارهای آرشیو در لینوکس را کامل کردیم و سپس به صورت مفصل در مورد مالکیت فایل‌ها و مدیریت مجوزهای دسترسی در لینوکس صحبت کردیم. در این جلسه با چگونگی جستجوی فایل‌ها در لینوکس آشنا می‌شویم، سپس به صورت خیلی کلی با مفاهیم اولیه شبکه آشنا می‌شویم و در نهایت، در مورد چگونگی انجام تنظیمات شبکه در لینوکس صحبت می‌کنیم.

پیدا کردن فایل اجرایی دستورات با *which*

با استفاده از دستور which، می‌توانیم موقعیت قرارگیری فایل اجرایی یک دستور یا برنامه را پیدا کنیم. برای استفاده از این دستور، کافی است نام برنامه‌ی مورد نظر را به آن بدهیم. برای مثال:

```
[root@localhost ~]# which yum
/usr/bin/yum
```

همانطور که می‌بینید، با ارائه‌ی نام دستور yum به which، این دستور موقعیت فایل اجرایی yum را در خروجی به ما نشان داد. جالب است بدانید که دستور which برای پیدا کردن فایل اجرایی دستورات، جستجوی خود را در مسیرهای مشخص شده در متغیر $PATH شِل انجام می‌دهد. ما قبلا در مورد Environment Variableها و همچنین $PATH صحبت کردیم، اما به طور کلی، $PATH متغیری است که به شِل می‌گوید در چه مسیرهایی به دنبال فایل‌های اجرایی بگردد.

دستور which، می‌تواند به ما بگوید که یک دستور از Alias خاصی استفاده می‌کند یا نه. می‌توانید به Alias، به عنوان نام مستعار نگاه کنید؛ Aliasها به ما اجازه می‌دهند که یک دستور را با نام دیگری اجرا کنیم. برای مثال:

```
[root@localhost ~]# which ll
alias ll='ls -l --color=auto'
      /usr/bin/ls
```

همانطور که می‌بینید، دستور ll که محتویات یک دایرکتوری را به صورت لیست شده به ما نشان میداد، در واقع نام مستعار دستور ls -l --color=auto می‌باشد که فایل اجرایی آن در موقعیت /usr/bin/ls قرار دارد. ما بعدا با Aliasها بیشتر آشنا می‌شویم.

پیداکردن فایل‌های اجرایی دستورات با *whereis*

یکی دیگر از دستوراتی که می‌توانیم از آن برای پیدا کردن موقعیت فایل‌های اجرایی دستورات استفاده کنیم، whereis می‌باشد. این دستور علاوه بر نشان دادن موقعیت فایل اجرایی، موقعیت فایل‌های تنظیمات و همچنین manpage یک دستور را نیز به ما نشان می‌دهد. استفاده از این دستور، بسیار شبیه به استفاده از which می‌باشد:

```
[root@localhost ~]# whereis yum
yum: /usr/bin/yum /etc/yum /etc/yum.conf /usr/share/man/man8/yum.8
```

همانطور که می‌بینید، با ارائه‌ی نام دستور مورد نظر به whereis، این دستور در خروجی خود موقعیت فایل باینری yum (/usr/bin/yum)، دایرکتوری مربوط به فایل‌های تنظیمات yum (/etc/yum)، فایل اصلی تنظیمات yum (/etc/yum.conf) و در نهایت موقعیت manpage دستور yum را به ما نشان داد.

این دستور حتی می‌تواند موقعیت لایبرری‌ها، موقعیت daemonها و... برخی از دستورات را به ما نشان دهد.

برای مثال:

```
[root@localhost ~]# whereis tar
tar: /usr/bin/tar /usr/include/tar.h /usr/share/man/man1/tar.1.gz
```

همانطور که می‌بینید، دستور which موقعیت فایل‌های هدر دستور tar (/usr/include/tar.h) را نیز به ما نشان داد.

استفاده از دستور *locate* برای پیدا کردن فایل‌ها

با استفاده از دستور locate، می‌توانیم به سادگی به دنبال فایل‌های مورد نظر در سیستم بگردیم. این دستور می‌تواند در ورودی خود، نام فایل یا حتی بخشی از نام فایل و همچنین regexها را دریافت کرده و سپس به جستجوی فایل مورد نظر ما بپردازد.

دستور locate، در جستجوی خود بسیار سریع می‌باشد و دلیل سرعت آن، جستجو درون دیتابیس کوچکی می‌باشد که توسط خود دستور locate مدیریت می‌شود. این دستور، دیتابیس خود را به صورت اتوماتیک آپدیت می‌کند. این آپدیت روزی یک بار یا هفته‌ای یک بار اجرا می‌شود، به همین دلیل، بعضا ممکن است locate درخروجی خود فایل‌هایی را به ما بازگرداند که دیگر روی سیستم وجود ندارند. البته ما می‌توانیم به صورت دستی نیز دیتابیس locate را آپدیت کنیم.

جالب است بدانید که این دستور به صورت پیش‌فرض روی CentOS 7 Minimal نصب نیست و باید با استفاده از yum آن را نصب کنیم:

```
[root@localhost ~]# yum install mlocate
...
Installed:
  mlocate.x86_64 0:0.26-8.el7

Complete!
```

پس از نصب این دستور، باید دیتابیس آن را برای اولین بار، آپدیت کنیم. برای این کار کافی است دستور updatedb را اجرا کنیم:

```
[root@localhost ~]# updatedb
```

حال می‌توانیم به سراغ استفاده از دستور locate برویم. فرض کنید می‌خواهیم دنبال کلیه‌ی فایل‌هایی که دارای نام main.cf می‌باشند بگردیم. برای این کار:

```
[root@localhost ~]# locate main.cf
/etc/postfix/main.cf
/usr/libexec/postfix/main.cf
/usr/share/doc/postfix-2.10.1/main.cf.default
```

همانطور که می‌بینید، در سیستم ما کلا سه فایل با نام main.cf وجود داشت که موقعیت دقیق آنها در خروجی به ما نمایش داده شد. اما به آخرین فایلی که در خروجی به ما نشان داده شده نگاه کنید. نام این فایل، main.cf.default می‌باشد؛ ما برای main.cf جستجو کرده بودیم، پس چرا این فایل در خروجی به ما نشان داده شده است؟

دلیل این امر این است که locate به صورت پیش‌فرض، علامت * را در ابتدا و انتهای نام جستجو شده قرار می‌دهد. یعنی این دستور، locate main.cf را به صورت locate *main.cf* می‌بیند؛ اگر صحبت‌هایی

که در مورد globbing کردیم را به خاطر بیاورید، این یعنی locate به دنبال کلیه‌ی فایل‌هایی می‌رود که در قسمتی از نام آنها، main.cf وجود داشته باشد؛ بدون توجه به این که در ابتدا و انتهای آن چه کاراکترهای دیگری قرار دارد.

برای این که به locate بگوییم که فقط دنبال نام فایل نوشته شده بگردد و کاراکترهای globbing را به آن اضافه نکند، از آپشن b- استفاده می‌کنیم و نام فایلی که دنبال آن هستیم را بین دو علامت ' قرار می‌دهیم و قبل از نوشتن نام فایل، یک علامت \ قرار می‌دهیم! یعنی:

```
[root@localhost ~]# locate -b '\main.cf'
/etc/postfix/main.cf
/usr/libexec/postfix/main.cf
```

همانطور که می‌بینید، این بار فقط فایل‌هایی که دقیقا نام main.cf را داشتند در خروجی به ما نشان داده شدند.

توجه کنید که اگر یک فایل جدید به سیستم اضافه کنیم یا فایلی را حذف کنیم و...، باید بار دیگر دیتابیس locate را آپدیت کنیم، در غیر این صورت، locate فایل‌های جدید را در جستجوی خود دخیل نمی‌کند.

دستور locate آپشن‌های بسیاری دارد که ما به آنها نمی‌پردازیم. پیشنهاد می‌کنیم که manpage این دستور را مطالعه کنید.

جستجو برای فایل‌ها با استفاده از *find*

دستور find یکی از پراستفاده‌ترین دستورها برای جستجو میان فایل‌های موجود در سیستم می‌باشد. این دستور علاوه بر پیدا کردن فایل‌ها بر حسب نام، به ما اجازه می‌دهد که فایل‌ها را بر حسب مالک آنها، مجوز آنها، تاریخ آخرین تغییر آنها و... نیز پیدا کنیم. به طور کلی، به صورت زیر از دستور find استفاده می‌کنیم:

```
find [PATH...] [OPTION] [EXPRESSION]
```

PATH مشخص کننده‌ی دایرکتوری که قصد داریم جستجو در آن انجام گیرد می‌باشد. دستور find جستجوی خود را از دایرکتوری مشخص شده شروع کرده و سپس داخل کلیه‌ی دایرکتوری‌های موجود درون دایرکتوری مشخص شده نیز رفته و جستجوی خود را ادامه می‌دهد. OPTION و EXPRESSION مشخص کننده‌ی فیلترهایی هستند که بر روی جستجو اعمال می‌شوند. در جدول ۱، برخی از معروف‌ترین ترکیب‌های OPTION و EXPRESSION برای دستور find را مشاهده می‌کنید:

جدول ۱- کاربردی‌ترین آپشن‌ها و اکسپرشن‌های دستور find

OPTION	EXPRESSION	عملکرد
-nmin	n	نام فایل‌هایی که محتویات آنها در n دقیقه‌ی قبل عوض شده را در خروجی نشان می‌دهد.
-group	name	فایل‌هایی که گروه آنها name می‌باشد را در خروجی نشان می‌دهد.
-gid	n	فایل‌های که Group ID آنها برابر با n می‌باشد را در خروجی نشان می‌دهد.
-name	pattern	فایل‌هایی که نامشان دارای الگوی ذکر شده در pattern باشد را در خروجی نشان می‌دهد. pattern می‌تواند شامل regexها باشد. برای عملکرد صحیح، regexها باید بین دو علامت ' قرار بگیرند.

-perm	mode	فایل‌هایی که مجوز آنها برابر با mode باشد را در خروجی نشان می‌دهد. Octal یا Symbolic بودن mode مهم نیست.
-user	name	فایل‌هایی که مالک آنها کاربر name می‌باشد را در خروجی نشان می‌دهد.

توجه داشته باشید مقادیر ستون EXPRESSION، توسط شما مشخص خواهند شد. احتمالا تا به اینجا کمی گیج شده باشید. بیایید این دستور را با یک سری مثال، بهتر درک کنیم. فرض کنید ما در دایرکتوری /etc، دنبال فایل‌هایی هستیم که پسوند .conf دارند:

```
[root@localhost ~]# find /etc/ -name "*.conf"
/etc/pki/ca-trust/ca-legacy.conf
/etc/yum/protected.d/systemd.conf
...
/etc/mke2fs.conf
/etc/tcsd.conf
```

همانطور که می‌بینید، دستور find داخل دایرکتوری /etc و سپس داخل کلیه‌ی دایرکتوری‌های موجود در دایرکتوری /etc شد و هر فایلی که با .conf تمام میشد را در خروجی به ما نشان داد. بیایید کمی در مورد آپشن‌های این دستور با هم صحبت کنیم. ما ابتدا دستور find را وارد کردیم. سپس دایرکتوری که می‌خواهیم جستجو در آن انجام شود را مشخص کردیم (/etc). سپس از آنجایی که می‌خواستیم دنبال یک الگوی خاص بگردیم (پیدا کردن کلیه‌ی فایل‌های دارای پسوند .conf)، از آپشن -name استفاده کردیم و سپس الگوی خاص مورد نظر ("*.conf") را وارد کردیم. توجه کنید که بهتر است الگوی مورد نظر را بین دو علامت " قرار دهیم.

دستور find برای بازرسی سیستم نیز به کار می‌رود. مثلا ما در جلسه‌ی قبل گفتیم که باید حواسمان به فایل‌هایی که دارای مجوز SUID هستند، باشد؛ چرا که این فایل‌ها می‌توانند امنیت سیستم را تحت تاثیر قرار دهند. ما می‌توانیم با استفاده از find، کلیه‌ی فایل‌هایی که دارای مجوز SUID هستند را پیدا کنیم. یعنی:

```
[root@localhost ~]# find / -perm /4000
...
/usr/bin/crontab
/usr/bin/pkexec
/usr/bin/passwd
...
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/libexec/dbus-1/dbus-daemon-launch-helper
```

همانطور که می‌بینید، دستور find کلیه‌ی فایل‌هایی که مجوز SUID را داشتند در خروجی به ما نشان داد. اما بیایید در مورد آپشن‌های اعمال شده به find صحبت کنیم. همانطور که می‌بینید، ما ابتدا دستور find را وارد کرده و سپس مشخص می‌کنیم که میخواهیم جستجو در کل سیستم انجام شود (/). سپس با استفاده از -perm، به find می‌گوییم که میخواهیم بر حسب مجوز در سیستم جستجو کنیم. سپس /4000 را وارد می‌کنیم. اول بیایید مفهوم / را با هم بررسی کنیم. علامت /، به find می‌گوید که اگر فایل، فقط یکی از مجوزهای مشخص شده را داشت، آن فایل را به ما بازگرداند . یعنی اگر دستوری مانند find / -perm 642 را وارد کنیم، هر فایلی که مجوز مالک آن ۶ باشد را، بدون توجه به مجوز سایر لایه‌ها، به ما نشان می‌دهد. به همین ترتیب، اگر مجوز گروه فایلی ٤ باشد، بدون توجه به مجوز سایر لایه‌ها، به ما بازگردادنده می‌شود.

پس از علامت /، مقدار 4000 نوشته شده است. اگر مفاهیم جلسه قبل را به خاطر داشته باشید، گفتیم که در

مجوزهای ٤ رقمی در حالت Octal، عدد اول نشان دهنده‌ی مجوز خاص، و ارقام بعد به ترتیب نشان دهنده‌ی مجوز مالک، گروه و سایرین می‌باشد. ما اینجا عدد 4 را به عنوان رقم اول قرار داده‌ایم. عدد 4، نشان دهنده‌ی مجوز SUID می‌باشد و 000 وقتی با / بیاید، به معنای هر مجوزی می‌باشد. اگر 4000/ را در کنار هم ببینیم، یعنی داریم به find می‌گوییم که دنبال فایل‌هایی که بیت مجوز ویژه‌ی آن 4 و سایر مجوزهای آن هر چیزی می‌باشد، بگردد.

دستور find آپشن‌ها و قابلیت‌های بیشتری دارد که در صورت تمایل به آشنایی با آنها، بهتر است manpage این دستور را مطالعه کنید.

استفاده از دستور type

در سیستم‌های لینوکسی، دستورها دارای ۳ نوع می‌باشند:

- دستورهای Alias
- دستورهای Builtin
- دستورهای External

قبلا به صورت خیلی ابتدایی در مورد Alias صحبت کردیم، اما بار دیگر نیز میگوییم که دستورهای Alias، دستورهایی هستند که به عنوان نام مستعار برای یک دستور دیگر عمل می‌کنند. شاید بهترین مثال برای نشان دادن این امر، دستور ls باشد:

```
[root@localhost ~]# type ls
ls is aliased to `ls --color=auto'
```

همانطور که می‌بینید، دستور ls، نام مستعار دستور ls --color=auto می‌باشد. یعنی ls، نام مستعار با یک آپشن خاص می‌باشد.

دستورهای Builtin دستورهایی هستند که همراه با خود شِل (در اینجا bash) آمده‌اند و ما آنها را به صورت جداگانه روی سیستم نصب نکرده‌ایم. به عبارت دیگر، دستورهای Builtin دستورهایی هستند که در هر سیستمی که شِل bash داشته باشد، موجود خواهند بود. برای مثال:

```
[root@localhost ~]# type cd
cd is a shell builtin
```

همانطور که می‌بینید، دستور cd یک دستور Builtin می‌باشد؛ یعنی اگر با یک سیستم که دارای شِل bash می‌باشد کار کنیم، دستور cd در آن سیستم موجود خواهد بود.

دستورهای External، دستورهایی هستند که توسط ما یا توسط کسانی که یک توزیع لینوکسی را ایجاد کرده‌اند روی سیستم نصب شده‌اند. برای مثال:

```
[root@localhost ~]# type locate
locate is /usr/bin/locate
```

همانطور که می‌بینید، خروجی type به ما می‌گوید که locate در موقعیت /usr/bin/locate قرار دارد. در واقع دستور type، موقعیت فایل اجرایی این دستور را به ما می‌گوید و ما از این میفهمیم که دستور locate، دستوری است که توسط خود ما یا کسانی که CentOS را ایجاد کرده‌اند روی سیستم نصب شده است و هیچ تضمینی نیست که این دستور روی یک سیستم لینوکسی دیگر، وجود داشته باشد.

مبانی شبکه

قبل از این که در مورد چگونگی انجام تنظیمات متفاوت شبکه در لینوکس صحبت کنیم، باید کمی با مفاهیم اولیه شبکه آشنا شویم. به طور کلی، شبکه‌های کامپیوتری به ما اجازه می‌دهند که اطلاعات را از یک کامپیوتر به کامپیوتر دیگر منتقل کنیم. می‌توان به شبکه‌های کامپیوتری، به عنوان یک سری سیستم لایه‌ای نگاه کرد؛ به طوری که هر لایه، نقش متفاوتی را در انتقال اطلاعات از یک کامپیوتر به کامپیوتر دیگر ایفا می‌کند.

روش‌های متفاوتی برای توصیف لایه‌های موجود در شبکه‌های کامپیوتری وجود دارد. برای مثال، یکی از معروف‌ترین استانداردهای شبکه به نام Open Systems Interconnection یا OSI، شبکه‌های کامپیوتری را به ۷ لایه تقسیم می‌کند. از آنجایی که ما می‌خواهیم به صورت خیلی کلی در مورد شبکه صحبت کنیم، از یک مدل ساده شده‌ی ٤ لایه‌ای استفاده می‌کنیم. این ٤ لایه به شرح زیر می‌باشند:

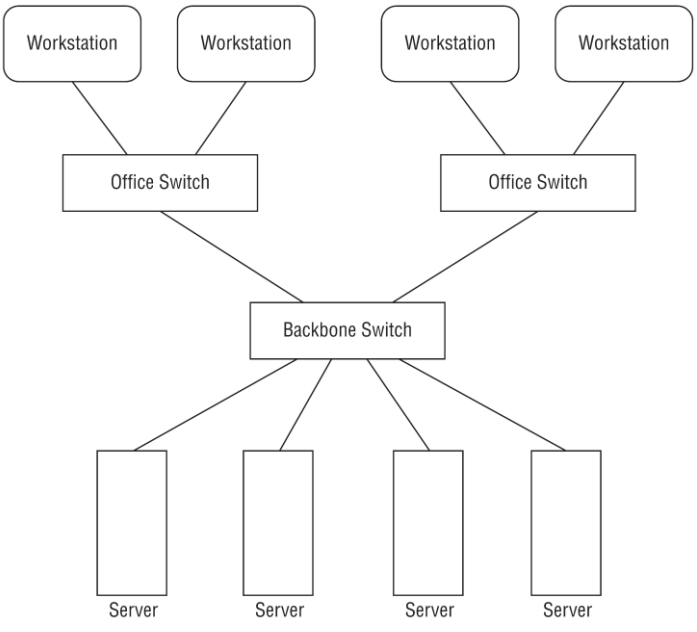
- لایه‌ی فیزیکی
- لایه‌ی شبکه
- لایه‌ی انتقال
- لایه‌ی کاربرد

در این بخش، تک‌تک این لایه‌ها را به طور مختصر توضیح می‌دهیم.

لایه‌ی فیزیکی

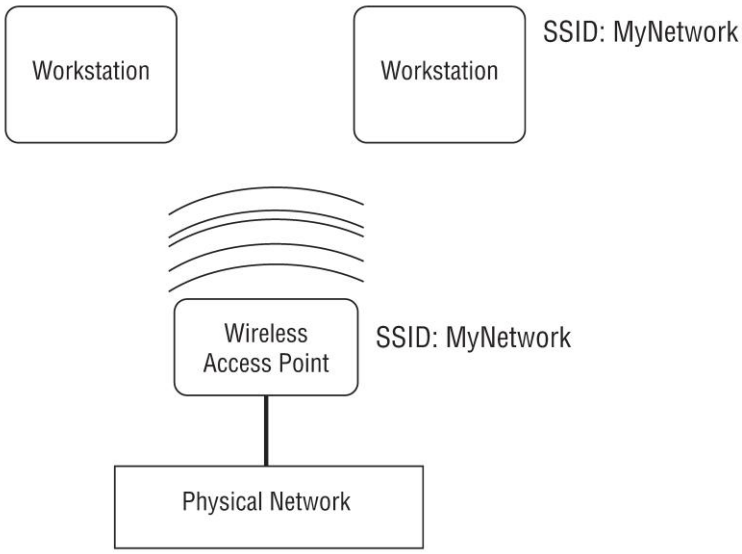
لایه‌ی فیزیکی، شامل کلیه‌ی سخت‌افزارهایی می‌باشد که با آن به شبکه متصل می‌شویم. دو روش کلی برای اتصال به شبکه وجود دارد: اتصال سیمی و اتصال بی‌سیم.

در اتصال سیمی، کامپیوترها با استفاده از یک سری سیم مخصوص، به دستگاه‌هایی به نام سوئیچ متصل می‌شوند. سوئیچ وظیفه‌ی دریافت اطلاعات از یک کامپیوتر و تحویل آن به مقصد صحیح را دارد. بحث بیشتر در مورد سوئیچ‌ها و معماری آنها از حوصله‌ی ما خارج است، اما به طور کلی، در یک ساختمان بزرگ، ممکن است شبکه دارای همچین معماری باشد:



تصویر ۱ - نمایی از معماری شبکه‌های دارای اتصال سیمی

در اتصال بی‌سیم، همانطور که از نامش پیداست، ما به صورت بی‌سیم به سایر سیستم‌ها متصل می‌شویم. شبکه‌های بی‌سیم از سیگنال‌های رادیویی برای انتقال اطلاعات بین یک سیستم و یک Access Point استفاده می‌کنند. Access Point در سیستم‌های بی‌سیم، عملکردی شبیه به سوئیچ در سیستم‌های سیمی را دارد؛ یعنی این دستگاه وظیفه‌ی مدیریت و کنترل انتقال اطلاعات به دستگاه‌هایی که به آن متصل هستند را بر عهده دارد. هر اکسس پوینت، یک شناسه‌ی به نام SSID دارد که کاربران می‌توانند از طریق آن، اکسس پوینت را دیده و به آن متصل شوند. SSID می‌تواند یک رشته یا یک عدد باشد. به طوری کلی اتصال بی‌سیم نمایی نظیر تصویر ۲ دارد:



تصویر ۲- نمایی از معماری شبکه‌های دارای اتصال بی‌سیم

مشکل شبکه‌ی بی‌سیم در این است که ما نمی‌توانیم مسیر سیگنال رادیویی را کنترل کنیم، یعنی کسانی که خارج از محیط مورد نظر ما هستند نیز می‌توانند اکسس‌پوینت را ببینند و سعی کنند به آن متصل شوند. به همین دلیل، ما باید اکسس‌پوینت را با استفاده از پروتکل‌های معمول، رمزگذاری کنیم. از معروف‌ترین پروتکل‌های رمزگذاری در شبکه‌های بی‌سیم، می‌توان WEP، WPA و WPA2 را نام برد.

لایه‌ی شبکه

لایه‌ی شبکه، چگونگی انتقال اطلاعات بین دستگاه‌های وصل شده به شبکه (شبکه‌ی خانگی یا شبکه‌ی اینترنت) را کنترل می‌کند. برای این که بتوانیم اطلاعات را به مقصد صحیح برسانیم، باید اطلاعاتی در مورد مقصد داشته باشیم؛ یعنی مثلا باید بدانیم که آدرس مقصد چیست. هر دستگاه موجود در شبکه، باید یک آدرس منحصر به فرد داشته باشد تا دیگران بتوانند آن دستگاه را به سادگی پیدا کنند. آدرس IP، یکی از معروف‌ترین روش‌ها برای آدرس‌دهی به سیستم‌ها می‌باشد.

به طور کلی، برای این که یک کامپیوتر را به یک شبکه وصل کنیم، به ٤ چیز نیاز داریم:

- آدرس IP
- Netmask
- Default Gateway
- Hostname

در ادامه، هر کدام از این موارد را توضیح می‌دهیم.

آدرس IP

دقیقا همانطور که هر فرد در یک شهر، برای خود آدرسی منحصر به فرد دارد، هر دستگاه موجود در شبکه نیز یک آدرس منحصر به فرد، که به آن IP Address یا آدرس IP می‌گویند، دارد. آدرس IP در یک شبکه به ما کمک می‌کند که بتوانیم به راحتی اطلاعات مورد نیاز خود را به یک سیستم دیگر بفرستیم. می‌توانید به این امر دقیقا مانند نوشتن یک ایمیل فکر کنید. ما هنگام ارسال ایمیل، باید آدرس ایمیل دریافت کننده را داشته باشیم. در شبکه نیز برای ارسال اطلاعات به یک فرد خاص، باید آدرس IP آن فرد را داشته باشیم.

آدرس IP، از ٤ عدد که با نقطه از هم جدا می‌شوند، تشکیل شده است. برای مثال:

۱۹۲,۱۶۸,۱,۱

هر کدام از این ٤ عدد، می‌توانند مقداری بین ۰ تا ۲۵۵ داشته باشند. به صورت دقیق‌تر، آدرس IP، یک آدرس ۳۲ بیتی می‌باشد که به چهار عدد ۸ بیتی تقیسم شده است.

به طور کلی، آدرس IP به دو بخش کلی تقیسم می‌شود:

- بخش آدرس شبکه (Network Address)
- بخش آدرس هاست (Host Address)

کلیه‌ی دستگاه‌هایی که در یک شبکه‌ی فیزیکی می‌باشند و می‌خواهند با هم ارتباط برقرار کنند، باید آدرس شبکه‌ی یکسانی داشته باشند و هر دستگاه موجود در شبکه، باید یک آدرس هاست منحصر به فرد داشته باشد. اما ما چگونه می‌توانیم بفهمیم که کدام بخش از آدرس IP، آدرس شبکه و کدام بخش آدرس هاست می‌باشد؟ ما این کار را با کمک Netmask انجام می‌دهیم، اما قبل از این که در مورد Netmaskها صحبت کنیم، باید با مسئله‌ای دیگر در مورد IPها، آشنا شویم.

به طور کلی، آدرس‌های IP دو نوع می‌باشند:

- آدرس‌های Private
- آدرس‌های Public

آدرس‌های Private، آدرس‌هایی هستند که در شبکه‌های محلی، مثل شبکه‌ی خانگی و... از آن استفاده می‌کنیم، اما نمی‌توانیم آن آدرس را به عنوان آدرس خود در اینترنت، به کار بریم (نمی‌توانیم با آن آدرس به اینترنت متصل شویم).

آدرس‌های Public، آدرس‌هایی هستند که می‌توانیم از آن به عنوان آدرس خود در اینترنت، استفاده کنیم (با آن آدرس به اینترنت متصل شویم). در اینترنت نیز دقیقا مانند شبکه‌های خانگی، نباید هیچ آدرس IP یکسانی وجود داشته باشد. به همین دلیل، سازمانی به نام IANA، وظیفه‌ی واگذاری IPهای Public به کسانی که می‌خواهند به اینترنت متصل شوند را بر عهده دارد.

برای جلوگیری از به وجود آمدن هر گونه سردرگمی، سه بلوک آدرس IP به عنوان آدرس‌های Private در نظر گرفته شده‌اند؛ یعنی بهتر است شبکه‌های محلی، از این سه بلوک IP جهت اختصاص آدرس به سیستم‌های درون خود استفاده کنند. این بلوک‌ها به شرح زیر می‌باشند:

- آدرس 10.10.10.0 تا 10.255.255.255
- آدرس 172.16.0.0 تا 172.16.255.255

- آدرس 192.168.0.0 تا 192.168.255.255

Netmask

گفتیم که Netmask، آدرس شبکه را از آدرس هاست جدا می‌کند. از نظر قیافه‌ای، Netmask بسیار شبیه به آدرس IP می‌باشد؛ یعنی Netmask نیز از ٤ عدد که با یک نقطه از هم جدا شده‌اند تشکیل شده و هر عدد، می‌تواند در محدوده‌ی ۰ تا ۲۵۵ باشد. با توجه به این حرف، می‌توانیم بگوییم که Netmask نیز یک عدد ۳۲ بیتی می‌باشد که به ٤ عدد ۸ بیتی، تقیسم شده است.

برای درک تفاوت Netmask با آدرس IP، باید کمی در مورد بیت‌ها و سیستم باینری صحبت کنیم. Netmask مشخص می‌کند که کدام بیت‌های موجود در آدرس IP، مربوط به آدرس شبکه می‌باشند؛ یعنی بیت‌هایی در آدرس IP که نشان دهنده‌ی آدرس شبکه می‌باشند، با بیت ۱ و بیت‌هایی که نشان دهنده‌ی آدرس هاست می‌باشند، با بیت ۰ نشان داده می‌شوند. فرض کنید ما یک آدرس IP با مقدار ۱۹۲,۱۶۸,۱,۱ را داریم و می‌خواهیم مشخص کنیم که ۳ عدد اول آدرس (۱۹۲,۱۶۸,۱) آدرس شبکه‌ی ما و عدد آخر (.۱) آدرس هاست ما می‌باشد.

برای این کار، آدرس IP را تبدیل به مقدار باینری آن می‌کنیم. گفتیم که می‌خواهیم سه عدد اول IP، بیانگر آدرس شبکه‌ی ما باشند. از آنجایی که در Netmask، بیت ۱ نشان دهنده‌ی آدرس شبکه می‌باشد، کلیه‌ی بیت‌های موجود در سه عدد اول Netmask را برابر با ۱ قرار می‌دهیم. این امر، به سیستم می‌گوید که سه عدد اول آدرس IP (یا ۲٤ بیت اول آدرس IP) نمایانگر آدرس شبکه می‌باشند و عدد آخر آدرس IP (یا ۸ بیت آخر آدرس IP)، نمایانگر آدرس هاست می‌باشد:

آدرس IP:	192.168.1.1
آدرس IP به باینری:	11000000.10101000.00000001.000000001
Netmask به باینری:	11111111.11111111.11111111.00000000

تصویر ۳- تبدیل آدرس IP به باینری و مشخص کردن Netmask

البته ما هیچ وقت Netmask را به صورت باینری به سیستم نمی‌دهیم و دقیقا مانند آدرس IP، هر کدام از قسمت‌های ۸ بیتی آن را تبدیل به اعداد دسیمال می‌کنیم. یعنی در بالا، Netmask ما برابر با ۲۵۵,۲۵۵,۲۵۵,۰ خواهد شد:

IP:	1	1.	168.	192.
Netmask:	0	255.	255.	255.
آدرس هاست	آدرس شبکه			

تصویر ٤- تبدیل آدرس Netmask به دسیمال

با نگاه کردن به تصویر ٤، نباید درک الگوی موجود در آدرس IP و Netmask دشوار باشد. اگر عدد اول در Netmask برابر با ۲۵۵ باشد، یعنی عدد اول در IP نیز بخشی از آدرس شبکه است، اگر عدد دوم در

Netmask برابر با ۲۵۵ باشد، یعنی عدد دوم در IP نیز بخشی از آدرس شبکه است و به همین ترتیب.
روش دیگری نیز برای مشخص کردن Netmask وجود دارد که به آن CIDR می‌گویند. در این روش، به جای ارائه‌ی ٤ عدد معمول Netmask، ما تعداد بیت‌هایی از آدرس IP که نشان دهنده‌ی آدرس شبکه می‌باشند را با یک علامت / مشخص می‌کنیم. مثلا در مثال بالا، ۲٤ بیت از ۳۲ بیت IP، نشان دهنده‌ی آدرس شبکه می‌باشند، پس Netmask این آدرس در حالت CIDR، برابر با 24/ خواهد شد.

Default Gateway

اگر گفته‌های ما در بخش آدرس IP را به خاطر داشته باشید، گفتیم که کلیه‌ی دستگاه‌هایی که در یک شبکه‌ی فیزیکی می‌باشند و می‌خواهند با هم ارتباط برقرار کنند، باید آدرس شبکه‌ی یکسانی داشته باشند. حالا اگر بخواهیم به یک شبکه‌ی دیگر، که آدرس شبکه‌ی متفاوتی دارد متصل شویم، باید چه کنیم؟
اتصال یک شبکه به یک شبکه‌ی دیگر، توسط دستگاهی به نام روتر صورت می‌پذیرد. به طور خیلی ساده، روتر می‌تواند اطلاعاتی که می‌خواهیم به یک شبکه‌ی دیگر ارسال کنیم را مسیریابی کرده و آن را در مسیر تحویل به مقصد، قرار دهد. در اکثر شبکه‌ها، معمولا یک روتر وجود دارد که وظیفه‌ی مسیریابی اطلاعات را بر عهده دارد. به این روتر، Default Gateway می‌گویند. Default Gateway، یک آدرس IP دارد که کلیه‌ی اطلاعاتی که قرار است به یک شبکه‌ی دیگر (مثل اینترنت) ارسال شوند، ابتدا برای آن آدرس ارسال شده و آن دستگاه، عمل مسیریابی اطلاعات و تحویل آن به یک شبکه‌ی دیگر را انجام می‌دهد.

Hostname

اگر به خاطر داشته باشید، گفتیم که برای ارسال اطلاعات و وصل شدن به سایر دستگاه‌ها در شبکه، باید آدرس IP آنها را داشته باشیم. اما چرا وقتی می‌خواهیم به یک وبسایت متصل شویم، به جای آدرس IP آن سایت، فقط نام آن را وارد می‌کنیم؟ دلیل این امر، وجود سیستمی به نام DNS یا Domain Name System می‌باشد. DNS، به هر آدرس IP، یک نام اختصاص می‌دهد. وجود نام، باعث می‌شود تا برقراری ارتباط با سایر سیستم‌ها راحت‌تر شود، چرا که در صورت عدم استفاده از نام، باید هزاران آدرس IP را حفظ باشیم.
DNS، به هر شبکه (آدرس شبکه) یک دامنه (Domain Name) اختصاص می‌دهد (چیزی شبیه linux.org).
این دامنه، منحصر به آن شبکه می‌باشد. سپس به هر هاست موجود در آن شبکه، یک Hostname منحصر به فرد اختصاص می‌دهد. بدین صورت، برای ارتباط با هر هاست موجود در یک شبکه، کافی است نام هاست و سپس نام دامنه را وارد کنیم؛ یعنی چیزی شبیه به thealbatross.linux.org.
البته برای این که بتوانیم در شبکه‌های محلی از این قابلیت استفاده کنیم، باید یک سرور DNS محلی داشته باشیم. سرورهای DNS محلی، نام هاست و نام دامنه را به آدرس IP اختصاص یافته به آن هاست تبدیل می‌کنند و برای پیدا کردن هاست‌های غیرمحلی، به DNSهای دیگر وصل شده و آدرس IP هاست مورد نظر را به ما می‌دهند.
برای استفاده از DNS، نیاز به آدرس IP یک سرور DNS داریم. با توجه به این که بسیاری از مردم در شبکه‌ی خانگی خود نیازی به داشتن یک سرور DNS اختصاصی ندارند، بسیاری از سرورهای DNS عمومی (نظیر 8.8.8.8) در اینترنت موجود می‌باشند که می‌توانند توسط کاربران به کار برده شوند.

Dynamic Host Configuration Protocol (DHCP)

حال که در مورد آدرس IP، Netmask، Default Gateway و DNS با هم صحبت کرده‌ایم، بهتر است در مورد DHCP نیز با هم صحبت کنیم. خیلی از اوقات، مدیریت این که به چه کسی، چه آدرس IP را اختصاص دهیم، کار بسیار دشواری می‌شود. به علاوه، در سیستم‌های بزرگتر، اختصاص دستی IP به تک تک سیستم‌های موجود در آن شبکه، کار بسیار زمان‌بری می‌شود.

DHCP، کار اختصاص IP، Netmask و... به سیستم‌ها را بسیار ساده‌تر می‌کند. DHCP به کلیه‌ی کاربران موجود در شبکه، یک آدرس IP، Netmask، Default Gateway و آدرس DNS، اختصاص می‌دهد. البته آدرس اختصاص یافته توسط DHCP، ثابت نیست و ممکن است در طول زمان، یا در صورت خاموش و روشن کردن سیستم، تغییر پیدا کند.

نکته: وجود DHCP، کار را برای کاربران معمولی شبکه بسیار ساده می‌کند، اما توجه داشته باشید که در صورت وجود یک سرور در شبکه، بهتر است آن سرور، یک IP ثابت داشته باشد. همانطور که گفتیم IP اختصاص یافته توسط DHCP، ثابت نیست و می‌تواند دچار تغییر شود.

لایه‌ی انتقال (Transport)

گفتیم که لایه‌ی شبکه، وظیفه‌ی انتقال اطلاعات به یک هاست یا یک سیستم خاص را دارد. لایه‌ی انتقال، وظیفه‌ی تحویل آن اطلاعات به یک برنامه‌ی خاص بر روی هاست را بر عهده دارد. این لایه، این کار را با استفاده از شماره‌ی پورت انجام می‌دهد.

میتوانید به پورت‌ها، مثل شماره‌ی واحد در یک آپارتمان نگاه کنید؛ یعنی برای ارسال اطلاعات به یک سیستم، باید آدرس IP آن سیستم (آدرس آپارتمان) و همچنین شماره‌ی پورت برنامه‌ای که باید اطلاعات را دریافت کند (شماره‌ی واحد) را داشته باشیم.

به طور کلی، لایه‌ی انتقال برای تحویل اطلاعات به یک برنامه بر روی یک هاست خاص، از دو پروتکل استفاده می‌کند:

- پروتکل Transmission Control Protocol (TCP)
- پروتکل User Datagram Protocol (UDP)

پروتکل TCP، به ما اطمینان می‌دهد که تک‌تک اطلاعاتی که ارسال می‌کنیم، حتما به مقصد خود می‌رسند. وجود این اطمینان، باعث می‌شود که مجبور باشیم انتقال اطلاعات را به صورت مدام تایید کنیم که باعث می‌شود عملیات انتقال، کم‌سرعت شود.

برای اطلاعاتی که در آن سرعت انتقال مهم است (ویدئو و...)، از پروتکل UDP استفاده می‌کنیم. UDP به ما تضمینی در مورد انتقال اطلاعات نمیدهد؛ بلکه فقط اطلاعات را می‌فرستد. ممکن است فکر کنید عدم وجود تضمین در انتقال اطلاعات یا از بین رفتن برخی بسته‌های اطلاعاتی، می‌تواند فاجعه‌بار باشد. اما برای ویدئو و صدا، این امر زیاد مهم نیست، چون ممکن است فقط یک فریم از ویدئو یا چند میلی‌ثانیه از صدا به مقصد نرسد؛ برای این نوع داده، این میزان افت اطلاعات برای ما مهم نیست.

لایه‌ی کاربرد

لایه‌ی کاربرد، جایی است که برنامه‌های موجود در سیستم، اطلاعاتی که به آنها ارسال شده را پردازش کرده و به آنها پاسخ می‌دهند. اکثر برنامه‌های شبکه‌ای از الگوی کلاینت سرور استفاده می‌کنند. در این الگو، یکی از دستگاه‌های موجود در شبکه به عنوان سرور عمل می‌کند؛ یعنی آن دستگاه، سرویس‌هایی را به چندین دستگاه در شبکه ارائه می‌دهد (مثل یک وب‌سرور که صفحات وب را در اختیار کاربران قرار می‌دهد). سرور، دائما بر روی یک آدرس پورت خاص (وب، DNS و...) در حال گوش کردن می‌باشد و هر کاربری که بخواهد درخواست یک خدمت را به آن سرور بفرستد، باید درخواست را به آن شماره پورت بفرستد.

برای ساده‌سازی این امر، هم TCP و هم UDP از پورت‌های Well-Known برای نمایان کردن برنامه‌های معروف اسفاده می‌کنند. این شماره پورت‌ها، به صورت رزرو شده برای برخی از خدمات می‌باشند و بدین ترتیب، کاربران می‌دانند که برای دریافت هر نوع خدمت، باید درخواست خود را روی چه پورتی ارسال کنند.

برخی از پورت‌های Well-Known به شرح زیر می‌باشند:

جدول ۲- شماره‌ی پورت‌های Well-Known

شماره پورت	پروتکل	سرویس
۲۰	TCP	اطلاعات FTP
۲۱	TCP	پیام‌های کنترلی FTP
۲۲	TCP	SSH
۲۳	TCP	Telnet
۲۵	TCP	SMTP
۵۳	TCP&UDP	DNS
۸۰	TCP	HTTP
۱۱۰	TCP	POP3
۱۲۳	UDP	NTP
۱۳۹	TCP	NetBIOS
۱۴۳	TCP	IMAP
۳۸۹	TCP	LDAP
۴۴۳	TCP	HTTPS

از آنجایی که حفظ کردن همه‌ی این پورت‌ها کار دشواری می‌باشد، در سیستم‌های لینوکسی، به هر پورت، یک نام اختصاص داده شده است. این نام‌ها و شماره‌ی پورت‌ها را می‌توانید در فایل `/etc/services` مشاهده کنید.

تنظیمات شبکه در لینوکس

برای این که سیستم لینوکس را برای اتصال به شبکه آماده کنیم، نیاز به ۵ قطعه اطلاعات داریم:

- آدرس IP سیستم
- Netmask شبکه
- Default Gateway
- Hostname
- آدرس سرور DNS

ما می‌توانیم این تنظیمات را با اعمال تغییرات در فایل‌های تنظیمات شبکه و همچنین استفاده از برنامه‌های کامندلاین انجام دهیم. در این بخش به بررسی هر دو روش می‌پردازیم.

اعمال تنظیمات شبکه با ایجاد تغییر در فایل‌های تنظیمات

یکی از ساده‌ترین روش‌های تنظیم شبکه در لینوکس، اعمال تغییرات در فایل‌های مربوط به شبکه می‌باشد. متاسفانه موقعیت قرارگیری فایل‌های تنظیمات شبکه در توزیع‌های متفاوت، دارای استاندارد خاصی نمی‌باشد و دچار تغییر می‌شود. از آنجایی که ما با یک توزیع Red Hat-based کار می‌کنیم، تمرکز خود را بر روی فایل‌های تنظیم شبکه در این توزیع‌ها می‌گذاریم.

در سیستم‌های Red Hat-based، دایرکتوری `/etc/sysconfig/network-scripts` میزبان فایل‌های تنظیمات شبکه می‌باشد. اطلاعات مربوط به تنظیمات آدرس IP، Netmask و... مربوط به هر کارت شبکه، در فایلی با پیشوند `ifcfg-` و سپس نام اینترفیس شبکه، قرار دارد. برای پیدا کردن نام اینترفیس شبکه، می‌توانیم از دستور `ip a` استفاده کنیم:

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
qlen 1000
    link/ether 00:0c:29:5d:15:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.50/24 brd 192.168.1.255 scope global noprefixroute ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::e836:9b5f:6eb:67/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

همانطور که می‌بینید، نام اینترفیس شبکه‌ی ما، `ens33` می‌باشد، پس فایل مربوط به تنظیمات این کارت شبکه، در موقعیت `/etc/sysconfig/network-scripts/ifcfg-ens33` قرار خواهد داشت. برای اعمال تغییرات در این فایل، باید آن را با یک ادیتور نظیر `vi`، باز کنیم:

```
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-ens33
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
```

```
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=56002cf0-9e73-4617-8e54-3556bdbeb508
DEVICE=ens33
ONBOOT=yes
IPADDR=192.168.1.50
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=8.8.8.8
DNS2=4.2.2.4
```

در این فایل، خط‌های مشخص شده برای ما مهم هستند. مفهوم هر کدام از آنها را در جدول ۳ می‌بینیم:

جدول ۳- مفهوم متغیرهای مهم در فایل تنظیمات کارت شبکه

متغیر	مقدار	توضیحات
BOOTPROTO	static (یا none) یا dhcp	چگونگی دریافت تنظیمات شبکه را مشخص می‌کند. مقدار static (یا none) یعنی تنظیمات را به صورت دستی وارد می‌کنیم و dhcp یعنی تنظیمات را به صورت اتوماتیک از یک سرور DHCP دریافت می‌کنیم.
ONBOOT	yes یا no	فعال بودن این کارت شبکه هنگام روشن شدن سیستم را مشخص می‌کند. مقدار yes باعث می‌شود این کارت شبکه هنگام روشن شدن سیستم فعال باشد و مقدار no باعث می‌شود که این کارت شبکه هنگام روشن شدن سیستم غیرفعال باشد.
IPADDR	آدرس IP	آدرس IP که باید روی این کارت شبکه تنظیم شود را مشخص می‌کند.
NETMASK یا **PREFIX**	مقدار Netmask یا در صورت استفاده از PREFIX، مقدار Netmask به صورت CIDR	Netmask مربوط به آدرس IP مشخص شده روی کارت شبکه را مشخص می‌کند.
GATEWAY	آدرس IP	آدرس IP مربوط Default Gateway موجود در شبکه را مشخص می‌کند.
DNS1 **DNS2**	آدرس IP	آدرس IP سرورهای DNS را برای کارت شبکه مشخص می‌کند. این آدرس می‌تواند آدرس سرور DNS محلی یا عمومی باشد.

توجه کنید که ممکن است که در فایل تنظیمات مربوط به کارت شبکه‌ی شما هر کدام از این متغیرها وجود نداشته باشند. شما می‌توانید در صورت عدم وجود یکی از مقادیر، خودتان آن را به صورت دستی وارد کنید.

علاوه بر این، بعضا ممکن است در صورت اضافه کردن یک کارت شبکه جدید به سیستم، فایل مربوط به تنظیمات آن در `/etc/sysconfig/network-scripts` وجود نداشته باشد. در این حالت، خودمان باید فایلی با نام

-ifcfg به همراه نام کارت شبکه‌ی جدید ایجاد کرده و مقادیر مورد نظر را در آن وارد کنیم.

در صورت اعمال تغییرات در یکی از فایل‌های تنظیمات شبکه، باید آن کارت شبکه‌ی خاص را یک بار غیر فعال و سپس فعال کنیم. برای این کار، می‌توانیم از دستور ifdown و ifup به همراه نام اینترفیس مورد نظر، استفاده کنیم:

```
[root@localhost ~]# ifdown ens33 && ifup ens33
Device 'ens33' successfully disconnected.
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/4)
```

دستور ifdown ens33، کارت شبکه‌ی ens33 را غیرفعال می‌کند و دستور ifup ens33 کارت شبکه‌ی ens33 را فعال می‌کند و باعث می‌شود که سیستم بار دیگر فایل مربوط به تنظیمات کارت شبکه‌ی ens33 را از اول بخواند. دلیل این که ما از علامت && بین این دو دستور استفاده کردیم، این بود که می‌خواستیم بلافاصله پس از اجرای موفقیت‌آمیز دستور ifdown، دستور ifup اجرا شود. این امر زمانی که به صورت SSH به سیستم متصل شده باشیم و بخواهیم یک کارت شبکه را غیرفعال و دوباره فعال کنیم به کار می‌آید، چرا که در غیر این صورت، به محض وارد کردن دستور ifdown، ارتباط ما با سیستم قطع شده و دیگر نمی‌توانیم دستور ifup را وارد کنیم.

برای تنظیم Hostname در سیستم‌های Red Hat-based، باید فایل /etc/hostname را با یک ادیتور نظیر vi باز کرده، Hostname قبلی را پاک کرده و مقدار Hostname جدید را درون آن وارد کنیم:

```
[root@localhost ~]# vi /etc/hostname
```

پس از وارد کردن Hostname مورد نظر خود در این فایل و ذخیره‌ی آن، باید سیستم را ریبوت کنیم تا Hostname جدید روی سیستم قرار گیرد:

```
[root@localhost ~]# init 6
```

پس از روشن شدن دوباره‌ی سیستم، می‌توانیم با استفاده از دستور hostname، مقدار Hostname سیستم خود را مشاهده کنیم:

```
[root@The Albatross ~]# hostname
The Albatross
```

همانطور که می‌بینید، اکنون Hostname سیستم ما تبدیل به The Albatross شده است. علاوه بر این، prompt شِل نیز این امر را به ما نشان می‌دهد ([root@The Albatross ~]).

اعمال تنظیمات شبکه با استفاده از برنامه‌های کامندلاین

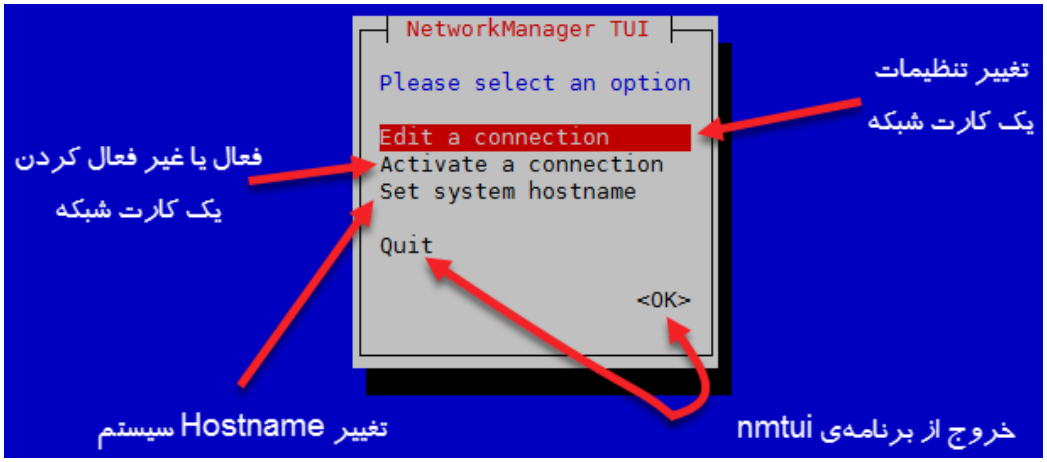
ما همیشه مجبور نیستیم فایل‌های تنظیمات شبکه را به صورت دستی تغییر دهیم و می‌توانیم از برخی برنامه‌ها که این کار را برای ما ساده‌تر می‌کنند، استفاده کنیم. در این بخش، با برخی از این برنامه‌ها آشنا می‌شویم.

تنظیم شبکه با نرم‌افزار *nmtui*

نرم‌افزار nmtui، یک نرم‌افزار کامندلاینی می‌باشد که دارای یک نمای گرافیکال تحت کامندلاین می‌باشد. با استفاده از این دستور، می‌توانیم تنظیمات شبکه را تغییر دهیم:

```
[root@The Albatross ~]# nmtui
```

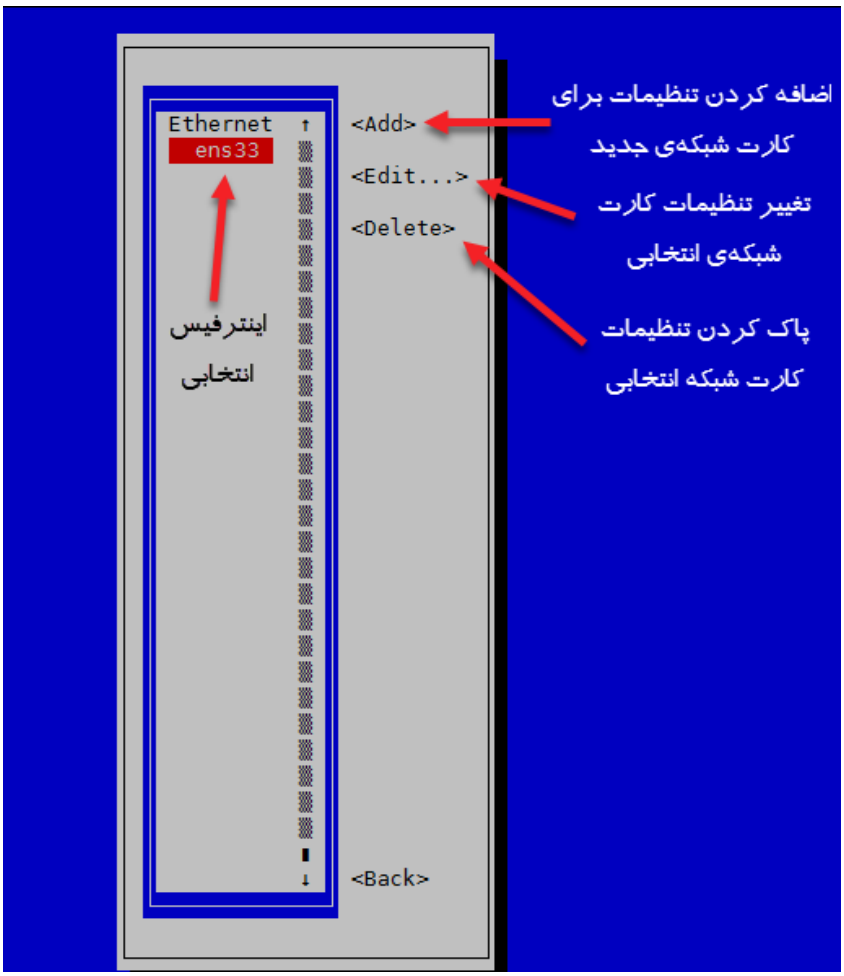
به محض وارد کردن این دستور، با نمایی نظیر تصویر ۵ مواجه می‌شویم. همانطور که می‌بینید، این برنامه به ما سه آپشن متفاوت می‌دهد. برای انتخاب هر کدام از این آپشن‌ها، کافی است با دکمه‌های ↑ و ↓ آپشن مورد نظر را انتخاب و سپس دکمه‌ی Enter را بزنیم. در ادامه، هر کدام از این آپشن‌ها را شرح می‌دهیم.



تصویر ۵- صفحه‌ی اول برنامه‌ی nmtui

- آپشن Edit a connection

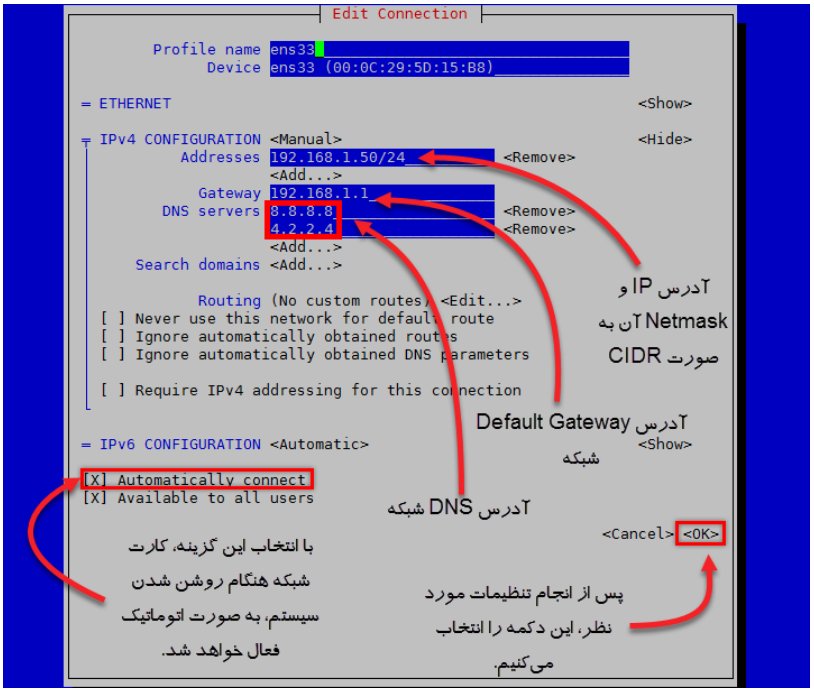
با انتخاب این گزینه، با تصویر زیر مواجه می‌شویم:



تصویر ۶- نتیجه‌ی انتخاب گزینه‌ی Edit a connection

در این صفحه، ما می‌توانیم تنظیمات مربوط به یک کارت شبکه جدید را اضافه کنیم، تنظیمات یک کارت شبکه را تغییر دهیم یا تنظیمات یک کارت شبکه را پاک کنیم.

برای تغییر تنظیمات یک کارت شبکه،کافی است کارت شبکه مورد نظر را انتخاب کرده و سپس دکمه‌ی Enter را بزنیم. به محض این کار، با صفحه‌ای نظیر تصویر ۷ مواجه می‌شویم:

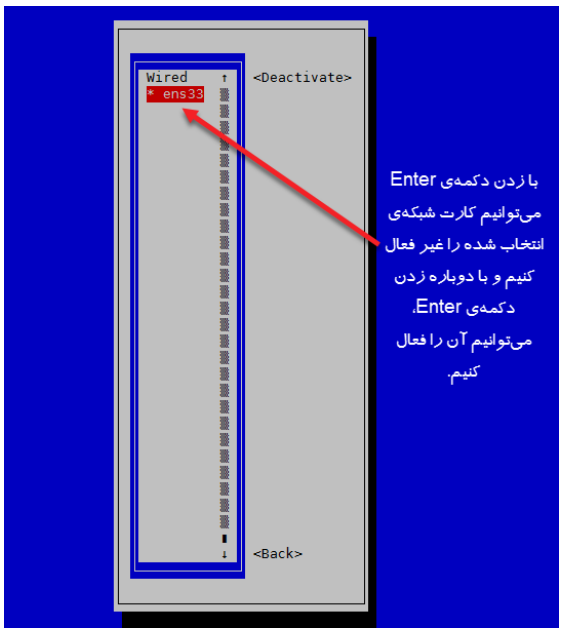


تصویر ۷- تغییر تنظیمات یک کارت شبکه در nmtui

در اینجا، می‌توانیم آدرس IP، Netmask، Gateway، DNS و... کارت شبکه انتخابی را تغییر دهیم. برای ذخیره‌ی تنظیمات، کافی است دکمه‌ی OK را انتخاب کرده و Enter را بزنیم. به محض زدن دکمه‌ی OK، به صفحه‌ی نمایش داده شده در تصویر ۶ باز می‌گردیم و در صورت نیاز می‌توانیم کارت شبکه‌ی دیگری را تنظیم کرده یا به صفحه‌ی اول باز گردیم.

- آپشن Activate a connection

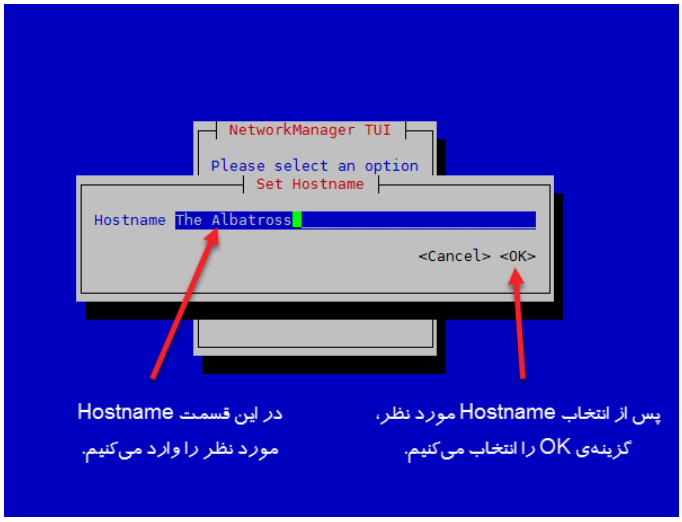
با انتخاب این گزینه، با نمایی نظیر تصویر ۸ مواجه می‌شویم. در اینجا ما می‌توانیم با زدن دکمه‌ی Enter، کارت شبکه‌ی انتخابی را فعال یا غیر فعال کنیم. پس از انجام این کار، کافی است دکمه‌ی Back را انتخاب کرده تا به صفحه‌ی اصلی `nmtui` باز گردیم.



تصویر ۸- فعال یا غیرفعال کردن یک کارت شبکه در nmtui

- آپشن Set system hostname

با انتخاب این گزینه، با نمایی نظیر تصویر ۹ مواجه می‌شویم. ما در اینجا می‌توانیم Hostname کنونی سیستم را مشاهده و در صورت نیاز آن را تغییر دهیم. توجه کنید که در صورت تغییر Hostname، باید سیستم را ریبوت کرده تا Hostname جدید بر روی سیستم قرار گیرد.



تصویر ۹- تغییر Hostname شبکه در nmtui

تنظیم شبکه با `nmcli`

این برنامه بر خلاف nmtui، هیچ ظاهر گرافیکالی ندارد و مانند یک برنامه‌ی کامندلاینی معمولی کار می‌کند. خوبی این برنامه در این است که علاوه بر امکان تغییر تنظیمات شبکه، می‌توانیم وضعیت شبکه را نیز به راحتی مشاهده کنیم.

برای مشاهده‌ی کلیه‌ی کارت‌های شبکه موجود بر روی سیستم، به صورت زیر از nmcli استفاده می‌کنیم:

```
[root@localhost ~]# nmcli dev status
DEVICE  TYPE      STATE      CONNECTION
ens33   ethernet  connected  ens33
lo      loopback  unmanaged  --
```

همانطور که می‌بینید، این دستور کارت‌های شبکه‌ی موجود در سیستم را به ما نشان می‌دهد. ستون اول خروجی این دستور، نام هر دستگاه را به ما نشان می‌دهد. ما در اینجا یک کارت شبکه و یک اینترفیس loopback داریم. اینترفیس loopback، یک کارت شبکه‌ی مجازی است که به برنامه‌های روی سیستم امکان می‌دهد که با یکدیگر ارتباط برقرار کنند. ستون دوم، نوع اتصال کارت شبکه را نشان می‌دهد. ستون سوم، وضعیت کارت شبکه را نشان داده و ستون آخر نام Connection را نشان می‌دهد.

برای مشاهده‌ی کانکشن‌هایی که در حال حاضر فعال هستند، دستور زیر را وارد می‌کنیم:

```
[root@localhost ~]# nmcli con show
NAME   UUID                                  TYPE      DEVICE
ens33  56002cf0-9e73-4617-8e54-3556bdbeb508  ethernet  ens33
```

همانطور که می‌بینید، در حال حاضر فقط یک کانکشن با نام ens33 از نوع ethernet، توسط کارت شبکه‌ی ens33 در سیستم ما فعال می‌باشد.

برای دریافت اطلاعات جزئی‌تر در مورد یک کانکشن، کافی است نام آن کانکشن را میان دو علامت ' قرار داده و آن را در انتهای دستور nmcli con show اضافه کنیم. یعنی:

```
[root@localhost ~]# nmcli con show 'ens33'
connection.id:                          ens33
connection.uuid:                        56002cf0-9e73-4617-8e54-3556bdbeb508
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              ens33
connection.autoconnect:                 yes
...
ipv4.method:                            manual
ipv4.dns:                               8.8.8.8,4.2.2.4
```

همانطور که می‌بینید، خروجی این دستور بسیار طولانی بوده و اطلاعات خیلی جزئی در مورد کانکشن ens33 به ما می‌دهد.

برای این که کاری کنیم که کارت شبکه ens33 یا کانکشن ens33، به صورت اتوماتیک هنگام روشن شدن سیستم فعال نشود، از دستور زیر استفاده می‌کنیم:

```
[root@localhost ~]# nmcli con mod 'ens33' connection.autoconnect no
```

می‌توانیم با استفاده از دستور زیر، صحت غیر فعال بودن کانکشن هنگام روشن شدن سیستم را بررسی کنیم:

```
[root@localhost ~]# grep 'ONBOOT' /etc/sysconfig/network-scripts/ifcfg-ens33
ONBOOT=no
```

اگر به خاطر داشته باشید، متغیر ONBOOT در فایل تنظیمات کارت شبکه، نشان دهنده‌ی فعال یا غیر فعال بودن یک کانکشن هنگام روشن شدن سیستم بود. این متغیر، اکنون مقدار no را دارد که نشان می‌دهد دستور nmcli کار خود را به درستی انجام داده است.

اگر بخواهیم کانکشن ens33 به صورت اتوماتیک هنگام روشن شدن سیستم فعال شود، کافی است به جای no در دستور بالا، از عبارت yes استفاده کنیم. یعنی:

```
[root@localhost ~]# nmcli con mod 'ens33' connection.autoconnect yes
```

اگر به خاطر داشته باشید، ما می‌توانستیم به صورت اتوماتیک (dhcp) یا به صورت دستی (static) به یک کارت شبکه، آدرس IP بدهیم. ما می‌توانیم چگونگی تنظیم IP بر روی یک کانکشن را نیز با nmcli تغییر دهیم. بیایید چگونگی تنظیم IP را تبدیل به dhcp کنیم:

```
[root@localhost ~]# nmcli con mod 'ens33' ipv4.method auto
[root@localhost ~]# grep 'BOOTPROTO' /etc/sysconfig/network-scripts/ifcfg-ens33
BOOTPROTO=dhcp
```

همانطور که می‌بینید، اکنون BOOTPROTO مقدار dhcp را دارد. این یعنی که تنظیمات شبکه باید به صورت اتوماتیک از یک سرور DHCP دریافت شود.

برای این که چگونگی تنظیم IP را تبدیل به حالت دستی کنیم:

```
[root@localhost ~]# nmcli con mod 'ens33' ipv4.method static
[root@localhost ~]# grep 'BOOTPROTO' /etc/sysconfig/network-scripts/ifcfg-ens33
BOOTPROTO=none
```

همانطور که می‌بینید، اکنون BOOTPROTO مقدار none را دارد که همان مفهوم static را دارد.

برای تنظیم آدرس IP و Netmask روی یک کارت شبکه، به صورت زیر از nmcli استفاده می‌کنیم. توجه داشته باشید که در این حالت، ما Netmask را به صورت CIDR وارد می‌کنیم:

```
[root@localhost ~]# nmcli con mod 'ens33' ipv4.address 192.168.1.60/24
```

حال بیایید از صحت تنظیم آدرس IP و Netmask دلخواه اطمینان حاصل کنیم:

```
[root@localhost ~]# grep 'IPADDR' /etc/sysconfig/network-scripts/ifcfg-ens33
IPADDR=192.168.1.60
[root@localhost ~]# grep 'PREFIX' /etc/sysconfig/network-scripts/ifcfg-ens33
PREFIX=24
```

همانطور که می‌بینید، آدرس IP مورد نظر و همچنین Netmask مورد نظر ما روی کارت شبکه‌ی ens33 تنظیم شده است. توجه کنید که به دلیل استفاده از روش CIDR برای نوشتن Netmask، در فال تنظیمات کارت شبکه‌ی ens33، متغیر PREFIX را مشاهده می‌کنیم.

برای تنظیم Default Gateway بر روی یک کارت شبکه به صورت زیر از دستور nmcli استفاده می‌کنیم:

```
[root@localhost ~]# nmcli con mod 'ens33' ipv4.gateway 192.168.1.1
```

بیایید از صحت تنظیم آدرس Default Gateway دلخواه اطمینان حاصل کنیم:

```
[root@localhost ~]# grep 'GATEWAY' /etc/sysconfig/network-scripts/ifcfg-ens33
GATEWAY=192.168.1.1
```

برای تنظیم DNS روی کارت شبکه، به صورت زیر از nmcli استفاده می‌کنیم:

```
[root@localhost ~]# nmcli con mod 'ens33' ipv4.dns 4.2.2.4
```

بیایید از صحت تنظیم این DNS اطمینان حاصل کنیم:

```
[root@localhost ~]# grep 'DNS' /etc/sysconfig/network-scripts/ifcfg-ens33
DNS1=4.2.2.4
```

برای اضافه کردن یک DNS دیگر (در کنار DNS کنونی)، به صورت زیر عمل می‌کنیم:

```
[root@localhost ~]# nmcli con mod 'ens33' +ipv4.dns 8.8.8.8
```

بیایید از صحت تنظیم این DNS جدید اطمینان حاصل کنیم:

```
[root@localhost ~]# grep 'DNS' /etc/sysconfig/network-scripts/ifcfg-ens33
DNS1=4.2.2.4
DNS2=8.8.8.8
```

دستور nmcli قابلیت‌های بسیار پیشرفته‌تری نیز دارد، اما ما بیشتر از این در مورد این دستور صحبت نخواهیم کرد. پیشنهاد می‌کنیم که در صورت تمایل، manpage این دستور را مطالعه کنید.

استفاده از دستور *ip*

یکی دیگر از برنامه‌های کامندلاینی که می‌توانیم از آن برای تنظیم و همچنین مشاهده‌ی وضعیت شبکه استفاده کنیم، پکیج iproute2 می‌باشد. ما در این بخش فقط به قابلیت این دستور در مشاهده‌ی تنظیمات شبکه می‌پردازیم. در واقع، ما تا به اینجا چندین بار از این پکیج استفاده کرده‌ایم. اگر به خاطر داشته باشید، ما برای مشاهده‌ی کارت‌های شبکه‌ی موجود در سیستم و آدرس IP آنها، از دستور زیر استفاده می‌کردیم:

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
qlen 1000
    link/ether 00:0c:29:5d:15:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.50/24 brd 192.168.1.255 scope global noprefixroute ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::e836:9b5f:6eb:67/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

هنگام استفاده از این دستور، ما داشتیم از پکیج iproute2 استفاده می‌کردیم. حال بیایید با برخی دیگر از آپشن‌های موجود این دستور آشنا شویم.

اگر فقط بخواهیم نام کارت شبکه‌های موجود در سیستم به ما نشان داده شود، به شکل زیر از دستور ip استفاده می‌کنیم:

```
[root@localhost ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP mode DEFAULT group default qlen 1000
    link/ether 00:0c:29:5d:15:b8 brd ff:ff:ff:ff:ff:ff
```

همانطور که می‌بینید، این دستور فقط اطلاعاتی کلی در مورد کارت‌های شبکه‌ی موجود در سیستم به ما نشان می‌دهد.

اگر بخواهیم اطلاعاتی در مورد تعداد بایت‌های دریافت شده و ارسال شده توسط یک کارت شبکه را به دست آوریم، به صورت زیر از دستور ip استفاده می‌کنیم:

```
[root@localhost ~]# ip -s link show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP mode DEFAULT group default qlen 1000
    link/ether 00:0c:29:5d:15:b8 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    69355      493      0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    30071      294      0       0       0       0
```

همانطور که می‌بینید، برای مشاهده‌ی تعداد بایت‌های دریافتی (RX) و ارسالی (TX)، ابتدا دستور ip را وارد کرده، سپس آپشن s- را به آن داده و پس از آن، از عبارت link و show استفاده و در نهایت نام کارت شبکه‌ای که می‌خواهیم این اطلاعات را در مورد آن دریافت کنیم را وارد می‌کنیم.

برای مشاهده‌ی جدول روتینگ، از دستور زیر استفاده می‌کنیم:

```
[root@localhost ~]# ip r
default via 192.168.1.1 dev ens33 proto static metric 100
192.168.1.0/24 dev ens33 proto kernel scope link src 192.168.1.50 metric 100
```

همانطور که می‌بینید، خروجی به ما می‌گوید که Default Gateway که ما از طریق آن می‌توانیم به سایر شبکه‌ها نظیر اینترنت متصل شویم، با ارسال اطلاعات به آدرس 192.168.1.1 از روی اینترفیس ens33 در دسترس می‌باشد.

اگر به خاطر داشته باشید، ما قبلا برای خاموش یا روشن کردن یک اینترفیس (فعال یا غیرفعال کردن اینترفیس)، از دستور ifup و ifdown استفاده می‌کردیم. اما ما می‌توانیم از دستور ip نیز برای انجام این کار استفاده کنیم.

برای خاموش یا غیرفعال کردن یک اینترفیس، به صورت زیر از دستور ip استفاده می‌کنیم:

```
[root@localhost ~]# ip link set ens33 down
```

همانطور که می‌بینید، ما دستور ip را با آپشن link وارد کرده، سپس عبارت set را وارد و پس از آن،

نام کارت شبکه‌ای که می‌خواهیم آن را خاموش یا غیرفعال کنیم را وارد کردیم. بلافاصله پس از وارد کردن نام کارت شبکه‌ی مورد نظر، از عبارت down، به معنای خاموش یا غیر فعال کردن کارت شبکه، استفاده کردیم.

برای روشن یا فعال کردن یک اینترفیس، به صورت زیر از دستور ip استفاده می‌کنیم.

```
[root@localhost ~]# ip link set ens33 up
```

این دستور بسیار شبیه به دستور مورد استفاده برای خاموش یا غیرفعال کردن یک کارت شبکه می‌باشد، پس به توضیح بیشتر آن نمی‌پردازیم.

دستور ip قابلیت‌های بسیار بیشتری نیز دارد که توضیح آنها از حوصله‌ی ما خارج است. پیشنهاد می‌شود که در صورت تمایل، manpage این دستور را مطالعه کنید.