

Advanced Editing Using sed



Andrew Mallett

Author and Trainer

@theurbanpenguin | www.theurbanpenguin.com



Overview



Accessing sed Features

- Using sed with scripts
- More complex expressions using sed files
- Remote edits with sed



Adding Shebang to Scripts



```
#!/bin/bash
for f in ~/test/sed-and-awk/scripts/*.sh; do
    firstline=$(sed -n '1p' $f)
    if [[ $firstline != "#!"* ]]; then
        echo "Adding shebang to $f"
        sed -i '1i #!/bin/bash' $f
    fi
done
```

Many Script : Many Inconsistencies

With many scripts created over many years; some have the shebang set, other don't. This can cause issues if the bash shell is not used



Demo



Writing shell scripts

The GIT repo has many scripts

- Some with shebangs and some without
- We need to ensure they all have the shebang but only the one shebang line





Using sed Files



```
$ sed '/^$/d;/^#/d' sshd_config
$ vim simple.sed
/^$/d
/^#/d

$ sed -f simple.sed sshd_config
```

Multiple Expressions

From the command line we can group expressions using the semi-colon. It may be easier by creating a sed file with each expression on a new line



The SSHD Configuration: Not Dropping Idle Clients

ClientAliveInterval

How often a keep alive packet is sent to clients

ClientAliveCountMax

How many intervals can pass without a client response




```
#!/bin/sed -Ef #Shebang and reminding that we need ERE
/ClientAliveInterval/ { #Search for this string
    s/^(ClientAliveInterval).*$ /\1 60/ #Test if starts with key and ensure set to 60
    t count #If substitution worked goto label count
    s/.* /ClientAliveInterval 60/ #If didn't start with key then replace line with setting
    t count #Then got to label
}

:count
/ClientAliveCountMax/ { #Search for this string
...
$ sed -Ef ssh_client_alive.sed sshd_config
```

More Complex Edits

A configuration file may have the setting you want, have it but with the wrong setting or not have the required line at all. Using sed files we can write much more complex scripts. Here we want this setting in our file: **ClientAliveInterval 60** before moving to ClientAliveMaxCount



Demo



Managing complexity with sed files

Learning sed files

- Understanding how to group expressions using sed files
- Creating backups



Demo



Extending sed files

- Branching in sed files to cater with differences
- Allowing us to cater for differing conditions





Working Remotely



```
$ echo "127.0.0.1" > server.txt
$ while read server ; do \
    echo "Starting work on $server"
    ssh -n -C $server "sudo sed -Ei ' /^(#|$)/d' /etc/ssh/sshd_config"
    echo "Finished work on $server"
done < server.txt
```

Remote SSH Commands

Using `ssh` and `-C` we can execute commands remotely without the need of an interactive session. The option `-n` is used to redirect `stdin` to `/dev/null`, allowing remote execution of commands on more than one server



Demo



Working Remotely

- Create a list of servers
- Remotely connect via SSH -C
- By doing this we can streamline administrative tasks
- We work with two systems so you can see that this does really work; it is up to you if you want to use more than one system



Summary



Scripting with sed

- Add shebang if it does not exist
- Using sed files
 - multiple expressions
 - sed -f
 - branching and labels
- Remote edit via SSH using sed
 - -n
 - -C

Up Next:

Filtering Text Files Using awk

