

You have 2 free member-only stories left this month.
[Sign up for Medium and get an extra one](#)



Prashant Lakhera

[Follow](#)

Apr 2, 2019 · 4 min read ·

Member-only

[Listen](#)



100 Days of DevOps — Day 51-Introduction to Bash Scripting

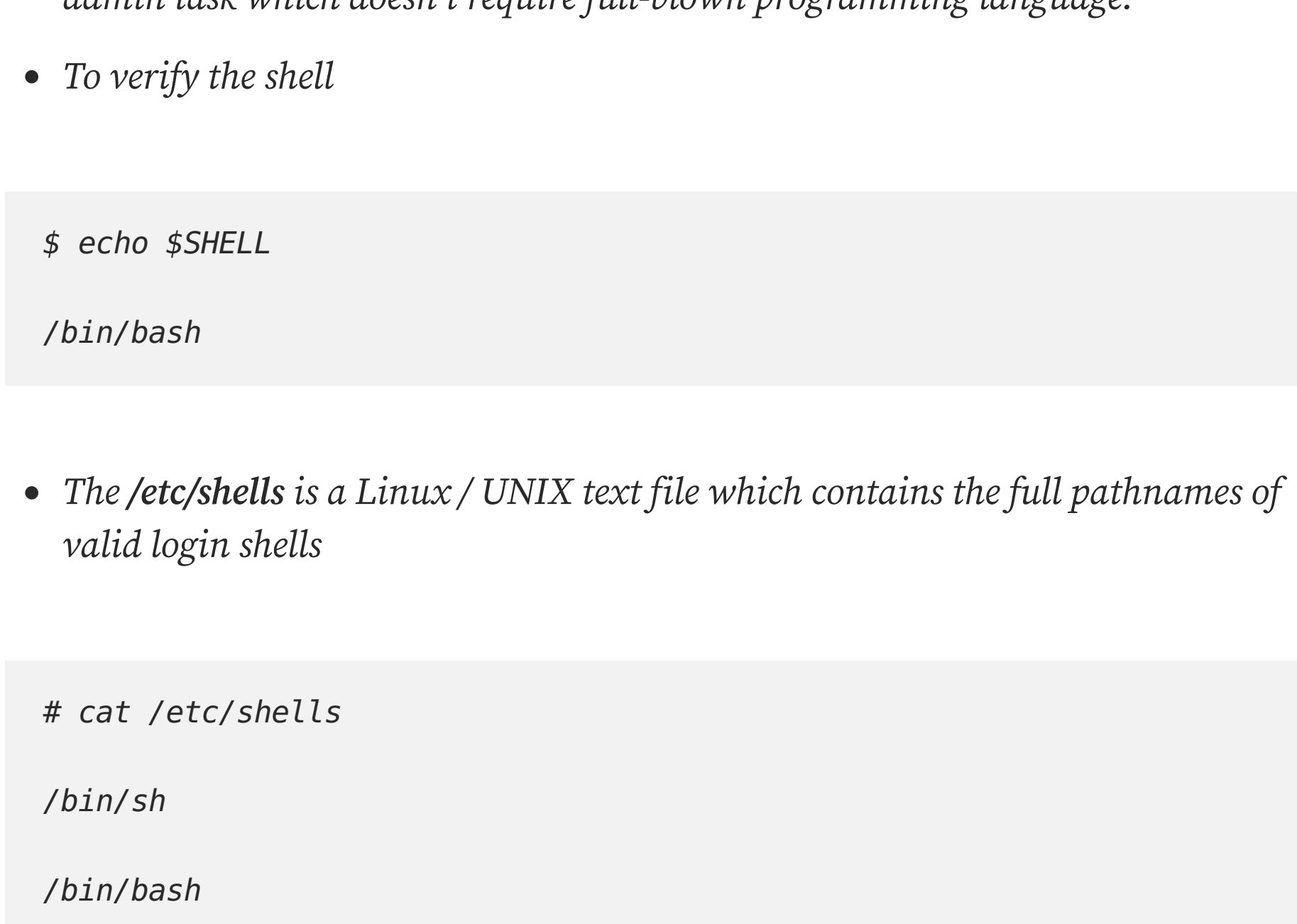
Welcome to Day 51 of 100 Days of DevOps, Focus for today is Bash Scripting.

Wow, we have completed 50 days and in the last 50 days hopefully, you learned something from it. I would like to thank all of you who read and give your valuable feedback. As you can see from the last 50 days, the main emphasis is AWS and terraform, for the next day it will be a mixed bag of other DevOps Tools(eg: Linux, GIT, Docker, Shell, Python...)

Today I am going to start with a really basic but important thing Shell.

What is Bash?

Bash is the shell, or command language interpreter, for the GNU operating system.



- It acts as an insulating layer between User and Kernel but at the same time its a fairly powerful programming language
- The shell script is particularly helpful for common repetitive tasks or for system admin task which doesn't require full-blown programming language.
- To verify the shell

```
$ echo $SHELL

/bin/bash
```

- The `/etc/shells` is a Linux / UNIX text file which contains the full pathnames of valid login shells

```
# cat /etc/shells

/bin/sh

/bin/bash

/usr/bin/sh

/usr/bin/bash
```

- Now once you logged in to the system, you will see a prompt like this

For root user

```
[root@ip-172-31-31-68 ~]#
```

For normal user

```
[centos@ip-172-31-31-68 ~]$
```

- The first field in it, indicate username(root)
- The second field indicates hostname(ip-172-31-31-68)
- The third field indicates the current working directory(~ or root home directory in this case or centos user home directory)
- # prompt** is for the root user and **\$ prompt** is for normal user

Default System File which Bash Relies on

- `/etc/profile` System-wide configuration for Bash

Some important configuration

```
USER=`/usr/bin/id -un`      <-- Username

LOGNAME=$USER

MAIL="/var/spool/mail/$USER"

HOSTNAME=`/usr/bin/hostname 2>/dev/null` <--Hostname

HISTSIZE=1000 <--History size
```

- `/etc/bashrc` System wide functions and aliases go to bashrc

```
$PS1 = printf "\033\0;%s@\%s:\%s\007" "${USER}" "${HOSTNAME%.*}"
"${PWD}/${HOME}/~}"

# This hold your bash prompt as we seen above

[root@ip-172-31-31-68 ~]#

and it's completly customizable
```

- If you go to the user home directory

```
$ cd /home/centos/
```

and run `ls -la`

```
$ ls -la

total 20

drwx-----. 3 centos centos 111 Apr  2 16:43 .

drwxr-xr-x. 3 root  root   20 Mar 28 03:38 ..

-rw-----. 1 centos centos   8 Mar 28 03:45 .bash_history

-rw-r--r--. 1 centos centos  18 Oct 30 17:07 .bash_logout

-rw-r--r--. 1 centos centos 193 Oct 30 17:07 .bash_profile

-rw-r--r--. 1 centos centos 231 Oct 30 17:07 .bashrc

-rw----- 1 centos centos  43 Apr  2 16:43 .lessht

drwx-----. 2 centos centos  29 Mar 28 03:38 .ssh
```

- You will see files similar to what we discussed above with the difference of dot(.) in front of it.
- In Linux analogy dot(.) means the hidden file
- These files are per user specific and it overrides the system-wide configuration(eg: Any settings in `.bash_profile` for a particular user will get more preference as compared to `/etc/profile`)

When we create a user, all these files copied from `/etc/skel` to the user home directory

```
$ ls -la /etc/skel/

total 24

drwxr-xr-x. 2 root root   62 Apr 11 2018 .

drwxr-xr-x. 77 root root 8192 Mar 28 03:38 ..

-rw-r--r--. 1 root root   18 Oct 30 17:07 .bash_logout

-rw-r--r--. 1 root root  193 Oct 30 17:07 .bash_profile

-rw-r--r--. 1 root root  231 Oct 30 17:07 .bashrc
```

To verify it

```
[root@ip-172-31-31-68 ~]# useradd testuser

[root@ip-172-31-31-68 ~]# ls -la /home/testuser/

total 12

drwx----- 2 testuser testuser  62 Apr  2 19:48 .

drwxr-xr-x. 4 root  root   36 Apr  2 19:48 ..

-rw-r--r-- 1 testuser testuser  18 Oct 30 17:07 .bash_logout

-rw-r--r-- 1 testuser testuser 193 Oct 30 17:07 .bash_profile

-rw-r--r-- 1 testuser testuser 231 Oct 30 17:07 .bashrc
```

If we try to check this user in `/etc/passwd` file(file used to keep track of every user)

```
# grep testuser /etc/passwd

testuser:x:1001:1001::/home/testuser:/bin/bash
```

- As we can see default shell for the user is bash(`/bin/bash`)
- In simplest word shell scripting is nothing but a list of system command stored in the file.

Sha-bang

```
#!/bin/bash
```

- The sha-bang at the top of the script tells our system that this file is a set of commands to be fed to the command interpreter indicated.
- `#!` is actually a two-byte magic number, a special marker that designates a file type or in this case executable shell script(for more info check **man magic**)
- Immediately following the sha-bang is a path name, this is the path to the program that interprets the commands in the script, whether its a shell, a programming language.
- Command interpreter then executes the commands in the script, starting at the top(line following the sha-bang and ignoring comments)

Looking forward from you guys to join this journey and spend a minimum an hour every day for the next 100 days on DevOps work and post your progress using any of the below medium.

- Twitter: [@100daysofdevops](#) OR [@lakhera2015](#)
- Facebook: <https://www.facebook.com/groups/795382630808645/>
- Medium: <https://medium.com/@devopslearning>
- Slack: <https://devops-myworld.slack.com/messages/CF41EFG49/>
- GitHub Link:<https://github.com/100daysofdevops>

Reference

100 Days of DevOps — Day 0

D-day is just one day away and finally, this is a continuation of the post(I posted a month earlier)

medium.com

21

Get an email whenever Prashant Lakhera publishes.

Subscribe

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

More from Prashant Lakhera

AWS Community Builder, Ex-Redhat, Author, Blogger, YouTuber, RHCA, RHCD, RHCE, Docker Certified, 4XAWS, CCNA, MCP, Certified Jenkins, Terraform Certified, 1XGCP

Apr 1, 2019

Member-only

100 Days of DevOps — Day 50-Introduction to Route53 Failover

Welcome to Day 50 of 100 Days of DevOps, Focus for today is Route53 Failover On Day 49, I talked about Route53, Let extend that concept...

AWS

4 min read

Share your ideas with millions of readers.

Write on Medium

Mar 31, 2019

Member-only

100 Days of DevOps — Day 49-Introduction to Route53

Welcome to Day 49 of 100 Days of DevOps, Focus for today is Route53 What is AWS Route53? Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 5...

AWS

4 min read

Mar 30, 2019

Member-only

100 Days of DevOps — Day 48- Threat detection and mitigation at AWS

Welcome to Day 48 of 100 Days of DevOps, Focus for today is Threat detection and mitigation at AWS This 27th Wednesday, I got the chance...

AWS

6 min read

Mar 29, 2019

100 Days of DevOps — Day 47-Introduction to Amazon Elastic File System (EFS)

Welcome to Day 47 of 100 Days of DevOps, Focus for today is Amazon Elastic File System (EFS) What is Amazon EFS? Amazon EFS provides...

AWS

3 min read

Mar 28, 2019

100 Days of DevOps — Day 46-Introduction to Amazon Glacier

Welcome to Day 46 of 100 Days of DevOps, Focus for today is Amazon Glacier What is Amazon Glacier? Data Archiving Solution It's designed...

AWS

3 min read

Love podcasts or audiobooks? Learn on the go with our new app.

Try Knowable

Recommended from Medium

nick3499

Python: Generate Color Design

Peter Aiello in The Startup

Adding Aliases to Shell in Catalina (Mac OS)

McKenna Boen in The Helium Blog

Connected Fresh Selects The People's Network to Keep Food & Beverages Safe

Zubair Ahmed in Grizzlybit

Setup OpenStreetMap Tile Server on Ubuntu 20.x

Danny Boahen in ETHA

ETHA Lend— Mainnet Soft Launch

Oren Oichman

Enable Linux Serial Console login

Bruce Lawson

Setting up Prince on EC2 with PHP

Uday Hiwarale in JsPoint

Manage how your site looks on search engines using Schema.org

Medium

About Help Terms Privacy