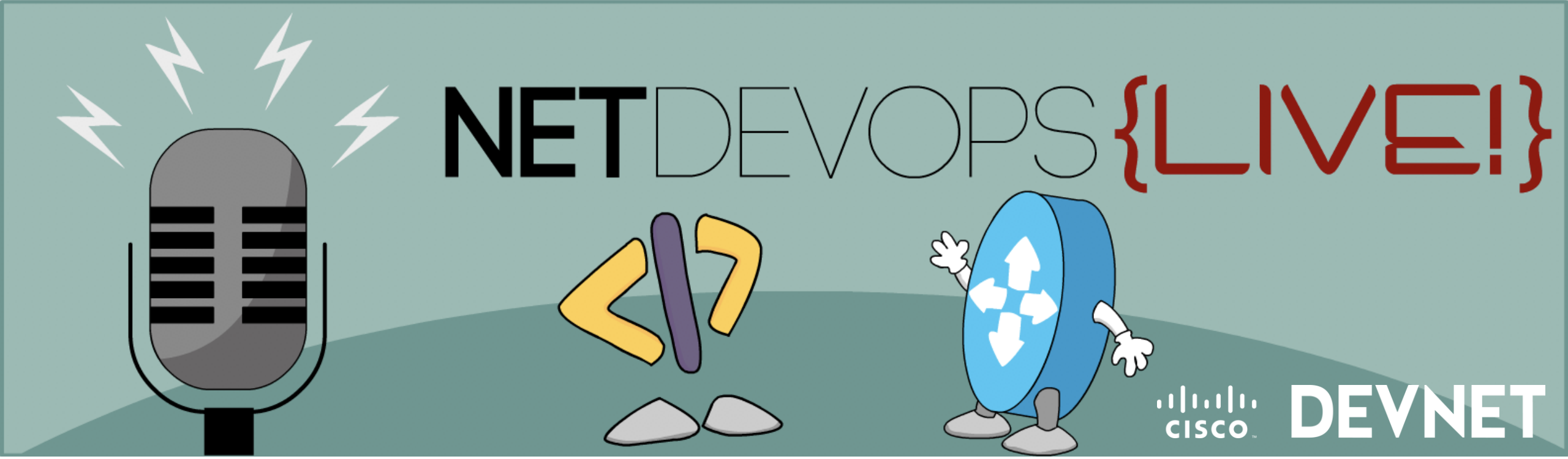


Laptop Tips and Tricks for the NetDevOps Engineer

Hank Preston, ccie 38336 R/S

NetDevOps Engineer, DevNet Sandbox

Twitter: @hfpreston

Season 2, Talk 5

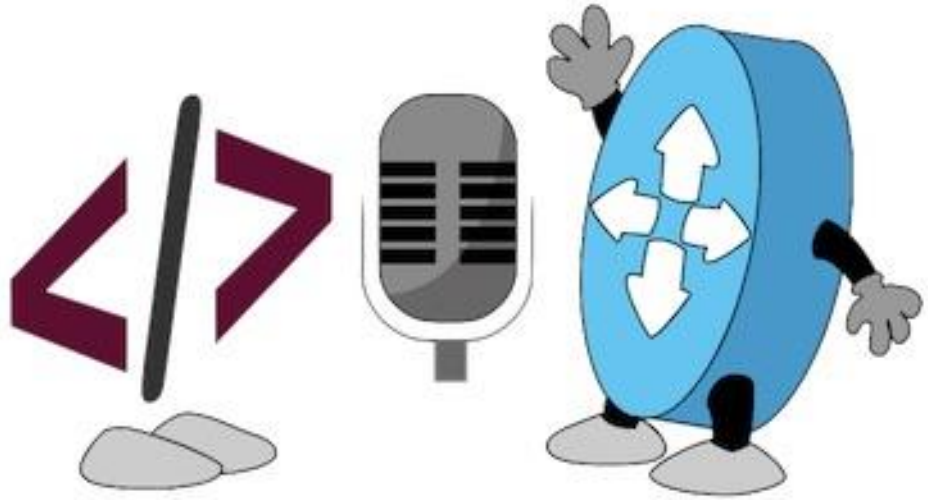
https://developer.cisco.com/netdevops/live

http://cs.co/ndl

Help us track NetDevOps Live Interest!

What are we going to talk about?

- Don’t be bashful, Love thy terminal
- Shhh… SSH like a pro
- Oscar worthy bash scripts!
  (and other bash-y stuff)
- Make your Apps Dance



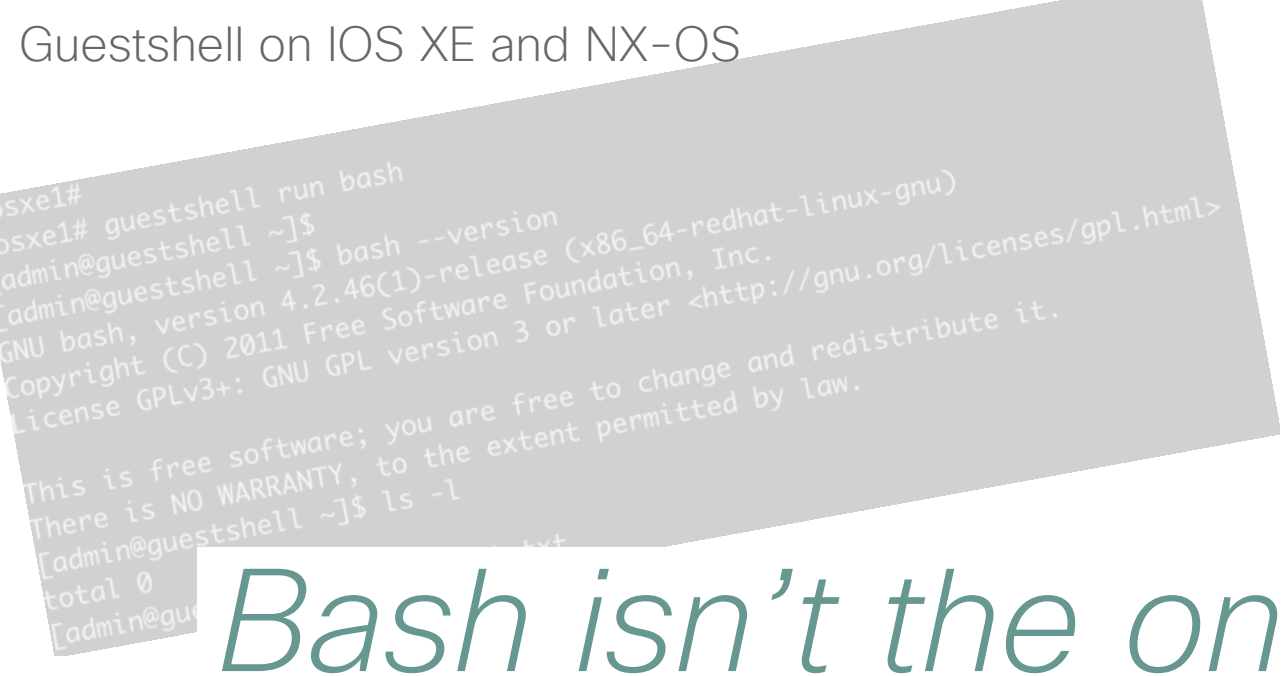
Before we start...

- ✓ What follows are all personal opinions, some tips might not be for you
- ✓ Some tips are very basic, and you might already know it... For others they'll be new
- ✓ I learn new stuff all the time... there are likely great tips I don't know, share them!

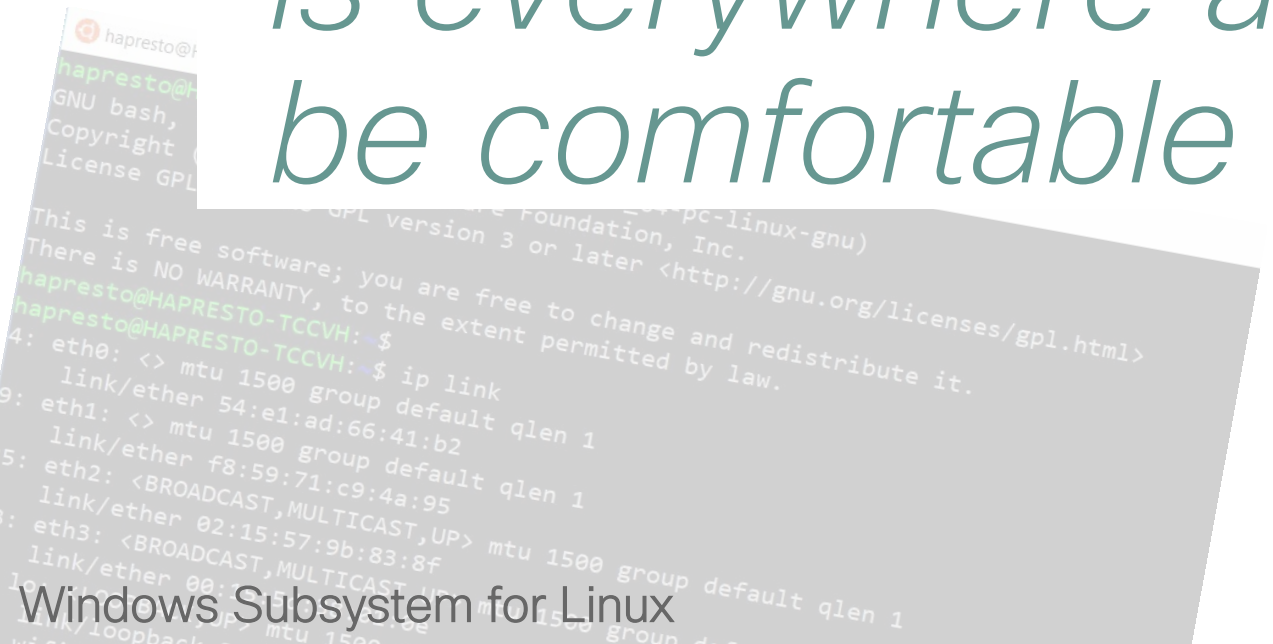
Don’t be bashful, Love thy terminal

Guestshell on IOS XE and NX-OS

git-bash on Windows



Bash isn't the only terminal... but it is everywhere and you need to be comfortable with it



Windows Subsystem for Linux

Direct bash access on NX-OS

May the “source” be with you

- Ever wonder what “source” does in bash?
- Given a bash script, what’s the difference between running and “sourcing”

Answer

- Running a script generates a new shell instance
- Sourcing runs the script in your current shell
- So what?
  - Environment variables, directory changes

File: env_demo1.sh

```
#! /bin/bash

echo "This is a sample bash script."
echo "I am going to set up an env var called 'DEMO_VAR1'"

export DEMO_VAR1="New ENV VAR!"

echo "Okay, see if it worked with 'echo \$DEMO_VAR1'"
echo ""
```

```
# Option 1
$ chmod +x env_demo1.sh
$ ./env_demo1.sh
$ echo $DEMO_VAR1


# Option 2
$ source env_demo1.sh
$ echo $DEMO_VAR1

# Which worked?
```

dev_tips/env_demo1.sh

Aliases – for more than restoring ”wr mem” to NX-OS

- Create shortcuts for common, long, or complicated commands
- Git commands
- File management
- VPN Connections
- Anything

```
sbx-n9kv-ao# wr me
             ^
% Invalid command at '^' marker.

sbx-n9kv-ao# conf t
sbx-n9kv-ao(config)# cli alias name wr copy running-config startup-config
sbx-n9kv-ao(config)# end

sbx-n9kv-ao# wr
[########################################] 100%
Copy complete, now saving to disk (please wait)...
Copy complete.
```

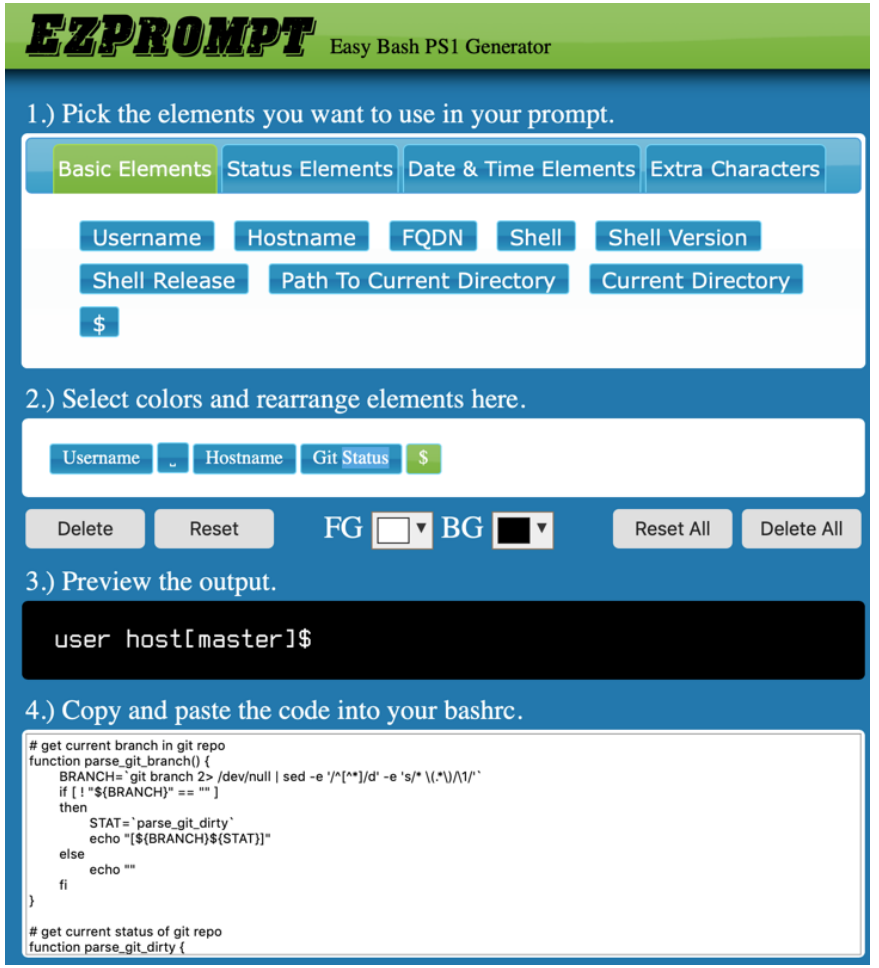
Bash Aliases

```
# A few handy bash aliases

# Display file list with colors, sizes, and in list format
alias ll='ls -FGlAhp'

# Handy aliases to find resource hogs
alias memHogsTop='top -l 1 -o rsize | head -20'
alias memHogsPs='ps wwaxm -o pid,stat,vsize,rss,time,command | head -10'
alias cpu_hogs='ps wwaxr -o pid,stat,%cpu,time,command | head -10'

# Some networking aliases for macOS
alias flushDNS='dscacheutil -flushcache'
alias openPorts='sudo lsof -i | grep LISTEN'

# Remove all exited/stopped docker containers
alias docker_rm='docker rm -v $(docker ps -aq -f status=exited)'
```

dev_tips/alias_examples.sh

The bash prompt

- Environment Variable PS1
- Series of codes and functions generate what you see
- You can make it whatever you want
- **export PS1=\W\ $**
  - Working directory and dollar sign
- Adding some git info
  - https://github.com/magicmonty/bash-git-prompt



http://ezprompt.net

.bash_profile (and .bashrc on Linux)

- Auto-load and customize your bash environment
- Terminal settings, prompts, colors, aliases, just about anything
- Many many examples of customizations online

```
#   ----------------------------
#   6.  NETWORKING
#   ----------------------------

alias myip='curl ip.appspot.com'                    # myip:         Public facing IP Address
alias netCons='lsof -i'                             # netCons:      Show all open TCP/IP sockets
alias flushDNS='dscacheutil -flushcache'            # flushDNS:     Flush out the DNS Cache
alias lsock='sudo /usr/sbin/lsof -i -P'             # lsock:        Display open sockets
alias lsockU='sudo /usr/sbin/lsof -nP | grep UDP'   # lsockU:       Display only open UDP sockets
alias lsockT='sudo /usr/sbin/lsof -nP | grep TCP'   # lsockT:       Display only open TCP sockets
alias ipInfo0='ipconfig getpacket en0'              # ipInfo0:      Get info on connections for en0
alias ipInfo1='ipconfig getpacket en1'              # ipInfo1:      Get info on connections for en1
alias openPorts='sudo lsof -i | grep LISTEN'        # openPorts:    All listening connections
alias showBlocked='sudo ipfw list'                  # showBlocked:  All ipfw rules inc/ blocked IPs

#   ii:  display useful host related informaton
#   ----------------------------------------------------------------------
    ii() {
        echo -e "\nYou are logged on ${RED}$HOST"
        echo -e "\nAdditionnal information:$NC " ; uname -a
        echo -e "\n${RED}Users logged on:$NC " ; w -h
        echo -e "\n${RED}Current date :$NC " ; date
        echo -e "\n${RED}Machine stats :$NC " ; uptime
        echo -e "\n${RED}Current network location :$NC " ; scselect
        echo -e "\n${RED}Public facing IP Address :$NC " ;myip
        #echo -e "\n${RED}DNS Configuration:$NC " ; scutil --dns
        echo
    }
```

https://natelandau.com/my-mac-osx-bash_profile/

Shhh... SSH like a Pro

Authentication Power User – Use the Cert Hank!

- Passwords are so 1980s…
- Let's see how to use certificate based authentication

```
[developer@devbox ~]$ssh-keygen

Generating public/private rsa key pair.
Your identification has been saved in
/home/developer/.ssh/id_rsa.
Your public key has been saved in
/home/developer/.ssh/id_rsa.pub.

[developer@devbox ~]$ssh-copy-id
developer@10.10.20.20
Number of key(s) added: 1

Now try logging into the machine, with:
       "ssh 'developer@10.10.20.20'"
```

SSH Certificate Notes

- Create a key pair
  - ssh-keygen
  - Default location for keys: ~/.ssh
- Copy key to another server
  - ssh-copy-id
  - Adds to ~/.ssh/authorized_keys
- Linux SSH Server Config File
  - /etc/ssh/sshd_config

- IOS SSH Auth
  - ip ssh pubkey-chain
    username admin
      key-string
        STRING
        STRING
  - Bash command to wrap key to length
    - fold –w 48 .ssh/id_rsa.pub
- NX-OS SSH Auth
  - username admin sshkey KEY

~/.ssh/config – SSH Host Aliases and Properties

- SSH “aliases” for Hosts
  - Make it easier to connect to common devices
- Enable non-standard ciphers and algorithms
- Wild Cards for settings

```
Host sbx-iosxe
  HostName ios-xe-mgmt.cisco.com
  User root
  Port 8181
  IdentityFile ~/.ssh/id_rsa
Host sbx-nxos
  HostName sbx-nxos-mgmt.cisco.com
  User admin
  Port 8181
Host OldSwitch
  HostName 10.10.2.1
  Ciphers aes128-cbc
  KexAlgorithms +diffie-hellman-group1-sha1
```

~/.ssh/config – Those pesky changing lab certs…

- How to handle hosts where you fully expect keys to change
- You can use wildcards in names

```
Host 172.20.*.*
  StrictHostKeyChecking no
  UserKnownHostsFile=/dev/null
```

Oscar worthy bash scripts!
(and other bash-y stuff)

Curl – REST APIs and Web Calls without Postman

- Check if service is up
  - Status Code
- Get some data to process
- Get token or ID for another request

```
$ curl -sk \
-u 'root:D_Vay!_10&' \
-H "Accept: application/yang-data+json" \
"https://10.10.20.48/restconf/data/ietf-interfaces:interfaces/interface=GigabitEthernet1" \
| python -c 'import sys, json;
print(
  json.load(sys.stdin)["ietf-interfaces:interface"]["ietf-ip:ipv4"]["address"][0]["ip"]
  )'

10.10.20.48
```

dev_tips/curl_examples.sh

Bash Script Example 1

- Using curl in bash script
- Creating variable based on output
- Conditional processing based on value

```
status_code=$(curl -ks \
-w "%{http_code}" \
-o /dev/null \
-u 'root:D_Vay!_10&' \
-H "Accept: application/yang-data+json" \
"https://10.10.10.48/restconf/data/ietf-
interfaces:interfaces"
)

if [ $status_code -eq 200 ]
then
  echo "✅ Yep, it's up and returning a 200. "
else
  echo "❌ Nope, it returned a $status_code”
fi
```

devtips/bash_script_ex1.sh

Bash Script Example 2

- Creating a bash function
- Using while loop to wait until ready

```
# Function to get the status code
get_status_code() {
code=$(curl -ks \
-w "%{http_code}" \
-o /dev/null \
-u 'root:D_Vay!_10&' \
-H "Accept: application/yang-data+json" \
"https://10.10.10.10/restconf/data/ietf-
interfaces:interfaces"
)
echo $code
}

# Loop until status code is 200, or 10 tries
while [ $status_code -ne 200 -a $COUNTER -lt 10 ]; do
sleep 30
status_code=$(get_status_code)
COUNTER=$(($COUNTER+1))
done
```

devtips/bash_script_ex2.sh

Bash Script Example 3

- Asking user for input
- Checking exit code with special **$?**

```
echo "What interface would you like to check?"
echo "For example 'GigabitEthernet1'"
read interface_name

ip=$(curl -sk \
-u 'root:D_Vay!_10&' \
-H "Accept: application/yang-data+json" \
"https://10.10.10.10/restconf/data/ietf-
interfaces:interfaces/interface=$interface_name" \
| python -c 'import sys, json;
print(
  json.load(sys.stdin)["ietf-
interfaces:interface"]["ietf-
ip:ipv4"]["address"][0]["ip"]
  )'
)

# Make sure the last command worked
if [ $? -ne 0 ]
then
  echo "Something went wrong, exiting."
  exit 1
fi
```

devtips/bash_script_ex3.sh

Bonus Bash Example

- Processing all files in a directory
- Here **ncs_load** is ran for EVERY file in the **device_configs/** directory

```
#! /bin/bash

for f in device_configs/*
do
  echo "Processing file $f"
  ncs_load -u admin -m -l $f
done
```

Make things easier with Makefiles

- Most often used for code compilation workflows
- But can make any workflow easier
- Simplify any repeated task for yourself, or others working on your project
- **make dev, make clean, make whatever**

```
dev: virl nso virl-generate nso-sync-from

clean: nso-clean virl-clean genie-clean

virl:
@echo "Starting the VIRL topology."
-@virl up --provision virlfiles/2-ios-router

virl-clean:
@echo "Tearing down VIRL topology."
-@virl down
-@rm -Rf default_testbed.yaml

nso:
@echo "Setting up NSO dev instance."
@ncs-setup --package cisco-ios --dest .
@ncs

nso-clean:
@echo "Destroying NSO dev instance."
-@ncs --stop
```

devtips/Makefile

Makefile Notes

- - before command to ignore error
- @ before command to NOT print command to screen
- Creates it's own shell, set any environment variables within Makefile
- Organize and group commands together into common workflows

Make your Apps Dance

Application Package Managers

- Installing applications is a necessary evil, make it as easy as possible
- Repeatability and script-ability should be a goal
- The less mouse and click, the better

- RedHat, CentOS, Other “EL” Linux
  - yum
- Ubuntu and other Debian
  - apt
- macOS
  - Homebrew – http://brew.sh
- Windows
  - Chocolatey – https://chocolatey.org

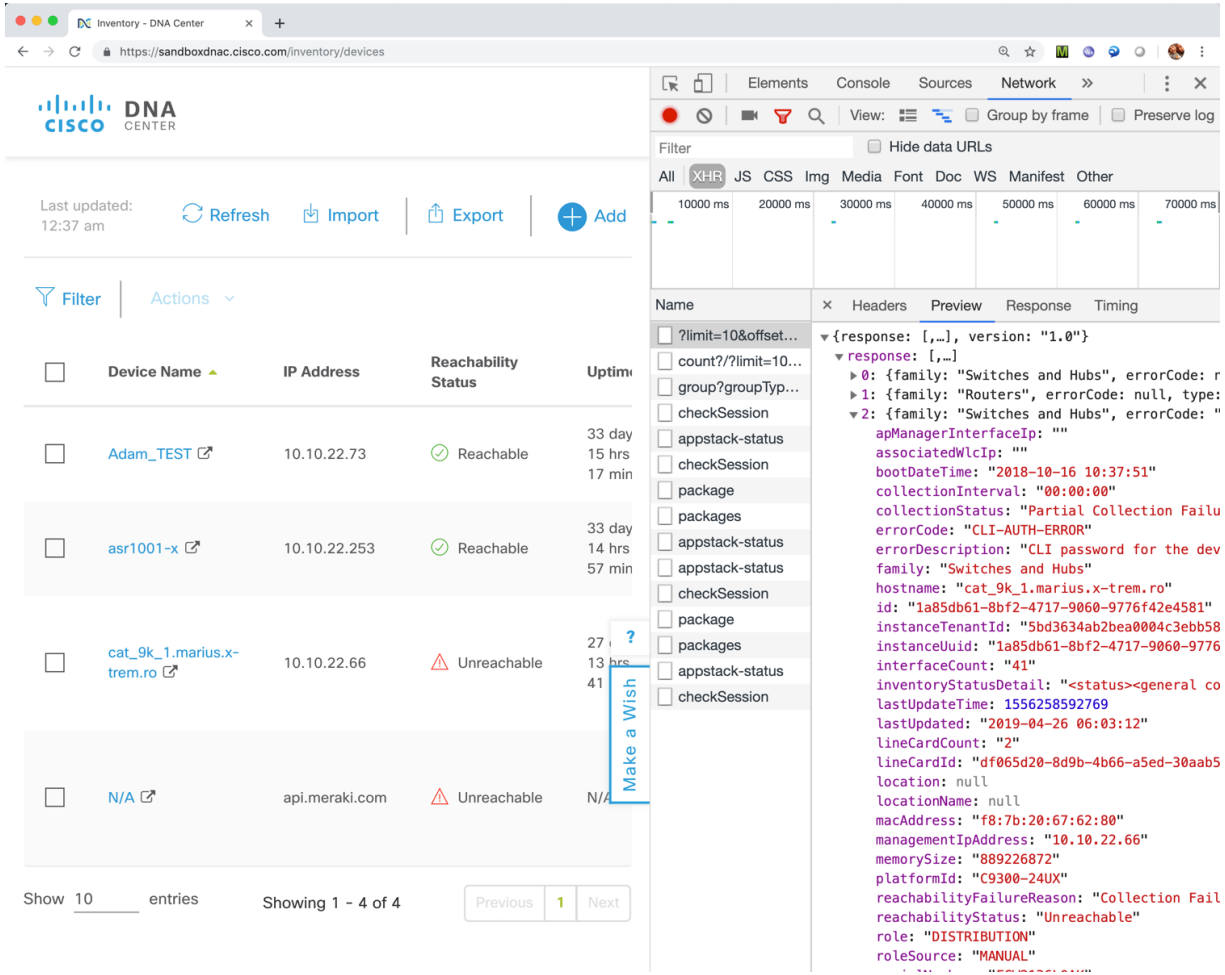
Extend your IDEs with Plug-Ins… to a point

- Language syntax checking and display coloring
- Handy tools for diffs, data
- Integrations to cloud services
- So much more
- *Potential Risks*
  - *Performance*
  - *Security Issues (Do you trust the community)*

- VS Code
  - https://marketplace.visualstudio.com
- Atom
  - https://atom.io/packages
- Sublime
  - https://packagecontrol.io

Peak under the hood with Developer Mode

- Web applications are everywhere today
- Displaying a nice GUI requires API calls, JavaScript, Cookies, and more
- You can get a lot of info about how stuff works by seeing what happens when you interact with a page
- Developer tools on all major browsers today

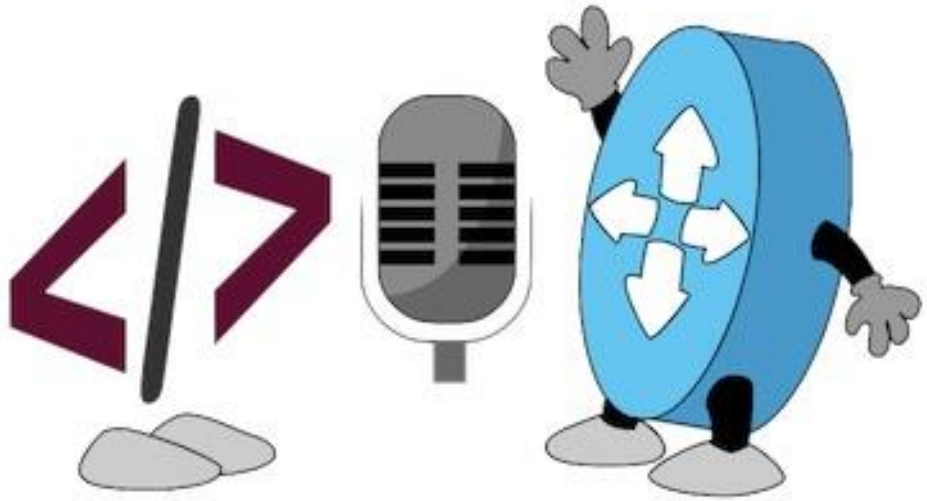


DEVNET developer.cisco.com

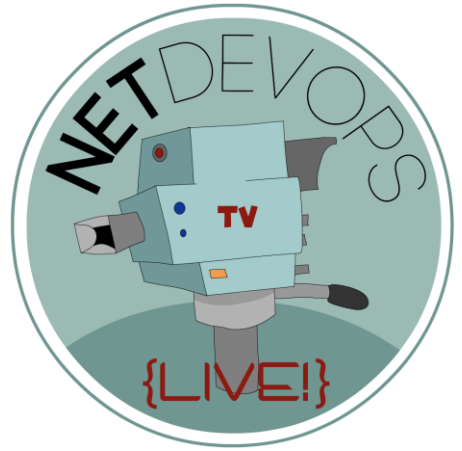
Summing up

What did we talk about?

- Don’t be bashful, Love thy terminal
- Shhh… SSH like a pro
- Oscar worthy bash scripts! (and other bash-y stuff)
- Make your Apps Dance



Webinar Resource List



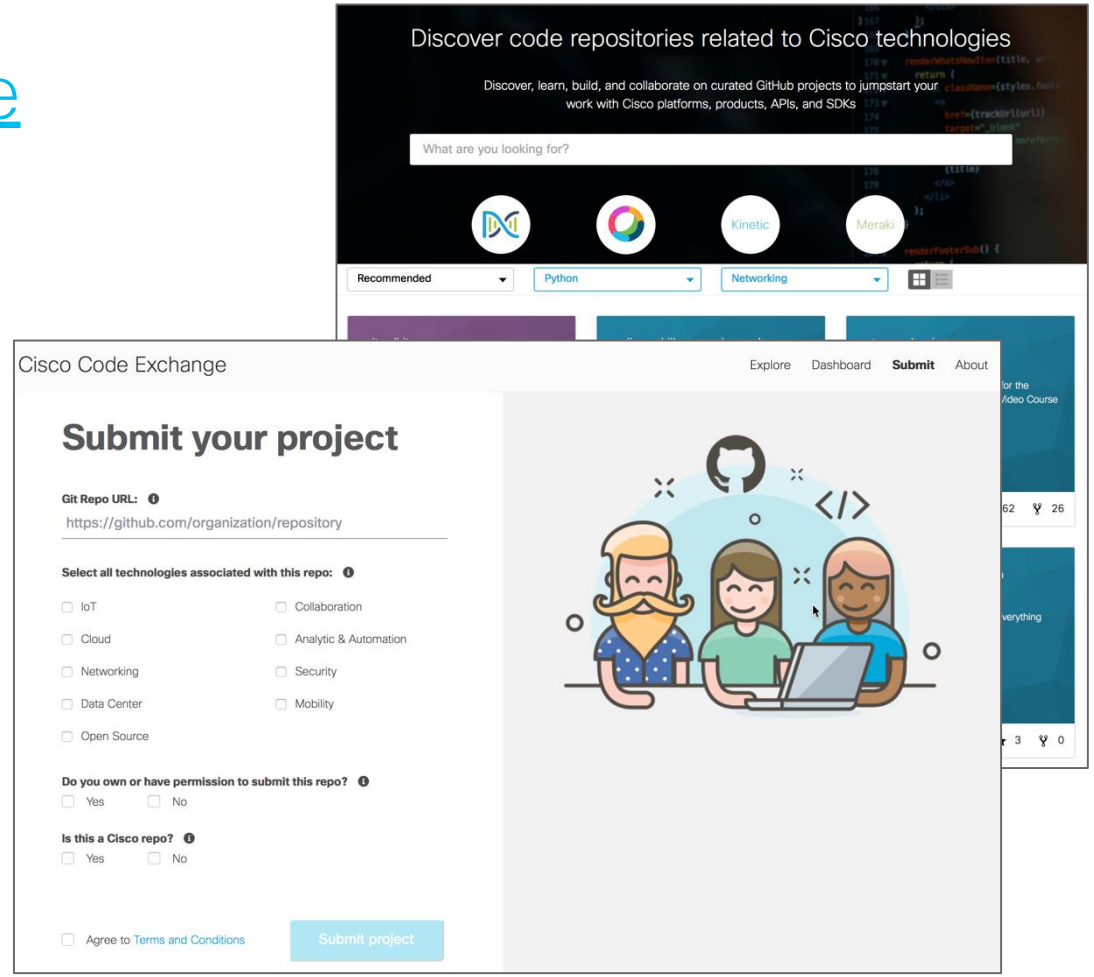
- Docs and Links
  - https://developer.cisco.com/python
  - Bash and Guestshell on NX-OS
  - Guest Shell on IOS XE
- Learning Labs
  - Laptop Setup http://cs.co/lab-dev-setup
  - Introduction to Guest Shell on IOS XE http://cs.co/lab-iosxe-guestshell
- DevNet Sandboxes
  - Multi-IOS Sandbox with VIRL http://cs.co/sbx-multi
- Code Samples
  - https://github.com/hpreston/netdevops_demos -> dev_tips folder

NetDevOps Live! Code Exchange Challenge

developer.cisco.com/codeexchange

Put together some bash scripts and/or a Makefile for one of your network automation projects to help setup the environment.

Example: Add ”make dev” and “make clean” capabilities to your project.



Looking for more about NetDevOps?

- NetDevOps on DevNet
  developer.cisco.com/netdevops
- NetDevOps Live!
  developer.cisco.com/netdevops/live
- NetDevOps Blogs
  blogs.cisco.com/tag/netdevops
- Network Programmability Basics Video Course
  developer.cisco.com/video/net-prog-basics/

Got more questions? Stay in touch!

Hank Preston

hapresto@cisco.com
@hfpreston
http://github.com/hpreston

CISCO DEVNET

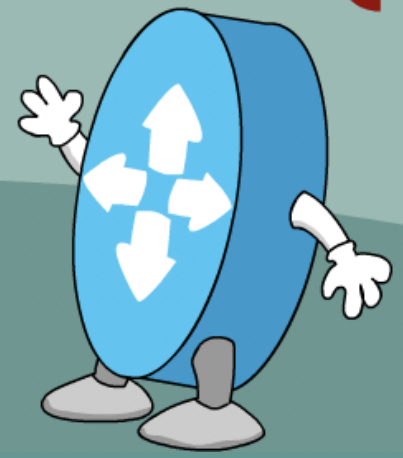
LEARN CODE INSPIRE CONNECT

developer.cisco.com

@CiscoDevNet
facebook.com/ciscodevnet/
http://github.com/CiscoDevNet

NETDEVOPS {LIVE!}

cisco DEVNET



https://developer.cisco.com/netdevops/live

@netdevopslive