**Shell**

# ln -s source destination

$ cd

$ ln -s ../httpd/html/users/ranga ./public\_html

$ ls -l ./public\_html

lrwxrwxrwx 1 ranga users 26 Nov 9 1997 public\_html ->

../httpd/html/users/ranga

#backgroup process

ls ch0\*.doc &

[1] + Done ls ch0\*.doc & #set –o monitor enable, disable set +o monitor, check all set -o

$ long\_running\_process

^Z[1] + Stopped (SIGTSTP) long\_running\_process

$ long\_running\_process2

^Z[2] + Stopped (SIGTSTP) long\_running\_process2

$

# To move the first one to the background, I use the following:

$ bg %1

[1] long\_running\_process &

$

# The second process is still suspended and can be moved to the background as follows:

$ bg %2

[2] long\_running\_process2 &

$

$ long\_running\_process

^Z[1] + Stopped (SIGTSTP) long\_running\_process

$ bg

[1] long\_running\_process &

$

You can move it back to the foreground as follows:

$ fg %1

long\_running\_process

$ kill %1 #kill job #1

$ nohup somecommand &

$ jobs

[3] + Running first\_one &

[2] - Stopped (SIGTSTP) second\_one

[1] Stopped (SIGTTIN) third\_one &

# array assignment 1

$ FRUIT=apple

$ FRUIT[1]=peach #FRUIT[0] = apple

# array assignment 2

set -A name value1 value2 ... valuen

# bash

name=(v1 v2 … vn)

$ set -A band derri terry mike gene or the bash command

$ band=(derri terry mike gene)

$ band[0]=derri

$ band[1]=terry

$ band[2]=mike

$ band[3]=gene

$ echo ${band[0]} #derri

#access all items

${name[\*]}

${name[@]}

$ FRUIT[0]=apple

$ FRUIT[1]=banana

$ FRUIT[2]=orange

$ echo ${FRUIT[\*]}

apple banana orange

FRUIT[3]="passion fruit"

$ **echo ${FRUIT[\*]}** #results in 5 items, not 4

apple banana orange passion fruit

$ **echo ${FRUIT[@]}** #results in 4

apple banana orange passion fruit

$ FRUIT=kiwi

**$ readonly FRUIT**

$ echo $FRUIT

kiwi

$ FRUIT=cantaloupe

/bin/sh: FRUIT: This variable is read only.

export PATH=/sbin:/bin

H8: Substitution

\* Matches zero or more occurrences of any character

? Matches one occurrence of any character

[characters] Matches one occurrence of any of the given characters

$ ls ch0?.doc

ch01.doc ch02.doc ch03.doc ch04.doc ch05.doc ch06.doc

$ ls ch0[0123456789].doc

ch01.doc ch02.doc ch03.doc ch04.doc ch05.doc ch06.doc

$ ls \*[a-zA-Z0-9] #matches all files ending with a letter or a number.

$ ls [!a]\* #list all files except those that start with letter a

# var substituation

${parameter:-word} If parameter is null or unset, word is substituted for parameter. The value of parameter does not change.

${parameter:=word} If parameter is null or unset, parameter is set to the value of word. ${parameter:?message} If parameter is null or unset, message is printed to standard error. This checks that variables are set correctly.

${parameter:+word} If parameter is set, word is substituted for parameter. The value of parameter does not change.

PS1=${HOST:-localhost}"$ " ; export PS1 ; #PS1 set to HOST if HOST is set, otherwise, set to localhost, HOST value doesn’t change

PS1=${HOST:=´uname -n´}"$ " ; export PS1 HOST ; #both HOST, PS1 are set

if [ -z "$HOST" ] ; then HOST=´uname -n´ ; fi ; PS1="$HOST$ "; export PS1 HOST ; #same as above

foo=$(( ((5 + 3\*2) - 4) / 2 ))

echo $foo #3.5

3

$ cp file1 file2 file3 file4 file5 file6 file7 \

> file8 file9 /tmp #PS2

$ echo 'Line 1

> Line 2'

Line 1

Line 2

$ echo -e “Line1 \nLine2”

H10 Flowcontrol

if list1

then

list2

elif list3

then

list4

else

list5

fi

if list1 ; then list2 ; elif list3 ; then list4 ; else list5 ; fi ;

test expression

[ expression ]

File Test:

**-d file True if file exists and is a directory.**

**-e file True if file exists.**

**-f file True if file exists and is a regular file.**

**-g file True if file exists and has its SGID bit set.**

**-h file True if file exists and is a symbolic link.**

**-k file True if file exists and has its "sticky" bit set.**

**-p file True if file exists and is a named pipe.**

**-r file True if file exists and is readable.**

**-s file True if file exists and has a size greater than zero.**

**-u file True if file exists and has its SUID bit set.**

**-w file True if file exists and is writable.**

**-x file True if file exists and is executable.**

**-o file True if file exists and is owned by the effective user ID.**

$ if [ -d /home/ranga/bin ] ; then PATH="$PATH:/home/ranga/bin" ; fi #dir

$ if [ -f $HOME/.bash\_aliai ] ; then . $HOME/.bash\_aliai ; fi #file

if [ -s $HOME/.bash\_aliai ] ; then . $HOME/.bash\_aliai ; fi #size>0

String Comparison

**-z string True if string has 0 length.**

**-n string True if string has nonzero length.**

**string1 = string2 True if the strings are equal.**

**string1 != string2 True if the strings are not equal.**

if [ -z "$FRUIT\_BASKET" ] ; then

echo "Your fruit basket is empty" ;

else

echo "Your fruit basket has the following fruit: $FRUIT\_BASKET"

fi

if [ "$FRUIT" = apple ] ; then

echo "An apple a day keeps the doctor away."

else

echo "You must like doctors, your fruit $FRUIT is not an apple."

Fi

Numeric Comparison

**int1 -eq int2 True if int1 equals int2.**

**int1 -ne int2 True if int1 is not equal to int2.**

**int1 -lt int2 True if int1 is less than int2.**

**int1 -le int2 True if int1 is less than or equal to int2.**

**int1 -gt int2 True if int1 is greater than int2.**

**int1 -ge int2 True if int1 is greater than or equal to int2.**

if [ $? -eq 0 ] ; then

echo "Command was successful." ;

else

echo "An error was encountered."

exit

fi

if [ $? -ne 0 ] ; then

echo "An error was encountered."

exit

fi

echo "Command was successful."

if [ -z "$DTHOME" ] && [ -d /usr/dt ] ; then DTHOME=/usr/dt ; fi

if [ -z "$DTHOME" -a -d /usr/dt ] ; then DTHOME=/usr/dt ; fi

if [ ! -d $HOME/bin ] ; then mkdir $HOME/bin ; fi

test ! -d $HOME/bin && mkdir $HOME/bin

FRUIT=kiwi

case "$FRUIT" in

apple) echo "Apple pie is quite tasty." ;;

banana) echo "I like banana nut bread." ;;

kiwi) echo "New Zealand is famous for kiwi." ;;

esac

H11 Loop

x=0

while [ $x -lt 10 ]

do

echo $x

x=`echo "$x + 1" | bc`

done

#same as above

x=1;

until [ $x -ge 10 ]

do

echo $x

x=`echo "$x + 1" | bc`

done

for i in 0 1 2 3 4 5 6 7 8 9

do

echo $i

done

select name in word1 workd2 word3

do

echo $name

break

done

select COMPONENT in comp1 comp2 comp3 all none

do

case $COMPONENT in

comp1|comp2|comp3) CompConf $COMPONENT ;;

all) CompConf comp1 CompConf comp2 CompConf comp3 ;;

none) break ;;

\*) echo "ERROR: Invalid selection, $REPLY." ;;

esac

done

The menu presented by the select loop looks like the following:

1) comp1

2) comp2

3) comp3

4) all

5) none

#?

for i in 1 2 3 4 5

do

mkdir -p /mnt/backup/docs/ch0${i}

if [ $? -eq 0 ] ; then

for j in doc c h m pl sh

do cp $HOME/docs/ch0${i}/\*.${j} /mnt/backup/docs/ch0${i}

if [ $? -ne 0 ] ; then break 2 ; fi **#break both loop**

done

else echo "Could not make backup directory."

fi

done

for FILE in $FILES ;

do

if [ ! -f "$FILE" ] ; then

echo "ERROR: $FILE is not a file."

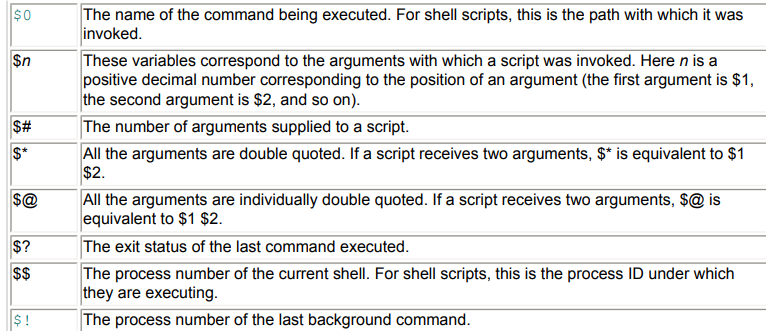
continue

fi

# process the file done

done

H12 Parameter



#!/bin/sh

case $0 in

\*listtar) TARGS="-tvf $1" ;;

\*maketar) TARGS="-cvf $1.tar $1" ;;

esac

tar $TARGS

$ ln -s mytar listtar

$ ln -s mytar maketar

$ ls fruits

apple banana mango peach pear

$ ./maketar fruits

$ ./listtar fruits.tar

$ basename /aa/bb

Bb

#!/bin/sh

case ´basename $0´ in

listtar) TARGS="-tvf $1" ;;

maketar) TARGS="-cvf $1.tar $1" ;;

esac

tar $TARGS

----

**Docker**

docker run busybox:1.24 echo "hello world"

docker images

busybox 1.24 47bcc53f74dc 3 years ago 1.11MB

[root@localhost ~]# docker run busybox:1.24 ls /

bin

dev

etc

home

proc

root

sys

tmp

usr

var

docker run -i -t busybox:1.24 #i-interactive #t-tty -stdin/stdout

[root@localhost ~]# docker run -i -t busybox:1.24

/ #

#image - java class

#container - java instance

#container is created from images, and run container, inside the container, it has all the binaries and dependencies needed to run the app

#registry: where to store images: docker hub

#inside registry, images are store din repoisotories, docker respo is collection of diff images with the same name and diff tages (ver)

#docker search wordpress | less

#docker pull busybox:1.24

#docker images

#docker rmi wordpress #remove image

docker verion

docker info

docker run --help

docker run -dit --name centos --hostname=centos centos /bin/bash

docker attach centos

docker start centos

docker top centos

docker status centos

docker cp centos:/etc/passwd /root/centos-passwd

docker exec centos cat /etc/hostname

docker stop centos

docker kill centos

docker rm centos

mkdir -p /var/local/mysql

chown -R 27:27 /var/local/mysql

docker pull mariadb

docker run --name mariadb -d -v /var/local/mysql:/var/lib/mysql -e MYSQL\_USER=user -e MYSQL\_PASSWORD=password -e MYSQL\_DATABASE=addresses -e MYSQL\_ROOTPASSWORD=password mariadb

docker exec -it mariadb /bin/bash

kubectl create ns secret

kubectl get ns

kubectl create -f busybox-ns.yaml --valide=false

kubectl get ns secret -o yaml

kubectl run ngix --image=nginx

kubectl get deployment

kubectl get rs [-o yaml]

kubectl scale rs {} --replicas=3 #not working

kubectl scale --replicas=3 deployment nginx

#^p^q to disconnect the shell

root@localhost ~]# docker run -d busybox:1.24 sleep 1000

8fb2ee204de2c185b2d89423c717cf474c95ac22ef2b101b1b44e45e6a068b22

[root@localhost ~]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

8fb2ee204de2 busybox:1.24 "sleep 1000" 6 seconds ago Up 5 seconds nostalgic\_edison

docker ps -a

docker run --rm busybox:1.24 sleep 1

docker run --name hello\_world busybox:1.24

docker run -d busybox:12.4 sleep 1000

[root@localhost ~]# docker run -d busybox:1.24 sleep 1000

da51e50300f66f9cb4132c2f18965420374a35183c0144bb3856703a7e6498eb

[root@localhost ~]# docker inspect da51e50300f66f9cb4132c2f18965420374a35183c0144bb3856703a7e6498eb

docker run -it -p 8888:8080 tomcat:8.0

docker run -it -d -p 8888:8080 tomcat:8.0

docker logs {containerid}

[root@localhost ~]# docker history busybox:1.24

IMAGE CREATED CREATED BY SIZE COMMENT

47bcc53f74dc 3 years ago /bin/sh -c #(nop) CMD ["sh"] 0B

<missing> 3 years ago /bin/sh -c #(nop) ADD file:47ca6e777c36a4cff… 1.11MB

----------------------------

#Dockerfile

FROM debian:jessie

RUN apt-get update

RUN apt-get install -y git

RUN apt-get install -y vim

#

RUN apt-get update && RUN apt-get install -y git \

\vim

COPY abc.txt /src/abc.txt #abc.txt in local, /src/abc.txt in image

CMD ["echo", "hello world"]

docker build -t jameslee/debian .

docker images

#docker build -t james/debian . --no-cache=true

----------------------------

docker run someimage

docker ps -a

docker commit {containerid} jameslee/debian:1.00

docker images

----------------------------

docker images

docker tag imageid jleetutorial/debian:1.01

docker login --username=jleetutorial

docker push jleetutorial/debain:1.01 #sends the image to docker hub

#docker container links

dokcer run -d --name redis redis:3.2.0

docker ps

docker build -t dockerapp:v3.0

docker run -d -p 5000:5000 --link redis dockerapp:v0.3

docker exec -it {containerid}

vi /etc/hosts

**Terraform**

terraform init

terraform plan

terraform apply

destroy:

1.

terraform destroy

terraform destroy -target digetialocean\_droplet.mydroplet

terraform destroy -target aws\_instance.myec2

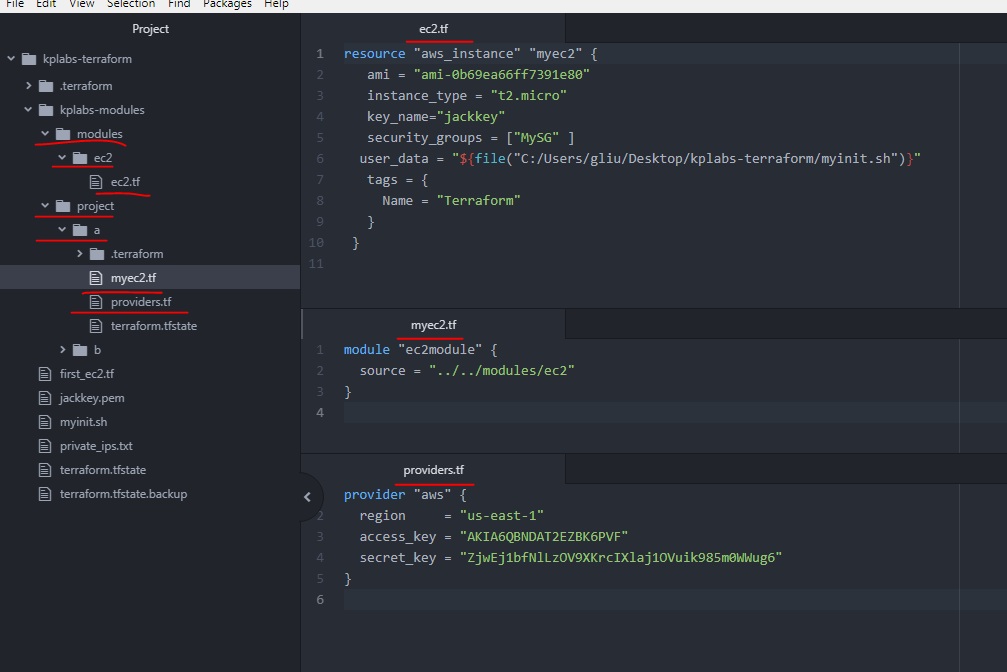
2.

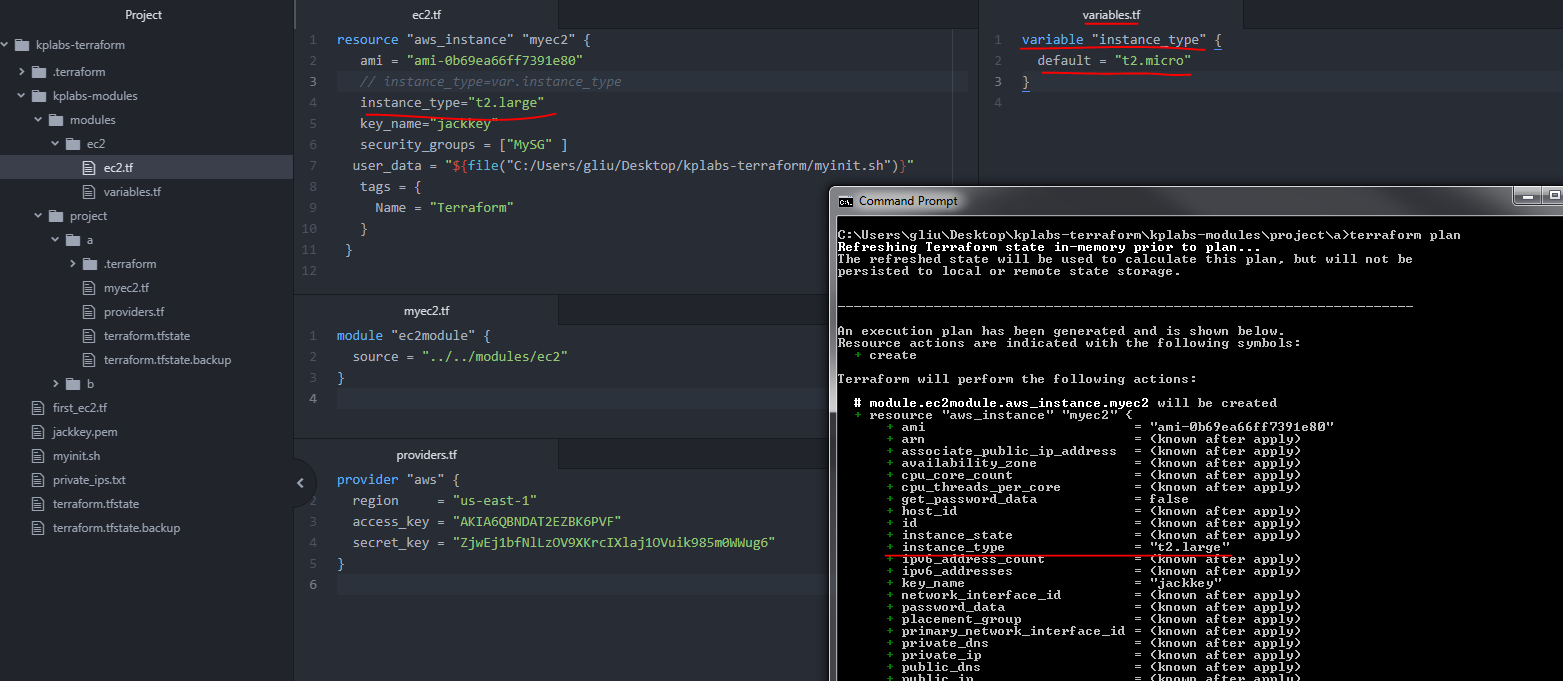
#comment out the aws\_instance.myec2 part in tf file and run terraform plan

terraform refresh

terraform show

Module





Workspace

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace

Usage: terraform workspace

New, list, show, select and delete Terraform workspaces.

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace show

default

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace list

\* default

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace new dev

Created and switched to workspace "dev"!

You're now on a new, empty workspace. Workspaces isolate their state,

so if you run "terraform plan" Terraform will not see any existing state

for this configuration.

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace new prd

Created and switched to workspace "prd"!

You're now on a new, empty workspace. Workspaces isolate their state,

so if you run "terraform plan" Terraform will not see any existing state

for this configuration.

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace list

default

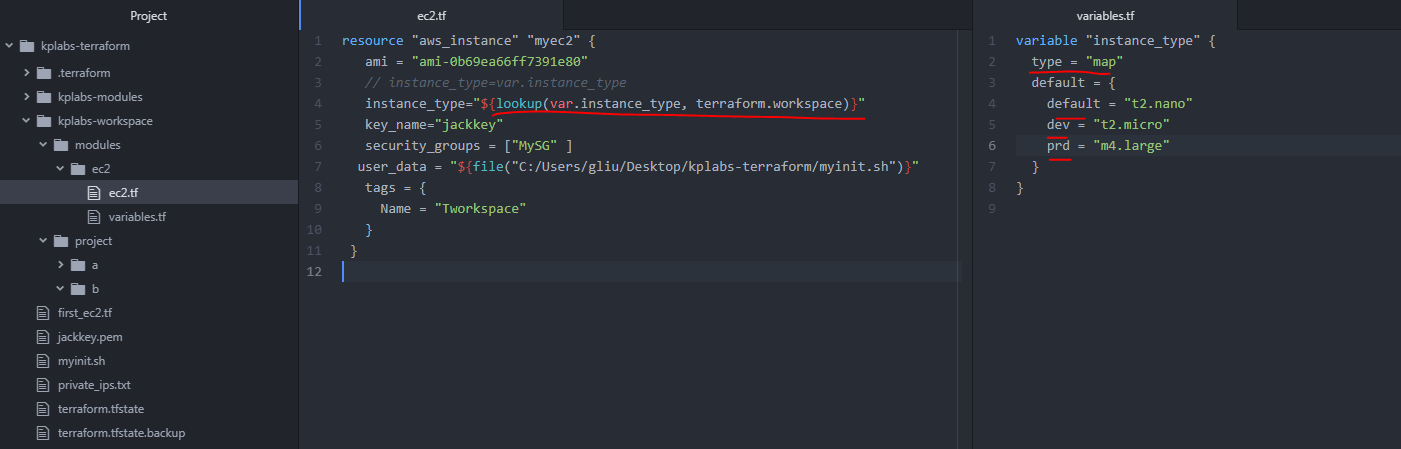
dev

\* prd

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace show

prd

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>



C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform plan

-----------------------------------------------------------------------

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

# module.ec2module.aws\_instance.myec2 will be created

+ resource "aws\_instance" "myec2" {

+ ami = "ami-0b69ea66ff7391e80"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

+ cpu\_core\_count = (known after apply)

+ cpu\_threads\_per\_core = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ id = (known after apply)

+ instance\_state = (known after apply)

**+ instance\_type = "m4.large"**

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ key\_name = "jackkey"

+ network\_interface\_id = (known after apply)

+ password\_data = (known after apply)

+ placement\_group = (known after apply)

+ primary\_network\_interface\_id = (known after apply)

+ private\_dns = (known after apply)

+ private\_ip = (known after apply)

+ public\_dns = (known after apply)

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace list

default

dev

\* prd

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace select dev

Switched to workspace "dev".

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace show

dev

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace list

default

\* dev

Prd

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace list

default

dev

\* prd

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform plan

Terraform will perform the following actions:

# module.ec2module.aws\_instance.myec2 will be created

+ resource "aws\_instance" "myec2" {

+ ami = "ami-0b69ea66ff7391e80"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

+ cpu\_core\_count = (known after apply)

+ cpu\_threads\_per\_core = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ id = (known after apply)

+ instance\_state = (known after apply)

**+ instance\_type = "t2.micro"**

+ ipv6\_address\_count = (known after apply)

+ ipv6\_addresses = (known after apply)

+ key\_name = "jackkey"

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace select default

Switched to workspace "default".

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace list

\* default

dev

prd

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform workspace show

Default

C:\Users\gliu\Desktop\kplabs-terraform\kplabs-workspace\project\a>terraform plan

Refreshing Terraform state in-memory prior to plan...

The refreshed state will be used to calculate this plan, but will not be

persisted to local or remote state storage.

------------------------------------------------------------------------

An execution plan has been generated and is shown below.

Resource actions are indicated with the following symbols:

+ create

Terraform will perform the following actions:

# module.ec2module.aws\_instance.myec2 will be created

+ resource "aws\_instance" "myec2" {

+ ami = "ami-0b69ea66ff7391e80"

+ arn = (known after apply)

+ associate\_public\_ip\_address = (known after apply)

+ availability\_zone = (known after apply)

+ cpu\_core\_count = (known after apply)

+ cpu\_threads\_per\_core = (known after apply)

+ get\_password\_data = false

+ host\_id = (known after apply)

+ id = (known after apply)

+ instance\_state = (known after apply)

**+ instance\_type = "t2.nano"**

+ ipv6\_address\_count = (known after apply)

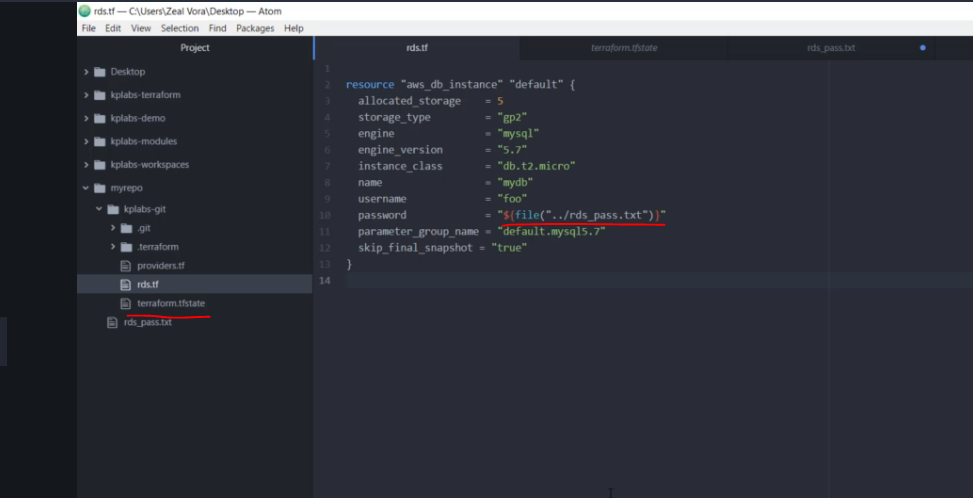
+ ipv6\_addresses = (known after apply)

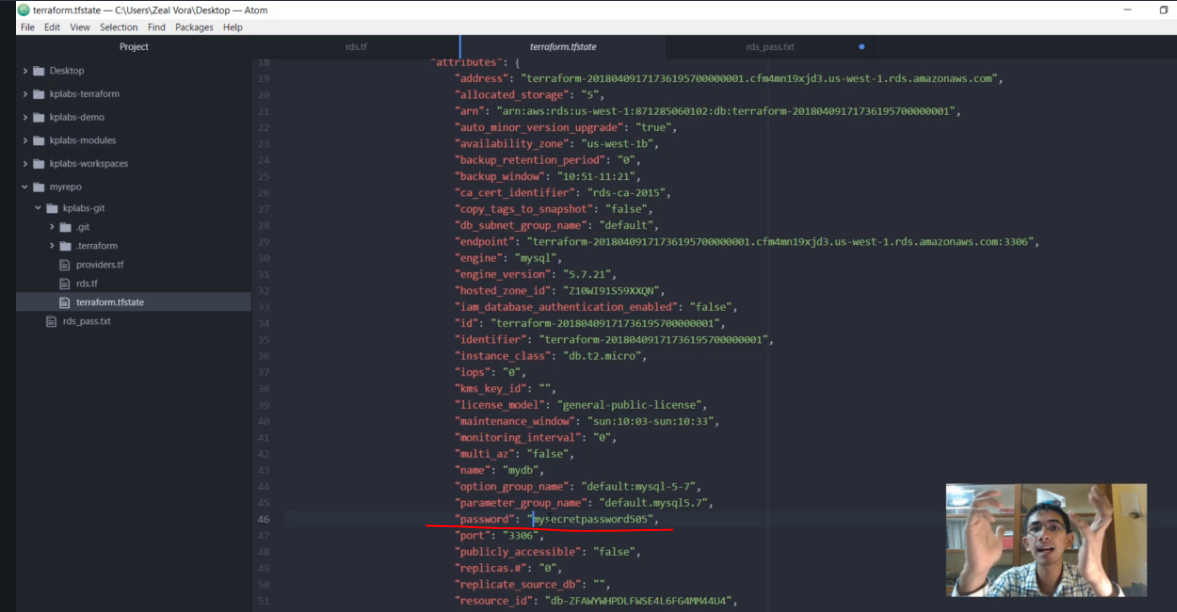
+ key\_name = "jackkey"

+ network\_interface\_id = (known after apply)

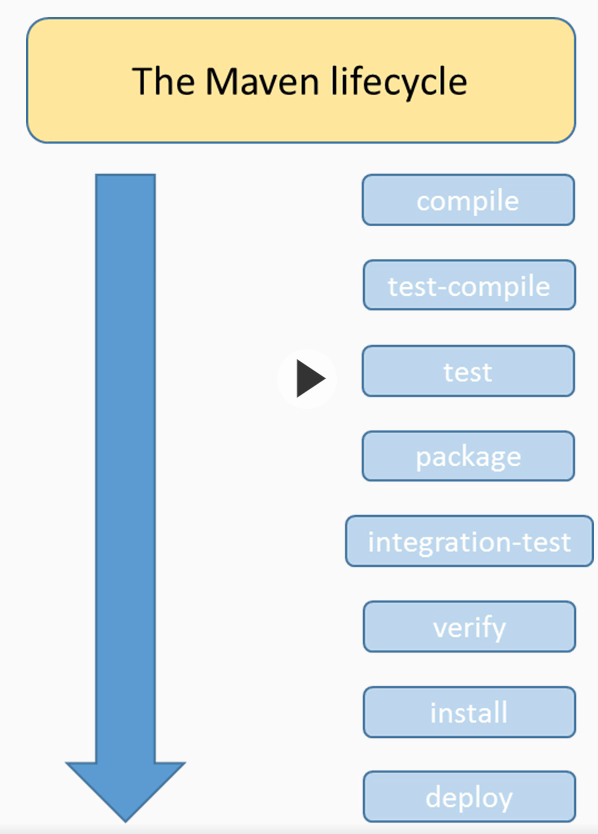
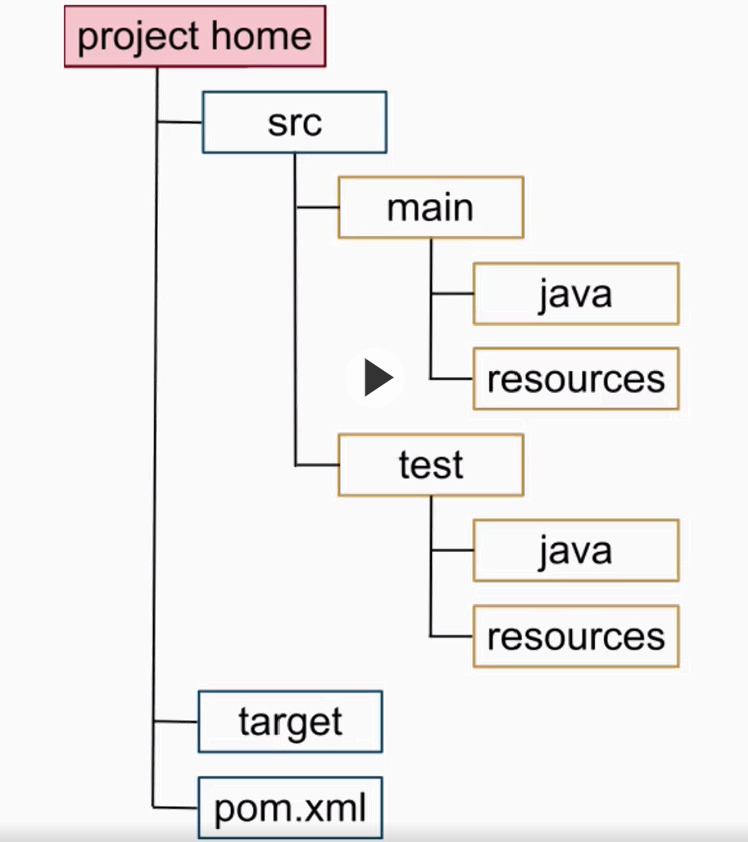
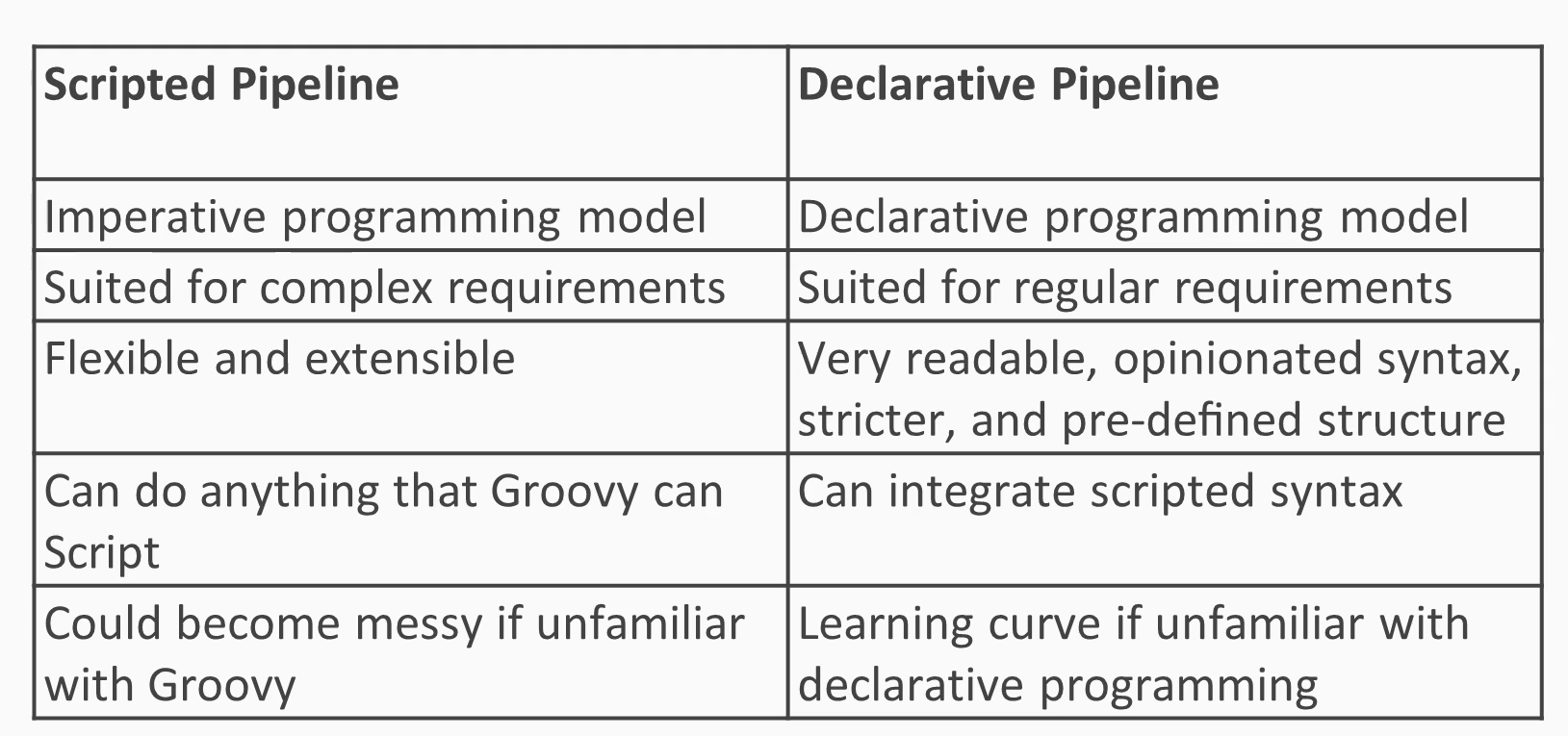
+ password\_data = (known after apply)

Integration with Git

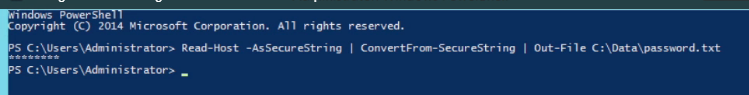


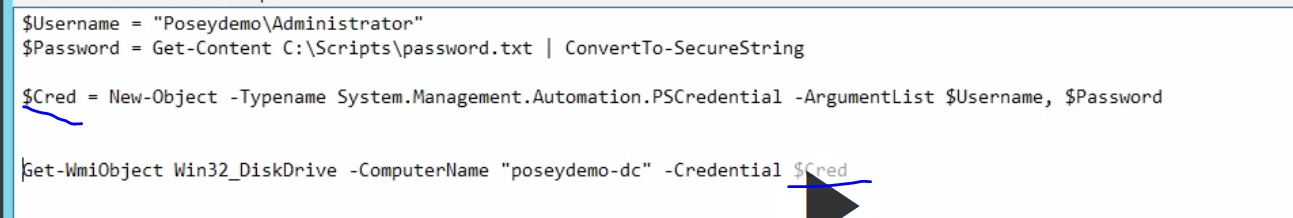


**Jenkins**

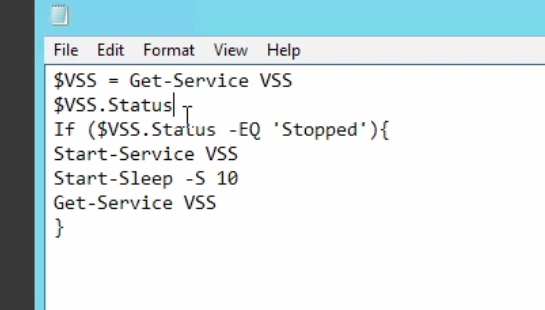
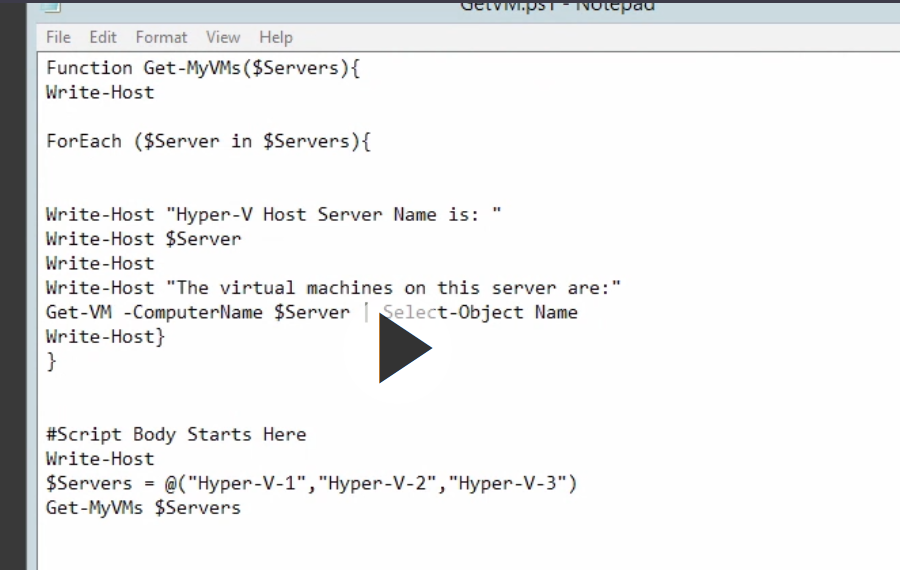
  


**PowerShell**





PS C:\temp1> read-host -AsSEcureString | ConvertFrom-SecureString | Out-File c:\temp1\buildpass.txt



wsadmin>print AdminTask.listServers('[-serverType APPLICATION\_SERVER ]')

server1(cells/ilc03zNode01Cell/nodes/ilc03zNode01/servers/server1|server.xml)

Get-Command

Get-Command Get-\*

Get-Command \*-Service

Get-Help Get-Service

www.microsoft.com/powershell

Start-Service -Name VSS -WhatIf

Stop-Service -Name VSS -Confirm

Get-Process | Sort-Object WS -Descending | Select-Object Name, WS -Last 5 #5 process using least of mem

Get-Service VSS | Select-Object Name, DisplayName, Status | Format-List #Format-Table, FL, FT

Copy-Item C:\data\readme.txt c:\scripts

Get-ChildItem c:\scripts\readme.txt

Remove-Item c:\scripts\readme.txt

Get-Process | Select-Object \* #show everything

Import-Module Hyper-V

Get-Module

Remove-Module Hyper-V

Get-ExecutionPolicy

Set-ExecutionPolicy UnRestricted

Get-ExcutiionPolicy -List

Set-ExecutionPolicy -ExecutionPolicy AllSigned -Scope CurrentUser

Get-ExecutionPolicy -List

Scope ExecutionPolicy

----- ---------------

MachinePolicy Undefined

UserPolicy Undefined

Process Undefined

CurrentUser AllSigned

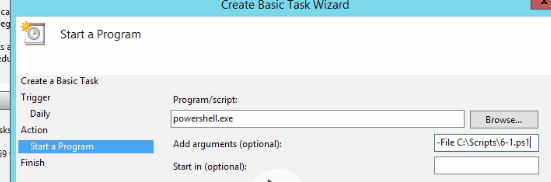
LocalMachine Unrestricted

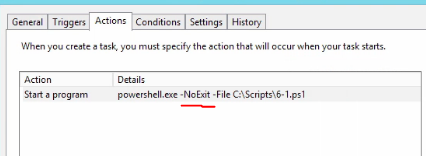
$Cred=Get-Credential

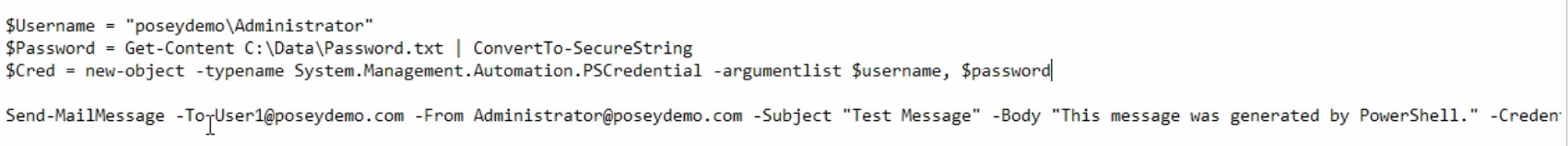
Get-WmiObject Win32\_DiskDrive -ComputerName "xyz-dc" -Credential $Cred

#run in the command line

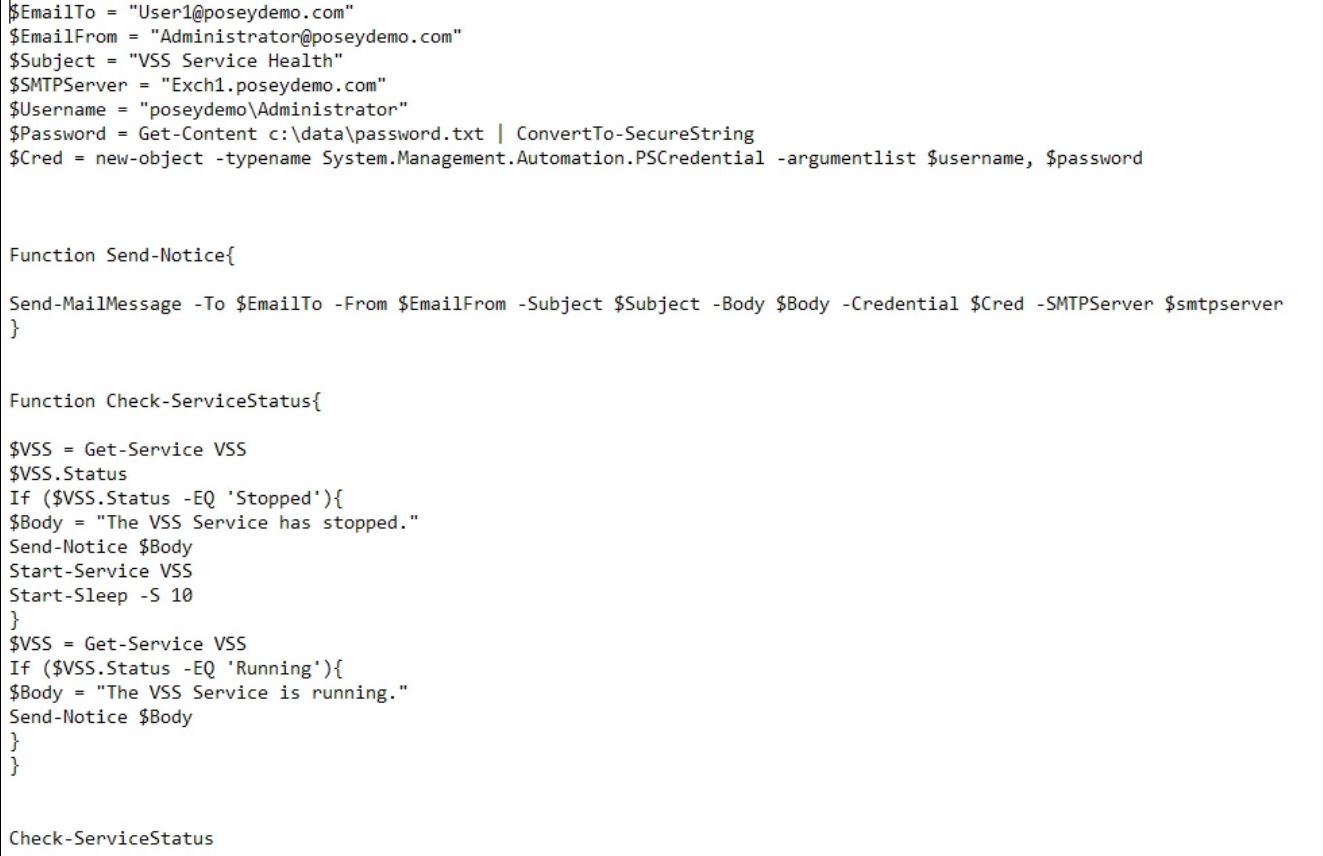
Powershell.exe -File C:\scripts\6-1.ps1





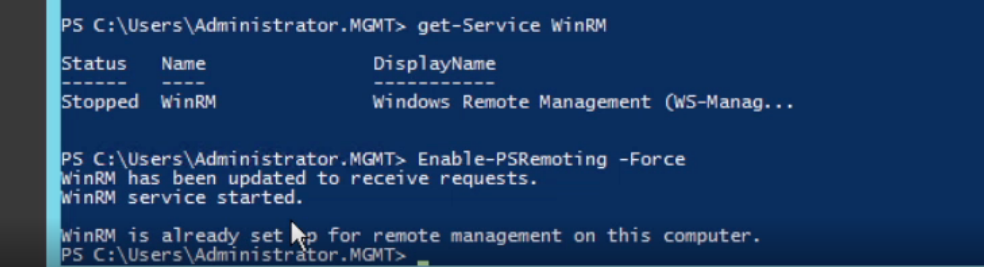
#sending email





Get-Service -ComputerName 172.16.100.49

#Remote manage:



PS C:\Windows\system32>

PS C:\Windows\system32> get-service winrm

Status Name DisplayName

------ ---- -----------

Stopped winrm Windows Remote Management (WS-Manag...

PS C:\Windows\system32> enable-psremoting -force

WinRM has been updated to receive requests.

WinRM service type changed successfully.

WinRM service started.

WinRM has been updated for remote management.

Created a WinRM listener on HTTP://\* to accept WS-Man requests to any IP on this machine.

WinRM firewall exception enabled.

PS C:\Windows\system32>

**PS P:\> $Cred=Get-Credential**

cmdlet Get-Credential at command pipeline position 1

Supply values for the following parameters:

Credential

PS P:\> Enter-PSSession -ComputerName 172.16.100.49 -Credential $Cred

Enter-PSSession : Connecting to remote server failed with the following error message : The WinRM client cannot process

the request. Default authentication may be used with an IP address under the following conditions: the transport is HT

TPS or the destination is in the TrustedHosts list, and explicit credentials are provided. Use winrm.cmd to configure T

rustedHosts. Note that computers in the TrustedHosts list might not be authenticated. For more information on how to se

t TrustedHosts run the following command: winrm help config. For more information, see the about\_Remote\_Troubleshooting

Help topic.

At line:1 char:16

+ Enter-PSSession <<<< -ComputerName 172.16.100.49 -Credential $Cred

+ CategoryInfo : InvalidArgument: (172.16.100.49:String) [Enter-PSSession], PSRemotingTransportException

+ FullyQualifiedErrorId : CreateRemoteRunspaceFailed

**PS P:\> Enter-PSSession -ComputerName prnj11063 -Credential $Cred**

[prnj11063]: PS C:\Users\CMBuild\Documents> get-service

Status Name DisplayName

------ ---- -----------

Running AeLookupSvc Application Experience

Running Albd Atria Location Broker

Stopped ALG Application Layer Gateway Service

Stopped AppIDSvc Application Identity

..

[prnj11063]: PS C:\Users\CMBuild\Documents> Exit-PSSession

PS P:\>

#interactive

$Cred=Get-Credential

Enter-PSSession -ComputerName **prnj11063** -Credential $Cred

Get-Service

Exit-PSSession

#auto

$Cred=Get-Credential

$Session=New-PSSession -Computer prnj11063 -Credential $Cred

Invoke-Command -Session $Session -ScriptBlock {

Get-Service

dir

Exit-PSSession

}

Invoke-Command -FilePath c:\scripts\test.ps1 -ComputerName Server01

PS C:\temp1> read-host -AsSEcureString | ConvertFrom-SecureString | Out-File c:\temp1\buildpass.txt

#type in the password to buildpass.txt

$Username="PROTECH\cmbuild"

$Pass=Get-Content C:\temp1\buildpass.txt | ConvertTo-SecureString

$Cred= New-Object -Typename System.Management.Automation.PSCredential -ArgumentList $Username,$Pass

#$Cred=Get-Credential

$Session=New-PSSession -Computer prnj11063 -Credential $Cred

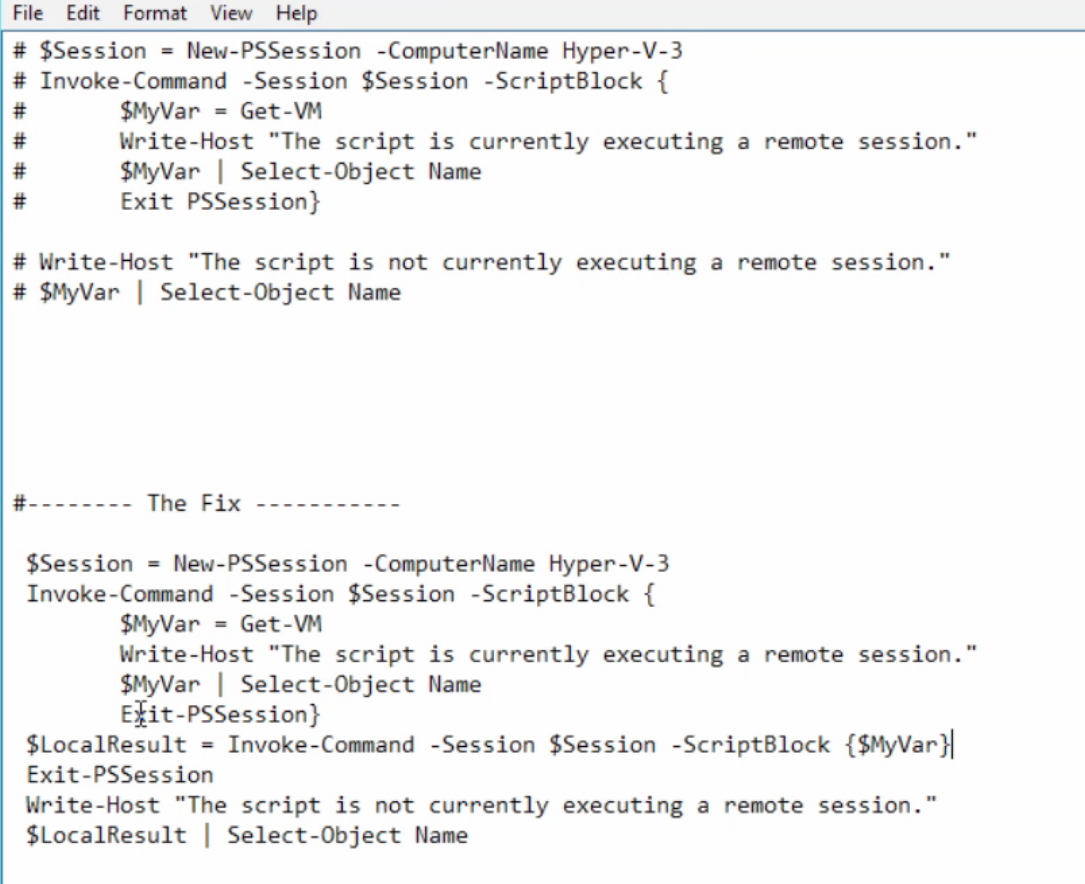
Invoke-Command -Session $Session -ScriptBlock {

Get-Service

Dir

Exit-PSSession

}



**Chef**



