

셸 스크립트를 이용한 클라우드 시스템 운영

Nalee Jang, Senior Consultant of Red Hat Korea





IT Expert on the Cloud!

한국 레드햇에서 오픈스택 및 클라우드 플랫폼 전문 컨설턴트로 근무하고 있습니다.

- 제 이름은 장현정입니다. 하지만, 저는 **NaleeJang** 이라고 불리는게 좋아요!
- 앞에서 소개했듯이 저는 한국 레드햇에서 오픈스택 전문 컨설턴트로 일하고 있어요.
- 그리고, 7년 전에는 HP Helion MVP(Most Valuable Professional, 2015.06.05)도 수상했어요.
- 저는 2권의 책을 출간했는데, 하나는 다들 아시는 “오픈스택을 다루는 기술”과 “오픈스택을 다루는 기술(최신 개정판)”이 있고, 또 다른 하나는 올 초에 출간한 “처음 배우는 셀 스크립트” 라는 책이에요!
- 물론 클라우드 마켓과 관련된 개인 특허를 재미삼아 내 본적도 있어요. 이것도 7년전예요.
- 그리고, 가끔은 제 블로그에 글을 쓰기도 해요. (<http://naleejang.tistory.com>)
- 페이스북은 매일 사용하죠~!! (<http://www.facebook.com/naleejang>)
- 마지막으로 다들 아시다시피 저는 오픈스택 한국 커뮤니티의 두번째 리더였어요. 지금은 아니지만요! (<http://www.facebook.com/groups/openstack.kr/>) (<http://www.openstack.or.kr/>)

생활 속의 클라우드



클라우드와 리눅스와 셸 스크립트의 관계



셸 스크립트를 이용한 클라우드 운영(?)

생활 속의 클라우드



주위를 둘러봐요~!!

생각하지 못한 곳에서

클라우드 환경을 만날 수 있어요.

생활 속의 클라우드



주위를 둘러 보세요!

코로나 바이러스가 발견되기 이전에도 있었지만, 코로나 바이러스로 인해 IT 기술은 매우 빠른 속도로 발전하고 있습니다. 재택근무를 하면서, 온라인 미팅을 하고, 학생들은 온라인으로 수업을 듣습니다. 예전에는 방청석을 예매해 참여했던 방송들은 온라인 방송 참여 시스템을 이용해 온라인으로 방송을 참여합니다. 물론 1인 방송을 하고, 세미나도 온라인으로 진행합니다. 모여서 밥을 못 먹으니 배달음식이 유행을 하고, 올해 초부터 시작된 코로나 백신 역시 예약시스템을 이용해 백신을 맞습니다. 이런 많은 IT 시스템들은 과연 어디에서 실행이 되는걸까요?

생활 속의 클라우드



우리가 사용하는 많은 서비스들을

과연 어디에서 실행이 되는걸까요? 우리는 알게 모르게 클라우드 환경을 사용하고 있습니다. 클라우드 서비스에서 제공하는 클라우드 스토리지에 데이터를 저장하고 클라우드 서비스에서 제공하는 가상서버와 컨테이너 환경을 이용하여 이와 같은 어플리케이션들을 개발하고 실행합니다. 그리고, 클라우드 환경에서 서비스를 합니다.

그럼, 이런 클라우드 환경을 구성하기 위해서는 어떤 기술들이 필요할까요? 함께 알아보까요?

생활 속의 클라우드

우리가 앞에서 살펴봤던 클라우드 환경을 구성하고, 어플리케이션들을 개발하고 서비스하기 위해서는 CPU, 메모리, 네트워크, 디스크와 같은 인프라가 필요 합니다. 그리고, 그 위에 운영체제를 설치하고, 개발이나 서비스에 필요로 하는 개발 플랫폼을 설치하죠 ~!! 그리고, 우리가 개발한 소스를 올리고, 데이터베이스에 데이터를 부어넣으면, 우리가 사용하는 소프트웨어 서비스가 되는 것입니다.

Software



Platform



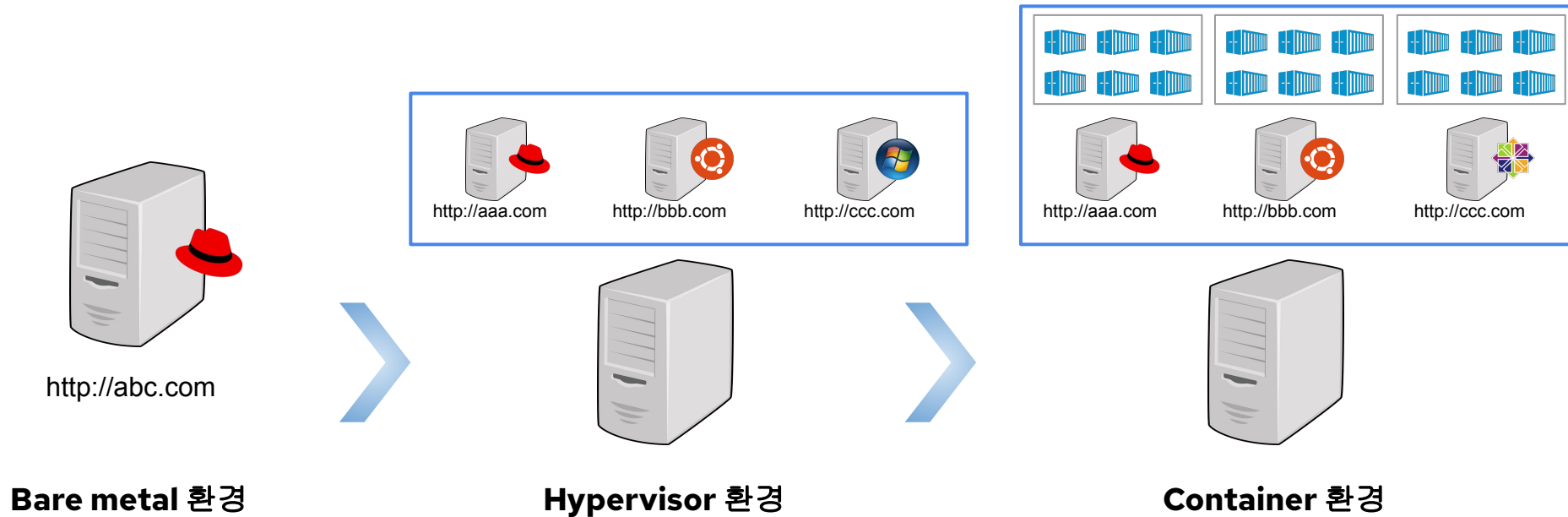
Infrastructure



생활 속의 클라우드

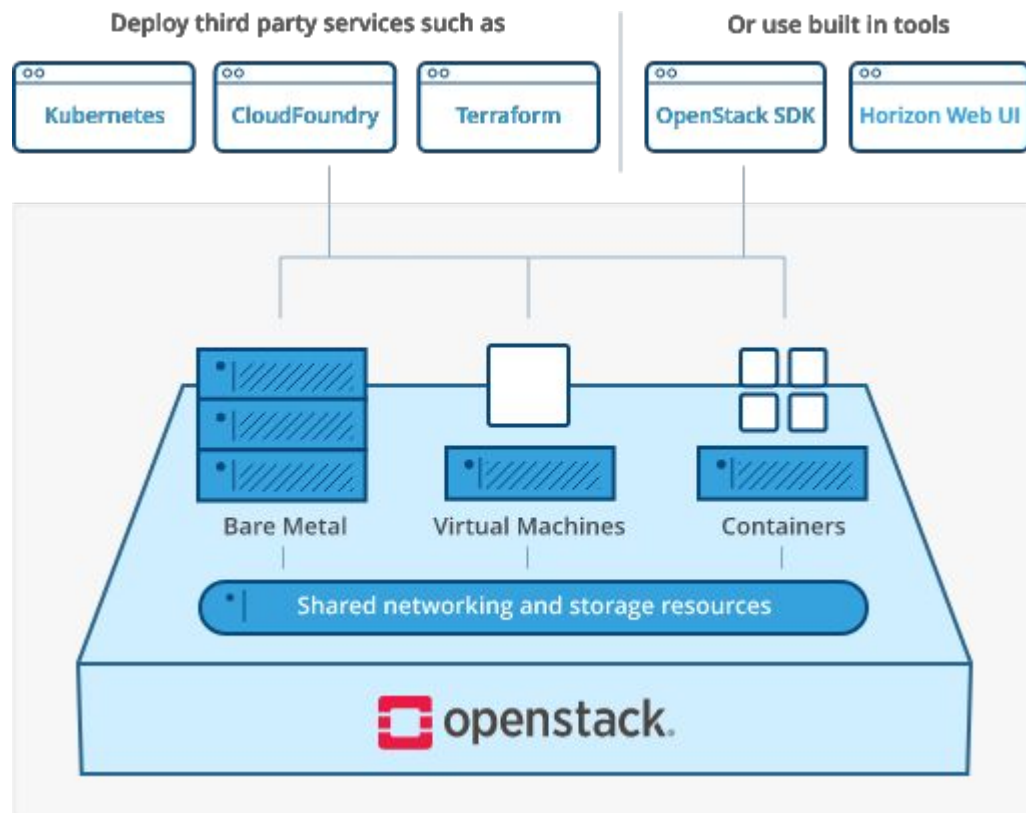
그럼, 인프라를 좀 더 자세하게 들여다 볼까요?

인프라는 아래와 같이 크게 3가지 타입의 인프라 종류가 있습니다. 물리 머신 환경을 그대로 사용하는 **Bare metal** 환경, 물리 머신에 하이퍼바이저를 설치하고 가상 서버를 생성하여 사용하는 **Hypervisor** 환경, 컨테이너 엔진을 사용하여 물리머신 혹은 가세서버 위에 컨테이너를 생성하여 사용하는 **Container** 환경이 있습니다.



클라우드와 리눅스

클라우드 환경을 구축하기 위해서는 리눅스가 필요합니다.



OpenStack packages

« » 📄 📌

Distributions release OpenStack packages as part of the distribution or using other methods because of differing release schedules. Perform these procedures on all nodes.

Note

The set up of OpenStack packages described here needs to be done on all nodes: controller, compute, and Block Storage nodes.

Warning

Your hosts must contain the latest versions of base installation packages available for your distribution before proceeding further.

Note

Disable or remove any automatic update services because they can impact your OpenStack environment.

- [OpenStack packages for SUSE](#)
 - [Enable the OpenStack repository](#)
 - [Finalize the installation](#)
- [OpenStack packages for RHEL and CentOS](#)
 - [Prerequisites](#)
 - [Enable the OpenStack repository](#)
 - [Finalize the installation](#)
- [OpenStack packages for Ubuntu](#)
 - [Archive Enablement](#)
 - [Sample Installation](#)
 - [Client Installation](#)

클라우드와 리눅스

클라우드 환경을 구축하기 위해서는 리눅스가 필요합니다.

Deploy third party services such as

Kubernetes

CloudFoundry

Terraform

Or use built in tools

OpenStack SDK

Horizon Web UI

OpenStack packages

Distributions release OpenStack packages as part of the distribution or using other methods because of differing release schedules. Perform these procedures on all nodes.

Note

needs to be done on all nodes: controller, compute, and Block Storage nodes.

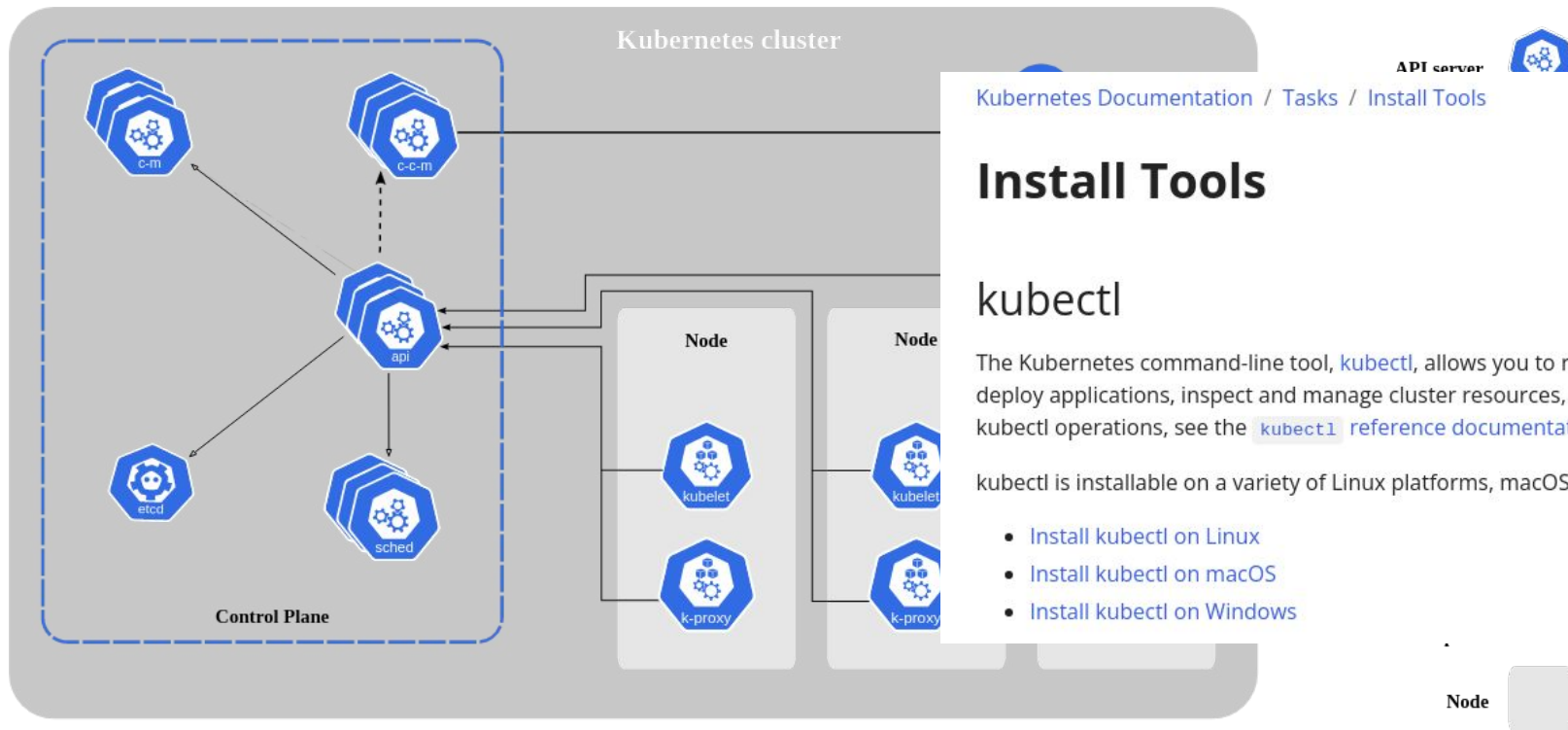
installation packages available for your distribution before proceeding further.

because they can impact your OpenStack environment.

- [OpenStack packages for SUSE](#)
 - [Enable the OpenStack repository](#)
 - [Finalize the installation](#)
- [OpenStack packages for RHEL and CentOS](#)
 - [Prerequisites](#)
 - [Enable the OpenStack repository](#)
 - [Finalize the installation](#)
- [OpenStack packages for Ubuntu](#)
 - [Archive Enablement](#)
 - [Sample Installation](#)
 - [Client Installation](#)

클라우드와 리눅스

클라우드 환경을 구축하기 위해서는 리눅스가 필요합니다.



[Kubernetes Documentation](#) / [Tasks](#) / [Install Tools](#)

Install Tools

kubectl

The Kubernetes command-line tool, [kubectl](#), allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs. For more information including a complete list of kubectl operations, see the [kubectl reference documentation](#).

kubectl is installable on a variety of Linux platforms, macOS and Windows. Find your preferred operating system below.

- [Install kubectl on Linux](#)
- [Install kubectl on macOS](#)
- [Install kubectl on Windows](#)

Node

클라우드와 리눅스

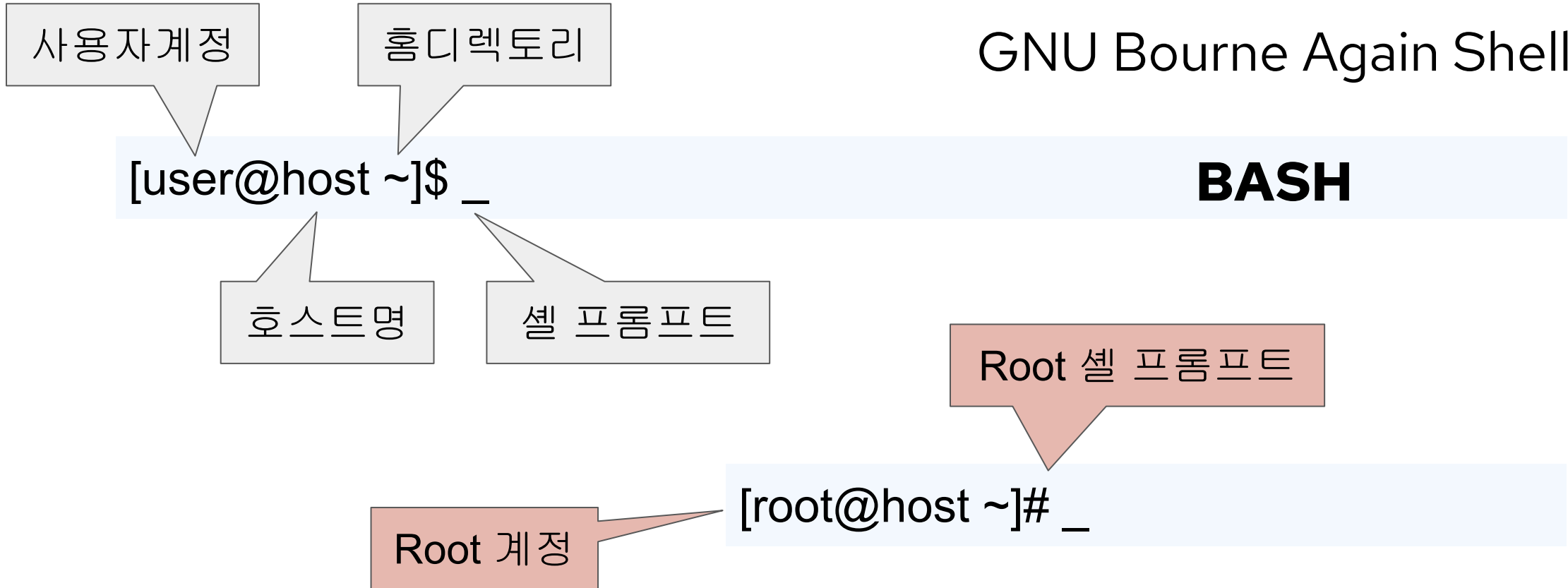
클라우드 환경을 구축하기 위해서는 리눅스가 필요합니다.



리눅스와 셸 스크립트

리눅스는 셸로 구성되어 있습니다.

GNU Bourne Again Shell



리눅스와 셸 스크립트

리눅스에서 셸 이용한 프로그램이 셸 스크립트입니다.

셸 스크립트는

- 우리가 자주 사용하는 명령어들을 프로세스화하여 파일에 작성한 것을 의미합니다.
- 스크립트라는 단어를 사용했기 때문에 if, for와 같은 간단한 제어문을 사용할 수 있으며,
- 명령어 및 실행결과를 저장할 수 있는 변수를 사용할 수 있습니다.

주요 셸 스크립트 구문

- 변수 : 함수, 전역변수, 지역변수, 예약변수, 환경변수, 위치 매개변수, 특수 매개변수
- 조건문 : if, switch - case
- 반복문 : for, while
- 연산자 : 문자열 연산자, 비교 연산자, 논리 연산자, 디렉터리 연산자, 파일 연산자
- 정규표현식 : POSIX 기본 및 확장 문법, POSIX 문자 클래스

주요 리눅스 명령어

주요 리눅스 명령어

- 문자열을 찾을 수 있는 grep
- 파일을 찾을 수 있는 find
- 특정 인덱스 문자열을 출력할 수 있는 awk
- 찾은 문자열을 바꿀 수 있는 sed
- 날짜와 시간을 알려주는 date

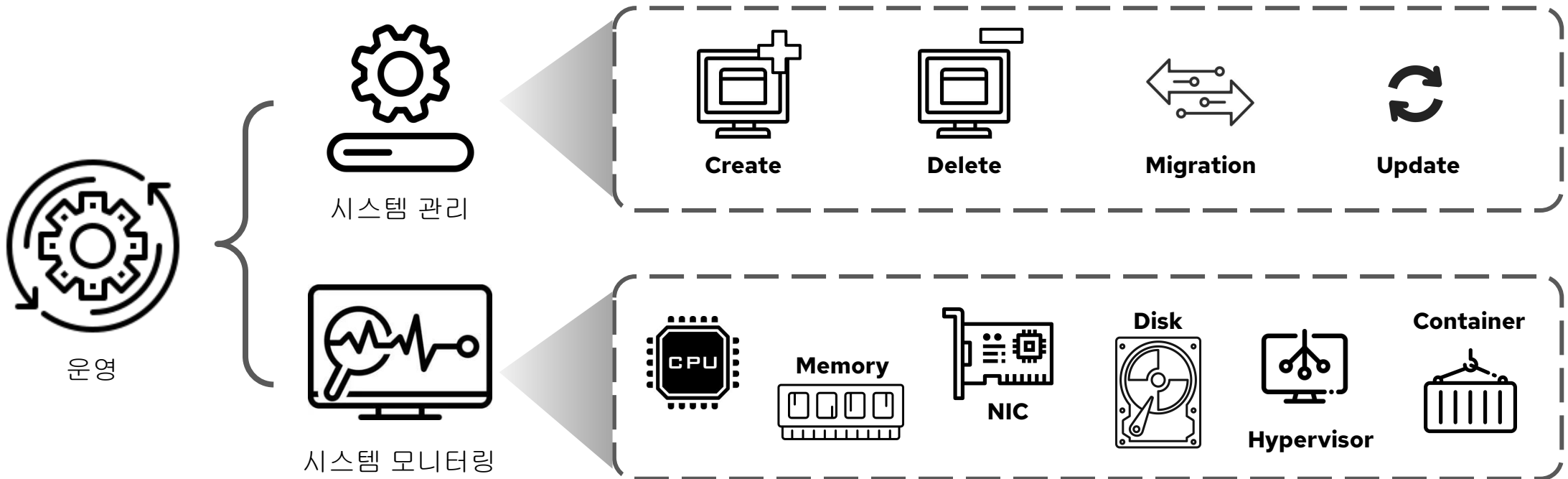
셀 스크립트를 이용한 클라우드 시스템 운영(?)



셀 스크립트 하나만 잘 써도
우리는 클라우드 시스템을
아주 효율적으로 관리할 수
있어요.

운영에는 어떤 일들이 있을까요?

운영 업무에는 크게 시스템 관리와 시스템 모니터링이 있습니다.

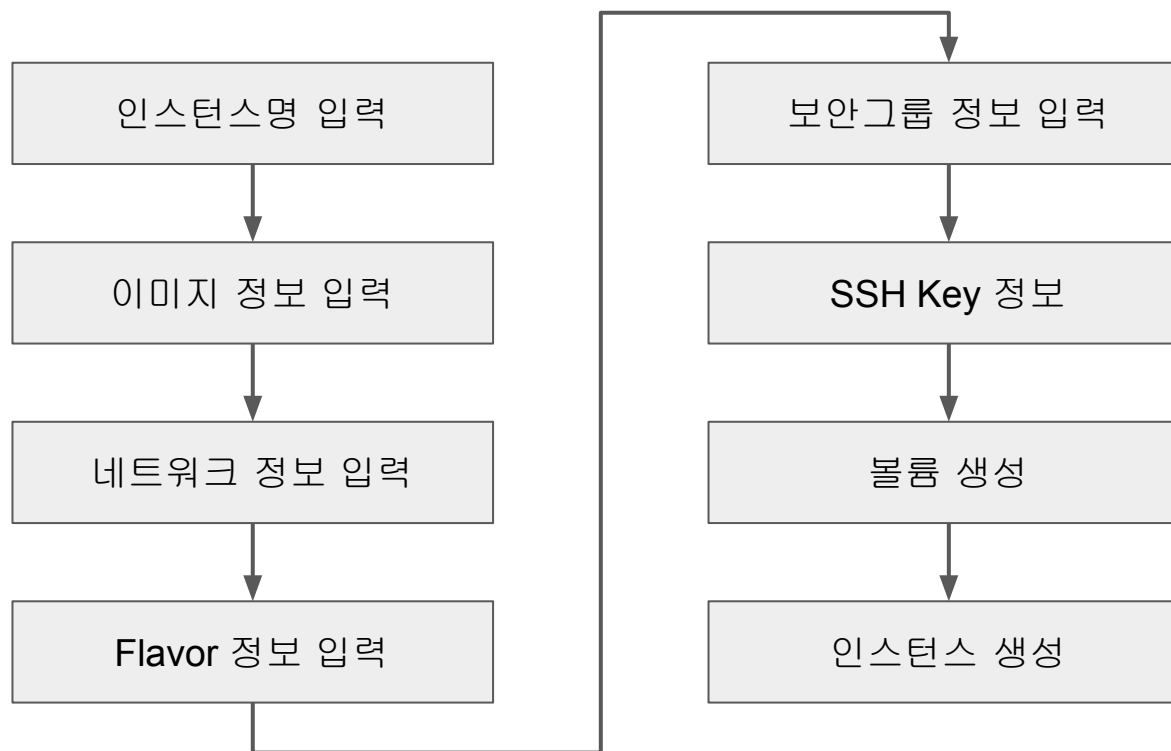


셀 스크립트를 이용하여 인스턴스를 만들어봅시다!

셀 스크립트도 프로세스가 필요합니다.

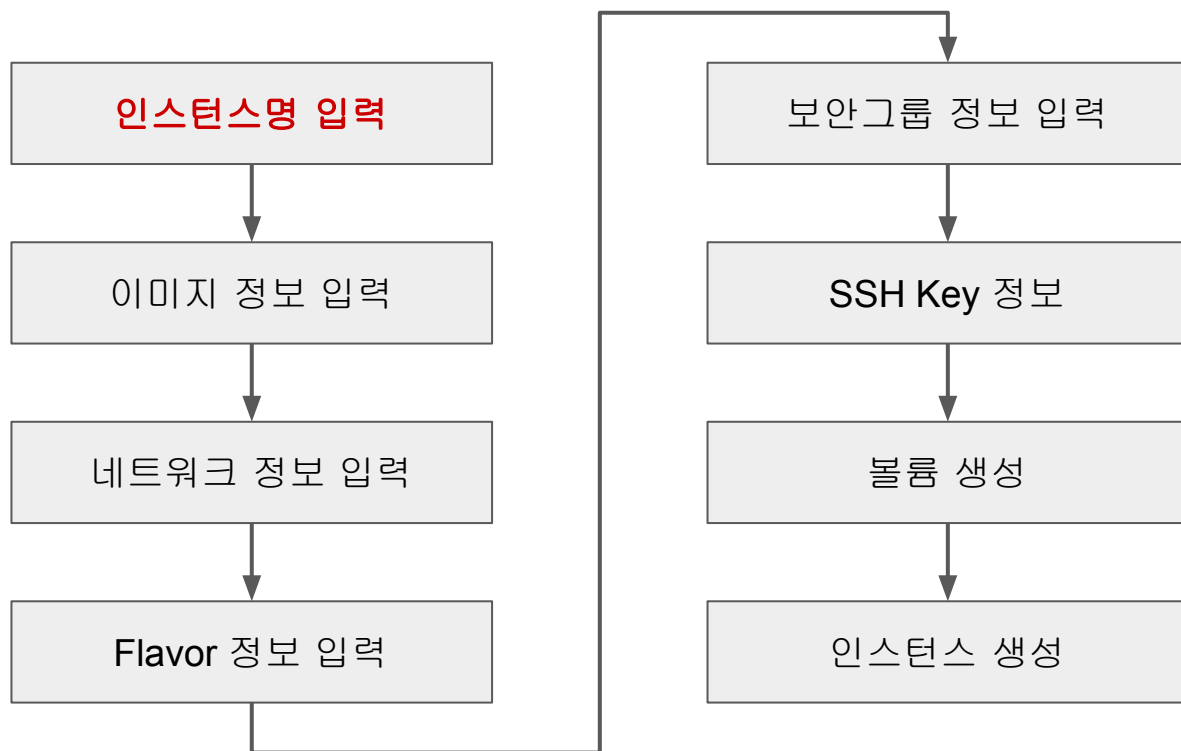


Instance Create



셀 스크립트를 이용하여 인스턴스를 만들어봅시다!

셀 스크립트도 프로세스가 필요합니다.



● 셀 스크립트로 구현하기

```
[nalee@localhost ~]$ vi create-instance.sh
#!/bin/bash

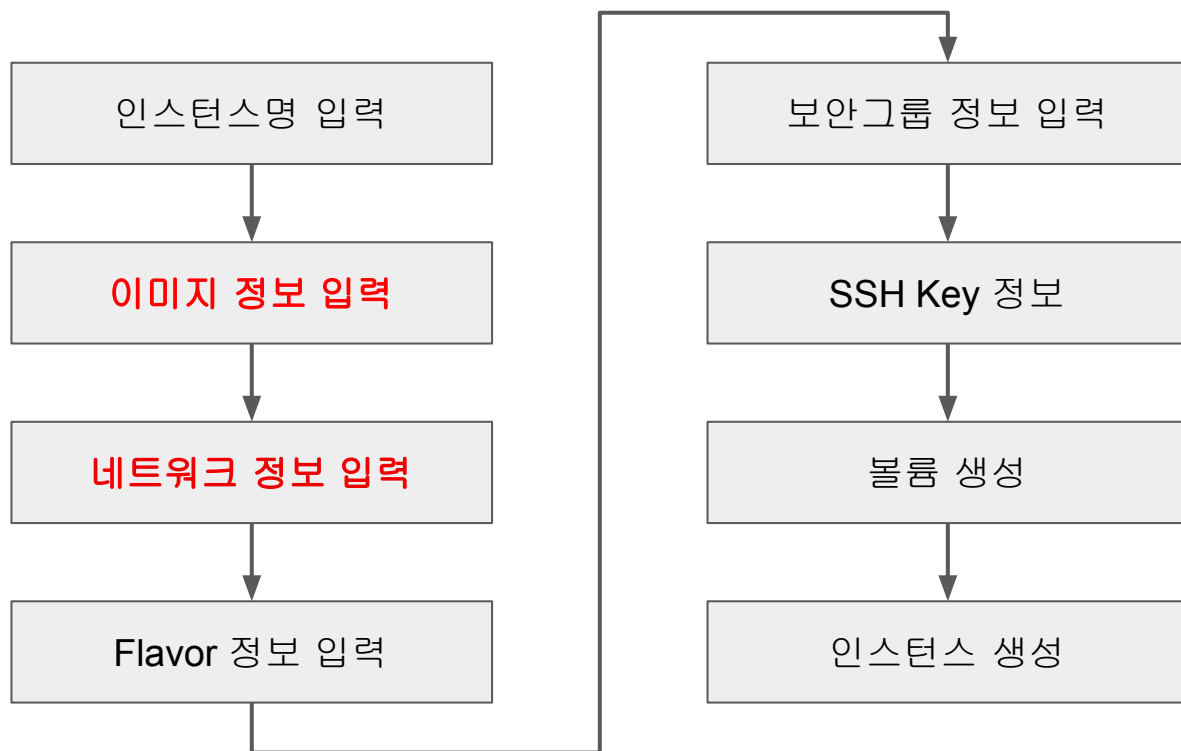
# 인스턴스명 입력
read -p "Input instance name : " vmname
```

read 명령어는 외부 사용자로부터는 직접 파라미터를 입력받을 때 사용하는 명령어입니다.

-p 옵션: 프롬프트 메시지를 함께 보여줄 때 사용하는 옵션
vmname : 사용자로부터 입력받은 문자열을 저장 하기 위한 변수

셀 스크립트를 이용하여 인스턴스를 만들어봅시다!

셀 스크립트도 프로세스가 필요합니다.



● 셀 스크립트로 구현하기

이미지 정보

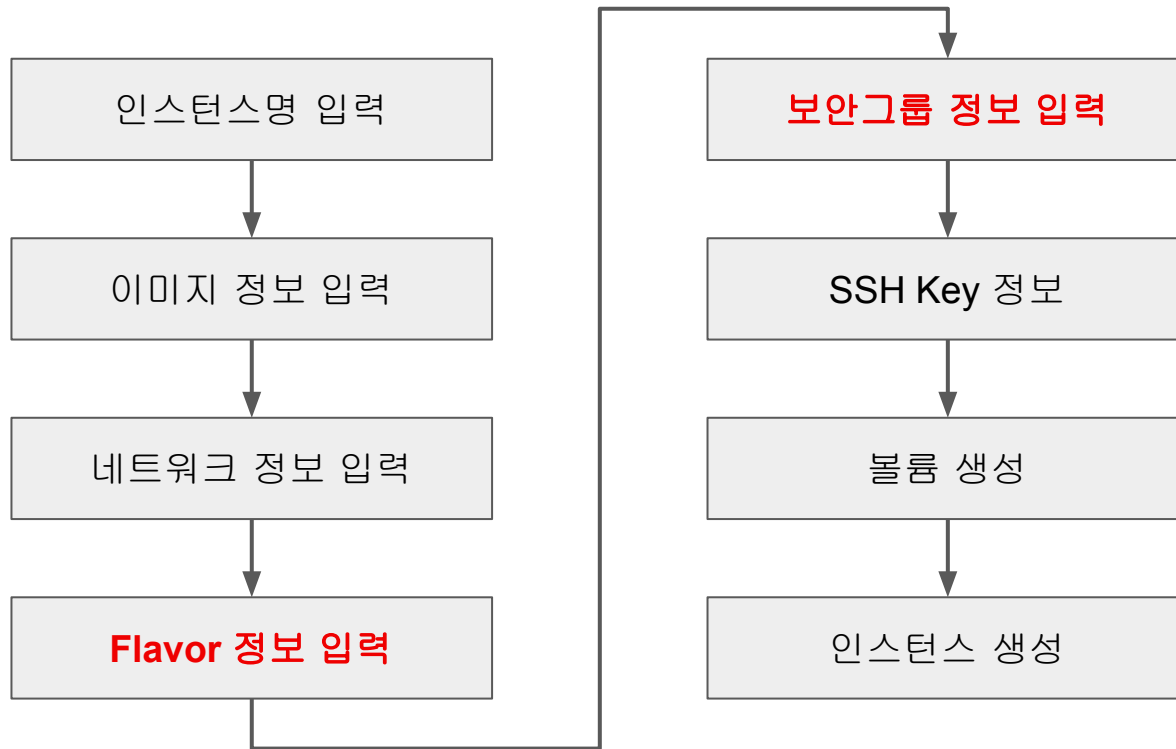
```
echo "== Image List =="
openstack image list -c Name -f value
read -p "Input image name : " image
```

네트워크 정보

```
echo "== Network List =="
openstack network list -c Name -f value
read -p "Input network name : " net
```

셀 스크립트를 이용하여 인스턴스를 만들어봅시다!

셀 스크립트도 프로세스가 필요합니다.



● 셀 스크립트로 구현하기

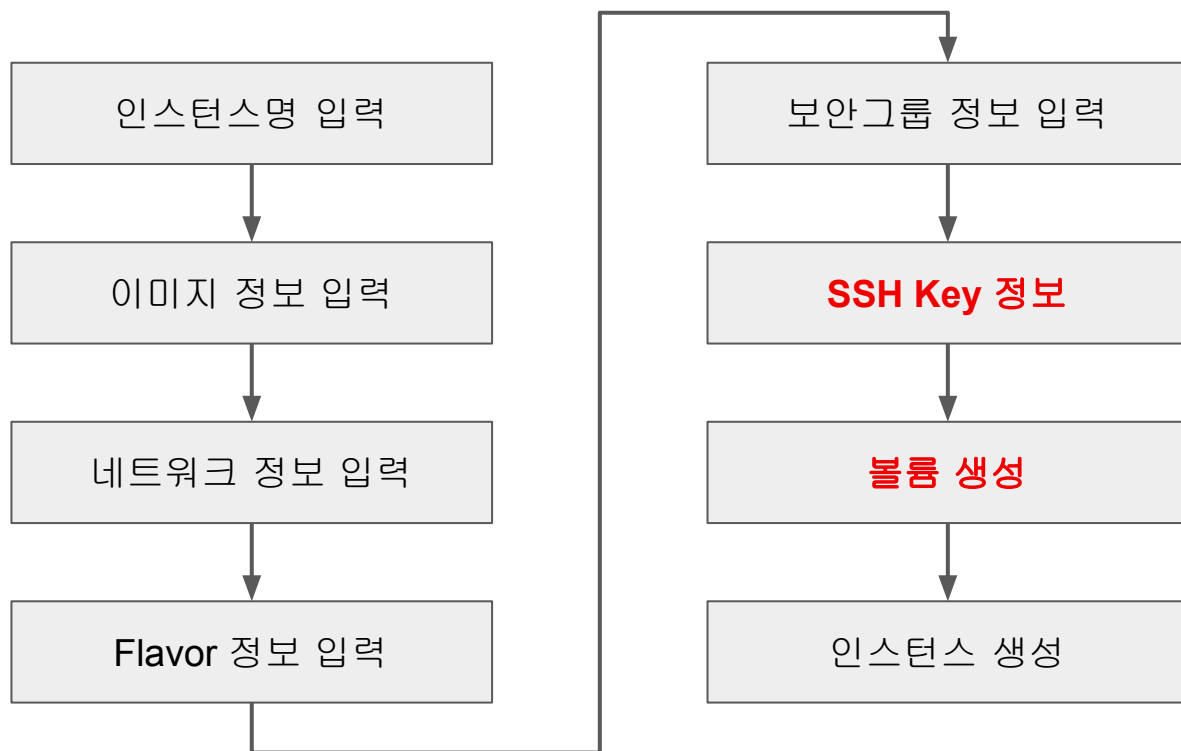
```

# Flaver 정보
echo "== Flavor List =="
openstack flavor list -c Name -f value
read -p "Input flavor name : " flavor

# 보안그룹 정보
echo "== Security group List =="
openstack security group list --project
$OS_PROJECT_NAME -c Name -f value
read -p "Input security group name : " sec
secgrp=$(openstack security group list --project
$OS_PROJECT_NAME -f value -c ID -c Name | grep
"$sec\$" | awk '{print $1}')
```

셀 스크립트를 이용하여 인스턴스를 만들어봅시다!

셀 스크립트도 프로세스가 필요합니다.



● 셀 스크립트로 구현하기

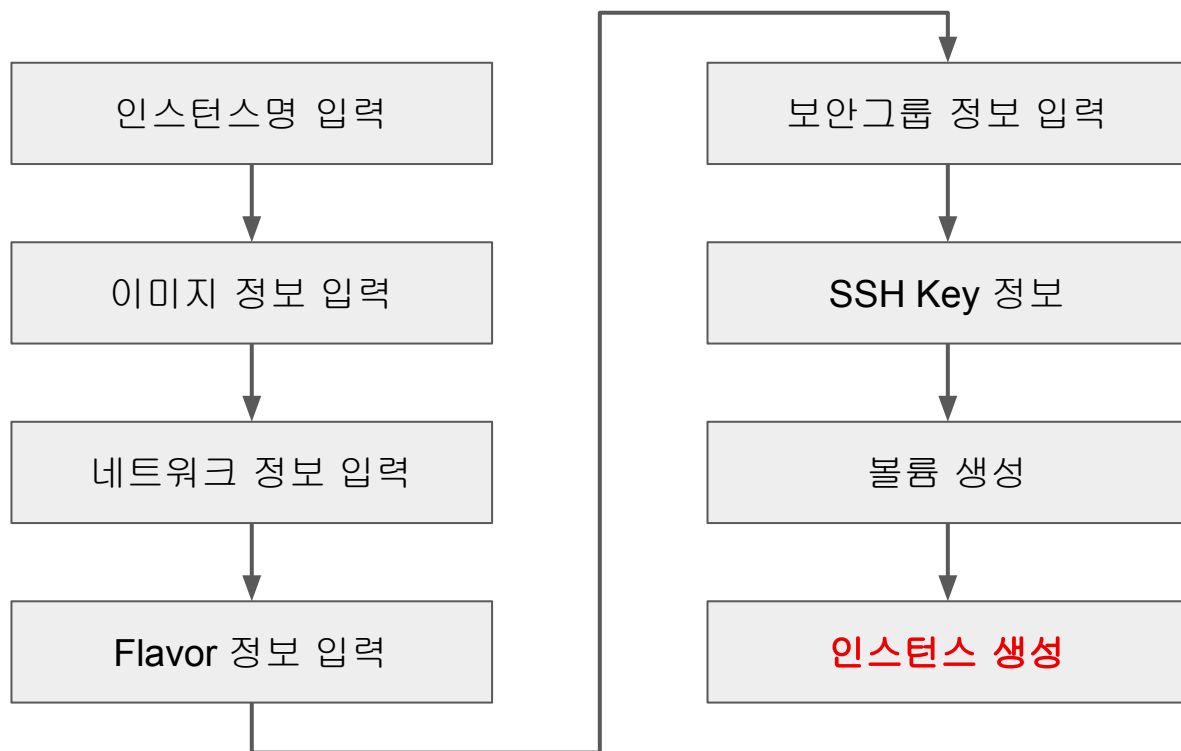
```

# SSH 키 정보
echo "== Keypair List =="
openstack keypair list -c Name -f value
read -p "Input keypair name : " keypair

# 볼륨 생성
echo "== Create volume =="
read -p "Input volume size: " size
openstack volume create --size $size --image $image --bootable $vmname
    
```

셸 스크립트를 이용하여 인스턴스를 만들어봅시다!

셸 스크립트도 프로세스가 필요합니다.



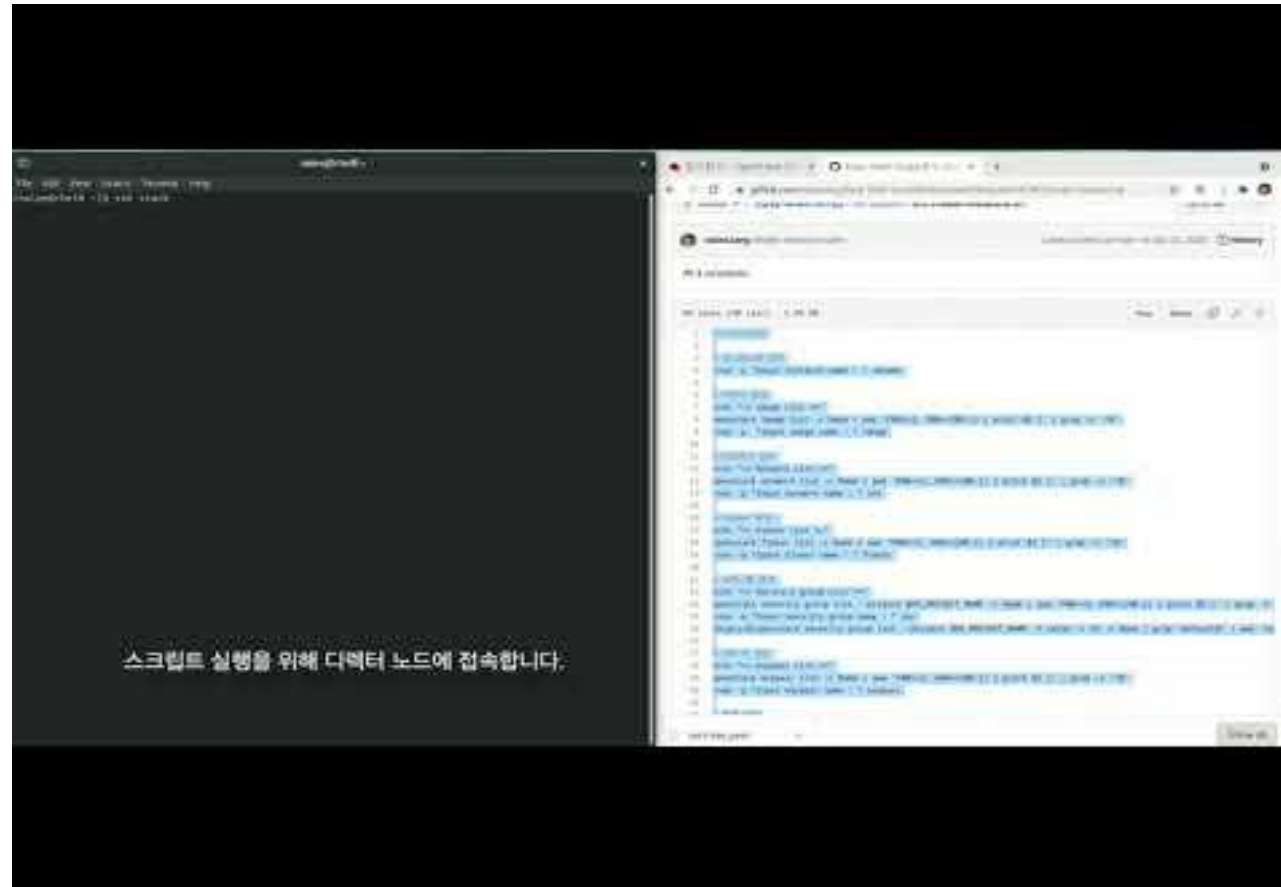
● 셸 스크립트로 구현하기

```

# 인스턴스 생성
echo "Create Instance Starting"
openstack server create \
--volume $(openstack volume list --name $vmname -f \
value -c ID) \
--flavor $flavor \
--security-group $secgrp \
--key-name $keypair \
--network $net \
--wait \
$vmname
    
```

셀 스크립트를 이용하여 인스턴스를 만들어봅시다!

데모를 같이 볼까요!



Youtube URL: https://youtu.be/C_iUDJ7ZPFE

그 외 시스템 관리를 위한 스크립트

처음 배우는 셸 스크립트 활용하기

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search Sign in Sign up

naleeJang / Easy-Shell-Script Public

Notifications Star 8 Fork 5

Code Issues Pull requests Actions Projects Wiki Security Insights

master Easy-Shell-Script / 12.Cloud-OpenStack / Go to file

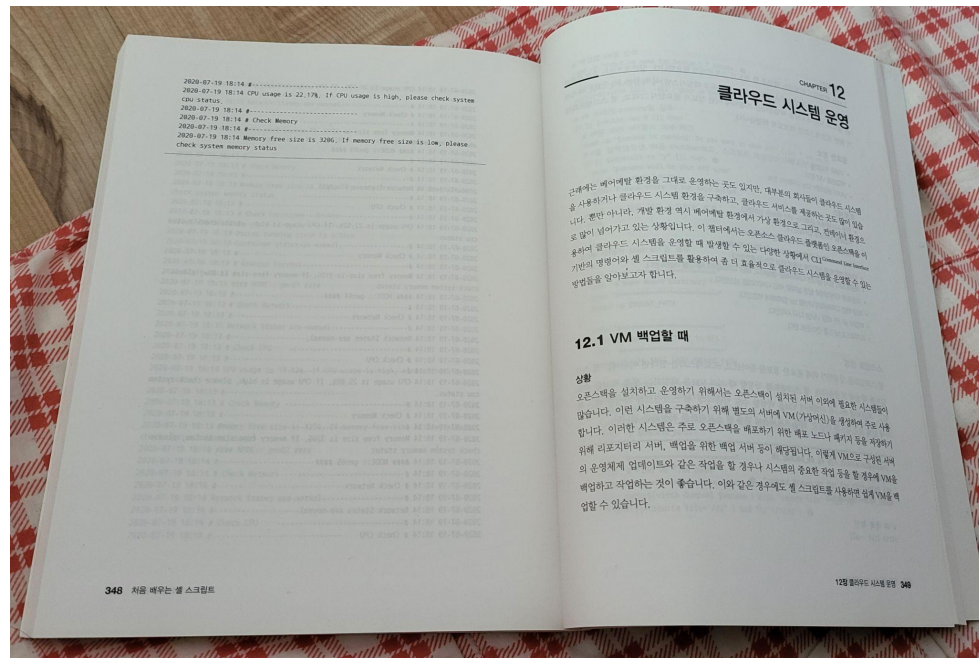
naleeJang Ninth commit 28c2eca on Aug 2, 2020 History

File	Commit	Time
12.1 vm_backup.sh	Eighth commit	16 months ago
12.2 chk_openstack.sh	Ninth commit	15 months ago
12.3 image-upload.sh	Ninth commit	15 months ago
12.4 flavor-create.sh	Ninth commit	15 months ago
12.5 vm-migrate.sh	Ninth commit	15 months ago
12.6 vm-poweroff.sh	Ninth commit	15 months ago
12.7 vm-start.sh	Ninth commit	15 months ago
12.8 vm-delete.sh	Ninth commit	15 months ago

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

GitHub URL

<https://github.com/naleeJang/Easy-Shell-Script>



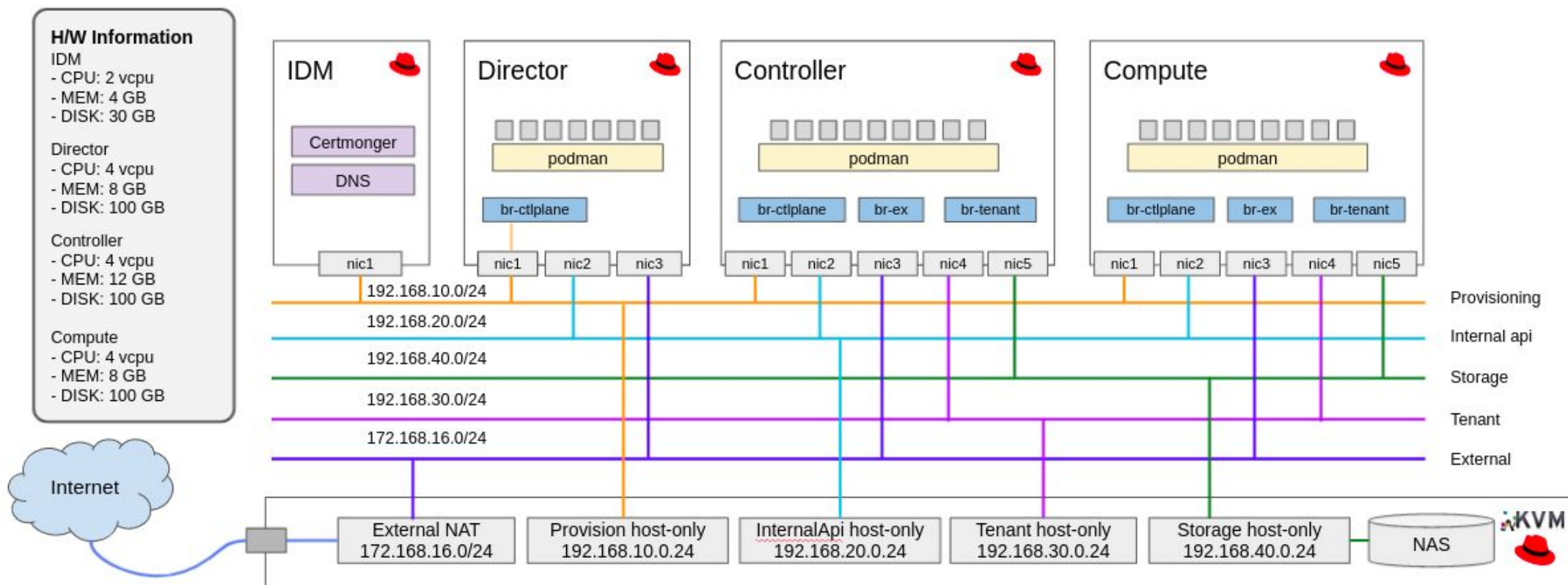
운영에는 어떤 일들이 있을까요?

주요 모니터링 항목을 알아보시다!

모니터링 항목	설명
Power	컨트롤러 및 컴퓨트 노드의 전원에 이상이 있는지 IPMI 명령어를 통해 확인합니다.
Network	다운되어 있는 네트워크 인터페이스가 있는지 확인하고, 다운되어 있는 인터페이스가 있다면 데이터센터 방문점검을 해야 합니다.
Cluster status	클러스터 관리 툴인 Pacemaker 상태를 점검합니다.
CPU	CPU 사용률을 확인합니다.
Memory	메모리 상태를 확인하고 캐시 메모리가 얼마나 찼는지를 확인합니다.
Container	컨테이너 실행상태를 점검합니다.
Mount NFS	Glance 에 잘 마운트 되어 있는지 확인하고, 마운트에 문제가 있다면 해당 구간의 네트워크를 확인해야 합니다.
Service Logs	서비스 로그를 확인하고 문제가 있는 서비스가 있는지 점검합니다.

운영에는 어떤 일들이 있을까요?

오픈스택은 주로 다음과 같은 노드들이 필요합니다.

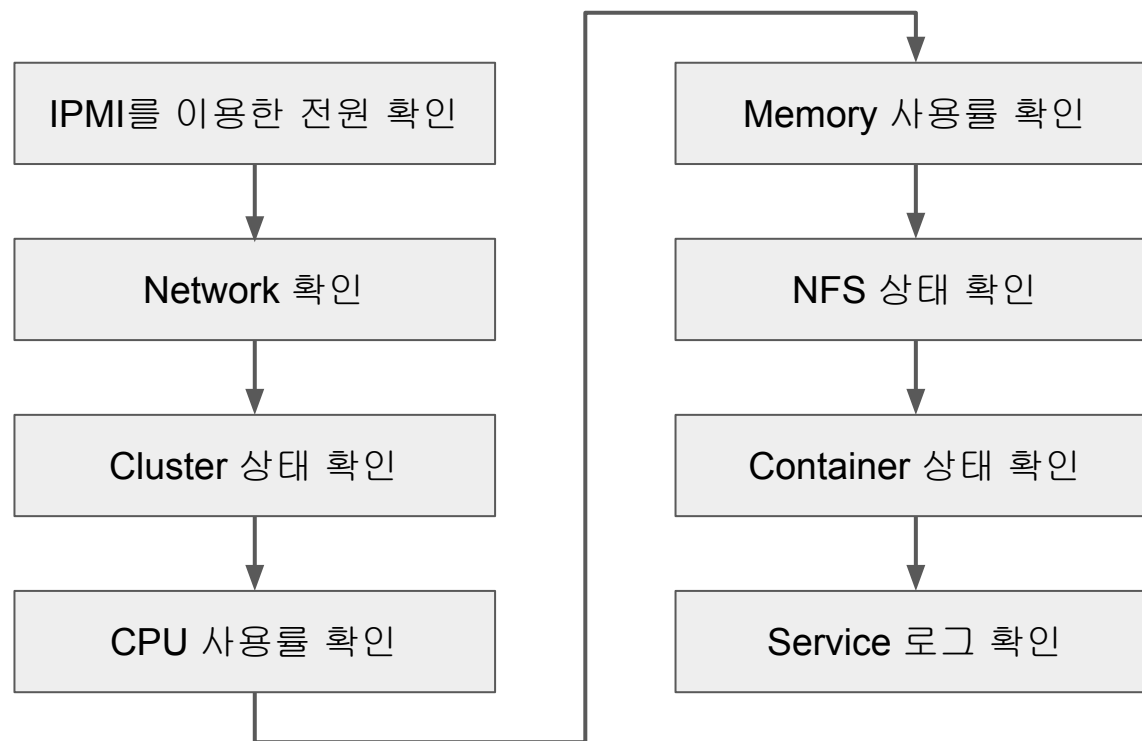


셀 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기

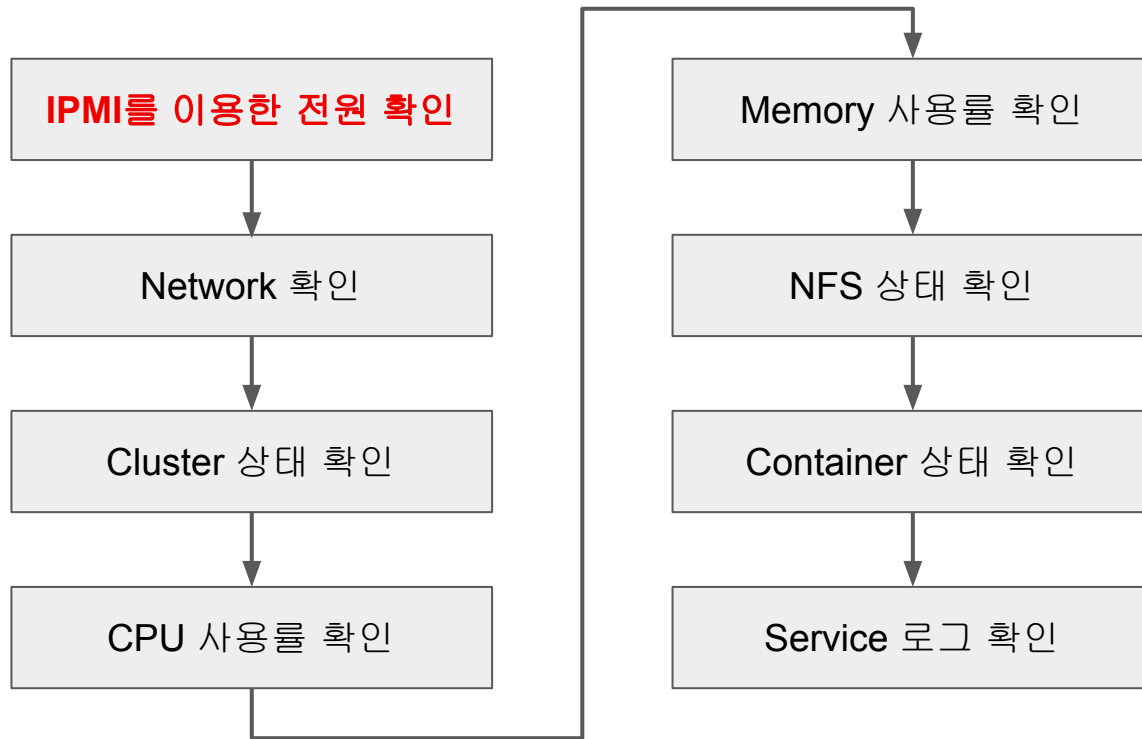


시스템 모니터링



셀 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



● 셀 스크립트로 구현하기

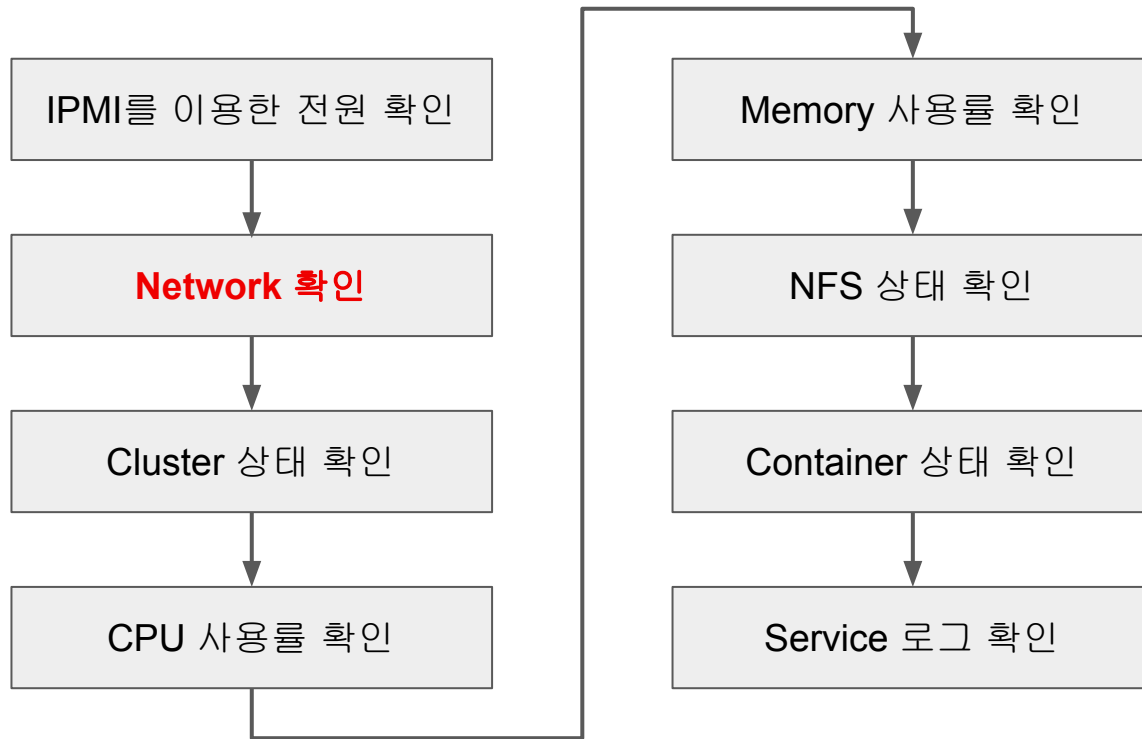
```

print_msg "#-----"
print_msg "# Check Overcloud Power"
print_msg "#-----"

for i in {0..2}
do
    print_msg "192.168.10.1:645$i"
    power_stat=$(ipmitool -H 192.168.10.1 -p 645$i
-I lanplus -U admin -P redhat power status)
    print_msg "$power_stat"
done
    
```

셸 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



● 셸 스크립트로 구현하기

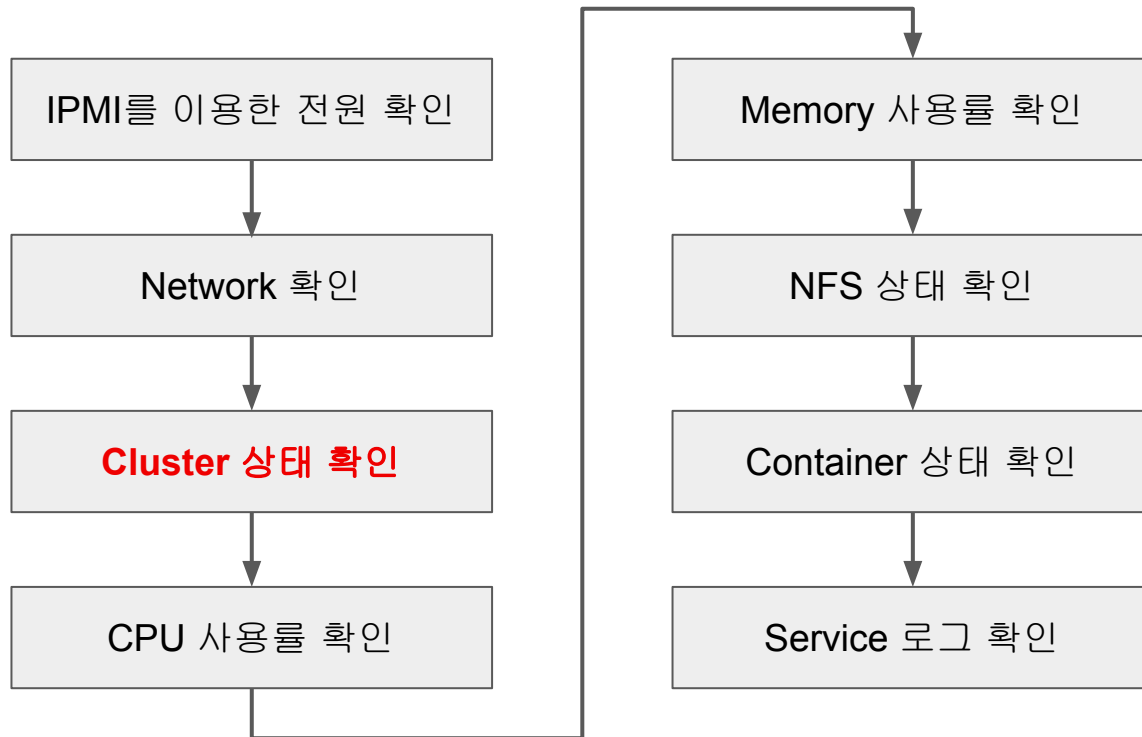
```

print_msg "#-----"
print_msg "# Check Network"
print_msg "#-----"
net_stat=$(ssh -q heat-admin@$i sudo ip a | grep
'state UP' | grep mq | wc -l)

if [ $net_stat -eq 5 ]
then
    print_msg "Network status is normal"
else
    print_msg "Please check network status"
    print_msg "$(ssh -q heat-admin@$i sudo ip a)"
fi
    
```

셀 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



● 셀 스크립트로 구현하기

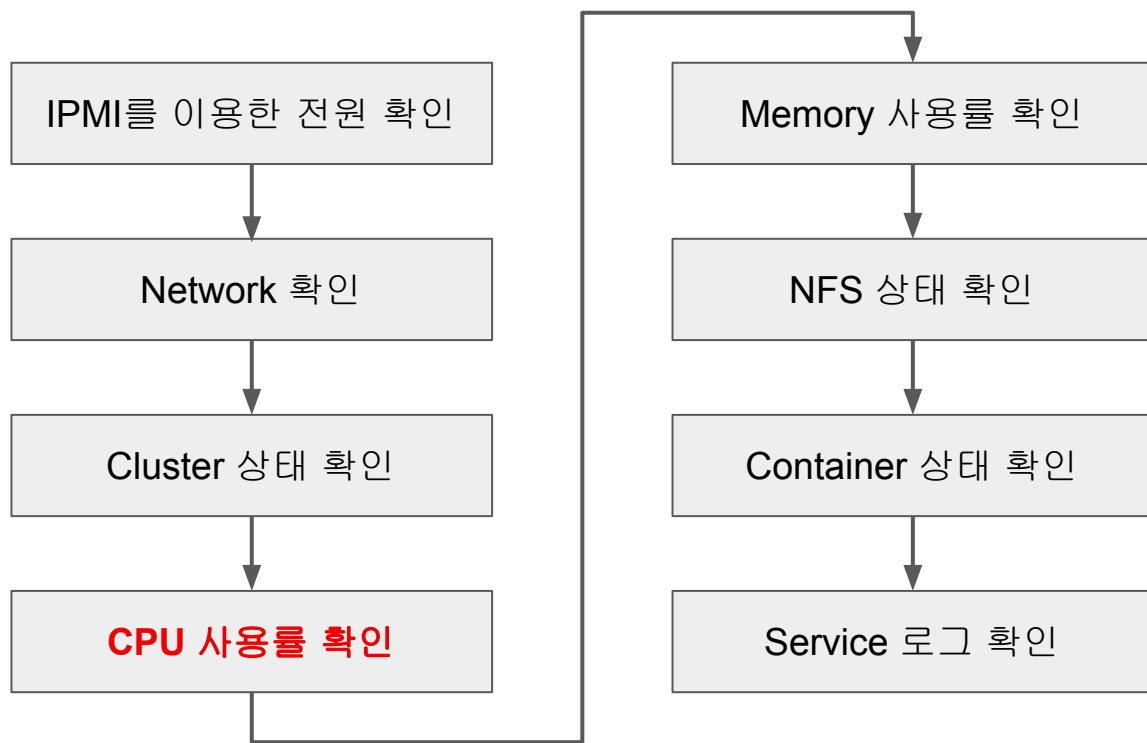
```

print_msg "#-----"
print_msg "# Check Clustering"
print_msg "#-----"
cluster_stat=$(ssh -q heat-admin@$i sudo pcs status
| grep 'failed' | wc -l)

if [ $cluster_stat -eq 0 ]
then
    print_msg "Pacemaker status is normal"
else
    print_msg "Please check pacemaker"
    print_msg "$(ssh -q heat-admin@$i sudo pcs
status)"
fi
    
```

셸 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



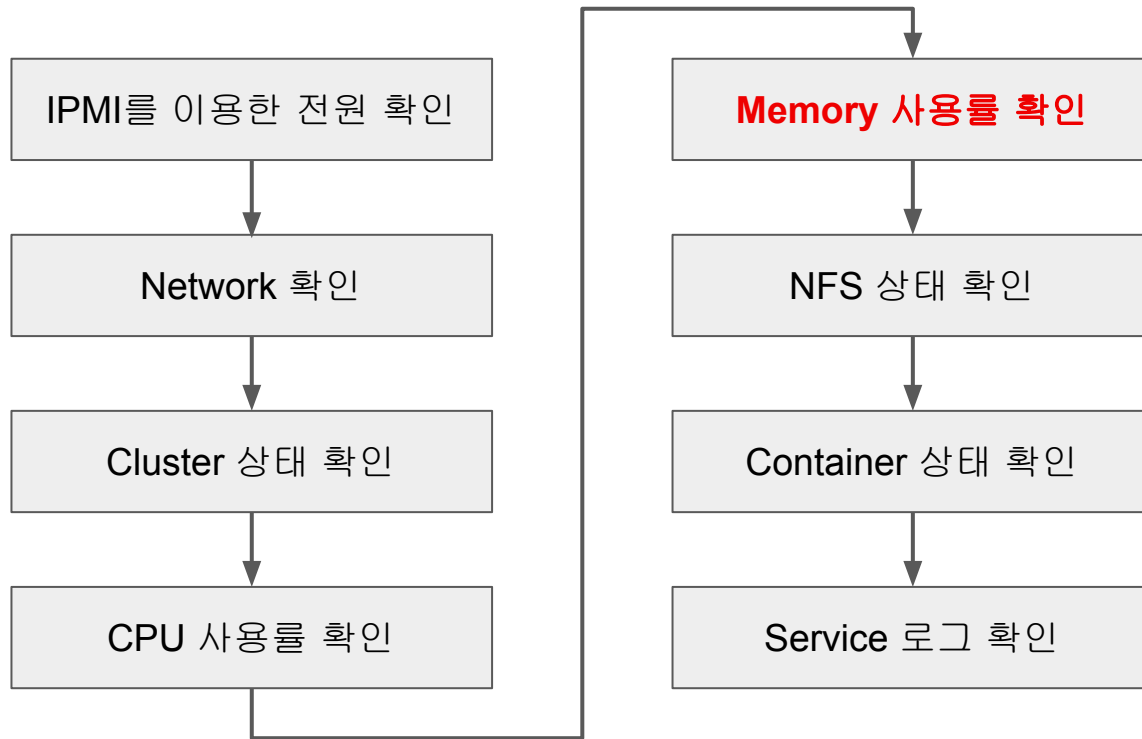
● 셸 스크립트로 구현하기

```

print_msg "#-----"
print_msg "# Check CPU"
print_msg "#-----"
cpu_stat=$(ssh -q heat-admin@$i sudo
mpstat | grep all | awk '{print $4}')
print_msg "CPU usage is $cpu_stat. If CPU
usage is high, please check system CPU
status"
    
```

셀 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



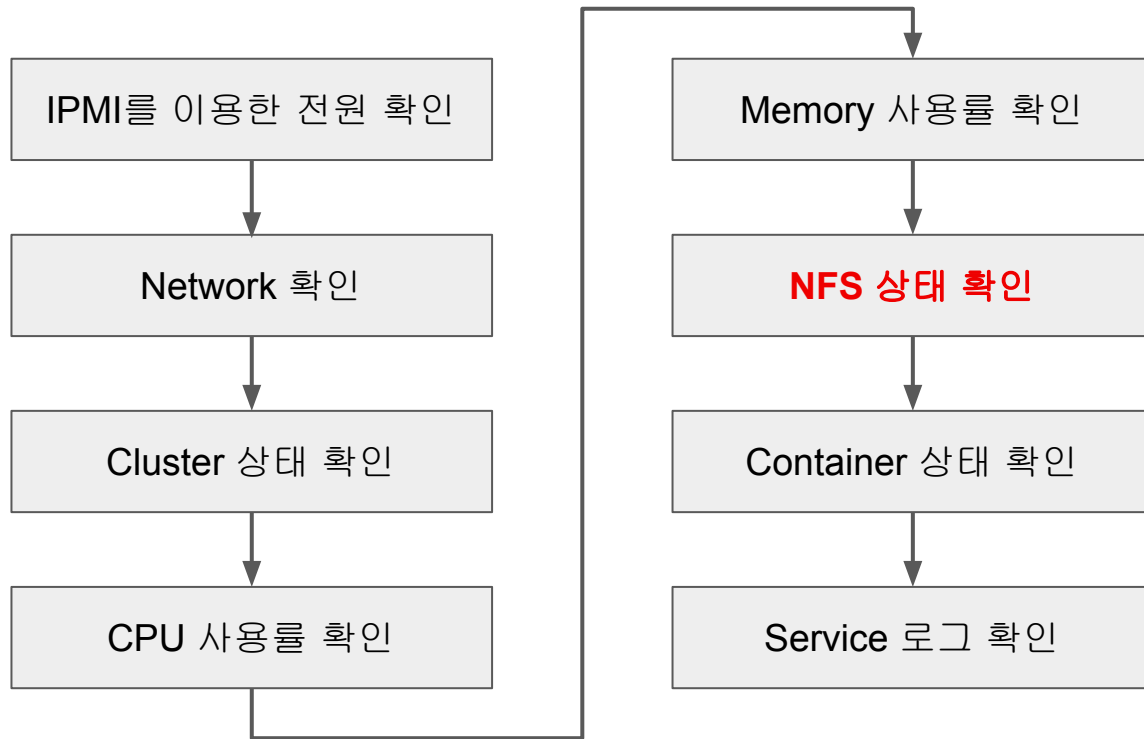
● 셀 스크립트로 구현하기

```

print_msg "#-----"
print_msg "# Check Memory"
print_msg "#-----"
mem_stat=$(ssh -q heat-admin@$i sudo free -h |
grep -i mem | awk '{print $4}')
print_msg "Free memory amount is $mem_stat. If
free memory amount is low, please check system
memory status"
    
```

셀 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



● 셀 스크립트로 구현하기

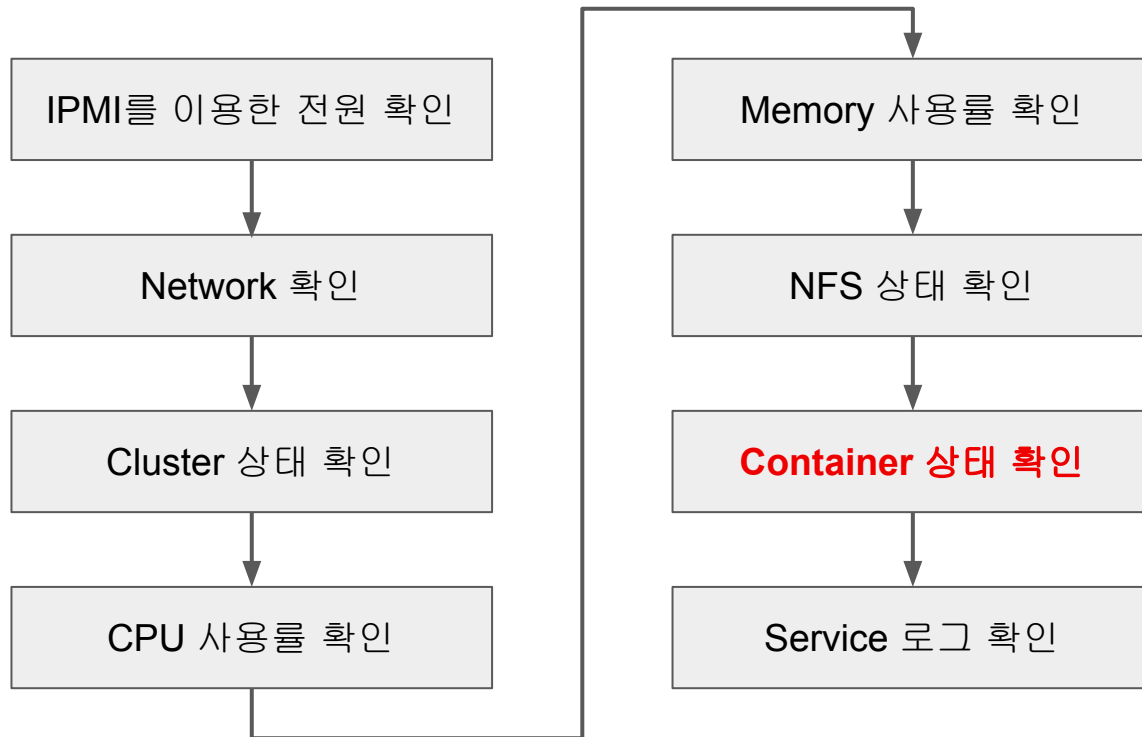
```

print_msg "#-----"
print_msg "# Check NFS - glance"
print_msg "#-----"
nfs_stat=$(ssh -q heat-admin@$i sudo df -h | grep
glance | wc -l)

if [ $nfs_stat -eq 1 ]
then
    print_msg "NFS status is normal."
else
    print_msg "Please check network status and nfs
status"
fi
    
```

셸 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



● 셸 스크립트로 구현하기

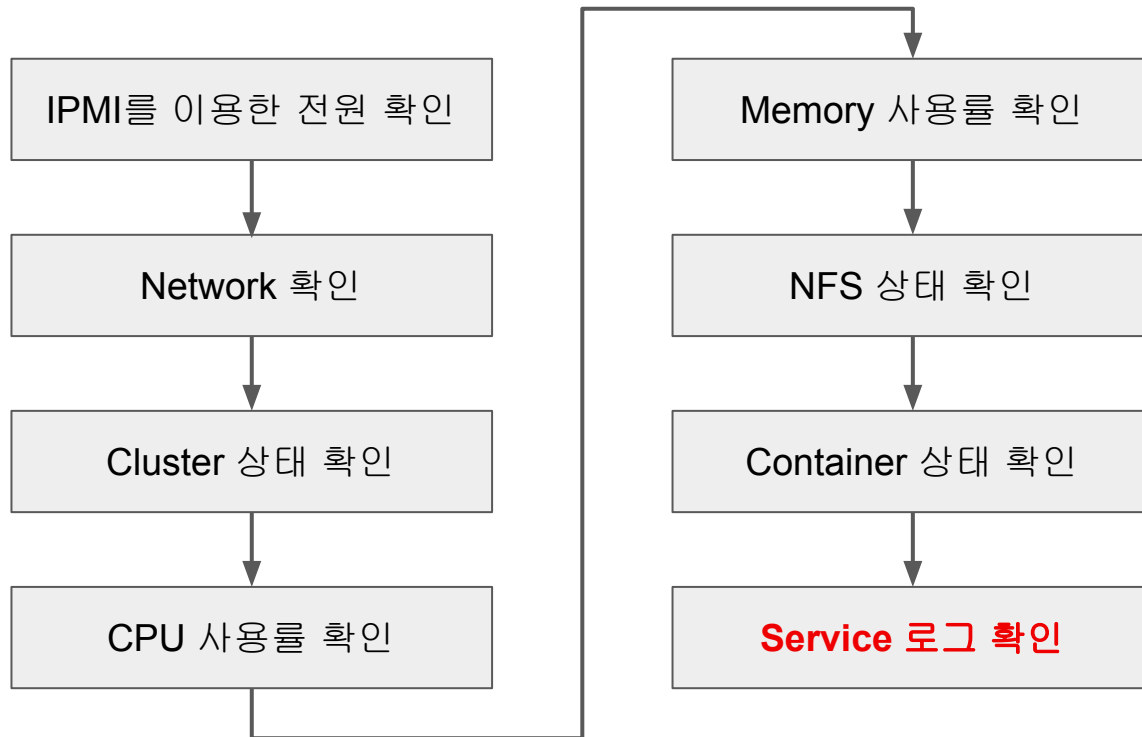
```

print_msg "#-----"
print_msg "# Check Container"
print_msg "#-----"
container_stat=$(ssh -q heat-admin@$i sudo podman ps |
grep -i sec | wc -l)

if [ $container_stat -eq 0 ]
then
    print_msg "Container status is normal."
else
    print_msg "Please check container status"
    print_msg "$(ssh -q heat-admin@$i sudo podman ps)"
fi
    
```

셸 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



● 셸 스크립트로 구현하기

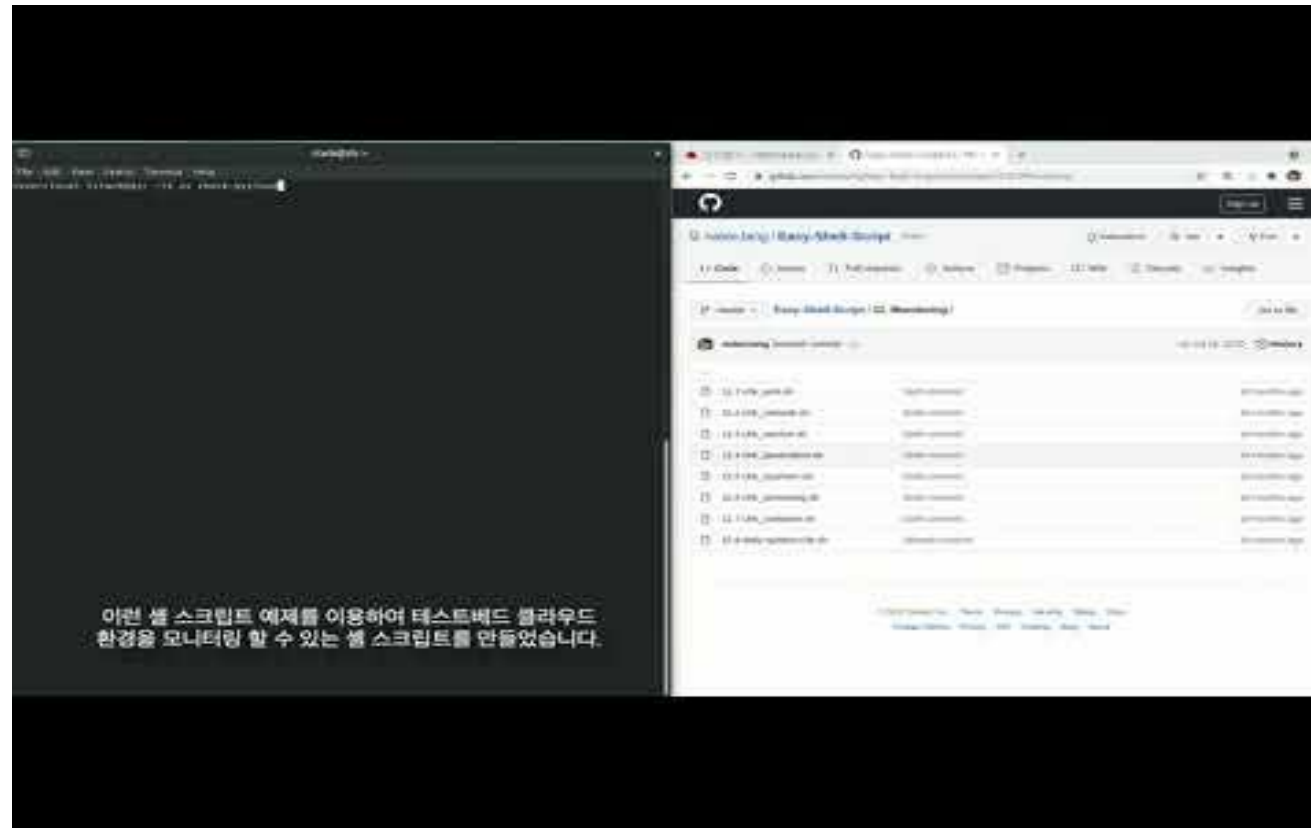
```

print_msg "#-----"
print_msg "# Check Service logs"
print_msg "#-----"
log_stat=$(ssh -q heat-admin@$i sudo sh chk-log.sh | wc -l)

if [ $log_stat -eq 0 ]
then
    print_msg "No error service logs. This system status is
normal."
else
    error_msg=$(ssh -q heat-admin@$i sudo sh chk-log.sh)
    print_msg "Please check system logs and container status."
    print_msg $error_msg
fi
    
```

셀 스크립트를 이용하여 모니터링하기

모니터링을 위한 프로세스 만들기



생활속의 클라우드

- 주위를 둘러보면 쉽게 클라우드 환경을 만날 수 있습니다.
- 클라우드 환경은 하이퍼바이저 환경에서 컨테이너 환경을 넘어가고 있습니다.

대표적인 Private 클라우드 환경

- Private 클라우드 환경을 구축하기 위해 가장 많이 사용되는 오픈소스 플랫폼이 오픈스택입니다.

클라우드! 리눅스! 셸 스크립트!

- Public 클라우드가 되든 Private 클라우드가 되든 우리는 리눅스 서버를 정말 많이 사용합니다.
- 리눅스는 셸로 구성되어 있고, 자주 사용하는 명령어를 나열하여 파일로 저장하면 셸 스크립트가 됩니다.

셸 스크립트를 활용한 클라우드 시스템 운영

- 자주 사용하는 리눅스 명령어와 셸 스크립트의 간단한 제어문을 이용해 클라우드 시스템을 효율적으로 운영할 수 있습니다.
- 인스턴스를 생성/삭제 할 수 있으며, 호스트 노드에서 매일 체크해야 하는 항목들을 쉽게 확인할 수 있습니다.

Thank you

OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.

 openstack.kr

 <http://www.openstack.or.kr>

 <http://bit.ly/openstack-kr-videos>

