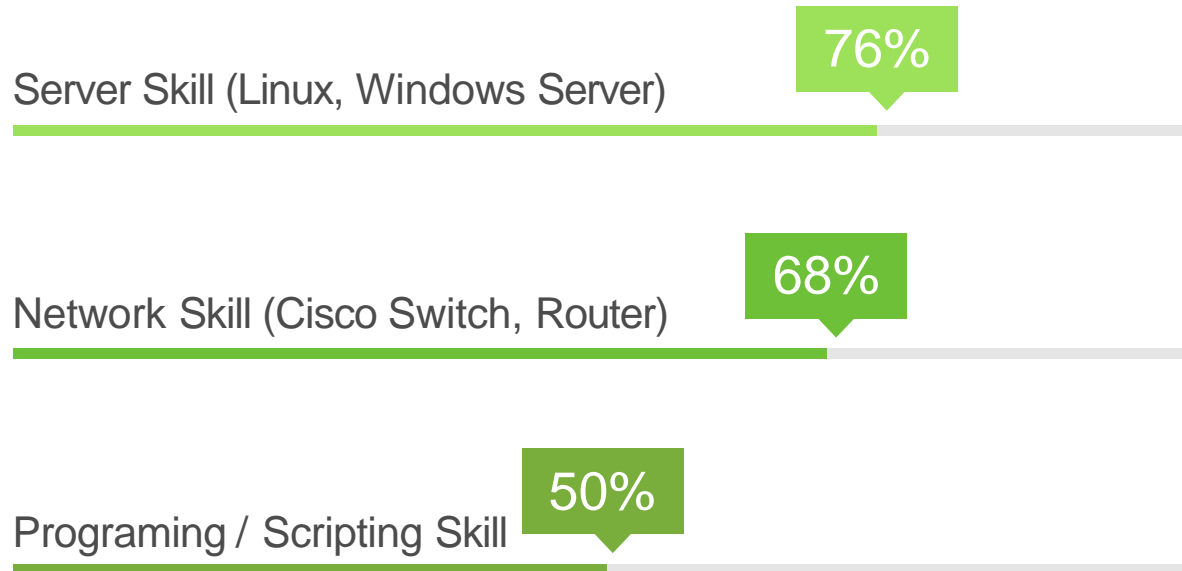


꿈꾸는 시스템엔지니어

시스템을 다루는것을 좋아하고, 새로운것을 알아가는게 좋은
19살 시스템엔지니어





받은 교육

세명컴퓨터고등학교 보안솔루션

(주)케이①지아①티뱅크

- 리눅스 서버
- 윈도우서버
- 네트워크 기초
- 네트워크보안
- 웹 보안
- CCNP

01



취득한 자격증

GTQ(그래픽기술자격) 1급

워드프로세서1급

인터넷정보관리사2급

리눅스마스터2급

네트워크관리사

정보처리기능사

02



공부한 내용

무료 웹호스팅서비스운영

cPanel 웹호스팅서비스운영

작구 통관어플 개발(졸업작품)

- 개발 총괄
- Swift 사용 개발
- APNS 구축
- 파싱서버 구축

03



보유기술

리눅스서버구축 및유지보수

윈도우서버구축 및유지보수

Apache 구축 및튜닝, 유지보수

Nginx 구축 및튜닝, 유지보수

IIS, DNS 구축 및유지보수

AD 구축 및유지보수

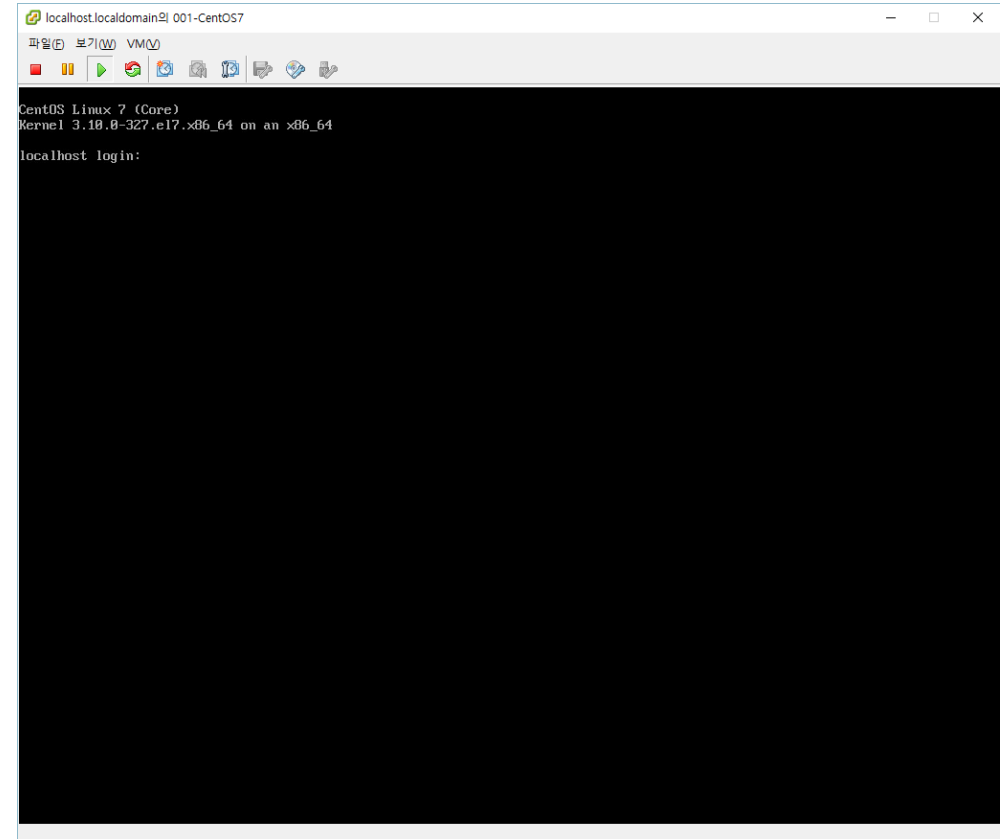
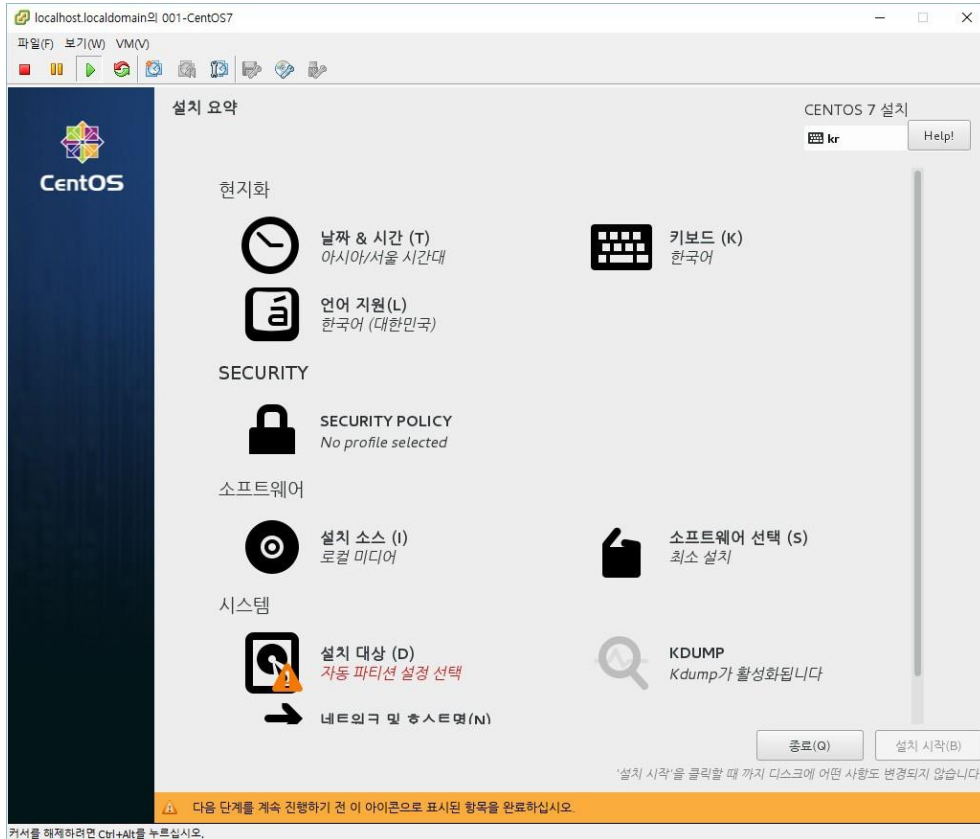
Exchange 구축 및유지보수

C=중, JAVA=중, PHP=중

Bash=기본, PowerShell=기본

04

[CentOS 7] 서버구축



ESXi 하①퍼바①자에 필요한 설정을 사전설정으로 선택하고 CentOS7을 설치했습니다.

자세한 설명은 블로그에 있습니다. BI
og : <https://jhok.kr/archives/322>

[CentOS 7] WebTatic 리포지토리 설치

```
Jaehyeok — root@localhost:~ — ssh root@121.131.239.124 — 80x24
[root@localhost ~]# rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release-7-3.noarch.rpm
https://mirror.webtatic.com/yum/el7/webtatic-release-7-3.noarch.rpm(을)를 복구합니다
경고 : /var/tmp/rpm-tmp.dKpMXq: Header V4 RSA/SHA1 Signature, key ID 62e74ca5: NO KEY
준비 중 ... ##### [100%]
Updating / installing...
 1:webtatic-release-7-3 ##### [100%]
[root@localhost ~]# rpm -qa | grep webtatic
webtatic-release-7-3.noarch
[root@localhost ~]#
```

```
Jaehyeok — root@localhost:~ — ssh root@121.131.239.124 — 80x24
[root@localhost ~]# yum repolist
Loaded plugins: fastestmirror
epel/x86_64/metalink | 4.1 kB 00:00
epel | 4.3 kB 00:00
webtatic | 3.6 kB 00:00
(1/5): epel/x86_64/group_gz | 170 kB 00:00
(2/5): epel/x86_64/updateinfo | 565 kB 00:00
(3/5): webtatic/x86_64/group_gz | 448 B 00:00
(4/5): webtatic/x86_64/primary_db | 120 kB 00:01
(5/5): epel/x86_64/primary_db | 4.1 MB 00:05
Loading mirror speeds from cached hostfile
* base: mirror.oasis.onnetcorp.com
* epel: mirror.premi.st
* extras: mirror.oasis.onnetcorp.com
* updates: mirror.oasis.onnetcorp.com
* webtatic: sp.repo.webtatic.com
repo id repo name status
base/7/x86_64 CentOS-7 - Base 9,007
epel/x86_64 Extra Packages for Enterprise Linux 7 - x86_64 10,115
extras/7/x86_64 CentOS-7 - Extras 305
updates/7/x86_64 CentOS-7 - Updates 1,687
webtatic/x86_64 Webtatic Repository EL7 - x86_64 272
repolist: 21,386
[root@localhost ~]#
```

최신버전의패키지를 Yum을 사용하여 설치하기 위해서 WebTatic 리포지토리를 설치했습니다.

안정화 버전의 패키지를 사용하는것도 중요하지만 호스팅용 웹서버특성상 사용자들①요구하는 버전에 맞추기 위해서 사용자들에 의해서 검
중된 버전을 설치할수 있는 리포지토리를 설정했습니다.

자세한 설명은 블로그에 있습니다. BI
og : <https://jhok.kr/archives/361>

[CentOS 7] Remi 리포지토리 설치

```
Jaehyeek — root@localhost:~ — ssh root@121.131.239.125 — 80x24
[[root@localhost ~]# rpm -Uvh http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
http://rpms.famillecollet.com/enterprise/remi-release-7.rpm(을)를 복구합니다
경고 : /var/tmp/rpm-tmp.4ApC0j: Header V3 DSA/SHA1 Signature, key ID 00f97f56: NO KEY
준비 중... ##### [100%]
Updating / installing...
 1:remi-release-7.1-3.el7.rem            ##### [100%]
[[root@localhost ~]# rpm -qa | grep remi*
remi-release-7.1-3.el7.rem.noarch
[[root@localhost ~]#
```

```
Jaehyeek — root@localhost:~ — ssh root@121.131.239.125 — 80x24
[[root@localhost ~]# yum repolist
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.premi.st
 * epel: mirror.premi.st
 * extras: mirror.premi.st
 * remi-safe: mirror.innosol.asia
 * updates: mirror.premi.st
repo id      repo name      status
base/7/x86_64  CentOS-7 - Base      9,007
epel/x86_64   Extra Packages for Enterprise Linux 7 - x86_64  10,126
extras/7/x86_64 CentOS-7 - Extras    310
remi-safe     Safe Remi's RPM repository for Enterprise Linux 7 - x86_64  948
updates/7/x86_64 CentOS-7 - Updates  1,687
repolist: 22,078
[[root@localhost ~]#
```

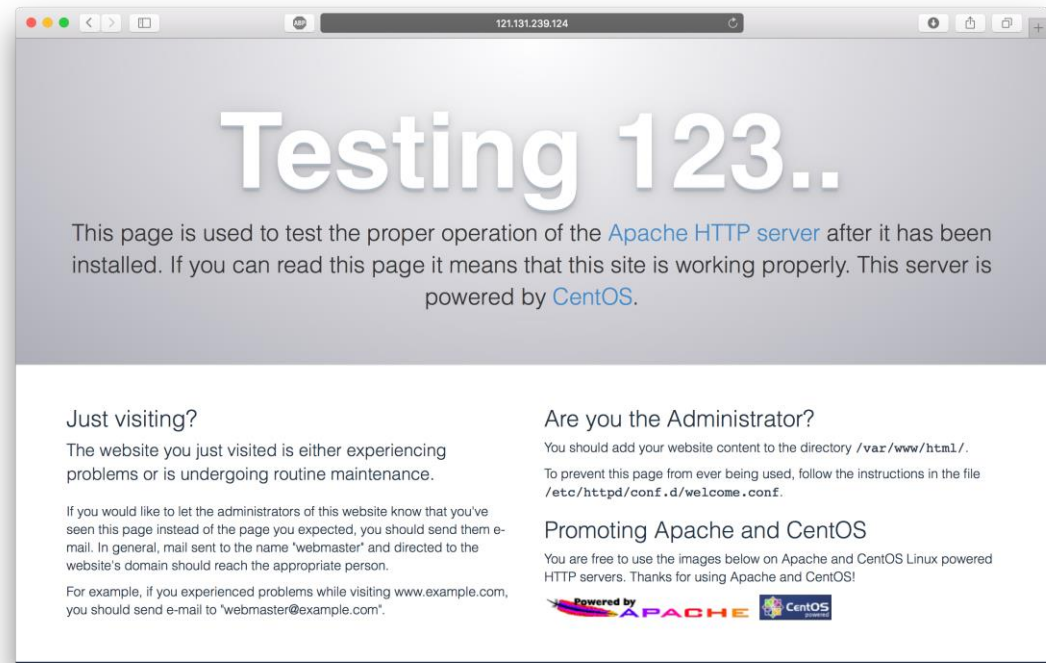
최신버전의패키지를 Yum을 사용하여 설치하기 위해서 Remi 리포지토리를 설치했습니다.

안정화 버전의 패키지를 사용하는것도 중요하지만 호스팅용 웹서버 특성상 사용자들①요구하는 버전에 맞추기 위해서 사용자들에 의해서 검증된 버전을 설치할수 있는 리포지토리를 설정했습니다.

자세한 설명은 블로그에 있습니다. BI
og : <https://jhok.kr/archives/418>

[CentOS 7] LAMP 구축

```
Jaehyeok — root@localhost:~ — ssh root@121.131.239.124 — 80x24
[root@localhost ~]# yum -y install httpd mariadb mariadb-server php56w php56w-mysql.x86_64
```



Apache, MariaDB, PHP 설치 해보고 설치한 Apache의 Virtual Host와 CentOS Quota를 ①용해서 웹호스팅을 제공해봤습니다.

웹호스팅을 제공하면서처리, 응답 속도를 개선하기 위해서 최적화도 같이 ①진행해봤습니다.

자세한 설명은 블로그에 있습니다. BI
og : <https://jhok.kr/archives/380>

[CentOS 7] Apache 튜닝

```
Jaehyeok — root@localhost:~ — ssh root@121.131.239.124 — 80x24
SSL/TLS Protocol:      TLSv1.2,ECDHE-RSA-AES128-GCM-SHA256,2048,128
Document Path:        /
Document Length:      52439 bytes

Concurrency Level:    1
Time taken for tests: 0.961 seconds
Complete requests:    1
Failed requests:      0
Write errors:         0
Total transferred:    52723 bytes
HTML transferred:    52439 bytes
Requests per second: 1.04 [#/sec] (mean)
Time per request:     960.908 [ms] (mean)
Time per request:     960.908 [ms] (mean, across all concurrent requests)
Transfer rate:        53.58 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    19    19   0.0    19    19
Processing: 942   942   0.0   942   942
Waiting:    488   488   0.0   488   488
Total:      961   961   0.0   961   961
[root@localhost ~]#
```

```
Jaehyeok — root@localhost:~ — ssh root@121.131.239.124 — 80x24
SSL/TLS Protocol:      TLSv1.2,ECDHE-RSA-AES128-GCM-SHA256,2048,128
Document Path:        /
Document Length:      52583 bytes

Concurrency Level:    1
Time taken for tests: 0.029 seconds
Complete requests:    1
Failed requests:      0
Write errors:         0
Total transferred:    52836 bytes
HTML transferred:    52583 bytes
Requests per second: 34.31 [#/sec] (mean)
Time per request:     29.142 [ms] (mean)
Time per request:     29.142 [ms] (mean, across all concurrent requests)
Transfer rate:        1770.56 [Kbytes/sec] received

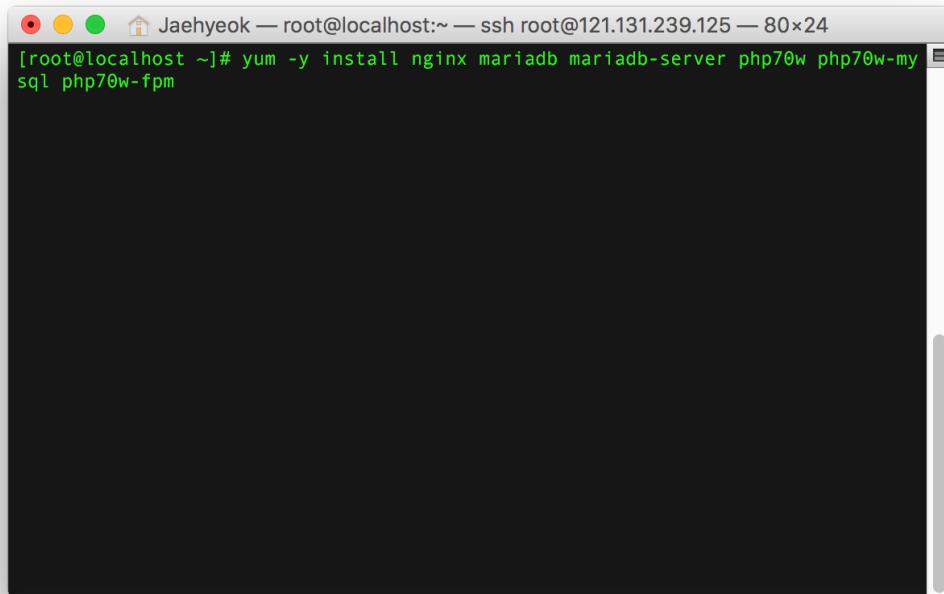
Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    19    19   0.0    19    19
Processing: 10    10   0.0    10    10
Waiting:     9     9   0.0     9     9
Total:      29    29   0.0    29    29
[root@localhost ~]#
```

Apache MPM을 Prefork 방식에서 Worker 방식으로 변경하고, MaxClient 값을 예상하여 컴파일 하였습니다.

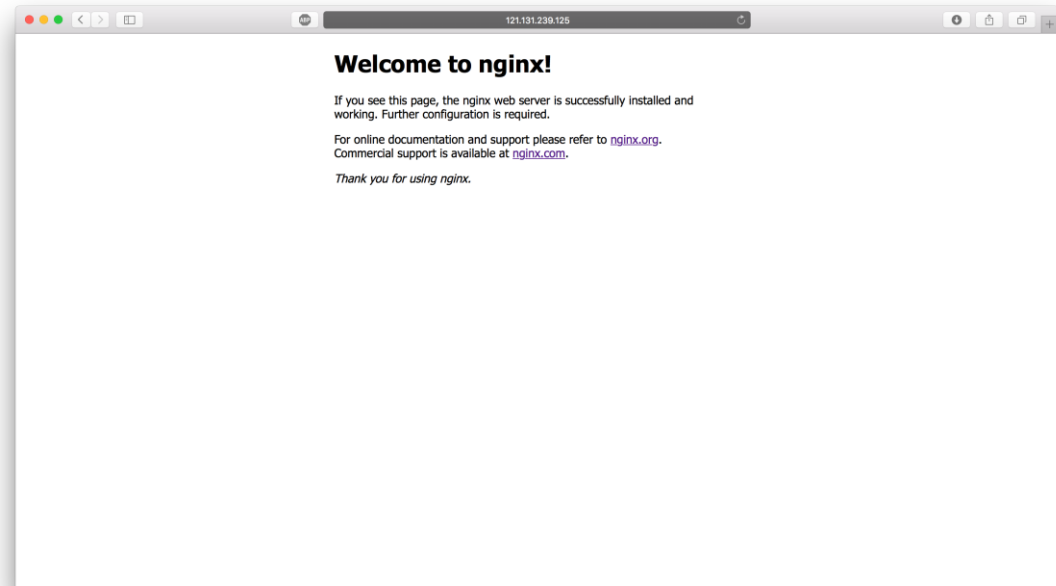
KeepAlive Timeout 값을 수정하고, PHP 캐시와 Client Browser 캐시를 사용하게하여 전반적인 웹서버의 속도를 높였습니다.

Apache 웹서버의 설정을 올바르게 변경하고 튜닝을 진행하였더니 프로세싱, 대기 시간① 획기적으로 줄어서 쾌적한 환경① 되었습니다.

[CentOS 7] LEMP 구축



```
Jaehyeok — root@localhost:~ — ssh root@121.131.239.125 — 80x24
[root@localhost ~]# yum -y install nginx mariadb mariadb-server php70w php70w-mysql php70w-fpm
```



Nginx, MariaDB, PHP, PHP-FPM을 설치하여 기존 Apache 서버와 성능을 비교해보고
프록시서버와 Nginx 웹서버를 별도로 구축하여 리버스프록시 적용해 로드밸런싱 테스트도 진행해봤습니다.

자세한 설명은 블로그에 있습니다. BI
og : <https://jhok.kr/archives/436>

[CentOS 7] 스크립트 만들기- 1

```
#!/bin/bash
day=$(date +%d)
if [ $month == Jul ]; then
    month=Jun
    day=30
elif [ $month == Aug ]; then
    month=Jul
    day=31
elif [ $month == Sep ]; then
    month=Aug
    day=30
elif [ $month == Oct ]; then
    month=Sep
    day=30
elif [ $month == Nov ]; then
    month=Oct
    day=31
else
    month=Nov
    day=30
fi

# Let "dayday-1"
echo "day"

# Nginx Log Check
echo "# Nginx Log #> /root/report/Server_log_report_STODAY.txt
cat /var/log/nginx/access_log | sed $(date +%p)$(date +%H)$(date +%d) >> /root/report/Server_log_report_STODAY.txt
echo " " >> /root/report/Server_log_report_STODAY.txt

# Mariadb Log Check
echo "# Mariadb Log #> /root/report/Server_log_report_STODAY.txt
cat /var/log/mariadb/mariadb.log | grep "10001" >> /root/report/Server_log_report_STODAY.txt
echo " " >> /root/report/Server_log_report_STODAY.txt

# PHP-FPM Log Check
echo "# PHP-FPM Log #> /root/report/Server_log_report_STODAY.txt
cat /var/log/php-fpm/error_log | grep "$day-$month-$0" >> /root/report/Server_log_report_STODAY.txt
echo " " >> /root/report/Server_log_report_STODAY.txt

# Secure Log Check
echo "# Secure Log #> /root/report/Server_log_report_STODAY.txt
A cat /var/log/secure | grep "$month $day" >> /root/report/Server_log_report_STODAY.txt
cat /var/log/secure | grep Accepted | awk '{print $9"-"$11"-"$14}' | sort | uniq >> /root/report/Server_log_report_STODAY.txt
echo " " >> /root/report/Server_log_report_STODAY.txt

# Spooler(cris) Log Check
echo "# Spooler(cris) #> /root/report/Server_log_report_STODAY.txt
cat /var/log/spooler | grep "$month $day" >> /root/report/Server_log_report_STODAY.txt
echo " " >> /root/report/Server_log_report_STODAY.txt

# Messages Log Check
echo "# Messages Log #> /root/report/Server_log_report_STODAY.txt
```

```
#!/bin/bash
today=$(date +%Y%m-%d-%S)
echo "# $today server daily report" > /root/report/Server_report_STODAY.txt
echo "# Server Information #> /root/report/Server_report_STODAY.txt
echo "===== " >> /root/report/Server_report_STODAY.txt
# Hostname Output
echo "# Hostname : " >> /root/report/Server_report_STODAY.txt
hostname >> /root/report/Server_report_STODAY.txt
# Server Uptime
echo "# Uptime : " >> /root/report/Server_report_STODAY.txt
uptime | grep -ohe "up " | sed 's//g' | awk '{print $2}' >> /root/report/Server_report_STODAY.txt
# Number of Login Users
echo "# Number of Login Users : " >> /root/report/Server_report_STODAY.txt
uptime | grep -ohe "(0-9+)" users: | >> /root/report/Server_report_STODAY.txt
# Server IP Output
echo "# Server IP : " >> /root/report/Server_report_STODAY.txt
ip addr | grep "inet " | grep brd | awk '{print $2}' | awk -F/ '{print $1}' >> /root/report/Server_report_STODAY.txt
echo "===== " >> /root/report/Server_report_STODAY.txt
# Disk Status
echo "# Disk Status #> /root/report/Server_report_STODAY.txt
echo "# Disk status #> /root/report/Server_report_STODAY.txt
df -h >> /root/report/Server_report_STODAY.txt
echo " " >> /root/report/Server_report_STODAY.txt
# Memory Status
echo "# Memory Status #> /root/report/Server_report_STODAY.txt
free -h >> /root/report/Server_report_STODAY.txt
echo " " >> /root/report/Server_report_STODAY.txt
# Network Status
echo "# Network Status (port) #> /root/report/Server_report_STODAY.txt
netstat -an | grep LISTEN >> /root/report/Server_report_STODAY.txt
echo " " >> /root/report/Server_report_STODAY.txt
# Zombie
echo "# Zombie Count #> /root/report/Server_report_STODAY.txt
ps -ef | grep defunct | grep -v grep | wc -l >> /root/report/Server_report_STODAY.txt
echo " " >> /root/report/Server_report_STODAY.txt
echo "# Zombie List #> /root/report/Server_report_STODAY.txt
ps -ef | grep defunct | grep -v grep >> /root/report/Server_report_STODAY.txt
echo " " >> /root/report/Server_report_STODAY.txt
# Temp Check(/tmp)
echo "# Temp Check #> /root/report/Server_report_STODAY.txt
ls -la /tmp >> /root/report/Server_report_STODAY.txt
echo " " >> /root/report/Server_report_STODAY.txt
```

CentOS 7 서버의 관리성을 향상시키고 보안에 있어서 놓치는 부분①생기는걸 방지하기 위해서

Bash Shell을 ①용해서 스크립트를 작성해봤습니다.

필요한 기능을 나누어스크립트를 작성했고, Crontab을 ①용해서 일정시간에 스크립트를 실행해 ①메일로 결과를 발송하도록 했습니다.

https://github.com/JaehyeokChoi/shell/blob/master/Server_check.sh https://github.com/JaehyeokChoi/shell/blob/master/Server_log_check.sh https://github.com/JaehyeokChoi/shell/blob/master/Server_secure_check.sh

[CentOS 7] 스크립트 만들기 - 2

받은 편지함 모두 ▾

오늘

root
[001-CentOS7] 2016년-06월-16일 Server_secure
2016년-06월-16일 Server_secure daily report ## Serv...

root
[001-CentOS7] 2016년-06월-16일 Server_log d
2016년-06월-16일 Server_log daily report ## Server I...

root
[001-CentOS7] 2016년-06월-16일 Server daily r
2016년-06월-16일 Server daily report ## Server Infor...

최요일

최
네
미

월
최
네
미

일
최
[h
"C

[001-CentOS7] 2016년-06월-16일 Server daily report

root
오늘 오전 12:32
최재혁

2016년-06월-16일 Server daily report

```
## Server Information ##
=====
Hostname : localhost.localdomain
Uptime : 14:47
Number of Login Users : 1 user,
Server IP : 121.131.239.125
=====

## Disk Status ##
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3        17G  1.4G  15G   9% /
devtmpfs         911M   0 911M   0% /dev
tmpfs            920M   0 920M   0% /dev/shm
tmpfs            920M  49M 872M   6% /run
tmpfs            920M   0 920M   0% /sys/fs/cgroup
/dev/sda2        477M 126M 322M  29% /boot
/dev/sda1        200M  9.5M 191M   5% /boot/efi
tmpfs            184M   0 184M   0% /run/user/0

## Memory Status ##
              total  used  free  shared buff/cache available
Mem:          1.8G   298M   951M   48M   590M   1.3G
Swap:         2.0G     0B   2.0G

## Network Status (Port) ##
tcp 0 0 0.0.0.0:3306 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN
tcp 0 0 0.127.0.0:1:25 0.0.0.0:* LISTEN
```

받은 편지함 모두 ▾

오늘

root
[001-CentOS7] 2016년-06월-16일 Server_secure
2016년-06월-16일 Server_secure daily report ## Serv...

root
[001-CentOS7] 2016년-06월-16일 Server_log d
2016년-06월-16일 Server_log daily report ## Server I...

root
[001-CentOS7] 2016년-06월-16일 Server daily r
2016년-06월-16일 Server daily report ## Server Infor...

최요일

최
네
미

월
최
네
미

일
최
[h
"C

[001-CentOS7] 2016년-06월-16일 Server_secure daily report

root
오늘 오전 12:32
최재혁

2016년-06월-16일 Server_secure daily report

```
## Server Information ##
=====
Hostname : localhost.localdomain
Uptime : 14:47
Number of Login Users : 1 user,
Server IP : 121.131.239.125
=====

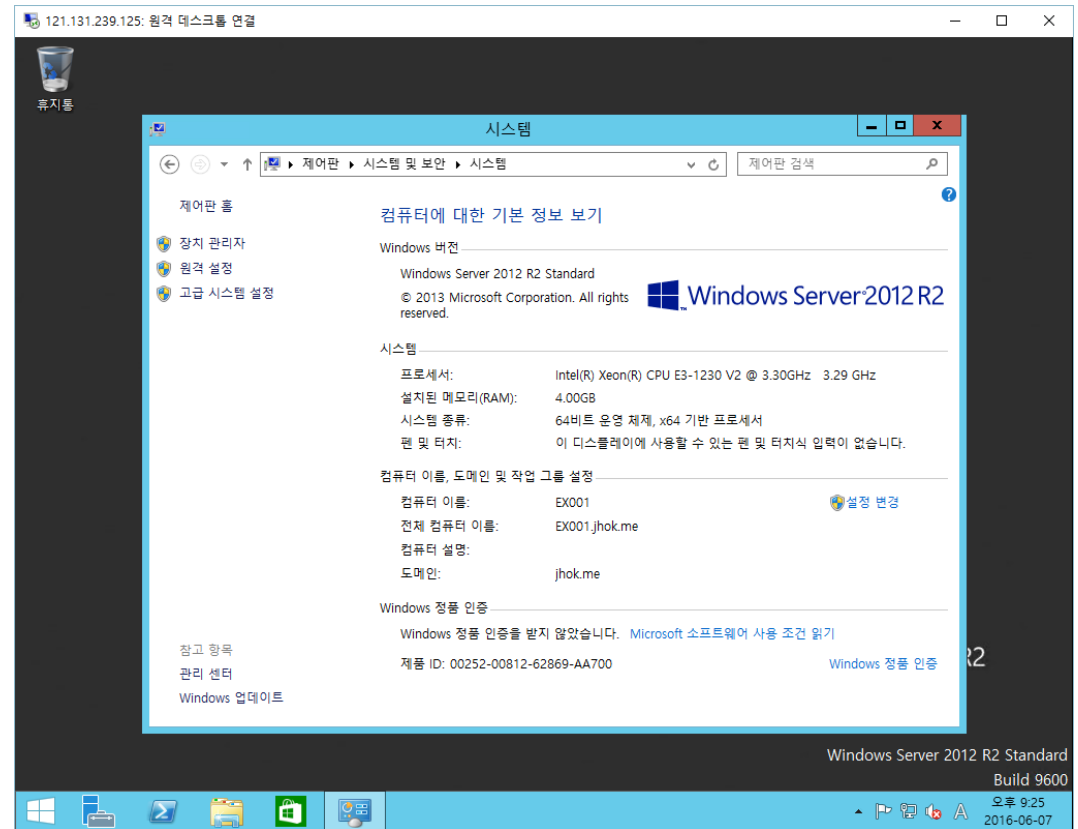
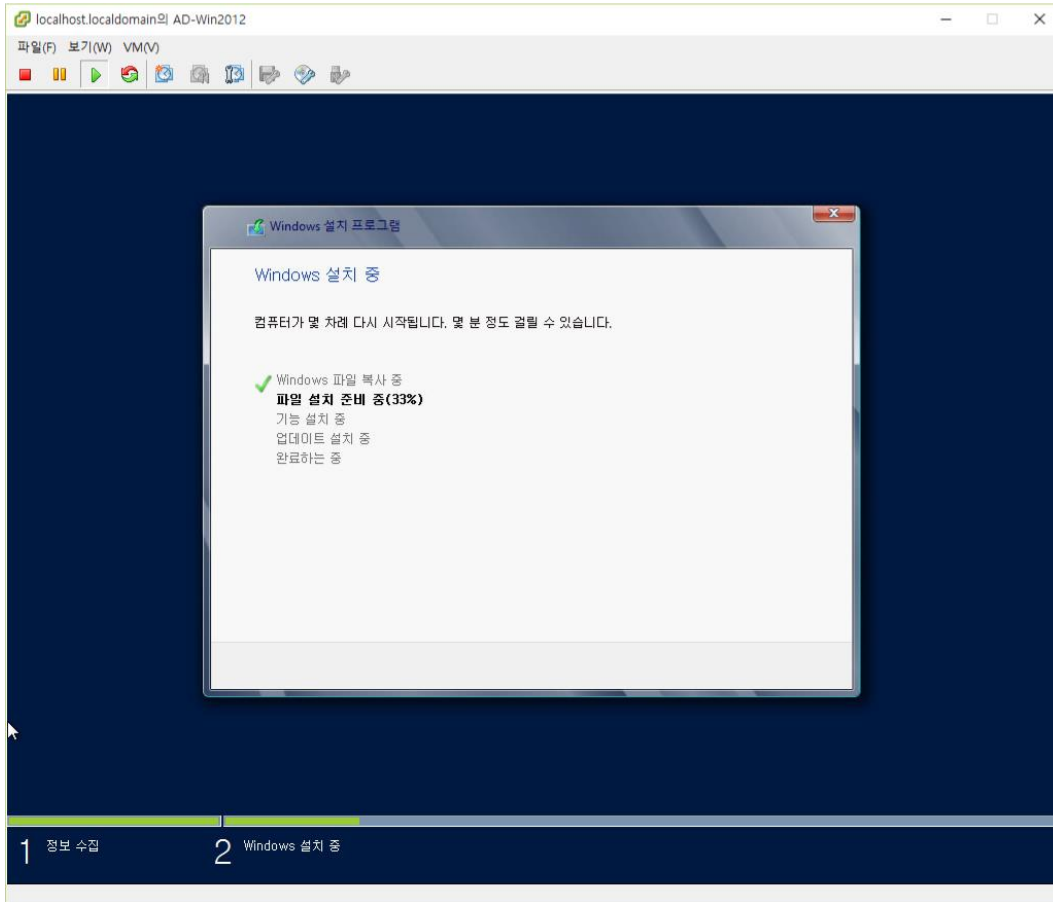
## SetUID, SetGID File Check ##
-rwsr-xr-x. 1 root root 31960  4월  1 02:37 /usr/bin/umount
--s-x-x. 1 root root 130720  4월  1 02:09 /usr/bin/sudo
-rwsr-xr-x. 1 root root 57552  4월  1 00:09 /usr/bin/crontab
-rwsr-xr-x. 1 root root 64200  3월  6 2015 /usr/bin/chage
--x-s-x. 1 root nobody 306304  3월 22 07:18 /usr/bin/ssh-agent
-rwsr-xr-x. 1 root root 27656  2월 17 00:45 /usr/bin/pkexec
-rwxr-sr-x. 1 root tty 19536  4월  1 02:37 /usr/bin/write
-rwsr-xr-x. 1 root root 78168  3월  6 2015 /usr/bin/gpasswd
-rws-x-x. 1 root root 23960  4월  1 02:37 /usr/bin/chfn
-rwsr-xr-x. 1 root root 27832  6월 10 2014 /usr/bin/passwd
-r-xr-sr-x. 1 root tty 15344  6월 10 2014 /usr/bin/wall
-rws-x-x. 1 root root 23856  4월  1 02:37 /usr/bin/chsh
-rwsr-xr-x. 1 root root 44232  4월  1 02:37 /usr/bin/mount
-rwsr-xr-x. 1 root root 32072  4월  1 02:37 /usr/bin/su
-rwxr-sr-x. 1 root mail 19800 11월 20 2015 /usr/bin/lockfile
-rwsr-xr-x. 1 root root 41752  3월  6 2015 /usr/bin/newgrp
-rwsr-xr-x. 1 root root 36264  8월 18 2015 /usr/sbin/umx_chkpwd
-rwxr-sr-x. 1 root root 11208  4월  1 02:39 /usr/sbin/netreport
-rwsr-xr-x. 1 root root 11208  8월 18 2015 /usr/sbin/pam_timestamp_check
-rwsr-xr-x. 1 root root 11272  4월  1 02:39 /usr/sbin/usernetctl
```

로그의 누락①없도록 00시가 넘어가면 전날 로그를 필터링해 메일로 전송하게 했습니다.

다량의 IO가 발생하는 스크립트(Find) Crontab에서 시간을 다르게 설정하여 사용자의 불편함을 줄였습니다.

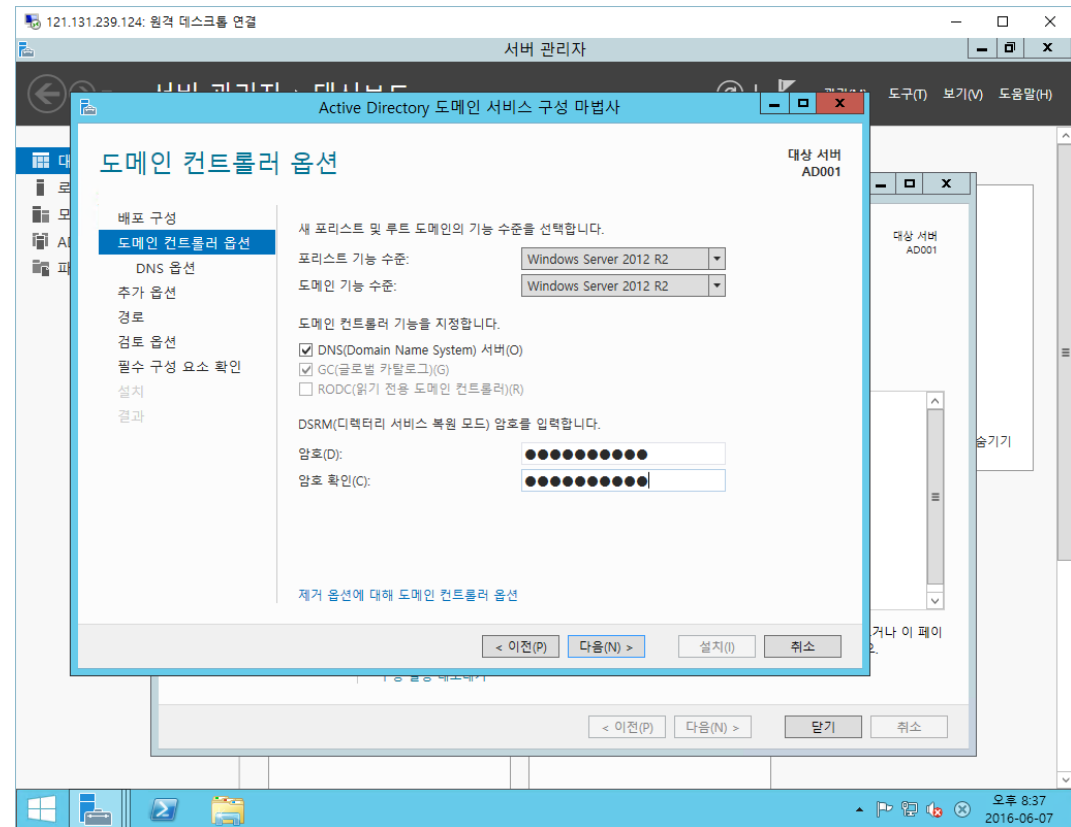
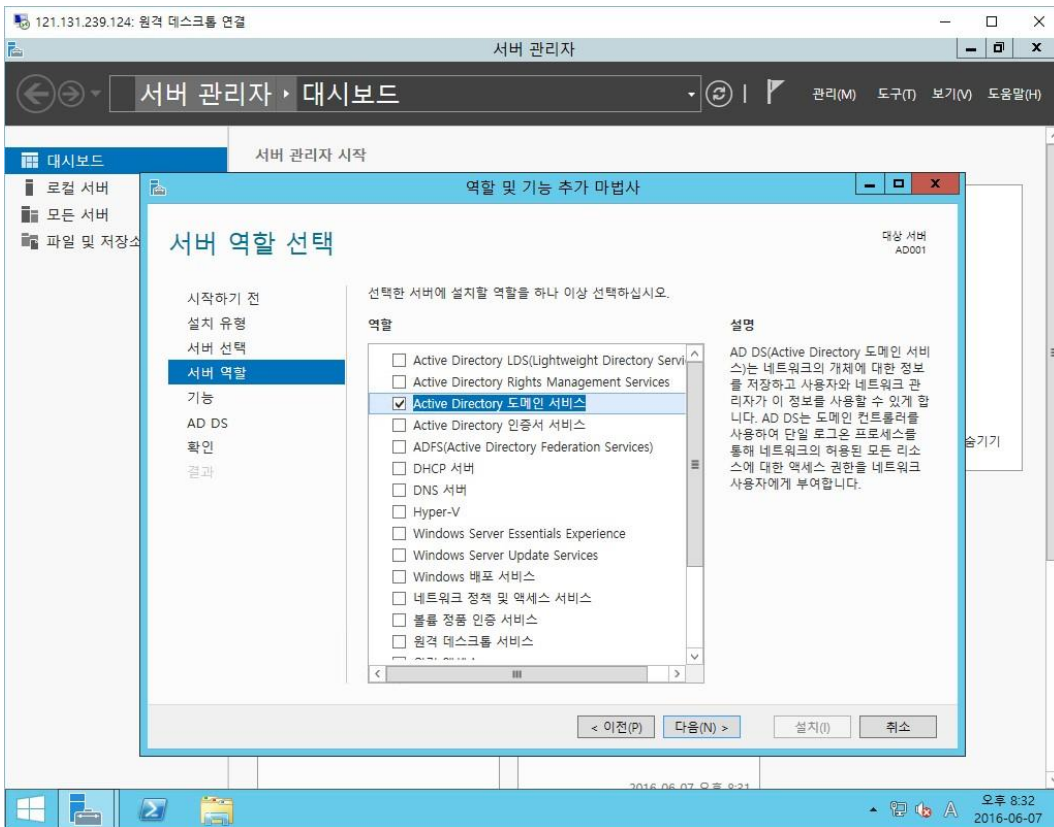
현재 옵션과 필터의 효율을 높①려고 관련 자료를 찾아보며 옵션과 필터 업데이트를 하고 있습니다.

[Windows] Windows Server 구축



ESXi 하①퍼바①저에Windows Server2 2012 R2를 설치했습니다.

[Windows] Active Directory 구축

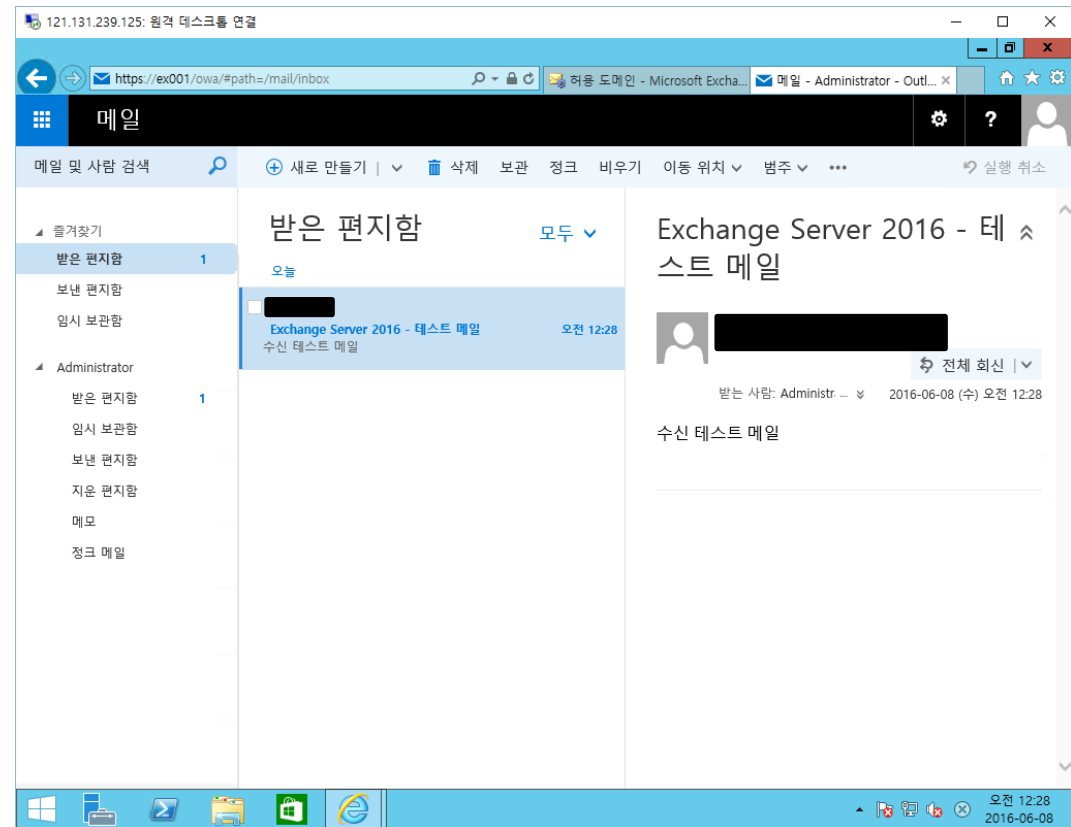
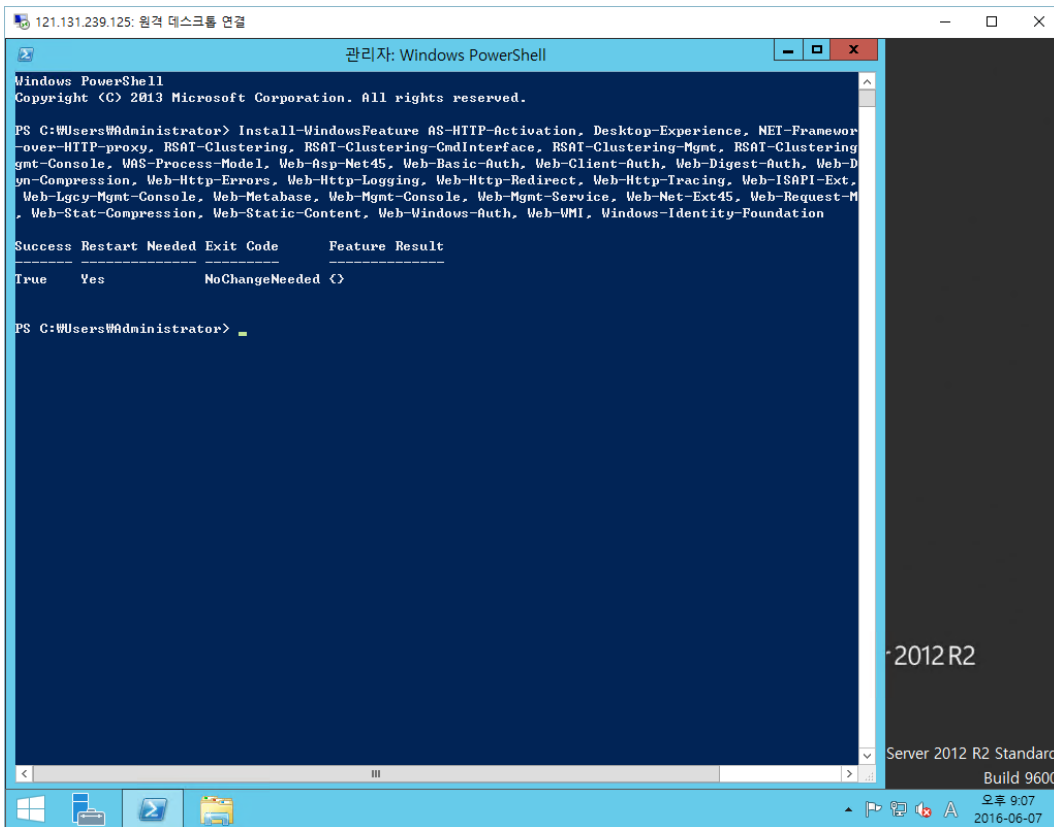


Windows Server 2012 R2에 Active Directory 서비스를 구축해보고

접근제어, 감사, 로깅등 정책을 설정해서 테스트도 진행해봤습니다.

자세한 설명은 블로그에 있습니다. BI
og : <https://jhok.kr/archives/144>

[Windows] Exchange Server 구축

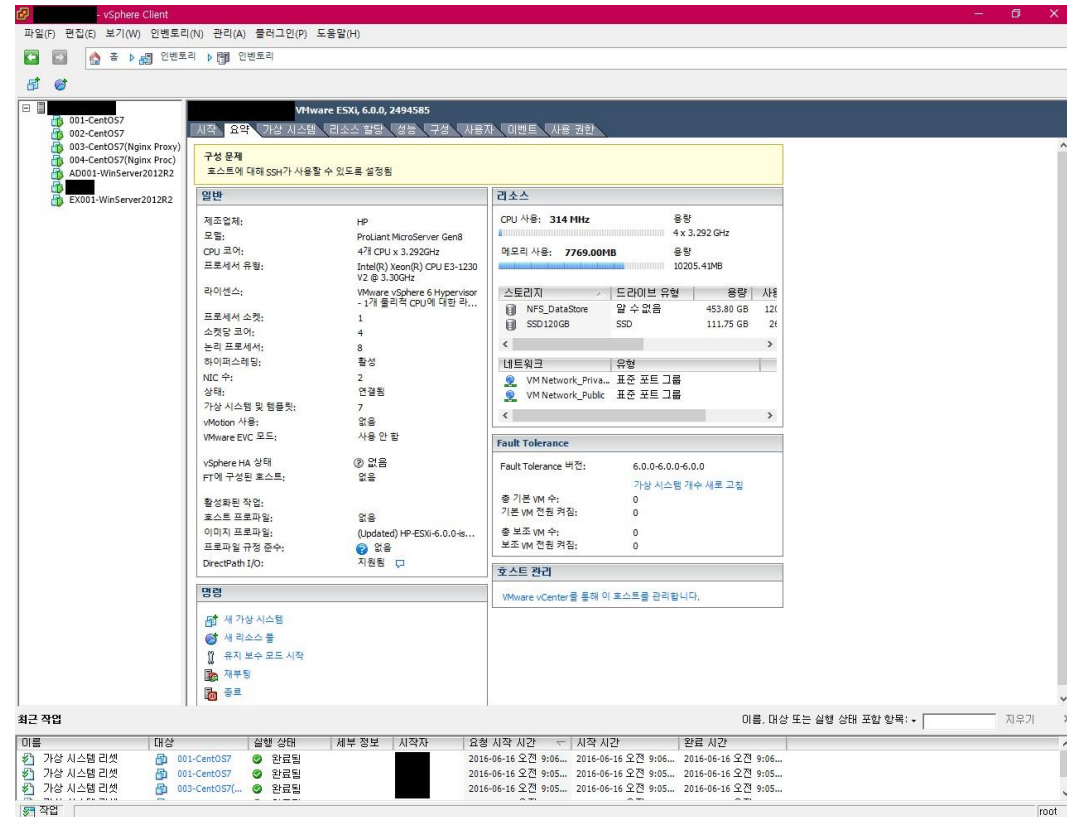
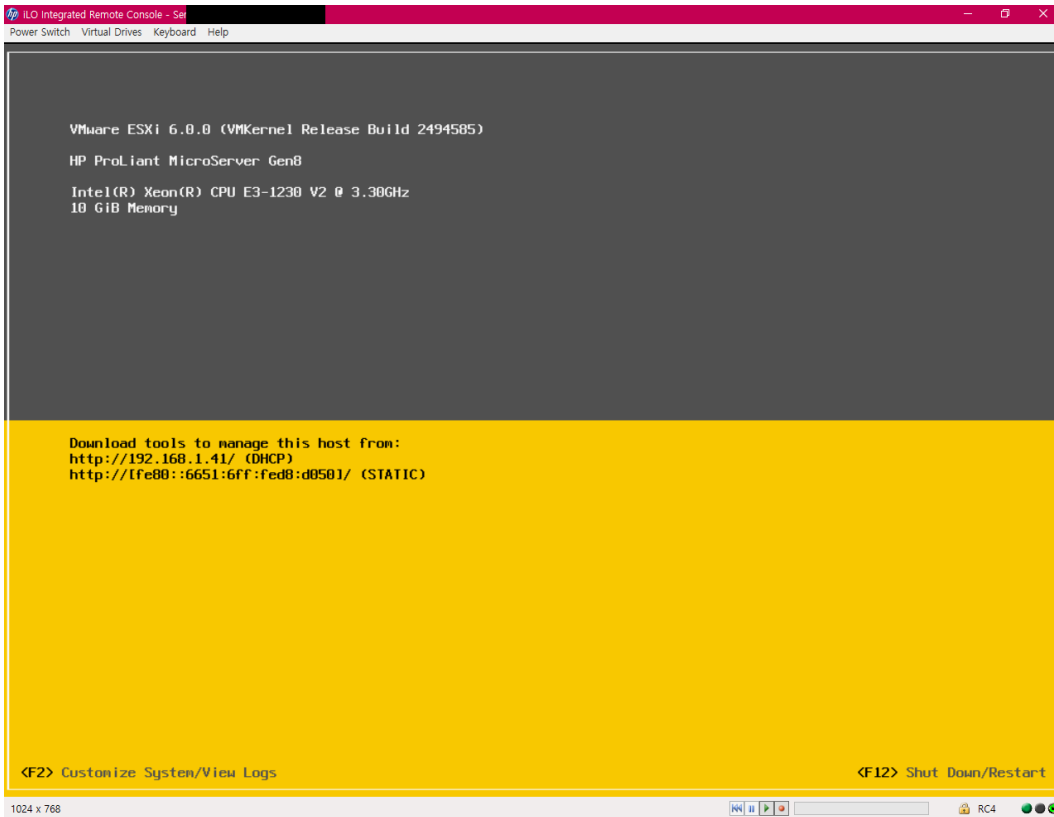


Windows Server 2012 R2에 Exchange Server 를 설치하여 개인 도메인의 메일 서버로 사용합니다.

메일서버의보안을 위해서 필터링, 멀웨어방지, 정책등 설정에 관한부분도 공부해봤습니다.

자세한 설명은 블로그에 있습니다. BI
og : <https://jhok.kr/archives/217>

[Vmware] ESXi 하①퍼바①저 구축



HP Gen8 서버에 Vmware ESXi 하①퍼바①저를 설치해서 서버 자원의 효율적인 분배를 통해서 여러 호스트 서버를 설치해보고 테스트 해봤습니다.

그리고 처음 하①퍼바①저를 선택할때 Xen, Hyper-V 등 여러 하①퍼바①저를 직접 설치해 성능을 비교해보고 호스트 OS 호환성 및 성능을 비교해서 가장 적합한 하①퍼바①저를 선택했습니다.



감사합니다