**Ubuntu Mirror Repository 구축 및 Ansible을 이용한 패키지 업데이트**

|  |  |  |  |
| --- | --- | --- | --- |
| 작성자 | 날짜 | 버전 | 내용 |
| 차 훈 | 2019.10.18 | v.1.2 | YAML 파일 추가 및 DOC 파일 내용 추가 |
| 차 훈 | 2019.10.16 | v1.1 | YAML 파일 수정 및 내용 추가 – 오타수정 |
| 차 훈 | 2019.10.10 | v1.0 | 내용 추가 및 1차 검토 |
| 차 훈 | 2019.10.02 | Draft v1.0 | 최초작성 |

**Case.**

**Ubuntu 14.06.02** 버전으로 설치된 400여대 서버를 **Ubuntu** **14.06.06** 버전으로 업데이트

**Problem.**

1. 고객의 스케줄에 따라 한번에 최대 진행 할 수 있는 수량은 40대로 제한되는데,

40대로 제한이 가능한지?

**Answer.** 인벤토리내에 작업대상 서버를 40대로 제한

1. 100 ~ 이상의 수량도 가능한지?

**Answer.** 1000 ~ 이상의 수량도 가능하며 inventory 설정을 어떻게 하는가에 따라 다름

**Spec.**

1. 가상화 : Oracle VM VirtualBox 6.0
2. Repo : Ubuntu 16.04.6 (Mem: 2048, Disk: 130G, NIC: AdapterBridge)
3. Client: Ubuntu 14.04.02 (Mem: 2048, Disk: 10G, NIC: AdapterBridge)

**Scenario.**

현재 고객 사이트 내부가 외부와 단절된 네트워크인 상황에서 업데이트 repo 사이트를 추가한다고 해도 외부에서 패키지를 다운받을 수 없는 없을것이고 내부에 설치된 Ubuntu 14.04.2 버전에 설치된 패키지내용을 파악하기 어려운 상태.

Ubuntu 에서는 CDROM을 활용한 Local Repo 부터 ISO Local Repo 등 많은 Repository 설정 방법등을 제공하고 있으나 좀더 깔끔하고 향후 배포 서버로써 활용할 수 있도록 Mirror 사이트를 구축하는 방법과 이를 설정하는 방법에 대해 기술함.

**Step 1. Depoly & Repository Server 설치**

1. Install Ubuntu14.06.06 Desktop
2. Apache2 설치 및 설정
3. Web 접속 테스트

**Step 2. Ubuntu Repo 서버 설정**

1. Packages 다운 및 설치
2. Repo 서버 설정 (Mirror 서버)
3. Repo 서버 테스트 (Web 테스트)

**Step 3. 업데이트 대상(Client) 서버 설치 및 설정**

1. Install Ubuntu14.06.02 Desktop
2. Repository 파일 초기화

**Step 4. 배포서버 Ansible 설치 및 YAML 작성**

1. Ansible 다운 및 설치
2. YAML 및 Jinja2 Template(필요시) 작성

**Step 5. 배포판 업데이트**

1. Ansible-Playbook 실행
2. 업데이트 확인

**Step 1. Deployment & Repository Server 설치**

* Install Ubuntu14.06.06 Desktop (ubuntu 16.04.6)

|  |
| --- |
| * **배포서버 설치 (설치시 배포패키지 공간으로 130G 이상 필요)**   root@hcha-VirtualBox:~# **cat /etc/lsb-release**  DISTRIB\_ID=Ubuntu  DISTRIB\_RELEASE=16.04  DISTRIB\_CODENAME=xenial  DISTRIB\_DESCRIPTION="Ubuntu 16.04.6 LTS"   * **업데이트 계정 생성** * root@hcha-VirtualBox:~# **adduser updater** * **업데이트 디렉토리 생성** * root@hcha-VirtualBox:~# **mkdir /home/updater/Archive** |

* Apache2 설치 및 설정

|  |
| --- |
| * **Apache2 설치 (2.4.18-2ubuntu3.13)** * root@hcha-VirtualBox:~# **apt-get install apache2** * root@hcha-VirtualBox**:~# dpkg -l | grep apache2**   ii apache2 2.4.18-2ubuntu3.13 amd64 Apache HTTP Server |

|  |
| --- |
| * **신규 DocumentRoot 권한설정**   /etc/apache2/apache2.conf 수정   * Mirror Directory 추가   <Directory **/home/updater/Archive/**>  Options Indexes FollowSymLinks  AllowOverride None  Require all granted  </Directory>   * **DocumentRoot 위치 변경 (Mirror Directory)** * sudo vi /etc/apache2/sites-enabled/000-default.conf   #DocumentRoot /var/www/html  DocumentRoot **/home/updater/Archive**   * **Apache2 재시작** * root@hcha-VirtualBox:~# systemctl restart apache2 * **Mirror Directory 생성 (130G 이상 공간필요)** * root@hcha-VirtualBox:~# df -h   …..  /dev/sr0 1.6G 1.6G 0 100% /media/updater/Ubuntu 16.04.6 LTS amd64  **/dev/sdb1 123G 60M 116G 1% /home/updater/Archive** |

* Web 접속 테스트

|  |
| --- |
| * http://<repo-ip> |

**Step 2. Ubuntu Repo 서버 설정**

* Mirror Packages 다운 및 설치

|  |
| --- |
| * **apt-mirror 패키지 설치** * root@hcha-VirtualBox:~# sudo apt-get install apt-mirror * root@hcha-VirtualBox:~# dpkg -l apt-mirror   ii apt-mirror 0.5.1-1ubuntu1 all APT sources mirroring tool |

* Repo 서버 설정 및 동기화 (Mirror 서버)

|  |
| --- |
| * **/etc/apt/mirror.list 수정**   set base\_path **/home/updater/Archive** # 주석제거 및 패치파일 경로 추가  # 다운받을 국내 공식 repo 서버 추가  **deb http://kr.archive.ubuntu.com/ubuntu/ trusty main restricted**  **deb http://kr.archive.ubuntu.com/ubuntu/ trusty-updates main restricted**  **deb http://kr.archive.ubuntu.com/ubuntu/ trusty universe**  **deb http://kr.archive.ubuntu.com/ubuntu/ trusty-updates universe**  **deb http://kr.archive.ubuntu.com/ubuntu/ trusty multiverse**  **deb http://kr.archive.ubuntu.com/ubuntu/ trusty-updates multiverse**  **deb http://kr.archive.ubuntu.com/ubuntu/ trusty-backports main restricted universe multiverse**  **deb http://security.ubuntu.com/ubuntu trusty-security main restricted**  **deb http://security.ubuntu.com/ubuntu trusty-security universe**  **deb http://security.ubuntu.com/ubuntu trusty-security multiverse**  # 나머지는 다 주석처리 +++++ **xenial 버전은 16.04 용 repo 임**  #deb http://archive.ubuntu.com/ubuntu xenial main restricted universe multiverse  #deb http://archive.ubuntu.com/ubuntu xenial-security main restricted universe multiverse  #deb http://archive.ubuntu.com/ubuntu xenial-updates main restricted universe multiverse  ##deb http://archive.ubuntu.com/ubuntu xenial-proposed main restricted universe multiverse  ##deb http://archive.ubuntu.com/ubuntu xenial-backports main restricted universe multiverse  #deb-src http://archive.ubuntu.com/ubuntu xenial main restricted universe multiverse  #deb-src http://archive.ubuntu.com/ubuntu xenial-security main restricted universe multiverse  #deb-src http://archive.ubuntu.com/ubuntu xenial-updates main restricted universe multiverse  ##deb-src http://archive.ubuntu.com/ubuntu xenial-proposed main restricted universe multiverse  ##deb-src http://archive.ubuntu.com/ubuntu xenial-backports main restricted universe multiverse   * 동기화 (상당한 시간소요됨)   root@hcha-VirtualBox:/home/updater/Archive# apt-mirror  Downloading 114 index files using 20 threads...  Begin time: Mon Oct 7 11:40:35 2019  [20]... [19]... [18]... [17]... [16]... [15]... [14]... [13]... [12]... [11]... [10]... [9]... [8]... [7]... [6].   * 동기화 디렉토리 확인   root@hcha-VirtualBox:/home/updater/Archive# pwd  /home/updater/Archive  root@hcha-VirtualBox:/home/updater/Archive# ls  **mirror skel var <= 신규 생성된 패키지 디렉토리 확인** |

* Repo 서버 테스트 (Web 테스트)

|  |
| --- |
| * **DocumentRoot 위치 변경 (Mirror Directory)**   /etc/apache2/sites-enabled/000-default.conf  #DocumentRoot /var/www/html  #DocumentRoot /home/updater/Archive # 기존 경로 주석  DocumentRoot **/home/updater/Archive/mirror/kr.archive.ubuntu.com**   * **Apache2 재시작** * root@hcha-VirtualBox:~# systemctl restart apache2 * **Web 접속**   http://<repo-ip> |

**Step 3. 업데이트 대상(Client) 서버 설치 및 설정**

* **기존 업데이트 대상이 있을 경우 설치 단계는 생략 가능하나, 설정은 진행해야 함**
* Install Ubuntu14.06.02 Desktop

|  |
| --- |
| * Oracle VM VirtualBox 에서 Client 설치시 네트워크 설정   네트워크 타입을 ‘어댑터에 브리지’로 선택하여 사내 DHCP Client 값을 가져와  배포서버와 통신 할 수 있도록 설정     * 설치시 계정추가   User: **testuser**  Password: **testuser012**     * **Ansible 통신을 위한 SSH 설정** **(필수)**   배포서버와 통신을 위한 SSH 설정 후 외부 Repository URL 을 다 삭제예정     * Client 배포판 버전 확인 * testuser@testuser-VirtualBox:~$ cat /etc/lsb-release   DISTRIB\_ID=Ubuntu  DISTRIB\_RELEASE=14.04  DISTRIB\_CODENAME=trusty  DISTRIB\_DESCRIPTION="**Ubuntu 14.04.2 LTS**"  - testuser@testuser-VirtualBox:~$ uname -r  3.16.0-30-generic |

* Repository 파일 초기화

|  |
| --- |
| * **기존의 repo 경로 파일들을 다 삭제하여 외부 업데이트 경로 차단**   **실제 업데이트 파일은 배포 서버를 통해서 받아야 함.**  **배포서버와 통신은 가능하지만 외부 연결이 안되는 상황을 가정해야 하고**  **실제로는 외부 연결이 가능하지만 업데이트 자체를 막아버리는 효과**   * testuser@testuser-VirtualBox:~$ sudo su * root@testuser-VirtualBox:/home/testuser# echo "" > /etc/apt/sources.list * root@testuser-VirtualBox:/home/testuser# apt-get update   Reading package lists... Done  root@testuser-VirtualBox:/home/testuser# |

**Step 4. 배포서버 Ansible 설치 및 YAML 작성**

* Ansible 다운 및 설치

|  |
| --- |
| * Ansible 설치 * root@hcha-VirtualBox:~# apt-get install ansible * root@hcha-VirtualBox:~# dpkg -l | grep ansible   ii ansible 2.0.0.2-2ubuntu1.3   * 로컬 ping 모듈 테스트 * root@hcha-VirtualBox:~# ansible -m ping localhost   localhost | SUCCESS => {  "changed": false,  "ping": "pong"  }   * ansible 최초 접속시 fingerprint 오류제거 설정 * root@hcha-VirtualBox:/home/updater# vi /etc/ansible/ansible.cfg   # uncomment this to disable SSH key host checking  **host\_key\_checking = False # 주석제거** |

* YAML 및 Jinja2 Template(필요시) 작성

|  |
| --- |
| * **디렉토리 구조** * updater@hcha-VirtualBox:~/Ansible$ pwd   /home/updater/Ansible   * updater@hcha-VirtualBox:~/Ansible$ ls   inven key\_exchange.yml main.yml pkgs ubuntu\_update.yml  /home/updater/Ansible  | |  | /pkgs --------------------------- => 패키지 디렉토리  | main.yml ----------------------- => 메인 yaml 파일  | inven --------------------------- => inventory  | key\_exchange.yml ------------- => 키교환 yaml 파일  | ubuntu\_update.yml ----------- => 업그레이드 yaml 파일 |

|  |
| --- |
| * **Inven // 인벤토리 파일로 변수설정 및 대상장비 정의**     [all:vars]  sshd\_user\_pam=no  sshd\_password\_authentication=no  sshd\_challenge\_response\_authentication=no  sshd\_permitroot\_login=no  update\_server=10.91.18.58 // Repository IP  user=partner // SSH remote user    [TEST]  #10.91.16.203 // Client 접속 IP  10.91.16.205   * **main.yml // 업그레이드 시작을 알리며 관련 YAML 파일 구동**     ---  - hosts: all  become: yes  tasks:  - debug:  msg: "Now that the updates will begin,  Ubuntu 14.04.2 -> Ubuntu 14.04.6"  - name: Start Key Exchange ------------  include: key\_exchange.yml  - name: Start Update ------------------  include: ubuntu\_update.yml   * **key\_exchange.yml // 배포서버와 대상서버간 공개키 교환**   ---  - hosts: all  become: yes  vars:  ssh\_known\_hosts\_command: "ssh-keyscan -T 10"  ssh\_known\_hosts\_path: "{{ lookup('env','HOME') + '/.ssh' }}"  ssh\_known\_hosts\_file: "{{ lookup('env','HOME') + '/.ssh/known\_hosts' }}"  ssh\_known\_hosts: "{{ groups['all'] }}"  # ---------------------- Known\_host For Repo Server -------------------------  tasks:  - name: Create a directory if it doesn't exist  file:  path: "{{ ssh\_known\_hosts\_path }}"  state: directory  - name: For each host, scan for its ssh public key  shell: "ssh-keyscan {{ item }},`dig +short {{ item }}`"  with\_items: "{{ ssh\_known\_hosts }}"  register: ssh\_known\_host\_results  ignore\_errors: yes  - name: Add/update the public key in the '{{ ssh\_known\_hosts\_file }}'  known\_hosts:  name: "{{ item.item }}"  key: "{{ item.stdout }}"  path: "{{ ssh\_known\_hosts\_file }}"  with\_items: "{{ ssh\_known\_host\_results.results }}"  #++++++++++++++ **아래라인 전부 주석 처리** ++++++++++++++++  #++ **키 교환 프로세스를 위해 배포 서버의 공개키를 전송하는 루틴** +++  #++ **배포 서버 ansible.cfg fingerprint 오류제거 설정을 했고** +++++++  #++ **배포 서버 known\_hosts 파일에 업데이트 대상 서버 key scan** +++  #++ **결과를 저장했기 때문에 연결에는 문제가 없다고 판단** +++++++++  # ---------------------- Authorized\_keys For Nodes -------------------------  #- name: ssh-keygen ansible-server  #connection: local  #command: "ssh-keygen -b 2048 -t rsa -f ~/.ssh/id\_rsa -q -N ''"  #ignore\_errors: yes  #run\_once: true  #- name: import id\_rsa.pub  #connection: local  #command: "cat ~/.ssh/id\_rsa.pub"  #register: id\_pub  #run\_once: true  #- name: create dir  #file:  #path: /home/{{user}}/.ssh/authorized\_keys  #state: touch  #- name: add ansible-node authrized keys  #lineinfile:  #dest: /home/{{user}}/.ssh/authorized\_keys  #line: "{{ id\_pub.stdout }}"   * **ubuntu\_update.yml // 업데이트 관련 패키지 설치 및 재부팅**   ---  - hosts: all  become: yes  tasks:  #----------------- Change file name (apt.conf, sources.list) -----------------  - name: change file name if exist (apt.conf)  stat:  path: /etc/apt/apt.conf  register: stat\_result  - name: move apt.conf to apt.conf.namu  command: mv /etc/apt/apt.conf /etc/apt/apt.conf.namu  when: stat\_result.stat.exists == True  - name: change file name if exist (sources.list)  stat:  path: /etc/apt/sources.list  register: stat\_result  - name: move sources.list to sources.list.namu  command: mv /etc/apt/sources.list /etc/apt/sources.list.namu  when: stat\_result.stat.exists == True  #---------------------------- Add Repositories -----------------------------  - name: add apt repo (trusty main restricted)  apt\_repository:  repo: 'deb [ arch=amd64 ] http://{{update\_server}}/ubuntu trusty main restricted'  state: present  update\_cache: yes  - name: add apt repo (trusty universe)  apt\_repository:  repo: 'deb [ arch=amd64 ] http://{{update\_server}}/ubuntu trusty universe'  state: present  update\_cache: yes  - name: add apt repo (trusty-updates restricted)  apt\_repository:  repo: 'deb [ arch=amd64 ] http://{{update\_server}}/ubuntu trusty-updates main restricted'  state: present  update\_cache: yes  #---------------------------- Upload aptitude.. -----------------------------  #- name: find packages (aptitude...) & upload packages (aptitude...)  # find: paths="/home/updater/Ansible/pkgs" recurse=no patterns="\*.deb"  # register: files\_to\_copy  #- copy: src={{ item.path }} dest=/var/cache/apt/archives  # with\_items: "{{ files\_to\_copy.files }}"  - name: upload aptitude..  copy: src={{ item.src }} dest={{ item.dest }}  with\_items:  - { src: '/home/updater/Ansible/pkgs/aptitude\_0.6.8.2-1ubuntu4\_amd64.deb', dest: '/var/cache/apt/archives/' }  - { src: '/home/updater/Ansible/pkgs/aptitude-common\_0.6.8.2-1ubuntu4\_all.deb', dest: '/var/cache/apt/archives/' }  - { src: '/home/updater/Ansible/pkgs/libcwidget3\_0.5.16-3.5ubuntu1\_amd64.deb', dest: '/var/cache/apt/archives/' }  - { src: '/home/updater/Ansible/pkgs/libboost-iostreams1.54.0\_1.54.0-4ubuntu3.1\_amd64.deb', dest: '/var/cache/apt/archives/' }  - { src: '/home/updater/Ansible/pkgs/libept1.4.12\_1.0.12\_amd64.deb', dest: '/var/cache/apt/archives/' }  #------------------- Delete lock files (for apt update)----------------------  - name: delete lock file  file: path={{ item }} state=absent  with\_items:  - /var/lib/dpkg/lock  - /var/cache/apt/archives/lock  - /var/lib/apt/lists/lock  - name: update apt lock  command: dpkg --configure -a  #---------------------------- Install aptitude.. -----------------------------  - name: install specific packages (aptitude...)  apt: name=aptitude state=present  with\_items:  - aptitude  - aptitude-common  - libboost-iostreams1.54.0  - libcwidget3  - libept1.4.12  #-------------------------- Upgrade All Packages -----------------------------  - name: upgrade all packages to the 14.04.6 version  apt: update\_cache=yes upgrade=full  #------------------------ Upgrade Specific Kernel ----------------------------  - name: upgrade specific kernel (4.4.0-142-generic)  apt:  name: linux-image-4.4.0-142-generic  state: present  #-------------- Upload initrd.img & megiaraid\_sas.ko (for R640) --------------  - name: upload device driver & initrd.img (for R640)  copy: src={{ item.src }} dest={{ item.dest }}  with\_items:  - { src: '/home/updater/Ansible/pkgs/initrd.img-4.4.0-142-generic', dest: '/boot' }  - { src: '/home/updater/Ansible/pkgs/megaraid\_sas.ko', dest: '/lib/modules/4.4.0-142-generic/kernel/drivers/scsi/megaraid/' }  #----------------- Change file name (apt.conf.namu, sources.list.namu) -----------------  - name: change file name if exist (apt.conf.namu)  stat:  path: /etc/apt/apt.conf.namu  register: stat\_result  - name: move apt.conf.namu to apt.conf  command: mv /etc/apt/apt.conf.namu /etc/apt/apt.conf  when: stat\_result.stat.exists == True  - name: change file name if exist (sources.list.namu)  stat:  path: /etc/apt/sources.list.namu  register: stat\_result  - name: move sources.list.namu to sources.list  command: mv /etc/apt/sources.list.namu /etc/apt/sources.list  when: stat\_result.stat.exists == True  #----------------------------------- Reboot --------------------------------  - name: reboot the server  command: reboot  async: 0  poll: 0   * **get\_upinfo\_ubuntu.yml // 업데이트 결과 회신**   ---  - hosts: all  become: yes  tasks:  #---------------------------- create a result file -----------------------------  - name: create result file  file:  path: "{{ ansible\_hostname }}.{{ ansible\_date\_time.date }}"  state: touch    - name: get kernel info  shell: uname -r > "{{ansible\_hostname}}.{{ansible\_date\_time.date}}"  - name: get kernel info  shell: cat /etc/lsb-release | grep DES | cut -d"=" -f2 >> "{{ansible\_hostname}}.{{ansible\_date\_time.date}}"  - name: megaraid\_sas  shell: modinfo megaraid\_sas | grep ^version | awk '{print $2}' >> "{{ansible\_hostname}}.{{ansible\_date\_time.date}}"  - name: create result dir  connection: local  file:  path: ./update\_result  state: directory  mode: 0777    - name: get file "{{ansible\_hostname}}.{{ansible\_date\_time.date}}"  fetch:  src: "{{ansible\_hostname}}.{{ansible\_date\_time.date}}"  dest: ./update\_result |

**Step 5. 배포판 업데이트**

* Ansible-Playbook 실행

|  |
| --- |
| * **~/Ansible$ ansible-playbook -i inven -vvv main.yml -k -K -u partner**   **Using /etc/ansible/ansible.cfg as config file**  **SSH password:**  **SUDO password[defaults to SSH password]:**  “Inven” 인벤토리 파일을 참조하여, “main.yml” 파일을 실행하고 업데이트 대상 서버의 “partner” 사용자 계정으로 SSH 계정 비번을 묻고, sudo 권한이 필요한 경우 sudo 비번을 입력하겠다는 command |

* 업데이트 확인 (개별파일)

|  |
| --- |
| * **업데이트 전** * hcha@hcha-VirtualBox:~$ cat /etc/lsb-release   DISTRIB\_ID=Ubuntu  DISTRIB\_RELEASE=14.04  DISTRIB\_CODENAME=trusty  DISTRIB\_DESCRIPTION="**Ubuntu 14.04.2 LTS**"  - hcha@hcha-VirtualBox:~$ uname -r  **3.16.0-30-generic**  - hcha@hcha-VirtualBox:~$ sudo modinfo megaraid\_sas  [sudo] password for hcha:  filename:/lib/modules/3.16.0-30-generic/kernel/drivers/scsi/megaraid/megaraid\_sas.ko  description: LSI MegaRAID SAS Driver  author: megaraidlinux@lsi.com  version: **06.803.01.00-rc1**   * **업데이트 후** * hcha@hcha-VirtualBox:~$ cat /etc/lsb-release   DISTRIB\_ID=Ubuntu  DISTRIB\_RELEASE=14.04  DISTRIB\_CODENAME=trusty  DISTRIB\_DESCRIPTION="**Ubuntu 14.04.6 LTS**"  - hcha@hcha-VirtualBox:~$ uname -r  **4.4.0-142-generic**  - hcha@hcha-VirtualBox:~$ sudo modinfo megaraid\_sas  [sudo] password for hcha:  filename: /lib/modules/4.4.0-142-generic/kernel/drivers/scsi/megaraid/megaraid\_sas.ko  description: Broadcom MegaRAID SAS Driver  author: megaraidlinux.pdl@broadcom.com  version: **07.709.08.00** |

* 업데이트 확인 (get\_upinfo\_ubuntu.yml 실행)

|  |
| --- |
| * **~/Ansible$ ansible-playbook -i inven -vvv get\_upinfo\_ubuntu.yml -k -K -u partner**   **Using /etc/ansible/ansible.cfg as config file**  **SSH password:**  **SUDO password[defaults to SSH password]:**   * ~/Ansible$ cd update\_result/ **// update\_result 생성** * ~/Ansible/update\_result$ pwd   **/home/updater/Ansible/update\_result**  - ~/Ansible/update\_result$ ls  **192.168.100.157 192.168.100.66 // 업데이트 대상 IP**  - ~/Ansible/update\_result$ cd 192.168.100.66  - ~/Ansible/update\_result/192.168.100.66$ cat hcha-VirtualBox.2019-10-18  hcha-VirtualBox.2019-10-18  - ~/Ansible/update\_result/192.168.100.66$ cat hcha-VirtualBox.2019-10-18  **4.4.0-142-generic // 업데이트 된 커널버전**  **"Ubuntu 14.04.6 LTS" // 업데이트 된 배포판 버전**  **07.709.08.00 // 패치된 megaraid\_sas 버전** |