엔서블 및 packer 자동화를 위한 코드

Status

자동화 (엔서블) 백업 패키지 채킹 코드

```
#!/bin/bash

cyan='\033[0;36m'
green='\033[0;32m'
lightgreen='\e[0;32m'
white='\e[0;37m'
red='\e[0;31m'
yellow='\033[0;33m'
blue='\033[0;34m'
orange='\e[38;5;166m'

## wget install
sudo yum -y install wget

######ansible_install_check

cd /etc/ansible
if [ "$?" -eq "0" ]; then
echo -e "$green" "[ ✔ ] Ensable installation completed ......................[ found ]"
sleep 1
else
echo -e "$red" "[ X ] NO ansible  -> not found"
echo "ansible-installin>>>>>>>> "
cd /home/ec2-user/lcj/ansible_install
./install.sh
fi

######gcc-make_install_check
gcc -v
if [ "$?" -eq "0" ]; then
echo -e "$green" "[ ✔ ] Ensable installation completed ......................[ found ]"
sleep 1
else
echo -e "$red" "[ X ] NO gcc - make  -> not found"
echo " gcc - make >>>>>>>> "
cd /home/ec2-user/lcj/gcc_install
./install.sh
fi

echo -e "$orange" "+-------------------------------------------------+"
echo -e "$orange" "|$white [$green Confirm$white ]$yellow ansible check complete $orange             |"
echo -e "$orange" "|$white [$green Confirm$white ]$yellow gccmake check complete $orange             |"
echo -e "$orange" "+-------------------------------------------------+"
clear

echo -e "$yellow" ">>>>packer_install_check>>>>"
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo
sudo yum -y install packer

echo ">>>>packer_install_check>>>"
packer --version
if [ "$?" -eq "0" ]; then
echo -e "$green" "[ ✔ ] Ensable installation completed ......................[ found ]"
sleep 1
else
echo -e "$red" "[ X ] NO ansible  -> not found"
echo "ansible-installin>>>>>>>> "
```

```
cd /home/ec2-user/lcj/packer_install
./install.sh
fi

echo -e "$orange" "+-------------------------------------------------+"
echo -e "$orange" "|$white [$green Confirm$white ]$yellow packer check complete $orange              |"
echo -e "$orange" "+-------------------------------------------------+"
clear

echo -e "$yellow" ">>>>tree_install_check>>>>"
yum install tree

git --version
if [ "$?" -eq "0" ]; then
echo -e "$green" "[ ✔ ] Ensable installation completed ......................[ found ]"
sleep 1
else
echo -e "$red" "[ X ] NO git && git hub  -> not found"
echo " gcc - make >>>>>>>> "
cd /home/ec2-user/lcj/git_install
./install.sh
fi

echo -e "$orange" "+-------------------------------------------------+"
echo -e "$orange" "|$white [$green 1$white ]$yellow tree rescue check complete $orange               |"
Type  :qa!  and press <Enter> to abandon all changes and exit Vim


echo -e "$orange" "+-------------------------------------------------+"
echo -e "$orange" "|$white [$green 1$white ]$yellow tree rescue check complete $orange               |"
echo -e "$orange" "|$white [$green 2$white ]$yellow git hub git check complete$orange                |"
echo -e "$orange" "|$white [$green 3$white ]$yellow General environment check complete$orange        |"
echo -e "$orange" "+-------------------------------------------------+"
clear
##python_install
cd /home/ec2-user/lcj/python_install
./install.sh

##boto3_install
cd /home/ec2-user/lcj/boto3_install
./install.sh

## aws cli
cd /home/ec2-user/lcj/aws_cli_install
./install.sh
echo -e "$orange" "+-------------------------------------------------+"
echo -e "$orange" "|$white [$green 1$white ]$yellow Full upload complete $orange                     |"
echo -e "$orange" "+-------------------------------------------------+"
cd
clear

echo -e "$orange" "+-------------------------------------------------+"

echo -e "$orange" "+-------------------------------------------------+"
echo -e "$orange" "|$white [$green 1$white ]$yellow ansible check complete $orange                   |"
echo -e "$orange" "|$white [$green 2$white ]$yellow aws cli check complete$orange                    |"
echo -e "$orange" "|$white [$green 3$white ]$yellow aws boto3 check complete$orange                  |"
echo -e "$orange" "|$white [$green 4$white ]$yellow packer check complete$orange                     |"
echo -e "$orange" "|$white [$green 5$white ]$yellow in normal operation  $orange                     |"
echo -e "$orange" "+-------------------------------------------------+"
aws ec2 describe-instances
clear
cd
cd project/
cd packer/
clear
packer build aws_ec2_build.json
```

ansible playbook

클라우드 환경 구성 (리전 및 인스턴스 관리)

```
---
 - name: This play is used to fetch instances info
   hosts: localhost
   connection: local
   gather_facts: fals
   tasks:
   - name: Info about ec2 using ec2_instance_facts
     ec2_instance_facts:
      region: us-east-1
      filters:
       "tag:Env": test
     register: ec2_info
     tags:
       - always
#   - name: Displaying output
#     debug: msg="{{item.instance_id}}"
#     with_items: "{{ec2_info.instances}}"
   - name: starting ec2 instance
     ec2:
      instance_ids: "{{item.instance_id}}"
      region: us-east-1
      state: running
      wait: yes
     tags:
      - start
      - never
     with_items: "{{ec2_info.instances}}"
```

ec2 생성 및 elb 구성

```
---
- hosts: tag_role_web
  become: true
  serial: 1
  pre_tasks:
    - name: Gathering ec2 facts
      ec2_facts:
      tags:
        - always
      register: ec2_find

    - debug: var=ec2_find

    - name: Instance De-register
      local_action: ec2_elb
      args:
        instance_id: "{{ ansible_ec2_instance_id }}"
        state: absent
        wait_timeout: 300
      tags:
        - always

    - debug: var=ec2_elbs

  roles:
  post_tasks:
    - name: Instance Register
      local_action: ec2_elb
      args:
        instance_id: "{{ ansible_ec2_instance_id }}"
        ec2_elbs: "{{ item }}"
        state: present
        wait_timeout: 300
      with_items: "{{ ec2_elbs }}"
      tags:
        - always
```

엔서블 aws 구성 체킹

```
---
- hosts: localhost
  connection: local
  vars_prompt:
    - name: "ec2_name"
      private: no
    - name: "ec2_instance_type"
      private: no
      default: "t2.micro"
    - name: "ec2_key_name"
      private: no
    - name: "ec2_group"
      private: no
      default: "default"
  roles:
    - { role: ec2, tags: [ec2] }
```

키 값 받아오기

```
---
- hosts: production
  user: root
  vars:
    username: ashley
    password: $6$rlLdG6wd1CT8v7i$7psP8l26lmaPhT3cigoYYXhjG28CtD1ifILq9KzvA0W0TH2Hj4.iO43RkPWgJGIi60Mz0CsxWbRVBSQkAY95W0
    public_key: ~/.ssh/id_rsa.pub
  roles:
   - yum
   - php
   - mariadb
```

```
- mariadb 설치
---
- name: Install MariaDB
  yum:
    name: mariadb-server
    state: present
    force: yes
```

php 설치 및 테스트 환경 구현

```
---
- name: Add PHP repo
  yum_repository:
    repo: ppa:ondrej/php

- name: Install PHP
```

```
  yum:
    name: "{{ item }}"
    state: present
    force: yes
    update_cache: yes
  with_items:
  - "php7.3-bcmath"
  - "php7.3-cli"
  - "php7.3-common"
  - "php7.3-curl"
  - "php7.3-fpm"
  - "php7.3-gd"
  - "php7.3-igbinary"
  - "php7.3-imagick"
  - "php7.3-mbstring"
  - "php7.3-mysql"
  - "php7.3-opcache"
  - "php7.3-redis"
  - "php7.3-soap"
  - "php7.3-xml"
  - "php7.3-xmlrpc"
  - "php7.3-zip"

- name: Set PHP user
  lineinfile:
    dest: /etc/php/7.3/fpm/pool.d/www.conf
    regexp: "^user"
    line: "user = {{ username }}"
    state: present
  notify: restart php

- name: Set PHP group
  lineinfile:
    dest: /etc/php/7.3/fpm/pool.d/www.conf
    regexp: "^group"
    line: "group = {{ username }}"
    state: present
  notify: restart php

- name: Set PHP listen owner
  lineinfile:
    dest: /etc/php/7.3/fpm/pool.d/www.conf
    regexp: "^listen\\.owner"
    line: "listen.owner = {{ username }}"
    state: present
  notify: restart php

- name: Set PHP listen group
  lineinfile:
    dest: /etc/php/7.3/fpm/pool.d/www.conf
    regexp: "^listen\\.group"
    line: "listen.group = {{ username }}"
    state: present
  notify: restart php

- name: Set PHP upload max filesize
  lineinfile:
    dest: /etc/php/7.3/fpm/php.ini
    regexp: "^upload_max_filesize"
    line: "upload_max_filesize = 128M"
    state: present
  notify: reload php

- name: Set PHP post max filesize
  lineinfile:
    dest: /etc/php/7.3/fpm/php.ini
    regexp: "^post_max_size"
    line: "post_max_size = 128M"
    state: present
  notify: reload php
```

아파치 설치 및 데스트 환경 구현

```
---
- hosts: apache
  sudo: yes
  tasks:
    - name: install apache2
      yum: name={{ item }}  update_cache=yes state=latestm force_yum-install=yes
      loop: [ 'yum' ] 설정

    - name: Install Apache
      yum: name=apache2 update_cache=yes state=latest

    - name: Create document root
      file:
        path: "/var/www/{{ http_host }}"
        state: directory
        owner: "{{ app_user }}"
        mode: '0755'

    - name: Copy index test page
      template:
        src: "files/index.html.j2"
        dest: "/var/www/{{ http_host }}/index.html"

    - name: Set up Apache virtualhost
      template:
        src: "files/apache.conf.j2"
        dest: "/etc/apache2/sites-available/{{ http_conf }}"

###방화벽 설정###
    - name: permit traffic in default zone for https service
      ansible.posix.firewalld:
         service: https
         permanent: yes
         state: enabled

    - name: restart apache2
      service: name=apache2
               state=restarted
```

패키지 업그레이드

```
---
- name: Upgrade packages
  yum: upgrade=safe

- name: Install packages
  yum:
    name: "{{ item }}"
    state: present
    update_cache: yes
  with_items:
  - fail2ba
```

mariadb 접속 테스트

```
#  Configuration
    - name: Sets the root password
      mariadb_user:
        name: root
        password: "{{ mysql_root_password }}"
        login_linux_socket: /var/run/mysql/mysqld.sock

    - name: Removes all anonymous user accounts
      mariadb_user:
        name: ''
        host_all: yes
        state: absent
        login_user: root
        login_password: "{{ mysql_root_password }}"

    - name: Removes the mariadb test database
      mariadb_db:
        name: test
        state: absent
        login_user: root
        login_password: "{{ mysql_root_password }}"
```

mariadb 접속

```
---
mysql_root_password: "mysql_root_password"
app_user: "sammy"
http_host: "your_domain"
http_conf: "your_domain.conf"
http_port: "80"
disable_default: true
```

```
<VirtualHost *:{{ http_port }}>
    ServerAdmin webmaster@localhost
    ServerName {{ http_host }}
    ServerAlias www.{{ http_host }}
    DocumentRoot /var/www/{{ http_host }}
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <Directory /var/www/{{ http_host }}>
          Options -Indexes
    </Directory>

    <IfModule mod_dir.c>
        DirectoryIndex index.php index.html index.cgi index.pl  index.xhtml index.htm
    </IfModule>

</VirtualHost>
```

packer (aws 생성)

```
{
  "variables": {
    "aws_access_key": "AKIA5G22QF354BBD5IUP",
    "aws_secret_key": "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"
  },
  "builders": [
    {
      "type": "amazon-ebs",
      "access_key": "{{user `aws_access_key`}}",
      "secret_key": "{{user `aws_secret_key`}}",
      "region": "us-east-1",
      "source_ami_filter": {
        "filters": {
          "virtualization-type": "hvm",
          "name": "rhel/images/*Red Hat Enterprise Linux 7.7 (HVM)*",
          "root-device-type": "ebs"
        },
        "owners": ["099720109477"],
        "most_recent": true
      },
      "instance_type": "t2.micro",
      "ssh_username": "ubuntu",
      "ami_name": "packer-example {{timestamp}}"
    }
  ]
}
```