SAS Viya SMP Installation Guide

0. Uninstall Viya

```
ansible-playbook deploy-cleanup.yml
```

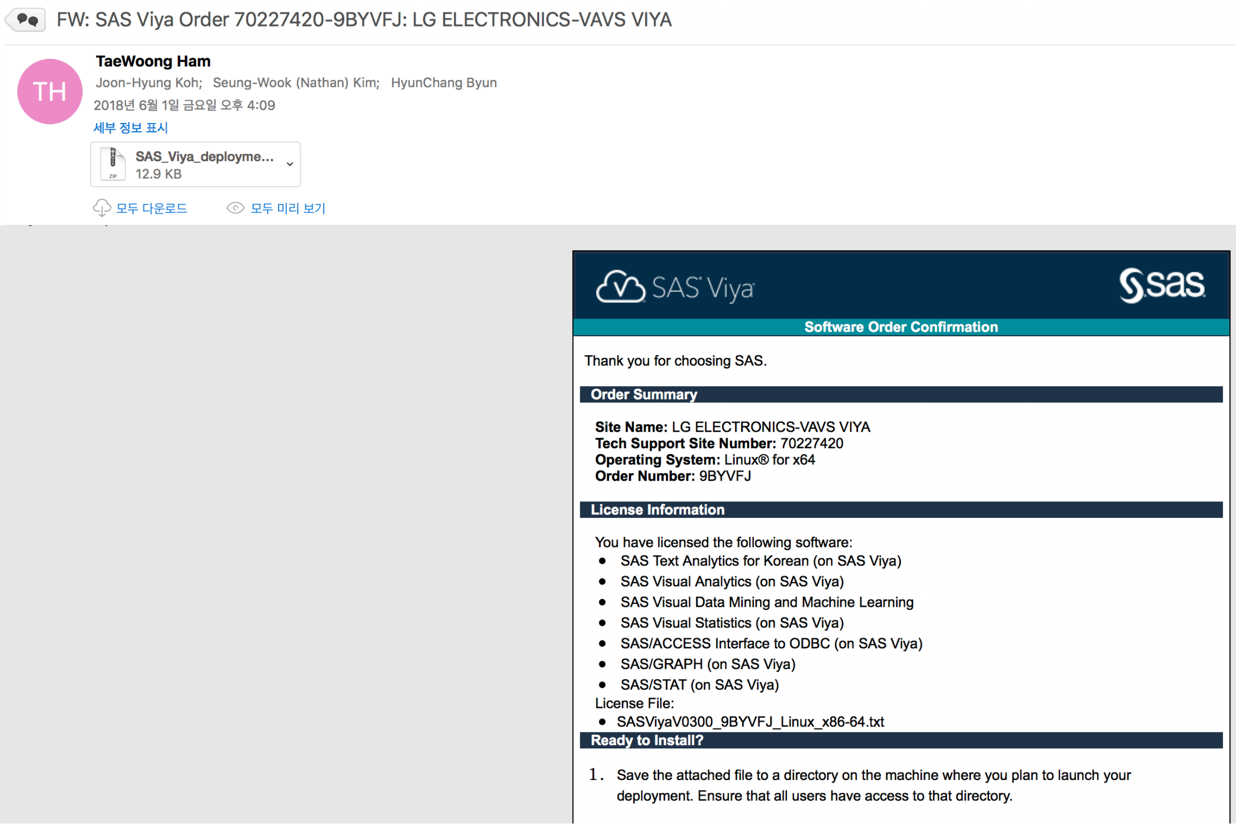
1. Pre Task Installation

사용자 생성및 디렉토리 생성

```
# 사용자 추가
useradd sas
useradd cas -g sas

# 인스톨 디렉토리 생성
mkdir /opt/sas
mkdir /opt/sas/install

chown -R sas:sas /opt/sas
```

File Upload

SOE(SAS Order Email) 에 첨부된 SAS_Viya_deployment.zip 파일을 다운받아 서버의 HOME 디렉토리에 업로드



FW: SAS Viya Order 70227420-9BYVFJ: LG ELECTRONICS-VAVS VIYA

TaeWoong Ham
Joon-Hyung Koh; Seung-Wook (Nathan) Kim; HyunChang Byun
2018년 6월 1일 금요일 오후 4:09

SAS_Viya_deployme...
12.9 KB

Software Order Confirmation

Thank you for choosing SAS.

Order Summary

Site Name: LG ELECTRONICS-VAVS VIYA
Tech Support Site Number: 70227420
Operating System: Linux® for x64
Order Number: 9BYVFJ

License Information

You have licensed the following software:

- SAS Text Analytics for Korean (on SAS Viya)
- SAS Visual Analytics (on SAS Viya)
- SAS Visual Data Mining and Machine Learning
- SAS Visual Statistics (on SAS Viya)
- SAS/ACCESS Interface to ODBC (on SAS Viya)
- SAS/GRAPH (on SAS Viya)
- SAS/STAT (on SAS Viya)

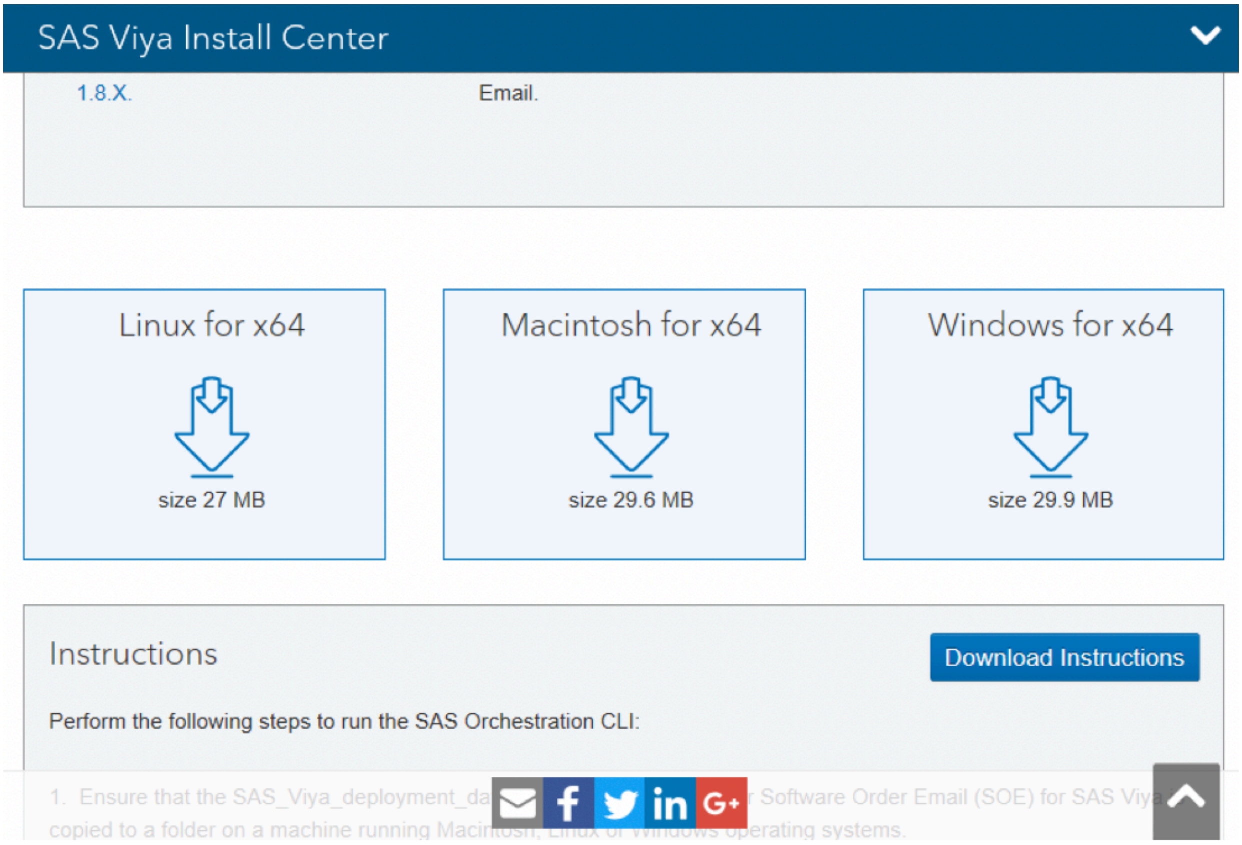
License File:

- SASViyaV0300_9BYVFJ_Linux_x86-64.txt

Ready to Install?

1. Save the attached file to a directory on the machine where you plan to launch your deployment. Ensure that all users have access to that directory.

SAS Viya Command line Interface Utility 웹사이트 에서 운영체제에 맞는 CLI 다운로드



Instructions

Perform the following steps to run the SAS Orchestration CLI:

1. Ensure that the SAS_Viya_deployment_da r Software Order Email (SOE) for SAS Viya copied to a folder on a machine running Macintosh, Linux or Windows operating systems.

다운로드 파일 : sas-orchestration.tgz

Ansible playbook 생성

```
# sas-orchestration.tgz 압축풀기 : sas-orchestration 실행파일 최종생성
gunzip sas-orchestration.tgz
tar -xvf sas-orchestration.tar

# 플레이북 생성 : SAS_Viya_playbook.tgz 파일 최종 생성
./sas-orchestration build --input SAS_Viya_deployment_data.zip

# 플레이북 압축 풀기
tar -xvf SAS_Viya_playbook.tgz

# 플레이북 파일 install 디렉토리로 이동
cp SAS_Viya_playbook.tgz /opt/sas/install/

# 플레이북 압축 풀기
cd /opt/sas/install
gunzip SAS_Viya_playbook.tgz
tar -xvf SAS_Viya_playbook.tar
```

Mirror Repository 생성

- centos yum repository 등록

  vi /etc/yum.repository.d/centos.repo

  ```
  [base]
  name=CentOS-$releasever - Base
  baseurl=http://ftp.daum.net/centos/7/os/$basearch/
  gpgcheck=1
  gpgkey=http://ftp.daum.net/centos/RPM-GPG-KEY-CentOS-7

  [updates]
  name=CentOS-$releasever - Updates
  baseurl=http://ftp.daum.net/centos/7/updates/$basearch/
  gpgcheck=1
  gpgkey=http://ftp.daum.net/centos/RPM-GPG-KEY-CentOS-7

  [extras]
  name=CentOS-$releasever - Extras
  baseurl=http://ftp.daum.net/centos/7/extras/$basearch/
  gpgcheck=1
  gpgkey=http://ftp.daum.net/centos/RPM-GPG-KEY-CentOS-7

  [centosplus]
  name=CentOS-$releasever - Plus
  baseurl=http://ftp.daum.net/centos/7/centosplus/$basearch/
  gpgcheck=1
  gpgkey=http://ftp.daum.net/centos/RPM-GPG-KEY-CentOS-7
  ```

- yum-utils 설치

  ```
  yum install yum-utils
  ```

- copy customized_deployment_script.sh

  ```
  cp /sas/install/sas_viya_playbook/customized_deployment_script.sh
  /sas/install/sas_viya_playbook/setup_repos.sh
  ```

- createrepo.sh

```
#!/bin/bash
sed -i -e 's/^\s*yum groupinstall/#yum groupinstall/' setup_repos.sh
./setup_repos.sh
MIRRORLOC=/sas/install/mirror
if [ ! -d ${MIRRORLOC} ]; then
mkdir -p ${MIRRORLOC}
fi
for f in $(ls /etc/yum.repos.d/sas-*.repo | cut -f4 -d/ | sed
s/.repo//g | grep -v sas-meta)
do
reposync -n -d -m --repoid=${f} --download_path=${MIRRORLOC} --
download-metadata
done
cd ${MIRRORLOC}
```

> MIRRORLOC 을 서버환경에 맞게 변경하면 해당 디렉토리로 설치에 필요한 파일들을 다운로드하게 됨

```
chmod 755 /sas/install/sas_viya_playbook/createrepo.sh
```

- createrepo.sh 실행

```
/sas/install/sas_viya_playbook/createrepo.sh
```

- yml 카피

```
cp /sas/install/sas_viya_playbook/internal/soe_defaults.yml
/sas/install/sas_viya_playbook/soe_defaults.yml
```

- createrepo 설치

```
yum install createrepo
```

- yumrepocreation.sh 생성

```
#!/bin/bash
#sudo yum install yum-utils createrepo httpd
REPOLOC=/sas/install/mirror
ORDERABLE=$(grep METAREPO_SOE_ORDERABLE soe_defaults.yml | awk -F"'" '{
print $2 }')
# Make the directory that will house the yum repository
if [ ! -d ${REPOLOC} ]; then
mkdir -p ${REPOLOC}
```

```
fi
echo ""
echo "Unpack the files from repomirror.tar.gz"
#tar xf repomirror.tar.gz -C ${REPOLOC}
echo ""
echo "Create the repository"
for repo in ${ORDERABLE}; do
NAME=$(sed -e 's/^"//' -e 's/"$//' <<<"$repo")
createrepo -v --update ${REPOLOC}/${NAME} -g ${REPOLOC}/${NAME}/comps.xml
done
```

> REPOLOC 을 MIRRORLOC 와 동일하게 맞춰준다.

- yumrepocreation.sh 실행

  ```
  /sas/install/sas_viya_playbook/yumrepocreation.sh
  ```

  > 각 파일 repository 에 repodata 폴더 및 repomd.xml 파일 생성 확인

2. Pre-Installation

YUM Repository 추가

Epel

```
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-$majversion.noarch.rpm
```

> $majversion 에 리눅스 버전 추가
>
> redhat 7.4 일경우 $majversion 은 7

확인사항

```
# 운영체제 확인
cat /etc/*-release

m
# sudo 권한 확인
sudo -v
```

```
# systemd 버전확인 (219 ~230버전)
rpm —qa | grep systemd

# 필요하면 systemd 업데이트
yum update systemd

# JAVA 버전확인 (Oracle JRE SE version 1.8.0_92 이상)
Java -version

# HTTPD 확인
service httpd status

# 필요시 HTTPD 설치
yum install httpd

# 유저 추가 (sas,cas 모두 sas 그룹으로 생성)
useradd sas
useradd cas -g sas

# 유저확인 ('sas','cas' 모두 'sas'그룹 이여야 함)
cat /etc/passwd | grep -e sas -e cas


# 필수 요구 패키지 설치 확인
rpm -qa at nfs-utils.x86_64 nfs-utils-lib.x86_64 gcc glibc firefox compat-libstdc++-33 compat-glibc GLIBC 2.12 libuuid libSM libXrender fontconfig libstdc++ zlib apr ksh numactl perl-Net-SSLeay libXext libXext.i686 libXp libXp.i686 libXts libXtst.i686 libgcc libgcc.i686 libpng12 libpng12.i686 python 2.7 xterm xauth libXmu uuid mod_ssl tcl

# 패키지 설치 스크립트
pkgs="                    \
 at                       \
 nfs-utils.x86_64         \
 nfs-utils-lib.x86_64     \
 gcc                      \
 glibc                    \
 firefox                  \
 compat-libstdc++-33      \
 compat-glibc             \
 GLIBC 2.12               \
 libuuid                  \
 libSM                    \
 libXrender               \
 fontconfig               \
 libstdc++                \
 zlib                     \
 apr                      \
 ksh                      \
 numactl                  \
 perl-Net-SSLeay          \
 libXext                  \
 libXext.i686             \
 libXp                    \
 libXp.i686               \
 libXtst                  \
 libXtst.i686             \
 libgcc                   \
 libgcc.i686              \
 libpng12                 \
 libpng12.i686            \
 python 2.7               \
 xterm                    \
 xauth                    \
 libXmu                   \
 uuid                     \
 mod_ssl                  \
 tcl                      \
"
yum install $pkgs -y

# 방화벽 중단
service firewalld stop
sudo systemctl disable firewalld.service

# Selinux 중단
sudo sestatus
만약 활성화 상태일 경우 모든 타겟 서버에 다음 명령을 통해 permissive mode 를 활성화 함

sudo setenforce 0    => 안먹을 때도 있음 재부팅 필요
sudo sed -i.bak -e 's/SELINUX=enforcing/SELINUX=permissive/g'
/etc/selinux/config

# Ansible 인스톨을 위한 EPEL 리포지토리 추가 스크립트

## Attach EPEL
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-
latest-$majversion.noarch.rpm

# Display the available repositories
sudo yum repolist
```

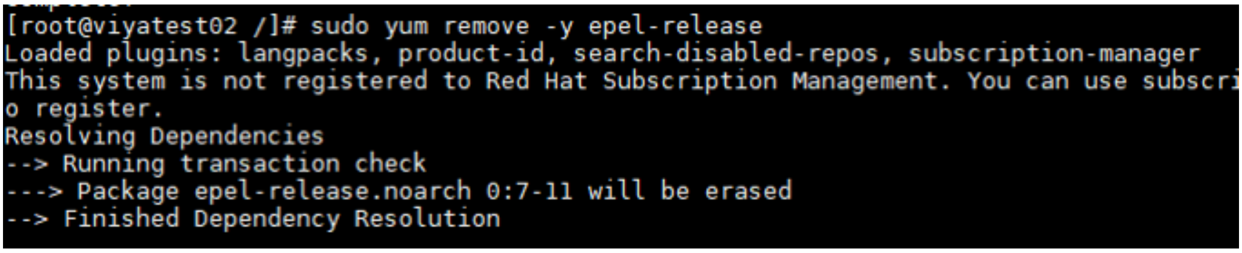
Ansible 설치

- Ansible 설치를 위한 패키지 설치

```
sudo yum install -y python python-setuptools python-devel openssl-devel
sudo yum install -y python-pip gcc wget automake libffi-devel python-six
```

- Epel 삭제

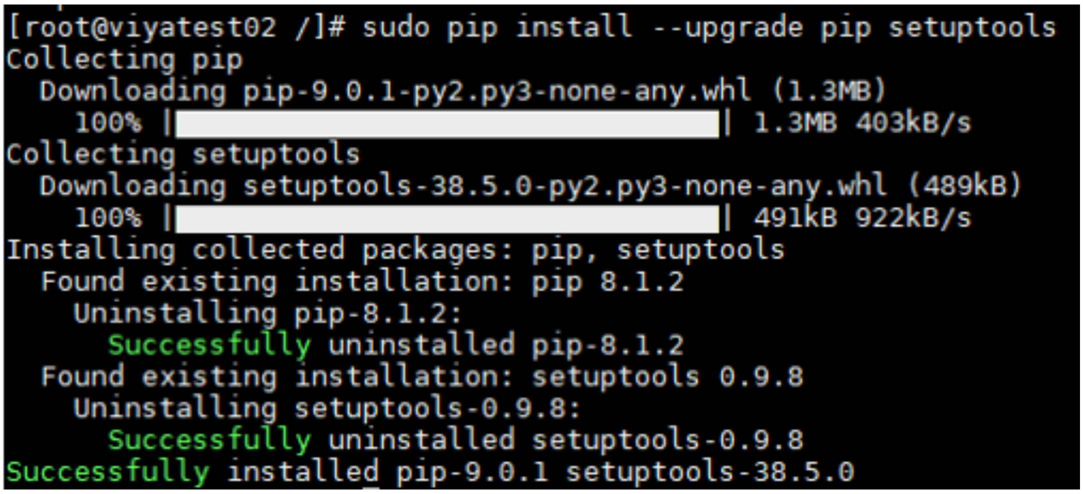
```
sudo yum remove -y epel-release
```



```
[root@viyatest02 /]# sudo yum remove -y epel-release
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscri
o register.
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be erased
--> Finished Dependency Resolution
```

- PIP 업그레이드

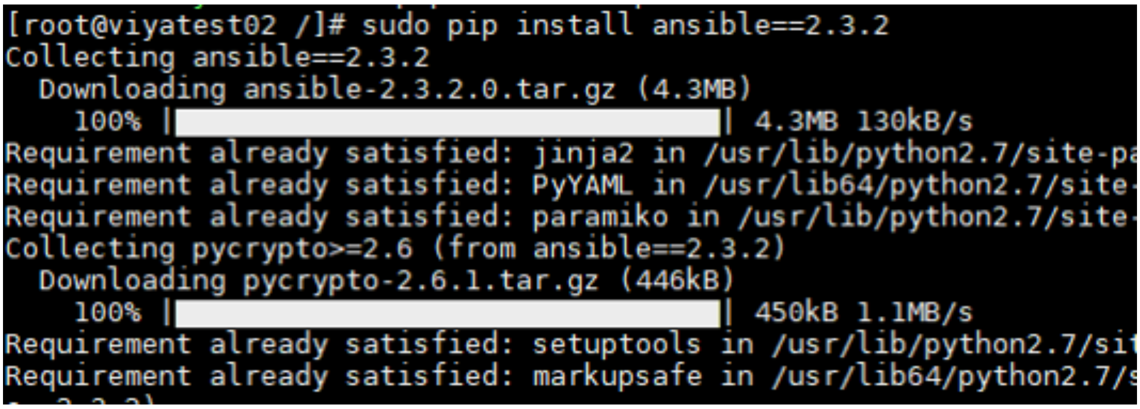
```
sudo pip install --upgrade pip setuptools
```



```
[root@viyatest02 /]# sudo pip install --upgrade pip setuptools
Collecting pip
  Downloading pip-9.0.1-py2.py3-none-any.whl (1.3MB)
    100% |████████████████████████████████| 1.3MB 403kB/s
Collecting setuptools
  Downloading setuptools-38.5.0-py2.py3-none-any.whl (489kB)
    100% |████████████████████████████████| 491kB 922kB/s
Installing collected packages: pip, setuptools
  Found existing installation: pip 8.1.2
    Uninstalling pip-8.1.2:
      Successfully uninstalled pip-8.1.2
  Found existing installation: setuptools 0.9.8
    Uninstalling setuptools-0.9.8:
      Successfully uninstalled setuptools-0.9.8
Successfully installed pip-9.0.1 setuptools-38.5.0
```

- Ansible 설치

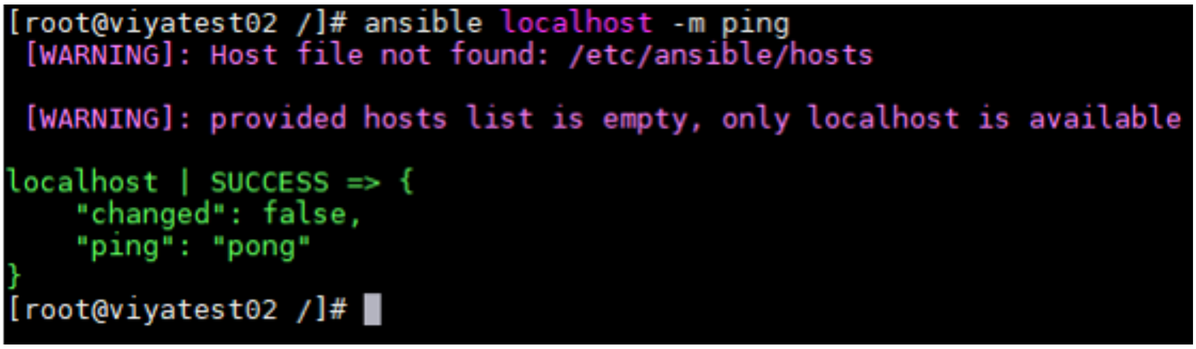
```
sudo pip install ansible==2.3.2
```



```
[root@viyatest02 /]# sudo pip install ansible==2.3.2
Collecting ansible==2.3.2
  Downloading ansible-2.3.2.0.tar.gz (4.3MB)
    100% |████████████████████████████████| 4.3MB 130kB/s
Requirement already satisfied: jinja2 in /usr/lib/python2.7/site-pa
Requirement already satisfied: PyYAML in /usr/lib64/python2.7/site-
Requirement already satisfied: paramiko in /usr/lib/python2.7/site-
Collecting pycrypto>=2.6 (from ansible==2.3.2)
  Downloading pycrypto-2.6.1.tar.gz (446kB)
    100% |████████████████████████████████| 450kB 1.1MB/s
Requirement already satisfied: setuptools in /usr/lib/python2.7/sit
Requirement already satisfied: markupsafe in /usr/lib64/python2.7/s
```

- 설치 확인

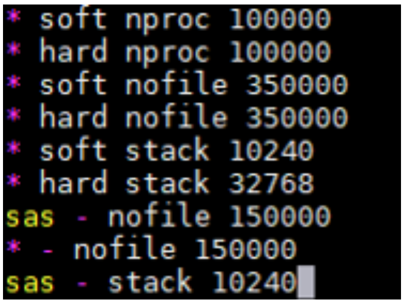
```
ansible --version
ansible localhost -m ping
```



```
[root@viyatest02 /]# ansible localhost -m ping
 [WARNING]: Host file not found: /etc/ansible/hosts

 [WARNING]: provided hosts list is empty, only localhost is available

localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[root@viyatest02 /]#
```

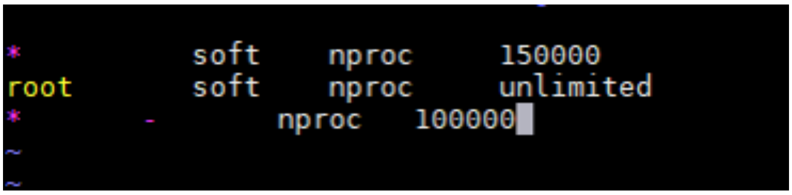
커널변수 설정

```
# vi /etc/ssh/sshd_config

MaxStartups 100
MaxSessions 100

# vi /etc/security/limits.conf

*   soft    nproc   100000
*   hard    nproc   100000
*   soft    nofile  350000
*   hard    nofile  350000
*   soft    stack   10240
*   hard    stack   32768
sas -   nofile  150000
*   -   nofile  150000
sas -   stack   10240
```



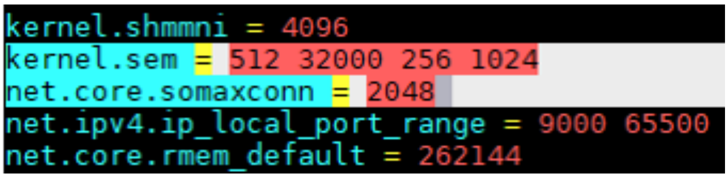
```
* soft nproc 100000
* hard nproc 100000
* soft nofile 350000
* hard nofile 350000
* soft stack 10240
* hard stack 32768
sas - nofile 150000
* - nofile 150000
sas - stack 10240
```

```
vi /etc/security/limits.d/20-nproc.conf

*          soft    nproc     150000
root       soft    nproc     unlimited
*       -       nproc   100000
```



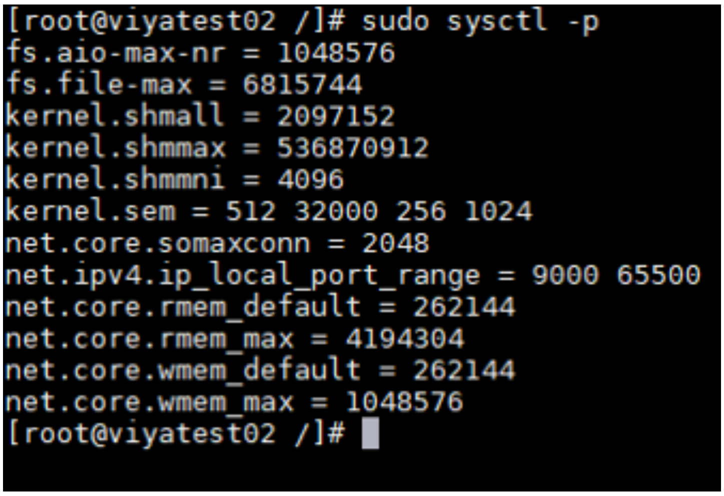
```
*          soft    nproc     150000
root       soft    nproc     unlimited
*      -        nproc    100000
~
~
```

```
# kernel.sem과 net.core.somaxconn 정보 설정
vi /etc/sysctl.conf

kernel.shmmni = 4096
kernel.sem = 512 32000 256 1024
net.core.somaxconn = 2048
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
```



```
kernel.shmmni = 4096
kernel.sem = 512 32000 256 1024
net.core.somaxconn = 2048
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
```

```
sudo sysctl -p
```



```
[root@viyatest02 /]# sudo sysctl -p
fs.aio-max-nr = 1048576
fs.file-max = 6815744
kernel.shmall = 2097152
kernel.shmmax = 536870912
kernel.shmmni = 4096
kernel.sem = 512 32000 256 1024
net.core.somaxconn = 2048
net.ipv4.ip_local_port_range = 9000 65500
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
[root@viyatest02 /]#
```

```
vi /etc/systemd/system.conf

DefaultTimeoutStartSec=1800s
DefaultTimeoutStopSec=1800s
```

3. Installation

/opt/sas/install/sas_viya_playbook 경로에서 다음 명령어 수행

```
cp sample-inventories/inventory_local.ini ./inventory.ini
```

vi /sas/install/sas_viya_playbook/inventory.ini 수행하여 ansible_connection 확인. ansible을 local로 사용시 default 설정 유지. remote 사용시 ansible_connection을 ansible_host로 변경 후 target host name으로 값 변경

```
 vi /sas/install/sas_viya_playbook/inventory.ini

 [host-definitions]
 deployTarget ansible_connection=local
```

cascache 생성

```
# root 계정으로 /opt 하위에 sas/cascache 디렉토리 생성.
# cascache 디렉토리에 chmod 777 cascache 명령어를 통해 권한 부여

mkdir /opt/sas/cascache
chmod 777 cascache
```

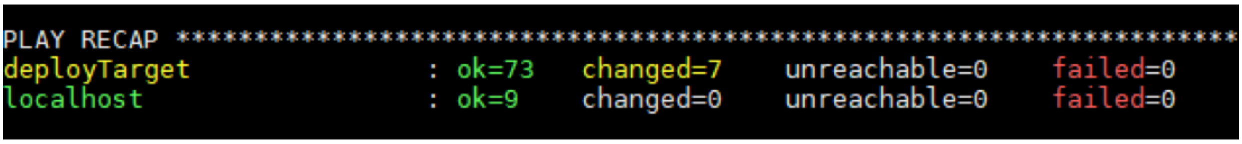
vars.yml 수정

```
# vi /sas/install/sas_viya_playbook/vars.yml 를 입력하여 파일을 연 후 적절한 DEPLOYMENT_LABEL 을 설정
DEPLOYMENT_LABEL : "{{ DEPLOYMENT_ID}}"

# sas_install_type 설정. default 설정은 all 이며 모든 소프트웨어를 설치.
# programming 옵션 설정 시 CAS, SAS Foundation, SAS Studio를 포함한 programming interface만 설치
sas_install_type : all

# casenv_group 값을 cas가 속해 있는 그룹의 이름으로 설정
casenv_group : sas

# Full Deployment를 위하여 LDAP user를 cas admin user로 설정. casenv_admin_user의 값을 주석 해제 후 해당하는 user 이름으로 값 설정.
casenv_admin_user : cas

# CAS_DISK_CACHE를 주석 해제 후 /opt/sas/cacscache 로 경로 설정
CAS_DISK_CACHE : /opt/sas/cascache
```

Validation 수행

/opt/sas/install/sas_viya_playbook/ 에서 아래 명령어 수행

```
ansible-playbook system-assessment.yml
```



```
PLAY RECAP *********************************************************************
deployTarget               : ok=73   changed=7    unreachable=0    failed=0
localhost                  : ok=9    changed=0    unreachable=0    failed=0
```

Deployment 수행

/opt/sas/install/sas_viya_playbook/ 에서 아래 명령어 수행

```
# 일반적인 방법
ansible-playbook -vvv site.yml

# 백그라운드 수행
nohup ansible-playbook -vvv site.yml &
```

> 백그라운드로 수행할 경우 deployment.log 를 확인하면 인스톨 상황을 알 수 있음

Yum Repository 등록

subscription-manager 미등록시 subscription-manager 등록 혹은 centOS repository 사용

> vi /etc/yum.repository.d/centos.repo

```
[base]
name=CentOS-$releasever - Base
baseurl=http://ftp.daum.net/centos/7/os/$basearch/
gpgcheck=1
gpgkey=http://ftp.daum.net/centos/RPM-GPG-KEY-CentOS-7

[updates]
name=CentOS-$releasever - Updates
baseurl=http://ftp.daum.net/centos/7/updates/$basearch/
gpgcheck=1
gpgkey=http://ftp.daum.net/centos/RPM-GPG-KEY-CentOS-7

[extras]
name=CentOS-$releasever - Extras
baseurl=http://ftp.daum.net/centos/7/extras/$basearch/
gpgcheck=1
gpgkey=http://ftp.daum.net/centos/RPM-GPG-KEY-CentOS-7
```

```
[centosplus]
name=CentOS-$releasever - Plus
baseurl=http://ftp.daum.net/centos/7/centosplus/$basearch/
gpgcheck=1
gpgkey=http://ftp.daum.net/centos/RPM-GPG-KEY-CentOS-7
```

```
rpm -qa at nfs-utils.x86_64 nfs-utils-lib.x86_64 gcc glibc firefox compat-libstdc++-33 •compat-glibc GLIBC 2.12 libuuid libSM libXrender fontconfig libstdc++ zlib apr ksh •numactl perl-Net-SSLeay libXext libXext.i686 libXp libXp.i686 libXts libXtst.i686 libgcc •libgcc.i686 libpng12 libpng12.i686 python 2.7 xterm xauth libXmu uuid mod_ssl tcl
```

```
service firewalld stop
systemctl disable firewalld.service
```