

TLS 적용


▼ KeyCloak TSL 인증키 발급

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8794d58b-eb3b-4933-b8c0-af0fbdeb9b1c/keycloak_tls_설정.txt

▼ ROOT CA 인증서 생성

OpenSSL 로 ROOT CA 생성 및 SSL 인증서 발급

웹서비스에 https 를 적용할 경우 SSL 인증서를 VeriSign 이나 Thawte, GeoTrust 등에서 인증서를 발급받아야 하지만 비용이 발생하므로 실제 운영 서버가 아니면 발급 받는데 부담이 될 수 있다. 이럴때 OpenSSL 을 이용하여 인증기관을 만들고 Self signed certificate 를 생성하고 SSL 인증서를 발급하는 법을 정리해 본다.

 <https://www.lesstif.com/system-admin/openssl-root-ca-ssl-6979614.html>

CA 가 사용할 RSA key pair(public, private key) 생성

```
$ openssl genrsa -aes256 -out lesstif-rootca.key 2048

-----
.....+++++
.....+++++
e is 65537 (0x010001)
Enter pass phrase for lesstif-rootca.key: paas-ta
Verifying - Enter pass phrase for lesstif-rootca.key: paas-ta
```

개인키 권한 설정

```
$ chmod 600 lesstif-rootca.key
```

CSR(Certificate Signing Request) 생성을 위한 openssl 설정 파일을 만들고 rootca_openssl.conf(변경 가능) 로 저장

```
#아래 내용 복사해서 rootca_openssl.conf 만들기
$ vim rootca_openssl.conf
```

```
[ req ]
default_bits          = 2048
default_md             = sha1
default_keyfile        = lesstif-rootca.key
distinguished_name     = req_distinguished_name
extensions            = v3_ca
req_extensions        = v3_ca

[ v3_ca ]
basicConstraints       = critical, CA:TRUE, pathlen:0
subjectKeyIdentifier   = hash
##authorityKeyIdentifier = keyid:always, issuer:always
keyUsage              = keyCertSign, cRLSign
nsCertType            = sslCA, emailCA, objCA
[req_distinguished_name ]
countryName           = Country Name (2 letter code)
countryName_default   = KR
countryName_min       = 2
countryName_max       = 2

# 회사명 입력
organizationName      = Organization Name (eg, company)
organizationName_default = PaaS-TA

# 부서 입력
#organizationalUnitName = Organizational Unit Name (eg, section)
#organizationalUnitName_default = Condor Project

# SSL 서비스할 domain 명 입력
```

```

commonName          = Common Name (eg, your name or your server's hostname)
commonName_default   = PaaS-TA's Self Signed CA
commonName_max       = 64

```

Root CA 용 CSR 요청 파일을 생성

```
$ openssl req -new -key lesstif-rootca.key -out lesstif-rootca.csr -config rootca_openssl.conf
```

```

-----
Enter pass phrase for lesstif-rootca.key: paas-ta
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [KR]: enter
Organization Name (eg, company) [PaaS-TA]: enter
Common Name (eg, your name or your servers hostname) [PaaS-TAs Self Signed CA]: enter

```

10년짜리 self-signed 인증서를 생성(인증서는 -out 옵션 뒤에 기술한 파일명(예: lesstif-rootca.crt)으로 생성)

```

openssl x509 -req -days 3650 \
-extentions v3_ca \
-set_serial 1 \
-in lesstif-rootca.csr \
-signkey lesstif-rootca.key \
-out lesstif-rootca.crt \
-extfile rootca_openssl.conf
-----
Signature ok
subject=C = KR, O = PaaS-TA, CN = PaaS-TAs Self Signed CA
Getting Private key
Enter pass phrase for lesstif-rootca.key: paas-ta

```


제대로 생성되었는지 확인을 위해 인증서의 정보를 출력

```
$ openssl x509 -text -in lesstif-rootca.crt
```

▼ SSL 인증서 발급

OpenSSL 로 ROOT CA 생성 및 SSL 인증서 발급

웹서비스에 https 를 적용할 경우 SSL 인증서를 VeriSign 이나 Thawte, GeoTrust 등에서 인증서를 발급받아야 하지만 비용이 발생하므로 실제 운영 서버가 아니면 발급 받는데 부담이 될 수 있다. 이럴때 OpenSSL 을 이용하여 인증기관을 만들고 Self signed certificate 를 생성하고 SSL 인증서를 발급하는 법을 정리해 본다.

 <https://www.lesstif.com/system-admin/openssl-root-ca-ssl-6979614.html>

SSL 호스트에서 사용할 RSA key pair(public, private key) 생성

```

$ openssl genrsa -aes256 -out lesstif.com.key 2048
-----
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
e is 65537 (0x010001)
Enter pass phrase for lesstif.com.key: paas-ta
Verifying - Enter pass phrase for lesstif.com.key: paas-ta

```

개인키 pass phrase 제거

```

$ cp lesstif.com.key lesstif.com.key.enc
$ openssl rsa -in lesstif.com.key.enc -out lesstif.com.key
-----
Enter pass phrase for lesstif.com.key.enc: paas-ta
writing RSA key

```

개인키의 유출 방지를 위해 group 과 other의 permission 을 모두 제거

```
$ chmod 600 lesstif.com.key*
```

CSR(Certificate Signing Request) 생성을 위한 openssl config 파일을 만들고 host_openssl.conf(변경 가능) 라는 이름으로 저장

```
$ vim host_openssl.conf
```

```
[ req ]
default_bits          = 2048
default_md             = sha1
default_keyfile        = lesstif-rootca.key
distinguished_name     = req_distinguished_name
extensions             = v3_user
## 인증서 요청시에도 extension 이 들어가면 authorityKeyIdentifier 를 찾지 못해 에러가 나므로 막아둔다.
## req_extensions = v3_user

[ v3_user ]
# Extensions to add to a certificate request
basicConstraints       = CA:FALSE
authorityKeyIdentifier = keyid,issuer
subjectKeyIdentifier   = hash
keyUsage               = nonRepudiation, digitalSignature, keyEncipherment
## SSL 용 확장키 필드
extendedKeyUsage       = serverAuth,clientAuth
subjectAltName          = @alt_names
[ alt_names ]
## Subject AltName의 DNSName field에 SSL Host 의 도메인 이름을 적어준다.
## 멀티 도메인일 경우 *.lesstif.com 처럼 쓸 수 있다.
DNS.1   = 115.68.250.15.nip.io #Master URL과 도메인 nip.io
##DNS.2   = lesstif.com
##DNS.3   = *.lesstif.com

[req_distinguished_name ]
countryName               = Country Name (2 letter code)
countryName_default       = KR
countryName_min           = 2
countryName_max           = 2

# 회사명 입력
organizationName          = Organization Name (eg, company)
organizationName_default  = lesstif Inc.

# 부서 입력
organizationalUnitName     = Organizational Unit Name (eg, section)
organizationalUnitName_default = lesstif SSL Project

# SSL 서비스할 domain 명 입력
commonName                = Common Name (eg, your name or your server's hostname)
commonName_default        = 115.68.250.15.nip.io
commonName_max            = 64
```

인증서 발급 요청(CSR) 파일을 생성한다.

```
$ openssl req -new -key lesstif.com.key -out lesstif.com.csr -config host_openssl.conf
```

```
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KR]: enter
Organization Name (eg, company) [lesstif Inc.]: enter
Organizational Unit Name (eg, section) [lesstif SSL Project]: enter
Common Name (eg, your name or your servers hostname) [115.68.250.15.nip.io]: enter
```

5년짜리 lesstif.com 용 SSL 인증서 발급 (서명시 ROOT CA 개인키로 서명)

```
openssl x509 -req -days 1825 -extensions v3_user -in lesstif.com.csr \
-CA lesstif-rootca.crt -CAcreateserial \
-CAkey lesstif-rootca.key \
-out lesstif.com.crt -extfile host_openssl.conf
-----
Signature ok
subject=C = KR, O = lesstif Inc., OU = lesstif SSL Project, CN = 115.68.250.15.nip.io
Getting CA Private Key
Enter pass phrase for lesstif-rootca.key: paas-ta
```

제대로 생성되었는지 확인을 위해 인증서의 정보를 출력

```
$ openssl x509 -text -in lesstif.com.crt
```

web server 에서 읽을수 있도록 시스템의 표준 개인키와 인증서 디렉터리에 복사

```
# 파일 생성
$ mkdir ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/keycloak_orig/tls-key/

# 복붙하는 경로 확인
$ cp lesstif.com.crt ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/keycloak_orig/tls-key/lesstif.com.crt
$ cp lesstif.com.key ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/keycloak_orig/tls-key/lesstif.com.key

# 경로 이동
$ cd ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/keycloak_orig/tls-key/

# 파일명 변경
$ mv lesstif.com.crt tls.crt
$ mv lesstif.com.key tls.key

# 확인
$ ls
tls.crt  tls.key
```

▼ CP 설치

▼ 공통 부분 CRI-O, PODMAN 설정

CRI-O insecure-registry 설정 (Master, Worker n개에 모두 설정 진행)



Private Repository에 컨테이너 플랫폼 포털 관련 이미지 및 패키지 파일 업로드 그리고 http 접속 설정을 위해 배포 전 Kubernetes **Master Node**, **Worker Node** 내 podman 설치 및 config 파일에 'insecure-registries' 설정을 진행 한다.

▼ master

Podman 설치

```
$ sudo apt-get update
$ sudo apt-get install -y podman
```

crio.conf 내 'insecure-registries' 설정

```
$ sudo vi /etc/crio/crio.conf
# 'insecure_registries' 항목에 "{K8S_MASTER_NODE_IP}:30002" 추가
...
insecure_registries = [
  "xx.xxx.xxx.xx:30002"
]
...

# crio 재시작
$ sudo systemctl restart crio
```

registries.conf 내 'registry' 항목 insecure 설정

```
$ sudo vi /etc/containers/registries.conf
# registries.conf 파일 내 '[[registry]]'가 주석 처리 되어있으므로 주석 해제 필요
# 아래 항목을 추가, location 값은 "{K8S_MASTER_NODE_IP}:30002" 설정
...
[[registry]]          #<=주석 해제 반드시
insecure = true
location = "xx.xxx.xxx.xx:30002"
...

# podman 재시작
$ sudo systemctl restart podman
```

▼ worker

Podman 설치

```
$ sudo apt-get update
$ sudo apt-get install -y podman
```

crio.conf 내 'insecure-registries' 설정

```
$ sudo vi /etc/crio/crio.conf
# 'insecure_registries' 항목에 "{K8S_MASTER_NODE_IP}:30002" 추가
...
insecure_registries = [
    "xx.xxx.xxx.xx:30002"
]
...

# crio 재시작
$ sudo systemctl restart crio
```

registries.conf 내 'registry' 항목 insecure 설정

```
$ sudo vi /etc/containers/registries.conf
# registries.conf 파일 내 '[[registry]]'가 주석 처리 되어있으므로 주석 해제 필요
# 아래 항목을 추가, location 값은 "{K8S_MASTER_NODE_IP}:30002" 설정
...
[[registry]]          #<=주석 해제 반드시
insecure = true
location = "xx.xxx.xxx.xx:30002"
...

# podman 재시작
$ sudo systemctl restart podman
```

▼ 포털 설치

컨테이너 플랫폼 포털 배포 (Master 실행)



컨테이너 플랫폼 포털 배포를 위해 컨테이너 플랫폼 포털 Deployment 파일을 다운로드 받아 아래 경로로 위치시킨다.

▼ master

컨테이너 플랫폼 포털 Deployment 파일 다운로드

```
# Deployment 파일 다운로드 경로 생성
$ mkdir -p ~/workspace/container-platform
$ cd ~/workspace/container-platform

# Deployment 파일 다운로드 및 파일 경로 확인
$ wget --content-disposition https://nextcloud.paas-ta.org/index.php/s/wYJ3wim3WCxG7Ed/download

$ ls ~/workspace/container-platform
paas-ta-container-platform-portal-deployment.tar.gz

# Deployment 파일 압축 해제
$ tar -xvf paas-ta-container-platform-portal-deployment.tar.gz
```

Deployment 파일 디렉토리 구성

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-portal-deployment

├── script      # 컨테이너 플랫폼 포털 배포 관련 변수 및 스크립트 파일 위치
├── images      # 컨테이너 플랫폼 포털 이미지 파일 위치
├── charts      # 컨테이너 플랫폼 포털 Helm Charts 파일 위치
├── values      # 컨테이너 플랫폼 포털 Helm Charts values.yaml 파일 위치
└── keycloak    # 컨테이너 플랫폼 포털 사용자 인증 관리를 위한 Keycloak 배포 관련 파일 위치
```

컨테이너 플랫폼 포털 변수 정의

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/script
$ vi container-platform-portal-vars.sh
```

```
# COMMON VARIABLE (Please change the values of the four variables below.)
K8S_MASTER_NODE_IP="3.36.89.176" # Kubernetes Master Node Public IP
K8S_AUTH_BEARER_TOKEN="eyJhbGciOiJIUzI1NiIsImtpZCI6IktzM2E3UjJfSkhadelMRWdQLW1CN09sNzVpd2FhSEUzSUZwM2xQZzAifQ.eyJpc3M"
NFS_SERVER_IP="10.0.41.184" # NFS Server Private IP
PROVIDER_TYPE="standalone" # Container Platform Portal Provider Type (Please enter 'sta
```

TLS 인증서 파일 준비(이 부분은 포털부분만 진행하면된다. (소스컨트롤, 파이프라인은 해당 안됨)이 유는 keycloak이 포털 배포할 때 배포가 되므로 이때만 하면 된다. 소스컨트롤, 파이프라인은 변수만 바꿔주면 됨)

```
# 컨테이너 플랫폼 포털 배포 전 TLS 인증서 파일 (ex: tls.key, tls.crt)이 사전에 준비되어야 한다.
# 컨테이너 플랫폼 포털 Deployment 파일 keycloak_orig 디렉토리 하위에 위치 필요
# 인증서 파일명은 tls.key, tls.crt 로 변경 필요
# 인증서 파일 권한 변경 필요

# 인증서 파일 keycloak_orig 디렉토리 하위에 위치
$ mkdir ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/keycloak_orig/tls-key
$ ls ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/keycloak_orig/tls-key
tls.crt  tls.key

# 인증서 파일 권한 변경
$ chmod ug+r ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/keycloak_orig/tls-key/*
```

```
-rw-rw-r-- 1 ubuntu ubuntu 1379 Jan 14 10:28 tls.crt
-rw-r----- 1 ubuntu ubuntu 1675 Jan 14 10:28 tls.key
```

TLS 인증서 파일 준비Dockerfile 내 인증서 파일 경로 추가

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/keycloak_orig
$ vi Dockerfile
# TLS_FILE_PATH : TLS 인증서 파일이 위치한 Deployment 파일 keycloak_orig 디렉토리 내 인증서 파일 경로
```

```
# TLS_FILE_PATH : TLS 인증서 파일이 위치한 Deployment 파일 keycloak_orig 디렉토리 내 인증서 파일 경로
FROM docker.io/jboss/keycloak:15.0.2

COPY tls-key/* /etc/x509/https/
COPY container-platform/ /opt/jboss/keycloak/themes/container-platform/
COPY {CONTAINER_PLATFORM_PORTAL_PROVIDER_TYPE}-realm.json /opt/jboss/keycloak/imports/container-platform-realm-realm.json

CMD ["-Dkeycloak.migration.action=import", "-Dkeycloak.migration.provider=singleFile", "-Dkeycloak.migration.file=/opt/jbos
```

Keycloak vlues.yaml 파일 수정

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/values_orig
$ vi paas-ta-container-platform-keycloak-values.yaml
```

```
# service.targetPort 값을 8443으로 변경 (https로 접속)

...
service:
  type: {SERVICE_TYPE}
  protocol: {SERVICE_PROTOCOL}
  port: 8080
  https:
    port: 8443
  targetPort: 8443 (수정)
  nodePort: 32710
...
```

컨테이너 플랫폼 포털 변수 파일 수정

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/script
$ vi container-platform-portal-vars.sh
```

```
# KEYCLOAK_URL 값 http -> https 로 변경
# Domain으로 nip.io를 사용하는 경우 아래와 같이 변경

# KEYCLOAK
KEYCLOAK_NAMESPACE="keycloak" # Keycloak namespace
KEYCLOAK_URL="https://\/${K8S_MASTER_NODE_IP}.nip.io:32710" # Keycloak url (include http://\/, if apply TLS, https://\
KEYCLOAK_DB_VENDOR="mariadb" # Keycloak database vendor
```

컨테이너 플랫폼 포털 배포 스크립트 실행

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-portal-deployment/script
$ chmod +x deploy-container-platform-portal.sh
$ ./deploy-container-platform-portal.sh
```

리소스 조회

```
#NFS 리소스 조회
$ kubectl get all -n nfs-storageclass

#Harbor 리소스 조회
$ kubectl get all -n harbor

#MariaDB 리소스 조회
$ kubectl get all -n mariadb

#Keycloak 리소스 조회
$ kubectl get all -n keycloak

#컨테이너 플랫폼 포털 리소스 조회
$ kubectl get all -n paas-ta-container-platform-portal
```

컨테이너 플랫폼 운영자/사용자 포털 접속



컨테이너 플랫폼 운영자포털 접속 URI : **http://{K8S_MASTER_NODE_IP}:32703**
 컨테이너 플랫폼 사용자포털 접속 URI : **http://{K8S_MASTER_NODE_IP}:32702**

▼ 서비스 설치(sc, pl)

▼ 소스컨트롤

▼ master

컨테이너 플랫폼 소스 컨트롤 Deployment 파일 다운로드

```
# Deployment 파일 다운로드 경로 생성
$ mkdir -p ~/workspace/container-platform
$ cd ~/workspace/container-platform
```

```
# Deployment 파일 다운로드 및 파일 경로 확인
$ wget --content-disposition https://nextcloud.paas-ta.org/index.php/s/6W69C29tjQ6Y8We/download

$ ls ~/workspace/container-platform
...
paas-ta-container-platform-source-control-deployment.tar.gz
...

# Deployment 파일 압축 해제
$ tar xvfz paas-ta-container-platform-source-control-deployment.tar.gz
```

이미지 파일 디렉토리 구성

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-source-control-deployment

├─ script      # 컨테이너 플랫폼 소스 컨트롤 배포 관련 변수 및 스크립트 파일 위치
├─ images      # 컨테이너 플랫폼 소스 컨트롤 이미지 파일 위치
├─ charts      # 컨테이너 플랫폼 소스 컨트롤 Helm Charts 파일 위치
└─ values      # 컨테이너 플랫폼 소스 컨트롤 Helm Charts values.yaml 파일 위치
```

컨테이너 플랫폼 소스 컨트롤 변수 정의

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-source-control-deployment/script
$ vi container-platform-source-control-vars.sh
```

```
#!/bin/bash
K8S_MASTER_NODE_IP="3.36.89.176" # Kubernetes master node public ip
PROVIDER_TYPE="standalone" # Container platform source control provider type (Please enter 'standalone' or 'se
```

▼ openstack

```
#!/bin/bash
K8S_MASTER_NODE_IP="10.100.1.211" # Kubernetes master node public ip
PROVIDER_TYPE="service" # Container platform source control provider type (Please enter 'standalone' or 'se
```

포털 설치 할 때 사전에

▼ tls일 경우 Keycloak TLS 설정이 되어 있어야 한다.

```
#!/bin/bash
K8S_MASTER_NODE_IP="115.68.250.15" # Kubernetes master node public ip
PROVIDER_TYPE="service" # Container platform pipeline provider type (Please enter 'standalone' or 'service')
CF_API_URL="https://api.10.0.0.120.nip.io" # e.g) https://api.10.0.0.120.nip.io, PaaS-T
KEYCLOAK_URL="https://\/${K8S_MASTER_NODE_IP}.nip.io:32710" #include http://\/, if apply TLS, https://\//
```

컨테이너 플랫폼 소스 컨트롤 배포 스크립트 실행

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-source-control-deployment/script
$ chmod +x deploy-container-platform-source-control.sh
$ ./deploy-container-platform-source-control.sh
```

컨테이너 플랫폼 소스 컨트롤 확인

```
# 소스 컨트롤 리소스 확인
$ kubectl get all -n paas-ta-container-platform-source-control
```



관리자: http://{K8S_MASTER_NODE_IP}:30084

사용자: Keycloak(http://{K8S_MASTER_NODE_IP}:32710)

▼ 파이프라인

▼ master

컨테이너 플랫폼 파이프라인 Deployment 파일 다운로드

```
# Deployment 파일 다운로드 경로 생성
$ mkdir -p ~/workspace/container-platform
$ cd ~/workspace/container-platform

# Deployment 파일 다운로드 및 파일 경로 확인
$ wget --content-disposition https://nextcloud.paas-ta.org/index.php/s/6BDzar68ck5jryq/download

$ ls ~/workspace/container-platform
...
paas-ta-container-platform-pipeline-deployment.tar.gz
...
# Deployment 파일 압축 해제
$ tar xvfz paas-ta-container-platform-pipeline-deployment.tar.gz
```

이미지 파일 디렉토리 구성

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-pipeline-deployment

├─ script      # 컨테이너 플랫폼 파이프라인 배포 관련 변수 및 스크립트 파일 위치
├─ images      # 컨테이너 플랫폼 파이프라인 이미지 파일 위치
├─ charts      # 컨테이너 플랫폼 파이프라인 Helm Charts 파일 위치
└─ values      # 컨테이너 플랫폼 파이프라인 Helm Charts values.yaml 파일 위치
```

컨테이너 플랫폼 파이프라인 변수 정의

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-pipeline-deployment/script
$ vi container-platform-pipeline-vars.sh
```

```
#!/bin/bash
K8S_MASTER_NODE_IP="115.68.250.15" # Kubernetes master node public ip
PROVIDER_TYPE="service" # Container platform pipeline provider type (Please enter 'standalone' or 'service')
CF_API_URL="https://api.10.0.0.120.nip.io" # e.g) https://api.10.0.0.120.nip.io, PaaS-TA API Domain, PROVIDER_T
#/workspace/portal-deployment/portal-api의 haproxy_public_ip 참고
```

▼ tls일 경우

```
#!/bin/bash
K8S_MASTER_NODE_IP="115.68.250.15" # Kubernetes master node public ip
PROVIDER_TYPE="service" # Container platform pipeline provider type (Please enter 'standalone' or 'service')
CF_API_URL="https://api.10.0.0.120.nip.io" # e.g) https://api.10.0.0.120.nip.io, PaaS-T

KEYCLOAK_URL="https://\/${K8S_MASTER_NODE_IP}.nip.io:32710" #include http://, if apply TLS, https://
```

컨테이너 플랫폼 파이프라인 배포 스크립트 실행

```
$ cd ~/workspace/container-platform/paas-ta-container-platform-pipeline-deployment/script
$ chmod +x deploy-container-platform-pipeline.sh
$ ./deploy-container-platform-pipeline.sh
```

컨테이너 플랫폼 소스 컨트롤 확인

```
# 소스 컨트롤 리소스 확인
$ kubectl get all -n paas-ta-container-platform-pipeline
```



관리자: http://{K8S_MASTER_NODE_IP}:30084

사용자: Keycloak(http://{K8S_MASTER_NODE_IP}:32710)