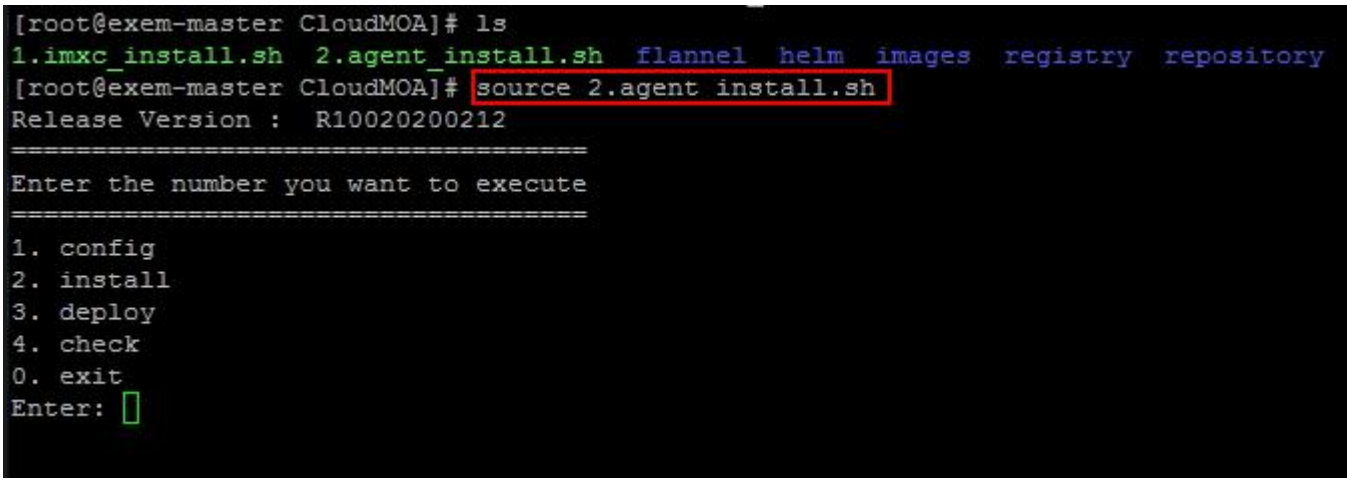
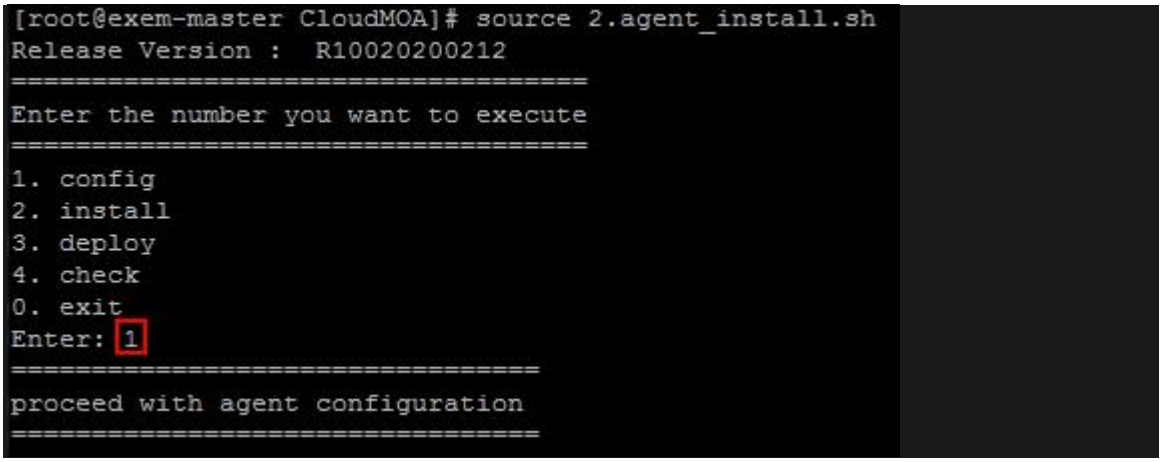
2.2. CloudMOA Agent 설치

CloudMOA agent 는 관제 대상 클러스터에 설치되는 모듈입니다. 배포 편이성을 위해 주요 모듈 및 기반 소프트웨어들을 Docker 이미지화 하였습니다. CloudMOA agent 설치는 Ansible 과 Helm 을 통해 관제 대상 클러스터에 설치되도록 진행되며, 모든 설치는 수집 서버 클러스터의 Master 노드에서 진행합니다.

1. "Source 2. Agent_install.sh" 로 실행합니다. (환경변수 적용을 위해 source 명령어로 실행해야 합니다.)



2. Main 화면에서 "1. Config" 실행합니다.



A. 수집서버의 Master 노드 IP 입력합니다.

B. Intermax APM 이 설치되어 있는 경우 Intermax APM 이 설치된 IP 입력합니다.

C. CloudMOA agent 설치파일(yaml 파일)을 위치시킬 경로를 입력합니다.

```
Enter the path to install CloudMOA agent
Enter: /home/cloudmoa agent
```

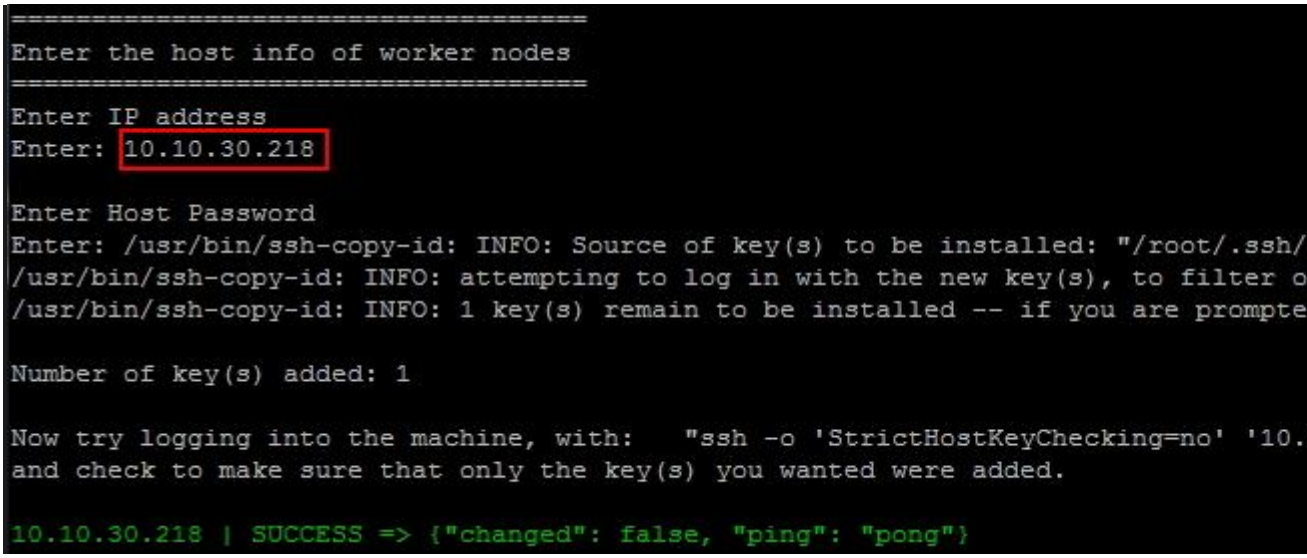
D. CloudMOA agent 설치를 진행할 클러스터들 각각의 Master 노드 IP / PW / 클러스터명 입력합니다. (클러스터명이 따로 없는 경우 원하는 이름으로 설정합니다.)

```
====================================
Enter the host info of master nodes
====================================
Enter IP address
Enter: 10.10.30.217

Enter Host Password
Enter:

Enter your cluster name
Enter: agent cluster
 [WARNING]: Could not match supplied host pattern, ignoring: all

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

 [WARNING]: Could not match supplied host pattern, ignoring: masters

 [WARNING]: No hosts matched, nothing to do

The authenticity of host '10.10.30.217 (10.10.30.217)' can't be established.
ECDSA key fingerprint is SHA256:9er5us2IKRiTX53sXpuC1ehlHbG23lrMxSA/kSboa7c.
ECDSA key fingerprint is MD5:82:43:0d:09:30:24:e1:7b:09:4b:1a:b5:4b:80:26:d5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.30.217' (ECDSA) to the list of known hosts.
root@10.10.30.217's password:
scp: /tmp/imxc/: Is a directory
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter o
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompte

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh -o 'StrictHostKeyChecking=no' '10.
and check to make sure that only the key(s) you wanted were added.

10.10.30.217 | SUCCESS => {"changed": false, "ping": "pong"}
```

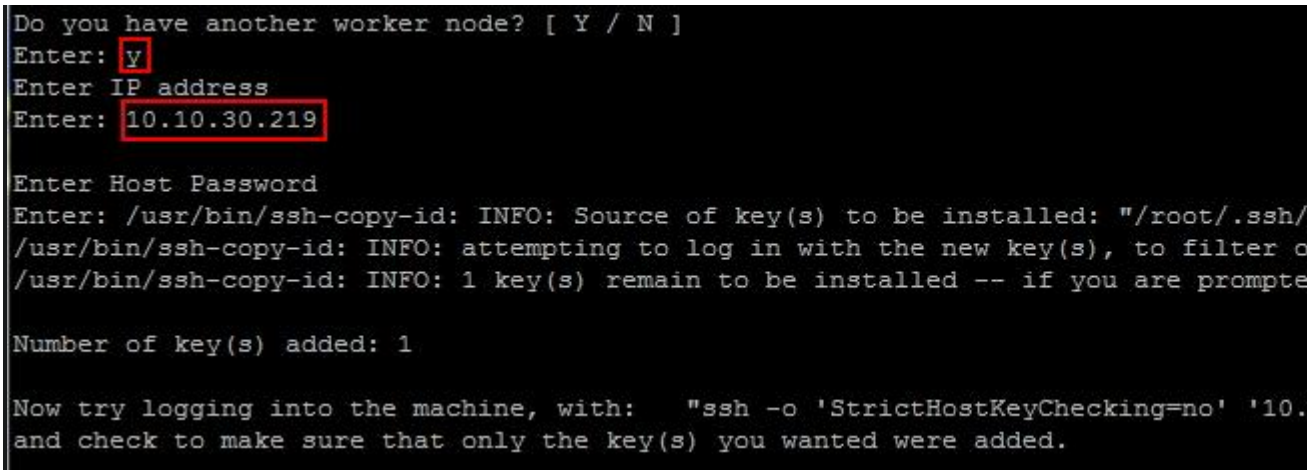
E. 다른 Master 노드 추가 여부 입력합니다.

```
Do you have another master node? [ Y / N ]
Enter: n
```

F. CloudMOA agent 설치를 진행할 클러스터의 Worker 노드 IP 입력합니다.



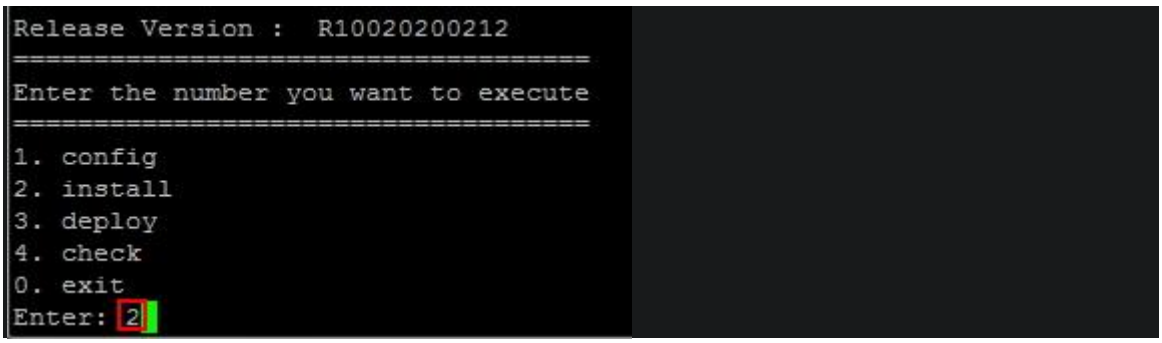
G. Worker 노드 추가합니다.



H. CloudMOA agent 설치를 진행할 클러스터에 수집서버 Doker Private Registry URL 세팅 후 Docker 재시작 합니다.

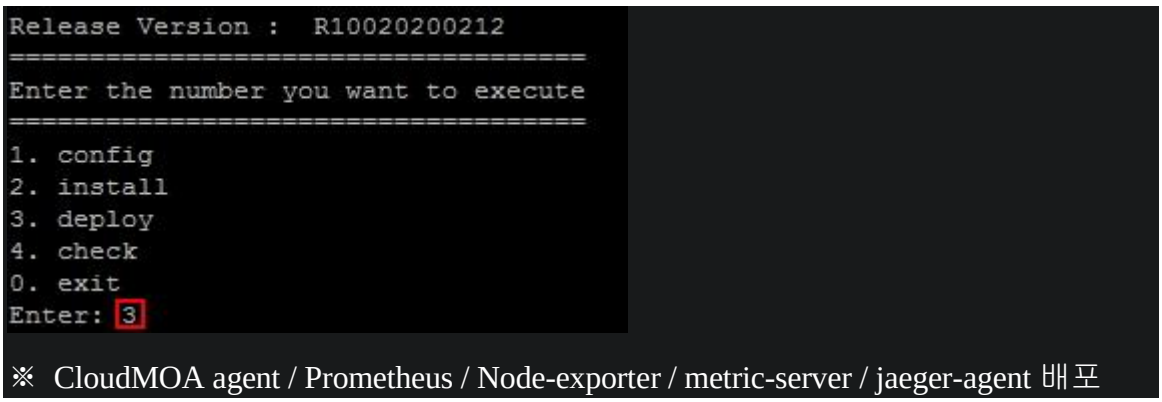
```
For using private registry, You should restart docker. Do you want to progress th
Enter: y
10.10.30.217 | SUCCESS => {
    "backup": "",
    "changed": true,
    "msg": "line added"
}
 [WARNING]: Consider using file module with state=directory rather than running m

10.10.30.217 | SUCCESS | rc=0 >>


 [WARNING]: Consider using file module with state=touch rather than running touch

10.10.30.217 | SUCCESS | rc=0 >>


10.10.30.217 | SUCCESS => {
    "backup": "",
    "changed": false,
    "msg": ""
}
10.10.30.217 | SUCCESS | rc=0 >>


10.10.30.218 | SUCCESS => {
    "backup": "",
    "changed": true,
    "msg": "line added"
}
10.10.30.219 | SUCCESS => {
    "backup": "",
    "changed": true,
    "msg": "line added"
}
 [WARNING]: Consider using file module with state=directory rather than running m

10.10.30.219 | SUCCESS | rc=0 >>


10.10.30.218 | SUCCESS | rc=0 >>


 [WARNING]: Consider using file module with state=touch rather than running touch

10.10.30.218 | SUCCESS | rc=0 >>


10.10.30.219 | SUCCESS | rc=0 >>


10.10.30.218 | SUCCESS => {
    "backup": "",
    "changed": false,
    "msg": ""
}
10.10.30.219 | SUCCESS => {
    "backup": "",
    "changed": false,
    "msg": ""
}
10.10.30.218 | SUCCESS | rc=0 >>
```

3. Main 화면에서 "2. Install" 실행합니다.



```
 [WARNING]: Consider using file module with state=directory rather than running mkdi

10.10.30.217 | SUCCESS | rc=0 >>


10.10.30.217 | SUCCESS => {
    "changed": true,
    "dest": "/home/cloudmoa_agent/",
    "src": "/home/CloudMOA/helm/agent"
}
10.10.30.217 | SUCCESS => {
    "changed": true,
    "dest": "/home/cloudmoa_agent/",
    "src": "/home/CloudMOA/helm/kubernetes"
}
10.10.30.217 | SUCCESS => {
    "changed": true,
    "dest": "/home/cloudmoa_agent/",
    "src": "/home/CloudMOA/helm/linux-amd64"
}
10.10.30.217 | SUCCESS => {
    "changed": true,
    "checksum": "e8988ef08fe4bf2c4f43b8aec8bc7ddcfd2a1582",
    "dest": "/home/cloudmoa_agent/tiller-rbac.yaml",
    "gid": 0,
    "group": "root",
    "md5sum": "eb4d03a4ab258c3409ecfe200b16d0cc",
    "mode": "0644",
    "owner": "root",
    "secontext": "unconfined_u:object_r:user_home_t:s0",
    "size": 254,
    "src": "/root/.ansible/tmp/ansible-tmp-1582632281.35-222159377431219/source",
    "state": "file",
    "uid": 0
}
10.10.30.217 | SUCCESS => {
    "changed": true,
    "checksum": "ce7022a85ad2ba7747d2da9524fce113e0a6c30a",
    "dest": "/usr/local/bin/helm",
    "gid": 0,
    "group": "root",
    "md5sum": "e6826316087692dedd619fe479e07a20",
    "mode": "0644",
    "owner": "root",
    "secontext": "system_u:object_r:bin_t:s0",
    "size": 38789120,
    "src": "/root/.ansible/tmp/ansible-tmp-1582632283.06-230904353015578/source",
    "state": "file",
    "uid": 0
}
 [WARNING]: Consider using file module with mode rather than running chmod

10.10.30.217 | SUCCESS | rc=0 >>


10.10.30.217 | SUCCESS => {
    "changed": false,
    "checksum": "4c8dc250a102f1b3be3de4e5b5d47ede73e195c7",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "path": "/home/cloudmoa_agent/agent/values.yaml",
```

4. Main 화면에서 "3. Deploy" 실행합니다.



※ CloudMOA agent / Prometheus / Node-exporter / metric-server / jaeger-agent 배포

> *Service Tracing 를 가능하게 하는 Jaeger Agent 가 고객사의 java Application 에 옵*
> *Java application 에 적용시 아래의 옵션을 입력해주시기 바랍니다.*

application 에 java option 으로 specialagent 를 같이 기동하도록 변경합니다.

```
$ java -Dsa.spring -Dsa.tracer=jaeger -
Djava.security.egd=file:/dev/./urandom -noverify -
javaagent:/opentracing-specialagent-1.3.3.jar -jar /app.jar
```

a. 고객사 **deployment.yaml** 파일에 아래의 환경변수 추가합니다.

jaeger 를 사용하기 위해선 고객사의 yaml 파일(deployment 등)에 아래의 환경변수 설정 내용이 추가되어야 하며 아래의 CLST_ID 환경변수의 value 값에 적당한 클러스터 이름을 입력해주시기 바랍니다.

```
apiVersion: v1
...
        env:
          - name: CLST_ID
            value: $CLUSTER_ID # need to edit
          - name: NODE_ID
            valueFrom:
              fieldRef:
                fieldPath: spec.nodeName
          - name: NAMESPACE
            valueFrom:
              fieldRef:
                fieldPath: metadata.namespace
          - name: POD_ID
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
          - name: JAEGER_SERVICE_NAME
            valueFrom:
              fieldRef:
                fieldPath: metadata.labels['app']
```

```
          - name: JAEGER_AGENT_HOST
            value: jaeger-agent
          - name: JAEGER_TAGS
            value:
xm_clst_id=$(CLST_ID),xm_node_id=$(NODE_ID),xm_namespace=$(
NAMESPACE),xm_pod_id=$(POD_ID)
```

5. 배포가 완료되면 Main 화면의 "4. Check" 실행을 통해 설치가 정상적으로 완료되었는지 확인 후 정상적으로 파드가 Running 되고 있다면 설치가 완료됐다고 판단하시면 됩니다.