**Cloudit™**

**DCE - PaaSTA 연동**

**가이드**

**문서버전 V0.2**

**법적공지**

Copyright ⓒ 2012 - 2019 ㈜이노그리드가 모든 판권을 소유 합니다. 이 제품은 대한민국 및 국제저작법과 지적 재산권법의 보호를 받습니다. Specifications are subject to change without notice. Innogrid, the Innogrid logo, Innogrid Cloudit, Innogrid Cloudit IaaS, Innogrid Cloudit 서버, Innogrid Cloudit 스토리지, Innogrid Cloudit Console are trademarks or registered trademarks of Innogrid. All other brands or products are trademarks or registered trademarks of their respective holders.

**㈜ 이노그리드**

서울시 서초구 서초대로 314, 10층 (서초동, 정보통신공제조합빌딩)

[www.innogrid.com](http://www.innogrid.com)

# 1. Cloudit 연동 설정

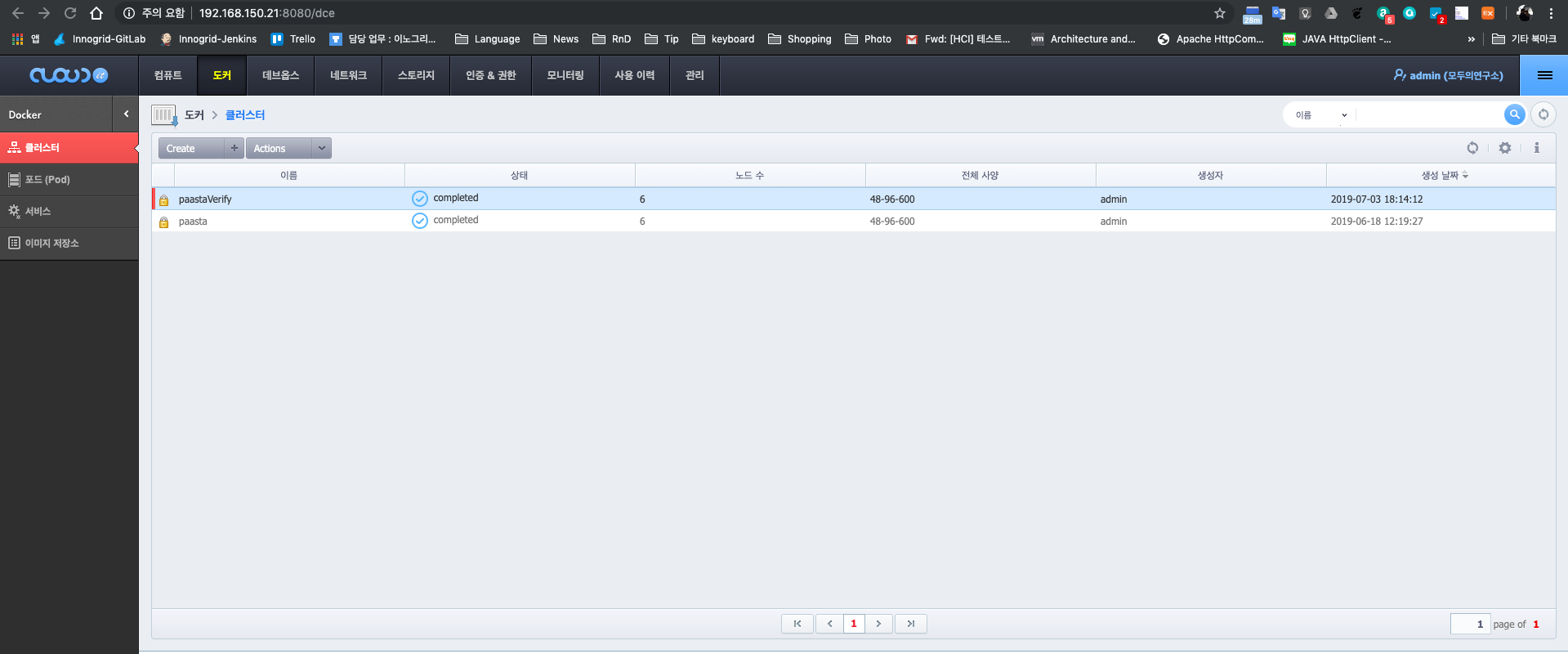
## 1.1. DCE 클러스터 생성

## 1.1.1 준비사항

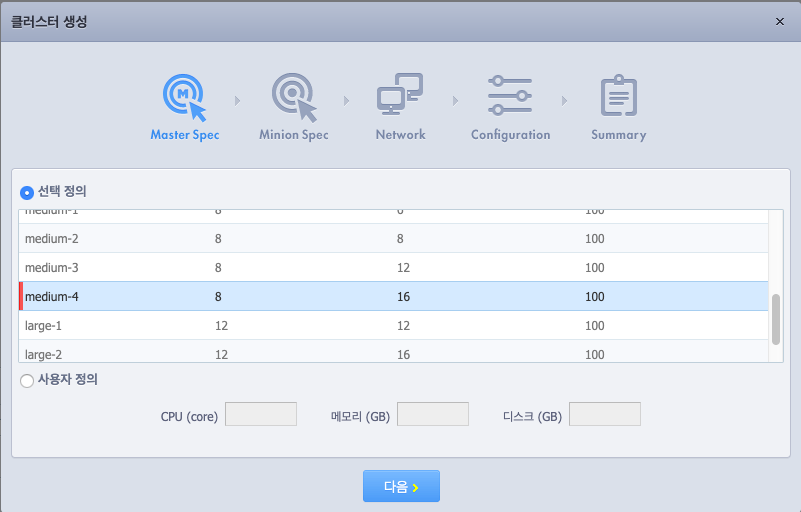
* Master VM 사양: medium-4(8-16-100) 이상
* Minion VM 사양: medium-4(8-16-100) 이상
* PaaS-TA 전용 Template 사용

## 1.1.2 DCE 클러스터 생성

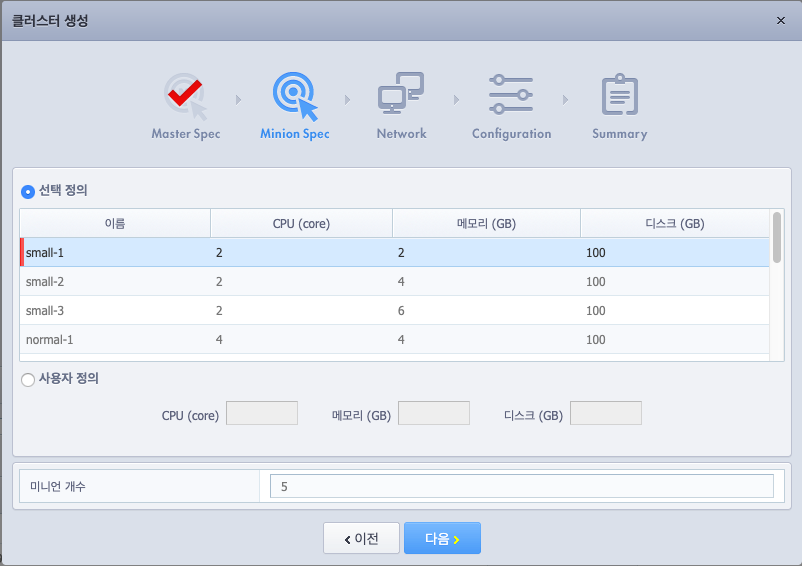
* "Create" 버튼 클릭



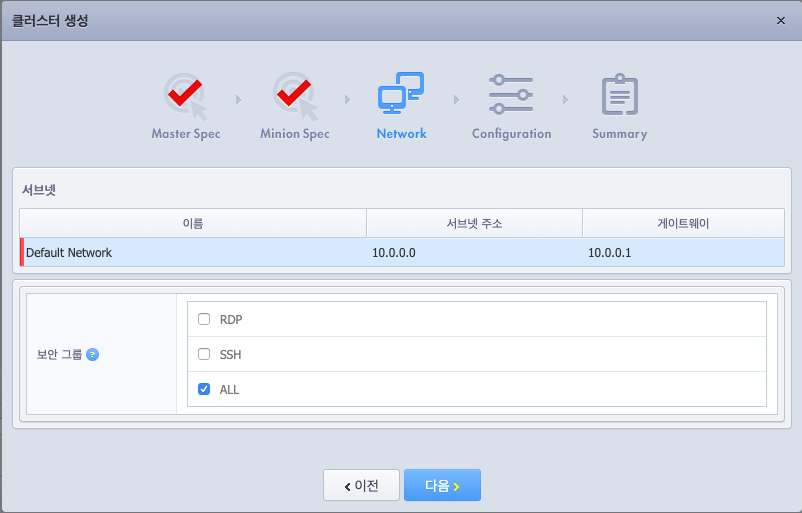
* Master Node Spec 설정(medium-4 이상)



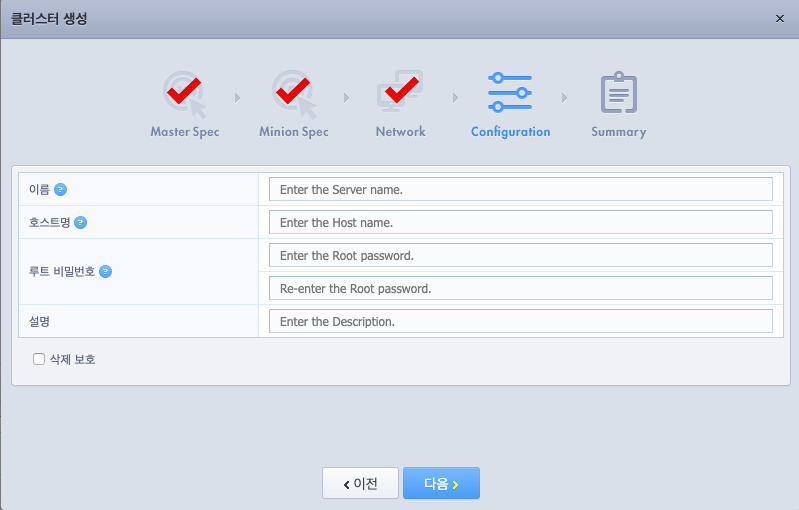
* Minion Node Spec 및 개수 설정(medium-4 이상, 최소 5노드 이상 권장)



* 보안 그룹 설정(ALL)



* 클러스터 노드(Master, Minion) 관련 설정



* 생성 요약 정보 확인 및 생성 진행



## 1.2. Ceph 설정 및 인증키 확인

## 1.2.1 준비사항

* Ceph CLI(Command LIne Interface) 환경 및 접속 가능한 Ceph 환경

## 1.2.2 Kubernetes - Ceph 연동을 위한 Ceph 관리자 인증키 확인

|  |
| --- |
| [root@CephNode:~#] ceph --cluster ceph auth get-key client.admin  **A2A1ZsFcQTP2IhAAlxgclKE/uv2qy6eFUrPmKg==** |

* 확인된 관리자 인증키는 **"1.3 Kubernetes 설정"**에서 사용

## 1.2.3 Kubernetes Storage Class 연동을 위한 Ceph Pool 생성 및 접근 권한 설정

* 환경에 적합한  ***[pool\_name], [client\_name]*** 명명 및 적용
* 환경에 적합한 ***[pg-num], [pgp-num]*** 설정
  + PG계산기: https://ceph.com/pgcalc/

|  |
| --- |
| [root@CephNode:~#] ceph --cluster ceph osd pool create ***[pool\_name]*** \  ***[pg-num] [pgp-num]***  [root@CephNode:~#] ceph --cluster ceph auth get-or-create \  client.***[client\_name]*** mon 'allow r' osd 'allow rwx pool=***[pool\_name]'***  [root@CephNode:~#] ceph --cluster ceph auth get-or-create \ client.***[client\_name]*** |

* 확인된 사용자 인증키는 **"1.3 Kubernetes 설정 및 Ceph Storage Class 생성"**에서 사용

## 1.3. Kubernetes 설정 및 Ceph Storage Class 생성

## 1.3.1 준비사항

* Master Node에 접속 가능한 DCE Cluster 환경
* Ceph 연동을 위한 Ceph 관리자, 사용자 인증 키 및 Pool(1.2에서 생성)

## 1.3.2 rbd-provisioner Cluster Role/Role 생성 및 바인딩

* YAML 파일 생성

|  |
| --- |
| [root@DceMaster:~#] vi ClusterRole.yaml  kind: ClusterRole  apiVersion: rbac.authorization.k8s.io/v1  metadata:  name: rbd-provisioner  rules:  - apiGroups: [""]  resources: ["persistentvolumes"]  verbs: ["get", "list", "watch", "create", "delete"]  - apiGroups: [""]  resources: ["persistentvolumeclaims"]  verbs: ["get", "list", "watch", "update"]  - apiGroups: ["storage.k8s.io"]  resources: ["storageclasses"]  verbs: ["get", "list", "watch"]  - apiGroups: [""]  resources: ["events"]  verbs: ["create", "update", "patch"]  - apiGroups: [""]  resources: ["services"]  resourceNames: ["kube-dns","coredns"]  verbs: ["list", "get"]  - apiGroups: [""]  resources: ["endpoints"]  verbs: ["get", "list", "watch", "create", "update", "patch"]  ---  kind: ClusterRoleBinding  apiVersion: rbac.authorization.k8s.io/v1  metadata:  name: rbd-provisioner  subjects:  - kind: ServiceAccount  name: rbd-provisioner  namespace: kube-system  roleRef:  kind: ClusterRole  name: rbd-provisioner  apiGroup: rbac.authorization.k8s.io  ---  apiVersion: rbac.authorization.k8s.io/v1beta1  kind: Role  metadata:  name: rbd-provisioner  rules:  - apiGroups: [""]  resources: ["secrets"]  verbs: ["get"]  ---  apiVersion: rbac.authorization.k8s.io/v1  kind: RoleBinding  metadata:  name: rbd-provisioner  roleRef:  apiGroup: rbac.authorization.k8s.io  kind: Role  name: rbd-provisioner  subjects:  - kind: ServiceAccount  name: rbd-provisioner  namespace: kube-system  ---  apiVersion: v1  kind: ServiceAccount  metadata:  name: rbd-provisioner  ---  apiVersion: extensions/v1beta1  kind: Deployment  metadata:  name: rbd-provisioner  spec:  replicas: 1  strategy:  type: Recreate  template:  metadata:  labels:  app: rbd-provisioner  spec:  containers:  - name: rbd-provisioner  image: "quay.io/external\_storage/rbd-provisioner:latest"  env:  - name: PROVISIONER\_NAME  value: ceph.com/rbd  serviceAccount: rbd-provisioner |

* kubectl create 명령어를 통해 생성 실행

|  |
| --- |
| [root@DceMaster:~#] kubectl create -n kube-system -f ClusterRole.yaml |

## 1.3.3 Ceph 연동을 위한 Ceph 관리자 인증 Secret 생성

* **"1.2.2"**에서 확인된 관리자 인증키를 이용한 Kubernetes Secret 생성

|  |
| --- |
| [root@DceMaster:~#] kubectl create secret generic ceph-secret \  --type="kubernetes.io/rbd" \  --from-literal=key=***[ceph 관리자 인증키]*** \  --namespace=kube-system |

## 1.3.4 Ceph 연동을 위한 Ceph 사용자 인증 Secret 생성]

* **"1.2.3"**에서 확인된 사용자 인증키를 이용한 Kubernetes Secret 생성
* ***[client\_name]***: "1.2.3"에서 생성한 Ceph Client명

|  |
| --- |
| [root@DceMaster:~#]kubectl create secret generic  ceph-secret-***[client\_name]*** \  --type="kubernetes.io/rbd" \  --from-literal=key=***[ceph 사용자 인증키]*** \  --namespace=kube-system |

## 1.3.5 Kubernetes Storage Class 생성

* YAML 파일 생성
* ***[StorageClass Name]***: Storage Class 명칭 명명
* ***[monitor-x-ip]***: 연동 대상 Ceph의 monitor 노드 IP
* ***[client\_name]***: "1.2.3"에서 생성한 Ceph Client명

|  |
| --- |
| [root@DceMaster:~#]vim StorageClass.yaml  apiVersion: storage.k8s.io/v1  kind: StorageClass  metadata:  name: ***[StorageClass Name]***  provisioner: ceph.com/rbd  parameters:  monitors: ***[monitor-1-ip]***:6789, ***[monitor-2-ip]***:6789, ***[monitor-3-ip]***:6789  adminId: admin  adminSecretName: ceph-secret  adminSecretNamespace: kube-system  pool: kube  userId: kube  userSecretName: ceph-secret-***[client\_name]***  userSecretNamespace: kube-system  imageFormat: "2"  imageFeatures: layering |

* kubectl create 명령어를 통해 생성 실행

|  |
| --- |
| [root@DceMaster:~#] kubectl create -f StorageClass.yaml |

## 1.3.6 Storage Class 정상 생성 확인을 위한 PVC 생성 테스트

* YAML 파일 생성
* ***[PVC\_name]***: 생성할 PVC명
* ***[StorageClass\_name]***: 생성할 Storage Class명

|  |
| --- |
| [root@DceMaster:~#]vim testPVC.yaml  kind: PersistentVolumeClaim  apiVersion: v1  metadata:  name: ***[PVC\_name]***  spec:  accessModes:  - ReadWriteOnce  resources:  requests:  storage: 8Gi  storageClassName: ***[StorageClass\_name]*** |

* kubectl create 명령어를 통해 생성 실행

|  |
| --- |
| [root@DceMaster:~#] kubectl create -f testPVC.yaml |

* 정상 생성 결과

|  |
| --- |
| [root@DceMaster:~#] kubectl get pvc myclaim  NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS AGE  ***[PVC\_name]*** Bound pvc-119e 8Gi RWO ***[StorageClass\_name]*** 1h |