

OpenStack 기반의 IaaS, PaaS 통합 Orchestration 플랫폼

DCOSTACK 소개 및 활용방안

2018.06

MTOSsolutions
"We actualize what you virtualize"

CONTENTS

I. DCOSTACK 소개 및 특징

II. DCOSTACK 활용방안

III. DCOSTACK 활용 사례

I. DCOSTACK 소개 및 특징

1. Why OpenStack?

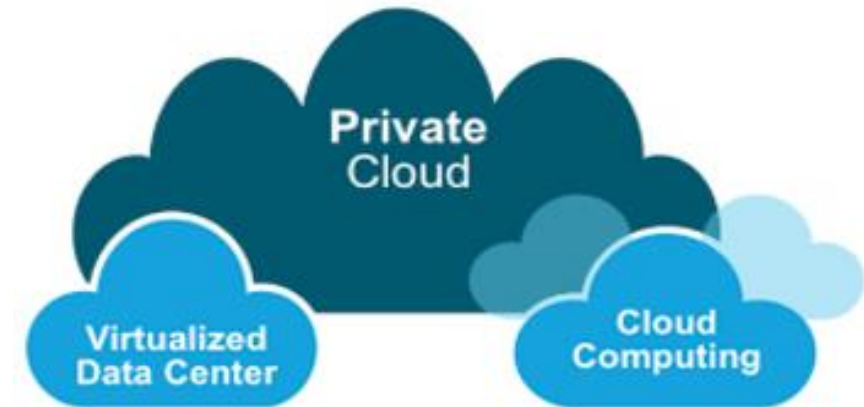
부족한 예산, 인력 부족

새로운 플랫폼 도입

통합 Platform

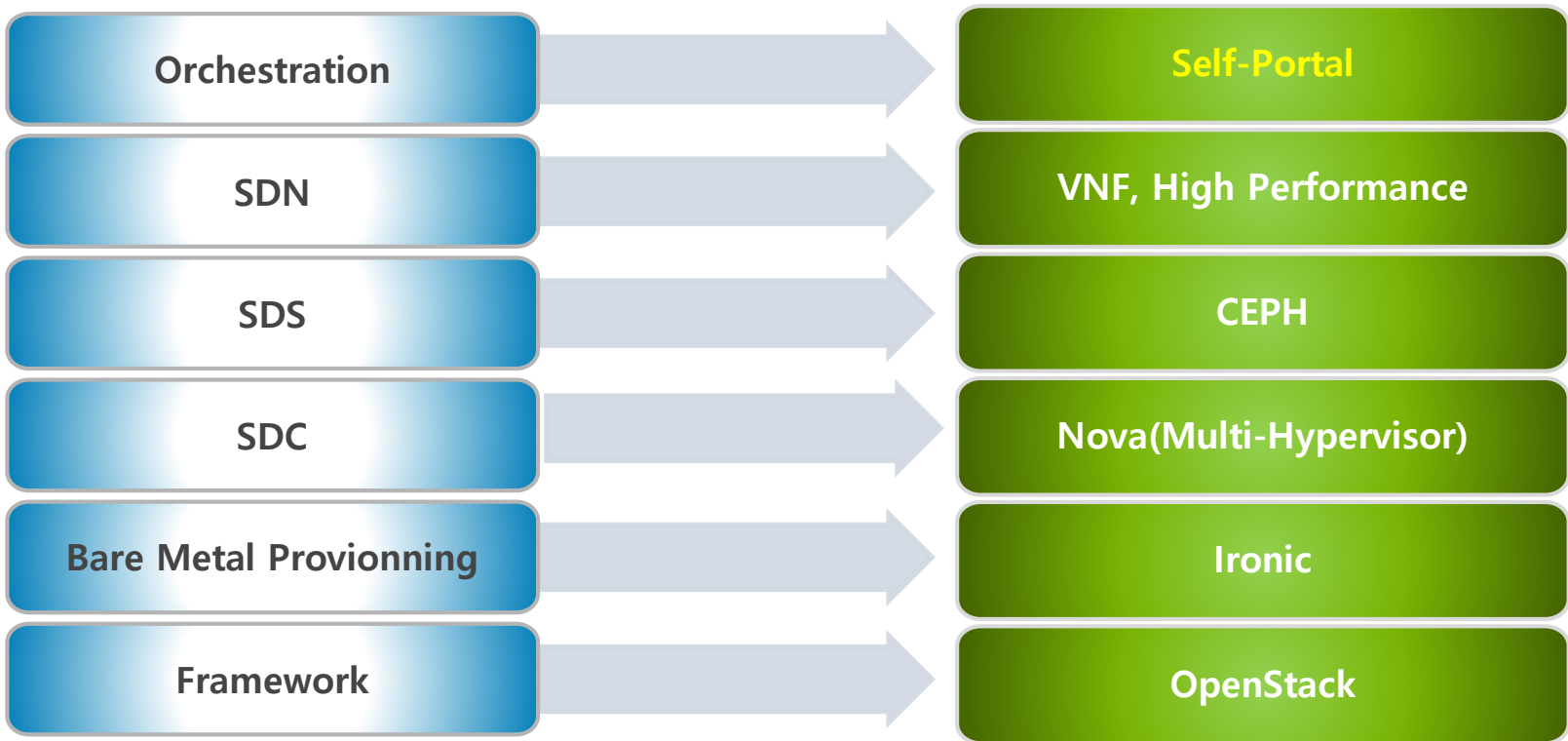
짧은 Service Life Cycle

기술력 내제화



2. DCOSTACK Product Positioning

- ✓ OpenStack을 Framework로 하여 각각의 Component를 가상화 및 Orchestration
- ✓ Self-portal에서 각각의 Component 설정, 자동화, 관리



3. OpenStack 안정성

✓ OpenStack의 역할 ?

- 가상리소스(VM, Volume) 구성 및 삭제 : 하드웨어 리소스를 사용하거나 반환하는 도구의 역할
- 가상리소스에 대한 Life Cycle 관리 기능 제공

✓ OpenStack 실제 운용 사례

▪ N사 Private Cloud

- OpenStack기반 제품(DCOSTACK)을 데이터센터에 적용
- 적용된 OpenStack Project → *Nova, Cinder, Neutron, Glance, Horizon* 등

▪ Private Cloud 장애 사례

- 최근 2년여 간 운영 중 Private Cloud 영역 장애 : 총 10여건
- 장애유형 : 하드웨어장애 / 커널BUG / VM운영체제 장애 등 DCOSTACK과 직접 관련 없는 장애로 분석

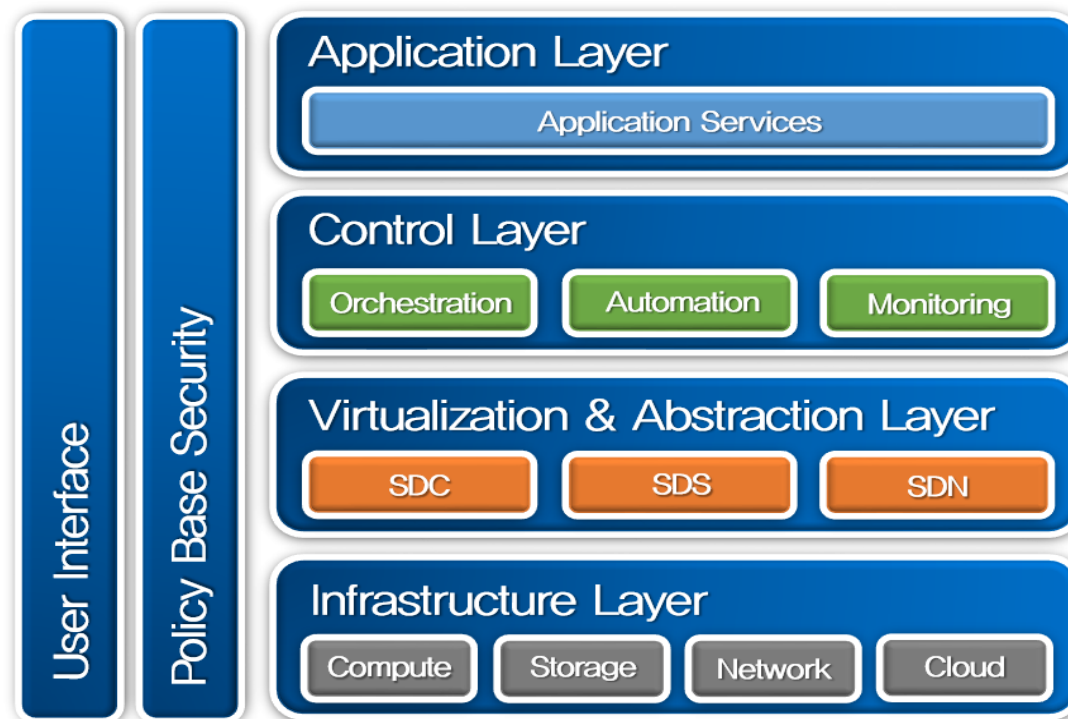
- 도입 안정화 이후 신규 패치 적용 전 검증절차 강화 및 운영기술 역량 내재화로 인해 OpenStack과 직접 연관된 장애는 발생하고 있지 않음

✓ Summary

- **DCOSTACK** 은 가상리소스를 구성하고 라이프 사이클을 관리하는 “**도구**”의 역할 !
- OpenStack의 장애 발생시에도 기존 운용되는 **VM 서비스에는 영향이 없으며**, DCOSTACK 제품 운용 중에 자체 문제로 인해 보고된 서비스 장애 이력 없음 → **오픈스택기반 제품의 안정성 검증**
참고) OpenStack에 장애 발생시에도 Running중인 VM에는 영향 없음 (추가적인 VM 생성,삭제,변경만 불가능)

4. DCOSTACK 소개

- ✓ DCOSTACK : Data Center Operating Stack
- ✓ 인프라 추상화를 기반으로 하는 데이터센터 자동화 및 오케스트레이션 플랫폼
- ✓ 고성능, 안정성, 편의성을 제공하는 IaaS, PaaS 통합 자동화 솔루션



5. DCOSTACK 특징

- ✓ On-Premise 환경을 포함하는 자체 데이터센터 Private Cloud
- ✓ Public Cloud 와 Private Cloud를 통합하는 Hybrid Cloud
- ✓ 오픈소스를 기반으로 개발된 소프트웨어 기반의 데이터센터 자동화 플랫폼



- **하나의 인터페이스로 다수의 데이터센터 통합 운용** : Multi Data Center, Unified Interface
- **DCOS(DataCenter Operating System) 모듈 구조** : 모듈화된 기능별 컴포넌트의 조합을 통한 솔루션 구성
- **Business Focus** : 클라우드 핵심 특성 지원 및 데이터센터 자동화

Public Cloud	Private Cloud	Software Defined Data Center
<ul style="list-style-type: none">• 빠른 요구 탄력성 (Elasticity)• 주문형(On-demand) 셀프서비스• 사용량 측정 (Metering)• 자원 공동관리 (Pooling)	<ul style="list-style-type: none">• 빠른 요구 탄력성 (Elasticity)• 주문형(On-demand) 셀프서비스• 사용량 측정 (Metering)• 자원 공동관리 (Pooling)• 관리성 (Manageability)• 보안성 (Security)	<ul style="list-style-type: none">• 프로세스 단순화 (Simplified)• 서비스 제공 자동화 (Automation)• 소프트웨어 정의 서비스 <p><i>SDx : Software Defined Anything</i> (ex : SDC, SDS, SDN ...)</p>



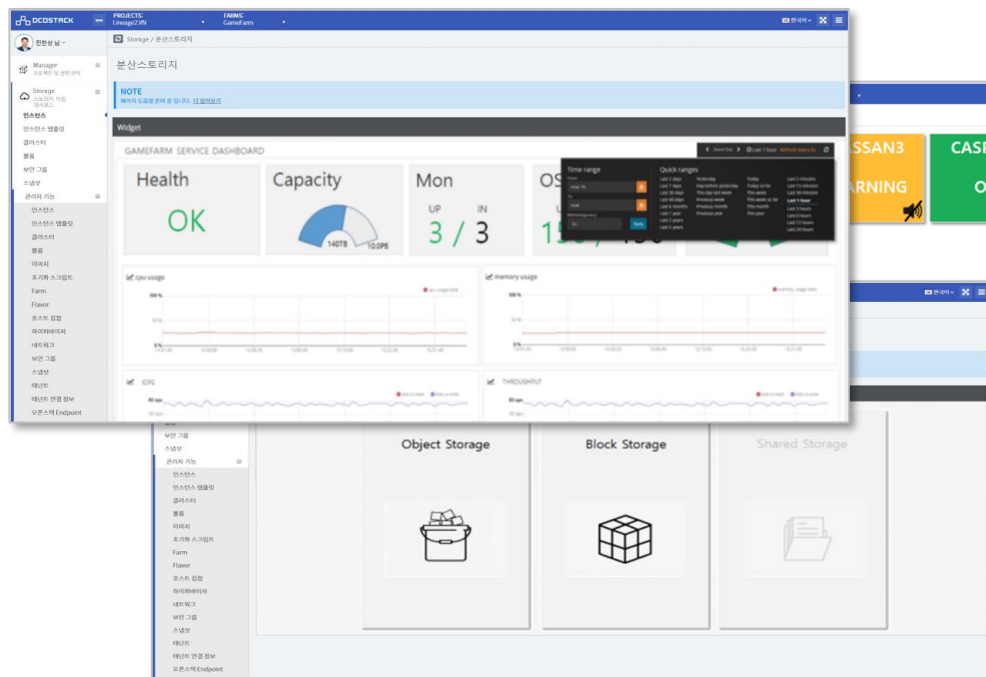
I. DCOSTACK 소개 및 특징

7. DCOSTORAGE - DCOSTACK SDS Model

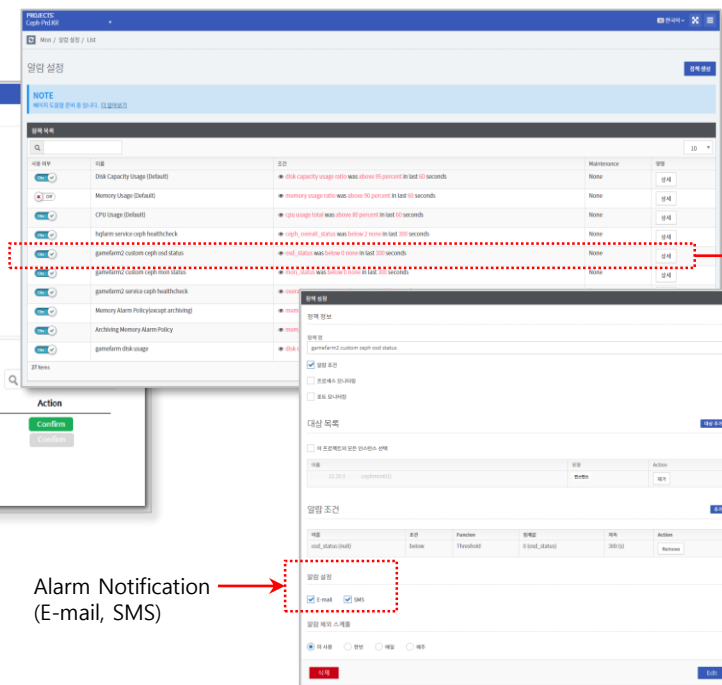
- ✓ 분산 구조 기반의 Ceph 스토리지 가상화 서비스 : Block Storage, Object Storage 제공
- ✓ Ceph 모니터링 및 DCOMON과 연계한 알람 도구 제공
- ✓ 별도의 독립된 스토리지 서비스, 또는 DCOSTACK에 통합된 스토리지 서비스 형태로 제공



[DCOSTORAGE > Ceph Monitoring / Admin / Management]



[DCOMON > Ceph Alarm]



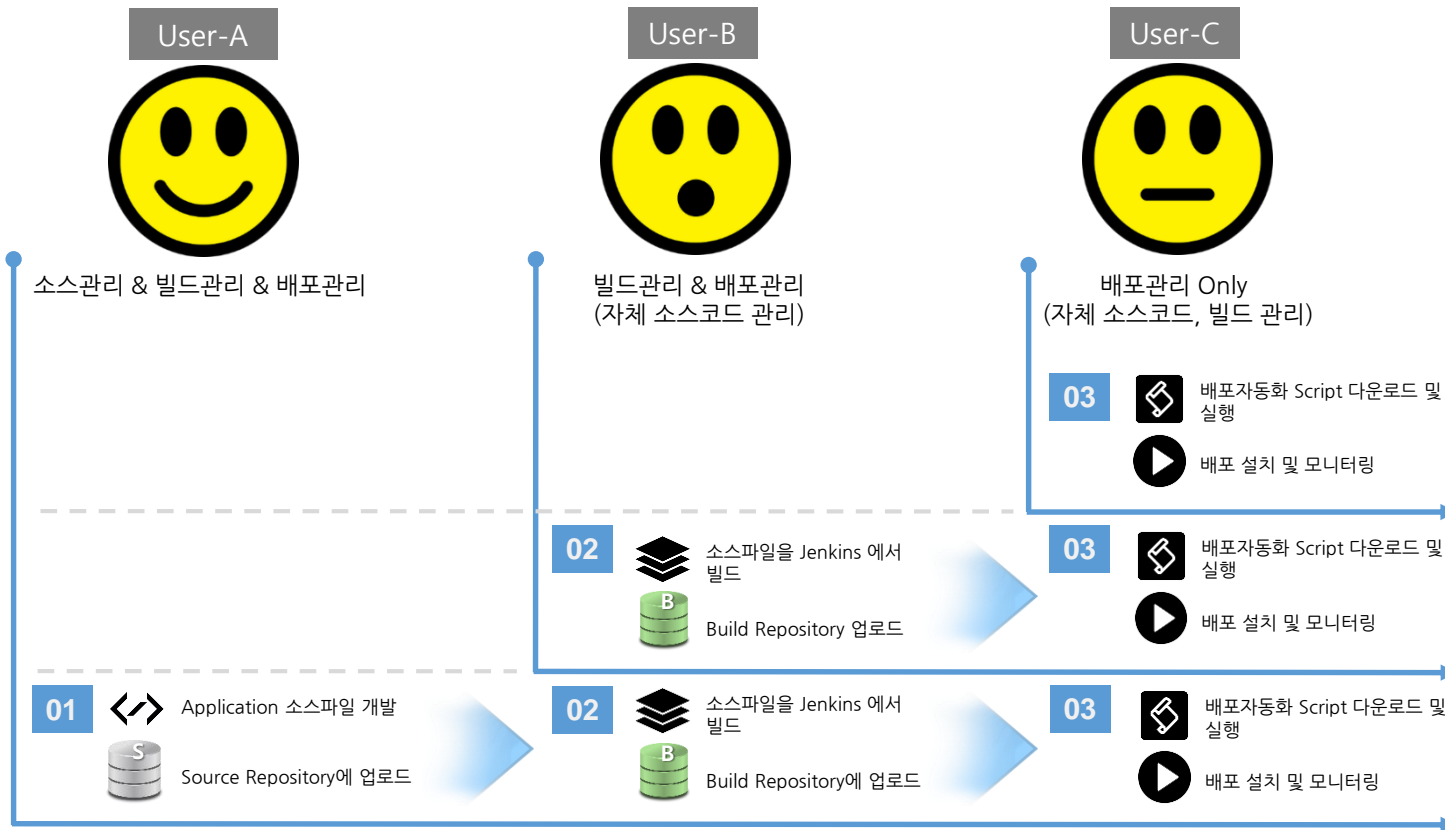
8. DCOSTACK Suite 구성

제품명	용도	기능 상세
Foundation Suite	기본 Private Cloud 구성 및 관리	Cloud Computing을 위한 Flavor, Volume 등 자원할당 구성 및 관리 포털 제공
Hybrid Suite	Public Cloud 연동 및 Private/Public 통합 관리	Public Cloud 연동 및 DCOSTACK Private Cloud 통합관리, Hybrid Live Migration 지원
Cloud Manager Suite	클라우드 인프라 구축 및 모니터링, 로그, 미터링, 자산 통합운영 관리	Cloud Compute, Virtual Network 구성 및 모니터링, 이벤트, 로그, 원격관리, 자산관리 지원
DevOps Suite	개발자를 위한 인프라 및 서비스 구성 관리 자동화	개발자를 위한 클라우드 인프라 할당 및 서비스 설정 자동화 지원
Automation Suite	플랜 및 스케줄에 의한 서비스 자동배포 적용 등 데이터센터 운영 자동화	운영자가 서비스 플랜을 사전에 정의하고, DCOSTACK을 통해 운영서비스에 대한 자동 배포 및 적용과 함께 스케줄링 지원
Cloud Network Suite	소프트웨어 기반의 데이터센터 SDN 구성	Virtual Network, VNF(로드밸런서,파이어월), Baremetal 스위치를 통한 SDN 구성까지 전체적인 데이터센터 클라우드 네트워크 구성
Cloud Storage Suite	CEPH 기반의 Object 및 Volume SDS 구성	별도의 소프트웨어 기반 클라우드 분산 스토리지 구성(CEPH) : File, Block, Object Storage
Custom Suite	고객 맞춤형 패키지	요구 기능에 대응하여 고객사 환경에 최적화된 구성 지원

II. DCOSTACK 활용방안

1. DCOSTACK DevOps Suite - 소스 빌드/배포 자동화

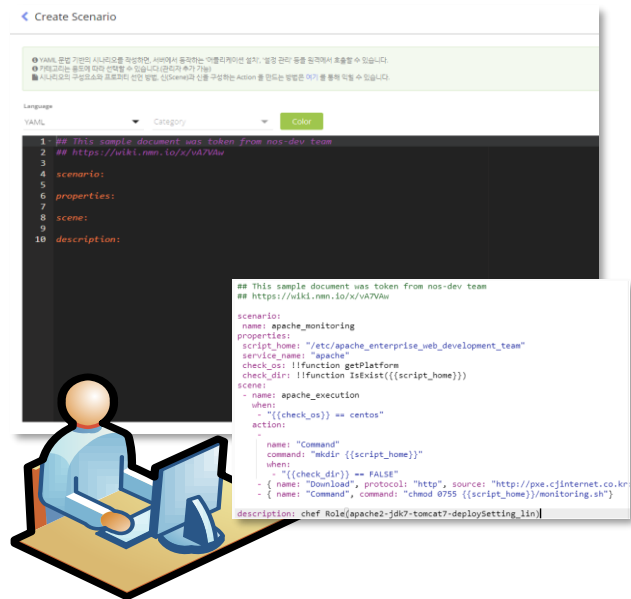
- ✓ 개발소스, 빌드, 배포에 대한 사용자들의 다양한 요구사항 반영필요
- ✓ 다양한 사용자 유형을 고려한 저장소 환경 및 배포 자동화 기능 구현
- ✓ 소스관리(Source) → 빌드관리(Build) → 배포관리(Deploy) 필요



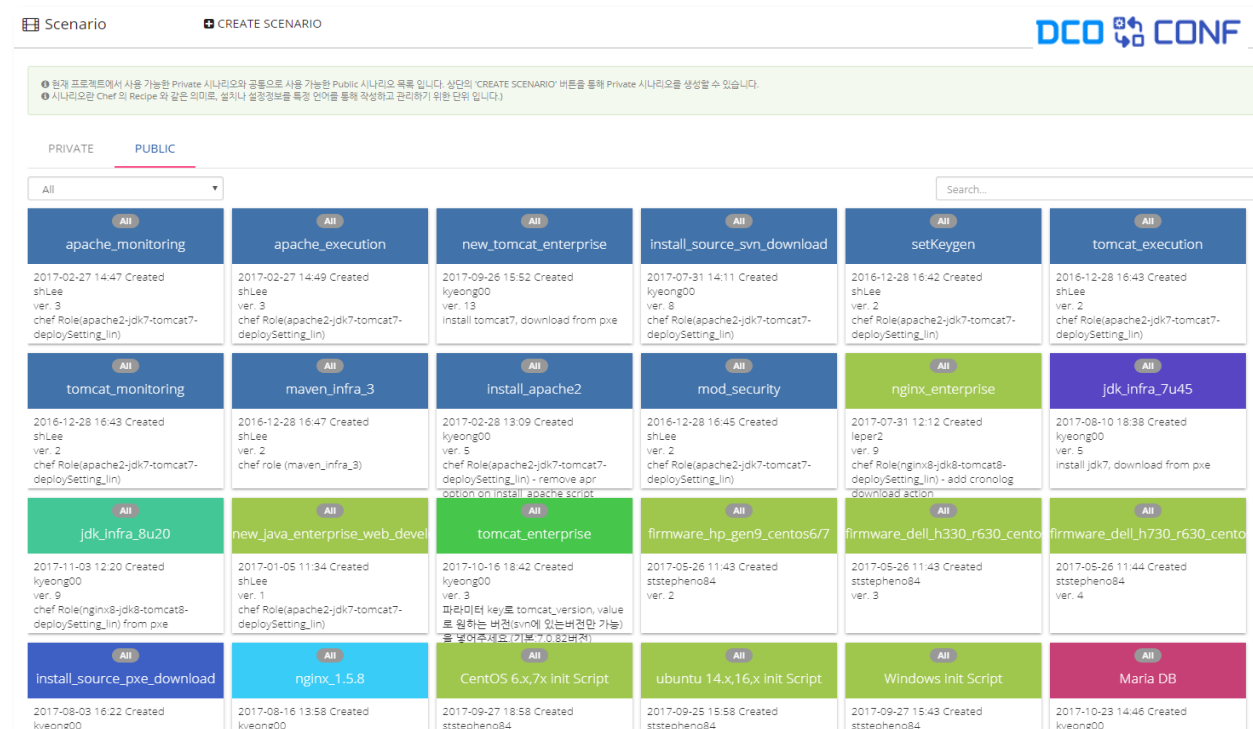
II. DCOSTACK 활용방안

2. DCOSTACK DevOps Suite - 어플리케이션/Patch 자동화

- ✓ DB, Web, OS, WAS, Firmware, Patch 및 어플리케이션의 서비스 자동화 반영 필요
- ✓ 다양한 고객 요구를 대응하는 시나리오(템플릿) 생성을 위한 개발도구 제공 필요
- ✓ 템플릿 재사용을 위한 카탈로그 서비스 필요



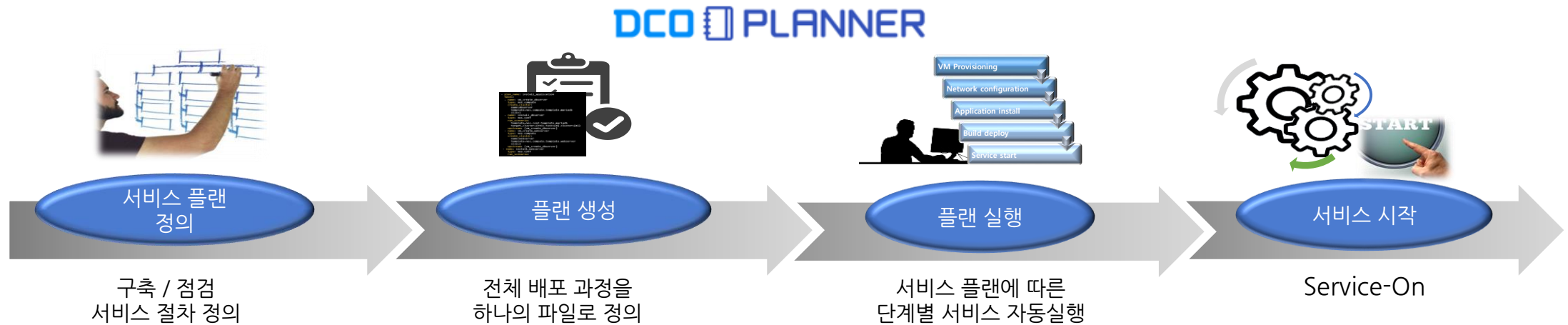
[어플리케이션 자동화 템플릿 개발도구 제공]



[어플리케이션 자동화 템플릿 : 서비스 카탈로그 형태로 제공]

3. DCOSTACK Automation Suite - 서비스 자동화

- ✓ 서비스 단위의 자동화 서비스 (Plan + Schedule) 지원
- ✓ 서비스 별 라이프사이클을 고려한 배포 플랜 및 스케줄 설정에 따른 자동화 실행
- ✓ 스케줄링 및 실행 순서 설정이 가능하며, 워크플로우 순서대로 서비스 자동 실행 지원

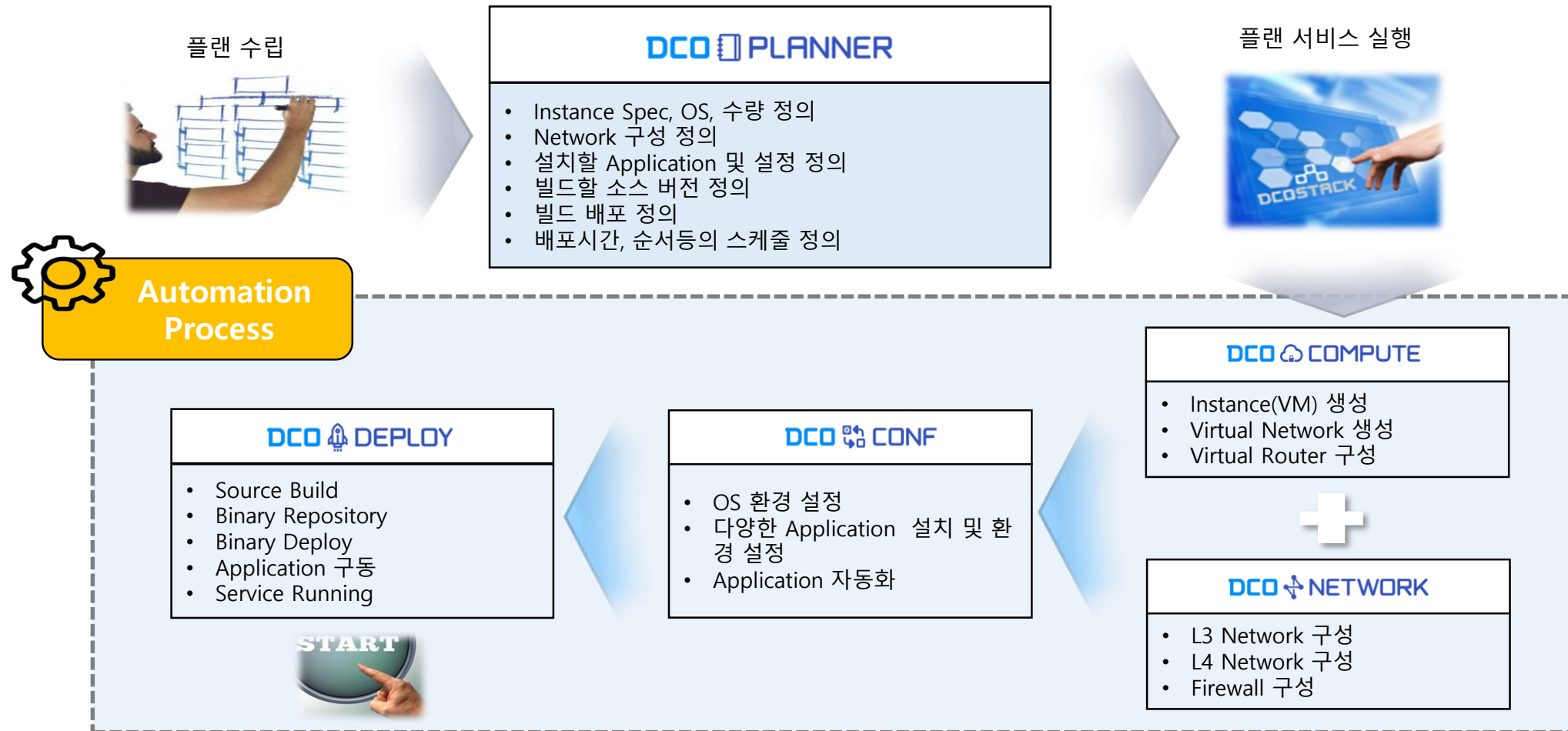


[Workflow 예시]



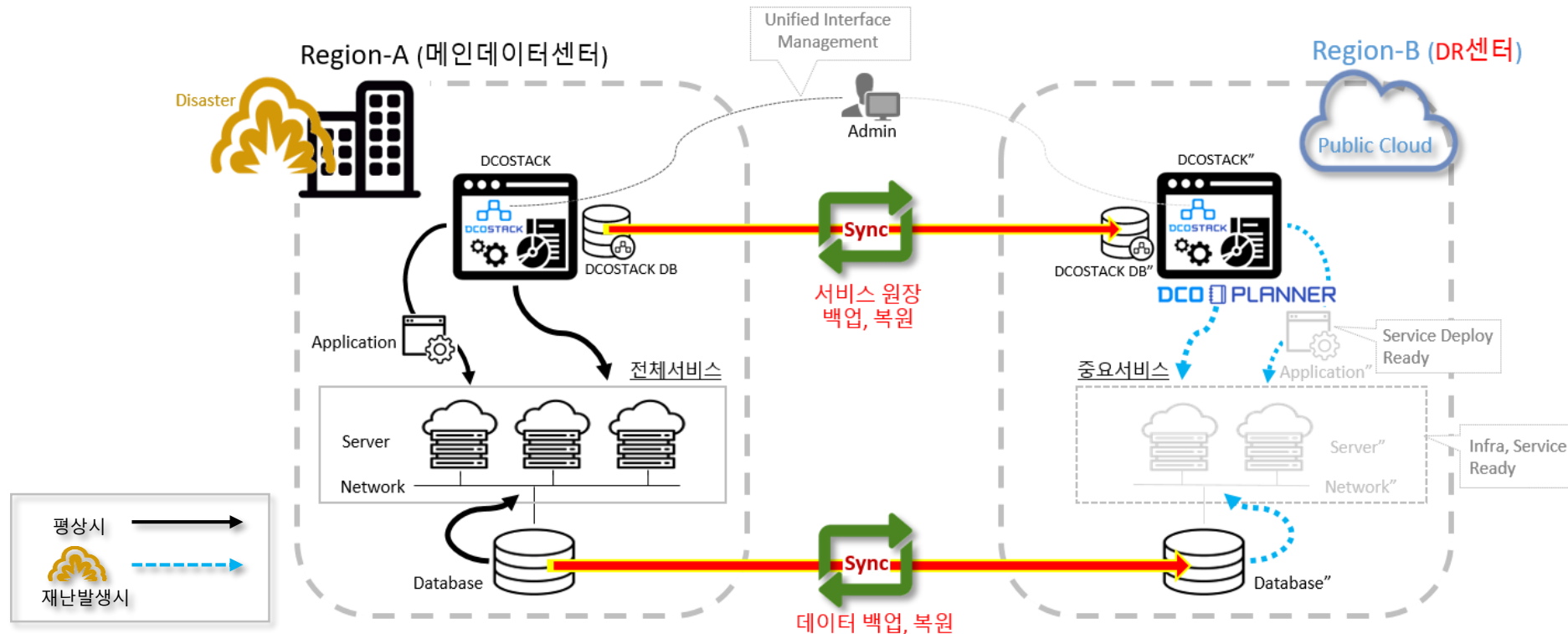
4. DCOSTACK Automation Suite - 데이터센터 자동화 프로세스 예시

- ✓ 실제 워크플로우를 기반으로 하는 서비스별 배포 플랜 정의 및 생성 지원
- ✓ 배포순서, 스케줄 등을 적용한 플랜 서비스 실행을 통해 인프라 및 애플리케이션 자동설치 환경 구현



5. DCOSTACK DR 구성방안 #1 - Cloud DR - Distributed DCOSTACK

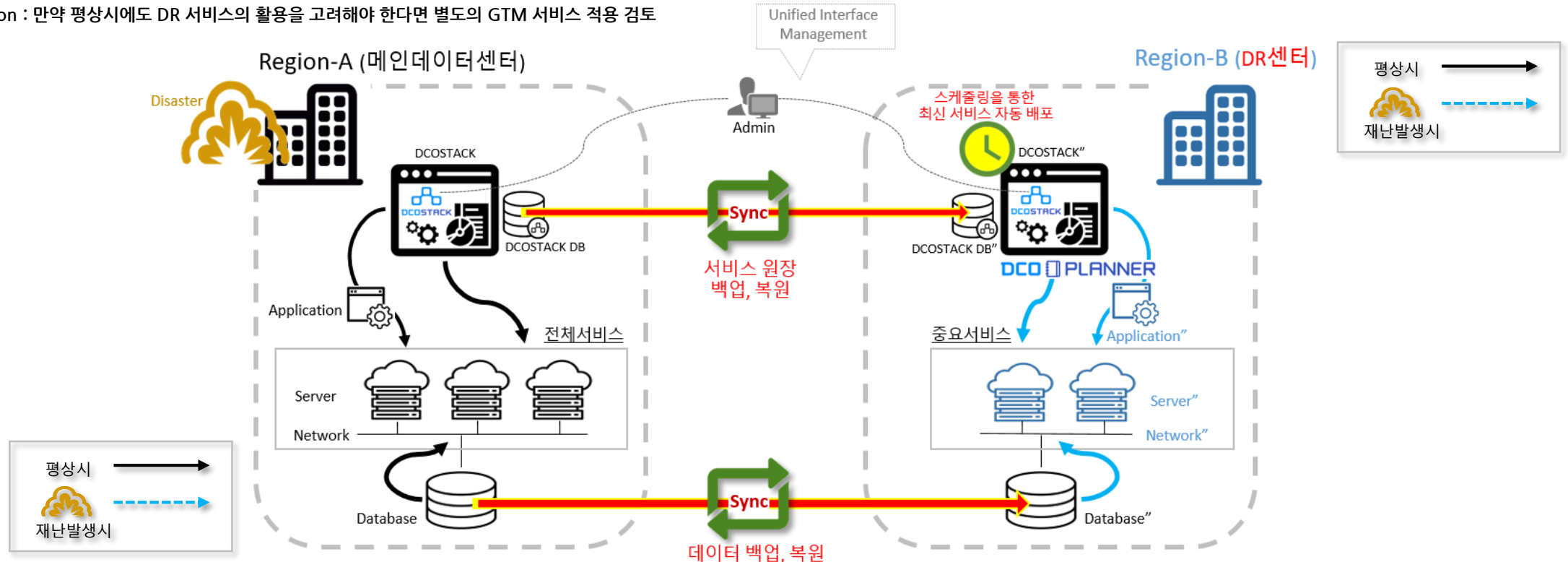
- ✓ 구성조건 : DCOSTACK을 메인데이터센터/DR(Cloud)에 각각 구축하고 서비스DB 동기화
- ✓ 고려사항 : DCOSTACK 및 DNS 위치는 반드시 재난발생시 영향이 없는 외부에 적용해야 함
- ✓ 장점 : 물리DR 보다 저렴한 비용 (평상시 DR 인프라 미사용 : 재난발생시 Cloud에 인프라/서비스 생성)
- ✓ 단점 : 물리DR 구성 대비 느린 복구시간



6. DCOSTACK DR 구성방안 #2- Physical DR - Distributed DCOSTACK

- ✓ 구성조건 : 물리DR에 DCOSTACK 구성 및 메인데이터센터와 서비스DB 동기화를 통한 자동 서비스 구성
- ✓ 고려사항 : 외부 DNS 적용
- ✓ 장점 : 최신 구성 정보 동기화 및 스케줄링된 서비스 배포 적용을 통해 자동화된 DR 구성 가능
- ✓ 단점 : 물리DR의 장점을 수용하고, 서비스 자동화 기능 적용을 통해 관리자의 DR 센터 운영업무는 최소화

※ Option : 만약 평상시에도 DR 서비스의 활용을 고려해야 한다면 별도의 GTM 서비스 적용 검토



III. DCOSTACK 활용 사례

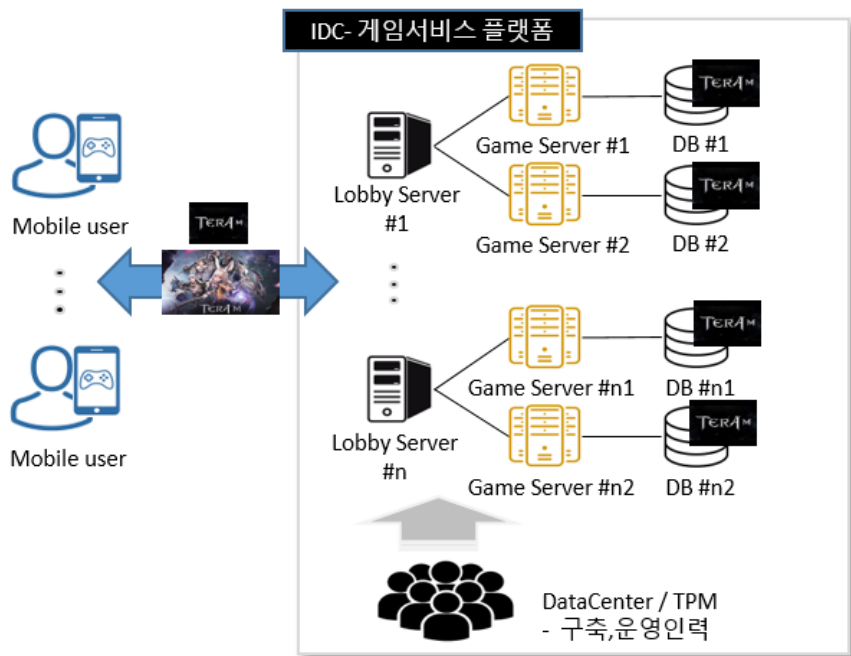
1. N사 도입 사례 - 서비스 요구 사양 : TERA-M

- ✓ 고품질 콘텐츠 기반의 게임 서비스를 위한 고성능 서버 자원
- ✓ 대용량 트래픽 처리를 위한 고성능 네트워크 처리
- ✓ 오픈 시점에 예상 유저 접속량 초과시 서비스 지연이 없도록 대비한 신속한 시스템 증설
- ✓ 서비스 안정화 시 불필요한 자원 효율성을 최적화하기 위해 유연한 시스템 회수
- ✓ 서비스 운영 및 유지관리가 용이하도록 자동화가 가능한 서비스 프로세스 제공



2. N사 도입 사례 - 서비스 구성 내역 : TERA-M

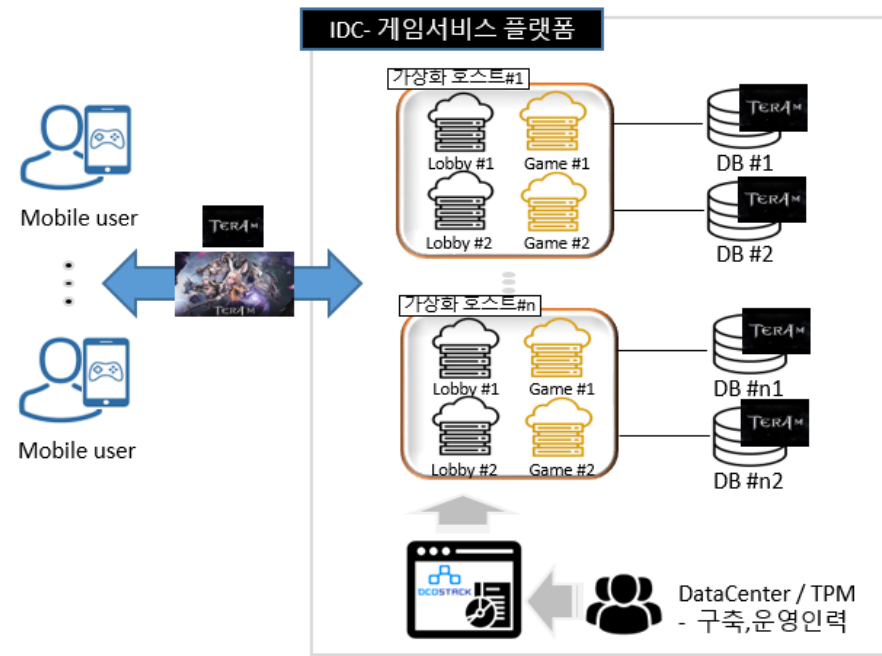
• 서비스 성능 요구사항



약 400여대의 물리머신이 필요한 서비스 사이즈

- 긴 준비시간 / 높은 인력 투입률
- 신속한 서비스 확장 및 축소 어려움

• 서비스 인프라 구성 및 적용



약 100여대의 가상화 호스트 머신으로 서비스 구성

- DCOSTACK을 이용한 가상화로 시스템 수량 절감
- 고성능 가상 서버 및 가상 네트워크 제공
- 자동화 서비스 적용을 통한 업무 효율 증가

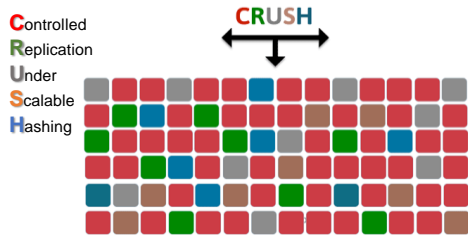
3. N사 도입 사례 - DCOSTORAGE(SDS)

- ✓ 서비스 용도 및 성능 요구사항에 맞춘 최적화된 하드웨어 및 아키텍처 구성
 - 대규모 Block Storage 및 Object Storage 혼합 운영

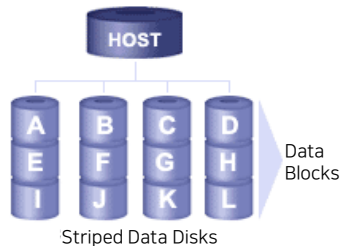
환경	Size	용도	비고
Farm #1	500TB +	Block Storage	OpenStack 연동
Farm #2	500TB +	Block Storage	OpenStack 연동
Farm #3	1.0PB +	Object Storage	Archiving Storage 용도

- Ceph 적용기술

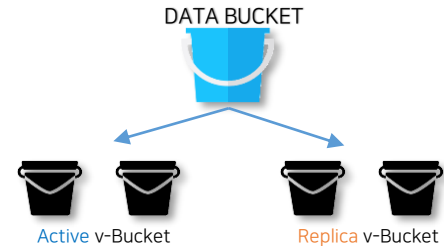
- CRUSH Rule Customizing
: 물리적 위치 및 디스크 종류에 맞추어 운영



- Data Striping
: 성능테스트 결과를 통해 적절한 Stripe Width 를 도출 및 적용



- Dynamic Bucket Index Sharding
: 단일 인덱스의 경우 오브젝트 증가에 따른 성능저하를 회피



감사합니다
THANK YOU