

명세서

발명의 명칭: 클라우드 플랫폼에서 어플리케이션 컨테이너의 볼륨(스토리지) 프로비저닝 방법

기술분야

- [1] 본 발명은 클라우드 플랫폼에서 어플리케이션 컨테이너의 볼륨(스토리지) 프로비저닝 방법에 관한 것으로서, 보다 상세하게는 어플리케이션 컨테이너에 필요한 저장 공간을 할당할 수 있는 클라우드 플랫폼에서 어플리케이션 컨테이너의 볼륨(스토리지) 프로비저닝 방법에 관한 것이다.

[2]

배경기술

- [3] 클라우드(Cloud)는 컴퓨팅 서비스 사업자 서버를 구름 모양으로 표시하는 관행에 따라 '서비스 사업자의 서버'로 통한다. 소프트웨어와 데이터를 인터넷과 연결된 중앙 컴퓨터에 저장하고 인터넷에 접속하기만 하면 언제 어디서든 데이터를 이용할 수 있도록 하는 것이다.
- [4] 이러한 클라우드는 서비스 제공 형태에 따라 Salesforce.com·Google e-mail 등과 같이 다수의 사용자에게 온디맨드(On-demand)로 제공되는 애플리케이션 서비스인 Software as a Service(SaaS), AWS RDS·Google AppEngine 등과 같이 개발용 플랫폼 또는 어플리케이션 실행에 필요한 소프트웨어 스택인 Platform as a Service(PaaS), AWS EC2 등과 같이 서버 또는 스토리지 등을 사용자에게 서비스 형태로 제공하는 Infrastructure as a Service(IaaS) 등으로 나눌 수 있다.
- [5] 또한 클라우드는 도입과 배포 형태에 따라 오직 하나의 단체를 위해서만 운영되는 프라이빗 클라우드(Private cloud), 공개적 이용을 위해 열린 네트워크를 통해 렌더링되는 퍼블릭 클라우드(Public cloud), 뚜렷한 실체는 유지하지만 함께 묶여 있는 둘 이상의 클라우드의 조합인 하이브리드 클라우드(Hybrid cloud) 등으로도 나눌 수 있다.
- [6] 한편 엔터프라이즈 클라우드(Enterprise Cloud) 경우 기업의 비즈니스와 IT 전략을 구현한 클라우드로 어플리케이션 서비스를 중심으로 기술과 인프라를 맞춤화하고 최적화하는 것이 무엇보다 중요하며, 또한 어플리케이션을 다양한 인프라에 구성하거나 배포하기에 용이하여야 한다.
- [7] 이에 본 출원인은 어플리케이션을 컨테이너화하는 방법을 출원하여 특허등록한 바 있다(특허출원 제10-2017-0056483호).
- [8] 그런데 상기 출원에서 어플리케이션 컨테이너에 있는 저장 공간은 임시적이고 재시작 시 저장 공간 내 파일이 삭제될 수 있다.

[9]

발명의 상세한 설명

기술적 과제

- [10] 이에 본 발명은 이러한 상기 문제점을 해결하기 위해 창출된 것으로, 어플리케이션 컨테이너에 필요한 저장 공간을 할당할 수 있는 클라우드 플랫폼에서 어플리케이션 컨테이너의 볼륨(스토리지) 프로비저닝 방법을 제공하는 데 그 목적이 있다.
- [11] 그러나 본 발명의 기술적 과제들은 위에서 언급한 과제들로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

[12]

과제 해결 수단

- [13] 본 발명의 실시예에 따른 클라우드 플랫폼에서 어플리케이션 컨테이너의 볼륨(스토리지) 프로비저닝 방법은 어플리케이션 컨테이너의 볼륨 할당이 요청되면, 상기 클라우드 플랫폼 시스템이 정적 볼륨 생성인지, 동적 볼륨 생성인지를 판단하는 단계; 상기 정적 볼륨 생성이면, 상기 클라우드 플랫폼 시스템이 정적 볼륨 등록 리스트를 조회하고, 등록된 정적 볼륨이 선택되고 할당 용량이 입력되면, 상기 클라우드 플랫폼 시스템이 가용 공간을 확인하며, 상기 가용 공간이 부족하면 상기 클라우드 플랫폼 시스템이 저장 공간이 부족함을 알리고, 다시 등록된 정적 볼륨이 선택되고 할당 용량이 입력되기를 대기하고, 상기 가용 공간이 부족하지 아니하면 상기 클라우드 플랫폼 시스템이 정적 볼륨 용량을 할당하고 정보를 생성하여 볼륨 할당을 완료하는 단계; 및 상기 동적 볼륨 생성이고 할당 용량이 입력되면, 상기 클라우드 플랫폼 시스템이 가용 공간을 확인하며, 상기 가용 공간이 부족하면, 상기 클라우드 플랫폼 시스템이 저장 공간이 부족함을 알리고, 다시 할당 용량이 입력되기를 대기하고, 상기 가용 공간이 부족하지 아니하면 상기 클라우드 플랫폼 시스템이 저장 경로를 생성하고 용량 할당 및 정보를 생성하여 볼륨 할당을 완료하는 단계를 포함한다.

[14]

발명의 효과

- [15] 본 발명에 의한 클라우드 플랫폼에서 어플리케이션 컨테이너의 볼륨(스토리지) 프로비저닝 방법은 어플리케이션 컨테이너에 필요한 저장 공간을 할당할 수 있는 효과를 갖는다.

[16]

도면의 간단한 설명

- [17] 도 1은 본 발명의 일 실시예에 따른 클라우드 플랫폼 시스템의 구성도를 나타낸다.
- [18] 도 2는 도 1의 클라우드 통합부의 기능을 간략히 도시한 것이다.
- [19] 도 3은 도 1의 서비스 관리부의 기능을 간략히 도시한 것이다.
- [20] 도 4는 도 1의 어플리케이션 오케스트레이션부의 기능을 간략히 도시한 것이다.

- [21] 도 5는 본 발명의 일 실시례에 따른 어플리케이션 컨테이너화의 프레임워크를 나타낸다.
- [22] 도 6 내지 도 11은 도 1의 개발/운영부의 기능을 간략히 도시한 것이다.
- [23] 도 12는 본 발명의 일 실시례에 따른 클라우드 플랫폼 시스템의 아키텍처를 나타낸다.
- [24] 도 13은 칵테일 서버의 구성과 그 주변 아키텍처를 나타낸다.
- [25] 도 14는 본 발명의 일 실시례에 따른 컨테이너 볼륨(스토리지) 개념도이다.
- [26] 도 15는 본 발명의 일 실시례에 따른 볼륨 정보 등록을 설명하기 위한 도면이다.
- [27] 도 16은 본 발명의 일 실시례에 따른 클라우드 플랫폼 시스템의 컨테이너 볼륨(스토리지) 프로비저닝 방법을 나타낸 흐름도이다.
- [28] 도 17은 본 발명의 일 실시례에 따른 볼륨 정보 등록 화면의 예시도이다.
- [29] 도 18은 본 발명의 일 실시례에 따른 등록 볼륨 정보 현황 화면의 예시도이다.
- [30] 도 19 및 도 20은 본 발명의 일 실시례에 따른 볼륨을 할당하기 위한 화면의 예시도이다.

[31]

발명의 실시를 위한 형태

- [32] 본 발명의 장점 및 특징 그리고 그것들을 달성하는 방법들은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시례들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시례들에 한정되는 것이 아니라 또 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시례들은 본 발명의 개시가 완전하도록 하고 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 단지 청구항에 의해 정의될 뿐이다.
- [33] 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성요소를 지칭한다.
- [34] 이하 첨부된 도면들을 참고하여 본 발명의 실시례에 따른 클라우드 플랫폼 시스템에 대해 설명하도록 한다.
- [35] 도 1은 본 발명의 일 실시례에 따른 클라우드 플랫폼 시스템의 구성도를 나타내고, 도 2는 도 1의 클라우드 통합부의 기능을 간략히 도시한 것이며, 도 3은 도 1의 서비스 관리부의 기능을 간략히 도시한 것이고, 도 4는 도 1의 어플리케이션 오케스트레이션부의 기능을 간략히 도시한 것이다.
- [36] 도 5는 본 발명의 일 실시례에 따른 어플리케이션 컨테이너화의 프레임워크를 나타내며, 도 6 내지 도 11은 도 1의 개발/운영부의 기능을 간략히 도시한 것이다.
- [37] 도 1의 클라우드 플랫폼 시스템은 멀티/하이브리드 클라우드 통합 관리를 기반으로 어플리케이션 가용성·확장성을 보장하고 개발·운영의 효율화를 위한 뷰와 도구를 제공한다. 이하 본 발명의 클라우드 플랫폼 시스템을 "칵테일 클라우드(Cocktail Cloud)"라고 칭하기로 한다.
- [38] 도 1을 참조하면, 칵테일 클라우드는 클라우드 통합부(Cloud Integration,

- 100)·서비스 관리부(Service Management, 110)·어플리케이션 오케스트레이션부(Orchestration, 120)·개발/운영부(DevOps View, 140) 및 DB/저장소(150)를 포함한다.
- [39] 클라우드 통합부(Cloud Integration, 100)는 멀티/하이브리드 클라우드의 인프라를 자동 구성하여 어플리케이션에 제공하고 관리를 위한 구성 정보를 동기화하는 역할을 수행한다.
- [40] 클라우드 통합부(100)는 클라우드 프로비저닝(Cloud Provisioning)과 클라우드 동기화(Cloud Synchronization)의 기능을 수행한다.
- [41] 도 2를 참조하면 클라우드 프로비저닝 기능은 어플리케이션 클러스터(각테일 클러스터)에 클라우드 네트워크 인프라를 구성 및 제공하고, 어플리케이션에 클라우드의 컴퓨팅 인프라를 구성 및 제공하는 기능이다. 그리고 물리 인프라(Bare Metal)의 경우 클러스터 설정 도구를 제공한다. 지원 클라우드는 Public의 경우 AWS·Azure·Aliyun·Google Computing Engine이고, Private의 경우 Openstack·VMWear이며, 이외에 On-premise·Datacenter BareMetal Infra가 있을 수 있다.
- [42] 클라우드 동기화 기능은 클라우드 인프라 구성 정보를 통합 구성 DB(160)에 저장 및 관리하고, 운영 시 인프라 변경 정보를 통합 구성 DB(160)와 동기화하는 기능이다.
- [43] 서비스 관리부(Service Management, 110)는 어플리케이션 클러스터를 관리하는 논리적 그룹으로 클라우드 계정과 사용자, 네트워크 자원을 할당 및 관리하는 역할을 수행한다. 즉 서비스 관리부(110)는 통합 계정 관리 기능·네트워크 관리 기능 및 사용자 관리 기능을 수행한다.
- [44] 도 3을 참조하면 통합 계정 관리(Cloud Provider) 기능은 멀티 클라우드 계정 및 접속 정보를 통합 관리하고, 네트워크와 클라우드 프로비저닝 구성에 사용되는 기능이다.
- [45] 네트워크 관리 기능은 클라우드 네트워크를 구성하고 서비스에 할당하는 기능이다. 예를 들면 AWS의 VPC·Subnet일 수 있다. 하나의 서비스는 멀티 클라우드의 공급자의 네트워크를 사용하여 클러스터를 생성하여 어플리케이션을 구성·운영한다.
- [46] 사용자 관리 기능은 서비스를 관리하는 팀 구성원과 개발/운영에 필요한 권한을 관리하는 기능이다. 여기서 권한은 전사 서비스 관리 권한(Admin), 전사 서비스 조회 권한(Manager), 구성원으로 배정된 서비스 관리 권한(DevOps) 등을 포함할 수 있다. 사용자는 여러 서비스에 구성원으로 참여 가능하다.
- [47] 어플리케이션 오케스트레이션부(Orchestration, 120)는 어플리케이션의 배포와 가용성·확장성을 보장하는 기능으로 각테일 클러스터(Cluster)의 핵심 기능을 담당한다.
- [48] 어플리케이션 오케스트레이션부(120)는 어플리케이션 배포(Deployment) 기능·복제(Replication Control) 기능·롤링 업데이트(Rolling Update)

- 기능·스케일링(Scaling) 기능 및 모니터링(Monitoring) 기능을 수행한다.
- [49] 도 4를 참조하면 어플리케이션 배포 기능은 컨테이너 이미지 기반의 배포로 별도 설정과 구성 작업이 필요 없는 용이성을 제공하며, 어플리케이션 배포 시 클라우드 인프라를 자동 프로비저닝하는 기능이다.
- [50] 여기서 어플리케이션은 컨테이너화되어 배포되게 되는데, 어플리케이션 컨테이너(이하 "컨테이너"라고 한다)는 어플리케이션 프로세스에 호스트 자원을 할당하고 격리하여 가상화한 OS상의 독립시스템을 말한다.
- [51] 컨테이너에 사용되는 핵심 기술은 Linux의 cgroup(control group)과 namespace이다. cgroup은 OS상의 프로세스에 호스트 자원을 할당하기 위해 해당 프로세스 그룹을 만들고 자원의 할당 및 관리를 수행한다. namespace는 프로세스, 네트워크, 마운트(mount) 등을 특정 name space로 격리하는 기술이다. 이에 따라 컨테이너는 cgroup을 통해 어플리케이션 프로세스에 자원을 할당하고, namespace로 격리한 OS상에 가상화된 독립 시스템을 말한다.
- [52] 컨테이너는 하이퍼바이저(Hardware emulator)와 게스트 OS를 사용하지 않는 가벼운 OS 가상화 방식으로 호스트 자원의 소모량이 거의 없고 기동에 드는 시간이 매우 적어 어플리케이션 가상화에 적합한 기술이다. 또한 OS상의 가상화로 기존 물리 서버(Bare Metal)·가상 서버(Virtual Machine) 등 인프라에 독립적인 구성과 배포가 가능하다.
- [53] 이렇게 기존 또는 신규 어플리케이션 구성을 컨테이너로 전환하기 위해서는 컨테이너화(Containerization) 과정이 수반되어야 한다. 그리고 이에 따른 개발·테스트·운영 방식의 전환 및 운영 인프라 구성(각테일 클라우드 플랫폼) 최적화 작업을 병행해야 한다.
- [54] 기존 어플리케이션을 컨테이너로 전환하기 위해서는 어플리케이션의 설정 및 소스가 아닌 구성의 전환이 필요하며, 배포와 운영 효율을 고려할 때 워크로드(Workload) 중심의 역할별 독립적 구성이 일반적이고, 복제를 통한 다중화와 스케일링을 고려한 구성이 설계되고 적용되어야 할 것이다.
- [55] 어플리케이션 개발·테스트·운영 방식의 전환을 위해서는 이미지 기반의 어플리케이션 빌드·테스트·배포와 베이스 이미지를 통한 어플리케이션 구성이 표준화되어야 할 것이다.
- [56] 어플리케이션 컨테이너 운영 인프라 구성 최적화를 위해서는 컨테이너 오케스트레이션을 위한 클러스터 중심의 인프라가 구성되고, 복제·스케일링을 고려한 컴퓨팅 용량이 산정(예비 용량 최소화, 필요 시 확장 용이)되어야 하며, 공유 스토리지·보안·네트워크 등 관련 인프라를 구성하여야 할 것이다.
- [57] 도 5를 참조하면 컨테이너화는 크게 분석 및 구성 설계(S100)·컨테이너 전환(S200)·운영 이관(S300)으로 구분된다.
- [58] 분석 및 구성 설계(S100)를 위해 컨테이너/클라우드 도입 목적과 전략을 고려하여 기존 어플리케이션 중 컨테이너 전환 대상을 선정한다(S110).
- [59] 대상 어플리케이션이 선정되면 대상 어플리케이션을 분석한다(S120). 이때

어플리케이션·인프라·데이터·연계 구조 등의 어플리케이션 현황 및 자료 조사를 하고, 개발 및 운영·관리자의 요구를 수집한다. 그리고 컨테이너 구성 방향·이슈 및 해결 방안을 도출한다.

- [60] 그리고 분리/통합·연계·가용성·확장성·보안 등을 고려하여 대상 어플리케이션별 컨테이너 구성을 설계한다(S130). 이때 베이스 이미지·환경 변수·포함 항목·코멘드 등의 이미지 빌드 템플릿을 정의할 수 있다.
- [61] 그후 인프라 구성을 설계한다(S140). 전환 인프라(클라우드/베어 메탈) 공급자를 선정하고, 어플리케이션 컨테이너별 용량을 산정한다. 그리고 컨테이너 클러스터 노드 수 및 인프라 용량을 산정하며, 스토리지·네트워크·보안 구성을 설계한다.
- [62] 인프라 구성이 설계되면 컨테이너 전환 방안을 수립한다(S150). 이때 어플리케이션별 전환 세부 방안을 수립하고, 전환 업무 및 조직/역할을 정의하며, 전환 일정을 수립한다. 그리고 보고 및 피드백을 반영한다.
- [63] 컨테이너 전환(S200)을 위해서는 반복/점증적 전환(S210)이 필요하다. 사전 테스트(PoC), 어플리케이션별 단계적 전환 등 반복적이고 점증적으로 전환한다.
- [64] 그리고 각테일 클러스터를 구성(S220)하기 위해 각테일 클라우드 플랫폼을 설치 및 구성하고, 네트워크·공유 스토리지·보안 등 기반 인프라를 구성한다(클라우드의 경우 각테일에서 프로비저닝). 기반 인프라 할당 및 사용자 등록을 통해 각테일 서비스와 클러스터를 생성하고, 클러스터 구성을 검증한다.
- [65] 그리고 어플리케이션 전환(S230)을 위해 어플리케이션 컨테이너를 구성하고, 필요 시 어플리케이션 설정 및 소스를 변경한다. 전환 컨테이너의 기능 및 설정 등을 검증하며, 컨테이너 배포 이미지 빌드 및 레지스트리에 등록한다. 그리고 각테일 서버를 생성하고 테스트 한다.
- [66] 데이터 전환(S240)을 위해 대상 어플리케이션 컨테이너 전환하며, Persistence 볼륨 설정 등을 통해 각테일 서버를 설정하고, 데이터를 추출하고 각테일 서버에 전송한다. 이 기중 DB 솔루션 적용의 경우 데이터 변환을 수행하며, 데이터 적합성을 확인한다. 운영 어플리케이션의 경우 다운타임을 최소화하기 위해 데이터 동기화 솔루션을 적용한다.
- [67] 그후 검증된 컨테이너를 각테일 서버에 배포하고, 어플리케이션 기능 및 성능 테스트를 수행하며, 컨테이너 및 인프라에 테스트 결과를 반영한다(S250, S260).
- [68] 운영 이관(S300)을 위해 운영 배포/오픈(S310)이 수행되는데 구체적으로 운영 각테일 클러스터를 생성하고 전환 완료된 이미지를 기반으로 각테일 서버를 생성하고 연계 구성한다. 그리고 운영 데이터를 이관하고 어플리케이션을 오픈한다. 이러한 어플리케이션 컨테이너를 배포·운영·관리하는 기술을 컨테이너 오케스트레이션(Orchestration)이라 칭한다.
- [69] 컨테이너 오케스트레이션은 물리/가상 인프라에 관리 클러스터(Managed Cluster)를 구성하여 어플리케이션 컨테이너를 배포·운영·관리하는 기술로, 컨테이너의 가볍고 빠른 기동성과 이동성의 장점을 활용하여 기존 사내, 데이터

센터 인프라의 클라우드화와 프라이빗/퍼블릭 클라우드의 어플리케이션 관리 플랫폼으로 확산되고 있다.

- [70] 각테일 클라우드 모니터링 뷰를 통해 어플리케이션 및 인프라 운영 모니터링을 수행하고 성능 이슈 및 오류를 반영한다(S320).
- [71] 개발·운영 체계 이관 및 적용(S330)을 위해 컨테이너 이관 결과를 리포트하고, 담당 개발 및 운영 조직에 컨테이너 기반 개발/운영 체계 교육을 실시하며, 각테일 클라우드 플랫폼 사용 교육을 실시한다.
- [72] 이에 따라 컨테이너는 다음과 같은 장점을 갖는다.
- [73] 첫째, 컨테이너는 독립성을 갖는다.
- [74] 격리된 어플리케이션 실행 환경이며, 독립적인 자원이 할당되고(CPU, Memory, Disk, Network 등), 동일 호스트상 다중 어플리케이션이 운영된다.
- [75] 둘째, 컨테이너는 가벼운 가상화를 구현한다.
- [76] OS 수준의 가상화(Non Hypervisor)가 가능하며, 빠른 조작성 가능하고(생성·실행·재시작 등), 적은 크기의 컨테이너 이미지로 배포 및 업데이트가 효율적이다.
- [77] 셋째, 컨테이너는 이동성을 갖는다.
- [78] 인프라 독립적 이미지를 가지며, 베어메탈(Bare Metal)·가상 머신(Virtual Machine)·클라우드(Cloud) 등 어디든지 이동이 가능하고, 이미지 레지스트리를 통한 온라인 배포 및 버전 관리가 가능하고, 주요 호스트 OS(Linux 계열, Windows)를 지원한다. 이러한 컨테이너의 이동성은 멀티/하이브리드 클라우드 환경하에 어플리케이션 운영/개발의 생산성 및 효율을 높이며 특히 규격화된 컨테이너 이미지로 이중의 인프라에 어플리케이션 배포 및 이전의 어려움을 해결하고 특정 클라우드에 종속되는 락인(Lock-in) 문제를 해결해 준다.
- [79] 복제 기능은 어플리케이션 안정성과 가용성을 위해 초기 지정한 복제수(다중화)를 유지하고, 어플리케이션 컨테이너 헬스 체크(Health Check)를 통해 이상 시 재기동하는 방식으로 OS 재부팅 방식보다 빠르고 효율적이다. 복제된 어플리케이션은 로드밸런싱을 통해 서비스된다.
- [80] 롤링 업데이트 기능은 어플리케이션 서비스의 중단 없이 배포·인프라 변경 등의 업데이트 작업을 수행하고, 여러 어플리케이션 간 의존성이 있을 경우 DevOps View의 작업(job) 관리 기능을 통해 자동화를 구성하는 기능이다.
- [81] 스케일링 기능은 어플리케이션의 모니터링을 통해 인스턴스의 스케일링을 인(In)/아웃(Out)하며, 어플리케이션 인프라의 경우 자원 용량의 스케일을 업(Up)/다운(Down)하는 기능이다. 그리고 모니터링 정보를 통해 스케일링 자동화를 구성한다.
- [82] 모니터링 기능은 어플리케이션 인스턴스(컨테이너+인프라)를 모니터링하고, 임계치 설정을 통한 알람을 발생 및 관리하는 기능이다.
- [83] 개발/운영부(DevOps View, 140)는 서비스 현황 기능, 클러스터 맵기능, 모니터링 뷰기능, 리소스 관리기능, 미터링기능, 작업 관리기능, 및 전사 현황

관리/분석기능을 포함한다. 각각의 기능을 도 6 내지 도 11을 참조하여 설명하면 다음과 같다.

- [84] 서비스 현황 기능은 각테일 클라우드의 전체 어플리케이션 클러스터의 현황을 서비스 중심으로 파악할 수 있는 뷰(도 6 참조)를 제공한다. 이에 서비스 현황·클러스터 현황·모니터링 알람 등의 항목이 표시될 수 있다.
- [85] 서비스 현황에서는 각테일 클라우드의 전체 서비스 현황을 조회할 수 있고, 서비스 내 클러스터의 구성 현황을 종합하여 클라우드 공급자·클러스터·서버·클라우드 컴포넌트·현재 월 사용 비용 등을 파악할 수 있다. 여기서 클러스터는 어플리케이션의 구성 단위를 의미하며, 서비스는 클러스터의 논리적 그룹을 의미한다.
- [86] 클러스터 현황에서는 클러스터의 공급자·리전·서버·클라우드 컴포넌트·월 사용 비용을 카드 형태로 조회 가능하고, 물리(Bare Metal) 클러스터의 경우 사용 비용은 제외될 수 있다.
- [87] 모니터링 알람 표시 기능에서는 클러스터 내 어플리케이션과 인프라에서 알람이 발생한 경우, 클러스터 카드에서 확인이 가능하다.
- [88] 클러스터 맵기능은 각테일 서버(어플리케이션)의 구성과 상태 정보를 맵 형태로 시각화하여 관리할 수 있는 뷰를 제공한다(도 7 참조).
- [89] 클러스터 맵은 클러스터의 서버와 클라우드 컴포넌트 구성을 맵 형태로 조회/관리하여 구성 정보의 가시성을 높인다. 클러스터 맵에서는 각테일 서버·클라우드 컴포넌트·서버 그룹 등의 항목을 포함할 수 있다.
- [90] 각테일 서버는 어플리케이션 오케스트레이션의 기본 단위로 로드 밸런싱·어플리케이션 컨테이너·인프라로 구성되며, 멀티/하이브리드 클라우드 관리에 표준화된 인터페이스를 제공한다. 각테일 서버는 서버 내 어플리케이션 상태와 복제, 자원 사용량을 확인하고 스케일링·롤링 업데이트 등을 관리 수행한다. 각테일 서버는 복제 기능의 유무에 따라 멀티와 싱글 인스턴스 타입으로 구분된다. AWS에서는 멀티존 옵션을 지원한다.
- [91] 클라우드 컴포넌트는 공급자가 제공하는 PaaS 서비스를 관리한다. 예를 들면 AWS의 DB 서비스인 RDS일 수 있다.
- [92] 서버 그룹은 서버 구성의 논리적 그룹을 관리적 편의성을 제공한다.
- [93] 모니터링 뷰기능은 클러스터 내 어플리케이션과 인프라의 자원 용량과 상태를 확인하고 클라우드 리소스의 상태를 확인할 수 있는 정보를 제공한다(도 8 참조).
- [94] 모니터링 뷰는 클러스터 내 어플리케이션과 인프라에 대한 모니터링 정보를 시각화하여 제공하고, CPU·메모리·디스크의 평균·TOP 정보 제공으로 자원의 사용량을 확인하고 운영에서 대응할 수 있도록 한다.
- [95] 모니터링 뷰는 뷰 전환(트렌드/테이터) 항목, 대상 전환(서버/리소스) 항목 등을 포함할 수 있다.
- [96] 뷰 전환 항목에서 트렌드뷰는 서버와 복제된 인스턴스·어플리케이션 컨테이너에 대한 시간별 모니터링 정보를 제공하고, 데이터뷰는 현재 시간의

평균·TOP 모니터링 수치를 제공한다.

- [97] 대상 전환 항목에서 모니터링 대상은 클러스터 내 서버와 클라우드 인프라의 리소스로 구분된다. 클라우드 리소스는 공급자가 제공하는 정보를 사용한다.
- [98] 리소스 관리 기능은 어플리케이션을 구성하는 클라우드 인프라의 리소스를 확인하고 필요 시 세부 설정을 조정할 수 있는 뷰(이하 "리소스 관리뷰"라고 한다)를 제공한다(도 9 참조).
- [99] 리소스 관리뷰는 각테일 서버를 구성하는 클라우드 인프라 리소스를 확인하고 설정을 세부적으로 변경할 수 있다. 여기서 각테일 서버는 어플리케이션 오케스트레이션을 위한 기본 구성을 자동으로 수행하지만, 필요한 경우 직접 클라우드 리소스를 조정할 필요가 있을 때 사용된다.
- [100] 리소스 관리뷰는 리소스 정보/액션 항목을 포함하는데, 리소스 정보 중 어플리케이션은 컨테이너 설정과 배포 정보를 관리한다. 클라우드 리소스 정보는 로드 밸런서·인스턴스(VM)·보안으로 구성되며, 인스턴스는 용량과 불륨을 관리한다. 조정이 필요한 리소스 정보는 액션을 통해 수행된다.
- [101] 미터링 기능은 어플리케이션이 사용하는 클라우드 인프라 리소스의 비용 정보를 확인할 수 있는 뷰(이하 "미터링뷰"라고 한다)를 제공한다(도 10 참조). 미터링뷰는 클러스터 인프라 사용 비용 항목·서버·리소스별 비용 항목 등을 포함할 수 있다.
- [102] 클러스터 인프라 사용 비용 항목에서는 클러스터와 각테일 서버가 사용하는 클라우드 리소스의 비용 현황을 확인할 수 있으며, 전월·현재월 비용 정보와 익월 추정 비용을 제공한다. 또한 월별로 비용 증감 추이 그래프를 제공한다.
- [103] 서버·리소스별 비용 항목은 각테일 서버별로 사용하는 클라우드 리소스 비용을 TOP을 기준으로 제공하고, 클라우드 리소스 종류별로 사용하는 비용을 TOP을 기준으로 제공한다.
- [104] 작업 관리기능은 배포·원격 명령·리소스 관리 등의 운영 작업을 스케줄링/자동화할 수 있는 관리 뷰(이하 "작업 관리뷰"라고 한다)를 제공한다(도 11 참조).
- [105] 작업 관리뷰는 어플리케이션과 인프라의 운영을 위한 스케줄링 및 일괄 처리 기능을 제공한다. 이러한 작업 관리뷰는 작업 현황 항목, 작업 관리 항목 등을 포함할 수 있다.
- [106] 작업 관리뷰에서 작업 현황 항목은 배포·원격 명령어·리소스 관리 태스크로 구분하고 각 태스크를 조합하여 구성된다. 여기서 배포는 어플리케이션 배포, 원격 명령어는 OS 명령어를 원격에서 수행, 리소스 관리는 스케일링, 상태/설정 변경을 의미한다.
- [107] 작업 관리뷰에서 작업 관리 항목은 즉시 수행·스케줄링·알람 발생에 따라 수행 방식을 설정할 수 있다. 알람 발생에 따른 수행은 용량 모니터링의 기준치에 따른 자동 스케일링 등에서 사용된다. 작업 관리 항목에서 작업의 실행 상태와 로그 확인을 제공한다.

- [108] 전사 현황 관리/분석기능은 전사 어플리케이션·클라우드·비용 현황을 파악하고 분석할 수 있는 각테일 대시보드(Dashboard)를 제공한다.
- [109] 각테일 대시보드는 전사 차원에서 어플리케이션과 클라우드 인프라의 현황을 조회하고 비용/예산 관리, 비용 최적화 분석, 통계 리포트를 제공하는 뷰이다. 이러한 각테일 대시보드는 어플리케이션 현황 항목, 클라우드 현황 항목, 비용/예산 관리, 비용 최적화 분석 항목, 통계/리포트 항목을 포함할 수 있다.
- [110] 어플리케이션 현황 항목을 통해 각테일 서버·클러스터·클라우드 컴포넌트의 표준화된 요소를 기준으로 어플리케이션과 인프라 현황을 전사적으로 파악하고 조회할 수 있고, 서비스 중심의 현황뷰를 제공한다.
- [111] 클라우드 현황 항목을 통해 전사에서 사용하는 클라우드를 공급자·리전·리소스별로 현황을 파악할 수 있으며, 인프라 중심의 현황뷰를 제공한다.
- [112] 비용/예산 관리, 비용 최적화 분석 항목을 통해 전사 클라우드 비용 현황을 파악하고 서비스별 예산 할당/통제와 최적화 분석을 통해 클라우드 리소스 비용 효율화를 할 수 있는 정보를 제공한다.
- [113] 통계/리포트 항목은 분석 및 보고에 필요한 통계 정보와 리포트뷰를 제공한다.
- [114] DB/저장소(150)에서 이미지 저장소(레지스트리)(180)는 어플리케이션 컨테이너의 등록·공유·다운로드·검색·버전을 관리하며, 모니터링 DB(170)는 어플리케이션과 인프라의 모니터링 정보를 관리하고, 통합 구성 DB(Configuration Management DB, CMDB, 160)는 프로바이더·네트워크·서비스·클러스터·서버·컴포넌트·클라우드 리소스의 구성 정보를 관리한다.
- [115] 도 12는 본 발명의 일 실시예에 따른 클라우드 플랫폼의 아키텍처를 나타내며, 도 13은 각테일 서버의 구성과 그 주변 아키텍처를 나타낸다.
- [116] 도 12를 참조하면 각테일 클라우드는 각테일 클러스터(200), 프로바이더 플러그인(210), 서버 매니저(220), DevOps 매니저, CMDB(160), 모니터링 DB(170), 이미지 레지스트리(180), API 서버(290), 사용자 콘솔(300)을 포함한다.
- [117] 각테일 클러스터(200)는 오케스트레이션 기반 아키텍처를 제공하고 프로바이더 플러그인(210)은 클라우드 공급자 API(280)를 통해 통합 관리를 위한 기본 모듈로 사용된다.
- [118] 클러스터(200)는 노드와 마스터로 구성되며, 노드의 경우 워커(worker, 310)를 통해 마스터의 명령어를 처리하는 구조이다. 워커(310)는 마스터와의 통신을 담당하고 수행 명령어에 따라 Executor가 지원된다. Monitoring Executor(320)는 노드와 컨테이너 모니터링 정보를 수집하고, Command Executor(330)는 OS와 컨테이너 명령을 수행한다. 그 외에 Container Engine(Docker, 340)이 있다.
- [119] 프로바이더 플러그인(210)은 멀티 클라우드와 Bare Metal을 위한 Kubernetes API 지원을 위한 API Rapper이며, 프로바이더 확장을 위한 플러그인 모듈로 구성된다. 각테일 서버는 어플리케이션 오케스트레이션의 기본 단위이며, 클러스터 마스터(200)와 프로바이더 플러그인(210)을 통해 컨테이너와 클라우드

인프라의 복제·스케일링·롤링 업데이트를 수행한다.

- [120] 각테일 서버는 도 13에 도시된 바와 같이 컨테이너와 클라우드 인프라로 구성되는데, 로드 밸런서·인스턴스(노드)·컨테이너·볼륨·보안 등으로 구성되며, AWS의 예를 들면, ELB·EC2 Instance·Security Group·ESB일 수 있다. 각테일 서버는 클라우드 제공자의 PaaS를 위해 클라우드 컴포넌트를 제공한다. 예를 들면 AWS의 RDS일 수 있다.
- [121] 서버 매니저(220)는 서버 내 어플리케이션 컨테이너와 인프라의 오케스트레이션을 수행하는 제어 모듈로서, 비정상 종료된 컨테이너를 재시작/복구하는 복제 제어, 스케일 인/아웃과 인스턴스 타입과 볼륨 확장을 통한 업다운을 수행하는 스케일링, 어플리케이션 컨테이너 배포를 순차적으로 무중단으로 수행하는 롤링 업데이트 기능을 제공한다.
- [122] DevOps 매니저는 멀티 클라우드 인프라 프로비저닝을 위한 구성 관리(Configuration Manager, 230), 멀티 클라우드 자원의 사용량 및 비용 관리를 위한 미터링 관리(Metering Manager, 240), 멀티 클라우드 자원 현황 및 설정 관리를 위한 자원 관리(Resource Manager, 250), 컨테이너/인프라 모니터링 정보 수집 및 관리를 위한 모니터링 관리(Monitoring Manager, 260), 여러 작업 태스크를 결합하여 일괄 수행하고 즉시 수행·수행 시간·이벤트 발생이 수행 조건이며, 배포·서버 액션·원격 명령어의 태스크를 위한 작업 관리(Job Manager, 270)를 제공하는 것으로서 DevOps를 위한 매니저 모듈이다.
- [123] 각테일 클라우드는 어플리케이션과 인프라의 구성 정보 관리·모니터링 정보 관리·어플리케이션 컨테이너 이미지 관리를 위한 DB를 제공하고, 사용자와 프로그래밍을 위한 인터페이스를 제공한다.
- [124] CMDB(160)는 프로바이더·네트워크·서비스·클러스터·서버·컴포넌트·클라우드 리소스의 구성 정보를 관리한다.
- [125] 모니터링 DB(170)는 어플리케이션과 인프라의 모니터링 정보를 관리한다.
- [126] 이미지 레지스트리(180)는 어플리케이션 컨테이너의 등록·공유·다운로드·검색·버전을 관리한다.
- [127] API 서버(290)는 각테일 클라우드의 모든 기능을 API(280)로 제공하고, 기업 전략에 따른 맞춤화와 타 솔루션과의 연계를 지원한다.
- [128] 사용자 콘솔(Console)(300)은 Web GUI 형태로 제공된다.
- [129] 이러한 각테일 클라우드는 다음과 같이 활용될 수 있다.
- [130] 첫째, 멀티 클라우드로서 활용될 수 있다.
- [131] 각테일 클라우드는 표준화 컴포넌트를 통해 이질적이고 복잡한 멀티클라우드 환경의 통합 관리를 위한 플랫폼이며, 또한 어플리케이션 중심의 기업 클라우드 전략을 구현한다. 구체적으로 각테일 클라우드는
 프로바이더·네트워크·서비스·클러스터·서버·클라우드 컴포넌트를 통해 관리
 대상을 표준화하고 이질적이고 복잡한 멀티 클라우드 리소스의 통합 관리(통합
 계정·자원·비용)하는 표준화 관리 컴포넌트이다. 또한 어플리케이션은

비즈니스의 핵심 자원인데, 각테일 클러스터를 통해 어플리케이션 가용성과 확장성이 강화되고, 각테일 DevOps View를 통한 개발/운영 업무 효율화를 통해 어플리케이션 중심의 기업 클라우드를 구현할 수 있다.

- [132] 둘째, 각테일 클라우드는 사내, 데이터 센터 Bare Metal 인프라의 클라우드화를 통해 하이브리드 클라우드를 구축/운영의 기반을 제공한다. 또한 복잡한 하이브리드 인프라의 통합 관리와 개발/운영 효율화를 제공한다.
- [133] 구체적으로 사내, 데이터 센터의 Bare Metal 인프라에 어플리케이션 클러스터를 구성하여 컨테이너 기반의 클라우드 환경을 구축함으로써 별도 가상화를 위한 플랫폼이 불필요하며, 가용성·스케일링 등 확장성을 제공하고, 기존 프라이빗과 퍼블릭 클라우드를 통합 관리할 수 있는 물리 인프라의 클라우드화를 구현할 수 있다.
- [134] 또한 각테일 클라우드의 표준 컴포넌트를 통해 관리하고, 각테일 클라우드 DevOps 뷰를 통한 개발/운영 업무 효율화를 제공한다.
- [135] 셋째, 각테일 클라우드는 컨테이너와 CI/CD를 위한 자동화를 통해 클라우드 상의 어플리케이션의 효율적 관리와 마이크로서비스의 구축 및 운영 플랫폼을 제공한다.
- [136] 각테일 클러스터는 컨테이너를 기반으로 클라우드 인프라에서 어플리케이션 배포 및 관리 환경을 제공(클라우드 네이티브 어플리케이션)한다. 여기서 각테일 클러스터는 마이크로 서비스를 구축하고 관리하는 기본 단위이다.
- [137] 각테일 DevOps 뷰의 작업 관리는 어플리케이션을 빌드하고 배포할 수 있는 자동화 기반을 제공하고, 컨테이너는 CI/CD를 보다 가볍고 용이하게 수행할 수 있는 기술이다. 각테일 클라우드는 멀티/하이브리드 클라우드 상에 어플리케이션을 배포/운영할 수 있는 플랫폼을 제공한다.
- [138] 넷째, 각테일 클라우드는 클라우드 서비스 브로커의 인프라 재판매 및 서비스 제공 플랫폼으로도 활용될 수 있다.
- [139] 퍼블릭 클라우드·데이터 센터 인프라를 통합 관리하고 사용자에게 재판매와 클라우드 관리 플랫폼을 서비스 형태로 제공하는 CSB용 플랫폼을 각테일 클라우드로 구축·운영하고, SaaS를 위한 멀티테넌시와 빌링 시스템을 제공하며, 큰 규모의 기업의 경우 계열사 클라우드 제공 및 관리 플랫폼으로 활용 가능하다.
- [140] 또한 기존 데이터 센터 사업자의 인프라를 클라우드화하여 제공하고, 퍼블릭 클라우드 제공자에 특화된 서비스(각테일 클라우드 컴포넌트(PaaS))를 제공한다.
- [141] 한편, 컨테이너에 있는 저장 공간은 임시적이고 제시작시 저장 공간 내 파일이 삭제되는 문제점이 있다. 이런 문제로 본 클라우드 플랫폼 시스템은 볼륨 객체를 이용하여 외부의 저장 공간과 컨테이너를 논리적으로 연결하여 사용하는 컨테이너 볼륨(스토리지) 프로비저닝 기능을 제공한다.
- [142] 본 발명의 컨테이너 볼륨(스토리지) 프로비저닝 기능은 GUI 상에서 다양한

저장 스토리지(nfs, 클라우드 스토리지 등)에 블록 정보를 생성하고, 컨테이너에 필요한 저장 공간을 할당 관리하는 기능이다(도 14 참조).

- [143] 블록 정보 등록은 도 15에 도시된 바와 같이 정적(static) 블록 생성과 동적(dynamic) 블록 생성으로 나뉠 수 있으며, 정적 블록 생성은 사전에 구성한 저장 공간 경로를 지정하여 해당 저장 공간에서만 사용되는 것이며, 동적 블록 생성은 저장 공간 풀(pool)을 생성하고 컨테이너가 블록 요청 시 필요한 공간만큼 동적으로 할당하여 제공하는 것이다.
- [144] 도 16은 본 발명의 일 실시예에 따른 클라우드 플랫폼 시스템의 컨테이너 블록(스토리지) 프로비저닝 방법을 나타낸 흐름도이다.
- [145] 먼저, 블록 할당이 요청되면(S400), 클라우드 플랫폼 시스템은 정적 블록 생성인지 동적 블록 생성인지를 판단한다(S410, S490).
- [146] 정적 블록 생성이면, 클라우드 플랫폼 시스템은 정적 블록 등록 리스트를 조회한다(S420). 등록된 정적 블록이 선택되고 용량 할당이 요청되면(S430, S440), 클라우드 플랫폼 시스템은 가용 공간을 확인한다(S450). 가용 공간이 부족하면, 클라우드 플랫폼 시스템은 저장 공간이 부족함을 알리고(S460), S430 단계로 복귀한다. 반면 가용 공간이 부족하지 아니하면 클라우드 플랫폼 시스템은 정적 블록 용량을 할당하고 정보를 생성하여 블록 할당을 완료한다(S470, S480).
- [147] 동적 블록 생성이면, 클라우드 플랫폼 시스템은 용량 할당을 요청하고 가용 공간을 확인한다(S500, S510). 가용 공간이 부족하면 클라우드 플랫폼 시스템은 저장 공간이 부족함을 알리고(S520), S500 단계로 복귀한다. 반면 가용 공간이 부족하지 아니하면 클라우드 플랫폼 시스템은 저장 경로를 생성하고 용량 할당 및 정보를 생성하여 블록 할당을 완료한다(S530, S480).
- [148] 도 17은 본 발명의 일 실시예에 따른 블록 정보 등록 화면의 예시도이며, 도 18은 본 발명의 일 실시예에 따른 등록 블록 정보 현황 화면의 예시도이다.
- [149] 도 17에 도시된 바와 같이 블록 정보를 생성하기 위해 스토리지에 블록 사용 정보를 등록할 수 있다. 도 17에서 등록된 블록 정보는 도 18의 등록 블록 정보 현황에서 확인할 수 있다.
- [150] 도 19 및 도 20은 본 발명의 일 실시예에 따른 블록을 할당하기 위한 화면을 나타낸 것으로, 도 19는 생성된 블록 정보를 선택하고 블록 이름, 액세스 모드, 할당 용량을 입력하는 블록 유형 설정 기능을 포함하고, 도 20은 할당한 블록의 경로와 컨테이너 상의 경로를 매핑하는 블록 마운트 기능을 포함한다.
- [151] 한편 상술한 본 발명의 실시예들은 컴퓨터에서 실행될 수 있는 프로그램으로 작성 가능하고, 컴퓨터로 읽을 수 있는 기록 매체를 이용하여 상기 프로그램을 동작시키는 범용 디지털 컴퓨터에서 구현될 수 있다. 상기 컴퓨터로 읽을 수 있는 기록매체는 마그네틱 저장매체(예를 들면 롬·플로피 디스크·하드디스크 등), 광학적 판독 매체(예를 들면 CD-ROM·DVD 등) 및 캐리어 웨이브(예를 들면 인터넷을 통한 전송)와 같은 저장 매체를 포함한다.

- [152] 이와 같이 본 발명에 의한 클라우드 플랫폼에서 어플리케이션을 컨테이너화하는 방법에 따르면 격리된 어플리케이션 실행 환경을 제공하고, 독립적인 자원 할당이 가능하며, 동일 호스트상 다중 어플리케이션 운영이 가능할 뿐만 아니라 OS 수준의 가상화로 빠른 조작성이 가능하고, 적은 크기의 컨테이너 이미지로 배포 및 업데이트가 효율적이며, 어디든지 이동이 가능하다.
- [153] 또한 본 발명에 의한 클라우드 플랫폼 시스템은 외부의 저장 공간과 컨테이너를 논리적으로 연결하여 어플리케이션 컨테이너에 필요한 저장 공간을 할당할 수 있다.
- [154] 이제까지 본 발명에 대하여 그 바람직한 실시예들을 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

청구범위

- [청구항 1] 어플리케이션 컨테이너의 볼륨 할당이 요청되면, 클라우드 플랫폼 시스템이 정적 볼륨 생성인지, 동적 볼륨 생성인지를 판단하는 단계;
 상기 정적 볼륨 생성이면, 상기 클라우드 플랫폼 시스템이 정적 볼륨 등록 리스트를 조회하고, 등록된 정적 볼륨이 선택되고 할당 용량이 입력되면, 상기 클라우드 플랫폼 시스템이 가용 공간을 확인하며, 상기 가용 공간이 부족하면 상기 클라우드 플랫폼 시스템이 저장 공간이 부족함을 알리고, 다시 등록된 정적 볼륨이 선택되고 할당 용량이 입력되기를 대기하고, 상기 가용 공간이 부족하지 아니하면 상기 클라우드 플랫폼 시스템이 정적 볼륨 용량을 할당하고 정보를 생성하여 볼륨 할당을 완료하는 단계;
 및
 상기 동적 볼륨 생성이고 할당 용량이 입력되면, 상기 클라우드 플랫폼 시스템이 가용 공간을 확인하며, 상기 가용 공간이 부족하면, 상기 클라우드 플랫폼 시스템이 저장 공간이 부족함을 알리고, 다시 할당 용량이 입력되기를 대기하고, 상기 가용 공간이 부족하지 아니하면 상기 클라우드 플랫폼 시스템이 저장 경로를 생성하고 용량 할당 및 정보를 생성하여 볼륨 할당을 완료하는 단계를 포함하는 클라우드 플랫폼에서 어플리케이션 컨테이너 볼륨(스토리지) 프로비저닝 방법.