기술동향보고서

데브옵스 기반 Provisioning 도구 기술분석 보고서

2016. 11

슈퍼컴퓨팅본부
슈퍼컴퓨터시스템개발실

한국과학기술정보연구원
Korea Institute of Science and Technology Information
www.kisti.re.kr

데브옵스 기반 Provisioning 도구
기술분석 보고서

부　서: KISTI 슈퍼컴퓨팅본부
작성자: 조혜영, 박주원, 안명호, 박주연

한국과학기술정보연구원

1. DevOps

(1) DevOps 개념

DevOps는 “Developer + Operator”의 합성어로 일종의 새로운 IT 문화라고 할 수 있다.

기존 레거시 환경에서는 IT 인프라를 개발과 운영이라는 2개의 큰 영역으로 분리했고, 각각의 영역에 개발자와 운영자라는 별도의 직업군이 있었다. 하지만 클라우드 시대에 접어들면서 개발과 운영이 하나로 합쳐지기 시작했다.

왜냐하면 클라우드에서는 “Standard API”를 통해 서버, 네트워크, 스토리지등을 관리할 수 있게 되었고, 이를 통해 자동화가 가능해졌기 때문이다. 이와 더불어 클라우드에서는 모든 컴퓨팅 리소스가 소프트웨어로 이루워진 가상의 것이기 때문에, 이전에 하던 것 처럼 서버에 이상이 생겼을때 리부팅을 하거나, 전원을 오프시키기 위해 누군가가 직접 데이터센터의 서버에 가서 조작을 해야 될 필요성이 없어졌다.

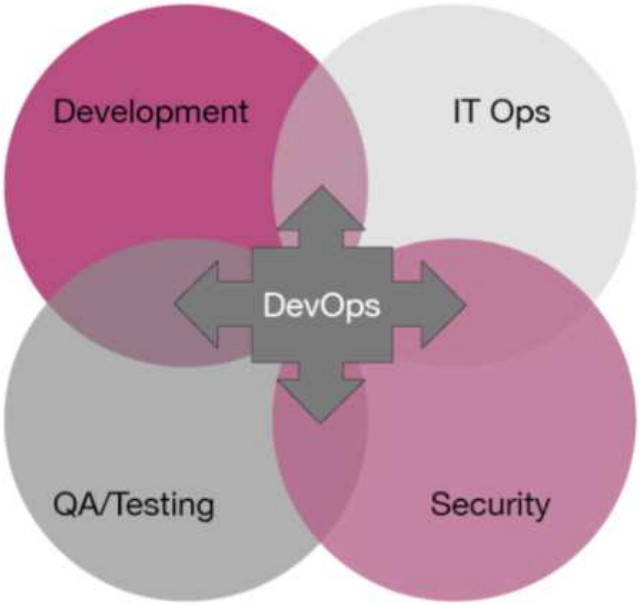
또한 클라우드에서는 에러에 대한 대처방법이 달라진 것도 큰 의미를 가지게 된다. 클라우드에서는 에러가 발생했을때, 문제가 발생한 서버를 중지시킨후에, 동일한 역할을 하는 별도의 서버를 프로비져닝해 사용하면 되기 때문에 이전보다는 훨씬 더 유연하고 빠르게 에러에 대처할 수 있게 되었다.

Wiki에서는 DevOps에 대해 다음과 같이 정의한다.

> ***DevOps*** *(a clipped compound of "development" and "operations") is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably.*

DevOps에 대해 새로운 개발방식 혹은 프로그래밍스타일이 아닌 “Culture, Movement or practice”라고 정의한 것이 유의할 부분이다. 이는 많은 것을 함축하는데, 간단히 이야기하면 DevOps는 단순한 관리방식의 변화가 아니라 프로

세스, 사람 그리고 목적에 관련된 것이기 때문에 “Culture, Movement or practice”라고 표현한 것이다.
아래의 그림은 DevOps의 전체적인 개괄에 대해 파악할 수 있는 것으로 연관된 영역을 잘 나타내주고 있다.
Development, IT Operation, QA Testing, Security등이 DevOps에서 다뤄야 할 주된 기능영역이다.
기존에는 위의 각각의 영역이 서로 다른 분야로 분리되고, 이를 처리하는 사람이 달랐지만 DevOps 에서는 이를 모두 동일한 사람 혹은 조직에서 모두 처리한다. 얼핏생각하면 개발뿐만 아니라 운영과 보안까지 그 광범위한 영역을 모두 일관되게 처리한다고 얘기할 수 있다.



< Functions in DevOps >

DevOps를 보다 간단하게 한마디로 표현하다면 “Quality Movement”라는 단어가 가장 적절한 문구일 것이다.
제조업에서는 Lean, Toyota Kanban, 그리고 Total Quality와 같은 방법들을 통해 생산성을 높임과 동시에 일정수준 이상의 품질을 확보할 수 있도록 많은 노력을 기울여 왔다.
DevOps는 앞서 예를 든 제조업과 비슷한 관점과 비슷한 목적으로 가지고 추진

된 IT에서의 생산성과 품질의 향상을 위한 방안이라고 말할 수 있다.
아래의 인용은 DevOps에 대한 이해를 넓혀 줄 수 있는 좋은 문구로 왜 DevOps를 지칭할때 개발자적인 관점에서의 용어를 사용하지 않고 Culture, Movement와 같은 단어를 사용하는지 잘 보여준다.

> *Agile was instrumental in Development regaining the trust in the business, but it unintentionally left IT Operations behind.*
> ***DevOpsisawayforthebusinesstoregaintrustintheentireITorganizationasawhole.”***

(2) Comparison

아래의 표는 기존의 IT운영방식과 DevOps의 운영방식을 비교한 표로 DevOps는 도대체 무엇이 다른지를 명확히 알 수 있도록 도와준다.

Yesterday's IT		Today's High Performing DevOps
IT as a cost center		IT as a competitive advantage
Developers blame Ops—Ops blames developers		High trust organization with shared goals
Waterfall development		Continuous integration and delivery
Performance considerations begin at code check-in	VS.	Performance is a critical requirement and is considered throughout the development lifecycle
Deployments only 1-2x a year		Multiple deployments a day
Change approval board required to deploy		Peer review of code quality
Version control for code only or no version control at all		Version control for code, system, application configurations and data
Failures found by NOC or worse, by customers		Proactive monitoring to find failures first

여러 가지 사항들이 있지만 가장 중요한 것은 다음의 항목들이다.

- IT As a competitive advantage
- High Trust organization with shared goals
- Multiple deployments a day

위의 사항들은 특히 첫번째와 두번째는 매우 밀접한 연관관계를 가지고 있는 사항이다. 물론 세번째 사항도 매우 깊은 연관을 가지고 있다.
위의 3가지를 관통하는 핵심 키워드는 "비지니스로의 관점 변화"라고 이야기 할 수 있다.
IT 연관자에게 있어 비지니스는 첫번째로 고려할 대상이 아니었다. 언제나 인프라. 시스템, 어플리케이션과 같은 것들이 첫번째 목표대상이었다.
DevOps에서는 관점이 더이상 IT 시스템이 아니고 Business이다.
따라서 IT가 비지니스 경쟁력을 가져야 하고, DevOps조직과 구성원들은 공통의 비지니스 목표를 가지고 이를 달성하기 위해 노력해야 한다. 더불어 고객의 만족과 제공하는 서비스의 가치를 높이기 위해 수차례 디플로이먼트를 실행한다.

(3) DevOps의 효과

DevOps를 제대로 이해하고 수행하게 되면 다음과 같은 효과가 있다고 있다.

* 아래의 자료는 Puppetlabs에서 2013,2014년에 조사한 결과에 기초하고 있다.

〈 Power of DevOps 〉

세계최대의 Public Cloud Provider인 아마존은 DevOps를 적용해 다음과 같은 결과를 보여주고 있다.

11.6초마다 23,000 deployments
0.001% deployment error

Source : Amazon, 2012 Velocity Presentation Presentation.

(4) DevOps Success Factors

DevOps가 성공하기 위해서는 다음과 같은 요소들이 중요하다.

■ **Collaboration**

- DevOps를 성공적으로 실현하기 위해서는 IT조직에는 익숙치 않은 모든 과정에 있어 협업이 중요하다.
- DevOps 조직은 타 IT 조직으로부터의 요구사항 뿐만 아니라, 전통적으로 협업을 하지 않았던 부서들, 예를 들어 마케팅, 세일즈 그리고 고객관리조직들로부터의 요구사항도 귀귀울여 들어야 한다.
- DevOps 조직에서는 다양한 조직으로부터의 요구를 수렴하여 보다 생산적인 그리고 고객에게 더욱 더 많은 가치를 선사할 수 있도록 협업이 중요하다.

■ Cloud and Virtualization

- DevOps를 기술적으로 실현하기 위해서는 Self-Service 그리고 개발과 테스트를 위한 보다 빠른 프로비져닝 시스템등이 반드시 필요하다. 이를 위해서는 클라우드 와 가상화가 필연적이라고 할 수 있다.

■ Automation

- 의심할 여지 없이 DevOps는 필요한 모든 프로세스를 지속적으로 반복할 수 있도록 자동화시키는 것이 필요하다.
- 이러한 자동화는 실행에 필요한 시간을 대폭 줄여 생산성을 줄일 수 있을뿐만 아니라 human error를 줄일 수 있기 때문에 매우 중요하다.

■ Version Control

- DevOps 환경에서는 Version Control과 Documentation 역시 매우 중요한 성공요소의 하나이다.
- DevOps에서는 영역에 대한 명확한 구분없이 필요한 일을 빨리 처리하는 것이 중요하기 때문에 누가, 언제, 무엇을, 왜 했는지등에 대한 내용을 상세히 기록하고 이를 문서화해야 한다.
- 이러한 프랙티스는 전체 프로세스를 투명하게 만들고 문제가 발생했을때 문

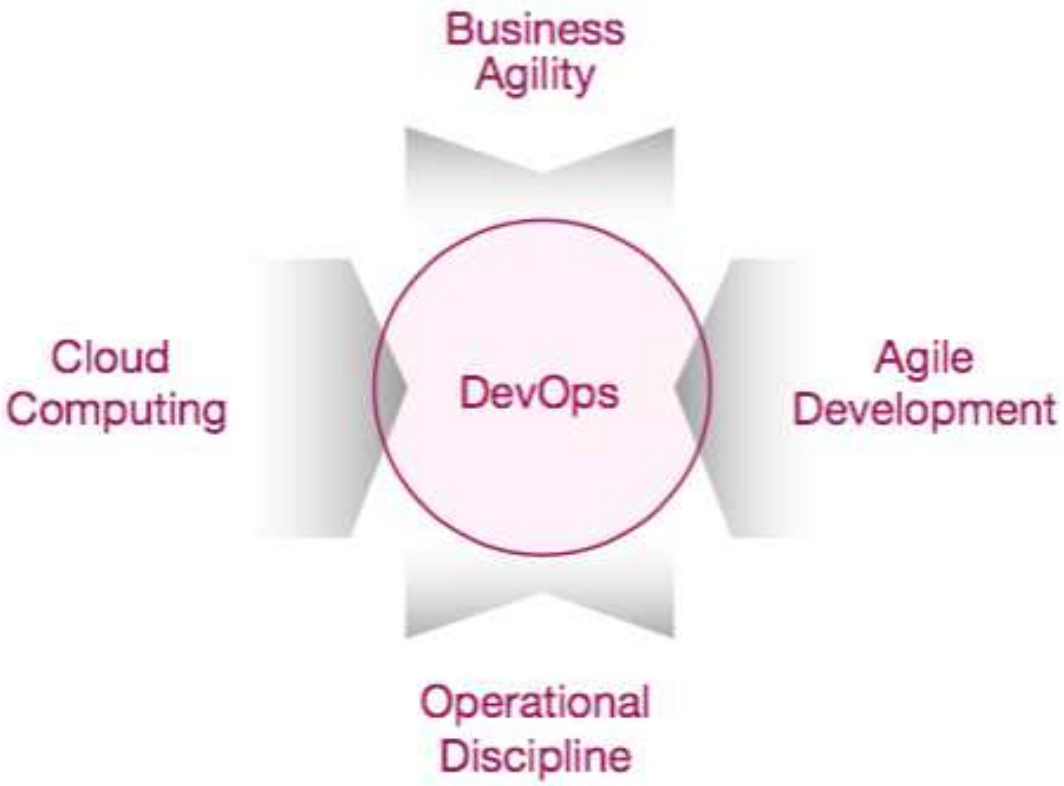
서화된 내용에 따라 빠른 시간에 조치할 수 있어 반드시 실현해야 한다.

■ Configuration Management

- 기술적인 측면에서 보다면 Configuration Management는 DevOps에서 가장 중요한 역할을 한다고 할 수 있다. 왜냐하면 Configuration Management가 DevOps의 가장 기본적인 업무들을 자동화시켜줄 수 있기 때문이다.
- DevOps를 도입하려는 곳에서는 자신의 조직과 해결하려는 문제에 적합한 Configuration Management Tool을 반드시 선택해 적용해야 한다.

■ Culture

- DevOps는 프로그램개발도 아니고, 새로운 프로그래밍언어도 아니다. 이것은 고객, 즉 비지니스 관점에서 IT 서비스를 개발하고 운영하려는 것이기 때문에 Culture라고 할 수 있다.
- DevOps 조직내의 모든 사람들에게 DevOps에서 필요로 하는 각종 도구와 프로세스 그리고 협업등이 문화로 스며들 수 있도록 노력해야 한다.



< DevOps Drivers >

(5) DevOps Benefits

DevOps를 통해 얻을 수 있는 이점들은 다음과 같다.

■ Reduced time to market

- DevOps를 도입하면 고객들의 요구를 더욱 빠른 시간내에 수용할 수 있어, 경쟁력을 높이고 고객충성도를 높일 수 있다.

■ Faster return on investment

- 고객의 요구사항을 빠른 시간내에 충족시켜 이를 통해 보다 높은 ROI를 빠르게 달성할 수 있다.

■ High Performance

- DevOps를 도입한 대부분의 기관에서는 퍼포먼스 향상을 경험하고 있다.
- 이러한 향상의 주된 이유는 critical application을 서비스 중지없이 deployment할 수 있게 되기 때문이다. Google, Facebook, Amazon, Twitter와 같은 서비스들은 이러한 능력을 바탕으로 하루에도 수차례 개선된 기능과 버그 패치가 된 application을 deploy해 경쟁력을 점점 높혀가고 있다.

■ Increased Quality

- DevOps 프랙티스에서 얻는 것은 Deployment speed가 전부는 아니다. Quality control도 역시 매우 중요한 요소로, 자동화와 Test provisioing을 이용해 배포하려는 시스템의 문제상황을 사전에 탐지해 대처할 수 있도록 해준다.

■ Customer Satisfaction

- 다양한 조직으로부터의 의견은 고객의 요구사항에 대한 이해를 높혀, IT 시스템이 비지니스적으로 어떤 목적을 가지고 서비스를 제공해야 하는지를 잘 이해하도록 한다.

■ Reduced IT Waste

- DevOps는 long lead time을 줄여 시간과 IT 리소스를 효율적으로 사용할 수 있도록 한다.
- 기존에는 Application Deployment를 위해 사람이 수작업으로 해야 했던 여러가지 일들, 예를들어 version control, continuous delivery 그리고 기타 프로세스들을 자동화시켜 이러한 낭비를 대폭 줄일 수 있다.

■ Human Error

- DevOps는 Human Error를 줄여 준다. 개발과 운영은 필연적으로 사람의 개입이 필요한데, 사람의 개입이 많으면 많을수록 human error에 의해 문제가 발생할 가능

성이 높아진다.
- DevOps는 자동화를 통해 이러한 가능성을 줄여준다.

(6) DevOps Challenges

DevOps를 도입하는 과정은 멀고도 험난한 과정이라 할 수 있기 때문에 신중한 준비와 과감한 실행이 필요하다.
아래의 내용들은 DevOps를 도입하는데 흔히 발생하는 문제점들에 대한 기술이다.

■ Misconception
- DevOps에 대한 개념을 명백히 이해하는 것이 중요하다.
- 혹자는 DevOps가 프레임워크나 프로세스를 무시하고 임기응변식으로 quality testing, deployment등을 수행하는 것으로 생각할 수도 있는데 이는 사실과 매우 다르다.
- DevOps는 체계적이고 유기적으로 필요한 일을 조직화해 처리하는 활동이다.

■ Need for automation
- 자동화는 표준화된 프로세스와 consistency ,repeatability 그리고 maturity를 필요로 한다.
- 앞서 언급한 요소들이 성숙되기 전까지는 신뢰할 수 있고, 안전하게 동작하는 자동화를 이루기 힘들다.
- 도입하려는 조직의 성숙도에 따라 단계별로 자동화를 적용하는 것이 좋다.

■ Skills
- DevOps에서 사용되는 도구들을 쓸줄 안다고 해서 DevOps를 적용하기 위한 모든 스킬을 갖추었다고 이야기하기 힘들다.
- 중요한 것은 DevOps 팀 전체 멤버들이 정의된 프로세스를, 정해진 도구를 사용해 지속적으로 수행할 수 있는 것이다. 이것이 진정한 스킬이라고 할 수 있다.

■ Culture
- DevOps를 적용하기 위한 문화가 매우 필요하다. DevOps 팀 멤버간의 협업과, 정해진 프로세스에 따라 일을 진행하고, 이를 문서화하는 등의 일련의 모든 활동이 문화로 정착되어야 한다.

■ Silo Mentality
- Silo Mentality는 어떻게 생각해보면 Devops 를 적용하는데 가장 큰 걸림돌이 될 수 있는 것으로 협업과 긴밀한 커뮤니케이션이 필수적인 DevOps 환경에서 이를 무

시하고 혼자 일을 처리하는 것이다.

- 업무를 수행하는 본인의 생각과 기술 그리고 여러가지 리소스들을 다른 사람과 공유하지 않거나, 다른 사람의 의견을 잘 받아들이지 않는다면 DevOps는 제대로 정착되기 어려울 수 밖에 없다.

(7) DevOps Recommendations

1. Make Deployments business as usual
   필요한 시점에, 혹은 자주 application deployment를 할 수 있도록 1) feature release를 decoupling하고 2) deployment size를 줄일 수 있어야 한다.

2. Make everyone a performance engineer
   위의 문장의 의미는 실제 운영환경과 비슷한 환경을 만들고 이를 모든 개발자와 DevOps에게 액세스 할 수 있도록 해야 한다. 새롭게 개발되는 기능이나, 혹은 기존의 버그 패치, application deployment를 모두 production-like 환경에서 실행해 문제를 사전에 찾아 해결할 수 있도록 한다.

3. Strengthen the safety net with automated testing
   개발종료시점이 아니라 개발하는 과정에도 사용할 수 있도록 Application Deployment에 필요한 테스트를 자동화 시켜놓아야 한다. 이렇게 해야만 발생가능한 문제를 사전에 탐지해 문제를 빠른 시간내에 해결할 수 있다.

4. Be Strict with version control
   Version Control은 개발뿐만 아니라 운영에 있어서도 매우 중요한 요소이다. Version Control을 운영에 사용하는 것은 DevOps 조직의 퍼포먼스에 막대한 영향을 미친다.

5. Implement end-to-end performance monitoring and metrics
   IT 시스템 퍼포먼스 모니터링 정보를 제공하는 것은 개발, 테스팅 그리고 운영에 있어 매우 유용한 직관을 제공하는 것과 다름없다. 연관된 각각의 부서와 사람들은 자신의 일이 시스템 전체 퍼포먼스에 어떤 영향을 미치는지를 잘 알게 되어 결과적으로 IT System Quality를 향상 시킬 수 있게 된다.
   특히 개발 혹은 실제운영 마지막에서 발견되는 “Last-Minute Surprise”를 방지하는데 매우 좋은 방법이다.

6. Peer code review
   Peer Code Review는 DevOps 조직의 퍼포먼스를 가늠해 볼 수 있는 매우 중요한 신호이다. 하지만 이는 큰 조직이라도 쉽게 행하기 어려운 것이 사실이다. 하지만

이러한 번거로움과 어려움이 있더라도 충분히 실행해볼만한 가치가 있다.

7. Allocate more cycles to the reduction of technical dept
DevOps를 실행하는데 있어 필요한 기술들은 적절한 시점에 모두 내재화를 하는 것이 필요하다. 시간상의 이유로 혹은 다른 핑계로 인해 차일피일 미루는 것은 어느날 그것으로 인해 엄청난 재앙을 불러올 것이 분명하기 때문이다. 특히 의심되는 부분은 연관된 작업을 실행하기 전에 반드시 확인하고, 이해되지 않는 부분을 없애 시스템의 안정성을 도모해야 한다.

"It is only through automated testing that we can transform fear into boredom... The only way you can get people productive is to show them there is a safety net underneath them that will catch errors long before they get into production."

— Eran Messeri, SCM, Google

Figure 4 Power of Automated Testing

2. OS provisioning tool

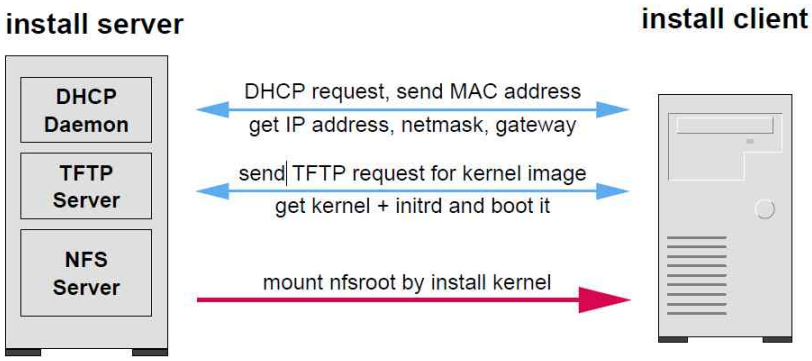
(1) Cobbler

* URL : http://cobbler.github.io/
* Feature
  - CLI 제공
  - 웹 인터페이스 : Django (python-django for Debian/Ubuntu)
  - Red Ha 기반 배포 : RHEL, CentOS, Scientific Linux
  - ISO Image를 갖고 system object를 만듬. = profile -> PXE booting.
  - Koan을 통하여 가상 프로비저닝 기능 제공
  - DHCP, DNS, rsync, TFTP 등 운영 관리

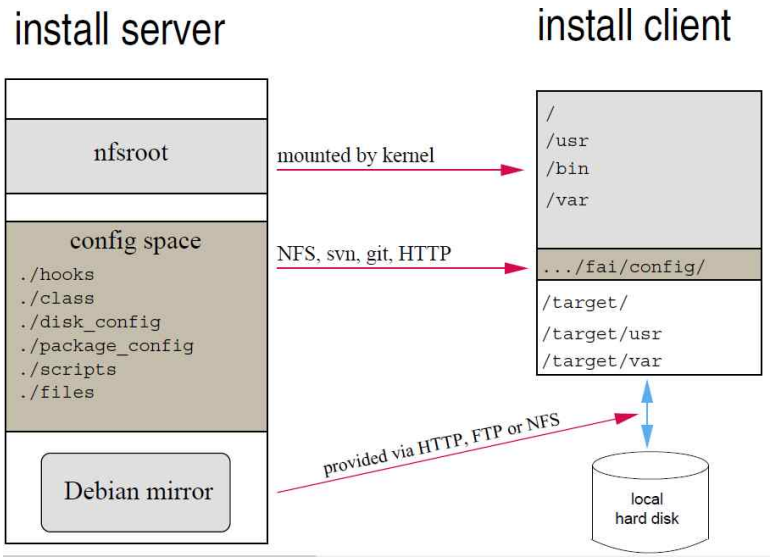
(2) FAI (Fully Automatic Installation)

* URL : http://fai-project.org/
* Features
- 설치 및 업데이트 : Debian, Ubuntu, CentOS, RHEL, SUSE, ...
- 중앙 집중식 배포 및 구성 관리
- KVM, XEN or VirtualBox and Vserver 등을 통한 가상머신 설치
- 손 쉬운 RAID 와 LVM 설정
- ssh을 통한 전체 설치 과정 원격 제어
- 통합 재해 복구 시스템
- 재설치 없이 사용 중인 시스템을 업데이트하여 매일 유지보수.
- 몇 대의 컴퓨터부터 수천 대의 머신까지 관리 가능한 확장성.
- 유사한 머신을 그룹으로 묶어 처리 가능.
- 커스터마이제이션을 위해 쉘, 펄, 익스펙트 스크립트, cfengine 스크립트 지원.
- 레드 햇의 kickstart나 cobbler, 수세의 autoyast, SUN 솔라리스의 Jumpstart와 비교
  해서 유연성이 높음. 훅을 이용해 대상에 맞은 설정들로 세세하게 튜닝 가능.
- GOsa를 통한 GUI 사용.

* Procedure
1. Boot host



2. Get configuration data



3. Run installation

- 디스크를 파티셔닝하고 파일 시스템을 만든다.
- apt-get 등의 명령어를 이용하여 소프트웨어를 설치한다.
- OS와 추가 소프트웨어를 설정한다.
- 설치와 관련된 로그 파일을 저장하고, 시스템을 리부팅한다.

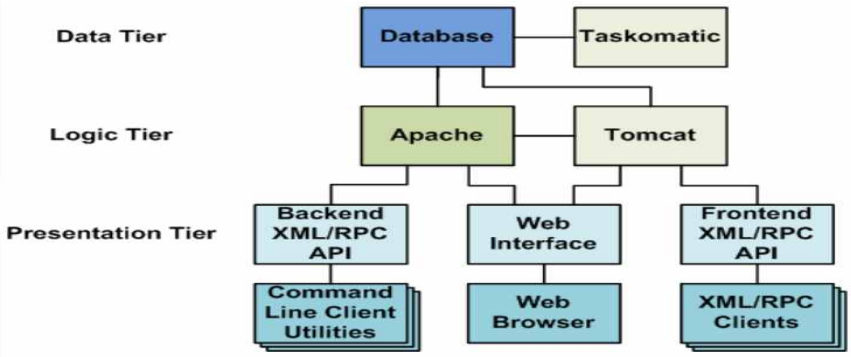
(3) Spacewalk

* URL : http://spacewalk.redhat.com/
* Feature

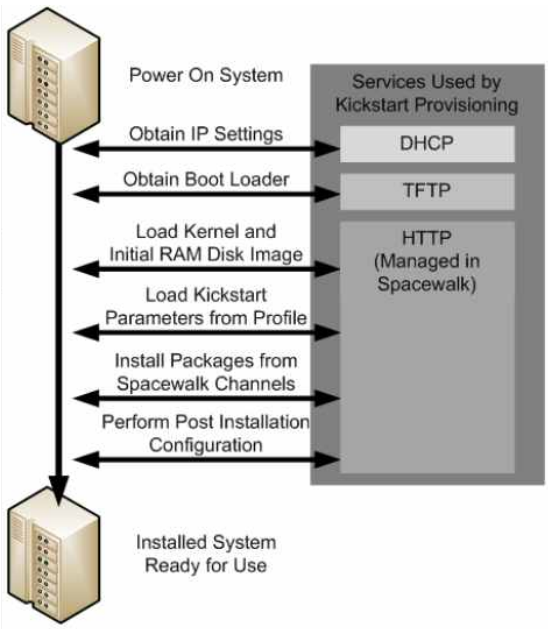
- 리눅스 시스템 관리를 위한 오픈 소스 프로그램
- Red Hat Satellite를 기반으로 제작된 소프트웨어
- 인벤토리 시스템 제공 (하드웨어와 소프트웨어 정보 관리)
- 시스템 설치 및 업데이트 제공

- 그룹 별 커스텀 소프트웨어 배포 및 관리
- RHEL, Fedora, CentOS, Scientific Linux, 등 지원.

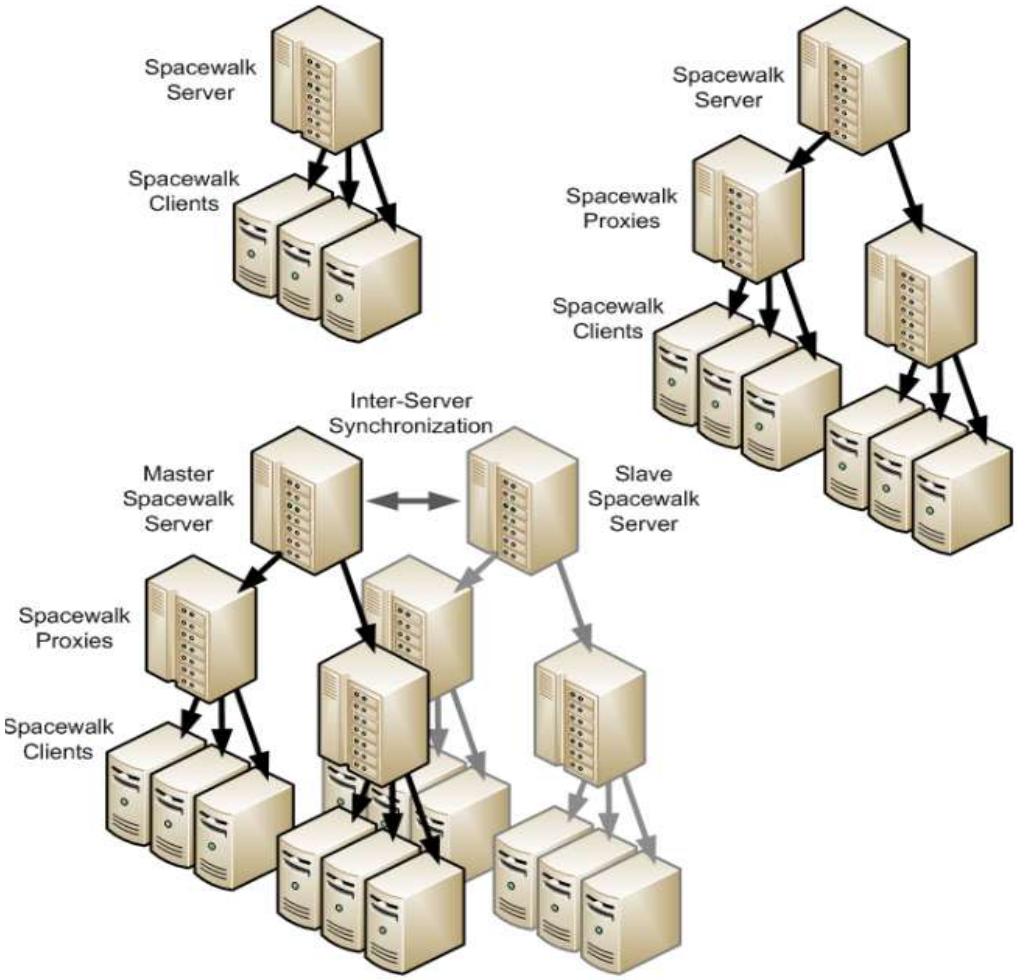
* Server Architecture _ 3 step



* 프로비저닝 프로세스



* 배포 환경 예제

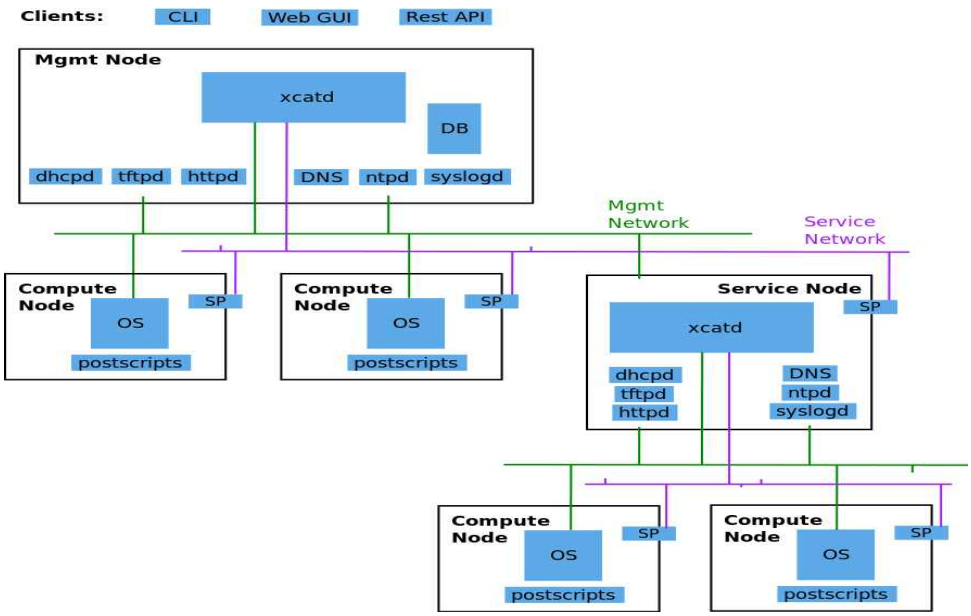

Spacewalk
Server
Spacewalk
Clients
Spacewalk
Server
Spacewalk
Proxies
Spacewalk
Clients
Inter-Server
Synchronization
Master
Spacewalk
Server
Slave
Spacewalk
Server
Spacewalk
Proxies
Spacewalk
Clients

(4) xCAT (Extreme Cluster/Cloud Administration Toolkit)

* URL : http://xcat.org
* Feature
- Support H/W : IBM Power, IBM Power LE, x86_64
- Support VM : IBM PowerKVM, KVM, IBM zVM, ESXI, XEN
- Support OS : RHEL, SLES, Ubuntu, Debian, CentOS, Fedora, Scientific Linux Oracle Linux, Windows, Esxi, RHEV, and more!
- Support Install Options : Diskful, Diskless, Cloning
- Built in Automatic discovery
- RestFUL API Support

* Operating System & Hardware Support Matrix

	Power	Power LE	zVM	Power KVM	x86_64	x86_64 KVM	x86_64 Esxi
RHEL	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SLES	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ubuntu	No	Yes	No	Yes	Yes	Yes	Yes
CentOS	No	No	No	No	Yes	Yes	Yes
AIX	Yes	No	No	No	No	No	No
Windows	No	No	No	No	Yes	Yes	Yes



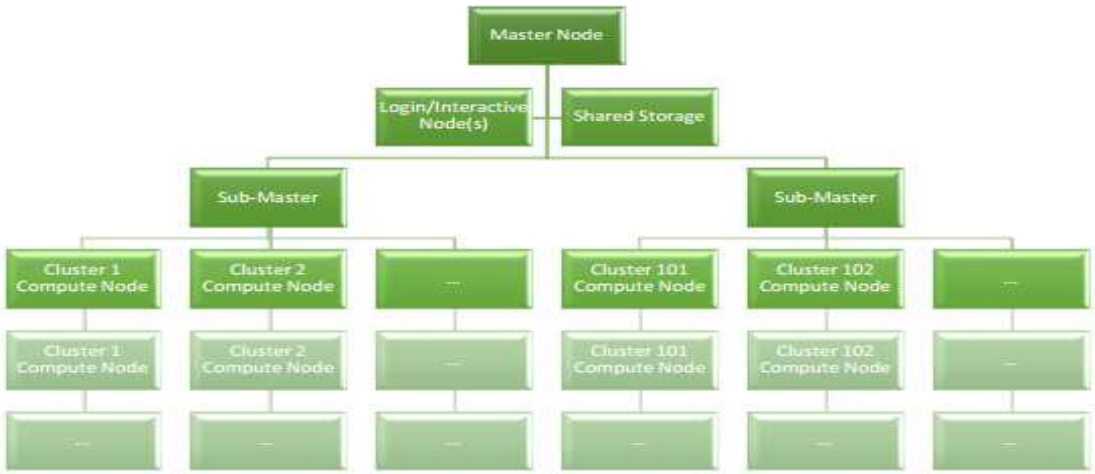
xCAT(Extreme Cluster / Cloud Administrator Toolkit)은 HPC Cluster, RenderFarms, Grids, WebFarms, 온라인 게임 인프라, Cloud, 데이터센터 등에 대한 지원이 가능하다. Provisioning은 스크립트를 사용하여 설치할 수 있으며, Statelite, iSCSI 또는 Cloning Remote manage system : DNS, TFTP, DHCP, HTTP, DNS 등의 관리 노드 서비스를 쉽고 빠르게 구성하고 관리할 수 있도록 lights-out management, remote console 그리고 분산 쉘을 지원한다. xCAT은 Hardware Server 검색, 원격 시스템 관리를 실행, 물리적 또는 가상 시스템, Diskful 또는 Diskless 상태의 물리적 서버에 OS 설치, 병렬 시스템 관리, 사용자 응용 프로그램 구성 등이 가능하다.

(5) Warewulf Cluster Managet

* URL : http://http://warewulf.lbl.gov/trac/wiki

Warewulf는 고성능 리눅스 클러스터를 관리하기 위하여 개발된, 확장 가능한 시스템 관리 제품이다. 일반적으로 확장 가능한 시스템 관리에 초점을 맞추어 개발되면서 모듈식 플러그인 (시스템 구성, 관리, 프로비저닝, 모니터링, 이벤트 알람) 아키텍쳐를 가지고 있다. 이에 사용자는 편리하게 사용하고자 하는 모듈 / 기능만을 따로 구성할 수 있어 사용자가 원하는 환경을 만들 수 있다. 따라서, Warewulf는 확장이 가능하고 사용하기 쉬운 솔루션이라고 말할 수 있다. Warewulf는 RHEL 기반의 배포판에 기반을 두고 개발되었으며, 우분투 및 우분투 기반의 리눅스 역시 지원하고 있다.

Warewulf는 처음부터 확장성 및 유연성을 염두에 두고 Object Oriented 프로그래밍을 사용하여 설계되었다. 또한 특정한 작업을 위해 Backend Data solution, Backend Interface, Module, Event handler를 가지고 있고, 어떤 특정 기능 또는 개선 사항들을 모듈로 제공하는 것을 기반으로 하고 있기에 하위 프로젝트는 모두 기능적 구성요소로 분리되어 관리된다.



2016년 현재 Warewulf는 3.7 버전(2016.06)이 배포되고 있으며, yum을 사용하는 레드햇· CentOS에서는 yum repository를 등록하여 바로 설치할 수 있고 다른 리눅스 배포판들에서는 소스코드를 다운받아 컴파일하여 사용할 수 있다.

◆ OS provisioning tool 비교 분석

- 기능 비교

	FAI	xCAT	Spacewalk	Cobbler	Warewulf
Language	Perl, Ruby, Python, etc	Perl	Java(C, Perl, Python, PL/SQL)	Python	
License	GNU GPL v2	Eclipse Public	GPLv2	GNU GPL v2	BSD
GUI	o	o	o	o	x
First release	1990	2011.05	2008.06	2011.01	2011.06
Latest release	2015.12	2016.05	2015.04	2016.01	2014.05

- Platform support

Support Platform	
FAI	Debian, Ubuntu, CentOS, RHEL, SUSE, ...
xCAT	RHEL, CentOS, Fedora, SLES, Ubuntu, AIX, Windows, VMWare, KVM, PowerVM, PowerKVM, zVM.
Spacewalk	Fedora, CentOS, SLE and Debian
cobbler	RPM-based
warewulf	Red Hat based distributions of Linux

3. S/W provisioning tool

(1) 기능 비교

	Ansible	Chef	Puppet	Salt	Juju
Language	Python	Ruby,Erlang	Ruby	Python	Python, Go
License	GPL	Apache 2.0	GPL Apache	Apache 2.0	Affero GPL
Mutual auth	o	o	o	o	o
Encrypts	o	o	o	o	o
Verify mode	o	o	o	o	x
Agentless	o	x	x	both	x
GUI	o	o	o	o	o
First release	2012.03	2009.01	2005.08	2011.03	2010.09
Latest release	2016.01	2016.03	2016.03	2015.07	2015.06

(2) Platform support

	Ansible	Chef	Puppet	Salt	Juju
AIX	o	o	o	o	x
*BSD	o	o	o	o	x
HP	o	o	o	Partial	x
Linux	o	o	o	o	o
OS X	o	o	o	o	x
Solaris	o	o	o	o	x
Window	Partial	o	o	o	o

참고 문헌

- How to choose tools for devops
- HighOps Operability eBook
- DevOps Toolchains Work to Deliver Integratable IT Process Management
- http://continuousdelivery.com/
- http://blog.octo.com/en/continuous-delivery-how-do-we-deliver-in-3-clicks-to-7000-machines/
- http://www.slideshare.net/CAinc/what-are-the-coolkidsdoingwithcontinuousdelivery?qid=4ae62edd-1432-4035-b8ca-b134d195888f&v=default&b=&from_search=1
- http://www.slideshare.net/dev2ops/devops-security
- http://www.bogotobogo.com/DevOps/DevOps_CI_CD_Pipeline_Sample.php
- http://www.slideshare.net/voux/owf2013-agnostic-continuous-deliveryowfmasks?qid=724a46b1-fb43-412f-a8b4-bc8fb2b52bb5&v=default&b=&from_search=1
- https://www.ibm.com/developerworks/community/blogs/c914709e-8097-4537-92ef-8982fc416138/entry/devops_best_practice_-_establishing_a_%E2%80%9Csingle_source_of_truth%E2%80%9D?lang=en
- http://www.slideshare.net/francoisledroff/master-chef-class-learn-how-to-quickly-cook-delightful-cqaem-infrastructures?qid=9198118a-5b10-4e3e-a7db-9d92e7c2d427&v=default&b=&from_search=7
- http://www.slideshare.net/CiscoDevNet/enabing-devops-in-an-sdn-world?qid=47004169-099b-4281-ac76-fed8fe1c09bd&v=default&b=&from_search=2

데브옵스 기반 Provisioning 도구 기술분석 보고서

2016 년 11 월 인쇄
2016 년 11 월 발행

발행처:

한국과학기술정보연구원
Korea Institute of Science and Technology Information
www.kisti.re.kr

대전시 유성구 과학로 335 번지
Tel: (042) 869-1004(代)

ISBN 000-00-000-0000-0 00000 보급가 : 비매품