**The Coconut Manual**

**Jihyun LEE**

**The Coconut Manual**

by Jihyun LEE

Published 2015

Copyright (c) 2015 L&K Logic Korea Co., Ltd.

This project is licensed under the terms of the GPL. This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. All product names are property of their respective owners. Such names are used for identification purposes only and are not indicative of endorsement by or of any company, organization, product, or platform.

이 프로그램은 자유 소프트웨어입니다. 소프트웨어의 피양도자는 자유 소프트웨어 재단의 GNU General Public License의 규정에 의해서 이 프로그램을, 개작된 2차적 프로그램과 함께 또는 개별적으로 재배포할 수 있습니다. 이 프로그램은 보다 유용하게 사용될 수 있으리라는 희망에서 배포되고 있지만 제품에 대한 어떠한 형태의 보증도 제공하지 않습니다. 보다 자세한 사항에 대해서는 GNU General Public License를 참고하시기 바랍니다. "제품의 모든 이름들은 원 제작자의 소유물입니다. 이런 이름들은 단지 구별을 위한 목적으로 사용되었으며, 어떤 회사, 조직, 제품 혹은 플렛폼의 혹은 그것들을 지지하는 목적으로 사용되지 않았음을 밝힙니다."(확인 필요)

GNU General Public License는 이 프로그램과 함께 제공됩니다. 만약 이 문서가 누락되어 있다면 자유 소프트웨어 재단(license-violation@gnu.org, Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA) 혹은 <license-violation@korea.gnu.org>으로 문의하시기 바랍니다.

**Table of Contents**

**I. Summary**

1. **History**
2. **Our goal**
3. **Essential particular**
4. **Architecture**
5. **Cautions**

**II. Installation**

1. **Install**
   1. **Required Packages**
   2. **Installing Redis**
   3. **Installing Coconut Manager/ Agent**
   4. **Installing Coconut Tool**

**III. Basics**

1. **Monitoring**
2. **Management**
3. **Control**

**IV. Advanced Topics**

1. **Commands**

**V. Goals**

1. **Goals**

**VI. Help**

1. **Help**

**VII. Reference**

1. **Reference**

**Chapter 1. History**

**2015. 05. 01**

**메뉴얼이 릴리즈 되었습니다.**

**Chapter 2. Our goal**

우리는 DevOps 팀에 알맞은 서비스 운영 툴 제작을 최우선 목표로 정하고 개발을 진행 중에 있습니다. 이를 위해 제작된 기능들에 대해서는 다음 Chapter에서 기술되어 있으며, 상세한 사용법은 "III. Basics"와 "IV. Advanced Topics"를 참고하세요.

우리가 원하는 목표를 이루기 위해 지속적으로 작업 중입니다. 추후 개발될 내용에 대해서는 "V. Goals"를 참고하세요.

**Chapter 3. Essential particular**

**Coconut**의 주요 기능은 아래 세가지로, 다음과 같습니다.

**시스템 관리**

하나의 서비스를 위해 많은 서버들이 활용되며, 여건에 따라 수십 대씩 늘어나거나 혹은 줄어들기도 합니다. 이런 동적인 운영 환경에서 원활한 서비스를 효율적으로 관리, 지원하기 위한 기능들을 제공합니다.

**모니터링**

등록된 시스템들의 중요한 현황들을 실시간으로 모니터링 합니다. 특정적인 것은 당신이 지정한 어플리케이션이 정상적으로 작동하는지를 실시간으로 모니터링 할 수 있으며, 당신이 지정한 서비스 프로세스의 오작동 여부를 판단하고, 문제점을 실시간으로 리포트하는 기능을 제공합니다.

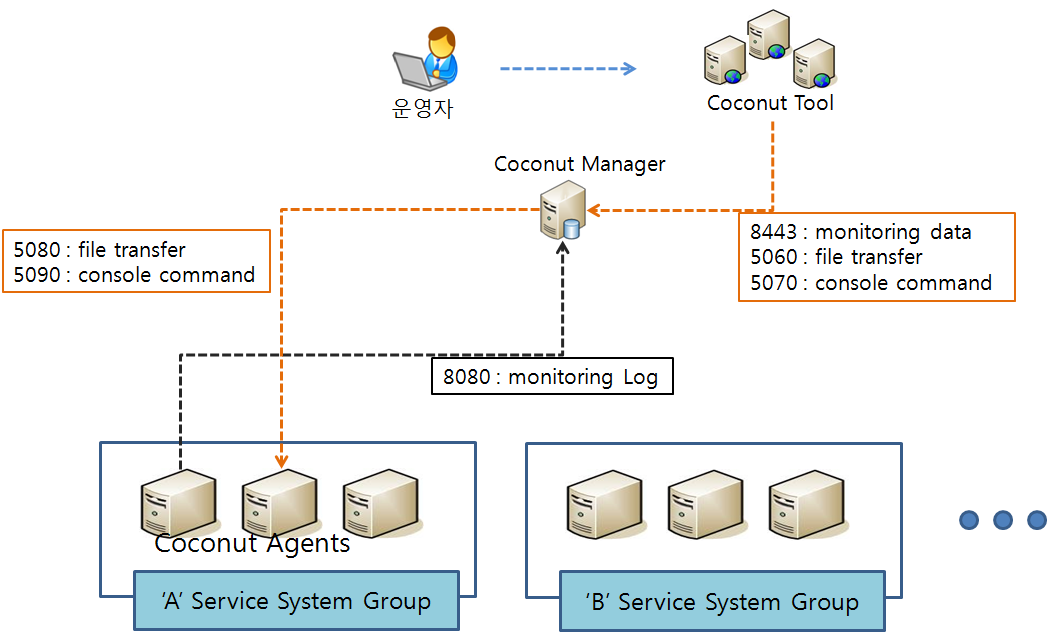
**손쉬운 원격 조작**

첫 화면에서 바로 특정 시스템으로 접속하여, 콘솔 화면으로 시스템을 조작할 수 있습니다. 지원된 콘솔 화면은 putty와 유사한 유틸리티들처럼 시스템이 직접 접근하는 것이 아닌, 중앙 서버를 통해 조작되는 시뮬레이팅 콘솔입니다. pwd, ls, cd 등의 주요 기능들을 제공하지만, vi와 같은 유틸리티를 실행하여 조작하는 아직 불가능합니다. 앞으로 지원하는 기능들을 지속적으로 추가할 예정입니다.

**Chapter 4. Architecture**

Coconut은 웹기반의 툴, 중앙관리를 위한 Manager, 그리고 각 시스템에서 정보를 전달하고 처리하는 Agent로 구성되어 있습니다.

시스템의 전체 구성은 아래 그림을 참고하세요.



(그림의 숫자와 내용은 각 포트와 해당 포트로 전송되는 데이터의 종류를 기술한 내용입니다.)

**Chapter 5. Cautions**

**Coconut**은 문서를 작성하는 현재, 보안에 취약한 상황입니다. 실 서비스에 적용시에는 항상 방화벽을 면밀히 설정하여 사용하길 바라며, 관리자의 비밀번호가 유출되지 않도록 주의하여 사용하시기 바랍니다.

앞으로 이 제품에 대한 기본적인 기능이 완료되면, 보안기능은 무엇보다 최우선으로 작업을 시작할 예정에 있습니다. 하지만 현재로서는 어느 시점인지를 약속할 수 없음을 양해 바랍니다.

**Chapter 6. Install**

Coconut을 사용하기 위해서는 사전에 설치해야 할 몇 가지 제품이 있습니다. 아래 내용을 참고하여 설치를 진행해 주시기 바랍니다.

**6.1. Required Packages**

* Redis (Manager에 각종 저장 정보를 기록하기 위함)
* httpd (Tool을 활용하기 위함)
* nodejs
* curl (Agent에 curl을 활용한 기능을 사용할 경우 필요함.)

**6.2. Install Redis**

현재 Manager와 Agent가 Redis로 접근할 때 pw없이 접근합니다. 참고하세요. Redis는 별도의 서버 혹은 Manager가 설치된 동일 서버에 설치하여도 문제없이 가동될 정도로 처리량이 많지 않습니다. docker를 사용하고 있다면 redis image를 다운받아 가동한다면 더욱 쉽게 설치하실 수 있습니다. Coconut은 Redis의 기본 포트를 사용하고 있으며, 원한다면 Coconut Manager 프로젝트에 포함된 "config.js"파일을 직접 수정하여 사용하실 수 있습니다.

**6.3. Installing Coconut Manager/Agent**

아래의 순서대로 진행하시기 바랍니다..

1. CoconutManager/Agent 폴더를 그대로 원하는 위치에 복사

2. CoconutManager/Agent 폴더에서 npm install 실행

3. config.js를 환경에 맞게 수정. 상세 설명은 코드 주석을 참고하세요.

(Coconut Agent에 포함된 config.js 파일과 동일하며, 해당 파일이 변경되면 Manager/Agent에도 동일하게 배포하여 사용해야 합니다.)

4. app\_mng.js를 실행

(\* 현재 사용중인 Agent들을 구별하기 위한 키는 해당 시스템의 IP(NIC 리스트 중 마지막 IP를 기준으로 결정)를 기반으로 하고 있습니다. 이는 하나의 물리적인 서버에 두 개의 agent가 존재할 수 없음을 이야기 합니다. 또한, 정상적인 관리를 위해서는 해당 IP로 접근이 가능하여야 합니다. 추후 이를 개선할 예정이지만 현재는 Agent를 설치할 때 이를 확실히 인지하고 있어야 합니다. 예를 들어 docker 이미지를 실행할 경우 도커는 이미지를 위해 가상의 NIC를 구축하여 내부에서만 사용하는 별도의 IP를 할당합니다. 이 IP로는 외부에서 접근할 수 없기 때문에 각별히 신경을 써야 합니다.)

**6.4. Installing Tool**

CoconutTool 폴더를 운영하는 httpd의 root 혹은 당신이 원하는 하위 폴더에 복사 하세요.

CoconutTool/ajax/ control\_box.html의 244line에 선언된 ipManager의 값을 설정한 Manager 주소로 변경하여 사용하시기 바랍니다.

**\* docker상에서 작동 시 주의 사항**

docker로 가동 시 가상 ip를 할당받게 되며, 이 ip는 외부에서 접근 불가능한 IP입니다. 현재 Coconut을 정상가동 하기 위해서는 docker run 시 --net=host 옵션을 사용하여 실행하시기 바랍니다. 이 옵션은 실행될 컨테이너의 eth0 interface에 host의 eth0가 매핑됩니다.

**Chapter 7. Monitoring**

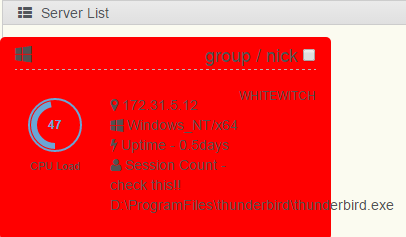
모니터링은 아래와 같은 종류를 지원합니다.

1) 시스템 모니터링

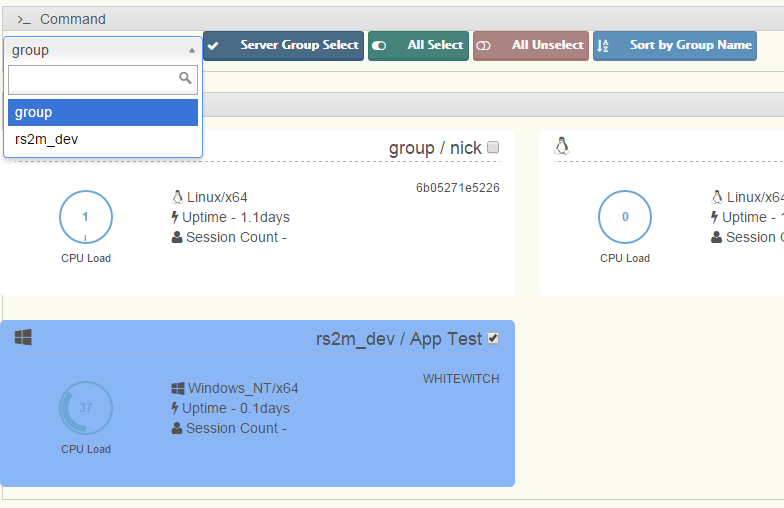
CPU Load, Uptime. (필요한 모니터링 요소를 추가할 예정. 12 Goal을 참조하세요.)

2) 어플리케이션 모니터링

지정한 프로세스의 정상 작동 여부. 이상이 발생하면 상태카드가 붉은색으로 변경되며, 이상이 발생한 프로세스 이름을 보여줍니다.



3) 시스템에 그룹/닉네임 지정 및 정렬 및 검색 지원

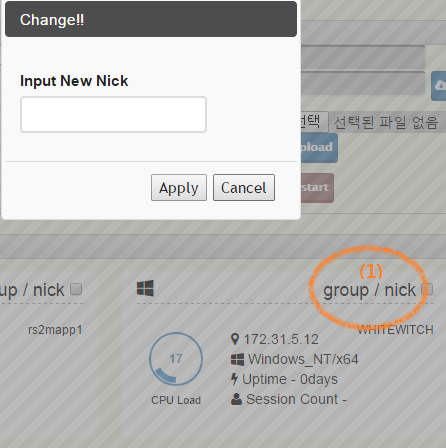


**Chapter 8. Management**

1) 그룹/ 닉네임 지정

**그림1**의 (1) 부분의 group 혹은 nick을 클릭하면 원하는 이름으로 변경할 수 있으며,

**그림 2**와 같이 반영된 결과를 볼 수 있다. 이후 그룹을 관리하는 창에서 **그림 3**처럼 그룹을 선택할 수 있게 됩니다.



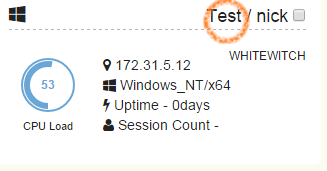


그림 2 - 변경된 그룹명

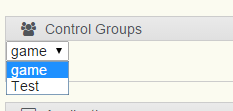


그림 3 - 등록된 그룹명

그림 1 - 그룹/닉 변경 다이얼로그

2) 관리할 어플리케이션 선정/해제

Agent를 설치한 시스템은 Manager 서버로 정보를 보내고 이를 제공된 Web 기반의 툴에서 볼 수 있습니다. Agent가 설치된 시스템은 아래와 같은 카드 형태의 정보창에 표시되는데, 상단의 기어 아이콘을 누르면 아래와 같은 메뉴를 볼 수 있습니다.

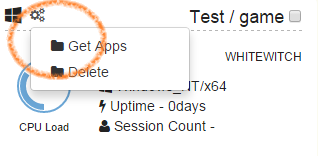


그림 4 - 시스템 카드에서 설정 옵션 선택

이 메뉴 중 "Get Apps"를 선택하면 해당 시스템에서 가동중인 모든 프로세스 리스트를 가져옵니다.

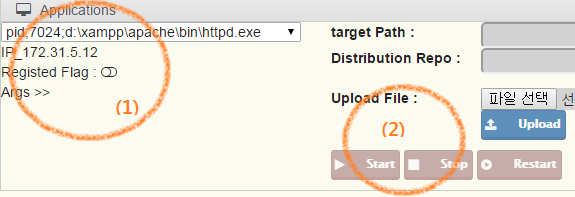


그림 5 - 어플리케이션 관리 화면

**그림 5**- (1)에서 콤보박스를 클릭하면 가동중인 모든 프로세스를 볼 수 있습니다.

**그림 5** -(2)에서는 아직 관리 대상이 아닌 프로세스 이므로 버튼들이 모두 비활성화 된 상태입니다.

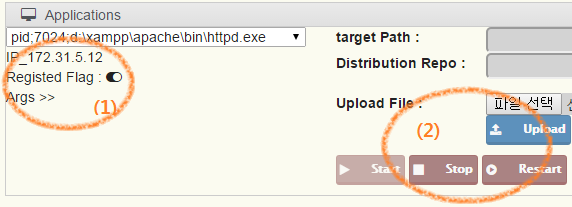


그림 6 - 특정 어플리케이션을 선정한 상태

**그림 6**에서 (1)번 영역을 클릭하면 선정/해제를 할 수 있으며, 선정할 경우 (2)의 영역에 해당 어플리케이션을 조정할 수 있는 버튼이 활성화 된 상태를 확인할 수 있습니다.

관리되는 어플리케이션의 경우 가동 시 설정한 옵션들을 그대로 적용하기 때문에, 별도로 옵션을 지정하지 않아도 정상적인 재시작이 가능합니다.

**\* 주의사항** : 어플리케이션을 가동 시 실행파일이 존재하는 전체 경로를 적어 가동하는 것을 권장합니다. 만약, 해당 실행 파일이 다른 경로에서 실행 시 경로 없이 인식할 수 없다면 정상 가동이 안될 수 있습니다.

3) 그룹별 검색

당신이 지정한 그룹별로 모니터 혹은 조작할 할 서버 리스트를 추릴 수 있습니다. **그림 3**을 참고하세요.

**Chapter 9. Control**

chapter 9에서 본, 관리할 어플리케이션을 등록하고, 시작/정지/재시작 버튼은 이곳에서 생략하겠습니다. UI를 지원하니 간단히 테스트 해보시면 됩니다.

이 장에서는 시스템 카드를 관리하고, 실제 해당 서버에 터미널로 접속하여 시스템을 조작하는 방법을 설명하겠습니다.

1) 시스템 카드

시스템 카드는 설치된 Agent 시스템을 기준으로 자동으로 등록됩니다. 설치된 Agent는 Management 서버로 정보를 전송하고, 이를 제공된 웹 기반 관리툴에서 확인이 가능합니다.

**그림 4**의 기어 모양의 버튼으로 등록된 카드를 삭제할 수 있지만, Agent가 가동되고 있는 서버라면 자동으로 재등록이 진행됩니다. 관리 대상이 아니라면 Agent를 삭제하는 것이 좋습니다.

2) 터미널 사용 방법

전통적으로는 한 시스템 당 하나의 (putty같은) 유틸리티를 사용하여 1:1로 접속하는 방식이 있었습니다. 현대는 하나의 서비스를 위해 수십대의 서버를 관리해야 합니다. 단순히 모니터링이 아닌 서비스 패치, 서비스 어플리케이션 조작, 환경의 변화에 따른 설정 등을 짧은 시간에, 그리고 오류없이 진행해야 합니다. 이는 전통적인 방법에서는 오류를 범할 많은 위험요소가 있었으며, 많은 인력과 비용이 들어갑니다.

**Coconut**에서는 이런 비용을 최소화 하기 위해 여러가지 기능을 지원하며, 그 중 지금 소개하는 터미널은 당신에게 분명히 반가운 기능일 것입니다.

**그림 7**에 보시는 것처럼, 제공되는 화면은 많은 개발자들에게 익숙한 화면일 것입니다.

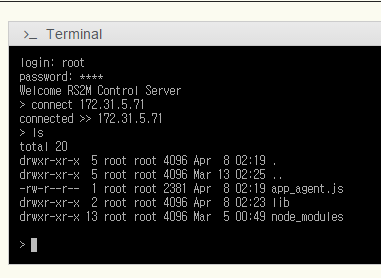


그림 7 - 터미널에서 특정 서버로 접속한 화면

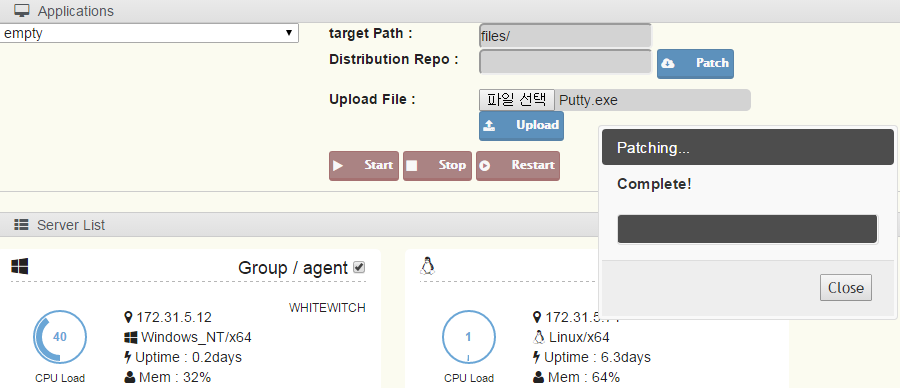
등록된 계정과 패스를 입력한 후, connect "접속하고자 하는 서버 IP"를 입력하면 특정 서버로 연결이 되고, 이후 ls, dir, curl 등의 명령어를 사용할 수 있습니다. 이 기능은 사실 완벽한 bash 환경을 제공하는 것이 아니, 가상으로 작동합니다. 즉, 해당 시스템에 직접 연결하여 명령을 전달하는 것이 아닌, Manager를 통해 Agent로 명령을 전달하도록 설계되어 있습니다.

이는 우리가 필요한 명령어를 쉽게 추가할 수 있으며, 불필요하거나 원하지 않는 기능들을 손쉽게 제거할 수도 있었습니다. 또한, 특정 기능을 제한적으로 사용하도록 수정할 수도 있습니다.

터미널에서 사용 가능한 명령어들은 이어지는 10장에서 상세히 다루겠습니다.

3) 파일 전송 기능

(\* 현재 지정한 저장소에서 파일을 다운받는 기능은 제공되지 않고 있습니다. 5월 중에 해당 기능이 추가되어 Github에 업데이트 될 예정입니다.)



파일을 전송하기 위해 세 가지 세팅이 필요합니다.

1) target path : 파일이 Agent에 저장될 때 어떤 경로에 저장될지를 지정할 수 있습니다. 비어 있다면 agent가 설치된 경로에 저장됩니다.

2) Upload File : Agent로 보낼 파일을 선택하세요. (추후 압축 파일일 경우 자동으로 풀어주는 옵션을 추가할 예정입니다.) 전송할 파일의 최대 용량은 50MB로 제한되어 있으며, 원한다면 control\_box.html 내의 스크립트 코드를 수정하여 적용할 수 있습니다.

3) 하단의 Server List탭에서 파일을 전송할 서버를 선택하세요. 한 개 이상 선택이 되어야 합니다.

파일 전송 시, 현재는 Manager 서버로 보낼 때의 상태값만 다이얼로그를 통해 볼 수 있습니다. 이후는 Manager에서 Agent로 보내는 과정을 볼 수 없으며, 이는 추후 개선될 예정입니다.

**Chapter 10. Commands**

이 장에서는 **Coconut**에서 제공하는 Terminal에서 활용할 수 있는 Commands들에 대해서 상세히 논의하겠습니다. 이 기능은 JQueryTerminal(http://terminal.jcubic.pl/)을 사용해 제작되었습니다. http를 통해 management로 전달되며, 이를 다시 Agent로 전달하여 결과를 다시 front로 전달합니다.

모든 명령어는 windows와 linux 모두를 지원합니다. 예를 들어 ls와 dir을 동시에 지원하며, 어느 시스템에서든 두 명령어를 같은 목적으로 사용할 수 있습니다.

이제 종류별로 각 기능들을 알아보겠습니다.

1) 특수 명령어

이 명령어들은 **Coconut**의 Terminal을 컨트롤 하기 위한 특수 기능들입니다.

|  |  |  |
| --- | --- | --- |
| 명령어 | 파라미터 | 상세 설명 |
| clear | - | 터미널 화면을 클리어합니다. login 화면에서도 입력됩니다.(JQueryTerminal 기본 기능) |
| whoru | - | 현재 자신이 어떤 계정으로, 어느 서버와 연결되어 있는지 보여줍니다. |
| connect | ipaddress | 지정한 서버와 매칭을 시켜줍니다. |

2) 기본 명령어

이 명령들은 bash 혹은 windows의 prompt에서 지원하는 기본 명령어들과 유사 혹은 동일하게 지원됩니다.

|  |  |  |
| --- | --- | --- |
| 명령어 | 파라미터 | 상세 설명 |
| pwd | - | 현재 디렉토리의 위치를 알려줍니다. |
| ls/dir | 경로 (생략가능) | 현재 디렉토리의 파일, 디렉토리를 보여줍니다 |
| cd | 경로 | 지정한 위치로 디렉토리를 변경합니다. |
| mkdir | 이름 | 지정한 이름으로 디렉토리를 생성합니다. |
| rm | 이름 | 지정한 이름의 파일을 삭제합니다. |
| mkdir | 이름 | 지정한 이름의 디렉토리를 삭제합니다. |
| tail | (기존옵션) | 지정한 파일의 마지막 몇 줄을 볼 수 있다. |
| find | (기존옵션) | 원하는 특정 파일, 디렉토리를 탐색하여 찾을 수 있다. |

3) 유틸리티

외부 유틸리티를 지원합니다.

|  |  |
| --- | --- |
| 명령어 | 상세 설명 |
| Curl | curl 지원 |
| svn | Svn 지원 |

**Chapter 11. Goal**

\* 기능 작업들

1. 파일 업로드에 대한 error 정보를 front로 전송, 표기하기
2. front에서 system card를 확장하여 모니터링 정보를 상세히 볼 수 있는 기능 디자인 및 기능 추가
3. front에서 system card에 network traffic 정보 추가
4. memory, disk, network 모니터링 정보를 각 코어별로 상세히 볼 수 있는 기능 추가
5. 관리 대상인 process별 cpu/memory 사용률 모니터링 기능 추가
6. log 검색 혹은 다운로드 기능
7. 지정한 log 필터링 기능(?)
8. https로 모두 교체
9. RPC를 ssh로 연결하도록 수정
10. console 창에서 특정 디렉토리 접근 제한 기능
11. 시스템 별 고유 인덱스 생성 방법 변경

\* 이후 작업들

1. 플러그인 시스템 지원
2. 서비스 가동에 대한 순서 정리(?) - 스케쥴러로 해결하는 방법 고민
3. 스케쥴 기능 추가
4. node-twitter 연결해서 각종 메세지 관리자에게 보내기
5. docker의 유동 ip 문제 해결 방법 고민 (고유 인덱스 문제가 일부 중첩됨.)
6. 자신의 시스템이 docker인지 여부 확인
7. aws ,azura 플래그 추가

**Chapter 12. Help**

Github 포럼에서 의견 받음.

**Chapter 13. Reference**

Front-end에 사용된 기술

1. JQuery : https://jquery.com/
2. JQuery-UI : https://jqueryui.com/
3. bootstrap : http://getbootstrap.com/
4. DevOOPS v2 : http://devoops.me/handmade/3/
5. JQueryTerminal : http://terminal.jcubic.pl/
6. dterm : http://decimus.net/dterm
7. socket.io : http://socket.io/

Management Server

1. nodejs : https://nodejs.org/
2. express : http://expressjs.com/
3. body-parser : https://www.npmjs.com/package/body-parser
4. node-schedule : https://www.npmjs.com/package/node-schedule
5. redis npm : https://www.npmjs.com/package/redis
6. json-rpc2 : https://www.npmjs.com/package/json-rpc2
7. MD5 npm : https://www.npmjs.com/package/MD

Agent

1. nodejs : https://nodejs.org/
2. express : http://expressjs.com/
3. body-parser : https://www.npmjs.com/package/body-parser
4. node-schedule : https://www.npmjs.com/package/node-schedule
5. redis npm : https://www.npmjs.com/package/redis
6. json-rpc2 : https://www.npmjs.com/package/json-rpc2
7. MD5 npm : https://www.npmjs.com/package/MD5
8. ps-node : https://www.npmjs.com/package/ps-node
9. windows-cpu : https://www.npmjs.com/package/windows-cpu