



RaspberryPi 와 클라우드 환경 을 이용한 실습실 설계 및 구현

201311263 김민환
201310173 이종하

- 2019 종합설계1 -

Index



I. 소개



II. 설계



III. 구현



VI. 시연



V. 발전방향

I. 소개

1. 요약

I. 소개

II. 설계

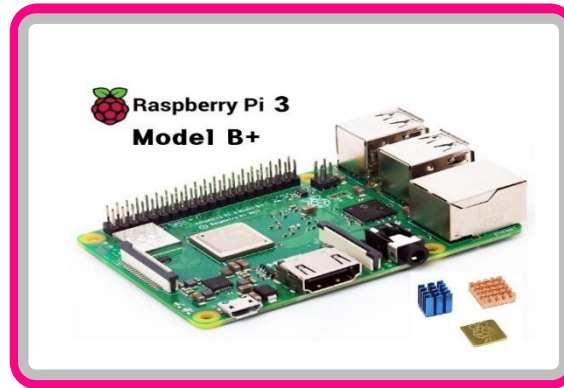
III. 구현

IV. 시연

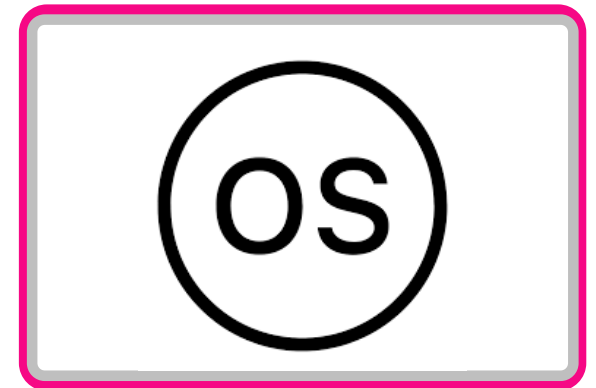
V. 발전방향



[Cloud Infra]



[Raspberry Pi]



[OS]

- OpenStack을 활용하여 Cloud Infra를 구성
- Raspberry Pi를 Desktop을 대체할 장비로 활용
- RDP프로토콜을 활용하여 Windows 와 Linux 데스크탑 환경을 자유롭게

2. 문제점 분석

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

[하드웨어]



[컴퓨터 실습실]



[OS 및 소프트웨어]



2. 문제점 분석

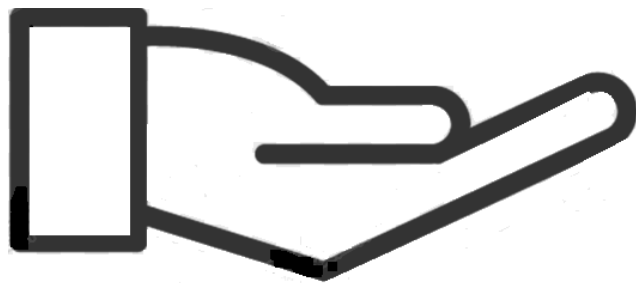
I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[수작업을 통한 관리]

- 실수 발생 가능성 존재
- 완벽히 같은 환경 구성의 어려움
- 시간과 비용의 소요 큼



[가상화를 통한 자동 관리]

- 실수 발생 가능성 차단
- 완벽히 같은 환경 구성 가능
- 시간과 비용의 소요 적음

3. 해결 방안

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

Virtual Desktop Infrastructure (VDI)



3. 해결 방안

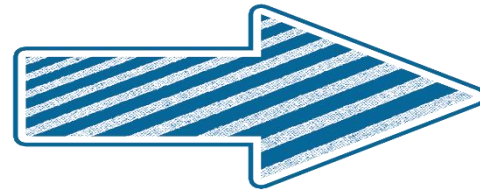
I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



활용



3. 해결 방안

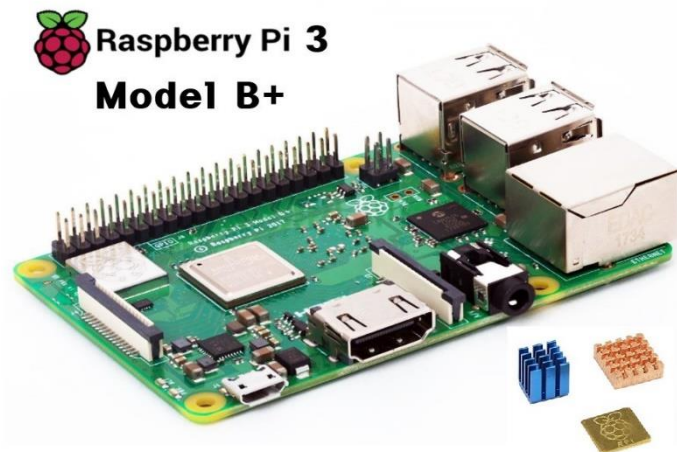
I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[ThinClient]



[RDP Server]

4. 이점

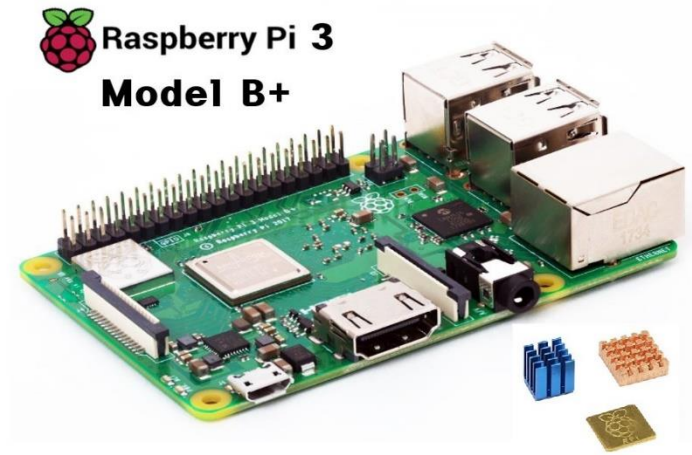
I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[선택의 이유]

- 생산 단가를 낮출 수 있음
- 좋은 품질의 VDI 제공

4. 이점

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

	본 과제	타 기업	일반
Quick Power On	○	X	X
Image 배포	○	X	X
SW 관리	○	X	X
Client 관리	○	○	X
SW 비용	Low	Middle	Low
HW 비용	Low	Middle	Low
전력소모	Low	Middle	Low

프. 설계

1. 시나리오

I. 소개

II. 설계

III. 구현

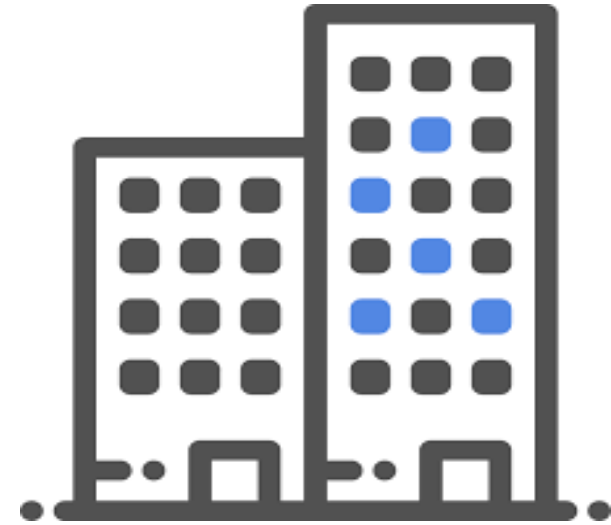
IV. 시연

V. 발전방향



[건국대학교 컴퓨터 실습실]

- VDI 솔루션 구축을 원함

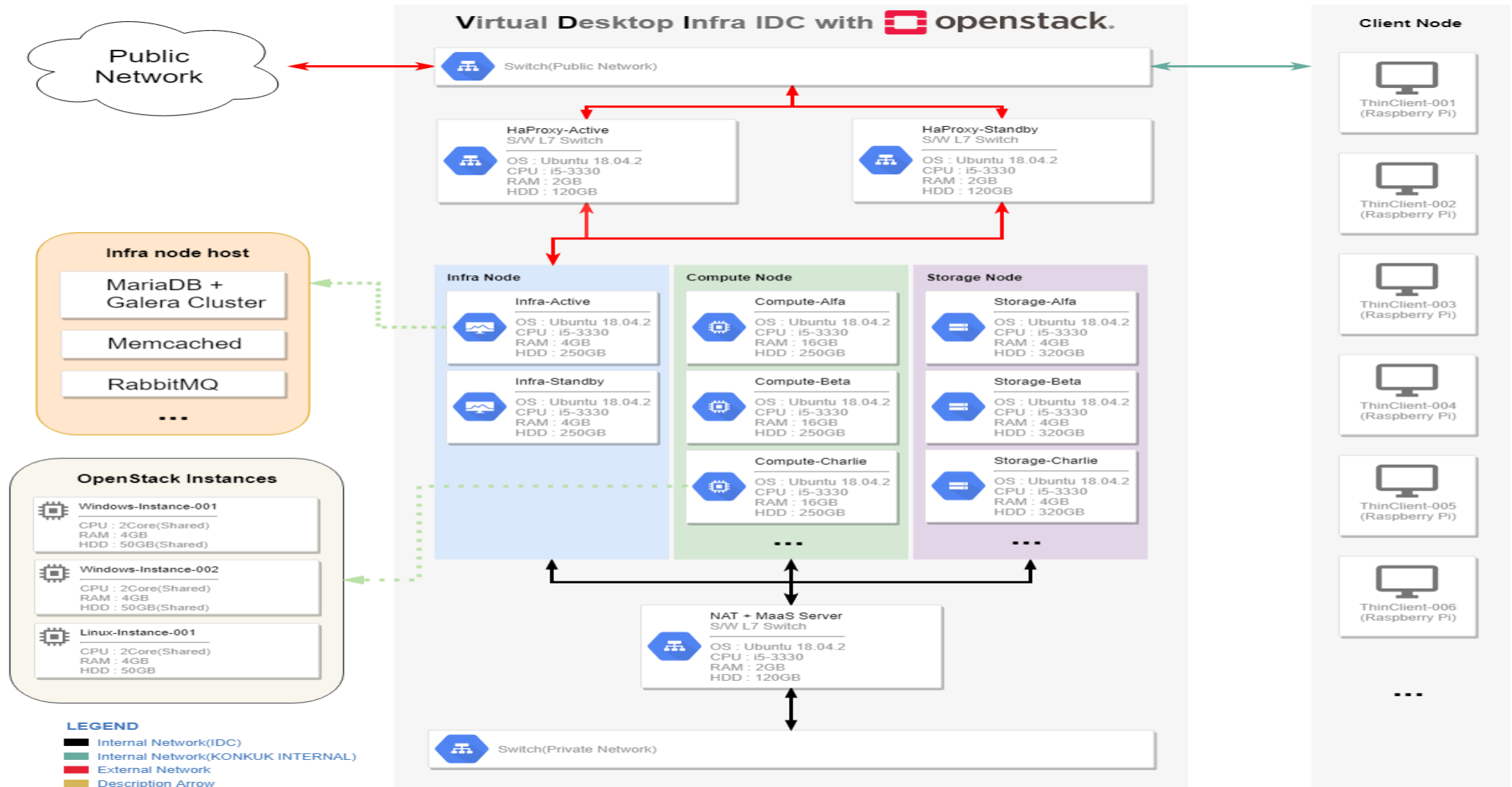


[A업체]

- VDI 솔루션 구축을 진행
- 업체별로 Lisence 정보 관리

2. 건국대학교 서버실 설계

Full VDI 설계도 (Prototype Version)



[건국대학교 신 공학관 서버 설계도]

2. 건국대학교 서버실 설계

MAAS

Machines

Devices

Controllers

Pods

Images

DNS

AZs

Subnets

Settings

admin

Logout

Machines 34 machines available

Add hardware

34 Machines 1 Resource pool

Filter by

Status +

Owner +

Resource pools +

Architectures +

OS/Release +

Tags +

Storage Tags +

Subnets +

Fabrics +

Zones +

Search machines

☐

FQDN

▼

MAC

Power

Status

Owner

Cores

RAM (GiB)

Disks

Storage (GB)

☐

awake-shrimp.maas

Unknown

New

0

0.0

0

0.0

☐

aware-cougar.maas

Unknown

Ubuntu 18.04 LTS

admin

16

8.0

1

250.1

☐

aware-swan.maas

Unknown

Ubuntu 18.04 LTS

admin

4

3.0

2

2320.5

☐

bold-insect.maas

Unknown

Ubuntu 18.04 LTS

admin

16

16.0

1

250.1

☐

bright-colt.maas

Unknown

⚠ Ubuntu 18.04 LTS

admin

4

3.0

⚠ 1

320.1

☐

close-kit.maas

Unknown

New

0

0.0

0

0.0

☐

cool-baboon.maas

Unknown

⚠ Ubuntu 18.04 LTS

admin

4

4.0

⚠ 2

2250.5

☐

crack-jackal.maas

Unknown

New

0

0.0

0

0.0

☐

daring-fly.maas

Unknown

Ubuntu 18.04 LTS

admin

8

8.0

1

320.1

☐

deep-trout.maas

Unknown

New

0

0.0

0

0.0

☐

enough-shark.maas

Unknown

Ubuntu 18.04 LTS

admin

8

32.0

1

2000.4

☐

equal-tomcat.maas

Unknown

New

0

0.0

0

0.0

☐

fleet-feline.maas

Unknown

Ubuntu 18.04 LTS

admin

8

8.0

1

320.1

☐

gentle-lynx.maas

Unknown

New

0

0.0

0

0.0

☐

grand-quagga.maas

Unknown

Ubuntu 18.04 LTS

admin

8

28.0

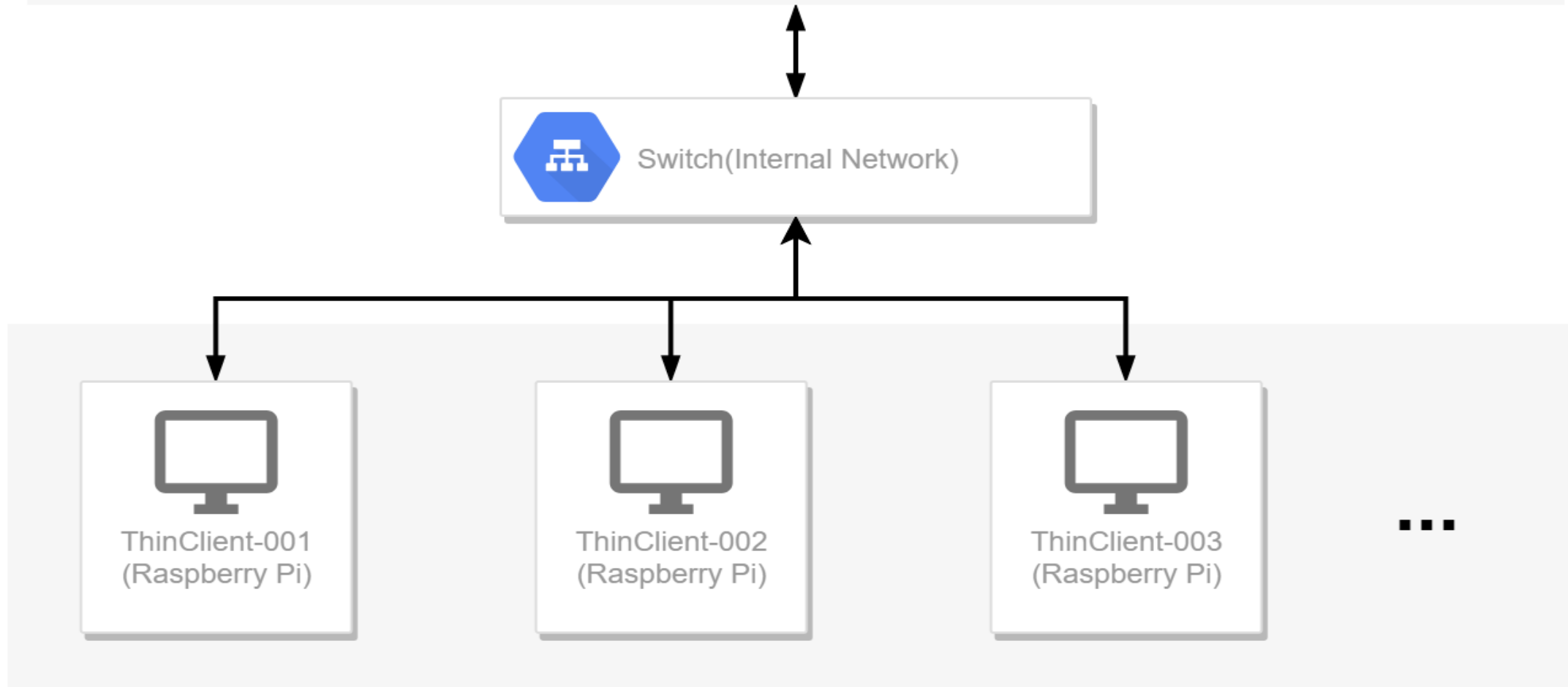
1

1000.2

[Ubuntu maas 관리자 화면]

3. 건국대학교 실습실 설계

Virtual Desktop Infra IDC with  openstack.®



[Client 구성 관리자 화면]

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

4. A업체 라이선스 관리 인프라 설계

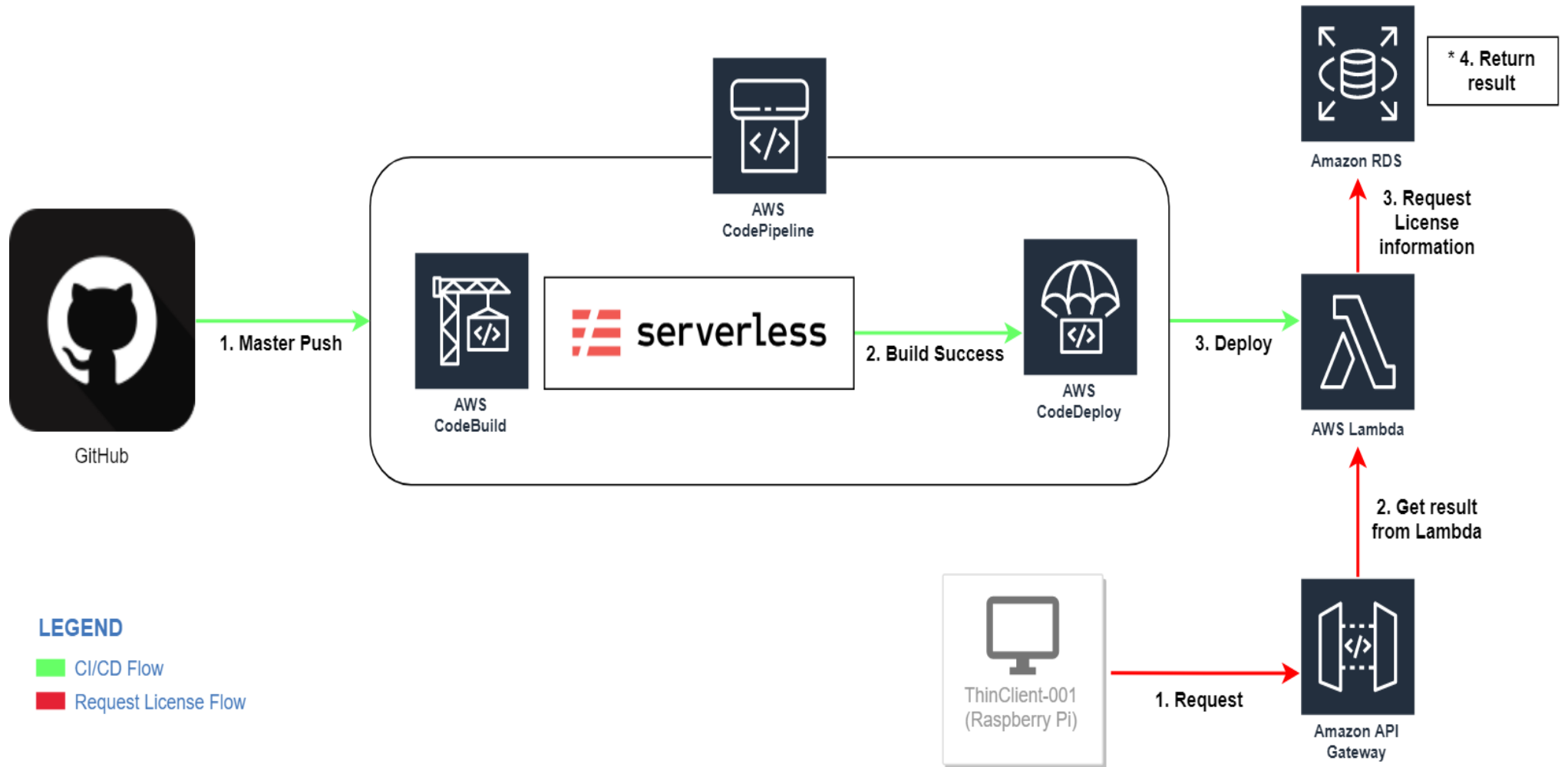
I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[AWS를 이용한 Serverless 설계도]

4. A업체 라이선스 관리 인프라 설계

I. 소개


II. 설계

III. 구현

IV. 시연







V. 발전방향



Branch: master ▾ New pull request Create new file Upload files Find File Clone or download ▾

 nerdkim

Client : start 진행 시 Internet connection 1초 단위로 체크

Latest commit 79390b8 1 hour ago

 _legacy_	Client 버그 패치	16 hours ago
 admin	Client 대규모 패치(for Raspi OS)	18 hours ago
 aws	Patch AWS-Client bug	10 hours ago
 client	Client : start 진행 시 Internet connection 1초 단위로 체크	1 hour ago
 xenserver	AWS 인프라 수정으로 인한 개선	12 hours ago
 README.md	AWS 인프라 수정으로 인한 개선	12 hours ago

 README.md 

Welcome to Tiny VDI project GitHub!

효율적인 Virtual Desktop Infra 구축을 위한 프로젝트 입니다.

```
git clone https://github.com/nerdkim/vdi
```

[GitHub 화면]

4. A업체 라이선스 관리 인프라 설계

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[SE][Linux]OpenStack에서 image upload가 되지 않을 때 조치 방법

Edit New Page

NerdKim edited this page 8 days ago · 1 revision

```
Access to XMLHttpRequest at 'https://openstack.konkuk.ac.kr:9292/v2/images/e2784757-2b57-419a'
```

OpenStack web에서 image(OS file) upload가 되지 않을 때, openstack-cli(openstack-client)를 이용하여 API를 통해 업로드 진행한다.

- OpenStack-Client : <https://github.com/openstack/python-openstackclient>

이 때, openstack-cli 명령 실행 시, insecure를 추가하여 SSL 인증을 진행하지 않는다.

```
# OpenStack image create with insecure SSL
openstack image create "Windows-10-Pro-PreBuilt" --disk-format qcow2 --container-format bare
```

※참고 : OpenStack 셋팅 시 다음과 같이 환경변수 설정을 진행해야 한다(Wiki에서 확인 가능).

+ Add a custom footer

Pages 11

Find a Page...

Home

[\[SE\]\[Linux\]Let's Encrypt SSL 생성](#)

[\[SE\]\[Linux\]OpenStack 전용 Windows image\(OS\) 생성 방법](#)

[\[SE\]\[Linux\]OpenStack 환경변수](#)

[\[SE\]\[Linux\]OpenStack에서 image upload가 되지 않을 때 조치 방법](#)

[\[SE\]서비스 최고 관리 정보](#)

[\[Windows\]Chrome을 insecure mode로 실행하기](#)

[Curl to AWS API Gateway](#)

[GitHub credential](#)

[vdi admin 계정 정보](#)

[Windows driver name 변경](#)

[GitHub Wiki 화면]

4. A업체 라이선스 관리 인프라 설계

vdi-license-server-production

삭제

업데이트

스택 작업 ▼

스택 정보

이벤트

리소스

출력

파라미터

템플릿

변경 세트

리소스 (10)

Q 리소스 검색

논리적 ID ▲	물리적 ID ▼	유형 ▼	상태 ▼	상태 사유 ▼
ApiGatewayDeployment1561220558799	mj0pz7	AWS::ApiGateway::Deployment	✔ CREATE_COMPLETE	-
ApiGatewayMethodLicenseGet	vdi-l-ApiGa-CXKAO9S7OTWV	AWS::ApiGateway::Method	✔ CREATE_COMPLETE	-
ApiGatewayResourceLicense	wt2ox8	AWS::ApiGateway::Resource	✔ CREATE_COMPLETE	-
ApiGatewayRestApi	9nwcq9gplk 🔗	AWS::ApiGateway::RestApi	✔ CREATE_COMPLETE	-
TestLambdaFunction	vdi-license-server-production 🔗	AWS::Lambda::Function	✔ UPDATE_COMPLETE	-
LambdaPermissionApiGateway	vdi-license-server-production-LambdaPermissionApiGateway-1WCCDFYPI54JW	AWS::Lambda::Permission	✔ CREATE_COMPLETE	-
HelloWorldLambdaVersionLlcwSRktFRobLuq2znJWCMAw7t4J58BUFiH8OPNyOc	arn:aws:lambda:ap-northeast-2:989565025424:function:vdi-license-server-production-helloWorld:9	AWS::Lambda::Version	✔ CREATE_COMPLETE	-
HelloWorldLogGroup	/aws/lambda/vdi-license-server-	AWS::Logs::LogGroup	✔ CREATE_COMPLETE	-

[AWS CloudFormation 화면]

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

III. 구현

1. 기존 자원

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[건국대학교 DMS 랩실]

2. 장비 조립 · 점검 · 교체

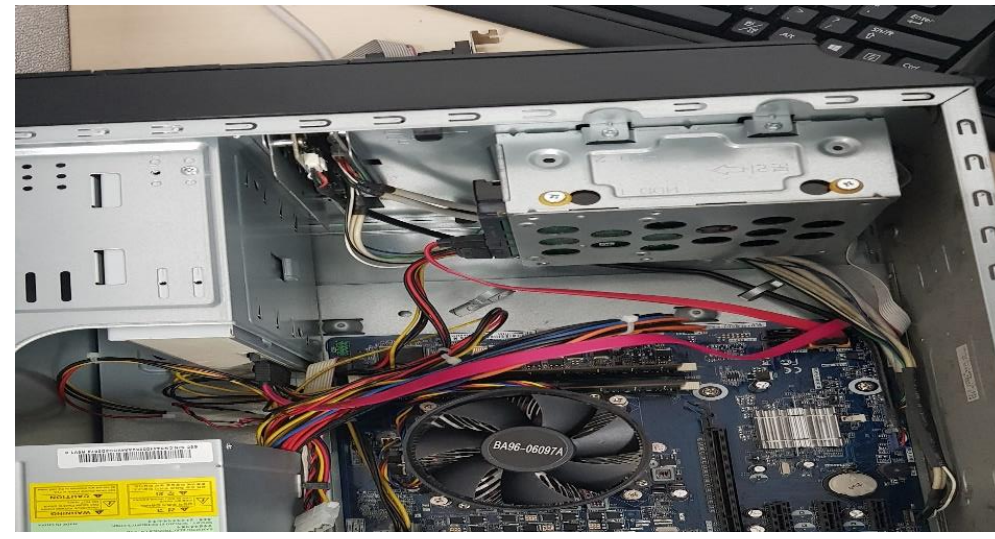
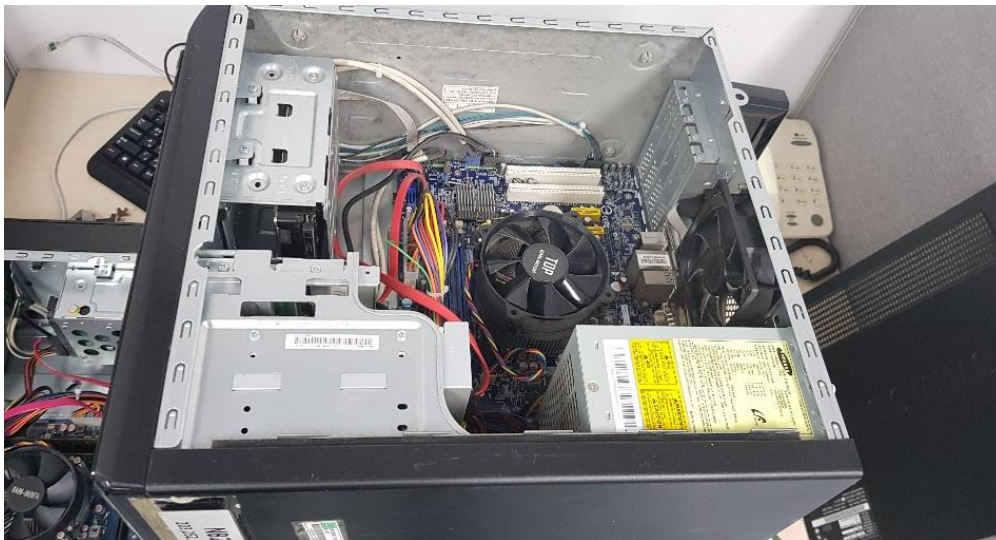
I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



3. 서버실 설계

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[Rack1: Proxy, Infra, Nat/MAAS, HaProxy서버 | Rack2: Storage, Compute node]

3. 서버실 설계

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[Rack1]

3. 서버실 설계

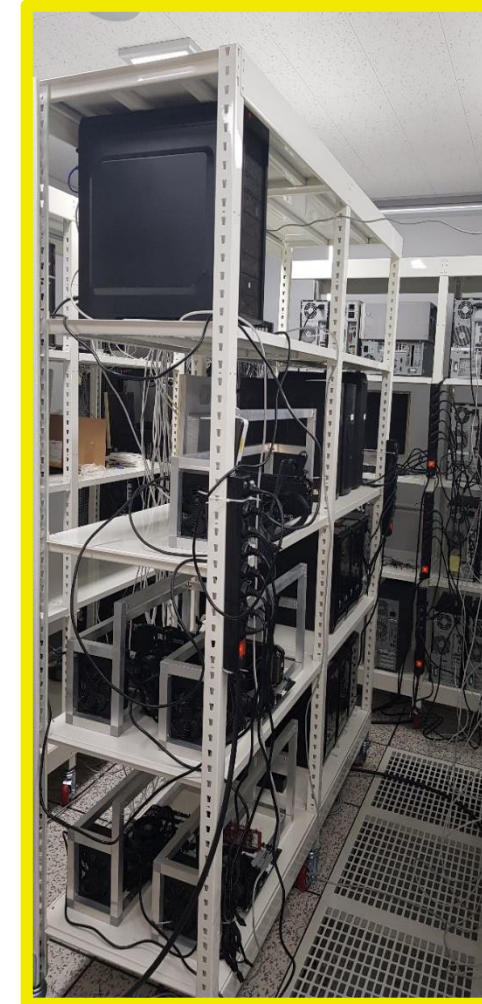
I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[Rack2]

IV. 시연

1. ThinClient 완제품

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



2. ThinClient 포트 연결

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



3. 가상PC환경 접속

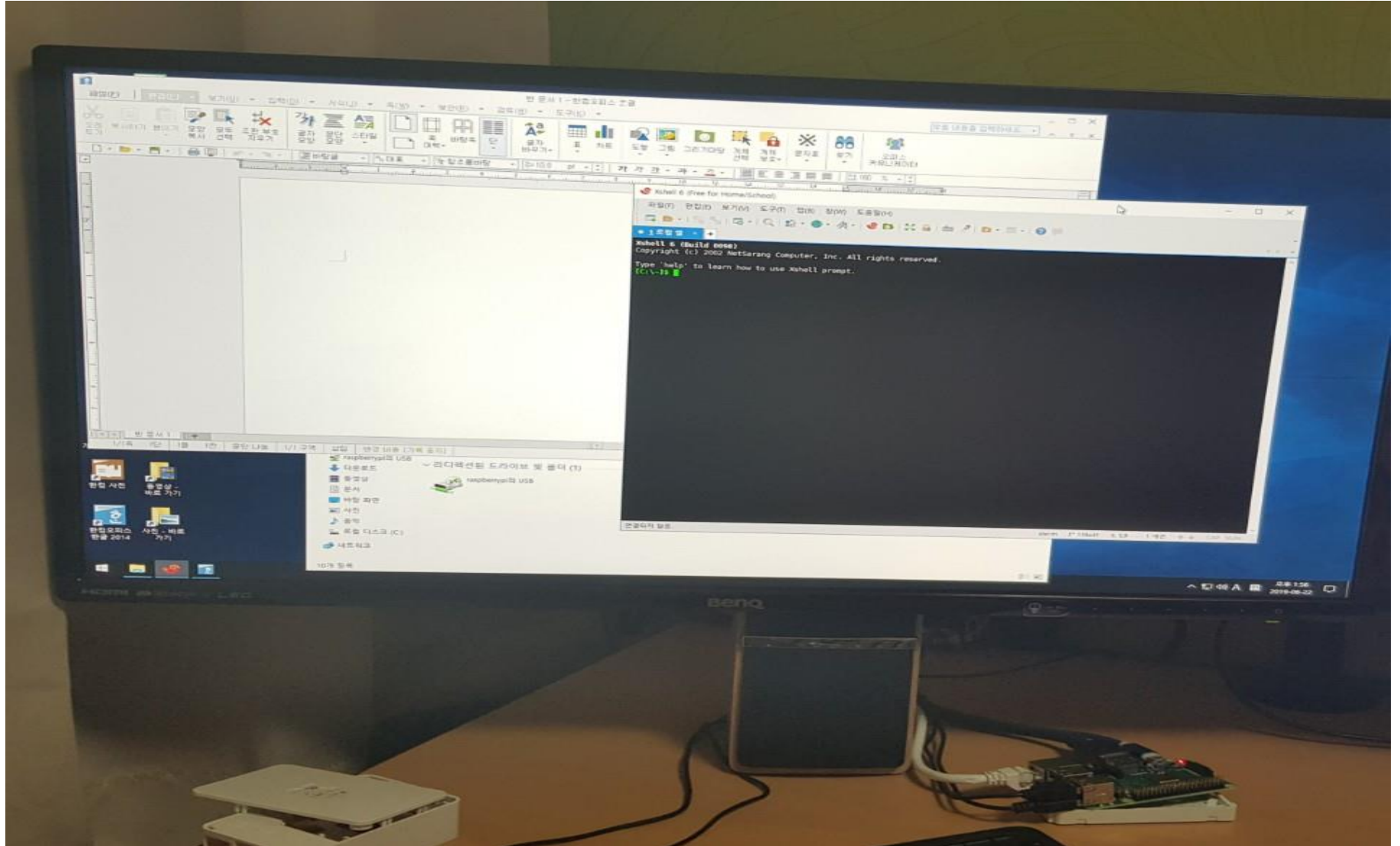
I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



V. 발전방향

1. 현재의 한계점

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

항목	값
OS 이름	Microsoft Windows 7 Enterprise K
버전	6.1.7601 Service Pack 1 빌드 7601
OS 추가 설명	사용할 수 없음
OS 제조업체	Microsoft Corporation
시스템 이름	NH03
시스템 제조업체	LG Electronics
시스템 모델	B70CV.AT558KE
시스템 종류	x64-based PC
프로세서	Intel(R) Core(TM) i5-4590 CPU @ 3.30GHz, 3301Mhz, 4 코어, 4 논리 프로세서
BIOS 버전/날짜	American Megatrends Inc. CVF07, 2015-04-27
SMBIOS 버전	2.7
Windows 디렉터리	C:\Windows
시스템 디렉터리	C:\Windows\system32
부팅 장치	#Device#HarddiskVolume1
로캘	대한민국
하드웨어 추상화 계층	버전 = "6.1.7601.24354"
사용자 이름	NH03\NH402
표준 시간대	대한민국 표준시
설치된 실제 메모리(RAM)	4.00GB
총 실제 메모리	3.92GB
사용 가능한 실제 메모리	2.66GB
총 가상 메모리	7.84GB
사용 가능한 가상 메모리	6.42GB
페이지 파일 공간	3.92GB
페이지 파일	C:\pagefile.sys

☒ 범주 이름만 검색(R)

[건국대학교 구형 실습실 컴퓨터 SPEC]

1. 현재의 한계점

○ 해결방안 -> 교체



[SSD]



[고사양 서버]

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

2. Client 개선 사항

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향

저사양의
Single Board
Computer
(단가 인하)



[BananaPi]



[Asus Tinker Board]

가벼운 OS
Embedded OS
(성능 향상)



[TinyCore OS]



[Zero Client]

3. Server 개선 사항

I. 소개

II. 설계

III. 구현

IV. 시연

V. 발전방향



[Xen Hypervisor]



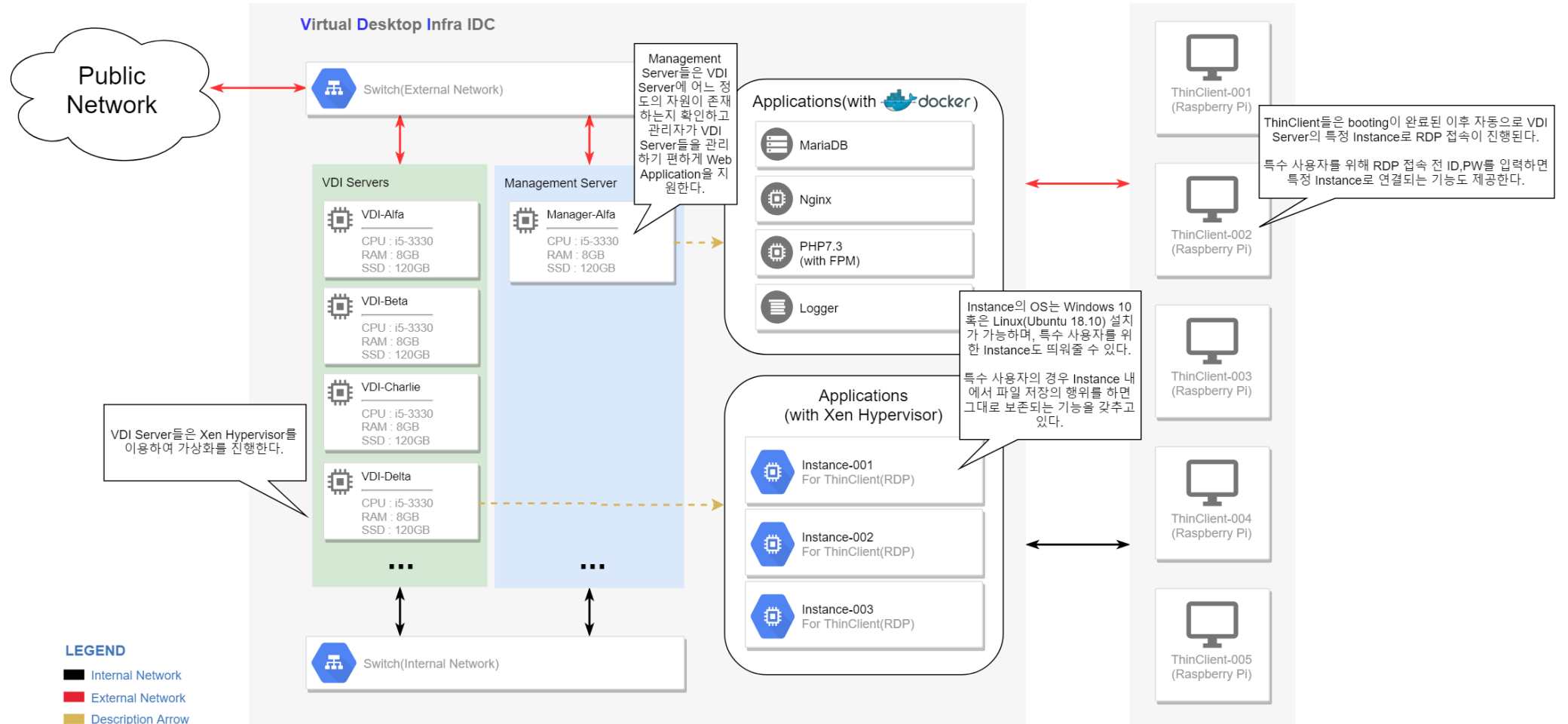
활용



[Web management service]

3. Server 개선 사항

Compact VDI 설계도 (Prototype Version)



[전체적인 프로젝트 설계도]



THANK YOU