
KGcinama

서버 구축 기술문서

K G C I N E M A

Contents

01. 사업개요

| | |
|-------------|------|
| 01. 사업명 | 05 P |
| 02. 사업 목적 | 05 P |
| 03. 요청사항 | 05 P |
| 04. 홈페이지 구성 | 06 P |

02. 네트워크 토플로지

| | |
|-----------------------|------|
| 01. 네트워크 논리적 구성도 | 08 P |
| 02. 네트워크 ,서버 장비 IP 할당 | 09 P |
| 03. 네트워크 설정 | 11 P |

03. 서버 구축

| | |
|----------------|------|
| 01. Ansible 구축 | 39 P |
| 02. DNS 구축 | 44 P |
| 03. PROXY 구축 | 49 P |
| 04. WEB 구축 | 64 P |
| 05. WAS 구축 | 66 P |
| 06. DB 구축 | 69 P |

Contents

04. 관리 도구

| | |
|------------------------|------|
| 01. 모니터링 도구(가비아) | 76 P |
| 02. 로그 관리 도구(goaccess) | 79 P |

05. 보안

| | |
|----------------------------|------|
| 03. Ansible을 이용한 서버 취약점 검사 | 84 P |
|----------------------------|------|

06. 검증

| | |
|-------------------|------|
| 01. PROXY VRRP 검증 | 91 P |
| 02. SSL/TLS 설정 검증 | 92 P |
| 03. WEB 검증 | 93 P |
| 04. DB 검증 | 95 P |

K G C I N E M A

사 업 개 요

01

01. 사업명 “KGcinema 서버 구축 및 운영 사업”

02. 사업 목적

INFRASTRUCTURE ENGINEER



KG cinema로부터 서버 구축 및 운영을 의뢰를 받았다. 2020년부터 코로나로 인해 인터넷 예매를 통한 영화 예매를 하는 사람들이 많이 줄어 들어 서버를 작게 구성했는데 정부가 코로나 종결을 2022년 4월부터 실행한다는 발표를 통해 KG cinema의 홈페이지에 접속 수의 증가와 홈페이지 예매를 통한 트래픽 유입이 증가할거라 예상해서 KG cinema의 자체 장비로는 원활한 서비스가 불가능하다 생각하여 서버 구축 및 운영을 의뢰했다.

03. 요구 사항 Requirements



통신성



안정성



보안성

서버에 갑작스러운
과도한 트래픽이 일어
나도 트래픽을 수용
할수 있는 서버 구성

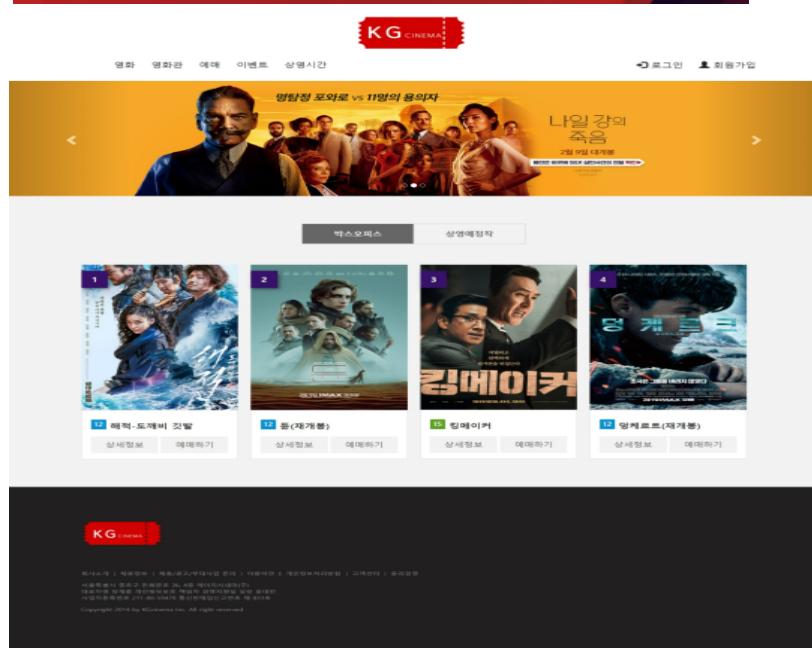
이중화를 통해
네트워크의 안정성을
확보하여 원활한
시스템 서비스 지원

유저의 로그 관리를
통해 보안 강화

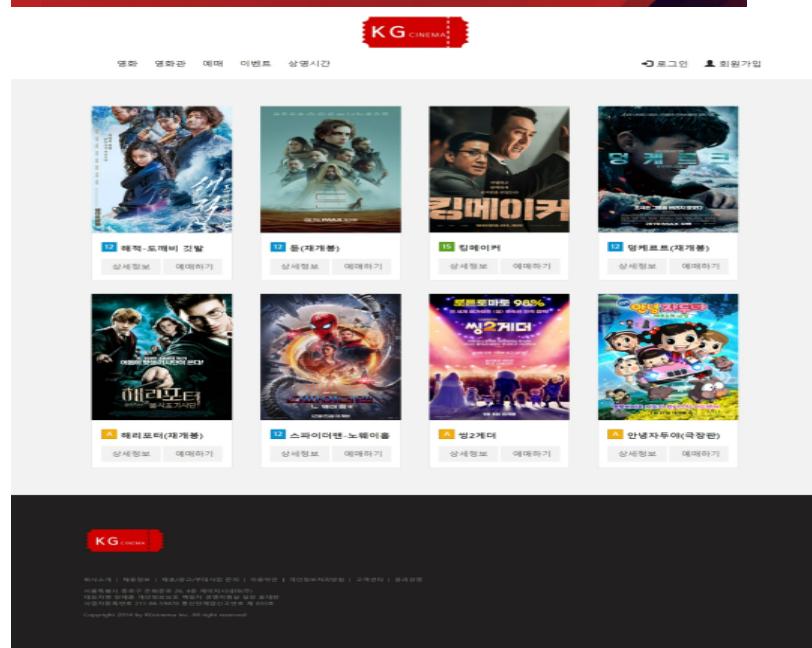
04. 홈페이지 구성

◆ URL : “ www.kgcinema1.com ”

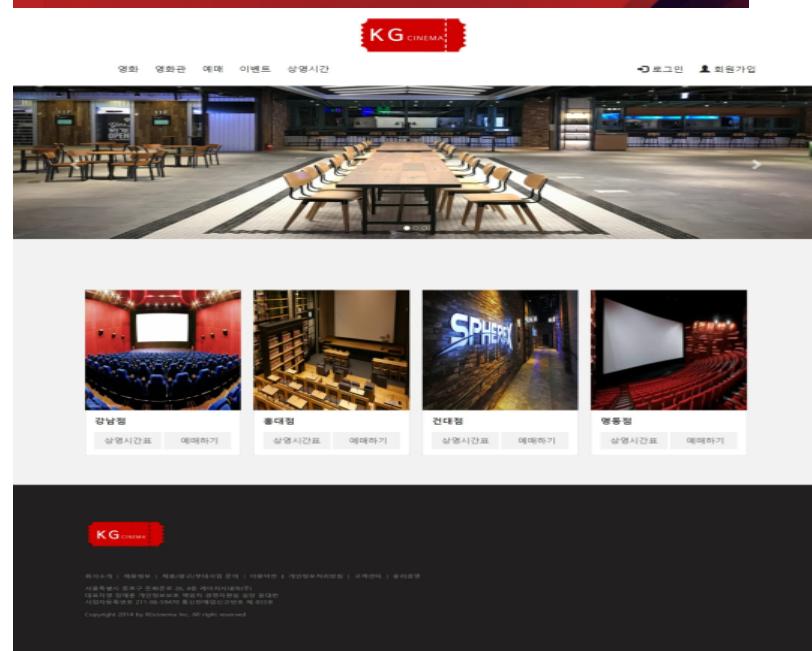
4-1) home.html



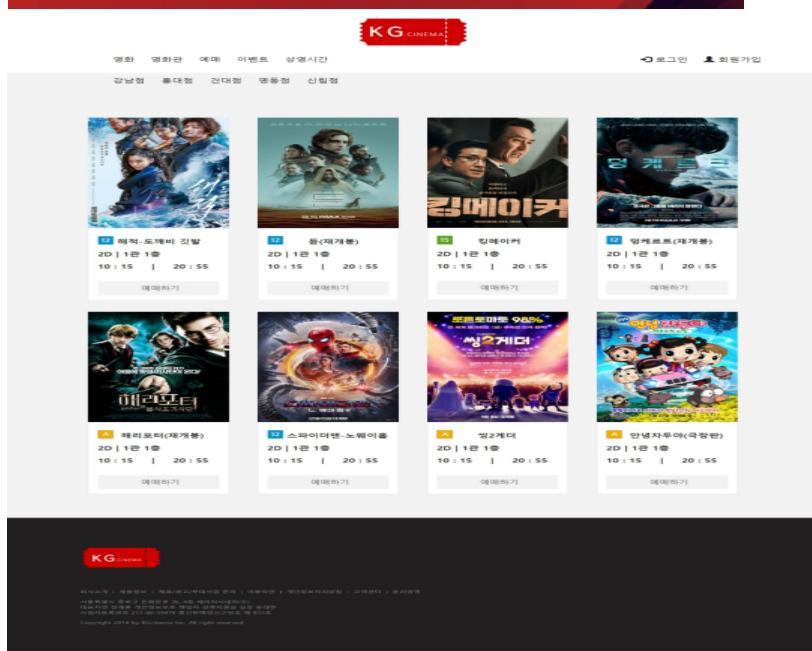
4-2) moive.html



4-3) cinema.html



4-4) time.html

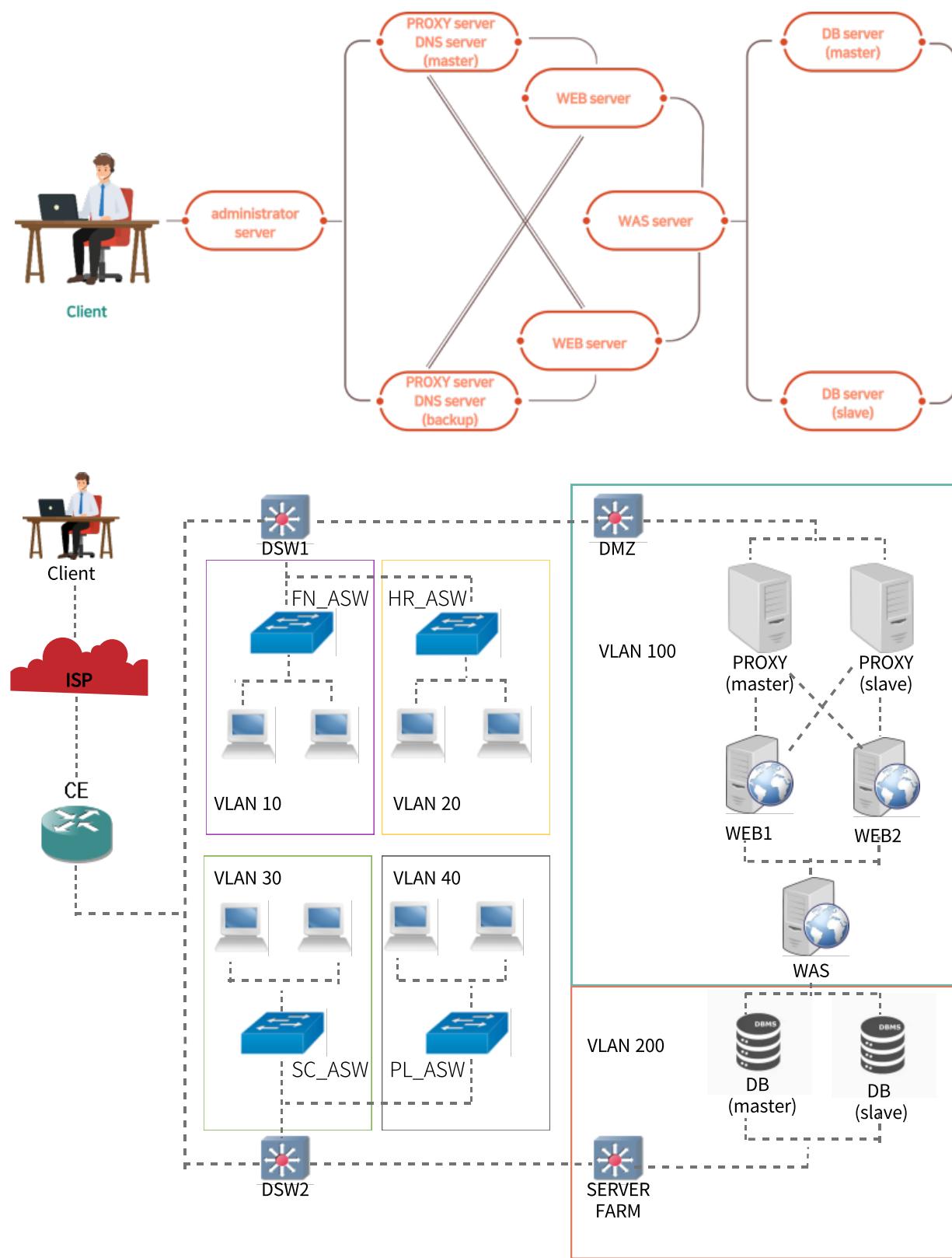


K G C I N E M A

네트워크
토론플랫폼
워로크지

02

01. 네트워크 논리적 구상도



02. 네트워크, 서버 장비 IP 할당

◆ IP 부여 정보 및 네트워크 대역

| | | |
|---------|----------|--------------|
| VLAN100 | proxy(M) | 192.168.1.10 |
| | proxy(S) | 192.168.1.20 |
| | web01 | 192.168.1.30 |
| | web02 | 192.168.1.40 |

| | | |
|---------------|-------|--------------|
| VLAN100 | was | 192.168.1.50 |
| VLAN200 | db(M) | 192.168.1.60 |
| | db(S) | 192.168.1.70 |
| administrator | | 192.168.1.80 |

| | | |
|--------|-----|--------------|
| VLAN10 | pc1 | 192.168.10.1 |
| | pc2 | 192.168.10.2 |
| VLAN20 | pc3 | 192.168.20.1 |
| | pc4 | 192.168.20.2 |

| | | |
|--------|-----|--------------|
| VLAN30 | pc5 | 192.168.30.1 |
| VLAN40 | pc6 | 192.168.30.2 |
| | pc7 | 192.168.40.1 |
| VLAN40 | pc8 | 192.168.40.2 |
| | | |

| | | |
|------|------|------------------|
| DSW1 | net | 192.168.100.0/16 |
| | DSW2 | 192.168.100.9/16 |
| | CE | 192.168.100.2/16 |

| | | |
|------|------|-------------------|
| DSW2 | net | 192.168.100.0/16 |
| | DSW1 | 192.168.100.10/16 |
| | CE | 192.168.100.3/16 |

| | | |
|----|------|------------------|
| CE | net | 192.168.100.0/16 |
| | DSW1 | 192.168.100.1/16 |
| | DSW2 | 192.168.100.5/16 |

02. 네트워크, 서버 장비 IP 할당

◆ VPS IP설정SVI 및 VLAN Gateway IP 부여 정보

| DSW1 | VLAN_GW | DSW2 | VLAN_GW |
|--------------|----------------|--------------|----------------|
| DSW1 SVI 10 | 192.168.10.252 | DSW2 SVI 10 | 192.168.10.253 |
| DSW1 SVI 20 | 192.168.20.252 | DSW2 SVI 20 | 192.168.20.253 |
| DSW1 SVI 30 | 192.168.30.252 | DSW2 SVI 30 | 192.168.30.253 |
| DSW1 SVI 40 | 192.168.40.252 | DSW2 SVI 40 | 192.168.40.253 |
| DSW1 SVI 100 | 192.168.50.252 | DSW2 SVI 100 | 192.168.50.253 |
| DSW1 SVI 200 | 192.168.60.252 | DSW2 SVI 200 | 192.168.60.253 |

◆ 장비 별 포트 연결

| | | | |
|---------------|-----------------|---------------|---------------|
| FN_ASW - DSW1 | e1/0-1 – e1/0-1 | FN_ASW - DSW2 | FN_ASW - DSW2 |
| HR_ASW - DSW1 | e1/0-1 – e1/2-3 | HR_ASW - DSW2 | HR_ASW - DSW2 |
| SC_ASW - DSW1 | e1/0-1 – e2/0-1 | SC_ASW - DSW2 | SC_ASW - DSW2 |
| PL_ASW - DSW1 | e1/0-1 – e2/2-3 | PL_ASW - DSW2 | PL_ASW - DSW2 |
| DMZ_SW - DSW1 | e3/0-1 – e3/0-1 | DMZ_SW - DSW2 | DMZ_SW - DSW2 |
| SF_SW - DSW1 | e3/2-3 – e3/2-3 | SF_SW - DSW2 | SF_SW - DSW2 |
| DSW2 - DSW1 | e0/1 – e0/1 | DSW1 - DSW2 | DSW1 - DSW2 |
| CE - DSW1 | e0/0 – f0/0 | CE - DSW2 | CE - DSW2 |
| CE - ISP | f1/0 - VMnet8 | | |

03. 네트워크 설정

◆ VPS IP설정

```
FN_PC1:ip 192.168.10.1 255.255.0.0 192.168.1.254  
FN_PC2:ip 192.168.10.2 255.255.0.0 192.168.1.254
```

```
HR_PC3:ip 192.168.20.1 255.255.0.0 192.168.1.254  
HR_PC4:ip 192.168.20.2 255.255.0.0 192.168.1.254
```

```
SC_PC5:ip 192.168.30.1 255.255.0.0 192.168.1.254  
SC_PC6:ip 192.168.30.2 255.255.0.0 192.168.1.254
```

```
PL_PC7:ip 192.168.40.1 255.255.0.0 192.168.1.254  
PL_PC8:ip 192.168.40.2 255.255.0.0 192.168.1.254
```

◆ FN_ASW 설정

```
Switch#config t  
Switch(config)#hostname FN_ASW  
FN_ASW(config)#enable se kgb123  
FN_ASW(config)#no ip domain lookup  
FN_ASW(config)#line con 0  
FN_ASW(config-line)#logging sy  
FN_ASW(config-line)#exec-time 0 0  
FN_ASW(config-line)#password kgbank123  
FN_ASW(config-line)#login  
FN_ASW(config-line)#exit  
FN_ASW(config)#int range e 0/0 - 3, e 1/0 - 3, e 2/0 - 3, e 3/0 - 3  
FN_ASW(config-if-range)#sh  
FN_ASW(config-if-range)#exit
```

◆ VLAN 설정

VLAN 10

```
FN_ASW(config)#vlan 10  
FN_ASW(config-vlan)#name FN  
FN_ASW(config-vlan)#exit
```

VLAN 20

```
FN_ASW(config)#vlan 20  
FN_ASW(config-vlan)#name HR  
FN_ASW(config-vlan)#exit
```

03. 네트워크 설정

◆ VLAN 설정

VLAN 30

```
FN_ASW(config)#vlan 30  
FN_ASW(config-vlan)#name SC  
FN_ASW(config-vlan)#exit
```

VLAN 100

```
FN_ASW(config)#vlan 100  
FN_ASW(config-vlan)#name DMZ  
FN_ASW(config-vlan)#exit
```

VLAN 40

```
FN_ASW(config)#vlan 40  
FN_ASW(config-vlan)#name PL  
FN_ASW(config-vlan)#exit
```

VLAN 200

```
FN_ASW(config)#vlan 200  
FN_ASW(config-vlan)#name SF  
FN_ASW(config-vlan)#exit
```

◆ VLAN Access Link 설정

FN_PC1

```
FN_ASW(config)#int e 0/0  
FN_ASW(config-if)#description ##FN_PC1##  
FN_ASW(config-if)#switchport mode access  
FN_ASW(config-if)#switchport access vlan 10  
FN_ASW(config-if)#spanning-tree portfast  
FN_ASW(config-if)#no shutdown
```

FN_PC1

```
FN_ASW(config)#int e 0/1  
FN_ASW(config-if)#description ##FN_PC2##  
FN_ASW(config-if)#switchport mode access  
FN_ASW(config-if)#switchport access vlan 10  
FN_ASW(config-if)#spanning-tree portfast  
FN_ASW(config-if)#no shutdown
```

◆ VLAN Trunk Link 설정 + Port Channel 구성

FN - DSW1

```
FN_ASW(config)#int range e 1/0-1  
FN_ASW(config-if range)#desc##DSW1_FN_Trunk##  
FN_ASW(config-if-range)#swi tru encapsulation dot1q  
FN_ASW(config-if-range)#swi tru allow vlan 10,20,30,40,  
100,200  
FN_ASW(config-if-range)#switchport mode trunk  
FN_ASW(config-if-range)#switchport nonegotiate  
FN_ASW(config-if-range)#channel-group 1 mode auto  
FN_ASW(config-if-range)#no shutdown  
FN_ASW(config-if-range)#exit
```

FN - DSW2

```
FN_ASW(config)#int range e 2/0-1  
FN_ASW(config-if-range)#desc##DSW2_FN_Trunk##  
FN_ASW(config-if-range)#switchport trunk encapsulation dot1q  
FN_ASW(config-if-range)#swi tru allow vlan 10,20,30,40,  
100,200  
FN_ASW(config-if-range)#switchport mode trunk  
FN_ASW(config-if-range)#switchport nonegotiate  
FN_ASW(config-if-range)#channel-group 2 mode auto  
FN_ASW(config-if-range)#no shutdown  
FN_ASW(config-if-range)#exit
```

◆ HR_ASW 설정

```
Switch#config
Switch(config)#hostname HR_ASW
HR_ASW(config)#enable se kgb123
HR_ASW(config)#no ip domain lookup
HR_ASW(config)#line con 0
HR_ASW(config-line)#logging sy
HR_ASW(config-line)#exec-time 0 0
HR_ASW(config-line)#password kgbank123
HR_ASW(config-line)#login
HR_ASW(config-line)#exit
HR_ASW(config)#int range e 0/0 - 3, e 1/0 - 3, e 2/0 - 3, e 3/0 - 3
HR_ASW(config-if-range)#sh
HR_ASW(config-if-range)#exit
```

◆ VLAN 설정

VLAN 10

```
HR_ASW(config)#vlan 10
HR_ASW(config-vlan)#name FN
HR_ASW(config-vlan)#exit
```

VLAN 30

```
HR_ASW(config)#vlan 30
HR_ASW(config-vlan)#name SC
HR_ASW(config-vlan)#exit
```

VLAN 100

```
HR_ASW(config)#vlan 100
HR_ASW(config-vlan)#name DMZ
HR_ASW(config-vlan)#exit
```

VLAN 20

```
HR_ASW(config)#vlan 20
HR_ASW(config-vlan)#name HR
HR_ASW(config-vlan)#exit
```

VLAN 40

```
HR_ASW(config)#vlan 40
HR_ASW(config-vlan)#name PL
HR_ASW(config-vlan)#exit
```

VLAN 200

```
HR_ASW(config)#vlan 200
HR_ASW(config-vlan)#name SF
HR_ASW(config-vlan)#exit
```

◆ ss Link 설정

HR_PC3

```
HR_ASW(config)#int e 0/0
HR_ASW(config-if)#description ##HE_PC3##
HR_ASW(config-if)#switchport mode access
HR_ASW(config-if)#switchport access vlan20
HR_ASW(config-if)#spanning-tree portfast
HR_ASW(config-if)#no shutdown
```

HR_PC4

```
HR_ASW(config)#int e 0/1
HR_ASW(config-if)#description ##HR_PC4##
HR_ASW(config-if)#switchport mode access
HR_ASW(config-if)#switchport access vlan20
HR_ASW(config-if)#spanning-tree portfast
HR_ASW(config-if)#no shutdown
```

◆ VLAN Trunk Link 설정 + Port Channel 구성

HR – DSW1

```
HR_ASW(config)#int range e 1/0-1
HR_ASW(config-if range)#desc##DSW1_HR_Trunk##
HR_ASW(config-if-range)#swi tru encapsulation dot1q
HR_ASW(config-if-range)#swi tru allow vlan 10,20,30,40,10
0,200
HR_ASW(config-if-range)#switchport mode trunk
HR_ASW(config-if-range)#switchport nonegotiate
HR_ASW(config-if-range)#channel-group 1 mode auto
HR_ASW(config-if-range)#no shutdown
HR_ASW(config-if-range)#exit
```

HR – DSW2

```
HR_ASW(config)#int range e 2/0-1
HR_ASW(config-if-range)#description ##DSW2_HR_Trunk##
HR_ASW(config-if-range)#switchport trunk encapsulation
dot1q
HR_ASW(config-if-range)#swi tru allow vlan 10,20,30,40,10
0,200
HR_ASW(config-if-range)#switchport mode trunk
HR_ASW(config-if-range)#switchport nonegotiate
HR_ASW(config-if-range)#channel-group 2 mode auto
HR_ASW(config-if-range)#no shutdown
HR_ASW(config-if-range)#exit
```

◆ SC_ASW 설정

```
Switch#config t
Switch(config)#hostname SC_ASW
SC_ASW(config)#enable se kgb123
SC_ASW(config)#no ip domain lookup
SC_ASW(config)#line con 0
SC_ASW(config-line)#logging sy
SC_ASW(config-line)#exec-time 0 0
SC_ASW(config-line)#password kgbank123
SC_ASW(config-line)#login
SC_ASW(config-line)#exit
SC_ASW(config)#int range e 0/0 - 3, e 1/0 - 3, e 2/0 - 3, e 3/0 - 3
SC_ASW(config-if-range)#sh
SC_ASW(config-if-range)#exit
```

◆ VLAN 설정

VLAN 10

```
SC_ASW(config)#vlan 10
SC_ASW(config-vlan)#name FN
SC_ASW(config-vlan)#exit
```

VLAN 30

```
SC_ASW(config)#vlan 30
SC_ASW(config-vlan)#name SC
SC_ASW(config-vlan)#exit
```

VLAN 100

```
SC_ASW(config)#vlan 100
SC_ASW(config-vlan)#name DMZ
SC_ASW(config-vlan)#exit
```

VLAN 20

```
SC_ASW(config)#vlan 20
SC_ASW(config-vlan)#name HR
SC_ASW(config-vlan)#exit
```

VLAN 40

```
SC_ASW(config)#vlan 40
SC_ASW(config-vlan)#name PL
SC_ASW(config-vlan)#exit
```

VLAN 200

```
SC_ASW(config)#vlan 200
SC_ASW(config-vlan)#name SF
SC_ASW(config-vlan)#exit
```

◆ VLAN Access Link 설정

SC_PC5

```
SC_ASW(config)#int e 0/0
SC_ASW(config-if)#description ##SC_PC5##
SC_ASW(config-if)#switchport mode access
SC_ASW(config-if)#switchport access vlan30
SC_ASW(config-if)#spanning-tree portfast
SC_ASW(config-if)#no shutdown
```

SC_PC6

```
SC_ASW(config)#int e 0/1
SC_ASW(config-if)#description ##SC_PC6##
SC_ASW(config-if)#switchport mode access
SC_ASW(config-if)#switchport access vlan30
SC_ASW(config-if)#spanning-tree portfast
SC_ASW(config-if)#no shutdown
```

◆ VLAN Trunk Link 설정 + Port Channel 구성

SC – DSW1

```
SC_ASW(config)#int range e 1/0-1
SC_ASW(config-if range)#desc##DSW1_SC_Trunk##
SC_ASW(config-if-range)#swi tru encapsulation dot1q
SC_ASW(config-if-range)#swi tru allow vlan 10,20,30,40,1
00,200
SC_ASW(config-if-range)#switchport mode trunk
SC_ASW(config-if-range)#switchport nonegotiate
SC_ASW(config-if-range)#channel-group 1 mode auto
SC_ASW(config-if-range)#no shutdown
SC_ASW(config-if-range)#exit
```

SC – DSW2

```
SC_ASW(config)#int range e 2/0-1
SC_ASW(config-if-range)#description ##DSW2_SC_Trunk
##
SC_ASW(config-if-range)#switchport trunk encapsulatio
n dot1q
SC_ASW(config-if-range)#swi tru allow vlan 10,20,30,40,1
00,200
SC_ASW(config-if-range)#switchport mode trunk
SC_ASW(config-if-range)#switchport nonegotiate
SC_ASW(config-if-range)#channel-group 2 mode auto
SC_ASW(config-if-range)#no shutdown
SC_ASW(config-if-range)#exit
```

03. 네트워크 설정

◆ PL_ASW 설정

```
Switch#config t
Switch(config)#hostname PL_ASW
PL_ASW(config)#enable se kgb123
PL_ASW(config)#no ip domain lookup
PL_ASW(config)#line con 0
PL_ASW(config-line)#logging sy
PL_ASW(config-line)#exec-time 0 0
PL_ASW(config-line)#password kgbank123
PL_ASW(config-line)#login
PL_ASW(config-line)#exit
PL_ASW(config)#int range e 0/0 - 3, e 1/0 - 3, e 2/0 - 3, e 3/0 - 3
PL_ASW(config-if-range)#sh
PL_ASW(config-if-range)#exit
```

◆ VLAN 설정

VLAN 10

```
PL_ASW(config)#vlan 10
PL_ASW(config-vlan)#name FN
PL_ASW(config-vlan)#exit
```

VLAN 30

```
PL_ASW(config)#vlan 30
PL_ASW(config-vlan)#name SC
PL_ASW(config-vlan)#exit
```

VLAN 100

```
PL_ASW(config)#vlan 100
PL_ASW(config-vlan)#name DMZ
PL_ASW(config-vlan)#exit
```

VLAN 20

```
PL_ASW(config)#vlan 20
PL_ASW(config-vlan)#name HR
PL_ASW(config-vlan)#exit
```

VLAN 40

```
PL_ASW(config)#vlan 40
PL_ASW(config-vlan)#name PL
PL_ASW(config-vlan)#exit
```

VLAN 200

```
PL_ASW(config)#vlan 200
PL_ASW(config-vlan)#name SF
PL_ASW(config-vlan)#exit
```

◆ VLAN Access Link 설정

PL_PC7

```
PL_ASW(config)#int e 0/0
PL_ASW(config-if)#description ##PL_PC7##
PL_ASW(config-if)#switchport mode access
PL_ASW(config-if)#switchport access vlan40
PL_ASW(config-if)#spanning-tree portfast
PL_ASW(config-if)#no shutdown
```

PL_PC8

```
PL_ASW(config)#int e 0/1
PL_ASW(config-if)#description ##PL_PC8##
PL_ASW(config-if)#switchport mode access
PL_ASW(config-if)#switchport access vlan40
PL_ASW(config-if)#spanning-tree portfast
PL_ASW(config-if)#no shutdown
```

◆ VLAN Trunk Link 설정 + Port Channel 구성

PL – DSW1

```
PL_ASW(config)#int range e 1/0-1
PL_ASW(config-if range)#desc##DSW1_PL_Trunk##
PL_ASW(config-if-range)#swi tru encapsulation dot1q
PL_ASW(config-if-range)#swi tru allow vlan 10,20,30,40,1
00,200
PL_ASW(config-if-range)#switchport mode trunk
PL_ASW(config-if-range)#switchport nonegotiate
PL_ASW(config-if-range)#channel-group 1 mode auto
PL_ASW(config-if-range)#no shutdown
PL_ASW(config-if-range)#exit
```

PL – DSW2

```
PL_ASW(config)#int range e 2/0-1
PL_ASW(config-if-range)#description ##DSW2_PL_Trunk##
PL_ASW(config-if-range)#switchport trunk encapsulation dot
1q
PL_ASW(config-if-range)#swi tru allow vlan 10,20,30,40,100,2
00
PL_ASW(config-if-range)#switchport mode trunk
PL_ASW(config-if-range)#switchport nonegotiate
PL_ASW(config-if-range)#channel-group 2 mode auto
PL_ASW(config-if-range)#no shutdown
PL_ASW(config-if-range)#exit
```

◆ DSW1 설정

```
IOU#conf t
IOU(config)#hostname DSW1
DSW1(config)#enable se kgb123
DSW1(config)#no ip domain lookup
DSW1(config)#line con 0
DSW1(config-line)#logging sy
DSW1(config-line)#exec-time 0 0
DSW1(config-line)#password kgbank123
DSW1(config-line)#login
DSW1(config-line)#exit
```

◆ VLAN 설정

VLAN 10

```
DSW1(config)#vlan 10
DSW1(config-vlan)#name FN
DSW1(config-vlan)#exit
```

VLAN 20

```
DSW1(config)#vlan 20
DSW1(config-vlan)#name HR
DSW1(config-vlan)#exit
```

VLAN 30

```
DSW1(config)#vlan 30
DSW1(config-vlan)#name SC
DSW1(config-vlan)#exit
```

VLAN 40

```
DSW1(config)#vlan 40
DSW1(config-vlan)#name PL
DSW1(config-vlan)#exit
```

VLAN 100

```
DSW1(config)#vlan 100
DSW1(config-vlan)#name DMZ
DSW1(config-vlan)#exit
```

VLAN 200

```
DSW1(config)#vlan 200
DSW1(config-vlan)#name SF
DSW1(config-vlan)#exit
```

◆ Interface 설정

◆ Router Connection 설정

```
DSW1(config)#interface e 0/0
DSW1(config-if)#description ##CE_Connection##
DSW1(config-if)#no switchport
DSW1(config-if)#ip add 192.168.100.2 255.255.0.0
DSW1(config-if)#duplex full
DSW1(config-if)#no shutdown
DSW1(config-if)#exit
```

◆ DSW2 Connection 설정

```
DSW1(config)#interface e 0/1
DSW1(config-if)#description ##DWS2_Connection##
DSW1(config-if)#no switchport
DSW1(config-if)#ip add 192.168.100.9 255.255.0.0
DSW1(config-if)#duplex full
DSW1(config-if)#no shutdown
DSW1(config-if)#exit
```

◆ Inter_VLAN설정 및 HSRP

```
DSW1(config)#track 1 interface e 0/0 line-protocol
DSW1(config)#exit
```

03. 네트워크 설정

◆ VLAN Trunk Link 설정 + port channel 구성

VLAN 10

```
DSW1(config)#interface range e 1/0-1
DSW1(config-if-range)#desc##FN_Trunk_Port##
DSW1(config-if-range)#swi truencap dot1q
DSW1(config-if-range)#swi tru allow vlan 10
DSW1(config-if-range)#swi mode trunk
DSW1(config-if-range)#swi nonegotiate
DSW1(config-if-range)#channel-group 1 mo auto
DSW1(config-if-range)#no sh
DSW1(config-if-range)#exit
```

VLAN 20

```
DSW1(config)#interface range e 1/2-3
DSW1(config-if-range)#desc##HR_Trunk_Port##
DSW1(config-if-range)#swi truencap dot1q
DSW1(config-if-range)#swi tru allow vlan20
DSW1(config-if-range)#swi mode trunk
DSW1(config-if-range)#swi nonegotiate
DSW1(config-if-range)#channel-group 2 mo auto
DSW1(config-if-range)#no sh
DSW1(config-if-range)#exit
```

VLAN 30

```
DSW1(config)#interface range e 2/0-1
DSW1(config-if-range)#desc##SC_Trunk_Port##
DSW1(config-if-range)#swi truencap dot1q
DSW1(config-if-range)#swi tru allow vlan30
DSW1(config-if-range)#swi mode trunk
DSW1(config-if-range)#swi nonegotiate
DSW1(config-if-range)#channel-group 3 mo auto
DSW1(config-if-range)#no sh
DSW1(config-if-range)#exit
```

VLAN 40

```
DSW1(config)#interface range e 2/2-3
DSW1(config-if-range)#desc##PL_Trunk_Port##
DSW1(config-if-range)#swi truencap dot1q
DSW1(config-if-range)#swi tru allow vlan40
DSW1(config-if-range)#swi mode trunk
DSW1(config-if-range)#swi nonegotiate
DSW1(config-if-range)#channel-group 4mo auto
DSW1(config-if-range)#no sh
DSW1(config-if-range)#exit
```

VLAN 100

```
DSW1(config)#interface range e 3/0-1
DSW1(config-if-range)#desc##DMZ_Trunk_Port##
DSW1(config-if-range)#swi truencap dot1q
DSW1(config-if-range)#swi tru allow vlan 100
DSW1(config-if-range)#swi mode trunk
DSW1(config-if-range)#swi nonegotiate
DSW1(config-if-range)#channel-group 5mo auto
DSW1(config-if-range)#no sh
DSW1(config-if-range)#exit
```

VLAN 200

```
DSW1(config)#interface range e 3/2-3
DSW1(config-if-range)#desc##SF_Trunk_Port##
DSW1(config-if-range)#swi truencap dot1q
DSW1(config-if-range)#swi tru allow vlan200
DSW1(config-if-range)#swi mode trunk
DSW1(config-if-range)#swi nonegotiate
DSW1(config-if-range)#channel-group 6mo auto
DSW1(config-if-range)#no sh
DSW1(config-if-range)#exit
```

03. 네트워크 설정

◆ Inter_VLAN설정 및 HSRP

VLAN 10 (Active)

```
DSW1(config)#interface vlan 10
DSW1(config-if)#desc##VLAN10_Gateway##
DSW1(config-if)#ip add 192.168.10.252 255.255.0.
0
DSW1(config-if)#standby 10 ip 192.168.10.254
DSW1(config-if)#standby 10 priority 110
DSW1(config-if)#standby 10 track 1 decrement 5
0
DSW1(config-if)#standby 10 preempt delay mini
mum 30
DSW1(config-if)#no sh
DSW1(config-if)#exit
```

VLAN 30 (Active)

```
DSW1(config)#interface vlan30
DSW1(config-if)#desc##VLAN30_Gateway##
DSW1(config-if)#ip add 192.168.30.252 255.255.0.
0
DSW1(config-if)#standby 30 ip 192.168.30.254
DSW1(config-if)#standby 10 priority 110
DSW1(config-if)#standby 30 track 1 decrement 5
0
DSW1(config-if)#standby 30 preempt delay mini
mum 30
DSW1(config-if)#no sh
DSW1(config-if)#exit
```

VLAN 100 (Active)

```
DSW1(config)#interface vlan 100
DSW1(config-if)#desc##VLAN100_Gateway##
DSW1(config-if)#ip add 192.168.100.252 255.255.0.0
DSW1(config-if)#standby 100 ip 192.168.100.254
DSW1(config-if)#standby 100 priority 110
DSW1(config-if)#standby 100 track 1 decrement 50
DSW1(config-if)#standby 100 preempt delay minimu
m 30
DSW1(config-if)#no sh
DSW1(config-if)#exit
```

VLAN 20 (Standby)

```
DSW1(config)#interface vlan20
DSW1(config-if)#desc##VLAN20_Gateway##
DSW1(config-if)#ip add 192.168.20.252 255.255.0.0
DSW1(config-if)#standby 20 ip 192.168.20.254
DSW1(config-if)#standby 20 preempt delay
DSW1(config-if)#no sh
DSW1(config-if)#exit
```

VLAN 40 (Standby) 40

```
DSW1(config)#interface vlan40
DSW1(config-if)#desc##VLAN40_Gateway##
DSW1(config-if)#ip add 192.168.40.252 255.255.0.0
DSW1(config-if)#standby 40 ip 192.168.40.254
DSW1(config-if)#standby 40 preempt delay
DSW1(config-if)#no sh
DSW1(config-if)#exit
```

VLAN 200 (Standby)

```
DSW1(config)#interface vlan200
DSW1(config-if)#desc##VLAN200_Gateway##
DSW1(config-if)#ip add 192.168.200.252 255.255.0.0
DSW1(config-if)#standby 200 ip 192.168.200.254
DSW1(config-if)#standby 200 preempt
DSW1(config-if)#no sh
DSW1(config-if)#exit
```

◆ RSTP 및 VLAN 우선순위 조정

```
DSW1(config)#spanning-tree vlan 10 priority 4096  
DSW1(config)#spanning-tree vlan 30 priority 4096  
DSW1(config)#spanning-tree vlan100 priority 4096
```

◆ DSW1 Routing RIP 설정

```
DSW1(config)#router rip  
DSW1(config-router)#version 2  
DSW1(config-router)#noauto-summary  
DSW1(config-router)#net 21.0.0.0  
DSW1(config-router)#exit  
DSW1(config)#no ipcef  
DSW1(config)#end  
DSW1#wr
```

03. 네트워크 설정

◆ DSW2 설정

```
IOU#conf t  
IOU(config)#hostname DSW2  
DSW2(config)#enable se kgb123  
DSW2(config)#no ip domain lookup  
DSW2(config)#line con 0  
DSW2(config-line)#logging sy  
DSW2(config-line)#exec-time 0 0  
DSW2(config-line)#password kgbank123  
DSW2(config-line)#login  
DSW2(config-line)#exit
```

◆ VLAN 설정

VLAN 10

```
DSW2(config)#vlan 10  
DSW2(config-vlan)#name FN  
DSW2(config-vlan)#exit
```

VLAN 20

```
DSW2(config)#vlan 20  
DSW2(config-vlan)#name HR  
DSW2(config-vlan)#exit
```

VLAN 30

```
DSW2(config)#vlan 30  
DSW2(config-vlan)#name SC  
DSW2(config-vlan)#exit
```

VLAN 40

```
DSW2(config)#vlan 40  
DSW2(config-vlan)#name PL  
DSW2(config-vlan)#exit
```

VLAN 100

```
DSW2(config)#vlan 100  
DSW2(config-vlan)#name DMZ  
DSW2(config-vlan)#exit
```

VLAN 200

```
DSW2(config)#vlan 200  
DSW2(config-vlan)#name SF  
DSW2(config-vlan)#exit
```

◆ Interface 설정

◆ Router Connection 설정

```
DSW2(config)#interface e 0/0
DSW2(config-if)#description ##CE_Connection##
DSW2(config-if)#no switchport
DSW2(config-if)#ip add 192.168.100.6 255.255.0.0
DSW2(config-if)#duplex full
DSW2(config-if)#no shutdown
DSW2(config-if)#exit
```

◆ DSW2 Connection 설정

```
DSW2(config)#interface e 0/1
DSW2(config-if)#description ##DWS1_Connection##
DSW2(config-if)#no switchport
DSW2(config-if)#ip add 192.168.100.10 255.255.0.0
DSW2(config-if)#duplex full
DSW2(config-if)#no shutdown
DSW2(config-if)#exit
```

◆ Inter_VLAN설정 및 HSRP

```
DSW2(config)#track 1 interface e 0/0 line-protocol
DSW2(config)#exit
```

03. 네트워크 설정

◆ VLAN Trunk Link 설정 + port channel 구성

VLAN 10

```
DSW2(config)#interface range e 1/0-1
DSW2(config-if-range)#desc##FN_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan 10
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 1 mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

VLAN 20

```
DSW2(config)#interface range e 1/2-3
DSW2(config-if-range)#desc##HR_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan20
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 2 mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

VLAN 30

```
DSW2(config)#interface range e 2/0-1
DSW2(config-if-range)#desc##SC_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan30
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 3 mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

VLAN 40

```
DSW2(config)#interface range e 2/2-3
DSW2(config-if-range)#desc##PL_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan40
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 4mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

VLAN 100

```
DSW2(config)#interface range e 3/0-1
DSW2(config-if-range)#desc##DMZ_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan 100
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 5mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

VLAN 200

```
DSW2(config)#interface range e 3/2-3
DSW2(config-if-range)#desc##SF_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan200
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 6mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

03. 네트워크 설정

★ Inter_VLAN설정 및 HSRP

VLAN 10 (Standby)

```
DSW2(config)#interface vlan 10
DSW2(config-if)#desc##VLAN10_Gateway##
DSW2(config-if)#ip add 192.100.10.253 255.2
55.0.0
DSW2(config-if)#standby 10 ip 192.168.10.25
4
DSW2(config-if)#standby 10 preempt
DSW2(config-if)#no sh
DSW2(config-if)#exit
```

VLAN 20 (Active)

```
DSW2(config)#interface vlan20
DSW2(config-if)#desc##VLAN20_Gateway##
DSW2(config-if)#ip add 192.168.20.253 255.255.0.0
DSW2(config-if)#standby 20 ip 192.168.20.254
DSW2(config-if)#standby 20 priority 110
DSW2(config-if)#standby 20 track 1 decrement 50
DSW2(config-if)#standby 20 preempt delay
DSW2(config-if)#no sh
DSW2(config-if)#exit
```

VLAN 30 (Standby)

```
DSW2(config)#interface range e 2/0-1
DSW2(config-if-range)#desc##SC_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan30
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 3 mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

VLAN 40 (Active)

```
DSW2(config)#interface range e 2/2-3
DSW2(config-if-range)#desc##PL_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan40
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 4mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

VLAN 100 (Standby)

```
DSW2(config)#interface range e 3/0-1
DSW2(config-if-range)#desc##DMZ_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan 100
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 5mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

VLAN 200 (Active)

```
DSW2(config)#interface range e 3/2-3
DSW2(config-if-range)#desc##SF_Trunk_Port##
DSW2(config-if-range)#swi truencap dot1q
DSW2(config-if-range)#swi tru allow vlan200
DSW2(config-if-range)#swi mode trunk
DSW2(config-if-range)#swi nonegotiate
DSW2(config-if-range)#channel-group 6mo auto
DSW2(config-if-range)#no sh
DSW2(config-if-range)#exit
```

◆ RSTP 및 VLAN 우선순위 조정

```
DSW2(config)#spanning-tree vlan20 priority 4096  
DSW2(config)#spanning-tree vlan40 priority 4096  
DSW2(config)#spanning-tree vlan200 priority 4096
```

◆ DSW1 Routing RIP 설정

```
DSW2(config)#router rip  
DSW2(config-router)#version 2  
DSW2(config-router)#noauto-summary  
DSW2(config-router)#net 21.0.0.0  
DSW2(config-router)#exit  
DSW2(config)#no ipcef  
DSW2(config)#end  
DSW2#wr
```

◆ CE (Router) 설정

```
R1#conf t
R1(config)#hostname CE
CE(config)#enable secret kgb123
CE(config)#no ip domain lookup
CE(config)#line console 0
CE(config-line)#logging synchronous
CE(config-line)#exec-timeout 0 0
CE(config-line)#password kgbank123
CE(config-line)#exit
CE(config)#int range e0/0-3, e1/0-3, e2/0-3, e3/0-3
CE(config-if-range)#sh
CE(config-if-range)#exit
```

◆ Interface 설정

◆ DSW1 Connection 설정

```
CE(config)#interface fastEthernet 0/0
CE(config-if)#description ##DSW1_Connection##
CE(config-if)#no switchport
CE(config-if)#ip address 192.168.100.1 255.255.0.0
CE(config-if)#duplex full
CE(config-if)#no shutdown
CE(config-if)#exit
```

◆ DSW2 Connection 설정

```
DSW2
CE(config)#interface fastEthernet 0/1
CE(config-if)#description ##DSW2_Connection##
CE(config-if)#no switchport
CE(config-if)#ip address 192.168.100.5 255.255.0.0
CE(config-if)#duplex full
CE(config-if)#no shutdown
CE(config-if)#exit
```

◆ ISP Connection 설정

```
CE(config)#interface fastEthernet 1/0
CE(config-if)#desc ##ISP_Connection##
CE(config-if)#ip address dhcp
CE(config-if)#no shutdown
CE(config-if)#exit
```

◆ Routing RIP 설정

```
CE(config)#router rip
CE(config-router)#version 2
CE(config-router)#no auto-summary
CE(config-router)#network 192.168.0.0
CE(config-router)#default-information originate
CE(config-router)#exit
CE(config)#end
CE#wr
```

◆ DMZ Server (Web / DNS / FTP) 설정

```
Switch>enable
Switch#configure terminal
Switch(config)#hostname DMZ_SW
DMZ(config)#enable secret kgb123
DMZ(config)#no ip domain lookup
DMZ(config)#line console 0
DMZ(config-line)#logging synchronous
DMZ(config-line)#exec-time 0 0
DMZ(config-line)#password kgbank123
DMZ(config-line)#login
DMZ(config-line)#exit
DMZ(config)#interface range e 0/0-3, e 1/0-3, e 2/0-3, e 3/0-3
DMZ(config-if-range)#shutdown
DMZ(config-if-range)#exit
```

◆ VLAN 설정

```
DMZ(config)#vlan 100
DMZ(config-vlan)#name DMZ
DMZ(config-vlan)#exit
```

◆ VLAN Access Link설정

```
DMZ(config)#interface range e 0/0-2
DMZ(config-if-range)#description ##DMZ##
DMZ(config-if-range)#switchport mode access
DMZ(config-if-range)#switchport access vlan 100
DMZ(config-if-range)#spanning-tree portfast
DMZ(config-if-range)#no shutdown
DMZ(config-if-range)#exit
```

◆ VLAN Trunk Link + Port Channel 구성

◆ DMZ – DSW1 설정

```
DMZ(config)#interface range e 3/0-1
DMZ(config-if-range)#desc ##DSW1_Trunk##
DMZ(config-if-range)#swi truencap dot1q
DMZ(config-if-range)#swi tru allow vlan 100
DMZ(config-if-range)#swi mode tru
DMZ(config-if-range)#swit none
DMZ(config-if-range)#channel-group 1 mo auto
DMZ(config-if-range)#no shutdown
DMZ(config-if-range)#exit
```

◆ DMZ – DSW2 설정

```
DMZ(config)#interface range e 3/2-3
DMZ(config-if-range)#desc ##DSW2_Trunk##
DMZ(config-if-range)#swi truencap dot1q
DMZ(config-if-range)#swi tru allow vlan 100
DMZ(config-if-range)#swi mode tru
DMZ(config-if-range)#swit none
DMZ(config-if-range)#channel-group 2mo auto
DMZ(config-if-range)#no shutdown
DMZ(config-if-range)#exit
```

◆ Proxy1 Connection 설정

```
DMZ(config)#interface ethernet 1/0
DMZ(config-if)#description #Proxy1_connection##
DMZ(config-if)#no switchport
DMZ(config-if)#ip address 192.168.1.10 255.255.0.0
DMZ(config-if)#no shutdown
DMZ(config-if)#exit
```

◆ Proxy2 Connection 설정

```
DMZ(config)#interface ethernet 1/1
DMZ(config-if)#description ##Proxy2_connection##
DMZ(config-if)#no switchport
DMZ(config-if)#ip address 192.168.100.20 255.255.0.0
DMZ(config-if)#no shutdown
DMZ(config-if)#exit
```

◆ ServerFarm(Database) 설정

```
Switch>enable
Switch#configure terminal
Switch(config)#hostname SF_SW
SF_SW(config)#enable secret kgb123
SF_SW(config)#no ip domain lookup
SF_SW(config)#line console 0
SF_SW(config-line)#logging synchronous
SF_SW(config-line)#exec-time 0 0
SF_SW(config-line)#password kgbank123
SF_SW(config-line)#login
SF_SW(config-line)#exit
SF_SW(config)#interface range e 0/0-3, e 1/0-3, e 2/0-3, e 3/0-3
SF_SW(config-if-range)#shutdown
SF_SW(config-if-range)#exit
```

◆ VLAN 설정

```
SF_SW(config)#vlan 200
SF_SW(config-vlan)#name SF
SF_SW(config-vlan)#exit
```

◆ VLAN Access Link설정

```
SF_SW(config)#interface e 0/0
SF_SW(config-if)#description ##SF_SW_SERVER##
SF_SW(config-if)#switchport mode access
SF_SW(config-if)#switchport access vlan 200
SF_SW(config-if)#spanning-tree portfast
SF_SW(config-if)#no shutdown
SF_SW(config-if)#exit
```

◆ VLAN Trunk Link + Port Channel 구성

◆ DMZ – DSW1 설정

```
SF_SW(config)#interface range e 3/0-1
SF_SW(config-if-range)#desc ##DSW1_Trunk##
SF_SW(config-if-range)#swi truencap dot1q
SF_SW(config-if-range)#swi tru allow vlan 200
SF_SW(config-if-range)#swi mode trunk
SF_SW(config-if-range)#swi =nonegotiate
SF_SW(config-if-range)#channel-group 1 mo auto
SF_SW(config-if-range)#no shutdown
SF_SW(config-if-range)#exit
```

◆ DMZ – DSW2 설정

```
SF_SW(config)#interface range e 3/2-3
SF_SW(config-if-range)#desc ##DSW2_Trunk##
SF_SW(config-if-range)#swi truencap dot1q
SF_SW(config-if-range)#swi tru allow vlan 200
SF_SW(config-if-range)#swi mode trunk
SF_SW(config-if-range)#swi =nonegotiate
SF_SW(config-if-range)#channel-group 2mo auto
SF_SW(config-if-range)#no shutdown
SF_SW(config-if-range)#exit
SF_SW(config)#end
SF_SW#wr
```

◆ DB1 Connection 설정

```
DTC(config)#interface ethernet 1/0
DTC(config-if)#description ##DB1_connection##
DTC(config-if)#no switchport
DTC(config-if)#ip address 192.168.100.60 255.255.0.0
DTC(config-if)#no shutdown
DTC(config-if)#exit
```

◆ DB2 Connection 설정

```
DTC(config)#interface ethernet 1/1
DTC(config-if)#description ##DB2_connection##
DTC(config-if)#no switchport
DTC(config-if)#ip address 192.168.100.70 255.255.0.0
DTC(config-if)#no shutdown
DTC(config-if)#exit
```

◆ NAT 및 ACL 접근제어 설정

```
CE(config)#ip nat pool PUBLIC_IP 192.168.1.60 192.168.1.100 netmask 255.255.0.0  
CE(config)#access-list 90 permit 192.168.10.0 0.0.255.255  
CE(config)#access-list 90 permit 192.168.20.0 0.0.255.255  
CE(config)#access-list 90 permit 192.168.30.0 0.0.255.255  
CE(config)#access-list 90 permit 192.168.40.0 0.0.255.255
```

```
CE(config)#ip nat inside source list 90 pool PUBLIC_IP  
CE(config)#interface fa 1/0  
CE(config-if)#ip nat outside  
CE(config-if)#exit
```

```
CE(config)#interface range fa 0/0  
CE(config-if-range)#ip nat inside  
CE(config-if-range)#exit
```

```
CE(config)#interface range fa 0/1  
CE(config-if-range)#ip nat inside  
CE(config-if-range)#exit
```

K G C I N E M A

서 버 구 축

03

01.Ansible 구축

◆ [root@administrator ~]# wget --no-check-certificate 'https://docs.google.com/uc?export=download&id=1p2T0PXJxPQWagUUzqMwPKuddL3sD-5HV' -O start.zip

```
[root@administrator ~]# unzip start.zip -d ~  
[root@administrator ~]# chmod +x ./start.sh  
[root@administrator ~]# ./start.sh
```

```
[root@administrator ~]# vi ./start.sh
```

```
#!/bin/bash

file_id="1jPUlqve9MNUABGx5lZn5k841BfMCm-mE"
wget -O playbook_script.zip "https://drive.google.com/uc?export=download&id=${file_id}"
unzip playbook_script.zip -d ~
chmod 755 dns1.sh dns2.sh web.sh db1.sh db2.sh ssl1.sh ssl2.sh proxy.sh haproxy1.sh haproxy2.sh check.sh

yum -y install centos-release-ansible-29
yum -y install ansible
yum -y install expect
sed -i 's/#host_key_checking = False/host_key_checking = False/g' /etc/ansible/ansible.cfg
mkdir ./ansible
touch ./ansible/inventory
echo '  
[dns1]  
ha-01 ansible_host=192.168.1.10  
[dns2]  
ha-02 ansible_host=192.168.1.20  
[web]  
web-01ansible_host=192.168.1.30  
web-02ansible_host=192.168.1.40  
[was]  
was-01ansible_host=192.168.1.50  
[db1]  
db-01 ansible_host=192.168.1.60  
[db2]  
db-02ansible_host=192.168.1.70  
' >> ./ansible/inventory
```

01. Ansible 구축

```
SECURE_MKD_DNS1=$(expect -c "
set timeout 3
spawn ssh root@192.168.1.10 mkdir -p .ssh
expect \"Are you sure you want to continue connecting (yes/no)?\"
send \"yes\r\"
expect \"root@192.168.1.10's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_MKD_DNS1"
SECURE_MKD_DNS2=$(expect -c "
set timeout 3
spawn ssh root@192.168.1.20 mkdir -p .ssh
expect \"Are you sure you want to continue connecting (yes/no)?\"
send \"yes\r\"
expect \"root@192.168.1.20's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_MKD_DNS2"
SECURE_MKD_WEB1=$(expect -c "
set timeout 3
spawn ssh root@192.168.1.30 mkdir -p .ssh
expect \"Are you sure you want to continue connecting (yes/no)?\"
send \"yes\r\"
expect \"root@192.168.1.30's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_MKD_WEB1"
SECURE_MKD_WEB2=$(expect -c "
set timeout 3
spawn ssh root@192.168.1.40 mkdir -p .ssh
expect \"Are you sure you want to continue connecting (yes/no)?\"
send \"yes\r\"
expect \"root@192.168.1.40's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_MKD_WEB2"
```

01. Ansible 구축

```
SECURE_MKD_WAS=$(expect -c "
set timeout 3
spawn ssh root@192.168.1.50 mkdir -p .ssh
expect \"Are you sure you want to continue connecting (yes/no)?\"
send \"yes\r\"
expect \"root@192.168.1.50's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_MKD_WAS"
SECURE_MKD_DB1=$(expect -c "
set timeout 3
spawn ssh root@192.168.1.60 mkdir -p .ssh
expect \"Are you sure you want to continue connecting (yes/no)?\"
send \"yes\r\"
expect \"root@192.168.1.60's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_MKD_DB1"
SECURE_MKD_DB2=$(expect -c "
set timeout 3
spawn ssh root@192.168.1.70 mkdir -p .ssh
expect \"Are you sure you want to continue connecting (yes/no)?\"
send \"yes\r\"
expect \"root@192.168.1.70's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_MKD_DB2"
SECURE_SSH=$(expect -c "
set timeout 3
spawn ssh-keygen -t rsa
expect \"Enter file in which to save the key (/root/.ssh/id_rsa):\"
send \"\r\"
expect \"Enter passphrase (empty for no passphrase):\"
send \"\r\"
expect \"Enter same passphrase again:\"
send \"\r\"
expect \"The key fingerprint is:\"
send \"\r\"
expect eof
")
echo \"$SECURE_SSH"
```

01. Ansible 구축

```
SECURE_SSH2=$(expect -c "
set timeout 3
spawn scp .ssh/id_rsa.pub root@192.168.1.10:~/ssh/authorized_keys
expect \"root@192.168.1.10's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_SSH2"
SECURE_SSH3=$(expect -c "
set timeout 3
spawn scp .ssh/id_rsa.pub root@192.168.1.20:~/ssh/authorized_keys
expect \"root@192.168.1.20's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_SSH3"
SECURE_SSH4=$(expect -c "
set timeout 3
spawn scp .ssh/id_rsa.pub root@192.168.1.30:~/ssh/authorized_keys
expect \"root@192.168.1.30's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_SSH4"
SECURE_SSH5=$(expect -c "
set timeout 3
spawn scp .ssh/id_rsa.pub root@192.168.1.40:~/ssh/authorized_keys
expect \"root@192.168.1.40's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_SSH5"
SECURE_SSH6=$(expect -c "
set timeout 3
spawn scp .ssh/id_rsa.pub root@192.168.1.50:~/ssh/authorized_keys
expect \"root@192.168.1.50's password:\"
send \"1\r\"
expect eof
")
echo \"$SECURE_SSH6"
```

01. Ansible 구축

```
SECURE_SSH7=$(expect -c "
set timeout 3
spawn scp .ssh/id_rsa.pub root@192.168.1.60:~/ssh/authorized_keys
expect \"root@192.168.1.60's password:\
send \"1\r\
expect eof
")
echo \"$SECURE_SSH7"
SECURE_SSH8=$(expect -c "
set timeout 3
spawn scp .ssh/id_rsa.pub root@192.168.1.70:~/ssh/authorized_keys
expect \"root@192.168.1.70's password:\
send \"1\r\
expect eof
")
echo \"$SECURE_SSH8"
ansible-galaxy collection install ansible.posix
```

02. DNS 구축_dns1.yml

◆ 관리자 서버에서 dns1 플레이북 실행

◆ [root@administrator ~]# ansible-playbook ~/dns1.yml -i ~/.ansible/inventory

[root@administrator ~]# vi ./dns1.yml

```
---
- hosts: dns1
  gather_facts: false
  tasks:
    - name: DNS Package Install
      yum: name=bind state=present
    - name: Service Started
      service: name=named state=started
    - name: ShellScripts Copy
      copy: src=~/dns1.sh dest=~/dns1.sh
    - name: Change script file permissions
      file: dest=~/dns1.sh mode=755
    - name: Run the script file
      shell: ~/dns1.sh
    - name: Firewall ADD
      ansible.posix.firewalld:
        service: dns
        permanent: yes
        state: enabled
    - name: Firewall Reload
      shell: firewall-cmd --reload
```

02. DNS 구축_dns1.sh

❖ [root@administrator ~]# vi ./dns1.sh

```
#!/bin/bash
sed -i 's/listen-on port 53 { 127.0.0.1; };/listen-on port 53 { any; };/ /etc/named.conf
sed -i 's/allow-query { localhost; };/allow-query { any; };/ /etc/named.conf
sed -i 's/recursion yes;/recursion no;/ /etc/named.conf
echo 'zone "kgcinema1.com" IN {
    type master;
    file "kgcinema1.zone";
    also-notify { 192.168.1.20; };
    allow-transfer { 192.168.1.20; };
';
ne "1.168.192.in-addr.arpa" IN {
    type master;
    file "kgcinema1.rev";
    also-notify { 192.168.1.20; };
    allow-transfer { 192.168.1.20; };
';
'>> /etc/named.rfc1912.zones
echo '$TTL 1D
@ IN SOA ns1.kgcinema1.com. root (
    0 ; serial
    60 ; refresh
    1H ; retry
    1W ; expire
    3H ) ; minimum
    IN NS ns1.kgcinema1.com.
    IN NS ns2.kgcinema1.com.
ns1 IN A 192.168.1.10
ns2 IN A 192.168.1.20
www IN A 192.168.1.10
aaa IN CNAME www.kgcinema1.com.
' >> /var/named/kgcinema1.zone
```

02. DNS 구축_dns1.sh

```
echo '$TTL 1D
@ IN SOA ns1.kgcinema1.com. root(
        0      ; serial
        60     ; refresh
        1H     ; retry
        1W     ; expire
        3H )   ; minimum
        IN    NS    ns1.kgcinema1.com.
        IN    NS    ns2.kgcinema1.com.
10    IN    PTR   ns1.kgcinema1.com.
20    IN    PTR   ns2.kgcinema1.com.
10    IN    PTR   www.kgcinema1.com.
' >> /var/named/kgcinema1.rev
chmod 660 /var/named/kgcinema1.*
chown .named /var/named/kgcinema1.*

systemctl restart named
```

02. DNS 구축_dns2.yml

◆ 관리자 서버에서 dns2 플레이북 실행

◆ [root@administrator ~]# ansible-playbook ~/dns2.yml -i ~/.ansible/inventory

[root@administrator ~]# vi ./dns2.yml

```
---
- hosts: dns2
  gather_facts: false
  tasks:
    - name: DNS Package Install
      yum: name=bind state=present
    - name: Service Started
      service: name=named state=started
    - name: ShellScripts Copy
      copy: src=~/dns2.sh dest=~/dns2.sh
    - name: Change script file permissions
      file: dest=~/dns2.sh mode=755
    - name: Run the script file
      shell: ~/dns2.sh
    - name: Firewall ADD
      ansible.posix.firewalld:
        service: dns
        permanent: yes
        state: enabled
    - name: Firewall Reload
      shell: firewall-cmd --reload
```

02. DNS 구축_dns2.sh

❖ [root@administrator ~]# vi ./dns2.sh

```
#!/bin/bash

sed -i 's/listen-on port 53 { 127.0.0.1; };/listen-on port 53 { any; }/' /etc/named.conf
sed -i 's/allow-query { localhost; };/allow-query { any; }/' /etc/named.conf
sed -i 's/recursion yes;/recursion no;/' /etc/named.conf
echo 'zone "kgcinema1.com" IN {
    type slave;
    file "slaves/kgcinema1.zone.slave";
    notify yes;
    masters { 192.168.1.10; };
    masterfile-format text;
};'

zone "1.168.192.in-addr.arpa" IN {
    type slave;
    file "slaves/kgcinema1.rev.slave";
    notify yes;
    masters { 192.168.1.10; };
    masterfile-format text;
};
'>> /etc/named.rfc1912.zones

systemctl restart named
```

03.PROXY 구축_proxy1.yml

◆ 관리자 서버에서 proxy1 플레이북 실행

◆ [root@administrator ~]# ansible-playbook ~/proxy1.yml -i ~/.ansible/inventory

[root@administrator ~]# vi ./proxy1.yml

```
---
- hosts: dns1
  gather_facts: false
  tasks:
    - name: PROXY Package Install
      yum: name=gcc state=present
    - name: PROXY Package Install
      yum: name=openssl state=present
    - name: PROXY Package Install
      yum: name=openssl-devel state=present
    - name: PROXY Package Install
      yum: name=systemd-devel state=present
    - name: ShellScripts Copy
      copy: src=~/proxy.sh dest=~/proxy.sh
    - name: Change script file permissions
      file: dest=~/proxy.sh mode=755
    - name: Run the script file
      shell: ~/proxy.sh
    - name: HAProxy Package Install
      yum: name=keepalived state=present
    - name: ShellScripts Copy
      copy: src=~/haproxy1.sh dest=~/haproxy1.sh
    - name: Change script file permissions
      file: dest=~/haproxy1.sh mode=755
    - name: Run the script file
      shell: ~/haproxy1.sh
    - name: Expect Package Install
      yum: name=expect state=present
    - name: ShellScripts Copy
      copy: src=~/ssl1.sh dest=~/ssl1.sh
```

03.PROXY 구축_proxy1.yml

```
- name: Change script file permissions
  file: dest=~/ssl1.sh mode=755
- name: Run the script file
  shell: ~/ssl1.sh
- name: Firewall ADD
  ansible.posix.firewalld:
    service: https
    permanent: yes
    state: enabled
- name: Firewall Reload
  shell: firewall-cmd --reload
```

03.PROXY 구축_proxy.sh

◆ [root@administrator ~]# vi ./proxy.sh

```
#!/bin/bash

mkdir /HAproxy
cd /HAproxy

wget http://www.haproxy.org/download/2.3/src/haproxy-2.3.10.tar.gz
tar xvzf haproxy-2.3.10.tar.gz
cd haproxy-2.3.10/

make TARGET=linux-glibc USE_OPENSSL=1 USE_SYSTEMD=1
make install
curl "https://git.haproxy.org/?p=haproxy-2.3.git;a=blob_plain;f=contrib/systemd/haproxy.service.in;" -o /etc/systemd/system/haproxy.service

sed -i 's/ExecStartPre=@SBINDIR@\!/haproxy -Ws -f $CONFIG -c -q $EXTRAOPTS/ExecStartPre=/usr \
\!local\!sbin\!haproxy -Ws -f $CONFIG -c -q $EXTRAOPTS/g' /etc/systemd/system/haproxy.service
sed -i 's/ExecStart=@SBINDIR@\!/haproxy -Ws -f $CONFIG -p $PIDFILE $EXTRAOPTS/ExecStart=/usr \
\!local\!sbin\!haproxy -Ws -f $CONFIG -p $PIDFILE $EXTRAOPTS/g' /etc/systemd/system/haproxy.service
sed -i 's/ExecReload=@SBINDIR@\!/haproxy -Ws -f $CONFIG -c -q $EXTRAOPTS/ExecReload=/usr\!\
ocal\!sbin\!haproxy -Ws -f $CONFIG -c -q $EXTRAOPTS/g' /etc/systemd/system/haproxy.service

mkdir /etc/haproxy
mkdir /etc/haproxy/certs
mkdir /etc/haproxy/errors
mkdir /var/log/haproxy
cd ./examples/errorfiles/

cp ./*.http /etc/haproxy/errors/

cd ~

useradd -c "HAproxy Daemon User" -s /sbin/nologin haproxy

touch /etc/rsyslog.d/haproxy.conf
echo '
$ModLoad imudp
$UDPServerAddress 127.0.0.1
$UDPServerRun 514
local0.* /var/log/haproxy/haproxy-traffic.log
' >> /etc/rsyslog.d/haproxy.conf
```

03.PROXY 구축_proxy.sh

```
firewall-cmd --permanent --add-port=514/udp
firewall-cmd --reload

touch /etc/logrotate.d/haproxy
echo '
/var/log/haproxy/*.log {
    daily
    rotate 30
    create 0600 root root
    compress
    notifempty
    missingok
    sharedscripts
    postrotate
        /bin/systemctl restart rsyslog.service > /dev/null 2>/dev/null || true
    endscript
}
' >> /etc/logrotate.d/haproxy

touch /etc/haproxy/haproxy.cfg
echo '
global
    daemon
    maxconn 4000
    user haproxy
    group haproxy
    log 127.0.0.1:514 local0
defaults
    mode http
    option redispatch
    retries 3
    log global
    option httplog
    option dontlognull
    option dontlog-normal
    option http-server-close
    option forwardfor
    maxconn 2000
    timeout connect 10s
    timeout http-request 10s
    timeout http-keep-alive 10s
    timeout client 1m
    timeout server 1m
    timeout queue 1m
'
```

03.PROXY 구축_proxy.sh

```
errorfile 400 /etc/haproxy/errors/400.http
errorfile 403 /etc/haproxy/errors/403.http
errorfile 408 /etc/haproxy/errors/408.http
errorfile 500 /etc/haproxy/errors/500.http
errorfile 502 /etc/haproxy/errors/502.http
errorfile 503 /etc/haproxy/errors/503.http
errorfile 504 /etc/haproxy/errors/504.http

listen stats
    bind *:9000
    stats enable
    stats realm Haproxy Stats Page
    stats uri /
    stats auth admin:haproxy1

frontend proxy
    bind *:80
    default_backend WEB_SRV_list

backend WEB_SRV_list
    balance roundrobin
    option httpchk HEAD /
    http-request set-header X-Forwarded-Port %[dst_port]
    cookie SRVID insert indirect nocache maxlife 10m
    server WEB_01 192.168.1.30:80 maxconn 1000 cookie WEB_01 check inter 3000 fall 5 rise 3
    server WEB_02 192.168.1.40:80 maxconn 1000 cookie WEB_02 check inter 3000 fall 5 rise 3

' >> /etc/haproxy/haproxy.cfg

haproxy -f /etc/haproxy/haproxy.cfg -c

systemctl start haproxy
systemctl enable haproxy
firewall-cmd --permanent --add-service=http
firewall-cmd --permanent --add-port=9000/tcp
firewall-cmd --reload
```

03.PROXY 구축_haproxy1.sh

❖ [root@administrator ~]# vi ./haproxy1.sh

```
#!/bin/bash

echo net.ipv4.ip_nonlocal_bind=1 >> /etc/sysctl.conf
sysctl -p

sed -i '1,1000d' /etc/keepalived/keepalived.conf
cat /etc/keepalived/keepalived.conf
echo '
global_defs {
    router_id HA_01
}

vrrp_script HA_Check {
    script "killall -0 haproxy"
    interval 1
    rise 3
    fall 3
    weight 2
}

vrrp_instance HAGroup_1 {
    state MASTER
    interface ens32
    garp_master_delay 5
    virtual_router_id 51
    priority 110
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass test123
    }
    virtual_ipaddress {
        192.168.1.150
    }
    track_script {
        HA_Check
    }
}
' >> /etc/keepalived/keepalived.conf
```

03.PROXY 구축_haproxy1.sh

```
firewall-cmd --direct --add-rule ipv4 filter INPUT 1 -i ens32 -d 224.0.0.18 -p vrrp -j ACCEPT  
firewall-cmd --direct --add-rule ipv4 filter OUTPUT 1 -o ens32 -d 224.0.0.18 -p vrrp -j ACCEPT  
firewall-cmd --runtime-to-permanent  
firewall-cmd --direct --get-all-rules  
systemctl start keepalived  
systemctl enable keepalived
```

03.PROXY 구축_ssl1.sh

❖ [root@administrator ~]# vi ./ssl1.sh

```
#!/bin/bash

openssl genrsa -out /etc/haproxy/certs/ha01.key 2048

SECURE_SSL=$(expect -c "
set timeout 3
spawn openssl req -new -key /etc/haproxy/certs/ha01.key -out /etc/haproxy/certs/ha01.csr
expect \"Country Name (2 letter code) \[XX\]:\
send \"KR\r\
expect \"State or Province Name (full name) \[\]:\
send \"Seoul\r\
expect \"Locality Name (eg, city) \[Default City\]:\
send \"Jonglo\r\
expect \"Organization Name (eg, company) \[Default Company Ltd\]:\
send \"KGCINEMA\r\
expect \"Organizational Unit Name (eg, section) \[\]:\
send \"CloudTeam\r\
expect \"Common Name (eg, your name or your server's hostname) \[\]:\
send \"www.kgcinema1.com\r\
expect \"Email Address \[\]:\
send \"root@kgcinema1.com\r\
expect \"A challenge password \[\]:\
send \"\r\
expect \"An optional company name \[\]:\
send \"\r\
expect eof
")

echo "$SECURE_SSL"
```

03.PROXY 구축_ssl1.sh

```
openssl x509 -req -days 365 -in /etc/haproxy/certs/ha01.csr -signkey /etc/haproxy/certs/ha01.key -out /etc/haproxy/certs/ha01.crt
cd /etc/haproxy/certs
cat ha01.crt ha01.key > ha01_ssl.crt
mv ha01.* /backup
cd ~

perl -p -i -e '$.==8 and print "ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets\n" /etc/haproxy/haproxy.cfg
perl -p -i -e '$.==8 and print "ssl-default-bind-ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256\n" /etc/haproxy/haproxy.cfg

perl -p -i -e '$.==47 and print "http-request redirect scheme https code 308 unless { ssl_fc }\n" /etc/haproxy/haproxy.cfg
perl -p -i -e '$.==47 and print "bind *:443 ssl crt /etc/haproxy/certs/ha01_ssl.crt\n" /etc/haproxy/haproxy.cfg

haproxy -f /etc/haproxy/haproxy.cfg -c
systemctl restart haproxy
```

03.PROXY 구축_proxy2.yml

◆ 관리자 서버에서 proxy2 플레이북 실행

◆ [root@administrator ~]# ansible-playbook ~/proxy2.yml -i ~/.ansible/inventory

[root@administrator ~]# vi ./proxy2.yml

```
---
- hosts: dns2
  gather_facts: false
  tasks:
    - name: PROXY Package Install
      yum: name=gcc state=present
    - name: PROXY Package Install
      yum: name=openssl state=present
    - name: PROXY Package Install
      yum: name=openssl-devel state=present
    - name: PROXY Package Install
      yum: name=systemd-devel state=present
    - name: ShellScripts Copy
      copy: src=~/proxy.sh dest=~/proxy.sh
    - name: Change script file permissions
      file: dest=~/proxy.sh mode=755
    - name: Run the script file
      shell: ~/proxy.sh
    - name: HAProxy Package Install
      yum: name=keepalived state=present
    - name: ShellScripts Copy
      copy: src=~/haproxy2.sh dest=~/haproxy2.sh
    - name: Change script file permissions
      file: dest=~/haproxy2.sh mode=755
    - name: Run the script file
      shell: ~/haproxy2.sh
```

03.PROXY 구축_proxy2.yml

```
- name: Expect Package Install
  yum: name=expect state=present
- name: ShellScripts Copy
  copy: src=~/ssl2.sh dest=~/ssl2.sh
- name: Change script file permissions
  file: dest=~/ssl2.sh mode=755
- name: Run the script file
  shell: ~/ssl2.sh
- name: Firewall ADD
  ansible.posix.firewalld:
    service: https
    permanent: yes
    state: enabled
- name: Firewall Reload
  shell: firewall-cmd --reload
```

03.PROXY 구축_haproxy2.sh

◆ [root@administrator ~]# vi ./haproxy2.sh

```
#!/bin/bash

echo net.ipv4.ip_nonlocal_bind=1 >> /etc/sysctl.conf
sysctl -p

sed -i '1,1000d' /etc/keepalived/keepalived.conf
cat /etc/keepalived/keepalived.conf
echo '
global_defs {
    router_id HA_02
}

vrrp_script HA_Check {
    script "killall -0 haproxy"
    interval 1
    rise 3
    fall 3
    weight 2
}

vrrp_instance HAGroup_1 {
    state BACKUP
    interface ens32
    garp_master_delay 5
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass test123
    }
    virtual_ipaddress {
        192.168.1.150
    }
    track_script {
        HA_Check
    }
}
' >> /etc/keepalived/keepalived.conf
```

03.PROXY 구축_haproxy2.sh

```
firewall-cmd --direct --add-rule ipv4 filter INPUT 1 -i ens32 -d 224.0.0.18 -p vrrp -j ACCEPT
firewall-cmd --direct --add-rule ipv4 filter OUTPUT 1 -o ens32 -d 224.0.0.18 -p vrrp -j ACCEPT
firewall-cmd --runtime-to-permanent
firewall-cmd --direct --get-all-rules
systemctl start keepalived
systemctl enable keepalived
```

03.PROXY 구축_ssl2.sh

◆ [root@administrator ~]# vi ./ssl2.sh

```
#!/bin/bash

openssl genrsa -out /etc/haproxy/certs/ha02.key 2048

SECURE_SSL=$(expect -c "
set timeout 3
spawn openssl req -new -key /etc/haproxy/certs/ha02.key -out /etc/haproxy/certs/ha02.csr
expect \"Country Name (2 letter code) \[XX\]:\""
send \"KR\r\"
expect \"State or Province Name (full name) \[\]:\""
send \"Seoul\r\"
expect \"Locality Name (eg, city) \[Default City\]:\""
send \"Jonglo\r\"
expect \"Organization Name (eg, company) \[Default Company Ltd\]:\""
send \"KG CINEMA\r\"
expect \"Organizational Unit Name (eg, section) \[\]:\""
send \"CloudTeam\r\"
expect \"Common Name (eg, your name or your server's hostname) \[\]:\""
send \"www.kgcinema1.com\r\"
expect \"Email Address \[\]:\""
send \"root@kgcinema1.com\r\"
expect \"A challenge password \[\]:\""
send \"\r\"
expect \"An optional company name \[\]:\""
send \"\r\"
expect eof
")

echo "$SECURE_SSL"

openssl x509 -req -days 365 -in /etc/haproxy/certs/ha02.csr -signkey /etc/haproxy/certs/ha02.key -out /etc/haproxy/certs/ha02.crt
cd /etc/haproxy/certs
cat ha02.crt ha02.key > ha02_ssl.crt
mkdir /backup
mv ha02.* /backup
cd ~

cat /etc/haproxy/certs/ha02_ssl.crt
```

03.PROXY 구축_ssl2.sh

```
perl -p -i -e '$.==8 and print "ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets\n"' /etc/haproxy/haproxy.cfg
```

```
perl -p -i -e '$.==8 and print "ssl-default-bind-ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256\n"' /etc/haproxy/haproxy.cfg
```

```
perl -p -i -e '$.==47 and print "http-request redirect scheme https code 308 unless { ssl_fc }\n"' /etc/haproxy/haproxy.cfg
```

```
perl -p -i -e '$.==47 and print "bind *:443 ssl crt /etc/haproxy/certs/ha02_ssl.crt\n"' /etc/haproxy/haproxy.cfg
```

```
haproxy -f /etc/haproxy/haproxy.cfg -c  
systemctl restart haproxy
```

04.WEB 구축_web.yml

◆ 관리자 서버에서 web 플레이북 실행

◆ [root@administrator ~]# ansible-playbook ~/web.yml -i ~/.ansible/inventory

[root@administrator ~]# vi ./web.yml

```
---
- hosts: web
  gather_facts: false
  tasks:
    - name: WEB Package Install
      yum: name=httpd state=present
    - name: WEB Package Install
      yum: name=httpd-manual state=present
    - name: WEB Package Install
      yum: name=httpd-devel state=present
    - name: Service Started
      service: name=httpd state=started
    - name: ShellScripts Copy
      copy: src=~/web.sh dest=~/web.sh
    - name: Change script file permissions
      file: dest=~/web.sh mode=755
    - name: Run the script file
      shell: ~/web.sh
    - name: Firewall ADD
      ansible.posix.firewalld:
        service: http
        permanent: yes
        state: enabled
    - name: Firewall Reload
      shell: firewall-cmd --reload
```

04.WEB 구축_web.sh

❖ [root@administrator ~]# vi ./web.sh

```
#!/bin/bash
sed -i 's/User apache/User nobody/g' /etc/httpd/conf/httpd.conf
sed -i 's/Group apache/Group nobody/g' /etc/httpd/conf/httpd.conf
sed -i 's/#ServerName www.example.com:80/ServerName www.kgcinema1.com:80/
g' /etc/httpd/conf/httpd.conf
sed -i 's/DirectoryIndex index.html/DirectoryIndex home.html index.html/g' /etc/http
d/conf/httpd.conf

file_id="1xgeKSbglJg-UPzngtckMnrvqN2zdyAvv"
wget -O html.zip "https://drive.google.com/uc?export=download&id=${file_id}"
unzip html.zip -d /var/www/html/

systemctl restart httpd
```

05.WAS 구축_was.yml

◆ 관리자 서버에서 was 플레이북 실행

◆ [root@administrator ~]# ansible-playbook ~/was.yml -i ~/.ansible/inventory

[root@administrator ~]# vi ./was.yml

```
---
- hosts: was
  gather_facts: false
  tasks:
    - name: WEB Package Install
      yum: name=httpd state=present
    - name: WEB Package Install
      yum: name=httpd-manual state=present
    - name: WEB Package Install
      yum: name=httpd-devel state=present
    - name: PHP Package Install
      yum: name/php state=present
    - name: PHP Package Install
      yum: name/php-mysql state=present
    - name: PHP Package Install
      yum: name/php-mbstring state=present
    - name: PHP Package Install
      yum: name/php-pdo state=present
    - name: PHP Package Install
      yum: name/php-gd state=present
```

05.WAS 구축_was.yml

```
- name: Service Started
  service: name=httpd state=started
- name: ShellScripts Copy
  copy: src=~/web.sh dest=~/web.sh
- name: Change script file permissions
  file: dest=~/web.sh mode=755
- name: Run the script file
  shell: ~/web.sh
- name: Firewall ADD
  ansible.posix.firewalld:
    service: http
    permanent: yes
    state: enabled
- name: Firewall Reload
  shell: firewall-cmd --reload
```

05.WAS 구축_was(=web).sh

❖ [root@administrator ~]# vi ./was.sh

```
#!/bin/bash
sed -i 's/User apache/User nobody/g' /etc/httpd/conf/httpd.conf
sed -i 's/Group apache/Group nobody/g' /etc/httpd/conf/httpd.conf
sed -i 's/#ServerName www.example.com:80/ServerName www.kgcinema1.com:80/
g' /etc/httpd/conf/httpd.conf
sed -i 's/DirectoryIndex index.html/DirectoryIndex home.html index.html/g' /etc/http
d/conf/httpd.conf

file_id="1xgeKSbglJg-UPzngtckMnrvqN2zdyAvv"
wget -O html.zip "https://drive.google.com/uc?export=download&id=${file_id}"
unzip html.zip -d /var/www/html/

systemctl restart httpd
```

06.DB 구축_db1.yml

◆ 관리자 서버에서 db1플레이북 실행

◆ [root@administrator ~]# ansible-playbook ~/db1.yml -i ~/.ansible/inventory

[root@administrator ~]# vi ./db1.yml

```
---
- hosts: db1
  gather_facts: false
  tasks:
    - name: DB Package Install
      yum: name=mariadb-server state=present
    - name: DB Package Install
      yum: name=expect state=present
    - name: Service Started
      service: name=mariadb state=started
    - name: ShellScripts Copy
      copy: src=~/db1.sh dest=~/db1.sh
    - name: Change script file permissions
      file: dest=~/db1.sh mode=755
    - name: Run the script file
      shell: ~/db1.sh
    - name: Firewall ADD
      ansible.posix.firewalld:
        service: mysql
        permanent: yes
        state: enabled
    - name: Firewall Reload
      shell: firewall-cmd --reload
```

06.DB 구축_db1.sh

❖ [root@administrator ~]# vi ./db1.sh

```
#!/bin/bash

perl -p -i -e '$.==2 and print "character-set-server=utf8\n"' /etc/my.cnf

systemctl start mariadb
systemctl enable mariadb

SECURE_MYSQL=$(expect -c "
set timeout 3
spawn mysql_secure_installation
expect \"Enter current password for root (enter for none):\""
send \"\r\"
expect \"Set root password?\"
send \"y\r\"
expect \"New password:\""
send \"itbank\r\"
expect \"Re-enter new password:\""
send \"itbank\r\"
expect \"Remove anonymous users?\"
send \"y\r\"
expect \"Disallow root login remotely?\""
send \"y\r\"
expect \"Remove test database and access to it?\""
send \"y\r\"
expect \"Reload privilege tables now?\"
send \"y\r\"
expect eof
")
echo "$SECURE_MYSQL"
```

06.DB 구축_db1.sh

```
firewall-cmd --permanent --add-service=mysql  
firewall-cmd --reload  
  
mysql -u root -pitbank -e "create database webdb;"  
mysql -u root -pitbank -e "grant all privileges on webdb.* to itbank@'192.168.1.%' identified  
by 'itbank';"  
mysql -u root -pitbank webdb -e "create table member(No int auto_increment primary key, I  
D varchar(50) not null, PW varchar(100) not null, Name varchar(50) not null, phone varchar(5  
0) not null, addr varchar(50) not null, Mail varchar(50) not null);"  
mysql -u root -pitbank webdb -e "create table book(userid varchar(50) not null, movie varcha  
r(100) not null, cinema varchar(50) not null, date varchar(50) not null, time varchar(50) not n  
ull, age varchar(50) not null, person varchar(50) not null);"  
  
mysql -u root -pitbank -e "grant replication slave on *.* to Rep_user@'%' identified by 'itban  
k';"  
mysql -u root -pitbank -e "flush privileges;"  
  
perl -p -i -e '$.==2 and print "server-id=1\n"' /etc/my.cnf  
perl -p -i -e '$.==2 and print "log-bin=mysql-bin\n"' /etc/my.cnf  
  
systemctl restart mariadb  
  
mysqldump -u root -pitbank --all-database > all_backup.sql
```

06.DB 구축_db2.yml

◆ 관리자 서버에서 db2플레이북 실행

◆ [root@administrator ~]# ansible-playbook ~/db2.yml -i ~/.ansible/inventory

[root@administrator ~]# vi ./db2.yml

```
---
- hosts: db2
  gather_facts: false
  tasks:
    - name: DB Package Install
      yum: name=mariadb-server state=present
    - name: DB Package Install
      yum: name=expect state=present
    - name: Service Started
      service: name=mariadb state=started
    - name: ShellScripts Copy
      copy: src=~/db2.sh dest=~/db2.sh
    - name: Change script file permissions
      file: dest=~/db2.sh mode=755
    - name: Run the script file
      shell: ~/db2.sh
    - name: Firewall ADD
      ansible.posix.firewalld:
        service: mysql
        permanent: yes
        state: enabled
    - name: Firewall Reload
      shell: firewall-cmd --reload
```

06.DB 구축_db2.sh

❖ [root@administrator ~]# vi ./db1.sh

```
#!/bin/bash

perl -p -i -e '$.==2 and print "character-set-server=utf8\n"' /etc/my.cnf
perl -p -i -e '$.==2 and print "replicate-do-db='webdb'\n"' /etc/my.cnf
perl -p -i -e '$.==2 and print "server-id=2\n"' /etc/my.cnf

SECURE SCP=$(expect -c "
set timeout 3
spawn scp root@192.168.1.60:/root/all_backup.sql ./
expect \"Are you sure you want to continue connecting (yes/no)?\""
send \"yes\r\"
expect \"root@192.168.1.60's password:\""
send \"1\r\"
expect eof
")
echo "$SECURE SCP"

systemctl start mariadb
systemctl enable mariadb

firewall-cmd --permanent --add-service=mysql
firewall-cmd --reload
```

06.DB 구축_db2.sh

```
SECURE_MYSQL=$(expect -c "
set timeout 3
spawn mysql_secure_installation
expect \"Enter current password for root (enter for none):\""
send \"\r\"
expect \"Set root password?\"
send \"y\r\"
expect \"New password:\""
send \"itbank\r\"
expect \"Re-enter new password:\""
send \"itbank\r\"
expect \"Remove anonymous users?\""
send \"y\r\"
expect \"Disallow root login remotely?\""
send \"y\r\"
expect \"Remove test database and access to it?\""
send \"y\r\"
expect \"Reload privilege tables now?\""
send \"y\r\"
expect eof
")
echo "$SECURE_MYSQL"

mysql -u root -pitbank < all_backup.sql
mysql -u root -pitbank -e "change master to master_host='192.168.1.60',master_user='Rep_user',
master_password='itbank',master_log_file='mysql-bin.000001',master_log_pos=245;"
systemctl restart mariadb
```

K G C I N E M A

모니터링

04

01. 모니터링 도구_ 가비아

◆ 가비아 모니터링 도구 설치 (운영하고 있는 서버에 다 설치해 준다)

```
rpm -Uvh http://monrepo.gabia.com/repo/centos/noarch/gabmon-repo-1.0.0-2.noarch.rpm
```

```
yum -y install gabia_mond
```

```
sudo env PATH=$PATH:/usr/local/gabia_mond/bin gabia_mond --start --userkey
```

```
User Key : xxxxxxxxxxxxxxxxxxxxxxxxx
```

The screenshot shows the Gabia Monitoring tool's web interface. On the left is a sidebar with navigation links: 이벤트 (Events), 모니터링 설치 (Monitoring Installation), 환경설정 (Environment Configuration), and 고객센터 (Customer Support). The main area is titled '서버 리스트' (Server List) and displays a table of monitored servers. The table columns are: 위험도 (Risk Level), 서버명 (Server Name), IP, CPU, 메모리 (Memory), Swap, and 디스크 (Disk). The table contains the following data:

| 위험도 | 서버명 | IP | CPU | 메모리 | Swap | 디스크 |
|-----|--------|--------------|-------|--------|--------|--------|
| �� | ha-01 | 192.168.1.10 | 2.1% | 61.81% | 66.96% | 35.35% |
| �� | ha-02 | 192.168.1.20 | 1.82% | 32.65% | 9.14% | 29.12% |
| 绿 | web-01 | 192.168.1.30 | 1.24% | 40.15% | 2.69% | 28.07% |
| 绿 | web-02 | 192.168.1.40 | 1.28% | 40.32% | 3.21% | 28.08% |
| 红 | was | 192.168.1.50 | 1.85% | 40.94% | 5.28% | 33.64% |
| 绿 | db01 | 192.168.1.60 | 1.17% | 40.76% | 0.00% | 30.71% |
| 绿 | db02 | 192.168.1.70 | 1.42% | 40.83% | 0.29% | 33.46% |

At the bottom left of the interface, there is a copyright notice: ©Gabia Inc. All Rights Reserved.

1-1) DNS,PROXY MONITORING

ha-01 Server Information:

- Server Status:** 0일 09시간 10분
- Server Name:** ha-01
- Host IP:** 192.168.1.10
- MAC Address:** 00:0c:29:c0:2e
- OS Version:** CentOS Linux release 7.3.1611 (Core) x86_64
- Event Log:** 0.81.98
- Monitoring Service:** 변경

ha-02 Server Information:

- Server Status:** 0일 09시간 16분
- Server Name:** ha-02
- Host IP:** 192.168.1.20
- MAC Address:** 00:0c:29:78:46
- OS Version:** CentOS Linux release 7.3.1611 (Core) x86_64
- Event Log:** 0.81.98
- Monitoring Service:** 변경

1-2) WEB MONITORING

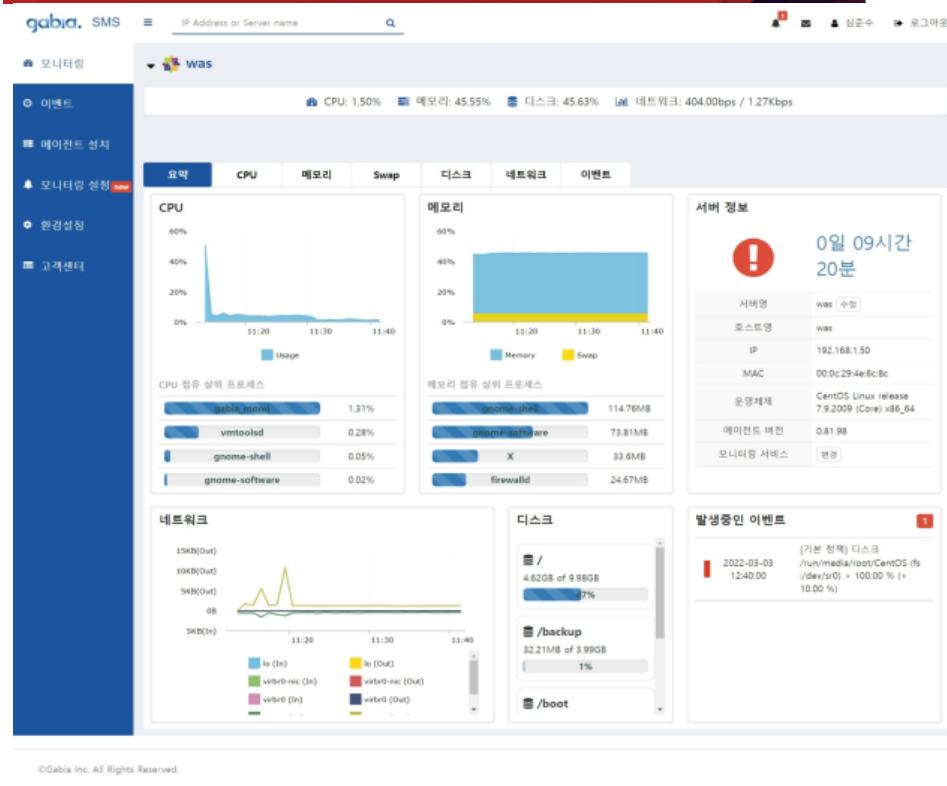
web-01 Server Information:

- Server Status:** 0일 09시간 14분
- Server Name:** web-01
- Host IP:** 192.168.1.30
- MAC Address:** 00:0c:29:57:b5:49
- OS Version:** CentOS Linux release 7.3.1611 (Core) x86_64
- Event Log:** 0.81.98
- Monitoring Service:** 변경

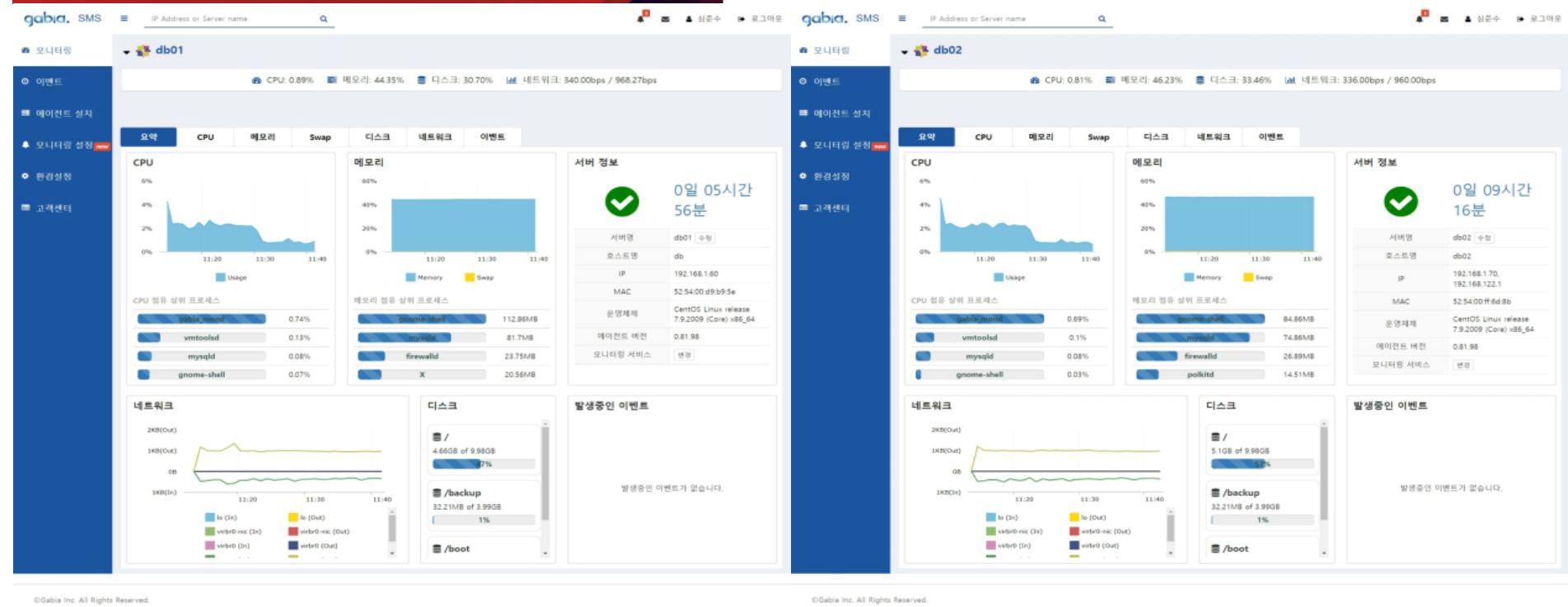
web-02 Server Information:

- Server Status:** 0일 09시간 19분
- Server Name:** web-02
- Host IP:** 192.168.1.40
- MAC Address:** 00:0c:29:a1:88:be
- OS Version:** CentOS Linux release 7.3.1611 (Core) x86_64
- Event Log:** 0.81.98
- Monitoring Service:** 변경

1-3) WAS MONITORING



1-4) DB MONITORING



02. 로그 관리 도구 _goaccess

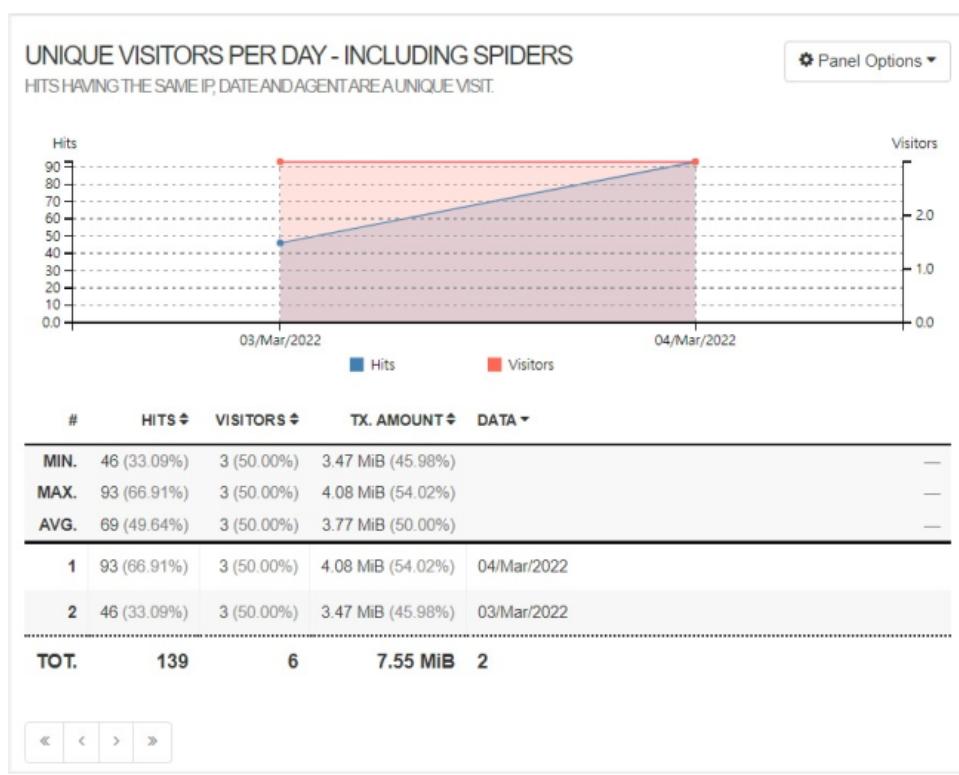
◆ goaccess 설치 (was 서버에 설치, apache가 설치되었는곳에서만 작동한다)

```
yum install ncurses-devel glib2-devel geoip-devel gcc  
wget https://tar.goaccess.io/goaccess-1.5.5.tar.gz  
wget https://tar.goaccess.io/goaccess-1.5.5.tar.gz `--no-check-certificate  
tar -xzvf goaccess-1.5.5.tar.gz  
cd goaccess-1.5.5/  
.configure --enable-utf8 --enable-geoip=legacy  
make  
make install  
goaccess -f /var/log/httpd/access_log
```

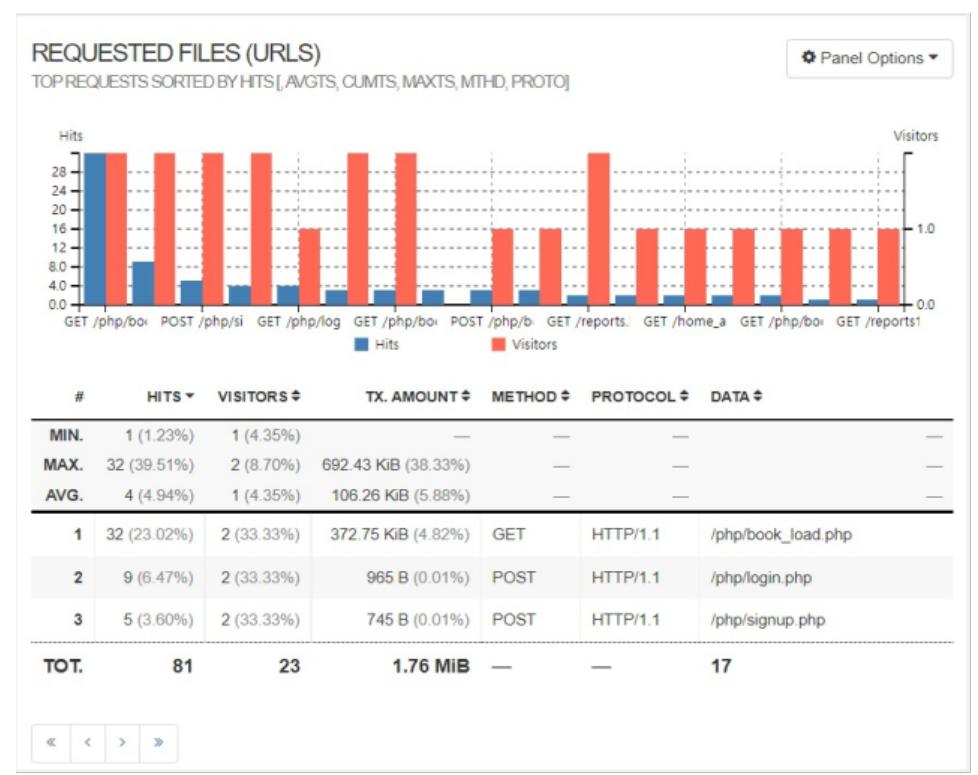
```
vi /usr/local/etc/goaccess/goaccess.conf  
#log-format COMBINED  
-> log-format COMBINED
```

```
goaccess -f /var/log/httpd/access_log > reports.html  
mv ./reports.html /var/www/html
```

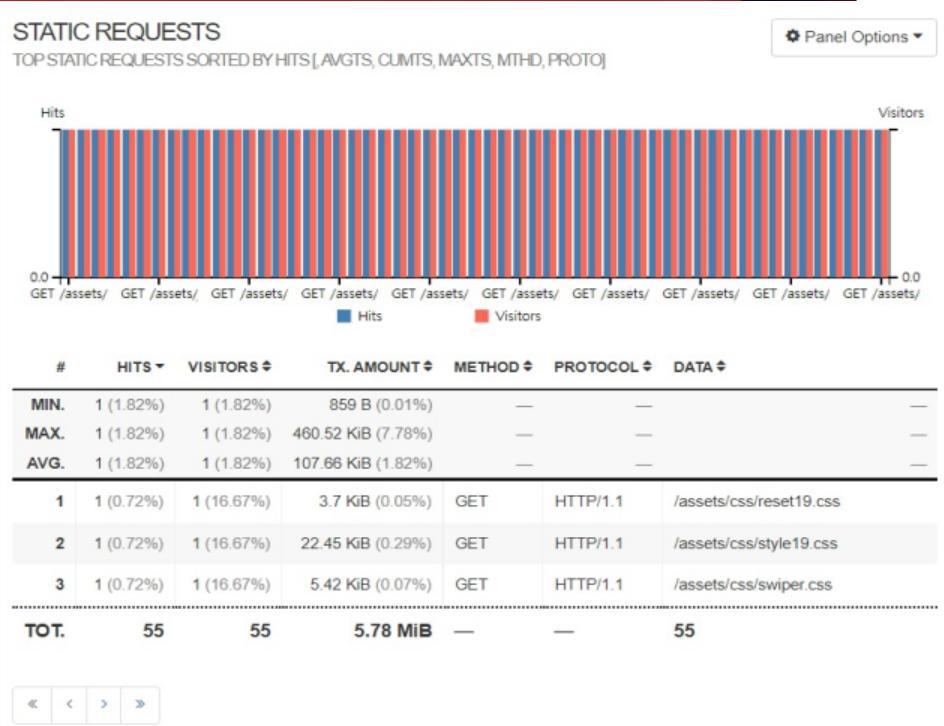
2-1) 일자별 방문자 정보



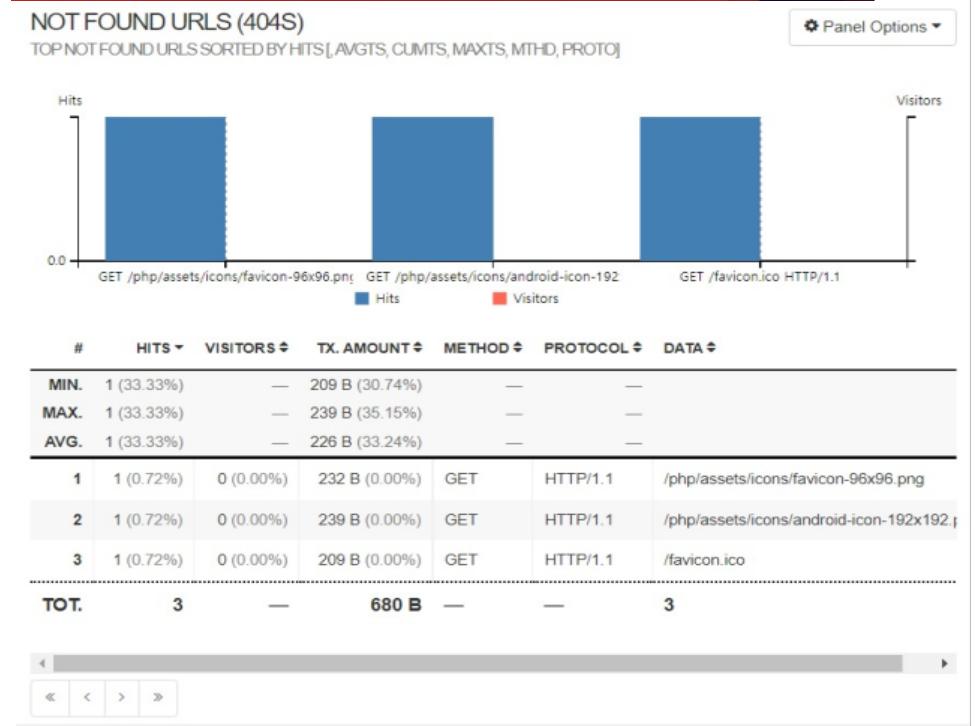
2-2) 요청 URL 정보



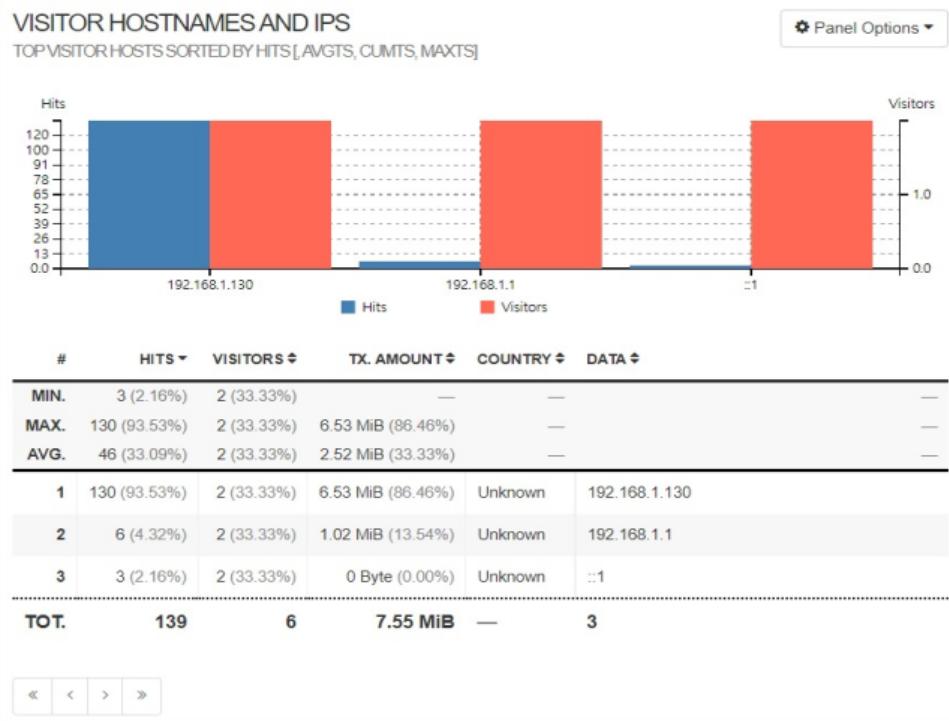
2-3) 정적 리소스 요청 정보



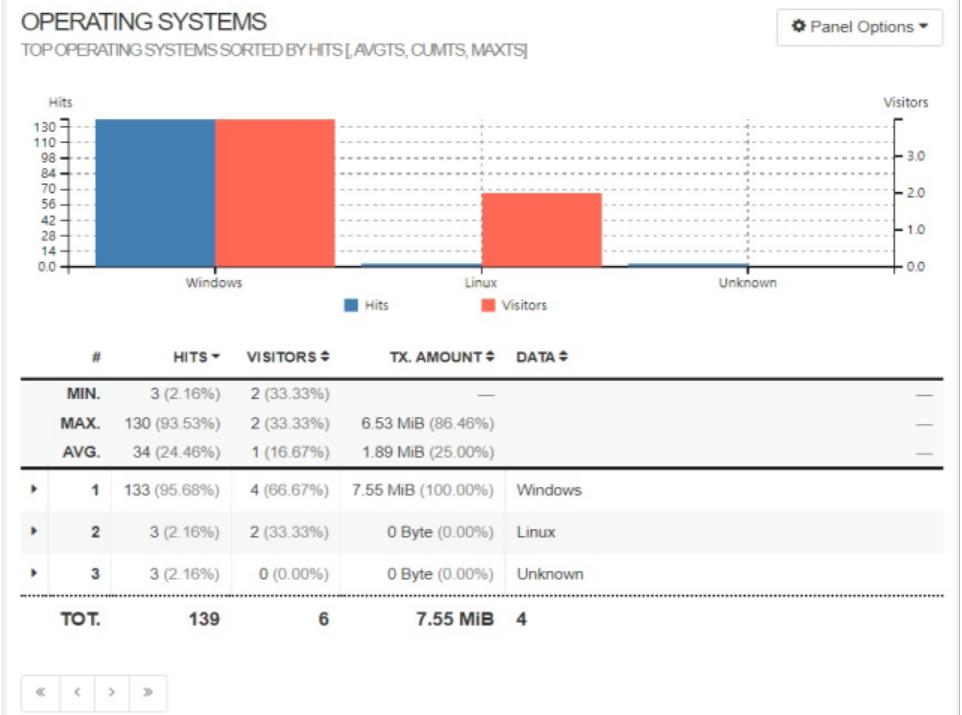
2-4) 404 요청 정보



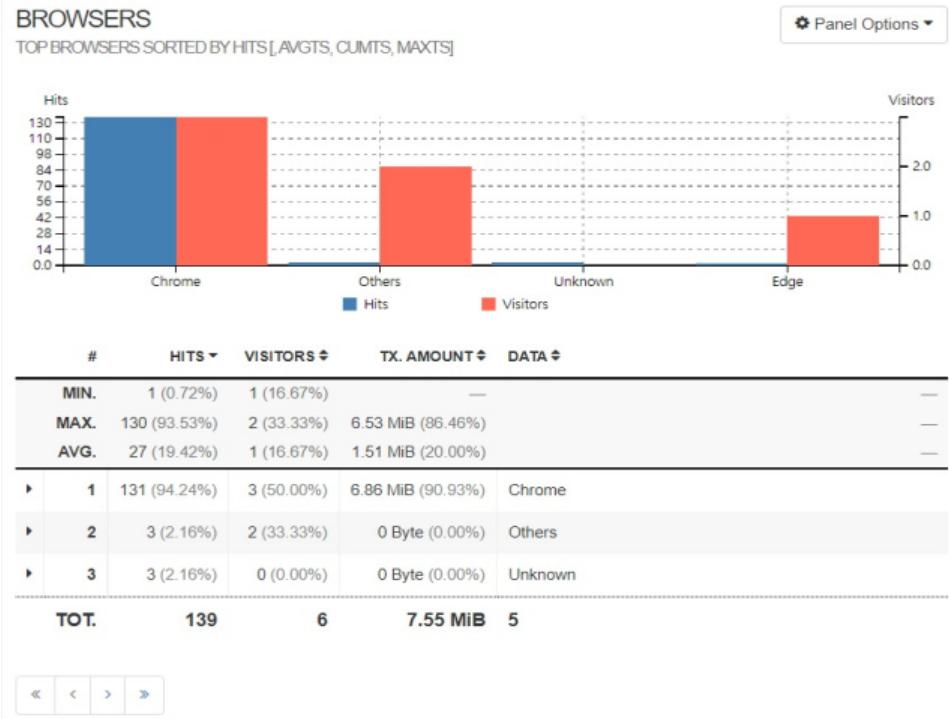
2-5) 방문자별 정보



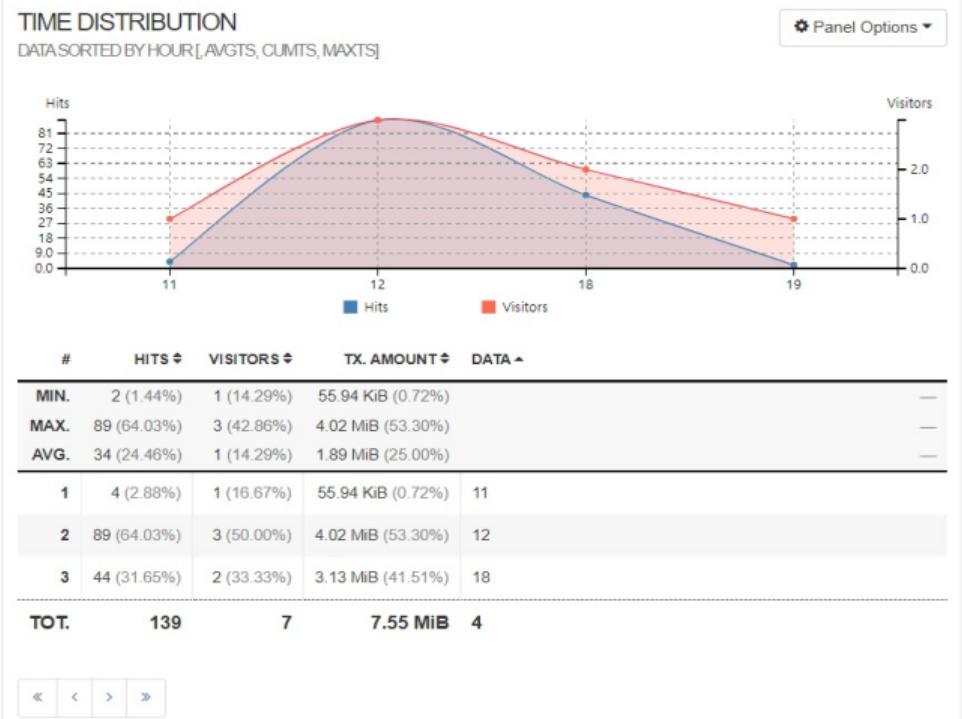
2-6) OS별 정보



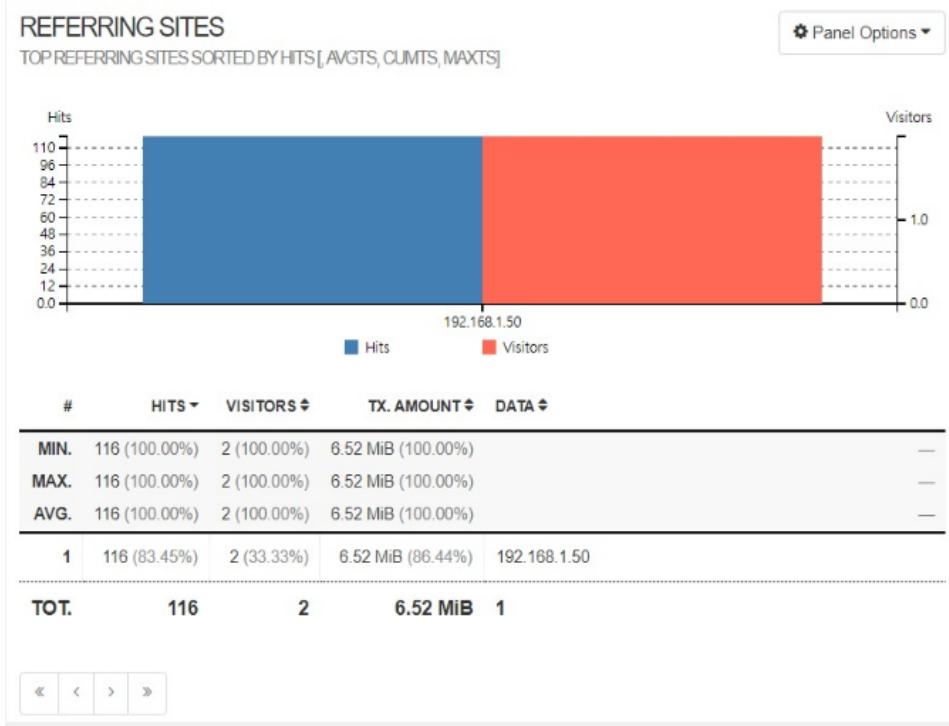
2-7) 브라우저별 정보



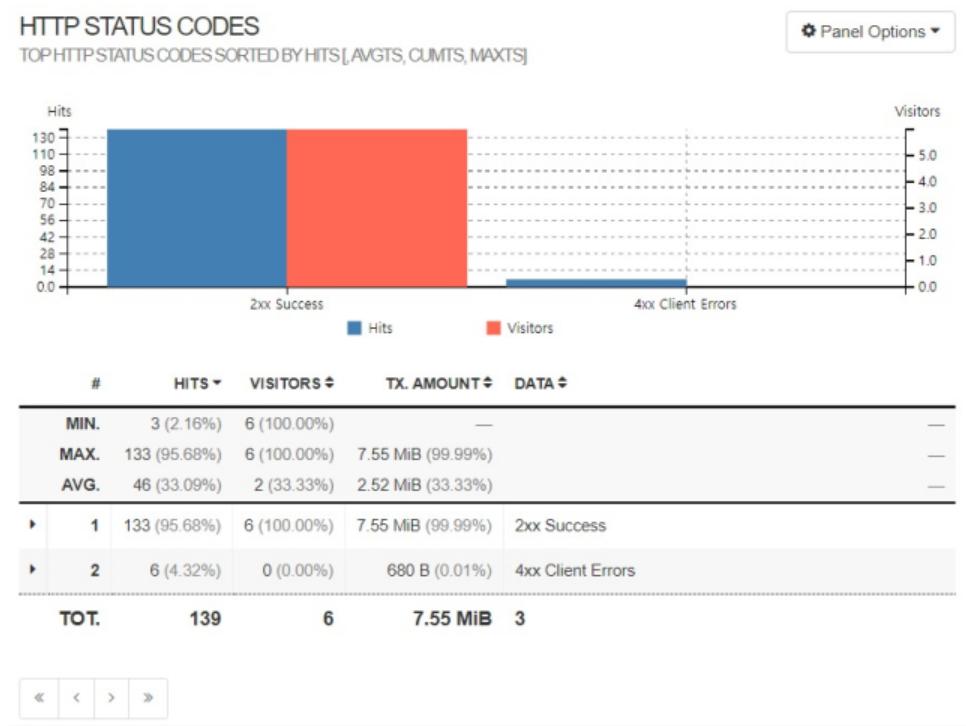
2-8) 시간별(00~23) 정보



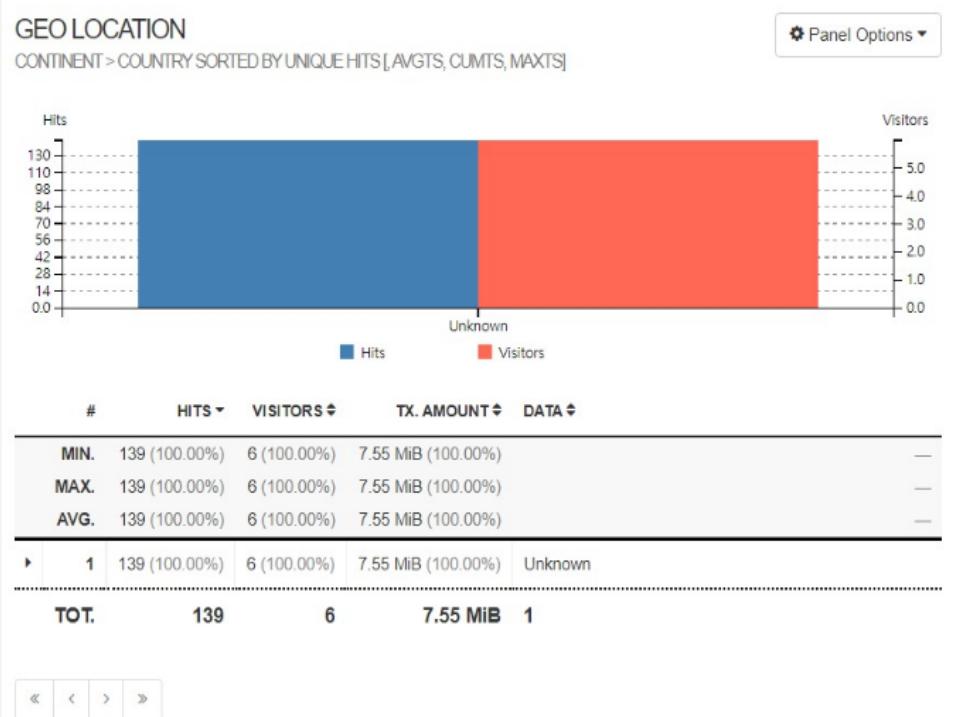
2-9) Refer 사이트 정보



2-10) HTTP 상태코드 정보



2-11) 국가별 정보



K G C I N E M A

05

보안

01. Ansible을 이용한 서버 취약점 검사

◆ 관리자 서버에서 check 플레이북 실행 (검사할 서버 Host)

◆ [root@administrator ~]# ansible-playbook ./check_dns1.yml -i ~/ansible/inventory

[root@administrator ~]# vi ./check_dns1.yml

```
---
- hosts: dns1
  gather_facts: false
  tasks:
    - name: ShellScripts Copy
      copy: src=~/check.sh dest=~/check.sh
    - name: Change script file permissions
      file: dest=~/check.sh mode=755
    - name: Run the script file
      shell: ~/check.sh >> ./check_dns1_res.txt
    - name: Copying files from remote server
      copy:
        src: "/root/check.sh"
        dest: "/root/check.sh"
    - name: Send mwavtst-db data to Stg-Server
      fetch:
        src: "/root/check_dns1_res.txt"
        dest: "./checkfile_dns1.txt"
        flat: yes
```

01. Ansible을 이용한 서버 취약점 검사

❖ [root@administrator ~]# vi ./check.sh

```
#!/bin/bash
echo "root 계정만 UID가 0이면 양호"
TARGET1=/etc/passwd
CHECK1=$(awk -F: '$3 == 0 {print $0}' $TARGET1)
CHECK2=$(printf '%s\n' $CHECK1 | wc -l)
if [[ -n $CHECK1 ]]; then
    if [[ 1 == $CHECK2 ]]; then
        echo -e "=> 양호\n"
    else
        echo -e "=> 변경 필요\n- UID가 0인 계정\n$CHECK1"
    fi
else
    echo -e "=> 점검 \nUID가 0인 계정 없음"
fi
echo -e "-----"
echo "계정 파일 퍼미션이 644 이하면 양호"
TARGET1=/etc/passwd
CHECK1=$(ls -al $TARGET1)
CHECK2=$(find $TARGET1 -perm 644 -o -perm 444 | wc -l)
if [[ 1 == $CHECK2 ]]; then
    echo -e "=> 양호\n"
else
    echo -e "=> 변경필요\n- 퍼미션 644 이하 설정 시 양호\n$CHECK1"
fi
echo -e "-----"
```

01. Ansible을 이용한 서버 취약점 검사

```
echo "그룹 파일 퍼미션이 644 이하면 양호"
TARGET1=/etc/group
CHECK1=$(ls -al $TARGET1)
CHECK2=$(find $TARGET1 -perm 644 -o -perm 444 | wc -l)
if [[ 1 == $CHECK2 ]]; then
    echo -e "=> 양호\n"
else
    echo -e "=> 변경필요\n- 퍼미션 644 이하 설정 시 양호\n$CHECK1"
fi
echo -e "-----"
echo "shadow 파일 퍼미션이 600 이하면 양호"
TARGET1=/etc/shadow
CHECK1=$(ls -al $TARGET1)
CHECK2=$(find $TARGET1 -perm 600 -o -perm 000 | wc -l)
if [[ 1 == $CHECK2 ]]; then
    echo -e "=> 양호\n"
else
    echo -e "=> 변경필요\n- 퍼미션 600 이하 설정 시 양호\n$CHECK1"
fi
echo -e "-----"
echo "일반 사용자의 su 명령어 제한"
echo "pam_wheel.so use_uid 설정"
TARGET1=/etc/pam.d/su
CHECK1=$(grep pam_wheel.so $TARGET1 | grep -v '#')
```

01. Ansible을 이용한 서버 취약점 검사

```
if [[ -n $CHECK1 ]]; then
    echo -e "=> 양호\n"
else
    echo -e "=> 변경필요\n$TARGET1"
fi
echo -e "-----"
echo "root 계정 Telnet 제한"
echo "/etc/pam.d/login 파일에 pam_securetty.so 설정"
TARGET1=/etc/pam.d/login
CHECK1=$(grep pam_securetty.so $TARGET1 | grep -v '#')
if [[ -n $CHECK1 ]]; then
    echo -e "=> 양호\n"
else
    echo -e "=> 변경필요\n$TARGET1"
fi
echo -e "-----"
echo "root계정 ftp 접속 제한"
echo "ftp 서비스를 사용하지 않거나 ftpusers에 root 계정 설정"
TARGET1=/etc/ftpusers
CHECK1=$(ps -ef | grep ftpd | grep -v pts)
if [[ -n $CHECK1 ]]; then
    CHECK2=$(grep -i root $TARGET1)
    if [[ -n $CHECK2 ]]; then
        echo -e "=> 양호\n"
    else
        echo -e "=> 변경필요\nftp 서비스 실행 상태\n$TARGET1 파일내 root 계정 추가 시 양호"
    fi
fi
```

01. Ansible을 이용한 서버 취약점 검사

```
else
    echo -e "=> 양호\nftp 서비스 중지 상태"
fi
echo -e "-----"
echo "익명 FTP 을 제한"
echo "ftp 서비스를 사용하지 않거나 ftp 계정이 없으면 양호"
TARGET1=/etc/passwd
CHECK1=$(ps -ef | grep ftpd | grep -v pts)
CHECK2=$(awk -F: '{print $1}' $TARGET1 | grep ftp)
if [[ -n $CHECK1 ]]; then
    if [[ -z $CHECK2 ]]; then
        echo -e "=> 양호\n"
    else
        echo -e "=> 변경필요\nftp 서비스 실행 상태\n$TARGET1 파일 ftp 계정 삭제 시 양호"
    fi
else
    if [[ -z $CHECK2 ]]; then
        echo -e "=> 양호\n"
    else
        echo -e "=> 변경필요\nftp 서비스 중지 상태\n$TARGET1 파일 ftp 계정 삭제 시 양호"
    fi
fi
echo -e "-----"
```

01. Ansible을 이용한 서버 취약점 검사

◆ 실행 결과가 로컬 컴퓨터에 저장된다

◆ [root@administrator ~]# ls -l

```
-rwxr-xr-x 1 root root 3704 3월 4 14:06 check.sh  
-rw-r--r-- 1 root root 553 3월 4 14:04 check_db1.yml  
-rw-r--r-- 1 root root 553 3월 4 14:05 check_db2.yml  
-rw-r--r-- 1 root root 557 3월 4 14:01 check_dns1.yml  
-rw-r--r-- 1 root root 557 3월 4 14:02 check_dns2.yml  
-rw-r--r-- 1 root root 553 3월 4 14:04 check_was.yml  
-rw-r--r-- 1 root root 557 3월 4 14:03 check_web1.yml  
-rw-r--r-- 1 root root 557 3월 4 14:04 check_web2.yml  
-rw-r--r-- 1 root root 1318 3월 5 22:27 checkfile_dns1.txt
```

◆ 실행 결과가 로컬 컴퓨터에 저장된다

◆ [root@administrator ~]# vi ./checkfile_dns1.txt

```
root 계정만 UID가 0이면 양호  
=> 양호  
  
-----  
계정 파일 퍼미션이 644 이하면 양호  
=> 양호  
  
-----  
그룹 파일 퍼미션이 644 이하면 양호  
=> 양호  
  
-----  
shadow 파일 퍼미션이 600 이하면 양호  
=> 양호
```

K G C I N E M A

검증

06

01. Proxy VRRP 검증

◆ VRRP Health Checking [Proxy]

A Wireshark capture window titled "VMnet8" showing VRRP traffic. The table lists 11 VRRP Announcement frames (v2) sent by source 192.168.1.10 to destination 224.0.0.18. The details pane shows the first frame: Frame 44982: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{AD633878-9D4A-48B1-95FB-0EFEDD4167D7}, id 0. The Ethernet II header shows Src: VMware_84:a7:2e (00:0c:29:84:a7:2e) and Dst: IPv4mcast_12 (01:00:5e:00:00:12). The Internet Protocol Version 4 header shows Src: 192.168.1.10 and Dst: 224.0.0.18.

◆ [root@ha-01 ~]# systemctl stop keepalived

A Wireshark capture window titled "VMnet8" showing VRRP traffic. The table lists 11 VRRP Announcement frames (v2) sent by source 192.168.1.10 to destination 224.0.0.18. The details pane shows the first frame: Frame 11: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{AD633878-9D4A-48B1-95FB-0EFEDD4167D7}, id 0. The Ethernet II header shows Src: VMware_84:a7:2e (00:0c:29:84:a7:2e) and Dst: IPv4mcast_12 (01:00:5e:00:00:12). The Internet Protocol Version 4 header shows Src: 192.168.1.10 and Dst: 224.0.0.18. A large hand-drawn arrow points from the previous capture to this one, indicating the state change.

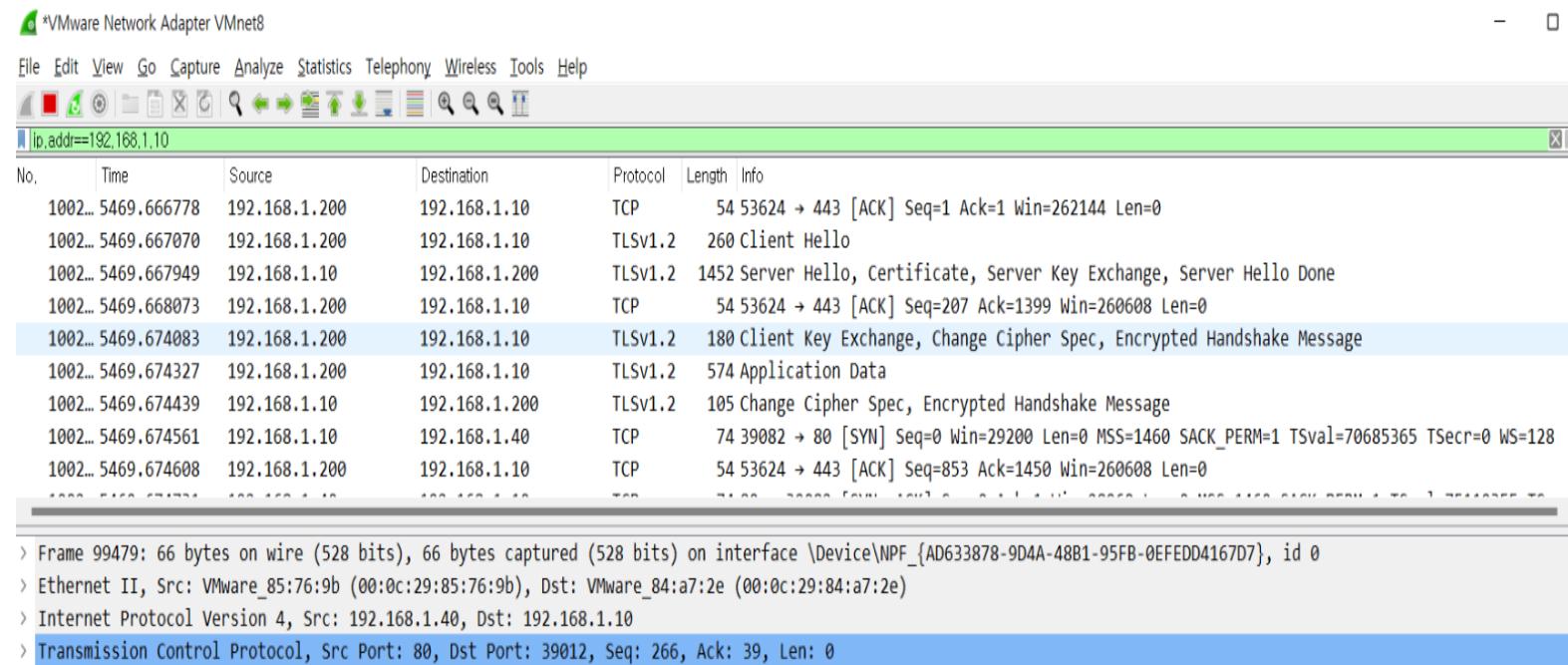
◆ keepalived 데몬을 끄고 실시간 상황

slave서버에서 vip가 잡히는 모습입니다.

02. SSL / TLS 설정 검증

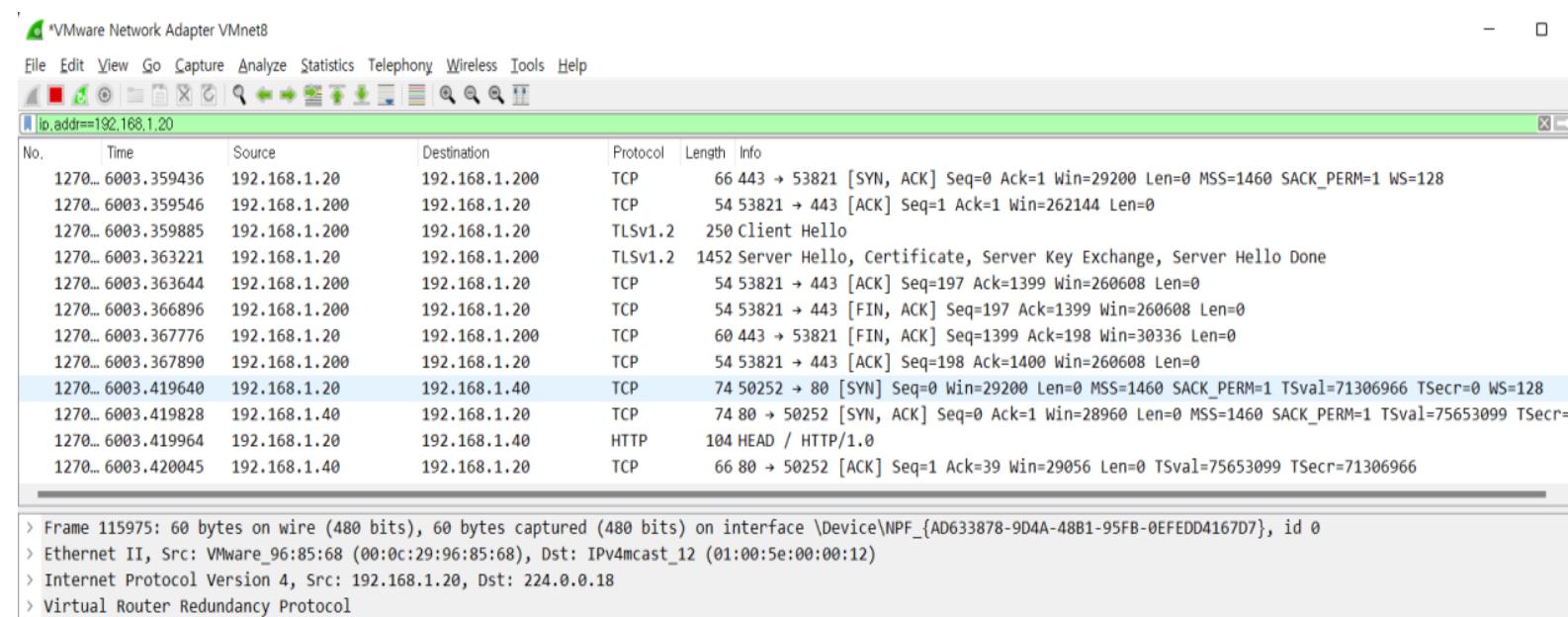
◆ Proxy 2대의 SSL / TLS 설정 확인

Wireshark VMnet8에서 ip.addr == 192.168.1.10



◆ Master Proxy 서버 끄고 Slave Proxy 서버에서 확인

Wireshark VMnet8에서 ip.addr == 192.168.1.20



03. WEB 검증_packet

◆ 패킷을 통한 WEB연결 확인

◆ proxy01 : 192.168.1.10
proxy02 : 192.168.1.20
web01 : 192.168.1.30
web02 : 192.168.1.40

| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|-------------|--------------|--------------|----------|--------|-----------------|
| 26701 | 1108.059040 | 192.168.1.30 | 192.168.1.10 | HTTP | 331 | HTTP/1.1 200 OK |
| 26711 | 1109.575738 | 192.168.1.20 | 192.168.1.30 | HTTP | 104 | HEAD / HTTP/1.0 |
| 26713 | 1109.577079 | 192.168.1.30 | 192.168.1.20 | HTTP | 331 | HTTP/1.1 200 OK |
| 26719 | 1110.758412 | 192.168.1.20 | 192.168.1.40 | HTTP | 104 | HEAD / HTTP/1.0 |
| 26721 | 1110.758946 | 192.168.1.40 | 192.168.1.20 | HTTP | 331 | HTTP/1.1 200 OK |
| 26729 | 1111.060885 | 192.168.1.10 | 192.168.1.40 | HTTP | 104 | HEAD / HTTP/1.0 |
| 26730 | 1111.060931 | 192.168.1.10 | 192.168.1.30 | HTTP | 104 | HEAD / HTTP/1.0 |
| 26733 | 1111.061333 | 192.168.1.40 | 192.168.1.10 | HTTP | 331 | HTTP/1.1 200 OK |
| 26734 | 1111.061338 | 192.168.1.30 | 192.168.1.10 | HTTP | 331 | HTTP/1.1 200 OK |
| 26748 | 1112.580158 | 192.168.1.20 | 192.168.1.30 | HTTP | 104 | HEAD / HTTP/1.0 |
| 26750 | 1112.580482 | 192.168.1.30 | 192.168.1.20 | HTTP | 331 | HTTP/1.1 200 OK |
| 26756 | 1113.763031 | 192.168.1.20 | 192.168.1.40 | HTTP | 104 | HEAD / HTTP/1.0 |
| 26758 | 1113.763476 | 192.168.1.40 | 192.168.1.20 | HTTP | 331 | HTTP/1.1 200 OK |
| 26766 | 1114.062822 | 192.168.1.10 | 192.168.1.40 | HTTP | 104 | HEAD / HTTP/1.0 |
| 26767 | 1114.062855 | 192.168.1.10 | 192.168.1.30 | HTTP | 104 | HEAD / HTTP/1.0 |
| 26770 | 1114.063173 | 192.168.1.40 | 192.168.1.10 | HTTP | 331 | HTTP/1.1 200 OK |
| 26772 | 1114.063288 | 192.168.1.30 | 192.168.1.10 | HTTP | 331 | HTTP/1.1 200 OK |

> Frame 26500: 331 bytes on wire (2648 bits), 331 bytes captured (2648 bits) on interface \Device\NPF_{CDD584CE-3991-4350-92B5-1^
> Ethernet II, Src: VMWare_a1:88:be (00:0c:29:a1:88:be), Dst: VMWare_29:c0:2e (00:0c:29:29:c0:2e)
> Internet Protocol Version 4, Src: 192.168.1.40, Dst: 192.168.1.10
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 51392, Seq: 1, Ack: 39, Len: 265
 Source Port: 80
 Destination Port: 51392
 [Stream index: 1511]

03. WEB 검증_round robin

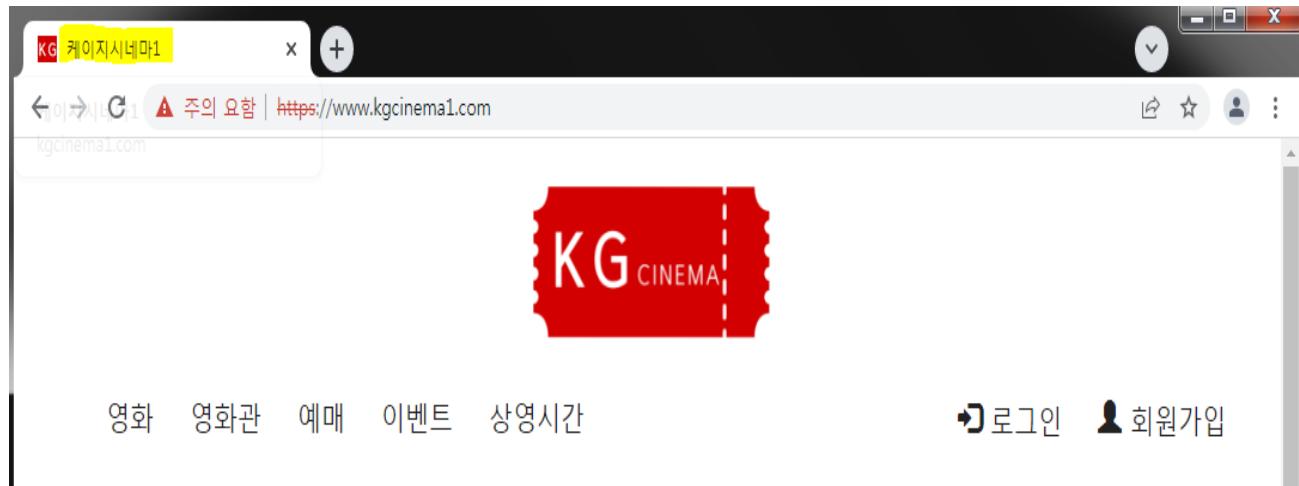
◆ proxy를 통한 라운드로빈

-라운드로빈을 위해 홈페이지의 타이틀 변경

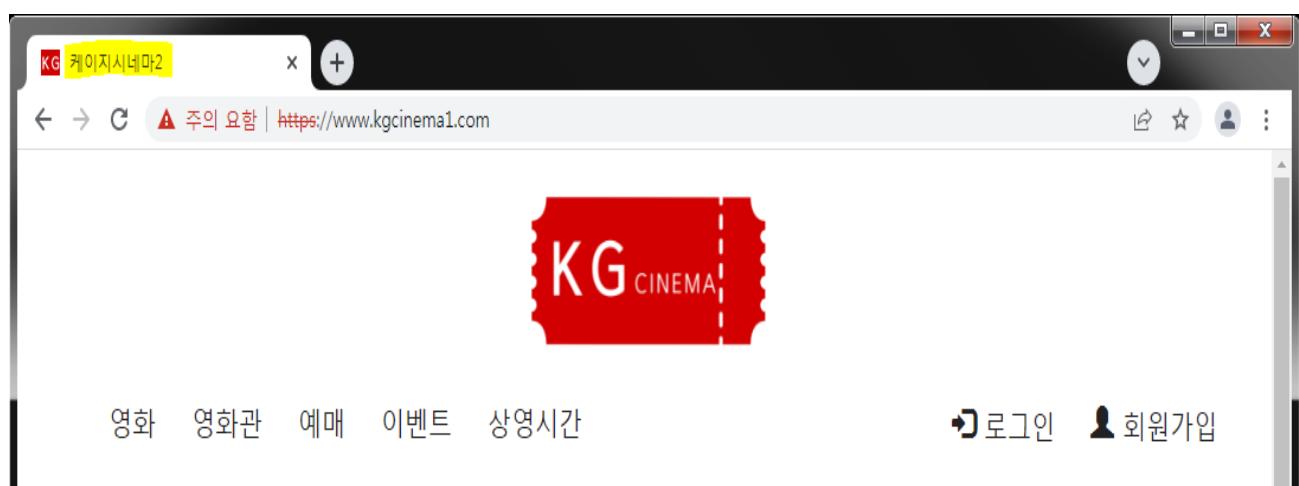
web01 : 케이지시네마1

web02 : 케이지시네마2

◆ www.kgcinema1.com 접속



◆ 쿠키 삭제 후 www.kgcinema1.com 접속



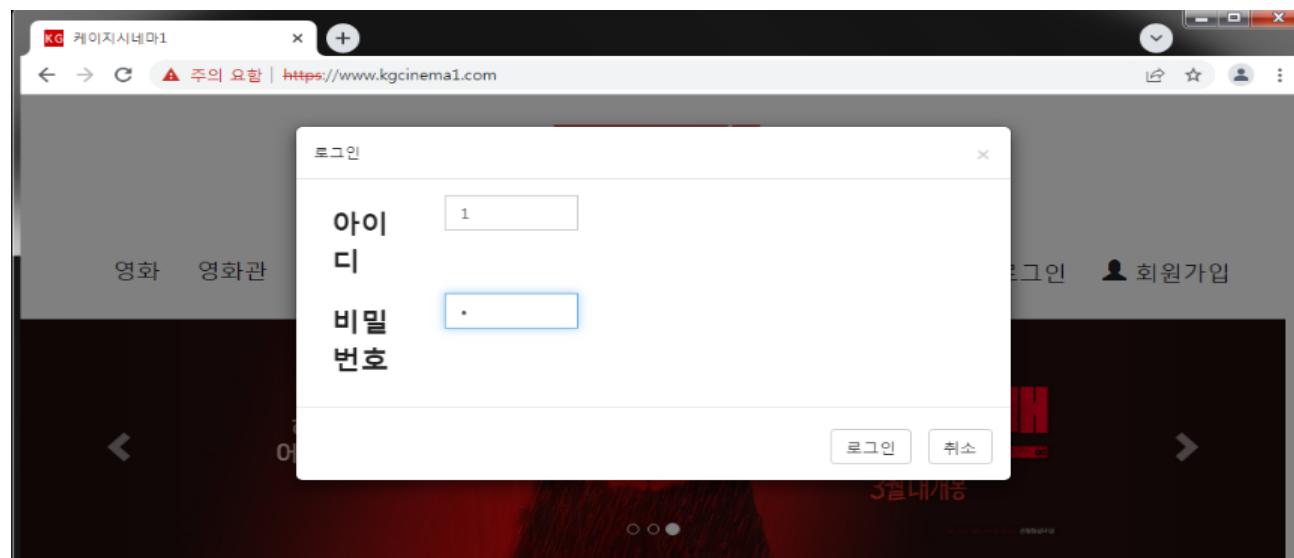
04. DB 검증_packet

◆ packet 통한 DB연결 확인 검증

◆ 기존 유저 : 1 을 통한 로그인

MariaDB [webdb]> select * from member;

| No | ID | PW | Name | phone | addr | Mail |
|----|----|---|------|-------|------|------|
| 1 | 1 | *E6CC90B878B948C35E92B003C792C46C58C4AF40 | 1 | 1 | 1 | 1@1 |

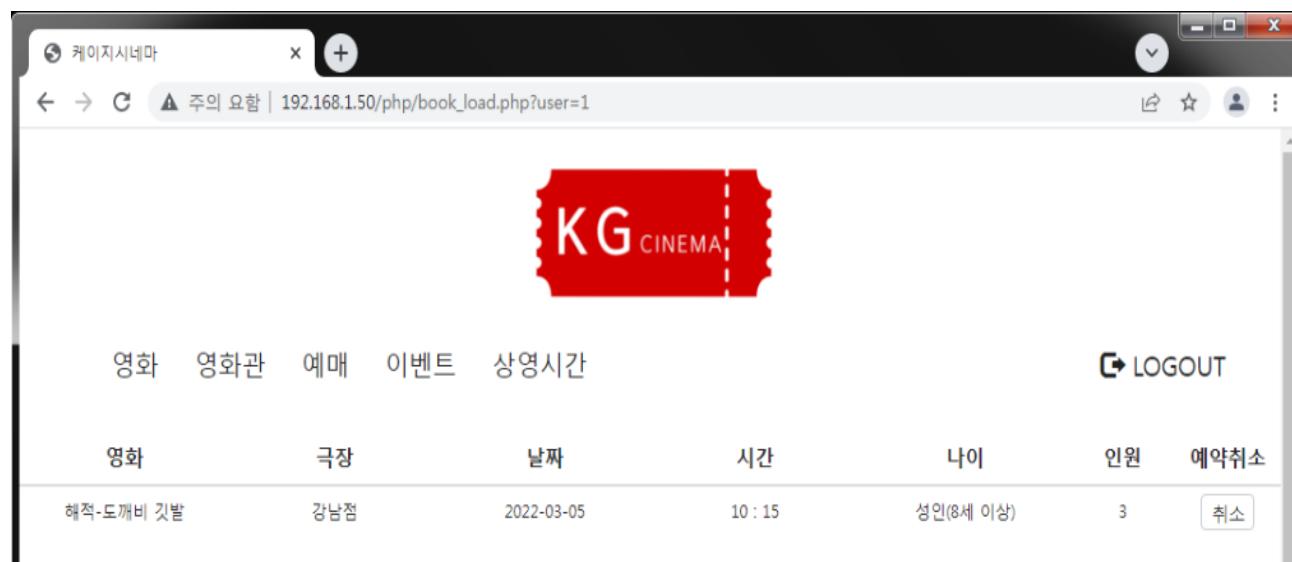
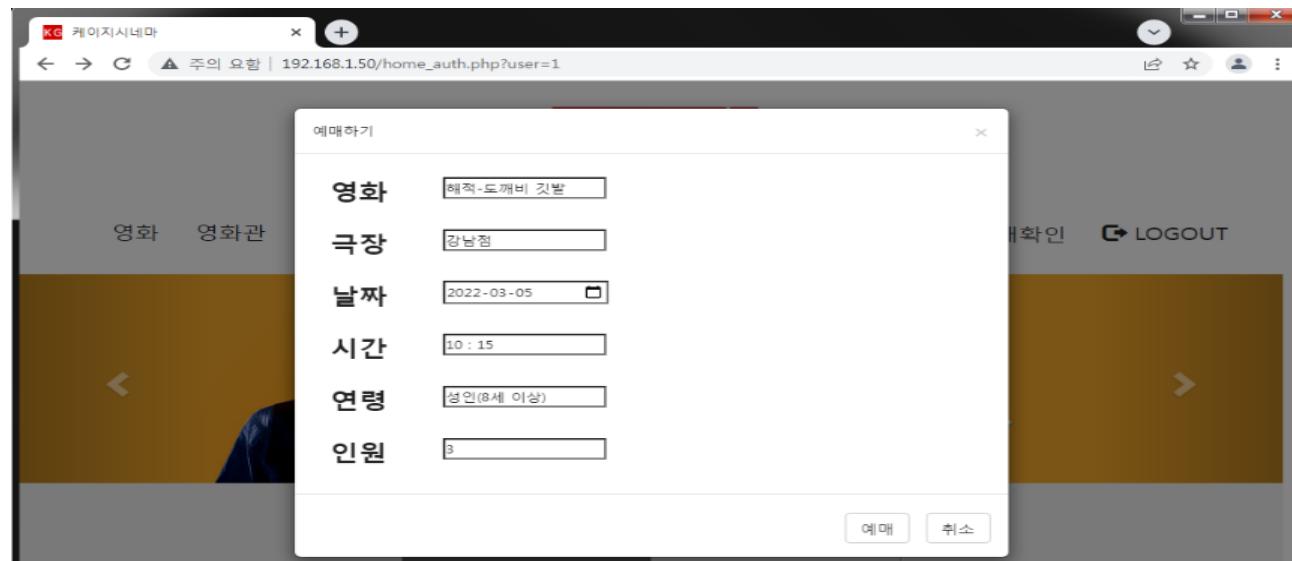


| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|-------------|--------------|--------------|----------|--------|--|
| 56807 | 2941.047343 | 192.168.1.50 | 192.168.1.60 | MySQL | 125 | Request Query |
| 56816 | 2941.050867 | 192.168.1.60 | 192.168.1.50 | MySQL | 430 | Response |
| 56952 | 2941.073094 | 192.168.1.50 | 192.168.1.60 | MySQL | 125 | Request Query |
| 56953 | 2941.073631 | 192.168.1.60 | 192.168.1.50 | MySQL | 430 | Response |
| 57022 | 2941.077656 | 192.168.1.50 | 192.168.1.60 | MySQL | 71 | Request Quit |
| 57497 | 2941.197740 | 192.168.1.60 | 192.168.1.50 | MySQL | 152 | Server Greeting proto=10 version=5.5.68... |
| 57499 | 2941.197859 | 192.168.1.50 | 192.168.1.60 | MySQL | 158 | Login Request user=itbank db=webdb |
| 57501 | 2941.197985 | 192.168.1.60 | 192.168.1.50 | MySQL | 77 | Response OK |
| 57502 | 2941.198308 | 192.168.1.50 | 192.168.1.60 | MySQL | 125 | Request Query |
| 57503 | 2941.199044 | 192.168.1.60 | 192.168.1.50 | MySQL | 430 | Response |
| 57504 | 2941.200338 | 192.168.1.50 | 192.168.1.60 | MySQL | 125 | Request Query |
| 57505 | 2941.203360 | 192.168.1.60 | 192.168.1.50 | MySQL | 430 | Response |
| 57506 | 2941.203730 | 192.168.1.50 | 192.168.1.60 | MySQL | 71 | Request Quit |

04. DB 검증_Replication

◆ Replication을 통한 DB master-slave 검증

◆ 기존 유저 : 1 을 통한 예매



04. DB 검증_Replication

◆ master db 확인

```
MariaDB [webdb]> select * from book;
```

| userid | movie | cinema | date | time | age | person |
|--------|-----------------------|---------|------------|---------|-------------------|--------|
| 1 | í•`ì-ë,ê¹°ë¹,, ê¹fë°œ | ê°•ë,°ì | 2022-03-05 | 10 : 15 | ì,,±ì,(8ì,, ì`if) | 3 |

◆ slave db 확인

```
MariaDB [webdb]> select * from book;
```

| userid | movie | cinema | date | time | age | person |
|--------|-----------------------|---------|------------|---------|-------------------|--------|
| 1 | í•`ì-ë,ê¹°ë¹,, ê¹fë°œ | ê°•ë,°ì | 2022-03-05 | 10 : 15 | ì,,±ì,(8ì,, ì`if) | 3 |

◆ packet 확인

