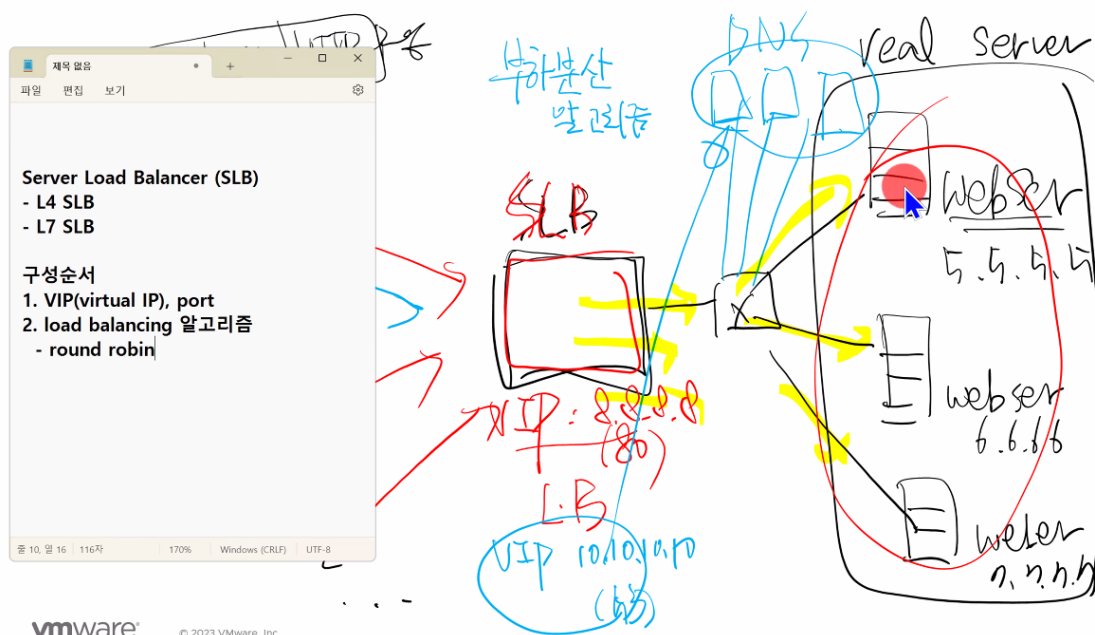
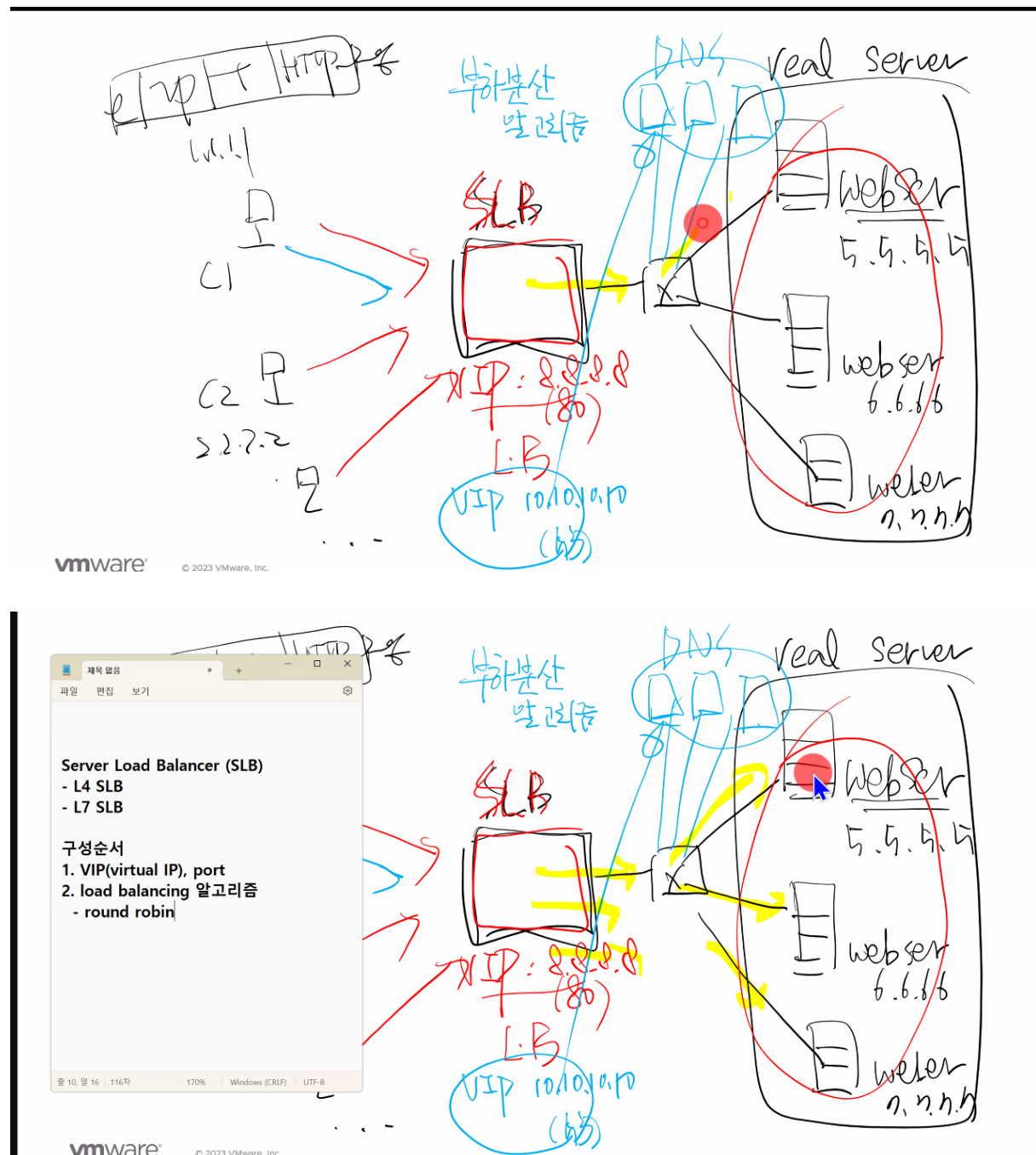


VM웨어 Cluster

DRS



- VIP(virtual IP, Port)
- 부하분산 알고리즘
 - 라운드로빈 방식 : 웨이트를 3을 주면 3번을 더 주고 나머지는 1번씩 줌
 - least connection 알고리즘 등등 상황에 맞게 선택
- real server 모니터링
 - 헬스 모니터링 해당 부하를 보낼 수 있게 해당 서버가 살아있는가?
 - 헬스모니터링 어떻게?
 - active 방식
 - SLB가 모니터링함
 - ARP, ICMP, tcp syn, dns query 등 사용
 - passive 방식
 - 클라이언트 ↔ 서버 간 트래픽 모니터링
 - HTTP 보내면 200 응답받는걸 확인
- Anti-Affinity Rules
 - 특정 가상 머신(VM)들이 서로 다른 호스트에서 실행되도록 보장하는 규칙
- 다른 호스트에 각 가상머신이 분배되어 있어야 로드밸런싱이 가능하기에 위의 설명을 했다.

DRS Anti-Affinity Rules

Anti-Affinity 규칙은 **vSphere DRS (Distributed Resource Scheduler)**에서 특정 가상 머신(VM)들이 **서로 다른 호스트**에서 실행되도록 보장하는 기능입니다. 이 규칙은 한 호스트에서 여러 중요한 VM이 동시에 실행될 경우, 그 호스트에 장애가 발생하면 전체 서비스가 중단되는 위험을 방지하기 위해 사용됩니다.

Anti-Affinity 규칙에 대한 설명:

1. 목적:

- Anti-Affinity 규칙은 특정 VM들이 **같은 호스트에서 실행되지 않도록** 합니다. 이는 주로 **중복성**을 확보하고, **고가용성**을 유지하며, 하드웨어 장애로 인한 위험을 최소화하기 위해 사용됩니다.

- 예를 들어, 여러 중요한 VM들이 같은 호스트에서 실행될 경우, 그 호스트가 다운되면 모든 VM들이 함께 중단될 수 있어 심각한 서비스 중단이 발생할 수 있습니다. Anti-Affinity 규칙은 이러한 VM들이 서로 다른 호스트에서 실행되도록 하여 이런 위험을 줄입니다.

2. 비즈니스 시나리오 (문서 내용에서):

- 예를 들어, 한 회사가 온라인 스토어를 운영하며 여러 웹 서버를 사용한다고 가정해 봅시다. 이 웹 서버들이 모두 같은 ESXi 호스트에서 실행 중인데, 해당 호스트에 장애가 발생하면 온라인 스토어가 완전히 중단될 수 있습니다. 이 경우 비즈니스에 큰 타격을 입게 되겠죠.
- 이런 상황을 방지하기 위해 **Anti-Affinity 규칙**을 설정하여 웹 서버 VM들이 서로 다른 호스트에서 실행되도록 할 수 있습니다. 이렇게 하면 한 호스트에 장애가 발생해도 다른 호스트에서 웹 서버가 계속 실행되므로 서비스가 유지됩니다.

3. Anti-Affinity 규칙의 작동 방식:

- Anti-Affinity 규칙이 설정되면, DRS는 지정된 VM들이 **같은 호스트에서 실행되지 않도록** 보장합니다.
- 만약 자원이 부족하여 여러 VM들이 같은 호스트에 배치되는 상황이 발생하면, DRS는 자동으로 VM을 다른 호스트로 이동시켜 규칙을 준수하면서 자원 할당을 조정합니다.

4. 사용 방법:

- Anti-Affinity 규칙은 클러스터 설정의 **VM/호스트 규칙** 섹션에서 구성할 수 있습니다.
- 관리자는 특정 VM 그룹이 항상 서로 다른 호스트에서 실행되도록 Anti-Affinity 규칙을 지정할 수 있습니다.

예시:

예를 들어, 웹 서버 VM 4개 (**Web-01** , **Web-02** , **Web-03** , **Web-04**)가 있다고 가정하면, Anti-Affinity 규칙을 생성하여 이 VM들이 같은 호스트에서 실행되지 않도록 할 수 있습니다. 만약 호스트 장애가 발생하더라도, vSphere HA는 해당 VM들을 서로 다른 호스트에서 재시작하면서 Anti-Affinity 규칙을 계속 준수합니다.

실질적인 이점:

- **중복성 확보:** 중요한 VM들이 한 호스트에서 실행되는 것을 방지하여, 호스트 장애 시 모든 VM이 동시에 중단되는 상황을 예방할 수 있습니다.
- **장애 내성:** 서비스를 지속적으로 유지하며 운영이 중단되지 않도록 돕습니다.

- **부하 분산:** 중요한 워크로드를 여러 물리적 자원에 분산시켜 성능을 유지합니다.

결론적으로, Anti-Affinity 규칙은 DRS에서 VM들의 가용성을 높이고 장애 상황에서 시스템의 내성을 강화하는 데 중요한 역할을 합니다. 이를 통해 중요한 애플리케이션이 항상 여러 호스트에 분산되어 안전하게 실행될 수 있습니다.

Server Load Balancer (SLB)

- L4 SLB
- L7 SLB

구성순서

1. VIP(virtual IP), port
2. load balancing 알고리즘
 - round robin (weighted RR)
 - least connection
3. real server 모니터링(health monitoring)
 - active: slb가 모니터링(ex. arp, icmp, tcp syn, dns query...)
 - passive: client ↔ server간 트래픽을 모니터링

About ESXi Host Maintenance Mode

ESXi Host Maintenance Mode는 ESXi 호스트를 유지 관리하기 위해 사용되는 기능으로, 호스트에 대한 유지 보수를 수행할 때 해당 호스트에서 실행 중인 모든 가상 머신(VM)을 안전하게 처리하는 기능입니다. 이 모드를 사용하면 호스트에 하드웨어 또는 소프트웨어 업데이트를 적용하거나, 메모리 추가, 패치 설치와 같은 작업을 수행할 수 있습니다.

Maintenance Mode의 주요 개념:

1. **Maintenance Mode 동작 원리:**
 - **호스트를 유지 관리 모드로 전환**하면, 호스트에서 실행 중인 모든 VM은 **다른 호스트로 이동**되거나, **종료**됩니다. VM들이 안전하게 다른 호스트로 옮겨지기 전까지 호스트는 완전히 유지 관리 모드로 들어가지 않습니다.

- VM들이 자동으로 다른 호스트로 이동하는 과정은 **vSphere DRS (Distributed Resource Scheduler)** 및 **vMotion** 기능과 협력하여 수행됩니다. vSphere DRS가 활성화되어 있으면, DRS가 VM을 자동으로 다른 호스트로 분산시키며 리소스를 최적화합니다.

2. 유지 관리 모드의 필요성:

- 호스트의 하드웨어 또는 소프트웨어를 업그레이드하거나 패치를 적용하는 작업을 수행할 때 호스트를 유지 관리 모드로 전환해야 합니다. 예를 들어, 메모리 추가 작업이나, ESXi 소프트웨어 업그레이드가 필요할 때 유지 관리 모드가 필요합니다.
- **가상 머신의 가용성**을 유지하기 위해, VM들을 자동으로 다른 호스트로 마이그레이션시키고 나서 호스트에 필요한 작업을 수행할 수 있습니다.

3. DRS 및 vMotion과의 통합:

- 유지 관리 모드가 활성화되면 **vMotion**을 통해 VM이 다른 호스트로 이동합니다. 이때 **vSphere DRS**가 자동화된 방식으로 VM이 이동할 최적의 호스트를 결정합니다.
- **Fully Automated** 모드에서 DRS는 완전히 자동으로 VM을 다른 호스트로 이동시키고, **Partially Automated** 모드나 **Manual** 모드에서는 관리자가 이동 권장 사항을 수동으로 적용해야 할 수도 있습니다.

4. DRS 설정에 따른 동작:

- **Fully Automated 모드:** 호스트가 유지 관리 모드로 전환되면 DRS가 자동으로 모든 VM을 다른 호스트로 마이그레이션합니다.
- **Partially Automated 모드:** DRS는 VM 이동에 대한 권장 사항을 제공하고, 관리자는 권장 사항을 수동으로 적용하여 VM을 이동해야 합니다.
- **Manual 모드:** DRS는 관리자가 직접 VM을 마이그레이션하거나 종료해야 하는 권장 사항을 제공합니다.

유지 관리 모드를 사용해야 하는 상황:

- **하드웨어 유지 보수:** 메모리 추가, 하드웨어 교체 등 물리적인 변경 작업이 필요할 때.
- **소프트웨어 업데이트:** ESXi 패치 설치 또는 버전 업그레이드가 필요할 때.
- **보안 패치:** 보안 취약점을 해결하기 위해 패치가 필요할 때.

유지 관리 모드로 호스트 전환하기:

1. **vSphere Web Client**에서 해당 ESXi 호스트를 선택합니다.
2. **호스트 설정**에서 **Enter Maintenance Mode** 옵션을 선택합니다.

3. VM들이 자동으로 다른 호스트로 마이그레이션되거나, 사용자가 수동으로 처리할 수 있습니다.
4. 호스트가 유지 관리 모드에 완전히 진입하면, 유지 보수 작업을 수행할 수 있습니다.

유지 관리 모드가 필요한 이유:

- **안전한 유지 보수:** 호스트에서 실행 중인 VM을 중단하지 않고도, 호스트에서 필요한 작업을 수행할 수 있습니다.
- **가용성 유지:** VM을 다른 호스트로 자동으로 이동하여 서비스 중단 없이 유지 보수 작업을 진행할 수 있습니다.
- **자원 최적화:** DRS를 통해 자원이 효율적으로 분배되어, 전체 클러스터 성능을 유지할 수 있습니다.

요약:

ESXi Host Maintenance Mode는 **호스트 유지 보수**를 위한 필수 기능으로, **VM의 가용성을 유지**하면서 호스트의 하드웨어나 소프트웨어에 대한 유지 보수를 안전하게 수행할 수 있습니다. **vSphere DRS**와 **vMotion**을 활용해 VM을 다른 호스트로 자동으로 이동시키며, 관리자가 유지 보수 작업을 간편하게 수행할 수 있도록 지원합니다.

HA Admission control

HA Admission Control은 **vSphere HA (High Availability)** 환경에서 클러스터의 자원을 관리하는 중요한 기능입니다. 이 기능은 클러스터의 가상 머신(VM)이 항상 가용한 상태를 유지할 수 있도록 **자원 할당을 제어**하는 역할을 합니다. 구체적으로, **호스트 장애 발생 시에도 VM을 재시작할 수 있는 충분한 자원이 있는지 확인**하여, 가용성을 보장하는 것이 핵심입니다.

HA Admission Control이란?

HA Admission Control은 클러스터의 각 호스트가 장애를 겪을 때, 해당 호스트에 있던 VM을 **다른 호스트에서 재시작**할 수 있도록 자원이 충분한지 확인하는 메커니즘입니다. 이를 위해 클러스터에서 일정량의 자원을 **예약**해 두어 장애 시에도 원활히 대응할 수 있게 합니다.

HA Admission Control의 주요 기능:

1. **호스트 장애 시 재시작 보장:**

- 클러스터에 포함된 호스트 중 하나가 장애를 일으켰을 때, 그 호스트에 있던 VM들은 다른 호스트에서 재시작해야 합니다. 하지만, 다른 호스트에 충분한 자원이 없으면 재시작이 불가능해집니다. HA Admission Control은 이러한 상황을 방지하기 위해 클러스터의 자원을 사전에 관리하고 할당합니다.

2. 자원 관리 및 예약:

- 클러스터 내에서 특정 양의 CPU와 메모리를 **예약**하여, 장애가 발생했을 때도 다른 호스트가 그 자원을 활용하여 VM을 재시작할 수 있도록 합니다.
- 이 기능은 각 호스트가 자신의 자원을 최대한 활용하지 않도록 제한함으로써, 클러스터 내에서 항상 여유 자원이 남아 있는 상태를 유지합니다.

HA Admission Control의 모드:

1. Host Failures Cluster Tolerates (호스트 장애 허용 모드):

- 이 모드는 클러스터에서 몇 개의 호스트 장애를 견딜 수 있는지 지정합니다. 예를 들어, 1개의 호스트 장애를 허용하도록 설정하면, 클러스터는 장애 발생 시 다른 호스트들이 해당 VM을 재시작할 수 있도록 충분한 자원을 유지합니다.
- VMware에서는 주로 **N+1** 방식으로 호스트 장애를 관리하는데, 여기서 **N**은 클러스터의 호스트 수이고, **+1**은 허용할 수 있는 호스트 장애 수를 의미합니다.

2. Percentage of Cluster Resources Reserved (클러스터 자원의 백분율 예약):

- 클러스터 내 자원의 특정 비율을 예약해 두어, 장애 시 VM을 재시작할 수 있는 여유 자원을 남깁니다. 예를 들어, 클러스터 자원의 25%를 예약해 두면, 장애가 발생했을 때 이 25%의 자원을 사용하여 VM을 재시작할 수 있습니다.
- 이 방식은 호스트 수가 적을 때보다 클러스터 내 호스트가 많을 때 유리하게 작동합니다.

3. Slot Policy (슬롯 기반 정책):

- 슬롯은 클러스터 내에서 VM이 필요한 최소 자원(CPU와 메모리)의 단위로 정의됩니다. **슬롯 기반 정책**은 클러스터에서 VM을 재시작하기 위해 필요한 슬롯 수를 미리 계산하고, 호스트 장애 시 필요한 슬롯을 확보해 둡니다.
- 모든 호스트에 대해 동일한 슬롯 크기가 계산되며, 이러한 슬롯 기반 정책을 통해 자원 부족 문제를 예방할 수 있습니다.

예시:

- 클러스터가 5개의 호스트로 구성되어 있고, 그 중 1개 호스트가 장애를 겪을 수 있도록 설정된 경우를 생각해봅시다. **HA Admission Control**은 장애 발생 시 나머지 4개의

호스트가 장애 호스트의 VM을 재시작할 수 있는 충분한 자원을 유지하도록 예약합니다. 이로 인해 각 호스트는 자신이 가진 자원의 일부만 사용할 수 있습니다.

- 만약 이 정책을 설정하지 않으면, 클러스터가 모든 자원을 사용 중일 때 한 호스트가 다운되면 재시작할 자원이 부족할 수 있어, VM이 정상적으로 재시작되지 못할 위험이 있습니다.

HA Admission Control의 중요성:

1. **가용성 보장:** 자원을 미리 예약하여 호스트 장애 시 VM이 중단되지 않고 즉시 재시작될 수 있도록 함으로써, 시스템의 **가용성을 극대화**합니다.
2. **리소스 관리 최적화:** 자원 낭비를 최소화하고, 장애 시에도 가상 환경이 계속해서 안정적으로 운영되도록 **자원을 효율적으로 관리**합니다.
3. **서비스 연속성:** 비즈니스 크리티컬 애플리케이션이 실행 중인 경우, 예기치 않은 호스트 장애로 인한 서비스 중단을 최소화하여 **서비스의 연속성**을 보장합니다.

요약:

HA Admission Control은 vSphere HA 클러스터에서 **가상 머신의 재시작을 보장**하기 위해 자원을 관리하는 핵심 기능입니다. 클러스터 내에서 **자원을 예약**하여, 호스트 장애 시에도 VM이 다른 호스트에서 안정적으로 재시작될 수 있도록 합니다. 이러한 자원 제어는 호스트 장애 발생 시 가용성을 유지하고, 클러스터 자원을 최적으로 활용하는 데 중요한 역할을 합니다.

HA Admission control vs DRS

vSphere HA Admission Control과 ****vSphere DRS (Distributed Resource Scheduler)****는 vSphere 클러스터 환경에서 모두 중요한 역할을 하는 기능이지만, 그 목적과 동작 방식은 다릅니다. 아래에서 두 기능을 비교하고 각각의 특징을 설명하겠습니다.

1. vSphere HA Admission Control

목적:

- **HA (High Availability)** Admission Control은 클러스터 내에서 **가용성**을 유지하기 위해, 장애 발생 시 가상 머신(VM)을 다른 호스트에서 **재시작할 수 있는 자원을 확보**하는 기능입니다. 클러스터가 특정 호스트의 장애를 견딜 수 있도록 자원을 예약하여, 시스템이 항상 준비된 상태를 유지합니다.

주요 역할:

- 호스트가 장애를 일으킬 경우, 다른 호스트가 그 호스트의 VM을 재시작할 수 있도록 **자원을 사전에 확보**합니다.
- 클러스터 내에서 **미리 자원을 제한**하여 모든 자원이 과도하게 사용되는 상황을 방지합니다.
- 관리자는 **클러스터에서 허용할 장애 수나 자원 예약 비율**을 지정할 수 있으며, 이 설정을 통해 몇 개의 호스트 장애가 발생해도 시스템이 정상 작동할 수 있게 자원을 관리합니다.

동작 방식:

- **자원 예약:** 클러스터 내에서 CPU와 메모리 자원을 일정 비율로 예약하여, 장애 발생 시 해당 자원을 사용해 VM을 재시작합니다.
- **슬롯 기반 정책:** 각 VM이 차지하는 최소 자원을 기준으로 "슬롯"을 정의하고, 장애 시 필요한 슬롯 수를 미리 계산하여 자원을 확보합니다.

결과:

- **고가용성 보장:** 호스트 장애 발생 시에도 VM이 신속히 다른 호스트에서 재시작되어, 서비스의 **중단을 최소화**합니다.

2. vSphere DRS (Distributed Resource Scheduler)

목적:

- **DRS**는 클러스터 내 **리소스의 효율적 분배와 부하 균형**을 맞추는 기능입니다. DRS는 클러스터의 호스트들 간에 가상 머신을 **자동으로 이동시켜 CPU, 메모리 등 리소스를 최적으로** 사용하도록 돕습니다.

주요 역할:

- 클러스터 내에서 **VM들이 필요한 자원을 충분히 얻을 수 있도록** 리소스를 자동으로 할당하고, **불균형한 자원 사용**을 방지합니다.
- 호스트 간 **부하 분산**을 위해 VM을 **vMotion**을 통해 자동으로 다른 호스트로 **마이그레이션**시킵니다.

동작 방식:

- **VM 마이그레이션:** DRS는 클러스터 내 호스트들의 자원 상태를 지속적으로 모니터링하며, VM이 필요한 자원을 충분히 사용할 수 있도록 호스트 간에 **자동으로 VM을 이동**시킵니다.
- **부하 균형:** 자원이 부족한 호스트에서 자원이 풍부한 호스트로 VM을 이동시키는 등 **부하를 분산**하여, 클러스터 전체의 자원 사용을 최적화합니다.

결과:

- **리소스 최적화:** 클러스터 내 호스트들 간 자원이 균등하게 사용되며, 모든 VM이 **충분한 CPU, 메모리**를 확보하여 원활하게 동작합니다.

차이점 요약:

기능	HA Admission Control	DRS (Distributed Resource Scheduler)
목적	장애 시 VM이 재시작될 수 있도록 자원을 사전에 예약	VM들이 충분한 자원을 받도록 리소스 분배 및 부하 분산
주요 기능	장애가 발생할 경우에도 VM이 재시작할 수 있도록 자원 예약	클러스터 자원을 효율적으로 사용하고 자동으로 VM을 이동 시켜 부하 균형
동작 방식	자원의 일정 비율 또는 슬롯을 미리 예약	vMotion 을 사용해 VM을 자동으로 이동시켜 자원을 최적화
사용 목적	호스트 장애에 대비 하여 서비스 중단을 최소화	부하 분산 과 리소스 효율성 을 위해 VM을 이동
결과	호스트 장애 시 가용성 유지 및 재시작	리소스 활용을 극대화하고 성능 최적화

어떻게 협력하는가?

- **HA**와 **DRS**는 상호 보완적인 기능입니다. HA Admission Control은 주로 **가용성**에 중점을 두며, **호스트 장애**에 대비하여 자원을 관리합니다. 반면에, DRS는 **부하 균형**과 **자원 최적화**에 초점을 맞춰 가상 환경에서 VM들이 항상 충분한 자원을 받도록 관리합니다.
- 예를 들어, 한 호스트에서 장애가 발생하면, **vSphere HA**는 해당 호스트의 VM을 다른 호스트에서 **재시작**하고, 그 이후에 **vSphere DRS**가 **리소스를 최적화**하기 위해 필요한 경우 VM을 다시 적절한 호스트로 **자동 마이그레이션**할 수 있습니다.

결론적으로, **HA Admission Control**은 **고가용성**에 초점을 맞춘 자원 예약 및 관리 기능이고, **DRS**는 클러스터 내 자원의 **효율적 사용**과 **부하 분산**을 돕는 기능으로, 둘은 함께 사용될 때 클러스터의 안정성과 성능을 극대화할 수 있습니다.

How DRS and vSphere HA Work Together

- *vSphere DRS (Distributed Resource Scheduler)**와 **vSphere HA (High Availability)**는 vSphere 환경에서 가상 머신(VM)의 **리소스 최적화**와 **고가용성**을 보장하기 위해 함께 작동하는 두 가지 중요한 기능입니다. 두 기능은 상호 보완적인 역할

을 하여, 시스템이 장애에 대응하고 자원을 효율적으로 분배할 수 있도록 도와줍니다. 다음은 DRS와 HA가 함께 작동하는 방식입니다.

각 기능의 개요:

- **vSphere HA:** 하드웨어 또는 소프트웨어 장애가 발생했을 때 **VM을 다른 호스트에서 재시작**하여 **다운타임을 최소화**하는 역할을 합니다. HA는 장애가 발생하면 즉시 VMs를 다른 가용한 호스트에서 실행시켜 시스템의 가용성을 유지합니다.
- **vSphere DRS:** 클러스터 내의 자원을 **효율적으로 분배**하여, VM이 **CPU, 메모리** 등의 자원을 충분히 받을 수 있도록 관리합니다. 자원이 부족하거나 불균형한 경우, DRS는 VM을 자동으로 다른 호스트로 **마이그레이션**하여 부하를 균형 있게 맞춥니다.

DRS와 HA가 함께 작동하는 방식:

1. 호스트 장애와 VM 복구:

- **ESXi 호스트**에 장애가 발생하면, **vSphere HA**는 해당 호스트에서 실행 중이던 VM을 **다른 가용한 호스트에서 재시작**합니다. HA는 가용성을 보장하기 위해 VM을 신속하게 재시작하지만, 이 과정에서 자원 사용량은 고려하지 않습니다.
- 이후, **vSphere DRS**는 **리소스 최적화**를 위해 개입합니다. HA는 자원 균형을 고려하지 않기 때문에, 재시작된 VM이 자원이 부족한 호스트에 배치될 수 있습니다. 이 경우 DRS가 **리소스를 모니터링**하고, 자원 분배가 고르게 이루어지지 않는다면 VM을 적절한 호스트로 **자동 마이그레이션**하여 **부하를 분산**시킵니다.

예시:

- 호스트 A가 장애를 겪고, **HA**가 VM을 호스트 B와 C에서 재시작했다고 가정해봅시다. 만약 호스트 B가 이미 많은 자원을 사용하고 있다면, DRS는 이 불균형을 인지하고 **vMotion**을 통해 VM을 자원이 더 여유 있는 호스트 D로 이동시켜 **자원 균형**을 맞춥니다.

2. 장애 복구 후 리소스 최적화:

- **vSphere DRS**는 클러스터 내의 모든 호스트의 **자원 사용량을 모니터링**하며, 모든 VM이 CPU, 메모리 등의 자원을 충분히 사용할 수 있도록 조정합니다.
- HA에 의해 VM이 재시작된 후, **DRS**는 **자원 할당을 다시 평가**하고 필요한 경우 **vMotion**을 사용해 자원을 더 효율적으로 활용할 수 있는 호스트로 VM을 이동시킵니다. 이를 통해 VM이 단순히 실행되는 것뿐만 아니라, **최적의 성능**을 유지할 수 있도록 보장합니다.

3. 자원 과부하 방지:

- HA와 DRS가 동시에 활성화된 상태에서, **HA Admission Control**은 항상 클러스터 내에 호스트 장애를 견딜 수 있는 충분한 자원이 있는지 확인합니다. 이를 통해, 장애 발생 시에도 다른 호스트에서 VM을 재시작할 수 있는 자원을 미리 확보해 둡니다.
- HA가 VM을 재시작한 후, DRS는 **부하 분산**을 위해 과부하가 발생하지 않도록 VM을 적절하게 재배포합니다.

4. 유지 보수 모드와의 협력:

- **ESXi 호스트를 유지 보수 모드**로 전환할 때, **vSphere DRS**는 해당 호스트에서 실행 중인 모든 VM을 자동으로 다른 호스트로 마이그레이션하여, 호스트가 안전하게 유지 보수에 들어갈 수 있도록 합니다.
- DRS와 HA가 함께 사용될 경우, 유지 보수 모드로 전환 시 DRS가 VM을 **부드럽게 마이그레이션**하여 **자원 균형을 유지**하면서 다른 호스트에 과부하가 발생하지 않도록 합니다.

DRS와 HA의 상호 작용:

- **초기 VM 배치:**
 - **VM이 처음으로 전원을 켤 때, DRS는 최적의 호스트를 선택**하여 VM을 배치합니다. 만약 호스트에 장애가 발생하면 **HA는 VM을 다른 호스트에서 재시작**합니다. 이후, **DRS는 자원 사용량을 다시 평가**하고 필요할 경우 VM을 적절한 호스트로 이동시킵니다.
- **장애 복구와 부하 균형:**
 - **HA는 장애 복구를 담당**하며, 호스트 장애 시 VM을 재시작합니다. 이후 **DRS는 리소스 최적화**를 위해 클러스터의 리소스 상태를 분석하고, 불균형이 있을 경우 **vMotion**을 통해 VM을 이동시킵니다.
- **사전 예방 조치:**
 - **Proactive HA**가 활성화된 경우, **호스트의 하드웨어 상태가 악화**되는 징후가 감지되면 DRS는 호스트 장애가 발생하기 전에 VM을 안전하게 다른 호스트로 **마이그레이션**하여 사전 대응할 수 있습니다.

DRS와 HA의 협력 예시:

1. 호스트 장애 발생:

- 클러스터에 네 개의 ESXi 호스트(A, B, C, D)가 있다고 가정해봅시다.

- 호스트 A가 하드웨어 문제로 인해 장애가 발생하면, **vSphere HA**는 호스트 A에 있던 모든 VM을 **호스트 B, C, D에서 재시작**합니다.
- 재시작된 후 호스트 B와 C는 기존 VM에 추가된 VM으로 인해 자원이 **과부하**될 수 있습니다.

2. 리소스 최적화:

- **vSphere DRS**는 클러스터 내 자원 상태를 모니터링하고, 호스트 B와 C가 과부하 상태에 있다고 판단하면, DRS는 **vMotion**을 통해 일부 VM을 **호스트 D로 이동**시켜 자원 사용을 균등하게 맞춥니다.
- 이를 통해 각 VM이 충분한 자원을 확보하고, 모든 호스트가 **효율적으로 자원**을 사용하게 됩니다.

요약:

- **vSphere HA**는 **고가용성**을 보장하기 위해 장애 발생 시 VM을 다른 호스트에서 **재시작**하며, 다운타임을 최소화합니다.
- **vSphere DRS**는 **자원 최적화**를 목표로 하여 VM을 클러스터 내에서 자동으로 **마이그레이션**시켜, 자원이 효율적으로 사용되도록 보장합니다.
- 두 기능이 함께 사용될 때, HA는 **장애 복구**를 처리하고, DRS는 장애 복구 후 **리소스 균형**을 맞추어 클러스터 내 VM들이 최적의 성능을 유지할 수 있도록 합니다.

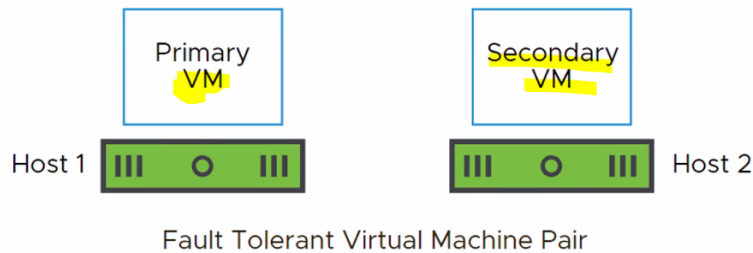
결론적으로, **HA**는 VM이 계속 실행되도록 보장하고, **DRS**는 VM이 **효율적으로 실행**되도록 보장합니다. 두 기능은 함께 사용되어 클러스터의 **가용성**과 **성능 최적화**를 동시에 보장합니다.

FT

Fault-Tolerant VM Pair

How does vSphere FT achieve this level of protection?

- A VM protected by vSphere FT is called the **primary VM**.
- vSphere FT protects the primary VM by **mirroring** it on a different host as a secondary VM.
- The primary and secondary VMs reside on different hosts to form a fault-tolerant VM pair.



- VM을 페어링해서 복제본 Primary VM, Secondary VM 두개를 만들어두어 다운타임 없이 서비스를 바로 복구시키는 원리
- vMotion 네트워크를 사용해서 실시간 메모리 비트맵을 공유함.
- 만약에 primary가 죽으면 secondary로 대체되고 secondary가 primary가되어 secondary를 하나 더 선정한다.
 - 그렇기 때문에 FT로 사용될 VM은 3개를 사용한다.

CPU Compatibility

CPU Constraints on vSphere vMotion Migration

CPU compatibility between source and target hosts is a vSphere vMotion requirement that must be met.

CPU Characteristics	Exact Match Required By Source Host and Target Host	Reason
Clock speeds, cache sizes, hyperthreading, and number of cores	N/A	The VMkernel virtualizes these characteristics.
Manufacturer (Intel or AMD) family and generation (Opteron4, Intel Westmere)	Applicable	Instruction sets contain many small differences.
Presence or absence of SSE3, SSSE3, or SSE4.1 instructions	Applicable	Multimedia instructions are usable directly by applications.
Virtualization hardware assist	For 32-bit VMs: N/A	The VMkernel virtualizes this characteristic.
	For 64-bit VMs on Intel: Applicable	Intel 64-bit with VMware implementation uses Intel VT.

- CPU의 명령어세트가 다르면 VM 마이그레이션이 불가능하다.
- 같은 종류의 cpu라도 시리즈가 다르다면? 불가
- 아예 다른 종류의 cpu 라면? 제조사가 다른것이라면 아예 불가
- 하이퍼 스레딩 x2 지원한다. VMkernel 덕분에
- CPU Compatibility를 보장하는 방법은 EVC를 사용하면 이 호환성을 맞춰줌

CPU Baseline Examples

CPU baselines hide from the guest OS any extra features that are on the newer hosts.

AMD Example

Scenario:

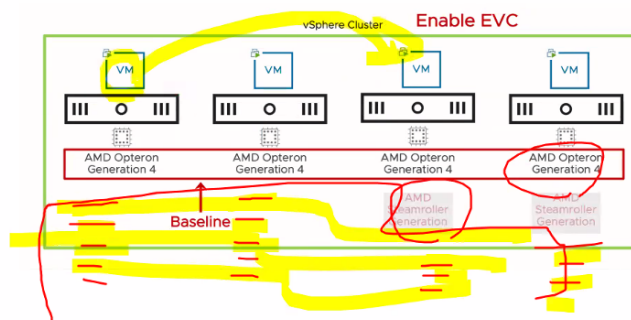
This cluster has two generations of processors:

- Older: AMD Opteron Generation 4
- Newer: AMD Steamroller Generation

Solution:

When enabling EVC for the cluster, select AMD Opteron Generation 4 as baseline.

The newer hosts still see AMD Steamroller Generation as their feature set, but EVC hides the new features from the VMs running on the newer hosts.



- 다른 버전의 cpu라면 마이그레이션이 안되는 베이스라인을 맞춰주어서 마이그레이션을 가능하게 만들어줌
- 이를 EVC라고 함.

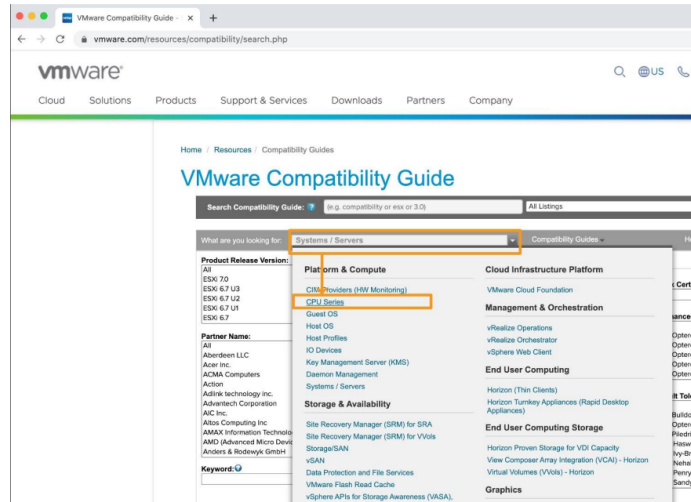
적용

1. 호스트 CPU 확인
2. 가이드라인 확인

2. View the VMware Compatibility Guide

Next, you go to the VMware Compatibility Guide at <https://www.vmware.com/resources/compatibility>.

You select **CPU Series** from the drop-down list.



3. CPU 버전확인

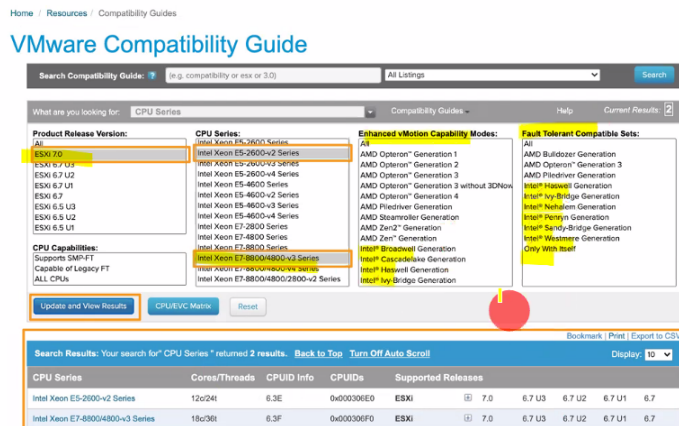
3. Select Product Version and CPU Series

You select the ESXi version that you are using in your cluster, for example, ESXi 7.0.

Then, you select all the processor types (CPU series) that are used by the hosts in your cluster.

In this example, let's say that the processor types used in the cluster are the following:

- Intel Xeon E5-2600-v2 Series
- Intel Xeon E7-8800/4800-v3 Series



초기 파악

1. H/W 종류 (시스코, 아리스타 등)
2. SW (OS, 솔루션종류)
3. 전체구성도

실습

HOL-2410-01-SDC: ESXi Installation and Configuration (vmware.com).

- 설치 시뮬레이션 url

Host에 접속하는 방법

- host client
- vSphere client
 - 만약에 web client라고 말해도 vSphere client라고 알아들으면 된다.

vmware hol에서 가이드보며 실습진행