Load Balancer

1. Load balancer란?

-서버에 가해지는 부하(load)를 분산(balancing)해주는 장치 또는 기술

-인터넷 서비스에 발생하는 트래픽이 많을 때, 여러 대의 서버가 분산처리하여 서버의 로드

증가, 부하량, 속도저하 등을 고려하여 적절히 분산처리하여 해결해주는 서비스

-클라이언트와 server pool(분산 네트워크를 구성하는 서버들의 그룹) 사이에 위치

-한 대의 서버로 부하가 집중되지 않도록 트래픽을 관리해, 각각의 서버가 최적의 퍼포먼스

를 보일 수 있도록 함

-여러 대의 서버를 두고 서비스를 제공하는 분산 처리 시스템에 필요한 기술

-scale up: 서버가 더 빠르게 동작하기 위해 서버 자체의 성능을 향상

-scale out: 서버를 여러 대 증설하여 운영. 여러 대의 서버로 트래픽을 균등하게 분산

(여러 대의 서버가 나눠서 일을 하는 것)

🡪 로드밸런싱

2. load balancer를 사용하는 이유 (scale out의 장점)

-H/W를 향상하는 방법보다 비용이 절감

-무중단 서비스를 제공

-유연한 구성이 가능

-짧은 대기 시간과 높은 처리량

3. load balancer의 종류

-공용 load balancer

-들어오는 트래픽의 Public IP주소와 포트를 VM의 사설IP 및 포트에 매핑

-특정 유형의 트래픽을 여러 VM 또는 서비스에 분산할 수 있음

-내부 load balancer

-Vnet 내부에 있는 리소스 또는 VPN을 사용하여 Azure인프라에 접근하는 리소스

로만 트래픽을 보냄

-Vnet 에서 동일한 Vnet으로 부하 분산

4. load balaning의 다양한 방식

-라운드로빈 방식(round robin method): 서버에 들어온 요청을 순서대로 돌아가며 배정

-여러 대의 서버가 동일한 스펙을 가짐

-서버와의 세션이 오래 지속되지 않는 경우에 활용하기 적합

-가중 라운드로빈 방식(weighted round robin method)

-각각의 서버마다 가중치를 매기고 가중치가 높은 서버에 클라이언트 요청을 우선적

으로 배분

-주로 서버의 트래픽 처리 능력이 상이한 경우 사용

-IP 해시 방식(IP Hash Method): 클라이언트 IP 주소를 특정 서버로 매핑하여 요청을 처리

-사용자가 항상 동일한 서버로 연결되는 것을 보장

-최소 연결 방식(least connection method)

-요청이 들어온 시점에 가장 적은 연결상태를 보이는 서버에 우선적으로 트래픽을 배

분

-서버에 분배된 트래픽들이 일정하지 않은 경우에 적합

-트래픽으로 인해 세션이 길어지는 경우 권장

-최소 리스폰타임(least reponse time method)

-서버의 현재 연결 상태와 응답시간을 모두 고려하여 트래픽을 배분

-가장 적은 연결 상태와 가장 짧은 응답시간을 보이는 서버에 우선적으로 로드를 배

분

5. L4 Load balance 와 L7 load balancer 비교

①L4: 네트워크 계층(IP, IPX)이나 전송 계층(TCP, UDP)의 정보를 바탕으로 로드를 분산

-IP주소나 포트번호, MAC주소, 전송 프로토콜에 따라 트래픽을 나누는 것이 가능

-데이터 안을 들여다보지 않고 패킷 레벨에서만 로드를 분산하기 때문에 속도가 빠

르고 효율이 높음

-데이터의 내용을 복호화할 필요가 없기에 안전

-L7 로드밸런서 보다 가격이 저렴

-패킷의 내용을 살펴볼 수 없기 때문에 섬세한 라우팅이 불가능

-사용자의 IP가 수시로 바뀌는 경우라면 연속적인 서비스를 제공하기 어려움

②L7: 애플리케이션 계층(HTTP, FTP, SMTP)에서 로드를 분산

-TCP/UDP 정보는 물론 HTTP의 URL, FTP의 파일명, 쿠키 정보 등을 바탕으로 함

-상위 계층에서 로드를 분산하기 때문에 훨씬 더 섬세한 라우팅이 가능

-캐싱 기능을 제공

-사용자의 요청을 기준으로 특정 서버에 트래픽을 분산하는 것이 가능

-패킷의 내용을 확인하고 그 내용에 따라 로드를 특정 서버에 분배

-특정한 패턴을 지닌 바이러스를 감지해 네트워크를 보호

6. Load balancer 구성 요소

-프로브: 백 엔드 풀에 있는 인스턴스의 상태를 확인하기 위해 사용

-프로브가 응답하지 않으면 load balancer는 비정상 인스턴스에 새 연결의 전송을 중지

-프로브 오류는 기존 연결에 영향을 주지 않음

-응용 프로그램에서 흐름을 종료하거나, VM이 종료될 때까지 연결이 계속됨

7. load balancer 의 기능

-NAT(Network Address Translation): 사설 IP주소를 공인IP 주소로 바꾸는데 사용하는 통신

망의 주소 변조기

-인바운드 NAT 규칙을 사용하여 특정 VM의 특정 포트에 트래픽을 전달

-터널링: 인터넷 상에서 눈에 보이지 않는 통로를 만들어 통신할 수 있게 하는 개념

데이터를 캡슐화해서 연결된 상호 간에만 패킷을 구별해 캡슐화를 해제

-DSR(Dynamic Source Routing protocol): 로드 밸런서 사용시 서버에서 클라이언트로 되돌

아가는 경우, 목적지 주소를 스위치의 IP 주소가 아닌 클라이언트의 IP주소로 전달

해서 네트워크 스위치를 거치지 않고 바로 클라이언트를 찾아가는 개념

-자동 재구성: 인스턴스를 확장 또는 축소하는 경우 즉시 재구성

-백 엔드 풀에서 VM을 추가 또는 제거하면 추가 작업 없이 load balancer가 다

시 구성됨

-아웃바운드 연결(SNAT): 가상 네트워크 내의 개인 IP 주소에서 인터넷의 퍼블릭 IP 주소로

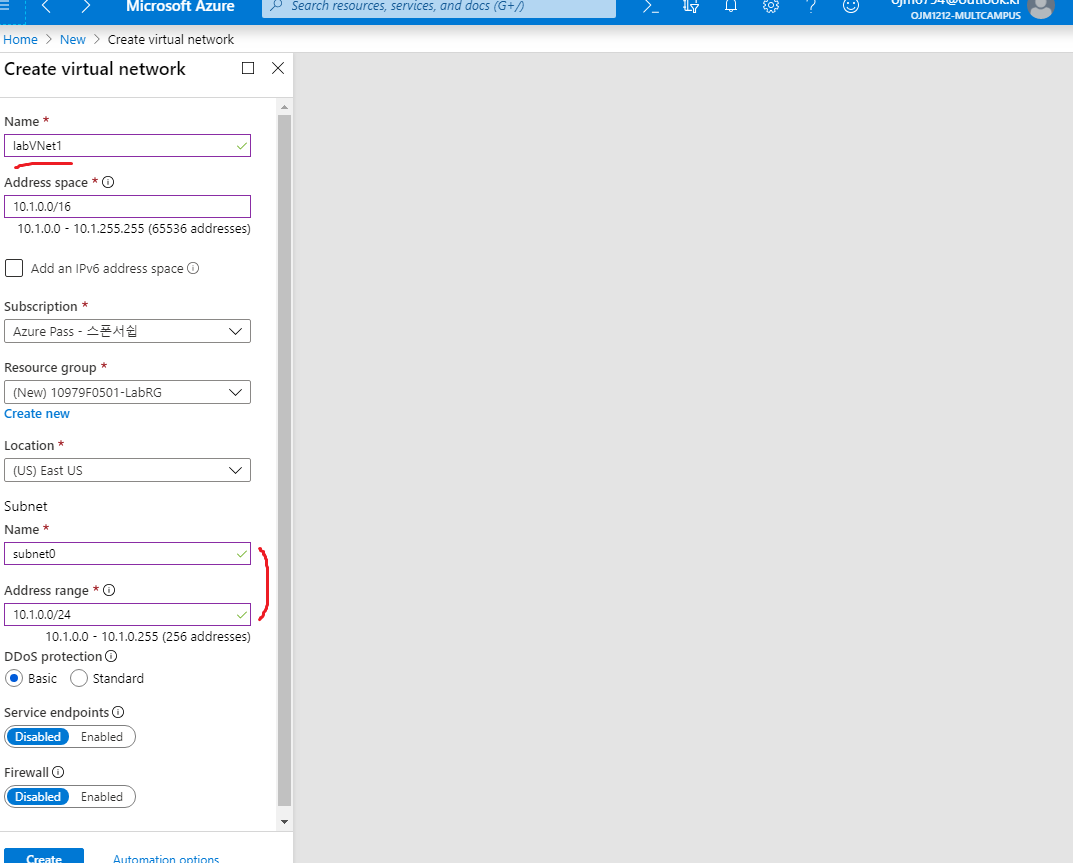
전달되는 모든 아웃바운드 흐름은 Load Balancer의 프런트 엔드 IP 주소로 변환될 수 있음

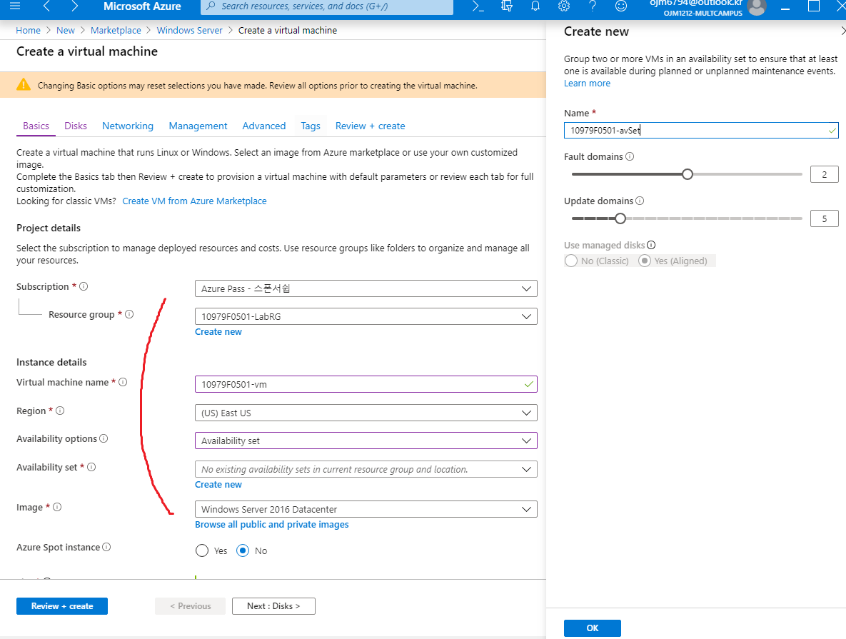
-퍼블릭 프런트 엔드가 부하 분산 규칙을 통해 백 엔드 VM에 연결되면 Azure는 아웃바운드 연결을 퍼블릭 프런트 엔드 IP 주소로 변환

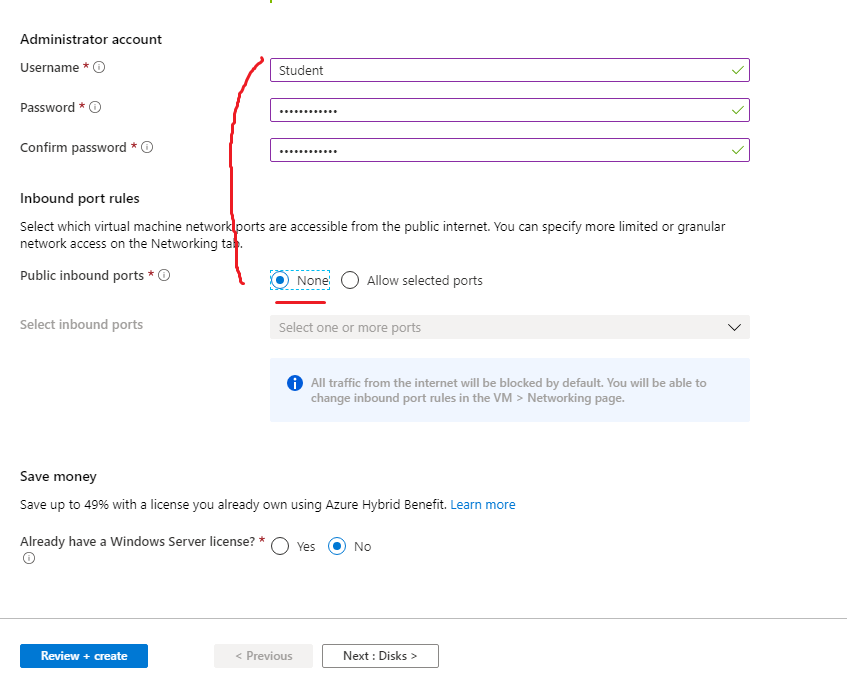
-프런트 엔드를 서비스의 다른 인스턴스에 동적으로 매핑할 수 있으므로 서비스의 업그레이드 및 재해 복구가 용이

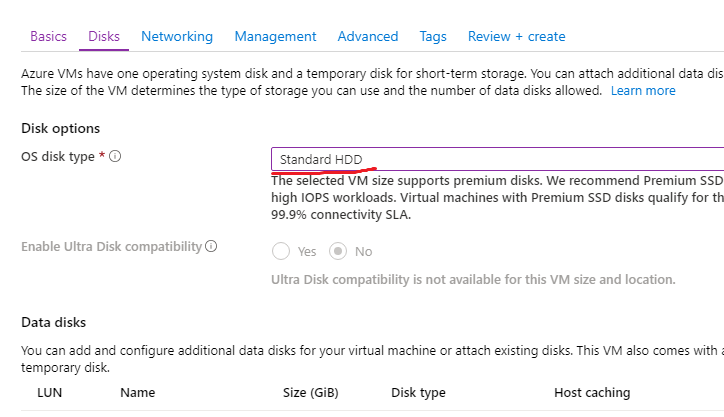
-ACL(액세스 제어 목록)을 보다 쉽게 관리할 수 있습니다. 프런트 엔드 IP로 표현되는 ACL은 서비스를 확장 또는 축소하거나 다시 배포해도 변경되지 않습니다. 아웃바운드 연결을 컴퓨터보다 더 적은 수의 IP 주소로 변환하면 수신 허용 - 받는 사람 목록을 구현해야 한다는 부담을 줄일 수 있다

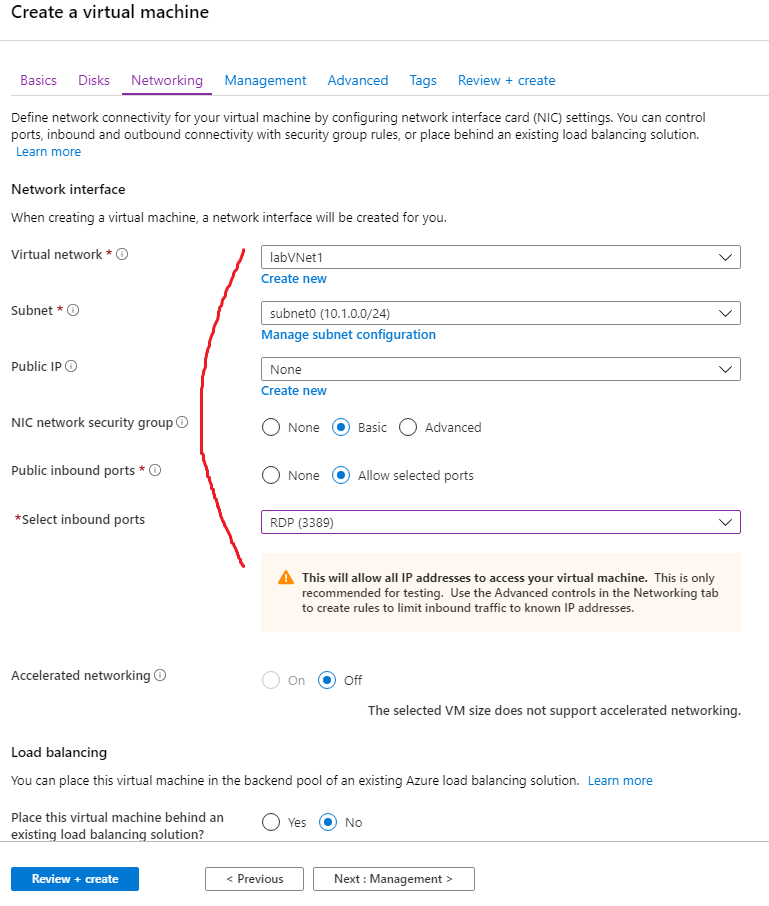
8. Load balancer 생성

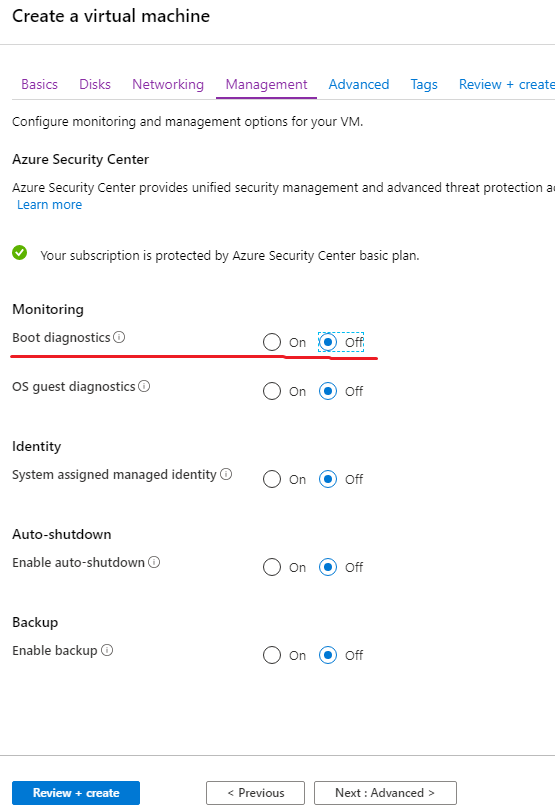
①리소스 그룹 만들기

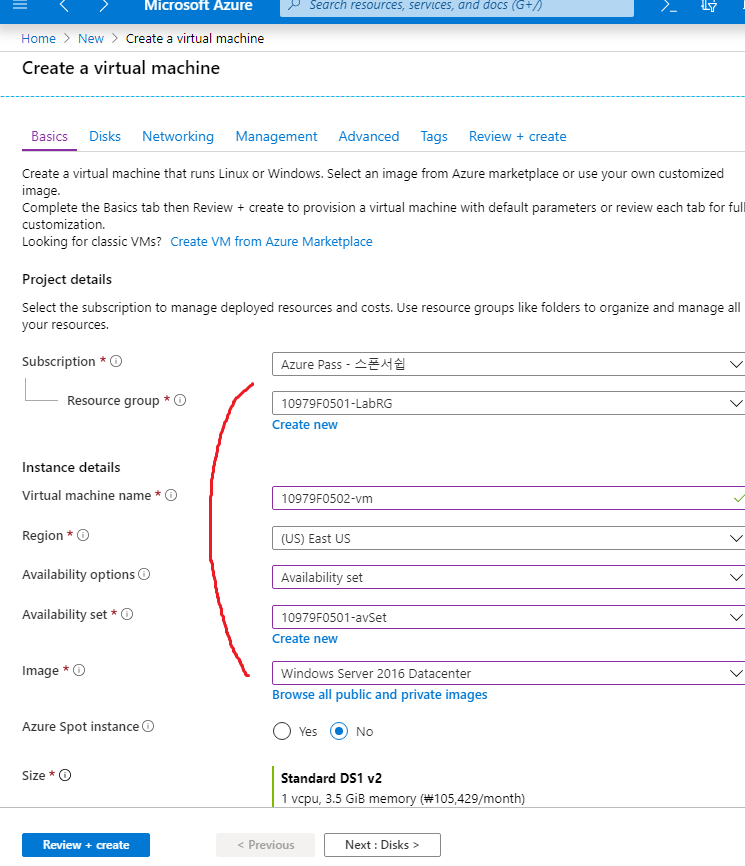
② 두개의 Vnet과 VM 만들기

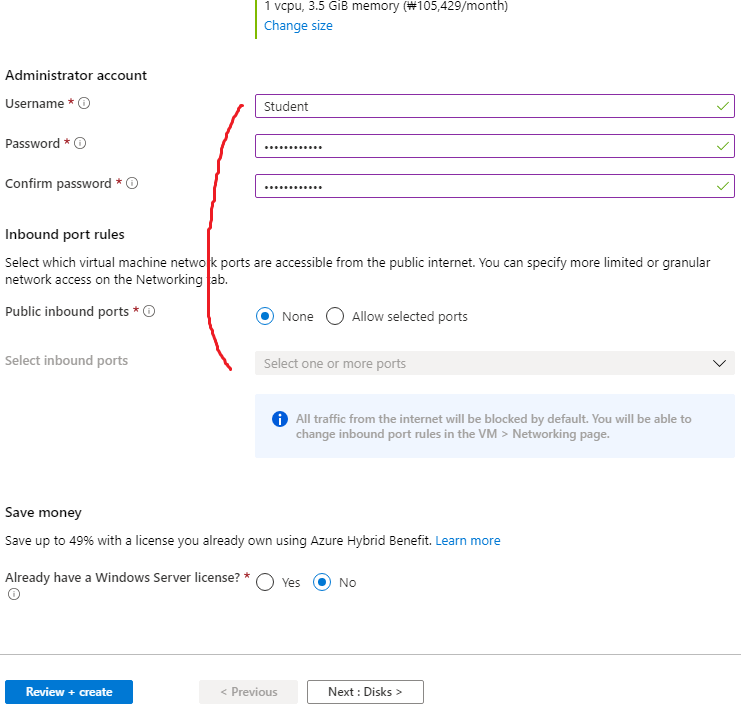


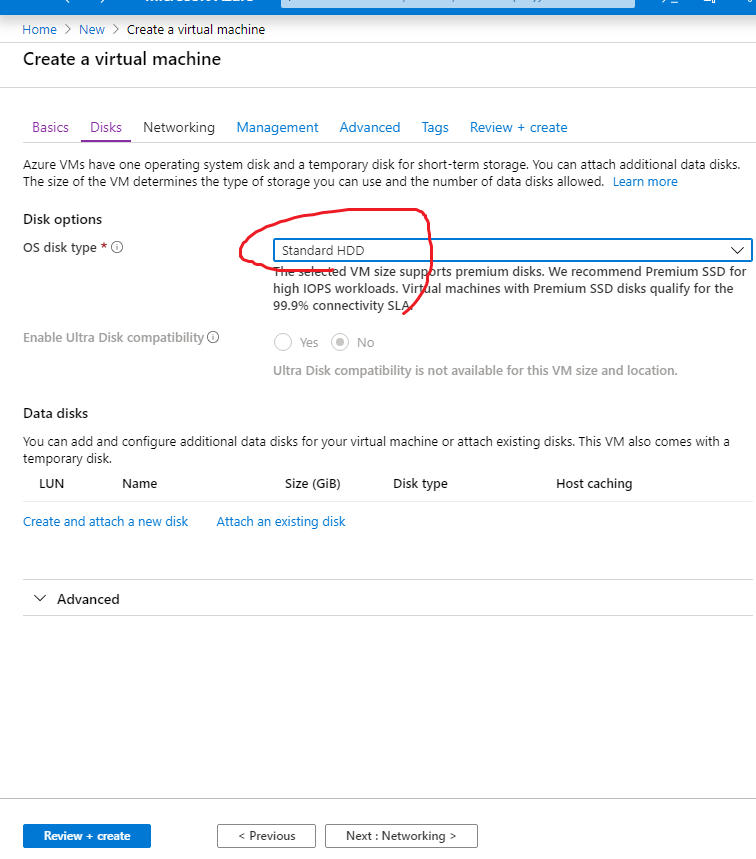


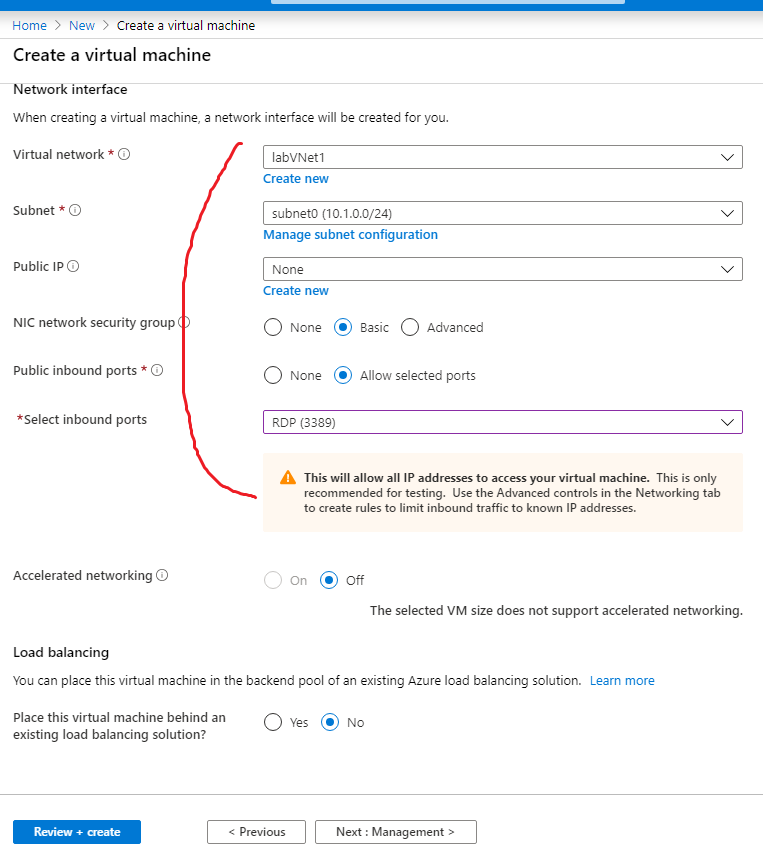


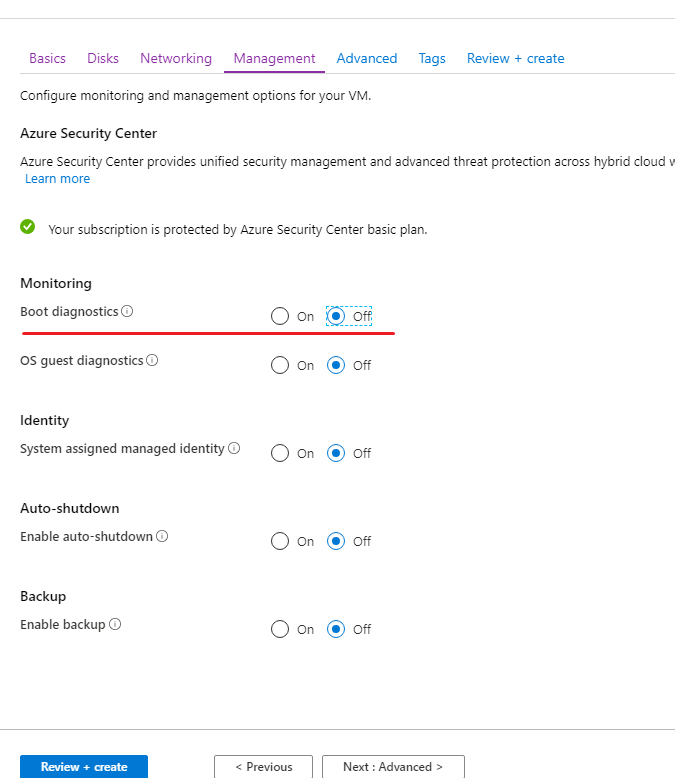


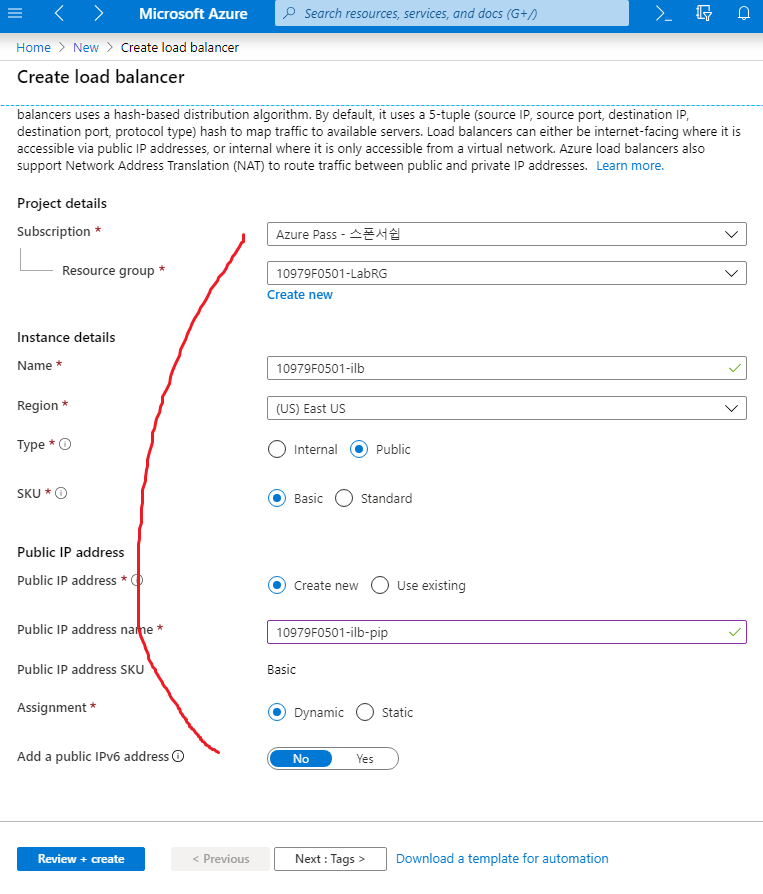


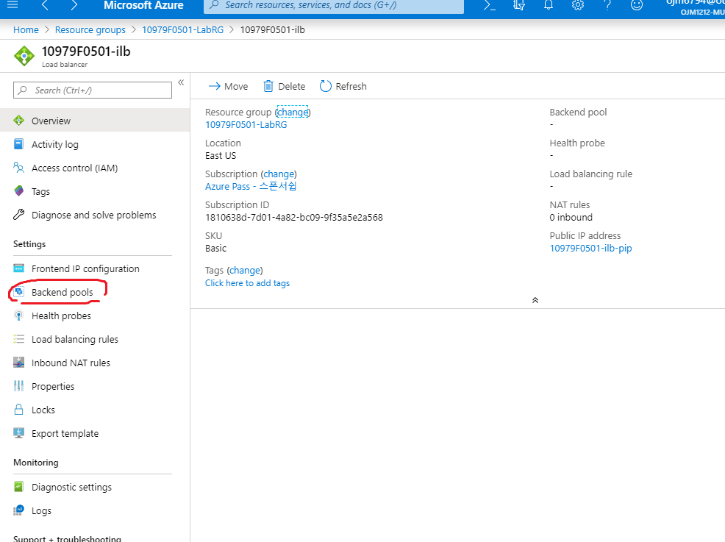
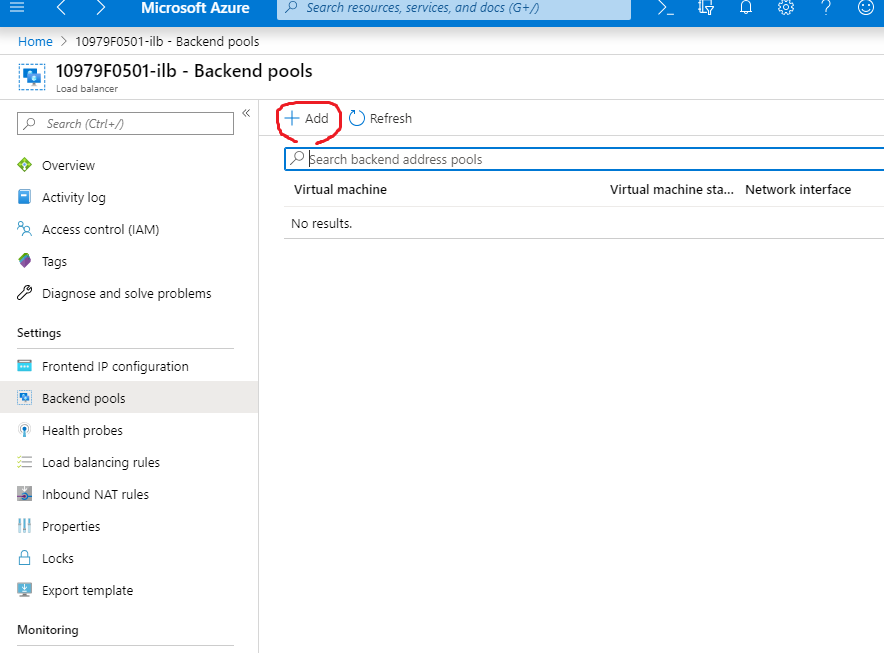


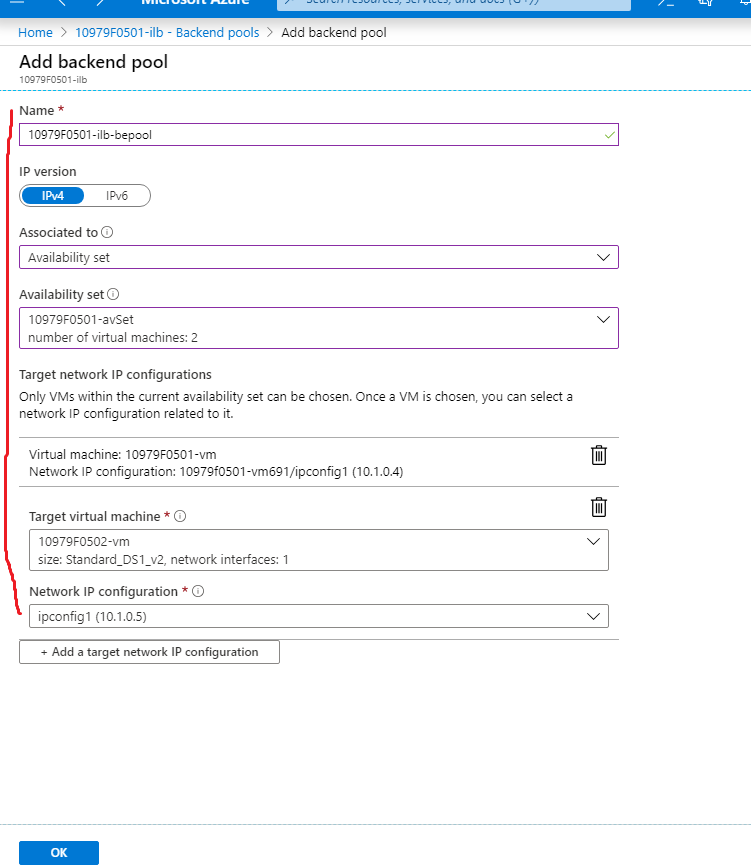


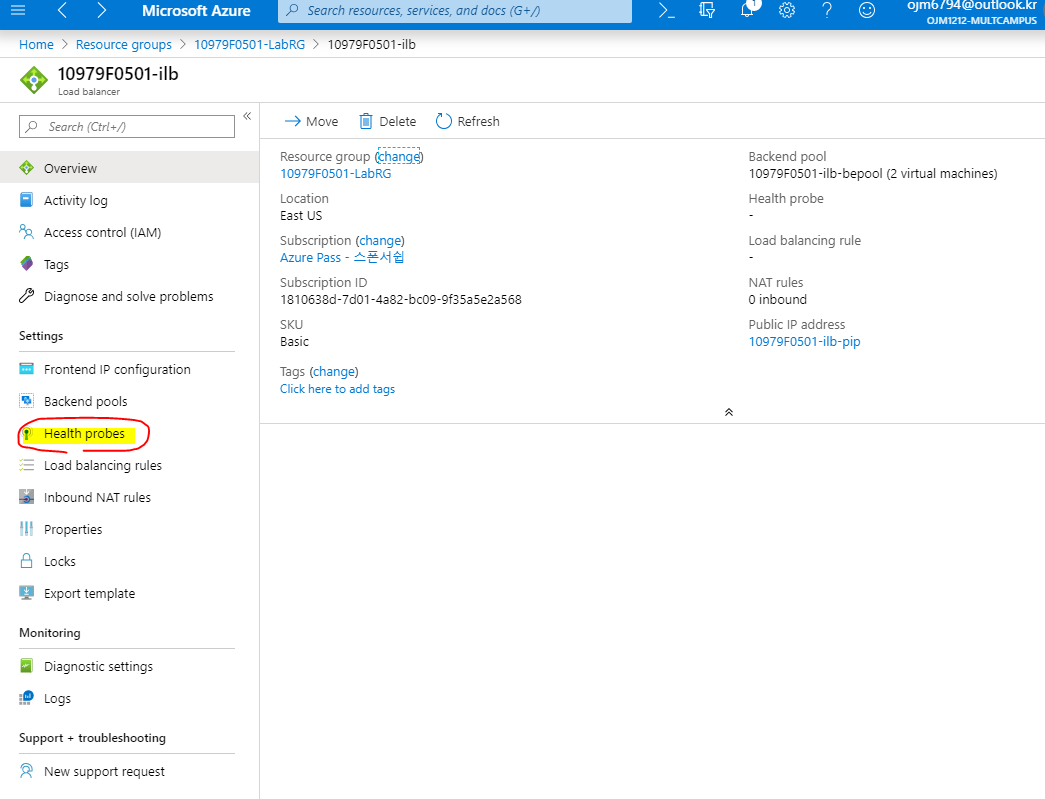


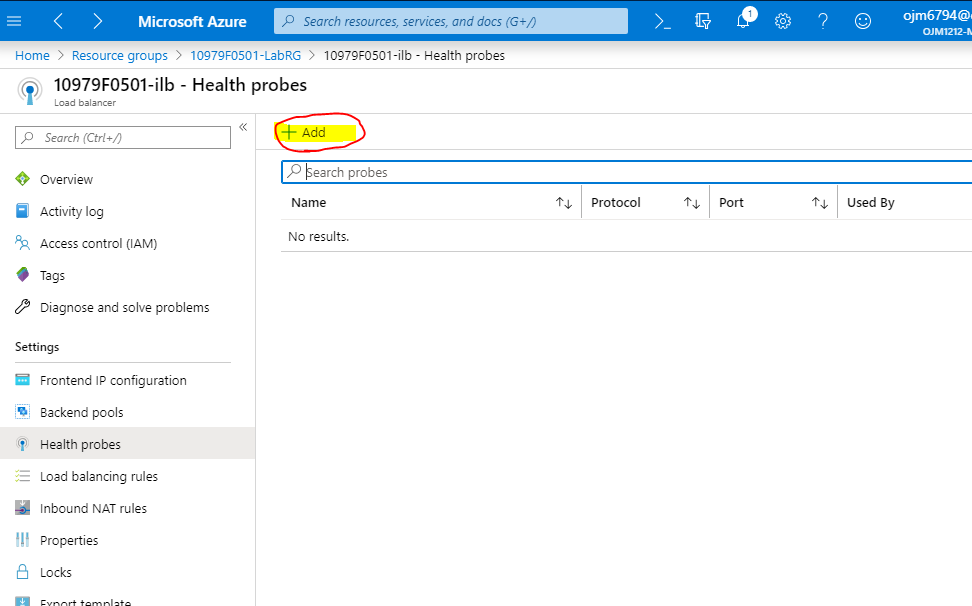


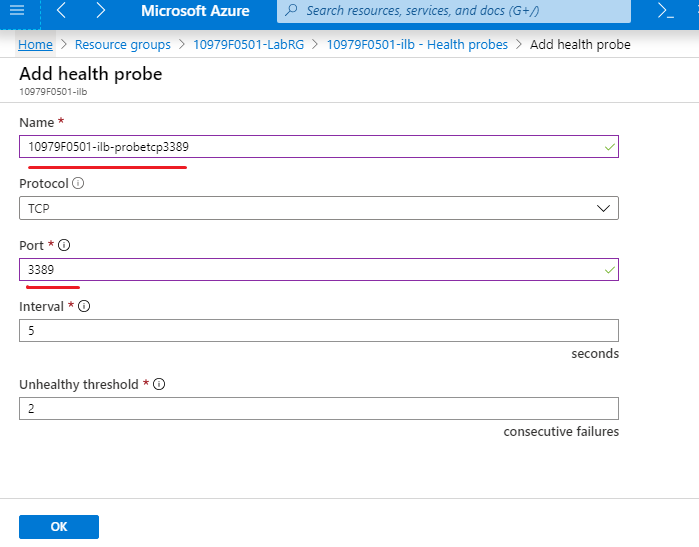
③Load Balancer 만들기

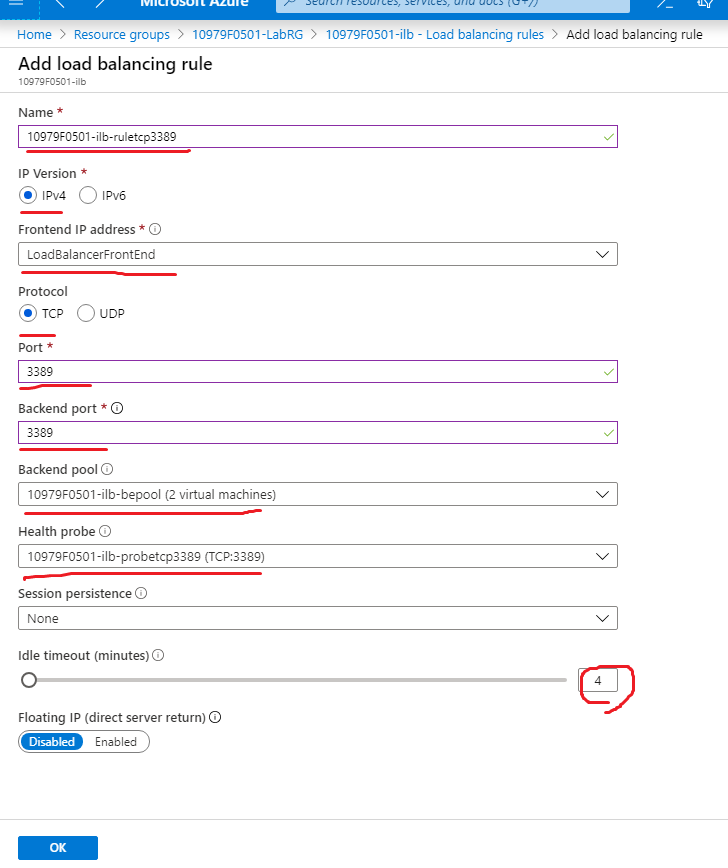
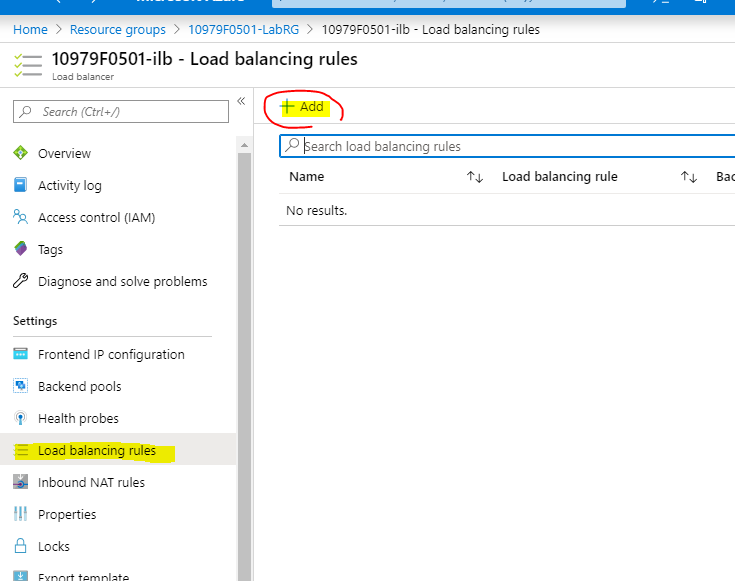
④Backend pool 만들기

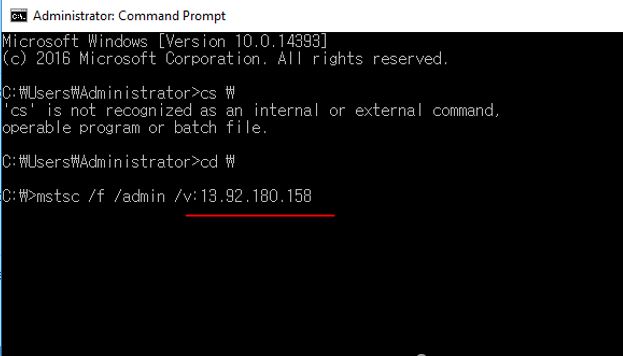


⑤probe 만들기





⑥Load balancing rule만들기

⑦실행

