|  |
| --- |
| **[600003] 클라우드컴퓨팅** |
| **~~실습 #07~~ => 숙제 #03 문제 및 보고서** |

|  |  |
| --- | --- |
| **이름** | 지현한 |
| **학번** | 20165164 |
| **소속**  **학과/대학** | 소프트웨어융합대학 빅데이터전공 |
| **분반** | 01 (담당교수: 김태운) |

## <주의사항>

* 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
* 파일명에 본인의 이름과 학번을 입력하세요.
* 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.
  + 소스코드/스크립트 등을 작성 한 경우, 본 문서에 코드를 붙여 넣거나 또는 별도의 파일로 첨부해서 제출하세요. 별도의 파일로 제출하는 경우 해당 파일의 이름도 적어주세요.
* 스마트캠퍼스 제출 데드라인: 10. 29. (목요일) 23:59 // 중간고사 다음주 목요일 까지
  + 데드라인을 지나서 제출하면 24시간 단위로 25%감점(4일 경과 시 0점)
  + 주말/휴일/학교행사 등 모든 날짜 카운트 함
  + 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0점 처리함
  + 예외 없음
* 스마트캠퍼스에 아래의 파일을 제출 해 주세요
  + 보고서(PDF 파일로 변환 후 제출 권장. 워드 문서로 제출해도 됨)
  + 보고서 파일명에 이름과 학번을 입력 해 주세요.
  + (소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 작성한 모든 파일을 본 문서에 붙여 넣기 하거나 또는 첨부파일로 제출)

## <개요>

이번 과제는 지난주에 이어서 Microsoft 의 클라우드 컴퓨팅 서비스인 Azure를 사용하는 내용입니다.

## <실습 과제: 기본>

|  |
| --- |
| **[Q 0] 요약 [20점]**  이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지를 3문장 이상으로 요약하세요. |

답변:

Azure를 통해 부하 분산자를 만드는 방법을 알게 되었습니다

Azure를 통해 가상머신 확장 집합을 만드는 방법을 알게 되었습니다

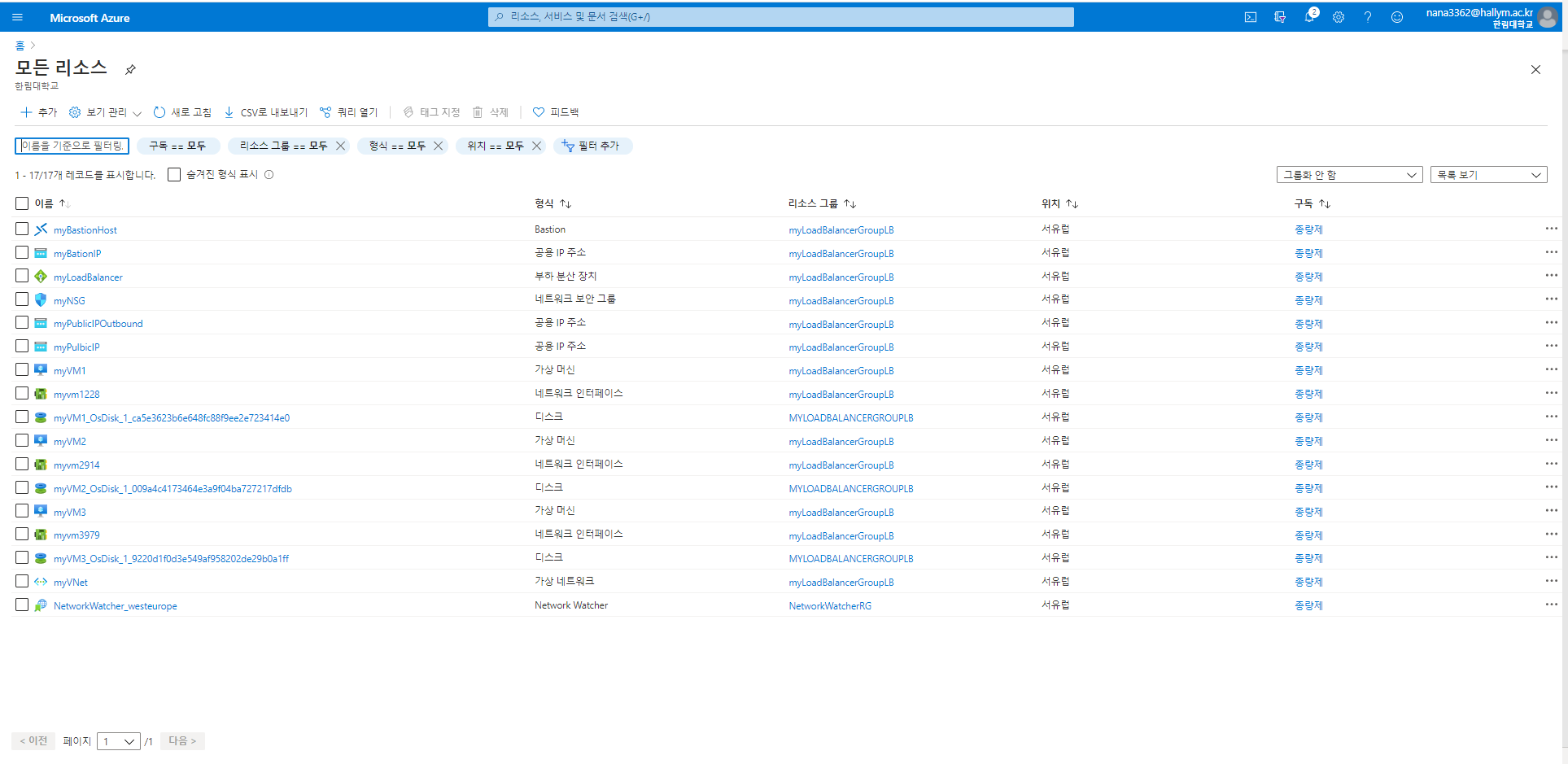
이전 과제로 $100가 모두 소진되어 있어서 결제수단을 등록하고 새로 구독을 만들어서 과제를 진행하였습니다

\*\* 주의 \*\*

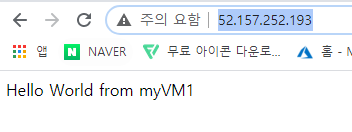
* 지난 과제와는 달리, 이번 과제에서 생성하는 리소스는 비용이 청구될 수 있습니다. Azure for Student 계정에는 $100 크레딧이 있지만, 과도하게 사용하면 금방 소진될 수 있습니다. 아래의 과제는 최대한 빠르게 완료하도록 하고, 과제가 완료된 뒤에는 모든 리소스를 삭제하여 불필요한 비용 청구를 방지하세요. 과제를 수행하는 도중 장시간 다른 일을 해야 한다면 모든 리소스를 삭제하고 나중에 다시 하거나, 또는 모든 리소스를 “사용 중단” 해서 불필요한 과금을 피하세요. 무료 요금을 초과하면 계정이 정지됩니다!!!
* 삭제되지 않는 리소스가 있다면, 모든 ‘리소스 그룹’을 삭제한 뒤에 다시 시도해 보세요.
* 주기적으로 잔여 크레딧을 확인하세요: <https://www.microsoftazuresponsorships.com/Balance>
* 화면 캡쳐 시, portal 웹 전체 화면(+ Cloud Shell이 있는 경우에는 터미널화면 포함)을 캡쳐하고, 우측 상단에 로그인 아이디 및 디렉터리 정보가 나와야 합니다.

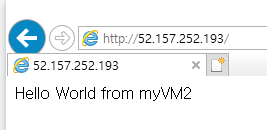
|  |
| --- |
| **[Q 1] Load Balancer (부하 분산) [30점]**  아래의 내용을 읽고 ‘부하 분산’에 대해 공부하세요.  https://docs.microsoft.com/ko-kr/azure/load-balancer/  아래의 Tutorial을 따라서 부하 분산 시스템을 구현하세요.  <https://docs.microsoft.com/ko-kr/azure/load-balancer/quickstart-load-balancer-standard-public-portal?tabs=option-1-create-load-balancer-standard>  [문제 1] ‘모든 리소스’ 페이지로 이동해서 전체 화면을 캡처한 뒤(우측 상단에 로그인 ID가 보여야 함) 아래에 첨부하세요. (아래의 [예시 화면 1]을 참고하세요.)  [문제 2] 컴퓨터, 노트북, 스마트폰 등에서 웹 브라우저를 이용해서 Load Balancer 공인 IP에 접속하세요. VM1, VM2, VM3에 접속한 화면을 각각 1개씩 첨부하세요. (아래의 [예시 화면 2]를 참고하세요)  [문제 3] ‘모든 리소스’에서 ‘myLoadBalancer’를 클릭하고, ‘백 엔드 풀’에서 ‘myBackendPool’을 클릭하세요. 전체 화면을 캡처한 뒤(우측 상단에 로그인 ID가 보여야 함) 아래에 첨부하세요. (아래의 [예시 화면 3]을 참고하세요.)  \*\* portal.azure.com의 ‘리소스 그룹’에 있는 모든 리소스 그룹을 삭제하세요. 또한, portal.azure.com의 ‘모든 리소스’에 있는 모든 자원을 삭제하세요. |

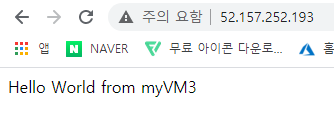
답변 1:



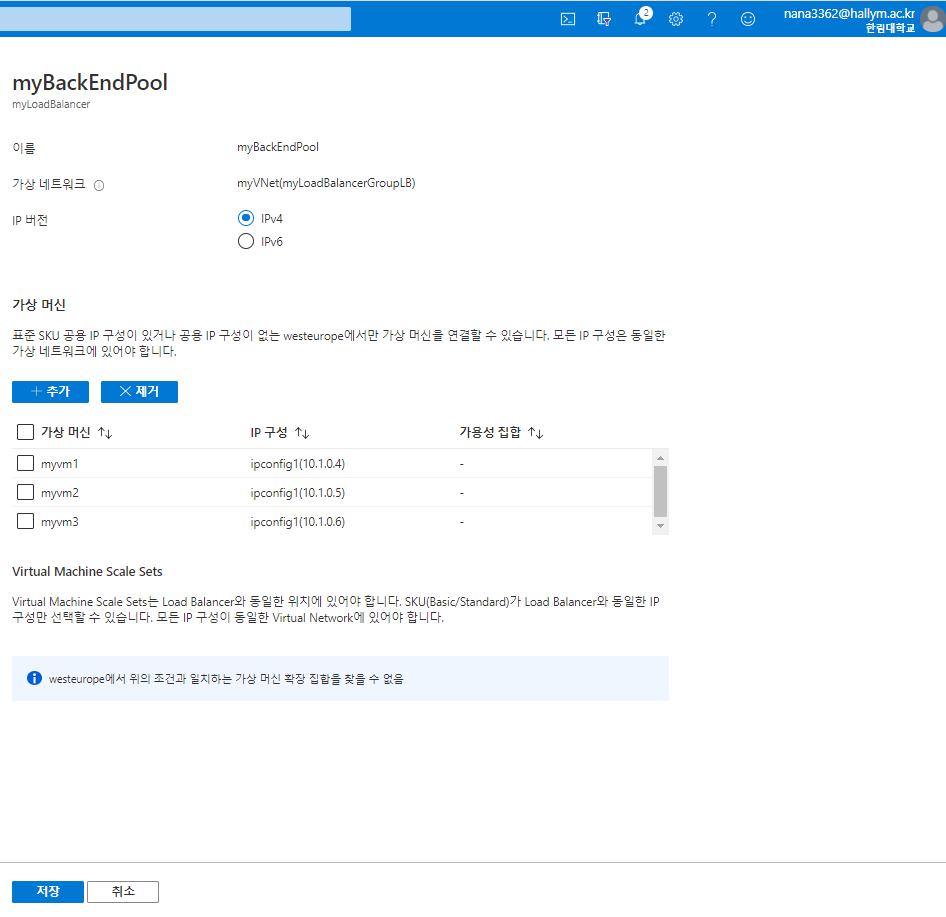
답변 2:



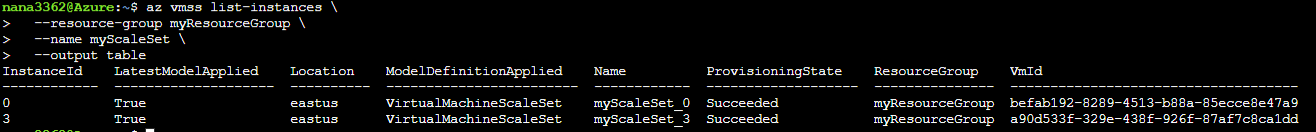




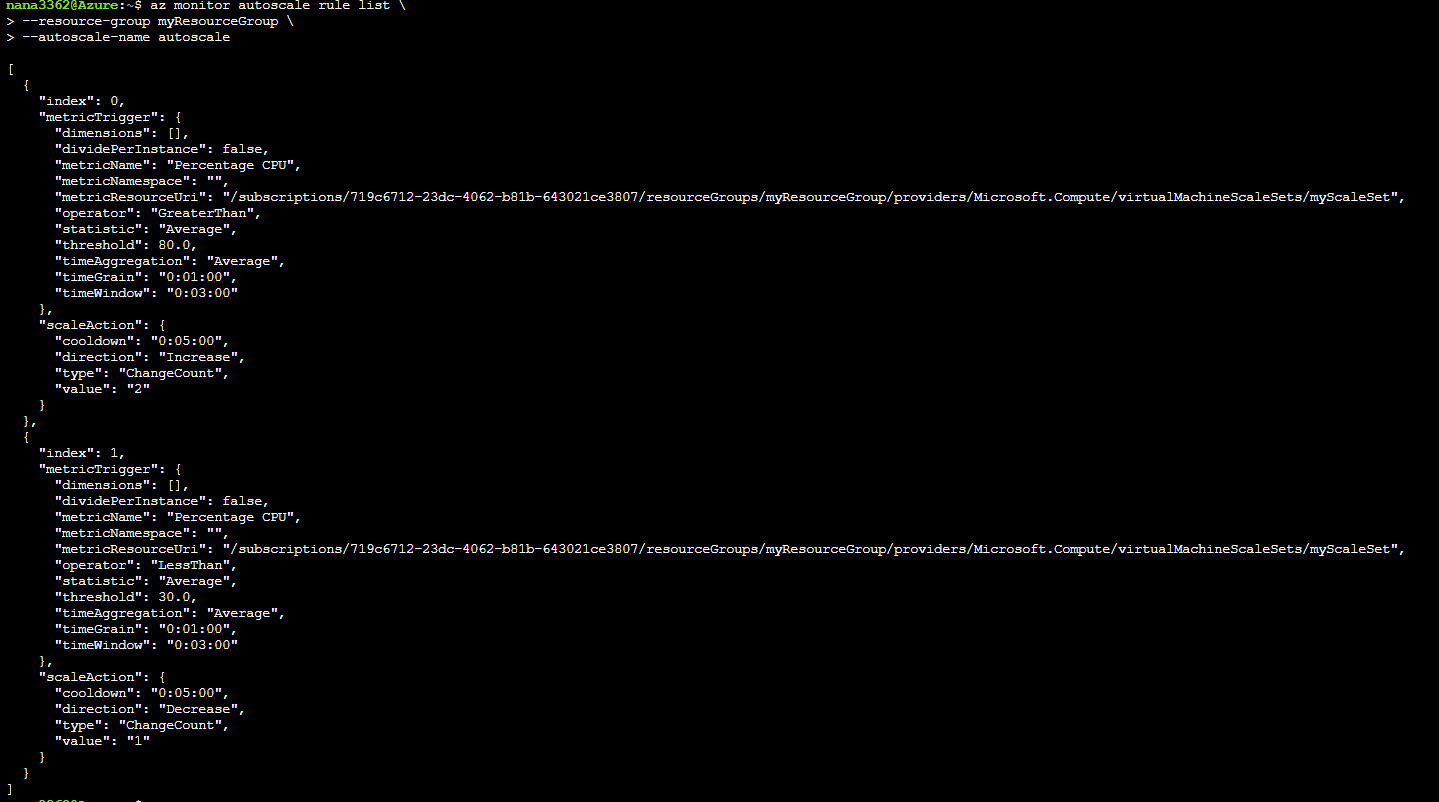
답변 3:



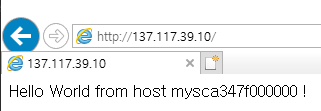
|  |
| --- |
| **[Q 2] Virtual Machine Scale Set (VMSS) 가상 머신 확장 집합 [50점]**  아래의 내용을 읽고, ‘가상 머신 확장 집합 (Virtual Machine Scale Set)’에 대해 공부하세요: https://docs.microsoft.com/ko-kr/azure/virtual-machine-scale-sets/overview  이번 문제에서는 command-line 기반의 Azure CLI (Azure Cloud Shell-Bash)를 사용해서 가상 자원을 관리합니다. VMSS 자습서를 참고하여, 아래의 과제를 완료하세요.  <자습서 위치>  <https://docs.microsoft.com/ko-kr/azure/virtual-machine-scale-sets/> 에 접속한 후, 화면 좌측의 자습서 클릭.    아래의 과제를 수행하기 위해서는 자습서 ‘1-확장 집합 만들기/관리’ 부터 ‘5-확장 집합 자동 크기 조정’ 내용을 모두 숙지하고 있어야 합니다. 아래의 [예시 화면 4]번부터 시작하는 화면을 참고하세요.  <과제>  [1] portal.azure.com 에서 Cloud Shell - Bash를 생성하고 (이 과정에서 스토리지도 생성함), 터미널에서 echo $0을 입력하여 bash shell을 사용하고 있는 것을 확인하세요.  [2] ‘myResourceGroup’ 이라는 리소스 그룹을 East-US 지역에 생성하세요. 지금부터 아래의 모든 과정은 ‘myResourceGroup’ 그룹 내에 정의됩니다.  [3] ‘az vmss create’명령을 사용해서 ‘myScaleSet’이라는 이름으로 가상 머신 확장 집합을 만드시오. 인스턴스 count가 2인 확장 집합을 만들고, SSH 키를 생성하도록 설정하시오. 가상머신 이미지는 ‘UbuntuLTS’를 사용하고, ‘upgrade-policy-mode’는 ‘automatic’으로 설정하시오. 관리자 계정 ID는 ‘azureuser’로 설정하시오.  ‘az vmss list-instances’ 명령을 사용해서, 확장 집합의 VM 인스턴스를 테이블 형식으로 터미널에 출력하시오.  🡺 (문제 1) 터미널 화면 캡처하여 아래에 붙여넣기  ‘az vmss get-instance-view’ 명령을 사용해서, 1번 및 2번 VM 인스턴스의 정보를 터미널에 출력하시오.  [4] ‘az monitor autoscale create’명령을 사용해서 자동 크기 조정 프로필을 정의하세요. 'resource-type’은 ‘Microsoft.Compute/virtualMachineScaleSets’, ‘name’은 ‘autoscale', 최소 VM 수는 2개, 최대 VM 수는 10개, 기본 인스턴스 수(= ‘count')는 2개로 설정하세요.  [5] 텍스트 에디터를 사용해서 VM 확장 시 사용할 JSON 스크립트를 생성하세요. 스크립트 이름을 customConfig.json으로 설정하세요. 스크립트는 nginx 웹 서버를 설치하고, VM 인스턴스의 호스트 이름을 출력하도록 구성되어 있고, 소스코드의 주소는 다음과 같음:  <https://raw.githubusercontent.com/Azure-Samples/compute-automation-configurations/master/automate_nginx.sh>  [6] ‘az monitor autoscale rule’명령을 사용해서 자동 크기 확장 규칙을 만드세요. CPU 사용량이 평균 80%를 초과한 상태로 3분이상 지속되면 VM 수를 +2 하세요.  [7] ‘az monitor autoscale rule’명령을 사용해서 자동 크기 축소 규칙을 만드세요. CPU 사용량이 평균 30% 미만인 상태로 3분이상 지속되면 VM 수를 -1 하세요.  'az monitor autoscale rule’ 명령을 사용해서, 지금까지 추가한 규칙을 터미널에 출력하세요.  🡺 (문제 2) 터미널 화면 캡처하여 아래에 붙여넣기  [8] ‘az vmss extension set’ 명령을 사용해서 지난 [5]번 에서 생성한 custonConfig.json 스크립트를 적용하여 확장 하도록 설정하세요. ‘publisher’는 ‘Microsoft.Azure.Extensions', ‘version’은 '2.0', 'name’은 ‘CustomScript’로 설정하세요.  [9] 트래픽(=사용자 요청)이 VM 웹 서버에 도달하도록 LB rule을 추가하세요. ‘az network lb rule create’ 명령을 사용하고, ‘name’은 ‘myLoadBalancerRuleWeb’, ‘lb-name’은 ‘myScaleSetLB’, ‘backend-pool-name’은 ‘myScaleSetLBBEPool’, ‘backend-port’는 ‘80’, ‘frontend-ip-name’은 ‘loadBalancerFrontEnd’, ‘frontend-port’는 ‘80’, ‘protocol’은 ‘tcp’로 설정하세요.  [10] 부하 분산 장치의 대표 IP를 가져오기 위해 ‘az network public-ip show’명령을 사용하세요. 웹 브라우저를 실행시키고, 해당 주소로 접속해서 웹 서버가 정상 동작 하는지를 확인하세요. 웹 브라우저를 강제 새로고침 하여, 2개의 VM에 접속 가능한지를 확인하세오.  🡺 (문제 3) 웹 브라우저에서 VM #1번과 VM #2번에 접속한 화면을 각각 캡처하여 아래에 붙여넣기  [11] ‘az vmss list-instance-connection-info’ 명령으로 Scale Set에 포함된 VM 1번과 2번에 각각 접속하는 주소/포트번호를 확인하고, 각각 VM에 접속한 뒤, 다음을 실행하세요.  sudo apt-get update  sudo apt-get -y install stress  sudo stress --cpu 10 --timeout 300 &  위의 명령은 CPU 로드를 생성하는 10개의 stress 작업자를 300초 동안 실행합니다.  다음으로, ‘watch az vmss list-instances’ 명령을 사용해서 VM 인스턴스 수가 어떻게 변화하는지 확인하세요 (약 2초마다 자동으로 화면이 갱신됩니다). 최소 11분 이상 지켜 보면서(휴지 기간 기본값: 5분), VM이 4개까지 늘어나는 것(= scale out)과 VM이 2개까지 줄어드는 것(= scale in)을 확인하세요.  🡺 (문제 4) VM 인스턴스가 4개까지 늘어난 상황에서 터미널 화면을 캡처하여 아래에 붙여넣기  🡺 (문제 5) 아래의 [예시 화면 4]와 같이 ‘모든 리소스’ 화면을 캡처하여 아래에 붙여넣기  [12] ‘az group delete --name myResourceGroup --yes --no-wait’ 명령을 실행해서 모든 리소스를 삭제하고, portal.azure.com의 ‘리소스 그룹’에 남아있는 모든 리소스 그룹을 삭제하세요. 또한, portal.azure.com의 ‘모든 리소스’에 남아있는 모든 자원을 삭제하세요. |

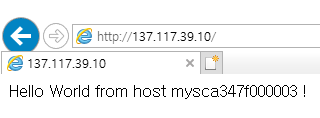
답변 1: 

답변 2:

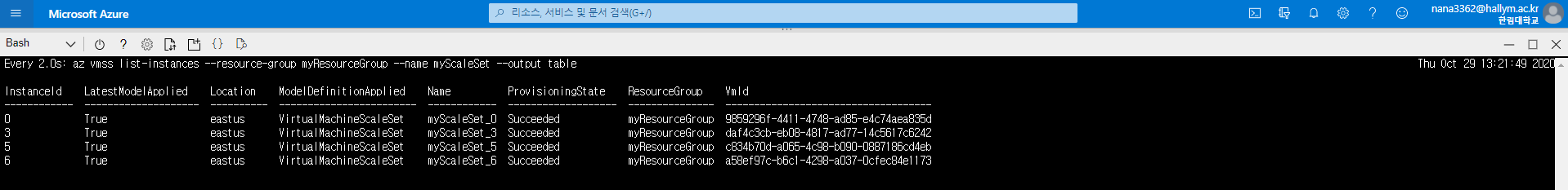


답변 3:

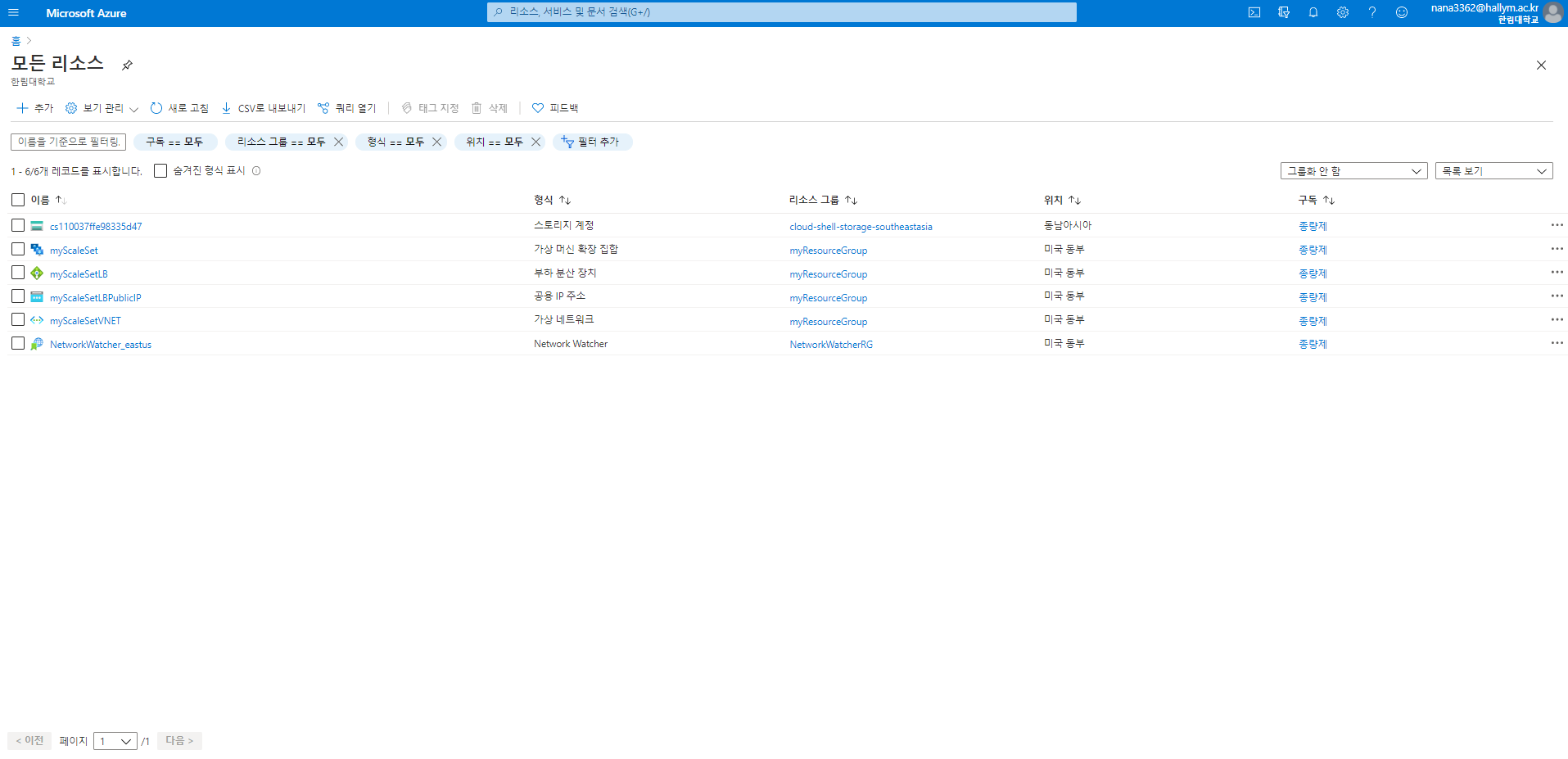




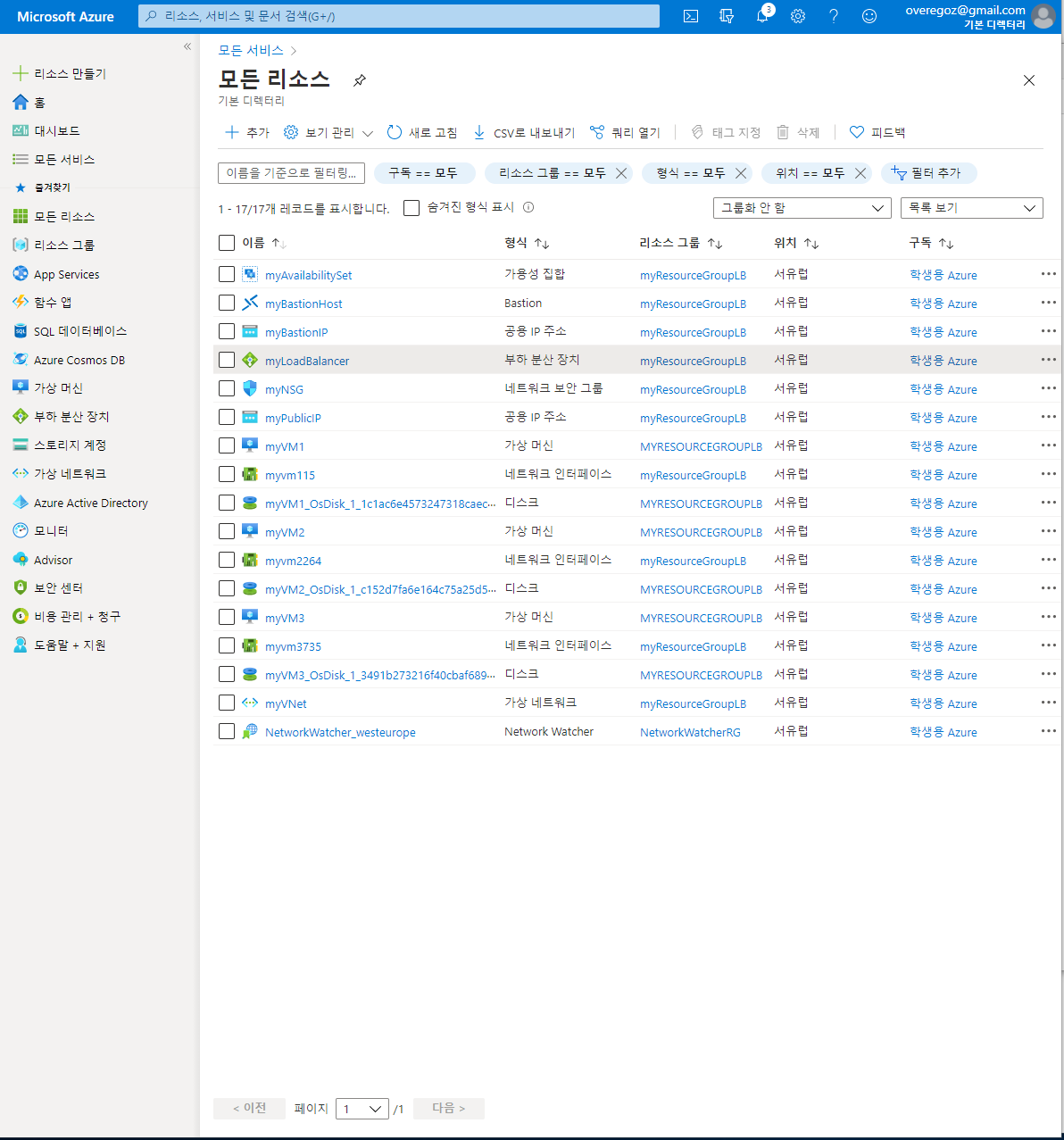
답변 4:



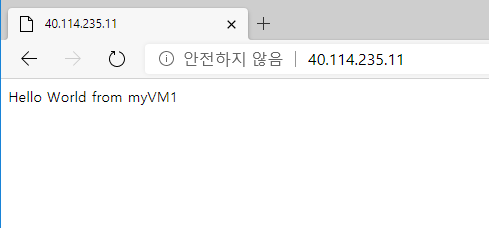
답변 5:

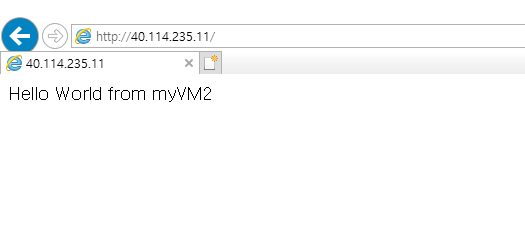


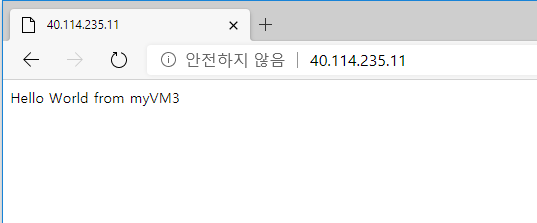
[예시 화면 1] [Q1] LB : 모든 리소스



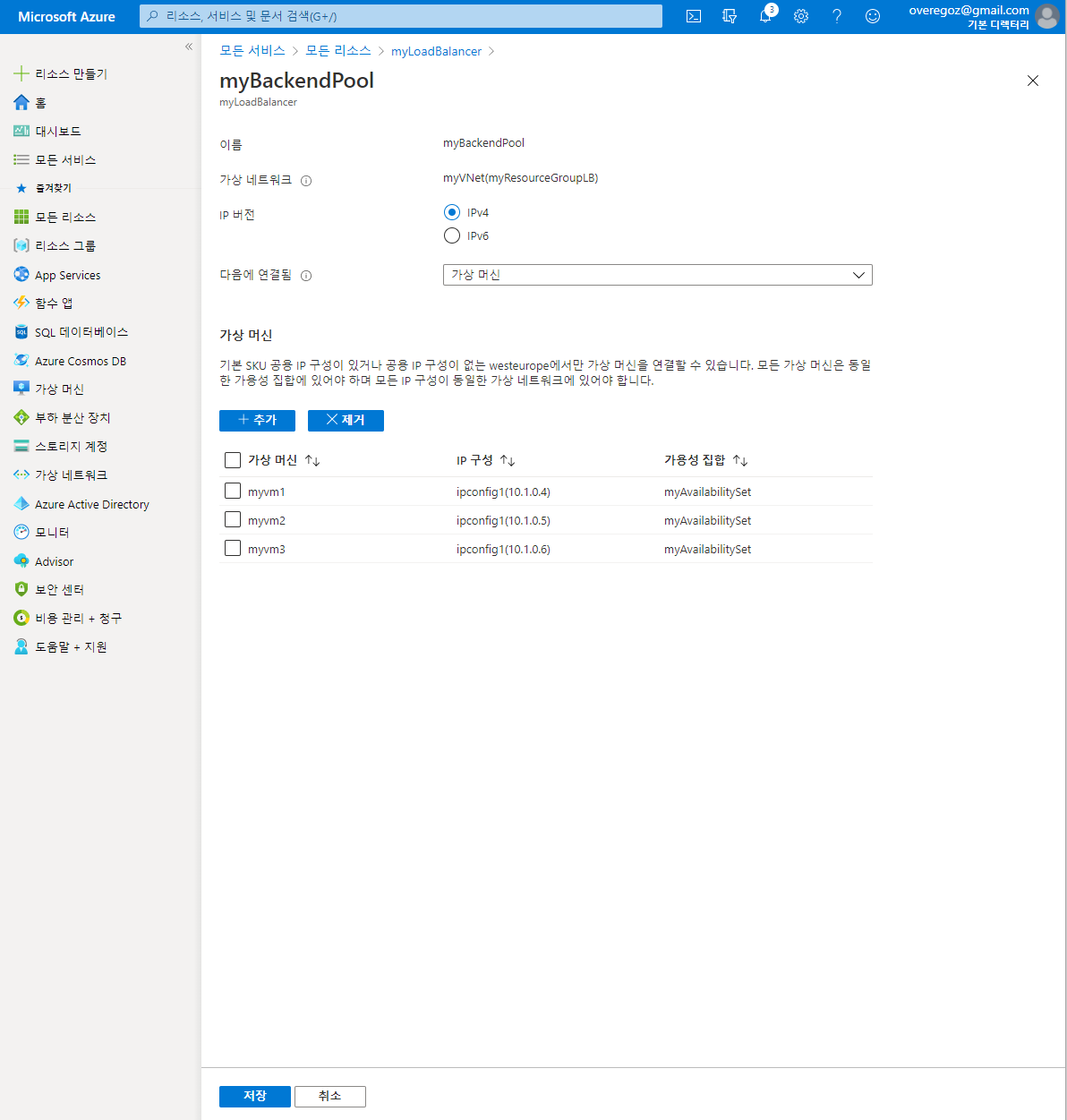
[예시 화면2] [Q1] 가상 머신 3대에 각각 접속한 화면 (웹 브라우저)



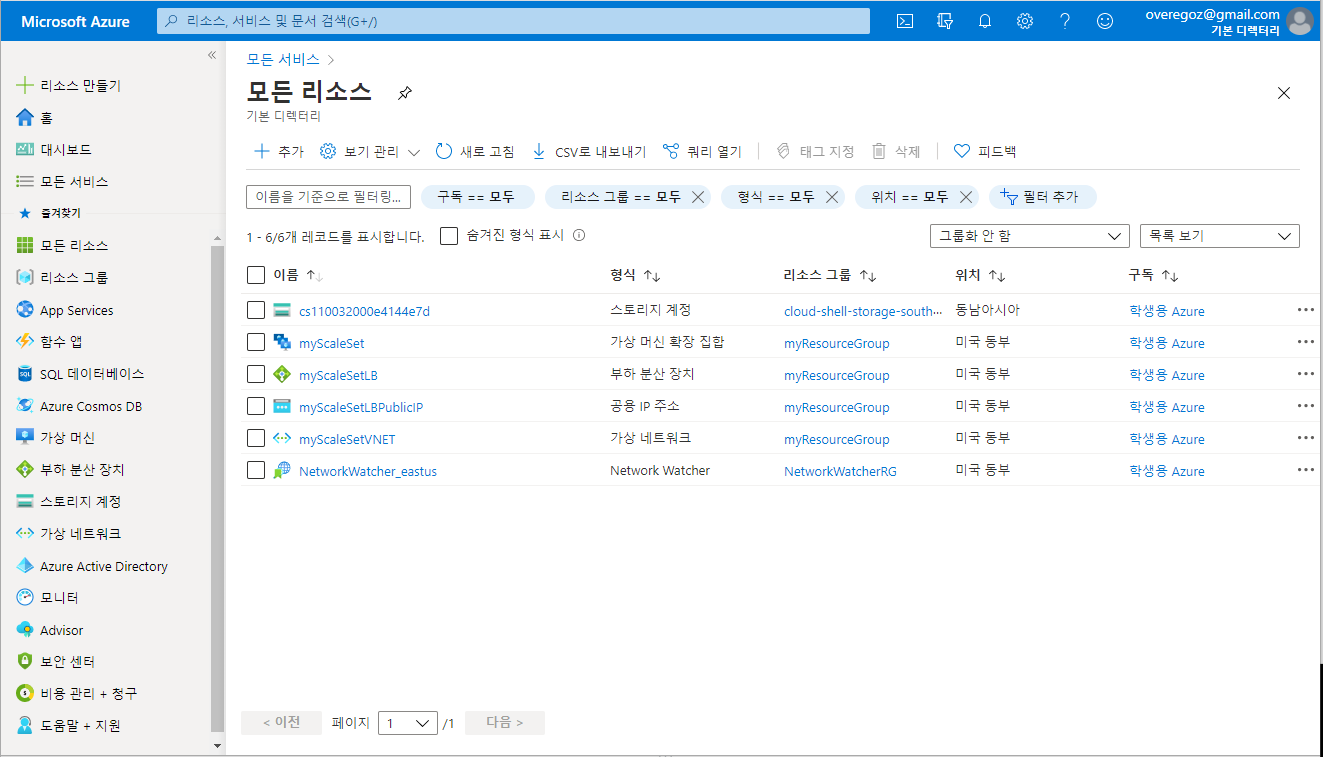




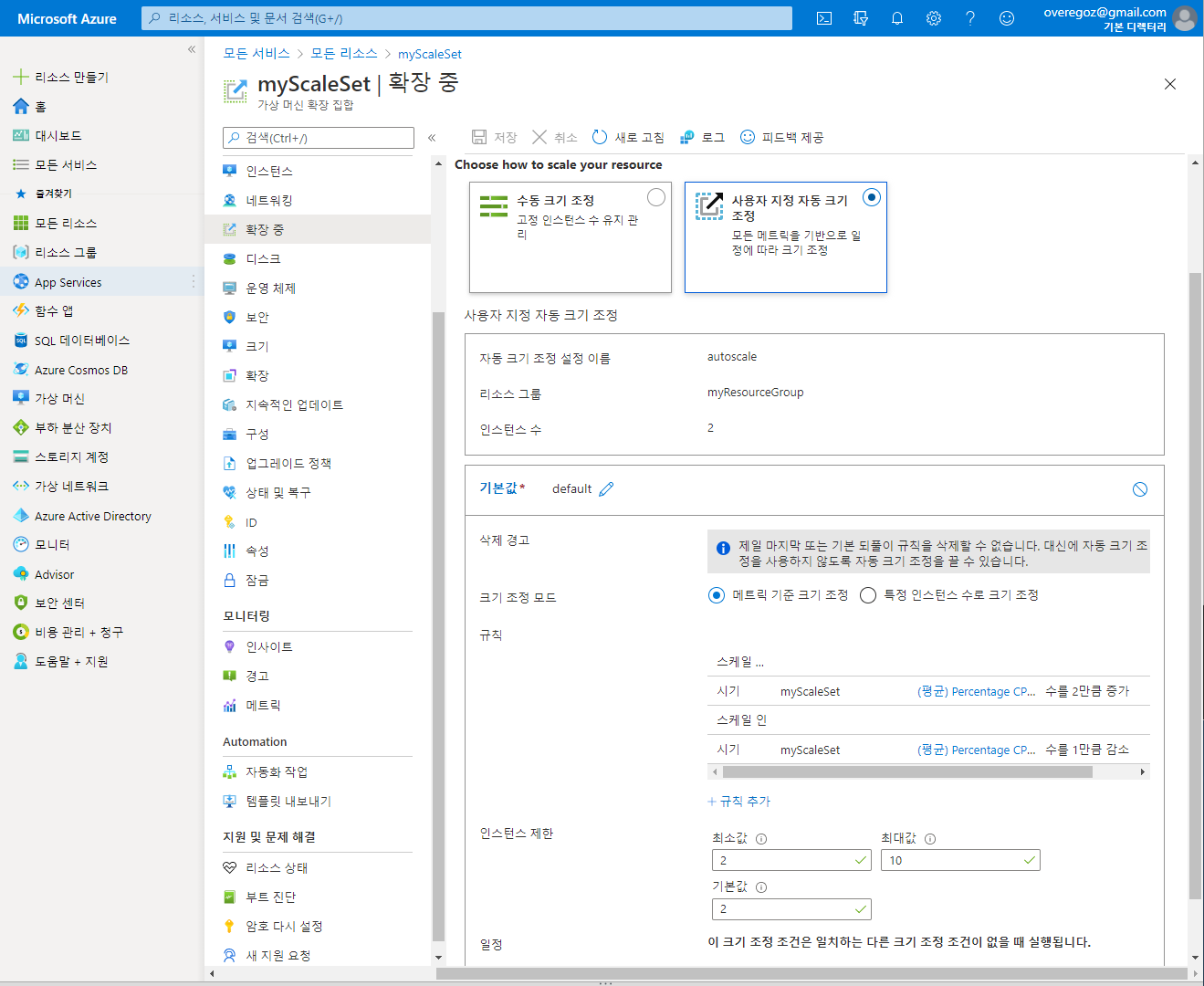
[예시 화면 3] [Q1] myLoadBalancer > myBackendPool



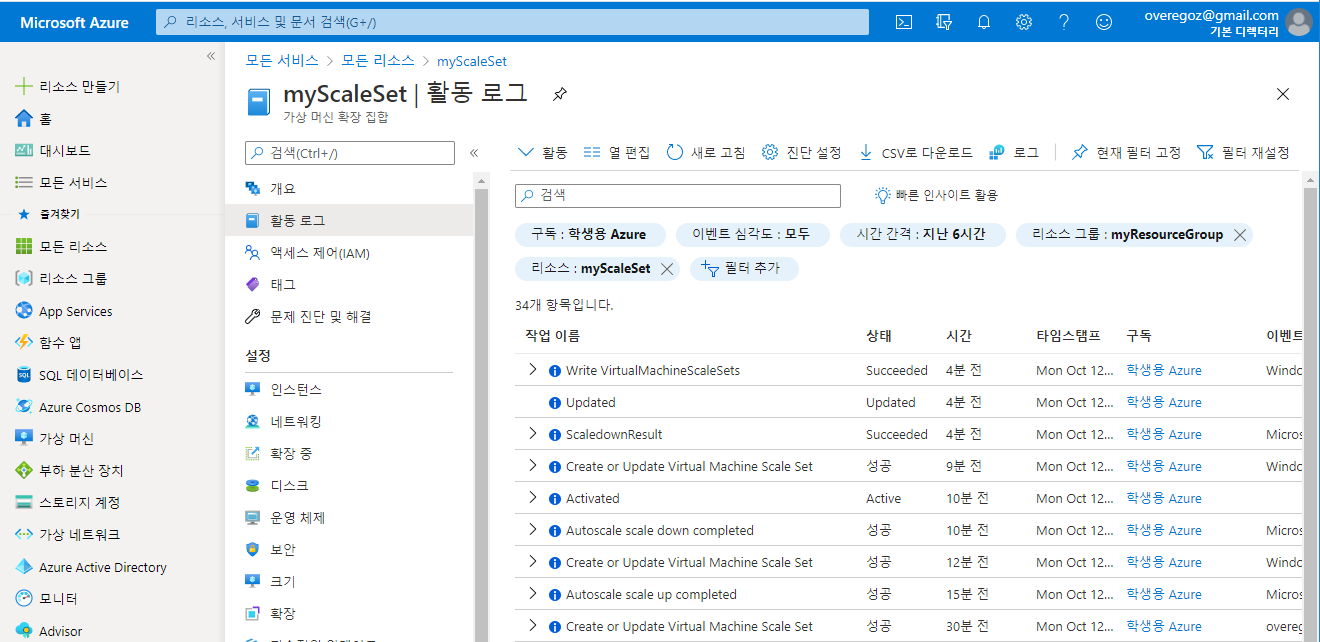
[예시 화면 4] [Q 2] 완료 후 ‘모든 리소스’



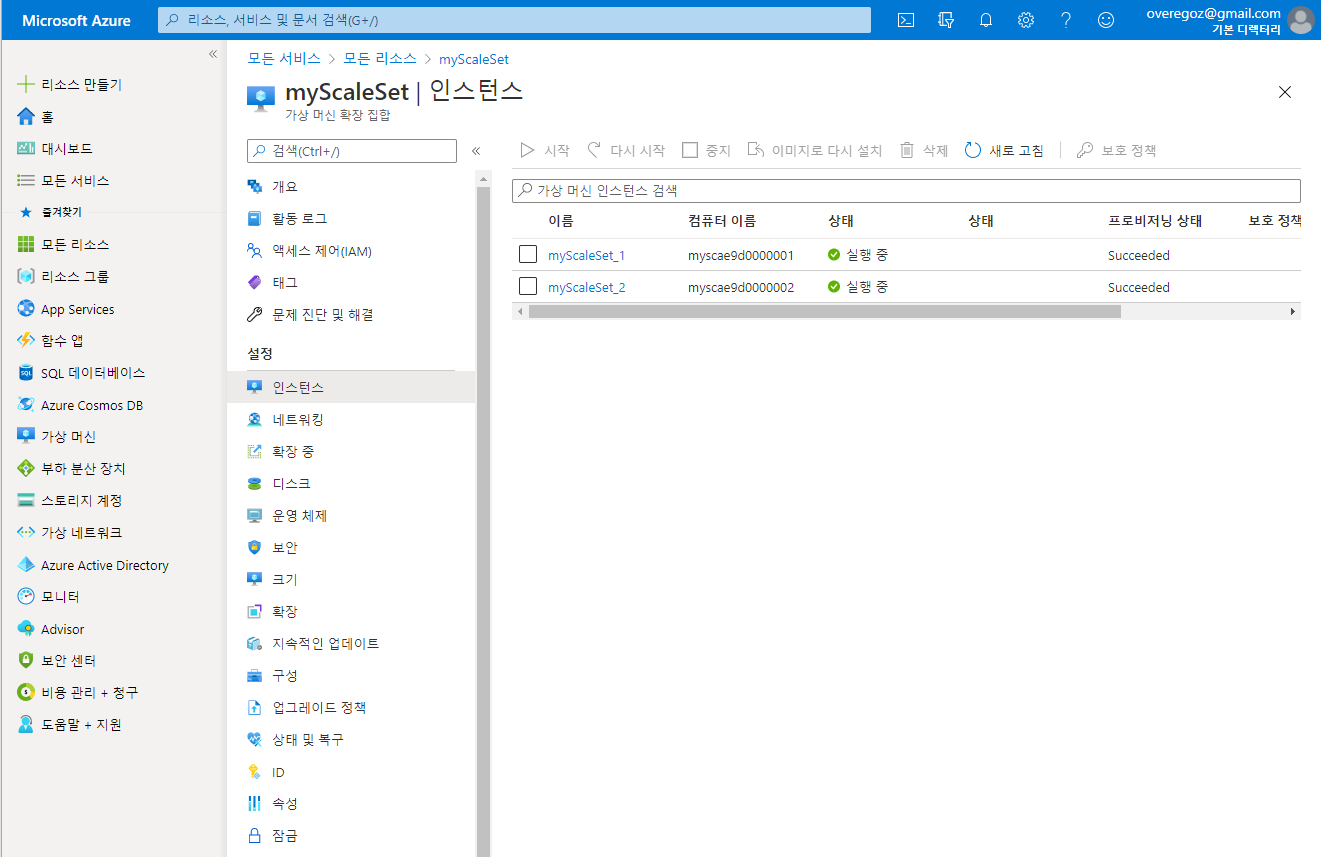
[예시 화면5] [Q2] 완료 후, ‘모든 리소스 > myScaleSet > 확장 중’에서 확인한 확장 규칙



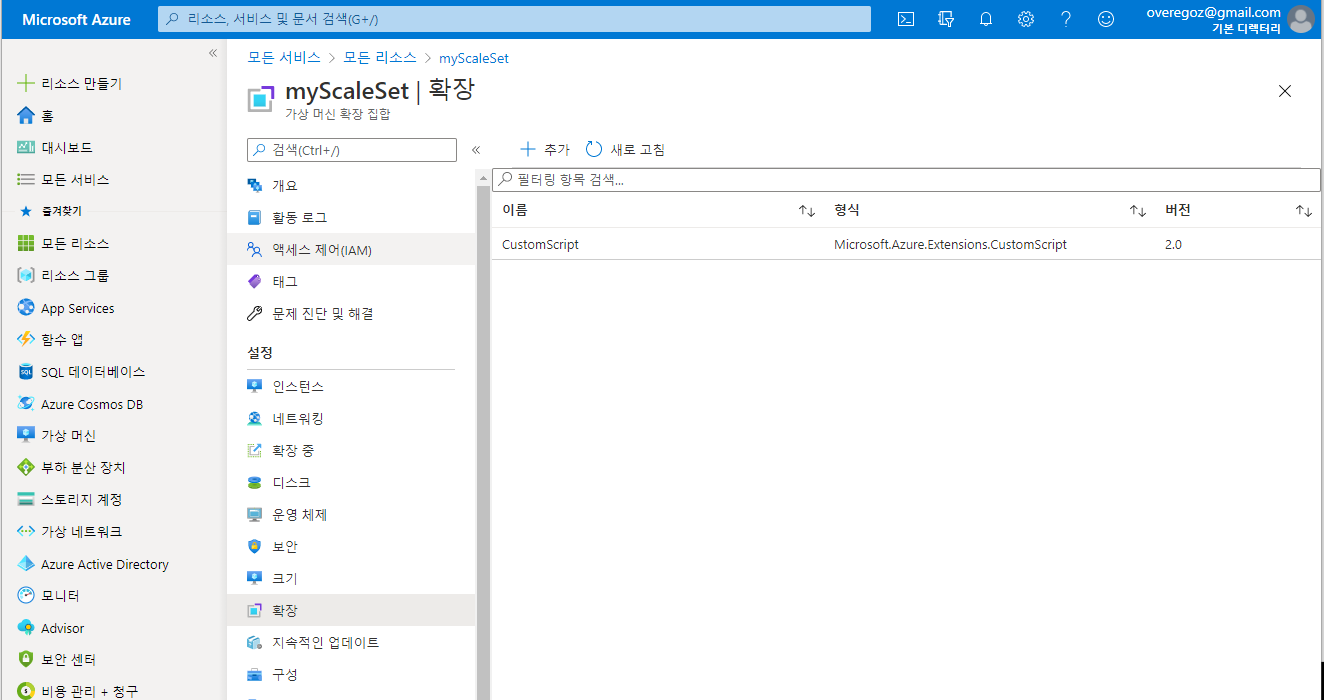
[예시 화면6] [Q2] 완료 후, ‘모든 리소스 > myScaleSet > 활동 로그’



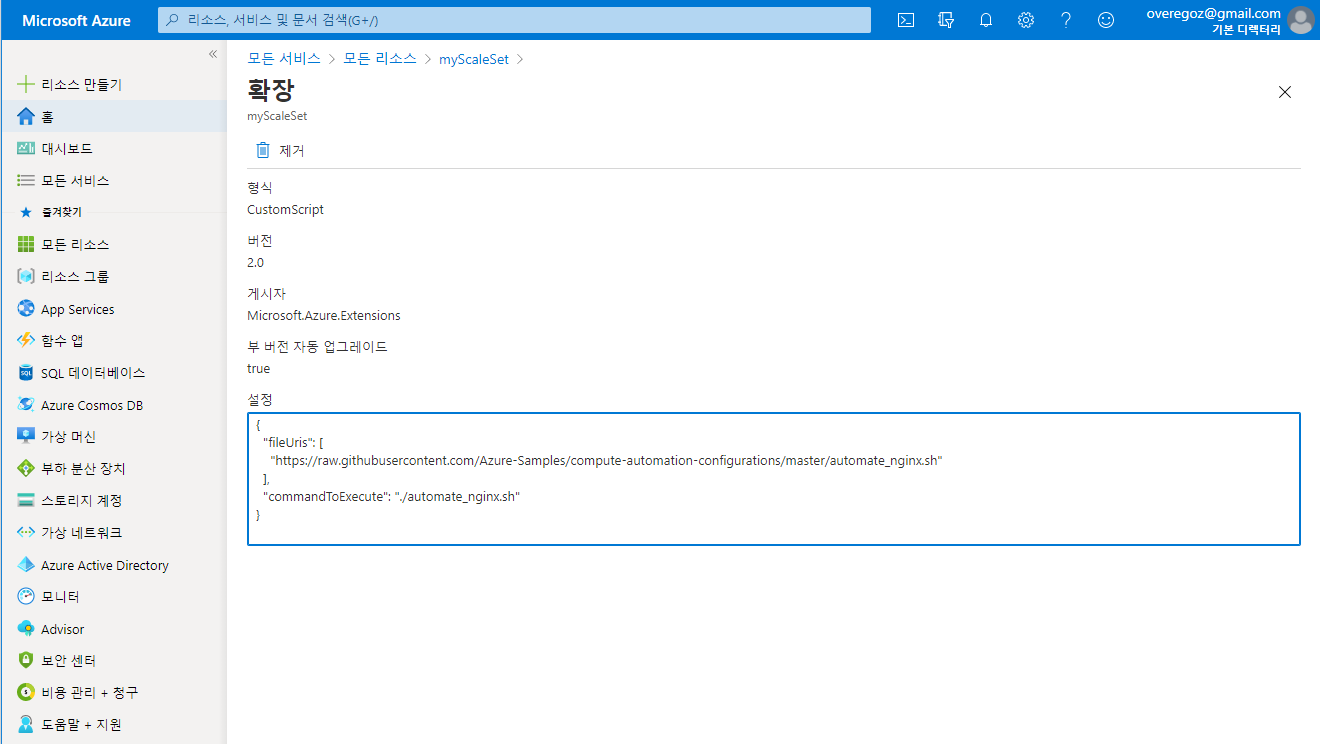
[예시 화면7] [Q2] 완료 후, VM이 2개로 줄어든 상황에서의 ‘모든 리소스 > myScaleSet > 인스턴스’



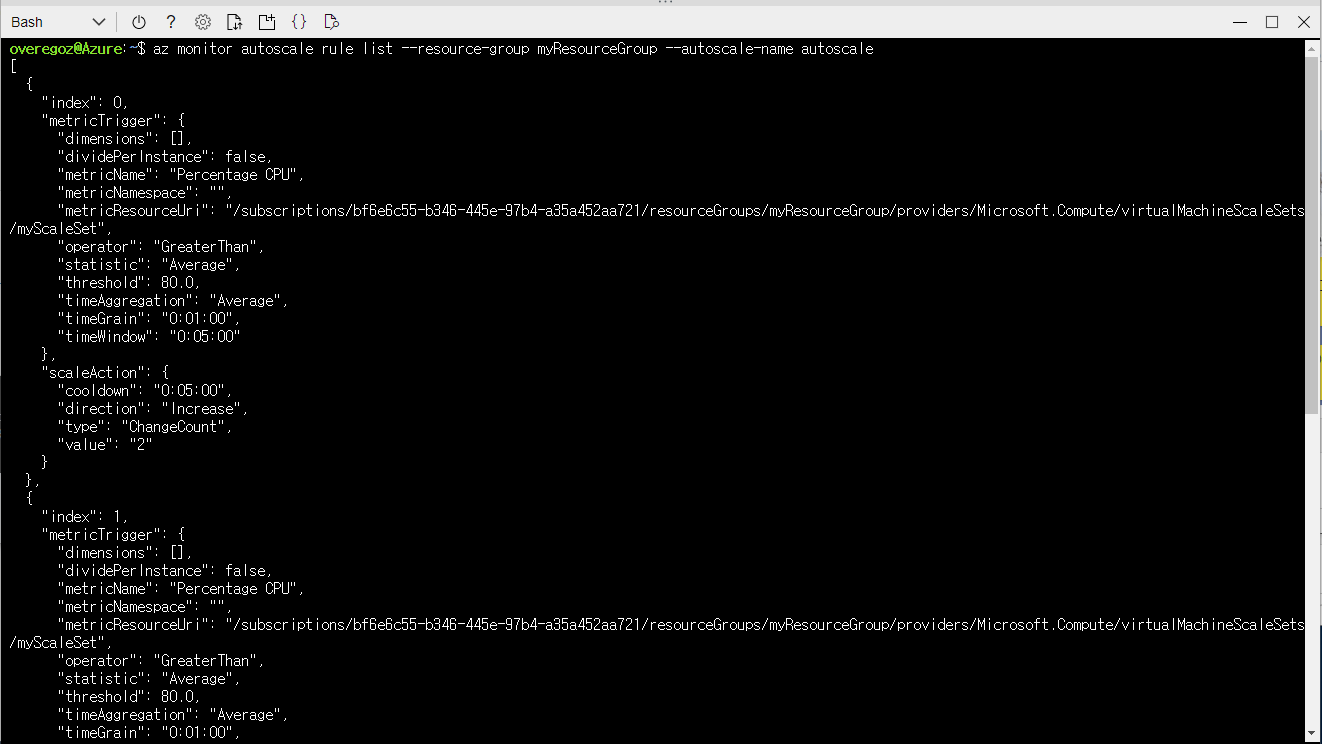
[예시 화면8] [Q2] 완료 후, ‘모든 리소스 > myScaleSet > 확장’



(CustomScript 클릭 시, 아래의 화면이 출력됨)



[예시 화면9] [Q2] 완료 후, 터미널에서 autoscale 규칙 출력 예시



**끝! 수고하셨습니다 ☺**