

제59회 전국기능경기대회 채점기준

1. 채점상의 유의사항	직 종 명	클라우드컴퓨팅
--------------	-------	---------

※ 다음 사항을 유의하여 채점하시오.

- 1) AWS의 지역은 ap-northeast-2을 사용합니다.
- 2) 웹페이지 접근은 크롬이나 파이어폭스를 이용합니다.
- 3) 웹페이지에서 언어에 따라 문구가 다르게 보일 수 있습니다.
- 4) shell에서의 명령어의 출력은 버전에 따라 조금 다를 수 있습니다.
- 5) 문제지와 채점지에 있는 < > 는 변수입니다. 해당 부분을 변경해 입력합니다.
- 6) 채점은 문항 순서대로 진행해야 합니다.
- 7) 삭제된 채점자료는 되돌릴 수 없으므로 유의하여 진행하며, 이의신청까지 완료 이후 선수가 생성한 클라우드 리소스를 삭제합니다.
- 8) 부분 점수가 있는 문항은 채점 항목에 부분 점수가 적혀져 있습니다.
- 9) 부분 점수가 따로 없는 문항은 모두 맞아야 점수로 인정됩니다.
- 10) 리소스의 정보를 읽어오는 채점항목은 기본적으로 스크립트 결과를 통해 채점을 진행하며, 만약 선수가 이의가 있다면 명령어를 직접 입력하여 확인해볼 수 있습니다.
- 11) [] 기호는 채점에 영향을 주지 않습니다.
- 12) 명령어 입력 Box 안의 명령줄은 한 줄 명령어입니다. 별도의 지시가 없으면 수정 없이 박스 안의 전체 내용을 복사하고 쉘에 붙여넣어 명령을 실행합니다.
- 13) (예상 출력)은 바로 이전 (명령어 입력)의 예상 출력을 의미합니다.

1) 주요항목별 배점			직 종 명		클라우드컴퓨팅			
과제 번호	일련 번호	주요항목	배점	채점방법		채점시기		비고
				독립	합의	경기 진행중	경기 종료후	
제1과제	1	API 구성	3.45		○		○	
	2	DynamoDB	0.3		○		○	
합 계			3.75					

2) 채점방법 및 기준

과제 번호	일련 번호	주요항목	일련 번호	세부항목(채점방법)	배점
1과제	1	API 구성	1	API Gateway 생성	0.4
			2	API POST - 정상 id	1.2
			3	API POST - 비정상 id	1.2
			4	API GET - id 조회	0.45
			5	api gateway integration 확인	0.2
	2	DynamoDB	1	DynamoDB Item 조회	0.3
	총점				3.75

3) 채점내용

순번	사전준비
0	<p>1) Bastion 서버에 SSH를 통해 접근합니다. (별도 명시가 없는 경우 모든 채점은 Bastion 서버에서 진행합니다.)</p> <p>2) Bastion 명령어 및 권한을 확인합니다. (awscli permission, curl, awscli region)</p> <p>3) marking 스크립트들을 /root/marking에 다운로드 합니다.</p> <p>4) /root/marking 경로에서 스크립트를 실행합니다. 실행 결과를 기반으로 채점을 진행하되 선수가 이의를 제기할 경우 수동으로 채점을 진행할 수 있도록 합니다.</p> <p>5) 채점을 진행하는 Bastion 서버의 셸을 초기 실행할 때 다음 명령어를 실행하여 환경 변수를 초기화합니다. (채점 스크립트로 진행 시 생략)</p>
	<p>6) 채점을 진행하기 전에 다음 명령어를 수행하여 채점 진행을 위한 사전 작업을 진행합니다. (채점 스크립트로 진행 시 생략)</p>
	<pre># set default region of aws cli aws configure set default.region ap-northeast-2</pre>

순번	채점 항목	
1-1	1-1-A	aws apigateway get-rest-apis --query items[].name
	1-1-A (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	["serverless-api-gw"]

1-2	1-2-A	aws dynamodb scan --table-name serverless-user-table --query Items
	1-2-A (예상 출력) <u>정확히 일치</u>	[]
	1-2-B	<p>apild=\$(aws apigateway get-rest-apis --query items[].id --output text)</p> <p>curl -sS -w "%n - status code: %{http_code} %n" -X POST</p> <p>"https://\$apild.execute-api.ap-northeast-2.amazonaws.com/v1/user?id=<심사위원 이 원하는 이름>&age=19&company=hrdkorea"</p> <p>※ 이름에 "admin" 이라는 단어가 포함되지 않도록 지정</p> <p>e.g) skillsadminuser -> X</p> <p>skillsuser -> O</p>
	1-2-B (예상 출력) <u>정확히 일치</u>	<pre>{ "msg": "Success insert data" }</pre> <p>- status code: 200</p>

1-3	1-3-A	<pre>apild=\$(aws apigateway get-rest-apis --query items[].id --output text) curl -sS -w "%{http_code} %n" -o /dev/null -X POST "https://\$apild.execute-api.ap-northeast-2.amazonaws.com/v1/user?id=<1-2-B에 서 입력한 이름>-admin&age=19&company=hrdkorea"</pre>
	1-3-A (예상 출력) 정확히 일치	- status code: 500
1-4	1-4-A	<pre>apild=\$(aws apigateway get-rest-apis --query items[].id --output text) curl -sS -w "%{http_code} %n" -X GET "https://\$apild.execute-api.ap-northeast-2.amazonaws.com/v1/user?id=<1- 2-B에 서 입력한 이름>"</pre>
	1-4-A (예상 출력) 정확히 일치	<pre>{ "id": "<1-2-B에서 입력한 이름>", <- 1-2-B에서 입력한 이름이 출력 "age": "19", "company": "hrdkorea" }</pre> <p>- status code: 200</p>
1-5	1-5-A	<pre>apild=\$(aws apigateway get-rest-apis --query "items[?name=='serverless-api-gw'].id" --output text) && resourceid=\$(aws apigateway get-resources --rest-api-id \$apild --query "items[?path=='/user'].id" --output text) && aws apigateway get-integration --rest-api-id \$apild --resource-id \$resourceid --http-method GET --query "uri" --output text grep -q 'dynamodb' && echo "DynamoDB integration found" echo "DynamoDB integration not found"</pre>
	1-5-A (예상 출력) 정확히 일치	DynamoDB integration found

순번	채점 항목	
2-1	2-1-A	aws dynamodb scan --table-name serverless-user-table --query Items
	2-1-A (예상 출력) <u>정확히 일치</u>	<pre>[{ "company": { "S": "hrdkorea" }, "id": { "S": "<1-2-B에서 입력한 이름>" <- 1-2-B에서 입력한 이름이 출력 }, "age": { "S": "19" } }]</pre> <p>위 내용만 존재해야 하며, 다른 item이 존재 시 해당 항목 0점 처리</p>