2025 전국기능경기대회 채점기준

1. 채점 상의 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 없음으로 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) 채점 내용의 $ 기호는 명령어에 포함되는 것이 아니라 쉘을 의미합니다.

2. 채점기준표

1) 주요항목별 배점

직 종 명	클라우드컴퓨팅

과제 번호	일련 번호	주요항목	배점	채점방법		채점시기		비고
				독립	합의	경기 진행중	경기 종료후	
제2과제	1	네트워크 구성	5		○		○	
	2	Bastion 서버	3		○		○	
	3	로그 생성 어플리케이션	12		○		○	
	4	Amazon Kinesis Data Stream	10		○		○	
	5	Kinesis Data Firehose	5		○		○	
	6	Glue	20		○		○	
	7	Athena	5		○		○	
합계			60					

2) 채점방법 및 기준

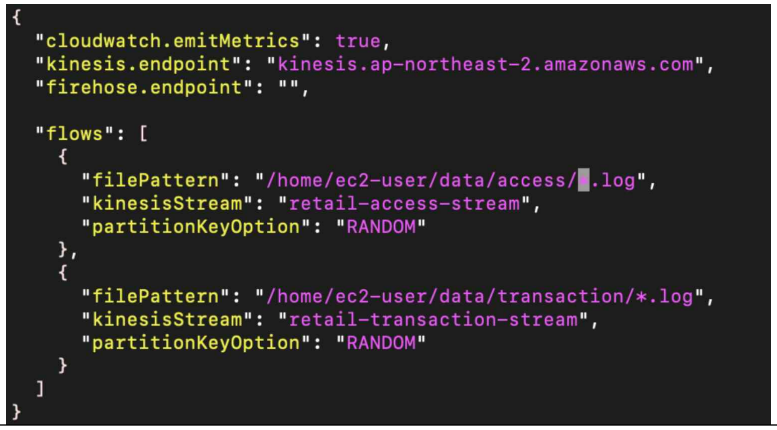
과제 번호	일련 번호	주요항목	일련 번호	세부항목(채점방법)	배점
2과제	1	네트워크 구성	1	VPC	1
			2	서브넷 구성	2
			3	라우팅 구성	2
	2	Bastion 서버	1	EC2 인스턴스 설정	1
			2	보안 및 권한 설정	2
	3	로그 생성 어플리케이션	1	EC2 인스턴스 구성	4
			2	바이너리 파일 설정	4
			3	Systemd 서비스 구성	4
	4	Amazon Kinesis Data Stream	1	Kinesis Data Stream 생성 및 설정	3
			2	Kinesis Agent 구성	4
			3	Agent 서비스 관리	3
	5	Kinesis Data Firehose	1	Firehose 전송 스트림 생성	2
			2	S3 버킷 연결	3
	6	Glue	1	데이터 카탈로그 설정	2
			2	커스텀 분류기 구성	3
			3	크롤러 구성	5
			4	ETL 작업 구현	10
	7	Athena	1	작업 그룹 생성	1
			2	Athena Query, 추가점수 5점 존재	4
	총점				60

3) 채점내용

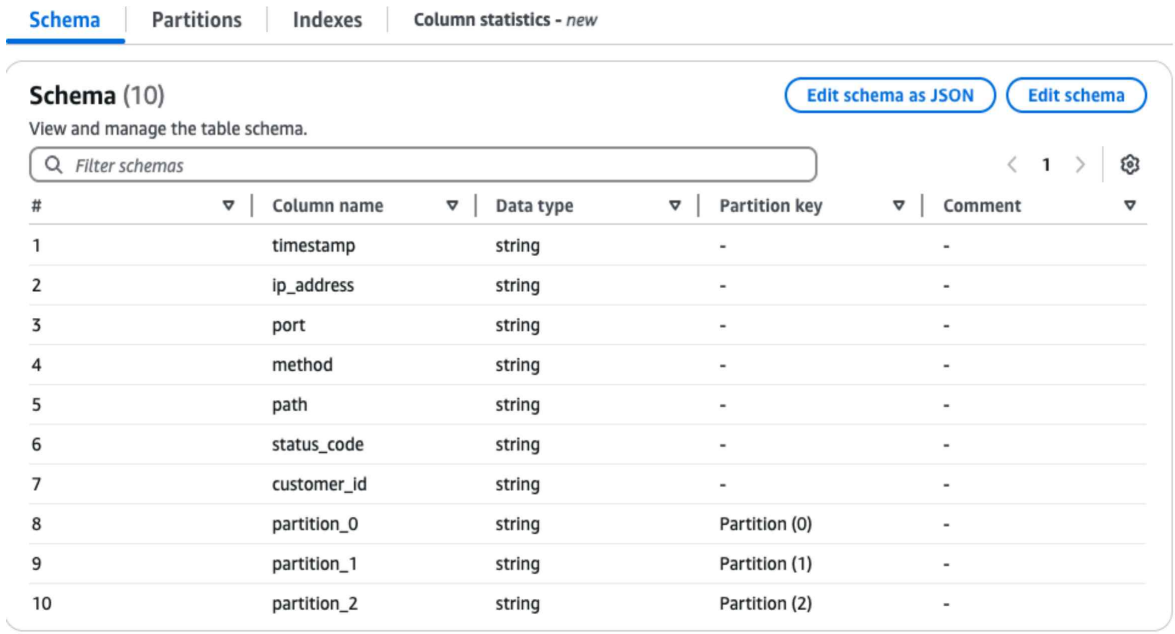
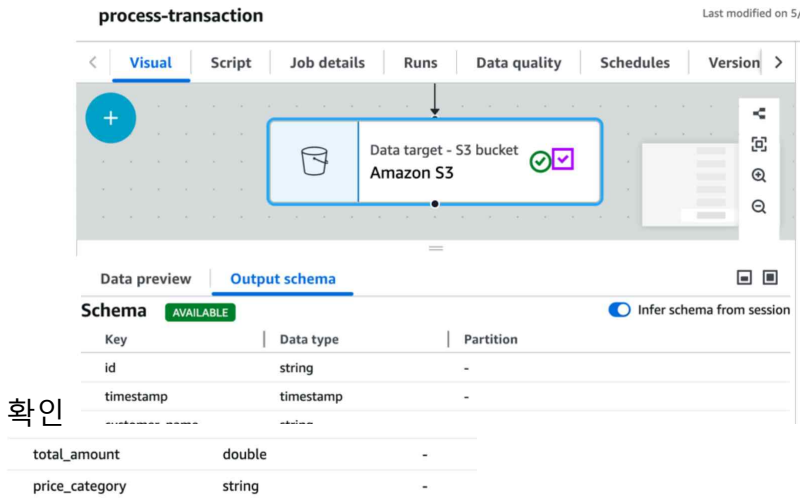
순번	사전준비
0	1) EC2 인스턴스에 SSH를 통해 접근합니다 (별도 명시가 없는 경우 모든 채점은 EC2 서버에서 진행합니다.) 2) EC2 명령어 및 권한을 확인합니다 (aws cli permission, awscli region, aws –version, aws sts get-caller-identity)

순번	채점 항목
1-1	- aws ec2 describe-vpcs --filters Name=tag:Name,Values=retail-vpc --query 'Vpcs[0].[CidrBlock,Tags[?Key==`Name`].Value]' --output text # 결과: 10.1.0.0/16 retail-vpc aws ec2 describe-internet-gateways --filters Name=tag:Name,Values=retail-igw --query 'InternetGateways[0].Tags[?Key==`Name`].Value' --output text # 결과retail-igw
1-2	- aws ec2 describe-subnets --filters Name=tag:Name,Values=retail-public-a --query 'Subnets[0].[CidrBlock,Tags[0].Value]' --output text # 결과10.1.2.0/24 retail-public-a - aws ec2 describe-subnets --filters Name=tag:Name,Values=retail-log-a --query 'Subnets[0].[CidrBlock,Tags[0].Value]' --output text # 결과10.1.1.0/24 retail-log-a
1-3	- aws ec2 describe-nat-gateways --filter Name=tag:Name,Values=retail-natgw-a --query 'NatGateways[*].[State,Tags[?Key==`Name`].Value]' --output text # 결과: available retail-natgw-a - aws ec2 describe-route-tables --filters Name=tag:Name,Values=retail-log-a-rt --query 'RouteTables[*].Routes[*].[DestinationCidrBlock,GatewayId,NatGatewayId]' --output text aws ec2 describe-route-tables --filters Name=tag:Name,Values=retail-public-a-rt --query 'RouteTables[*].Routes[*].[DestinationCidrBlock,GatewayId]' --output text # 결과 : 다음 형식과 같은지 확인 10.1.0.0/16 local None 0.0.0.0/0 None nat-054d5a0b5aba884bc 10.1.0.0/16 local 0.0.0.0/0 igw-0484794f0e3daa652

2-1	- aws ec2 describe-instances --filters "Name=tag:Name,Values=retail-bastion" --query 'Reservations[*].Instances[*].[InstanceType,ImageId,SubnetId,Tags[?Key==`Name`].Value]' --output text # 결과t3.large ami-xxxxx subnet-xxxxx retail-bastion - aws ec2 describe-instances --filters Name=tag:Name,Values=retail-bastion --query 'Reservations[0].Instances[0].PublicIpAddress' --output text # 결과 : EIP 형태
2-2	- aws ec2 describe-security-groups --filters Name=group-name,Values=retail-bastion-sg --query 'SecurityGroups[0].IpPermissions[?ToPort==`22`].[IpProtocol,FromPort,ToPort]' --output text # 결과: tcp 22 22 - aws iam get-role --role-name retail-app-role --query 'Role.RoleName' --output text # 결과: retail-app-role - aws iam list-attached-role-policies --role-name retail-app-role --query 'AttachedPolicies[?PolicyName==`AdministratorAccess`].PolicyName' --output text # 결과: AdministratorAccess
3-1	- aws ec2 describe-instances --filters "Name=tag:Name,Values=retail-log" --query 'Reservations[*].Instances[*].[InstanceType,SubnetId,Tags[?Key==`Name`].Value]' --output text # 결과: t3.large subnet-xxxxx retail-log - aws iam get-role --role-name retail-app-role --query 'Role.RoleName' --output text # 결과: retail-app-role - aws ec2 describe-security-groups --filters Name=group-name,Values=retail-log-sg --query 'SecurityGroups[0].IpPermissions[*]' --output text # 결과: 보안그룹의 인바운드 규칙 확인 (Bastion 서버의 Ip 혹은retail-bastion-sg만 허용되어야 함)
3-2	**retail-log 서버에서 채점 실시** - ls -l /home/ec2-user/access-log-generator /home/ec2-user/transaction-log-generator # 결과가 다음 형식과 같은지 확인 

3 - 3	**retail-log 서버에서 채점 실시** - systemctl status access-generator.service # 결과: active (running) - systemctl status transaction-generator.service # 결과: active (running) - systemctl is-enabled access-generator.service # 결과: enabled - systemctl is-enabled transaction-generator.service # 결과: enabled
4 - 1	- aws kinesis describe-stream --stream-name retail-access-stream --query 'StreamDescription.[StreamName,RetentionPeriodHours,Shards[0].ShardId]' --output text # 결과: retail-access-stream 24 shardId-000000000000 - aws kinesis describe-stream --stream-name retail-transaction-stream --query 'StreamDescription.[StreamName,RetentionPeriodHours,Shards[0].ShardId]' --output text # 결과: retail-transaction-stream 24 shardId-000000000000
4 - 2	**retail-log 서버에서 채점 실시** - sudo vi /etc/aws-kinesis/agent.json # 결과가 다음 형식과 같은지 확인 
4 - 3	**retail-log 서버에서** - systemctl status aws-kinesis-agent # 결과: active (running) 확인 - systemctl is-enabled aws-kinesis-agent # 결과: enabled 확인

5-1	- aws firehose describe-delivery-stream --delivery-stream-name retail-access-delivery --query 'DeliveryStreamDescription.[DeliveryStreamName,DeliveryStreamStatus,Source.KinesisStreamSourceDescription.KinesisStreamARN,DestinationDescriptions[0].S3DestinationDescription.[BufferingHints.SizeInMBs,BufferingHints.IntervalInSeconds,BucketARN,Prefix]]' --output text # 결과: retail-access-delivery ACTIVE arn:aws:kinesis:[region]:[account]:stream/retail-access-stream 1 60 arn:aws:s3:::retail-data-lake-[계정번호 앞 6자리] raw/access/ - aws firehose describe-delivery-stream --delivery-stream-name retail-transaction-delivery --query 'DeliveryStreamDescription.[DeliveryStreamName,DeliveryStreamStatus,Source.KinesisStreamSourceDescription.KinesisStreamARN,DestinationDescriptions[0].S3DestinationDescription.[BufferingHints.SizeInMBs,BufferingHints.IntervalInSeconds,BucketARN,Prefix]]' --output text # 결과: retail-transaction-delivery ACTIVE arn:aws:kinesis:[region]:[account]:stream/retail-transaction-stream 1 60 arn:aws:s3:::retail-data-lake-[계정번호 앞6자리] raw/transaction/
5-2	- aws s3api head-bucket --bucket retail-data-lake-$(aws sts get-caller-identity --query 'Account' --output text | cut -c1-6) # 결과과 다음 형식과 같은지 확인 
6-1	- aws glue get-database --name retail_analytics_db --query 'Database.[Name,Description]' --output text # 결과: retail_analytics_db "Online Store Data Analysis Database"
6-2	- aws glue get-classifier --name access-log-classifier --query 'Classifier.GrokClassifier.[Name,Classification,GrokPattern]' --output text # 결과: access-log-classifier grokLog %{TIMESTAMP_ISO8601:timestamp} %{IP:ip_address}:%{NUMBER:port} %{WORD:method} /%{DATA:path} %{NUMBER:status_code} %{USERNAME:customer_id}
6-3	- aws glue get-crawler --name retail-transaction-crawler --query 'Crawler.[Name,DatabaseName,Targets.S3Targets[0].Path]' --output text # 결과: retail-transaction-crawler retail_analytics_db s3://retail-data-lake-XXXXXX/raw/transaction/ - aws glue get-crawler --name retail-access-crawler --query 'Crawler.[Name,DatabaseName,Targets.S3Targets[0].Path]' --output text # 결과: retail-access-crawler retail_analytics_db s3://retail-data-lake-XXXXXX/raw/access/

<table>
<tr><td></td><td>- Glue Data Catalog tables 에서 raw_access테이블이 아래와 같은 Schema를 구성했는지 확인
 
</td></tr>
<tr><td>6
-
4</td><td>- AWS Glue 콘솔에서 visual ETL -> process-transaction 에서timestamp 필드의 데이터 타입이 'timestamp'로 올바르게 변경되었는지,total_amount 필드가 새로 생성되었는지, price_category 필드가 새로 생성되었는지
확인
 
</td></tr>
<tr><td>7
-
1</td><td>- aws athena get-work-group --work-group retail-analytics-workgroup --query 'WorkGroup.[Name,Configuration.ResultConfiguration.OutputLocation]' --output text
결과: retail-analytics-workgroup s3://retail-data-lake-XXXXXX/athena-result/</td></tr>
</table>

7-2	- **AWS Athena 콘솔에서**retail-analytics-workgroup 에서 다음과 같은 쿼리문을 작성하였는지 확인 SELECT price_category, COUNT(*) as order_count, SUM(total_amount) as total_sales FROM processed_transaction GROUP BY price_category ORDER BY total_sales DESC; - 쿼리 결과가 다음 형식과 같은지 확인. (해당 데이터셋에서 3번째 행의 price_category가 비어있는 것을 확인 가능) - 3번째 행의 price_category에는 'Invalid', 'Unknown', 'NULL' 등과 같은 적절한 레이블을 지정하여 누락된 데이터를 명시적으로 표시할 시 **5점 추가점수**