

2025년도 전국기능경기대회 채점기준

1. 채점상의 유의사항	직 종 명	클라우드컴퓨팅
<p>※ 다음 사항을 유의하여 채점하시오.</p> <ol style="list-style-type: none"> 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) 모든 채점 사항은 ws25-ec2-bastion에서 SSH 접속 후 진행합니다. 12) 별도의 지시가 없으면 수정 없이 박스 안의 명령어 입력 상자 안의 전체 내용을 복사하고 셸에 붙여넣어 명령을 실행합니다. 13) (예상 출력)은 바로 이전 (명령어 입력)의 예상 출력을 의미합니다. 14) 채점 시에는 별도로 제공한 채점 스크립트(mark.sh)를 실행하여 채점할 수 있습니다. 다만, 선수가 직접 입력을 원할 경우 채점기준표에 명시된 명령어 그대로 입력하여 채점할 수 있습니다. 채점 스크립트는 ec2-user의 root 경로에 지정하도록 합니다. 		

2. 채점기준표

1) 주요항목별 배점			직 종 명			클라우드컴퓨팅		
과제 번호	일련 번호	주요항목	배점	채점방법		채점시기		비고
				독립	합의	경기 진행중	경기 종료후	
제1과제	1	네트워크 구성	2.25		○		○	
	2	Bastion	1.0		○		○	
	3	Secret Management	1.0		○		○	
	4	RDBMS	2.55		○		○	
	5	ECR	1.0		○		○	
	6	ECS	4.5		○		○	
	7	Logging	1.5		○		○	
	8	Load Balancer	3.5		○		○	
	9	Application	4.2		○		○	
	10	Monitoring	3.2		○		○	
	11	Continuous Delivery	5.3		○		○	
합 계			30					

2) 채점방법 및 기준

과제 번호	일련 번호	주요항목	일련 번호	세부항목(채점방법)	배점
1과제	1	Networking	1	VPC	0.25
			2	Subnets	0.25
			3	VPC Peering	0.25
			4	Routing Table	0.5
			5	VPC Flow Log	0.5
			6	ECR Endpoints	0.5
	2	Bastion	1	EC2 Instance Config	0.3
			2	IAM Role	0.3
			3	Security Group	0.4
	3	Secret Management	1	Secret Manager Config	0.35
			2	Secret Manager Encryption	0.65
	4	RDBMS	1	Database Config	0.5
			2	Database Monitoring	0.45
			3	Database Options	0.8
			4	Database Security	0.8
	5	ECR	1	Repository Tag Mutability	0.15
			2	Repository Encryption	0.25
			3	Repository Images	0.35
			4	Repository Scanning	0.25
	6	ECS	1	ECS Cluster	0.3
			2	ECS Task Definition - Green	1.0
			3	ECS Task Definition - Red	1.0
			4	ECS Service - Green	1.1
			5	ECS Service - Red	1.1
	7	Logging	1	FirelensConfiguration	0.5
			2	Exclude Health path	1.0
	8	Load Balancer	1	Load Balancer Config	1.0
			2	Load Balancer Target	1.5
			3	Load Balancer Listener Rule	1.0
	9	Application	1	Green App	2.1
			2	Red App	2.1
	10	Monitoring	1	Container Insights	0.4
			2	Dashboards	1.0
			3	Alarms	0.9
			4	Requesting Data	0.9
	11	Continuous Delivery	1	CodeDeploy - Green	0.4
			2	CodeDeploy - Red	0.4
			3	CodePipeline - Green	0.5
			4	CodePipeline - Red	0.5
			5	File Validation	0.3
			6	Deployment Check - Green	1.6
			7	Deployment Check - Red	1.6

3) 채점내용

순번	사전준비	
0	1) ws25-ec2-bastion 서버에 SSH를 통해 접근합니다. 2) 제공된 mark.sh 채점 스크립트를 실행합니다. 3) “[!] 사전 준비가 완료되었습니다. 채점을 시작합니다.” 이라는 문구 출력 시 1번 항목이 자동으로 진행됩니다. 4) 채점이 자동으로 시작됩니다.	
	수동 채점 시 aws sts get-caller-identity 명령으로 자격증명이 적절한지 확인한 후 적절하다면 1번 항목부터 채점합니다.	
순번	채점항목	
1-1	1-1-A (명령어 입력)	<pre>aws ec2 describe-vpcs --filter Name=tag:Name,Values=ws25-hub-vpc --query "Vpcs[0].CidrBlock" \ ; aws ec2 describe-vpcs --filter Name=tag:Name,Values=ws25-app-vpc --query "Vpcs[0].CidrBlock"</pre>
	1-1-A (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	<pre>"172.28.0.0/16" "10.200.0.0/16"</pre>

순번	채점항목	
1-2	1-2-A (명령어 입력)	<pre>aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-hub-pub-a --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-hub-pub-c --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-app-pub-a --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-app-pub-b --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-app-pub-c --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-app-pri-a --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-app-pri-b --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-app-pri-c --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-app-db-a --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text ; aws ec2 describe-subnets --filter Name=tag:Name,Values=ws25-app-db-c --query "Subnets[0].[CidrBlock, AvailabilityZone]" --output text</pre>
	1-2-A (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	<pre>172.28.0.0/20 ap-northeast-2a 172.28.16.0/20 ap-northeast-2c 10.200.10.0/24 ap-northeast-2a 10.200.11.0/24 ap-northeast-2b 10.200.12.0/24 ap-northeast-2c 10.200.20.0/24 ap-northeast-2a 10.200.21.0/24 ap-northeast-2b 10.200.22.0/24 ap-northeast-2c 10.200.30.0/24 ap-northeast-2a 10.200.31.0/24 ap-northeast-2c</pre>

순번	채점항목	
1-3	1-3-A (명령어 입력)	aws ec2 describe-vpc-peering-connections --filters Name=tag:Name,Values=ws25-peering --query "VpcPeeringConnections[0].{Requester:RequesterVpcInfo.VpcId, Acceptor:AcceptorVpcInfo.VpcId}" --output text
	1-3-A (예상 출력)	"vpc-" <u>로 시작하는 문구가 정상적으로 2개 출력 되는지 확인</u>
1-4	1-4-A (명령어 입력)	aws ec2 describe-route-tables --filters "Name=tag:Name,Values=ws25-hub-pub-rt" --query "RouteTables[].{IGW: Routes[?GatewayId != null && starts_with(GatewayId, 'igw')].GatewayId, Subnets: Associations[?SubnetId != null].SubnetId, Peering: Routes[?VpcPeeringConnectionId != null].VpcPeeringConnectionId}" --output text echo -e "\n" aws ec2 describe-route-tables --filters "Name=tag:Name,Values=ws25-app-pub-rt" --query "RouteTables[].{IGW: Routes[?GatewayId != null && starts_with(GatewayId, 'igw')].GatewayId, Subnets: Associations[?SubnetId != null].SubnetId}" --output text echo -e "\n" aws ec2 describe-route-tables --filters "Name=tag:Name,Values=ws25-app-pri-rt-a" --query "RouteTables[].{NATGW: Routes[?NatGatewayId != null].NatGatewayId, Subnets: Associations[?SubnetId != null].SubnetId, Peering: Routes[?VpcPeeringConnectionId != null].VpcPeeringConnectionId}" --output text echo -e "\n" aws ec2 describe-route-tables --filters "Name=tag:Name,Values=ws25-app-pri-rt-b" --query "RouteTables[].{NATGW: Routes[?NatGatewayId != null].NatGatewayId, Subnets: Associations[?SubnetId != null].SubnetId, Peering: Routes[?VpcPeeringConnectionId != null].VpcPeeringConnectionId}" --output text echo -e "\n" aws ec2 describe-route-tables --filters "Name=tag:Name,Values=ws25-app-pri-rt-c" --query "RouteTables[].{NATGW: Routes[?NatGatewayId != null].NatGatewayId, Subnets: Associations[?SubnetId != null].SubnetId, Peering: Routes[?VpcPeeringConnectionId != null].VpcPeeringConnectionId}" --output text echo -e "\n" aws ec2 describe-route-tables --filters "Name=tag:Name,Values=ws25-app-db-rt-a" --query "RouteTables[].{Subnets: Associations[?SubnetId != null].SubnetId, Peering: Routes[?VpcPeeringConnectionId != null].VpcPeeringConnectionId}" --output text echo -e "\n" aws ec2 describe-route-tables --filters "Name=tag:Name,Values=ws25-app-db-rt-c" --query "RouteTables[].{Subnets: Associations[?SubnetId != null].SubnetId, Peering: Routes[?VpcPeeringConnectionId != null].VpcPeeringConnectionId}" --output text echo -e "\n"

순번	채점항목	
1-4	1-4-A (예상 출력) <u>순서 중요</u>	<p>“IGW igw-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“PEERING pcx-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“IGW igw-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“NATGW nat-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“PEERING pcx-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“NATGW nat-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“PEERING pcx-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“NATGW nat-” 로 시작하는 문구가 위 NATGW와 같은 ID로 정상적으로 출력되는지 확인</p> <p>“PEERING pcx-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“PEERING pcx-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“PEERING pcx-” 로 시작하는 문구가 정상적으로 출력되는지 확인</p> <p>“SUBNETS subnet-” 으로 시작하는 문구가 정상적으로 출력되는지 확인</p>

순번	채점항목	
1-5	1-5-A (명령어 입력)	<pre> VPCHUBID=\$(aws ec2 describe-vpcs --filters "Name=tag:Name,Values=ws25-hub-vpc" --query "Vpcs[*].VpcId" --output text) aws ec2 describe-flow-logs --filter "Name=resource-id,Values=\$VPCHUBID" --query "FlowLogs[*].FlowLogId" --output text VPCAPPID=\$(aws ec2 describe-vpcs --filters "Name=tag:Name,Values=ws25-app-vpc" --query "Vpcs[*].VpcId" --output text) aws ec2 describe-flow-logs --filter "Name=resource-id,Values=\$VPCAPPID" --query "FlowLogs[*].FlowLogId" --output text </pre>
	1-5-A (예상 출력)	<p><u>“fi-” 로 시작하는 문구가 정상적으로 출력 되는지 확인</u></p> <p><u>“fi-” 로 시작하는 문구가 위와 다른 값으로 정상 출력 되는지 확인</u></p>
	1-5-B (명령어 입력) 1-5-A 오답 시 채점 X	<pre> for group in /ws25/flow/hub /ws25/flow/app; do count=\$(aws logs describe-log-streams --log-group-name \$group --query 'logStreams' --output json jq length); echo "\$group : \$count"; done </pre>
	1-5-B (예상 출력) 순서 무관	<p><u>/ws25/flow/hub : 1 이상의 값이 출력 되는지 확인</u></p> <p><u>/ws25/flow/app : 1 이상의 값이 출력 되는지 확인</u></p>

순번	채점항목	
1-6	1-6-A (명령어 입력)	aws ec2 describe-vpc-endpoints --query "VpcEndpoints[].ServiceName"
	1-6-A (예상 출력) 순서 무관	["com.amazonaws.ap-northeast-2.ecr.dkr", "com.amazonaws.ap-northeast-2.ecr.api", "com.amazonaws.ap-northeast-2.s3"]
	1-6-B (명령어 입력) 1-6-A 오답 시 채점 X	for service in com.amazonaws. \$(aws configure get region).ecr.dkr com.amazonaws. \$(aws configure get region).ecr.api; do eni=\$(aws ec2 describe-vpc-endpoints --filters Name=service-name,Values=\$service --query 'VpcEndpoints[0].NetworkInterfaceIds[0]' --output text); for sg in \$(aws ec2 describe-network-interfaces --network-interface-ids \$eni --query 'NetworkInterfaces[0].Groups[].GroupId' --output text); do echo "\$service - \$sg"; aws ec2 describe-security-groups --group-ids \$sg --query 'SecurityGroups[0].IpPermissions[?FromPort==`443` && ToPort==`443`].IpRanges[].CidrIp' --output text; done; done
	1-6-B (예상 출력)	com.amazonaws.ap-northeast-2.ecr.dkr - “sg-” <u>로 시작하는 문구가 정상적으로 출력 되는지 확인</u> 10.200.0.0/16 (또는 ws25-app-pri 서브넷의 CIDR 범위) com.amazonaws.ap-northeast-2.ecr.api - “sg-” <u>로 시작하는 위와 같은 문구가 정 상적으로 출력 되는지 확인</u> 10.200.0.0/16 (또는 ws25-app-pri 서브넷의 CIDR 범위)

순번	채점항목	
2-1	2-1-A (명령어 입력)	aws ec2 describe-images --image-ids \$(aws ec2 describe-instances --filters Name=tag:Name,Values=ws25-ec2-bastion Name=instance-state-name,Values=running --query "Reservations[].Instances[].ImageId" --output text) --query "Images[].Name" --output text; aws ec2 describe-instances --filters Name=tag:Name,Values=ws25-ec2-bastion Name=instance-state-name,Values=running --query "Reservations[].Instances[].InstanceType" --output text; aws ec2 describe-subnets --subnet-ids \$(aws ec2 describe-instances --filters Name=tag:Name,Values=ws25-ec2-bastion Name=instance-state-name,Values=running --query "Reservations[].Instances[].SubnetId" --output text) --query "Subnets[].Tags[?Key=='Name'].Value [0]" --output text; aws ec2 describe-addresses --filters Name=instance-id,Values=\$(aws ec2 describe-instances --filters Name=tag:Name,Values=ws25-ec2-bastion Name=instance-state-name,Values=running --query "Reservations[].Instances[].InstanceId" --output text) --query "Addresses[].PublicIp"
	2-1-A (예상 출력) <u>순서 일치</u> 오류 발생 시 채점 X	“al2023-“ <u>로 시작하는 문구가 정상적으로 출력되는지 확인</u> t3.small ws25-hub-pub-c [<u>IP가 정상적으로 출력되었는지 확인</u>]
2-2	2-2-A (명령어 입력)	aws iam list-attached-role-policies --role-name \$(aws ec2 describe-instances --filters Name=tag:Name,Values=ws25-ec2-bastion --query "Reservations[].Instances[].IamInstanceProfile.Arn" --output text awk -F'instance-profile/' '{print \$2}') --query "AttachedPolicies[?PolicyName=='AdministratorAccess'].PolicyName [0]" --output text
	2-2-A (예상 출력) <u>정확히 일치</u>	AdministratorAccess
2-3	2-3-A (명령어 입력)	aws ec2 describe-security-groups --group-ids \$(aws ec2 describe-instances --filters Name=tag:Name,Values=ws25-ec2-bastion --query "Reservations[].Instances[].SecurityGroups[].GroupId" --output text) --query "SecurityGroups[].IpPermissions" --output json jq -c '.[[]]'
	2-3-A (예상 출력) <u>정확히 일치</u>	{"IpProtocol":"tcp","FromPort":10100,"ToPort":10100,"UserIdGroupPairs":[],"IpRanges":[{"CidrIp":"0.0.0.0/0"}],"Ipv6Ranges":[],"PrefixListIds":[]}

순번	채점항목	
3-1	3-1-A (명령어 입력)	aws secretsmanager describe-secret --secret-id ws25/secret/key --query "Name" --output text; aws secretsmanager get-secret-value --secret-id ws25/secret/key --query "SecretString" --output text jq -r 'keys[]'
	3-1-A (예상 출력) 키의 순서는 바뀌어도 무방	ws25/secret/key DB_PASSWD DB_URL DB_USER
3-2	3-2-A (명령어 입력)	aws secretsmanager describe-secret --secret-id ws25/secret/key --query "KmsKeyId" --output text
	3-2-A (예상 출력)	arn:aws:kms:ap-northeast-2:122610509788:key/5561bfd6-f685-49c1-8ca3-09cc394ad286 (키의 arn이 ws25-kms 키의 arn과 동일한지 확인)
4-1	4-1-A (명령어 입력)	aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].Engine" --output text; aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].EngineVersion" --output text; aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "length(DBClusters[0].DBClusterMembers[?IsClusterWriter==\`false\`])" --output text; aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].Port" --output text; aws rds describe-db-instances --filters Name=db-cluster-id,Values=ws25-rdb-cluster --query "DBInstances[0].DBInstanceClass" --output text; aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].Status" --output text
	4-1-A (예상 출력) 순서 중요 정확히 일치	aurora-mysql 8.0.mysql_aurora.3.08.2 1 10101 db.t4g.medium available

순번	채점항목	
4-2	4-2-A (명령어 입력)	aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].PerformanceInsightsEnabled" --output text; aws rds describe-db-instances --filters Name=db-cluster-id,Values=ws25-rdb-cluster --query "DBInstances[0].EnabledCloudwatchLogsExports" --output json
	4-2-A (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	True ["audit", "error", "general", "instance"]
4-3	4-3-A (명령어 입력)	aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].DatabaseName" --output text; aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].BackupRetentionPeriod" --output text; aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].BacktrackWindow" --output text; aws rds describe-db-subnet-groups --db-subnet-group-name \$(aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].DBSubnetGroup" --output text) --query "DBSubnetGroups[0].Subnets[].SubnetIdentifier" --output text
	4-3-A (예상 출력) <u>순서 중요</u>	day1 34 10800 "subnet-" <u>로 시작하는 문구가 2개이며, ws25-app-db 서브넷의 ID와 일치하는지</u> <u>확인</u>

순번	채점항목	
4-4	4-4-A (명령어 입력)	aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].KmsKeyId" --output text; aws kms get-key-rotation-status --key-id \$(aws rds describe-db-clusters --db-cluster-identifier ws25-rdb-cluster --query "DBClusters[0].KmsKeyId" --output text) --query "KeyRotationEnabled" --output text
	4-4-A (예상 출력) 순서 중요	arn:aws:kms:ap-northeast-2:122610509788:key/5451bfd6-f685-49c1-8ca3-09cc394ad28f (키의 arn이 ws25-kms 키의 arn과 동일한지 확인) True
5-1	5-1-A (명령어 입력)	aws ecr describe-repositories --repository-names green --query "repositories[0].imageTagMutability" --output text; aws ecr describe-repositories --repository-names red --query "repositories[0].imageTagMutability" --output text
	5-1-A (예상 출력) 정확히 일치 순서 중요	IMMUTABLE IMMUTABLE
5-2	5-2-A (명령어 입력)	aws ecr describe-repositories --repository-names red --query "repositories[0].encryptionConfiguration" --output json; aws ecr describe-repositories --repository-names green --query "repositories[0].encryptionConfiguration" --output json
	5-2-A (예상 출력) 순서 중요	{ "encryptionType": "KMS", "kmsKey": "arn:aws:kms:ap-northeast-2:122610509788:key/5451bfd6-f685-49c1-8ca3-09cc394ad28f" (키의 arn이 ws25-kms 키의 arn과 동일한지 확인) } { "encryptionType": "AES256" }

순번	채점항목	
5-3	5-3-A (명령어 입력)	aws ecr list-images --repository-name green --query "imageIds[].imageTag" --output text
	5-3-A (예상 출력) 순서 무관	v1.0.0 v1.0.1
	5-3-B (명령어 입력)	aws ecr list-images --repository-name red --query "imageIds[].imageTag" --output text
	5-3-B (예상 출력) 순서 무관	v1.0.0 v1.0.1
5-4	5-4-A (명령어 입력)	aws ecr describe-repositories --repository-names green --query "repositories[0].imageScanningConfiguration.scanOnPush" --output text; aws ecr describe-repositories --repository-names red --query "repositories[0].imageScanningConfiguration.scanOnPush" --output text; aws ecr describe-image-scan-findings --repository-name green --image-id imageTag=v1.0.0 --query "imageScanFindings.findingSeverityCounts" --output json; aws ecr describe-image-scan-findings --repository-name green --image-id imageTag=v1.0.1 --query "imageScanFindings.findingSeverityCounts" --output json; aws ecr describe-image-scan-findings --repository-name red --image-id imageTag=v1.0.0 --query "imageScanFindings.findingSeverityCounts" --output json; aws ecr describe-image-scan-findings --repository-name red --image-id imageTag=v1.0.1 --query "imageScanFindings.findingSeverityCounts" --output json
	5-4-A (예상 출력)	True True <u>이 외의 다른 문구 출력 시 출력되는 문구가 LOW 이상의 취약성인 경우 오답 처리</u>

순번	채점항목	
6-1	6-1-A (명령어 입력)	aws ecs describe-clusters --clusters ws25-ecs-cluster --include CONFIGURATIONS --region ap-northeast-2 --output json jq -r '.clusters[0].configuration.managedStorageConfiguration.kmsKeyId'
	6-1-A (예상 출력)	arn:aws:kms:ap-northeast-2:122610509788:key/5532f52f-059c-4f19-ada2-befd1fd9d273 <u>(키의 arn이 ws25-kms 키의 arn과 동일한지 확인)</u>
6-2	6-2-A (명령어 입력)	bash -c 'for td in \$(aws ecs list-task-definitions --family-prefix ws25-ecs-green-taskdef --region ap-northeast-2 --query "taskDefinitionArns[]" --output text); do aws ecs describe-task-definition --task-definition \$td --region ap-northeast-2 --query "taskDefinition.containerDefinitions[].image" --output text grep -qw green:v1.0.0 && { echo True; exit 0; }; done; echo False'
	6-2-A (예상 출력) <u>정확히 일치</u> 0.2점	True
	6-2-B (명령어 입력)	aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-green --region ap-northeast-2 --output text --query "services[0].taskDefinition" xargs -l{} bash -c 'td="{ }"; name=\$(basename "\${td%:*}"); echo "\$name"; aws ecs describe-task-definition --task-definition "\$td" --region ap-northeast-2 --output json jq -r "(.taskDefinition.cpu tonumber/1024 tostring) + \" vCPU, \" + (.taskDefinition.memory tonumber/1024 tostring) + \"GB\""'
	6-2-B (예상 출력) <u>정확히 일치</u> 0.4점	ws25-ecs-green-taskdef 1 vCPU, 1GB
	6-2-C (명령어 입력)	aws ecs describe-task-definition --task-definition \$(aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-green --query 'services[0].taskDefinition' --output text --region ap-northeast-2) --output json --region ap-northeast-2 jq -r '"ENV", (.taskDefinition.containerDefinitions[] select(.firelensConfiguration not) .environment[]?.name), "", "Secrets", (.taskDefinition.containerDefinitions[] select(.firelensConfiguration not) .secrets[]?.name)'
	6-2-C (예상 출력) <u>순서 무관</u>	ENV Secrets DB_USER DB_PASSWD DB_URL

순번	채점항목	
6-3	6-3-A (명령어 입력)	<pre>bash -c 'for td in \$(aws ecs list-task-definitions --family-prefix ws25-ecs-red-taskdef --region ap-northeast-2 --query "taskDefinitionArns[]" --output text); do aws ecs describe-task-definition --task-definition \$td --region ap-northeast-2 --query "taskDefinition.containerDefinitions[].image" --output text grep -qw red:v1.0.0 && { echo True; exit 0; }; done; echo False'</pre>
	6-3-A (예상 출력) 정확히 일치 0.2점	True
	6-3-B (명령어 입력)	<pre>aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-red --region ap-northeast-2 --output text --query "services[0].taskDefinition" xargs -l{} bash -c 'td="{ }"; name=\$(basename "\${td%:*}"); echo "\$name"; aws ecs describe-task-definition --task-definition "\$td" --region ap-northeast-2 --output json jq -r "(.taskDefinition.cpu tonumber/1024 tostring) + \" vCPU, \" + (.taskDefinition.memory tonumber/1024 tostring) + \"GB\""'</pre>
	6-3-B (예상 출력) 정확히 일치 0.4점	ws25-ecs-red-taskdef 0.5 vCPU, 1GB
	6-3-C (명령어 입력)	<pre>aws ecs describe-task-definition --task-definition \$(aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-red --query 'services[0].taskDefinition' --output text --region ap-northeast-2) --output json --region ap-northeast-2 jq -r '"ENV", (.taskDefinition.containerDefinitions[] select(.firelensConfiguration not) .environment[]?.name), "", "Secrets", (.taskDefinition.containerDefinitions[] select(.firelensConfiguration not) .secrets[]?.name)'</pre>
	6-3-C (예상 출력) 순서 무관	ENV Secrets DB_USER DB_PASSWD DB_URL

순번	채점항목	
6-4	6-4-A (명령어 입력)	<pre>ids=\$(aws ec2 describe-instances --filters Name=tag:Name,Values=ws25-ecs-container-green --region ap-northeast-2 --query 'Reservations[].Instances[].SubnetId' --output text); echo \$([\$(echo \$ids wc -w) -eq 3] && echo True echo False); names=\$(for id in \$ids; do aws ec2 describe-subnets --subnet-ids \$id --region ap-northeast-2 --query 'Subnets[0].Tags[?Key=='Name'].Value' --output text; done sort xargs); echo \$(["\$names" = "ws25-app-pri-a ws25-app-pri-b ws25-app-pri-c"] && echo True echo False); types=\$(aws ec2 describe-instances --filters Name=tag:Name,Values=ws25-ecs-container-green --region ap-northeast-2 --query 'Reservations[].Instances[].InstanceType' --output text); echo \$(echo \$types xargs -n1 grep -v t3.medium >/dev/null && echo False echo True)</pre>
	6-4-A (예상 출력) <u>정확히 일치</u>	<pre>True True True</pre>
	6-4-B (명령어 입력) <u>오답 시</u> <u>6-4 남은 항목</u> <u>채점 X</u>	<pre>aws ecs describe-tasks --cluster ws25-ecs-cluster --region ap-northeast-2 --tasks \$(aws ecs list-tasks --cluster ws25-ecs-cluster --service-name ws25-ecs-green --desired-status RUNNING --region ap-northeast-2 --query 'taskArns[]' --output text) --query 'tasks[].containerInstanceArn' --output text xargs -r aws ecs describe-container-instances --cluster ws25-ecs-cluster --region ap-northeast-2 --container-instances --query 'containerInstances[].ec2InstanceId' --output text xargs -r aws ec2 describe-instances --region ap-northeast-2 --instance-ids --query 'Reservations[].Instances[].Tags[?Key=='Name'].Value' --output text sort -u grep -qFx ws25-ecs-container-green && echo True echo False</pre>
	6-4-B (예상 출력) <u>정확히 일치</u>	<pre>True</pre>

순번	채점항목	
6-4	6-4-C (명령어 입력)	aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-green --region ap-northeast-2 --output json jq -r '.services[0] .runningCount, .availabilityZoneRebalancing, .deploymentController.type'
	6-4-C (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	3 ENABLED CODE_DEPLOY
	6-4-D (명령어 입력)	aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-green --region ap-northeast-2 --query 'services[0].loadBalancers[].targetGroupArn' --output text \\\n xargs -l {} aws elbv2 describe-target-groups --target-group-arns {} --region ap-northeast-2 --query 'TargetGroups[].LoadBalancerArns[]' --output text \\\n xargs -l {} aws elbv2 describe-load-balancers --load-balancer-arns {} --region ap-northeast-2 --query 'LoadBalancers[].LoadBalancerName' --output text
	6-4-D (예상 출력) <u>정확히 일치</u>	ws25-app-alb
	6-4-E (명령어 입력)	aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-green --region ap-northeast-2 --query 'services[0].status' --output text grep -q '^ACTIVE\$' && echo True echo False; aws ecs list-tasks --cluster ws25-ecs-cluster --service-name ws25-ecs-green --desired-status RUNNING --region ap-northeast-2 --query 'taskArns[]' --output text xargs -r aws ecs describe-tasks --cluster ws25-ecs-cluster --region ap-northeast-2 --tasks --query 'tasks[].lastStatus' --output text xargs -n1 grep -vx RUNNING >/dev/null && echo False echo True
	6-4-E (예상 출력) <u>정확히 일치</u>	True True

순번	채점항목	
6-5	6-5-A (명령어 입력)	aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-red --region ap-northeast-2 --output json \n jq -r '.services[0] if .launchType != null and .launchType != "" then .launchType else (.capacityProviderStrategy[0].capacityProvider if test("FARGATE") then "FARGATE" else "EC2" end) end'
	6-5-A (예상 출력) <u>정확히 일치</u>	FARGATE
	6-5-B (명령어 입력)	aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-red --region ap-northeast-2 --output json jq -r '.services[0] .runningCount, .availabilityZoneRebalancing, .deploymentController.type'
	6-5-B (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	3 ENABLED CODE_DEPLOY
	6-5-C (명령어 입력)	aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-red --region ap-northeast-2 --query 'services[0].loadBalancers[].targetGroupArn' --output text \n xargs -l {} aws elbv2 describe-target-groups --target-group-arns {} --region ap-northeast-2 --query 'TargetGroups[].LoadBalancerArns[]' --output text \n xargs -l {} aws elbv2 describe-load-balancers --load-balancer-arns {} --region ap-northeast-2 --query 'LoadBalancers[].LoadBalancerName' --output text
	6-5-C (예상 출력) <u>정확히 일치</u>	ws25-app-alb
	6-5-D (명령어 입력)	aws ecs describe-services --cluster ws25-ecs-cluster --services ws25-ecs-red --region ap-northeast-2 --query 'services[0].status' --output text grep -q '^ACTIVE\$' && echo True echo False; aws ecs list-tasks --cluster ws25-ecs-cluster --service-name ws25-ecs-red --desired-status RUNNING --region ap-northeast-2 --query 'taskArns[]' --output text xargs -r aws ecs describe-tasks --cluster ws25-ecs-cluster --region ap-northeast-2 --tasks --query 'tasks[].lastStatus' --output text xargs -n1 grep -vx RUNNING >/dev/null && echo False echo True
	6-5-D (예상 출력) <u>정확히 일치</u>	True True

순번	채점항목	
7-1	7-1-A (명령어 입력)	<pre>for service in ws25-ecs-green ws25-ecs-red; do task_def=\$(aws ecs describe-services --cluster ws25-ecs-cluster --services \$service --query 'services[0].taskDefinition' --output text); aws ecs describe-task-definition --task-definition \$task_def --query 'taskDefinition.containerDefinitions[?firelensConfiguration].firelensConfiguration length(@) > `0`' --output text && aws ecs describe-task-definition --task-definition \$task_def --query 'taskDefinition.containerDefinitions[?firelensConfiguration].firelensConfiguration.type' --output text grep -E 'fluentbit fluentd' && echo True; done</pre>
	7-1-A (예상 출력)	<pre>True "fluentbit" 또는 "fluentd" True True "fluentbit" 또는 "fluentd" True</pre> <p><u>단 하나라도 잘못된 값이 출력될 경우 7. Logging 항목 채점하지 않습니다</u></p>
7-2	7-2-A (명령어 입력)	<pre>for log_group in /ws25/logs/green /ws25/logs/red; do aws logs describe-log-groups --log-group-name-prefix \$log_group --query 'logGroups[0].logGroupName' --output text grep -q \$log_group && echo True echo False; done</pre>
	7-2-A (예상 출력) <u>정확히 일치</u>	<pre>True True</pre>
	7-2-B (명령어 입력)	<pre>for log_group in /ws25/logs/green /ws25/logs/red; do echo -n "\$log_group : "; total=0; log_streams=\$(aws logs describe-log-streams --log-group-name "\$log_group" --query 'logStreams[].logStreamName' --output text); count=\$(aws logs filter-log-events --log-group-name "\$log_group" --log-stream-names \$log_streams --filter-pattern "GET health" --query 'events length(@)' --output text); echo \$count; done</pre>
	7-2-B (예상 출력) <u>출력되는 모든 숫자가 0인지 확인</u>	<pre>/ws25/logs/green : 0 0 0 /ws25/logs/red : 0 0</pre>

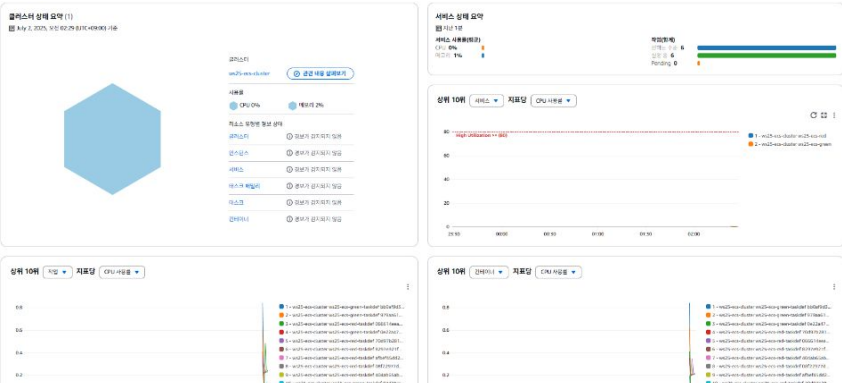

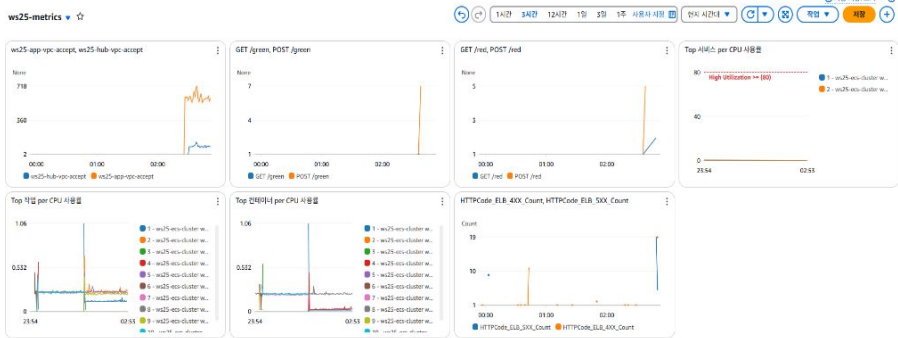
순번	채점항목	
8-1	8-1-A (명령어 입력)	aws elbv2 describe-load-balancers --names ws25-hub-nlb --query "LoadBalancers[0].[Type, Scheme]" --output text tr '\t' '\n' && aws elbv2 describe-listeners --load-balancer-arn \$(aws elbv2 describe-load-balancers --names ws25-hub-nlb --query "LoadBalancers[0].LoadBalancerArn" --output text) --query "Listeners[0].Port" --output text && aws elbv2 describe-target-groups --load-balancer-arn \$(aws elbv2 describe-load-balancers --names ws25-hub-nlb --query "LoadBalancers[0].LoadBalancerArn" --output text) --query "TargetGroups[0].{TargetGroupName:TargetGroupName, Protocol:Protocol, TargetType:TargetType}" --output text tr '\t' '\n'
	8-1-A (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	network internet-facing 80 TCP ws25-hub-nlb-tg ip
	8-1-B (명령어 입력)	aws elbv2 describe-load-balancers --names ws25-app-nlb --query "LoadBalancers[0].AvailabilityZones[]" --output text tr '\t' '\n' && for zone in \$(aws elbv2 describe-load-balancers --names ws25-app-nlb --query "LoadBalancers[0].AvailabilityZones[0].ZoneName" --output text); do ip_address=\$(aws elbv2 describe-load-balancers --names ws25-app-nlb --query "LoadBalancers[0].AvailabilityZones[?ZoneName=='\${zone}'].LoadBalancerAddresses[0].IpAddress" --output text); echo "\$zone \$ip_address"; done && aws elbv2 describe-target-groups --load-balancer-arn \$(aws elbv2 describe-load-balancers --names ws25-app-nlb --query "LoadBalancers[0].LoadBalancerArn" --output text) --query "TargetGroups[0].{TargetGroupName:TargetGroupName, Protocol:Protocol, TargetType:TargetType}" --output text
	8-1-B (예상 출력) <u>순서 무관</u>	"subnet-" 으로 시작하는 문구가 출력되는지 확인 ap-northeast-2a LOADBALANCERADDRESSES "10.200.20." 으로 시작하는 IP가 출력되는지 확인 "subnet-" 으로 시작하는 문구가 출력되는지 확인 ap-northeast-2b LOADBALANCERADDRESSES "10.200.21." 으로 시작하는 IP가 출력되는지 확인 "subnet-" 으로 시작하는 문구가 출력되는지 확인 ap-northeast-2c LOADBALANCERADDRESSES "10.200.22." 으로 시작하는 IP가 출력되는지 확인 ap-northeast-2a ap-northeast-2b ap-northeast-2c TCP ws25-app-nlb-tg alb

순번	채점항목	
8-1	8-1-C (명령어 입력)	aws elbv2 describe-load-balancers --names ws25-app-alb --query "LoadBalancers[0].[Type,Scheme]" --output text tr '\t' '\n'; aws elbv2 describe-listeners --load-balancer-arn \$(aws elbv2 describe-load-balancers --names ws25-app-alb --query "LoadBalancers[0].LoadBalancerArn" --output text) --query "Listeners[0].Protocol" --output text
	8-1-C (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	application internal HTTP
8-2	8-2-A (명령어 입력)	aws elbv2 describe-target-health --target-group-arn \$(aws elbv2 describe-target-groups --names ws25-hub-nlb-tg --query "TargetGroups[0].TargetGroupArn" --output text) --query "TargetHealthDescriptions[0].Target.Id" --output text tr '\t' '\n'; aws elbv2 describe-target-health --target-group-arn \$(aws elbv2 describe-target-groups --names ws25-hub-nlb-tg --query "TargetGroups[0].TargetGroupArn" --output text) --query "TargetHealthDescriptions[0].TargetHealth.State" --output text
	8-2-A (예상 출력) <u>순서 무관</u>	10.200.20.x (8-1-B에서 출력된 IP와 동일한지 확인) 10.200.21.x (8-1-B에서 출력된 IP와 동일한지 확인) 10.200.22.x (8-1-B에서 출력된 IP와 동일한지 확인) healthy healthy healthy (<u>2개 이상의 healthy가 출력되는지 확인</u>) <u>8-1-B에서 출력되지 않은 IP가 출력될 경우 8-2 항목 전체 오답 처리</u>

순번	채점항목	
8-2	8-2-B (명령어 입력)	aws elbv2 describe-target-health --target-group-arn \$(aws elbv2 describe-target-groups --names ws25-app-nlb-tg --query "TargetGroups[0].TargetGroupArn" --output text) --query "TargetHealthDescriptions[].TargetHealth.State" --output text tr '\t' '\n'
	8-2-B (예상 출력) <u>정확히 일치</u>	healthy
	8-2-C (명령어 입력)	aws elbv2 describe-load-balancers --names ws25-app-alb --query "LoadBalancers[0].LoadBalancerArn" --output text xargs -I {} aws elbv2 describe-target-groups --load-balancer-arn {} --query "TargetGroups[].TargetGroupArn" --output text tr '\t' '\n' xargs -I {} aws elbv2 describe-target-health --target-group-arn {} --query "TargetHealthDescriptions[?TargetHealth.State=='healthy'].Target.Id" --output text wc -w
	8-2-C (예상 출력)	6 이상의 숫자가 출력되는지 확인

순번	채점항목															
8-3	8-3-A (명령어 입력)	NLB=\$'〈본인의 ws25-hub-nlb 로드밸런서 엔드포인트〉' curl -s -o /dev/null -w '%{http_code}\n' http://\${NLB}/error; curl -s http://\${NLB}/error; echo curl -o /dev/null -s -w "%{http_code}\n" http://\${NLB}/green curl -o /dev/null -s -w "%{http_code}\n" http://\${NLB}/red														
	8-3-A (예상 출력) <u>정확히 일치</u>	500 <center><h1>500 Internal Server Error</h1></center> 400 400														
	8-3-B	<div><div><div>리스너 규칙 (6) 정보</div><div>리스너가 수신한 트래픽은 기본 작업 및 추가 규칙에 따라 라우팅됩니다. 규칙은 가장 낮은 값에서 가장 높은 값까지 우선 순위에 따라 평가됩니다.</div><div><div>Q 규칙 필터링</div><table><tr><th><input type="checkbox"/></th><th>이름 태그</th><th>우선 순위 ▲</th><th>조건(인 경우)</th><th>작업(다음 수행)</th></tr><tr><td><input type="checkbox"/></td><td>-</td><td>1</td><td>경로 패턴은 /green 또는 /green/입니다</td><td>대상 그룹으로 전달<ul style="list-style-type: none">ws25-ecs-green-tg-blue [2]: 1 (100%)ws25-ecs-green-tg-green [2]: 0 (0%)대상 그룹 고정성: 끄</td></tr><tr><td><input type="checkbox"/></td><td>-</td><td>2</td><td>경로 패턴은 /red/ 또는 /red/입니다</td><td>대상 그룹으로 전달<ul style="list-style-type: none">ws25-ecs-red-tg-green [2]: 1 (100%)ws25-ecs-red-tg-blue [2]: 0 (0%)대상 그룹 고정성: 끄</td></tr></table></div></div><div>1. ws25-app-alb 리스너를 클릭합니다. 2. 위와 같이 /green 경로로 들어오는 요청은 Green 서비스로 전달되며, /red 경로로 들어오는 요청은 Red 서비스로 전달되는지 확인합니다. (대상그룹의 이름과 개수는 달라도 무방)</div></div>	<input type="checkbox"/>	이름 태그	우선 순위 ▲	조건(인 경우)	작업(다음 수행)	<input type="checkbox"/>	-	1	경로 패턴은 /green 또는 /green/입니다	대상 그룹으로 전달 <ul style="list-style-type: none">ws25-ecs-green-tg-blue [2]: 1 (100%)ws25-ecs-green-tg-green [2]: 0 (0%)대상 그룹 고정성: 끄	<input type="checkbox"/>	-	2	경로 패턴은 /red/ 또는 /red/입니다
<input type="checkbox"/>	이름 태그	우선 순위 ▲	조건(인 경우)	작업(다음 수행)												
<input type="checkbox"/>	-	1	경로 패턴은 /green 또는 /green/입니다	대상 그룹으로 전달 <ul style="list-style-type: none">ws25-ecs-green-tg-blue [2]: 1 (100%)ws25-ecs-green-tg-green [2]: 0 (0%)대상 그룹 고정성: 끄												
<input type="checkbox"/>	-	2	경로 패턴은 /red/ 또는 /red/입니다	대상 그룹으로 전달 <ul style="list-style-type: none">ws25-ecs-red-tg-green [2]: 1 (100%)ws25-ecs-red-tg-blue [2]: 0 (0%)대상 그룹 고정성: 끄												

순번	채점항목	
9-1	9-1-A (명령어 입력)	NLB=\$'http://<본인의 ws25-hub-nlb 로드밸런서 엔드포인트>' curl --silent -X POST -H "Content-Type: application/json" -d '{"x": "eks", "y": 81}' http://\${NLB}/green
	9-1-A (예상 출력)	{"id":"1A7Fy0qi","status":"inserted"} (출력된 id 기록)
	9-1-B (명령어 입력)	curl --silent http://\${NLB}/green?id=<9-1-A에서 출력된 id 입력>
	9-1-B (예상 출력) <u>정확히 일치</u>	{"version":"1.0.0","x":"eks","y":81}
9-2	9-2-A (명령어 입력)	curl --silent -X POST -H "Content-Type: application/json" -d '{"name":"kubectrl"}' http://\${NLB}/red
	9-2-A (예상 출력)	{"id":"1Om46BTU","status":"inserted"} (출력된 id 기록)
	9-2-B (명령어 입력)	curl --silent http://\${NLB}/red?id=<9-2-A에서 출력된 id 입력>
	9-2-B (예상 출력) <u>정확히 일치</u>	{"name":"kubectrl","version":"1.0.0"}

순번	채점항목	
10-1	10-1-A	 <p>CloudWatch 콘솔의 Container Insights에서 사진과 같이 ws25-ecs-cluster의 지표를 볼 수 있는지 확인합니다.</p>
10-2	10-2-A	 <p>ws25-metrics 대시보드가 생성되었는지 확인합니다.</p>
	10-2-B	 <p>대시보드의 지표들이 사진과 같은지 확인합니다. (지표 이름 및 위젯 유형, 순서 무관)</p>
10-3	10-3-A	<ol style="list-style-type: none"> 1. NLB=\$'본인의 ws25-hub-nlb 로드밸런서 엔드포인트'을 입력합니다. 2. curl http://\$ {NLB}/4xx을 빠르게 10번 입력합니다. 3. 약 3분 이내에 4xx에 대한 경보가 경고 상태로 전환되었는지 확인합니다.
	10-3-B	<ol style="list-style-type: none"> 1. NLB=\$'본인의 ws25-hub-nlb 로드밸런서 엔드포인트'을 입력합니다. 2. curl http://\$ {NLB}/error을 빠르게 5번 입력합니다. 3. 약 3분 이내에 5xx에 대한 경보가 경고 상태로 전환되었는지 확인합니다.

순번	채점항목	
10-4	10-4-A	<ol style="list-style-type: none"> 1. NLB='\$<본인의 ws25-hub-nlb 로드밸런서 엔드포인트>' 를 입력합니다. 2. curl --silent -X POST -H "Content-Type: application/json" -d '{"x": "test", "y": 111}' http://\${NLB}/green 를 입력하여 출력된 id를 기록합니다. 3. curl http://\${NLB}/green?id=<기록한 id> 을 입력합니다. 4. curl --silent -X POST -H "Content-Type: application/json" -d '{"name":"test2"}' http://\${NLB}/red 를 입력하여 출력된 id를 기록합니다. 5. curl http://\${NLB}/red?id=<기록한 id> 를 입력합니다. 6. 약 1분 후 green, red 요청에 대한 대시보드의 지표 값이 변화했는지 확인합니다.

순번	채점항목	
11-1	11-1-A (명령어 입력)	for deployment_group in \$(aws deploy list-deployment-groups --application-name ws25-cd-green-app --query 'deploymentGroups' --output text); do echo "Deployment Group: \$deployment_group"; aws deploy get-deployment-group --application-name ws25-cd-green-app --deployment-group-name "\$deployment_group" --query 'deploymentGroupInfo.ecsServices' --output json; done
	11-1-A (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	Deployment Group: ws25-cd-green-dg [{ "serviceName": "ws25-ecs-green", "clusterName": "ws25-ecs-cluster" }]
11-2	11-2-A (명령어 입력)	for deployment_group in \$(aws deploy list-deployment-groups --application-name ws25-cd-red-app --query 'deploymentGroups' --output text); do echo "Deployment Group: \$deployment_group"; aws deploy get-deployment-group --application-name ws25-cd-red-app --deployment-group-name "\$deployment_group" --query 'deploymentGroupInfo.ecsServices' --output json; done
	11-2-A (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	Deployment Group: ws25-cd-red-dg [{ "serviceName": "ws25-ecs-red", "clusterName": "ws25-ecs-cluster" }]
11-3	11-3-A (명령어 입력)	S3_BUCKET=\$(aws codepipeline get-pipeline --name ws25-cd-green-pipeline --query 'pipeline.stages[?name=='Source'].actions[0].configuration.S3Bucket' --output text); \ DEPLOY_GROUP=\$(aws codepipeline get-pipeline --name ws25-cd-green-pipeline --query 'pipeline.stages[?name=='Deploy'].actions[0].configuration.DeploymentGroupName' --output text); \ echo "S3 Bucket: \$S3_BUCKET"; echo "Deploy Group: \$DEPLOY_GROUP"
	11-3-A (예상 출력) <u>순서 중요</u>	S3 Bucket: ws25-cd-green-artifact-〈비번호〉 Deploy Group: ws25-cd-green-dg

순번	채점항목	
11-4	11-4-A (명령어 입력)	<pre> S3_BUCKET=\$(aws codepipeline get-pipeline --name ws25-cd-red-pipeline --query 'pipeline.stages[?name=='Source'].actions[0].configuration.S3Bucket' --output text); \ DEPLOY_GROUP=\$(aws codepipeline get-pipeline --name ws25-cd-red-pipeline --query 'pipeline.stages[?name=='Deploy'].actions[0].configuration.DeploymentGroupNam e' --output text); \ echo "S3 Bucket: \$S3_BUCKET"; echo "Deploy Group: \$DEPLOY_GROUP" </pre>
	11-4-A (예상 출력) <u>순서 중요</u>	<pre> S3 Bucket: ws25-cd-red-artifact-<비번호> Deploy Group: ws25-cd-red-dg </pre>
11-5	11-5-A (명령어 입력)	<pre> echo -e "green.sh: \$([-f /home/ec2-user/pipeline/green.sh] && echo True echo False)\nred.sh: \$([-f /home/ec2-user/pipeline/red.sh] && echo True echo False)\nartifact dir: \$([-d /home/ec2-user/pipeline/artifact] && echo True echo False)\ngreen dir: \$([-d /home/ec2-user/pipeline/artifact/green] && echo True echo False)\nred dir: \$([-d /home/ec2-user/pipeline/artifact/red] && echo True echo False)\ngreen files: \$([\$(ls /home/ec2-user/pipeline/artifact/green wc -l) -ge 1] && echo True echo False)\nred files: \$([\$(ls /home/ec2-user/pipeline/artifact/red wc -l) -ge 1] && echo True echo False)" </pre>
	11-5-A (예상 출력) <u>정확히 일치</u> <u>순서 중요</u>	<pre> green.sh: True red.sh: True artifact dir: True green dir: True red dir: True green files: True red files: True </pre> <p>한 개 이상의 False 발생 시 다른 항목이 성공해도 오답 처리됩니다.</p>

순번	채점항목
11-6	<div data-bbox="293 658 379 689">11-6-A</div> <div data-bbox="461 275 1415 633"> <p>1. NLB=\${<본인의 ws25-hub-nlb 로드밸런서 엔드포인트>}' <code>curl --silent -X POST -H "Content-Type: application/json" -d '{"x": "codebuild", "y": 105}' http://\${NLB}/green</code> 을 입력하여 출력된 ID를 기록합니다.</p> <p>2. <code>curl --silent http://\${NLB}/green?id=<기록한 id></code> 를 입력하여 <code>{"version": "1.0.0", "x": "codebuild", "y": 105}</code> 가 정상적으로 출력되는지 확인합니다.</p> <p>3. <code>/home/ec2-user/pipeline/</code> 경로에서 <code>"bash green.sh"</code> 명령을 실행합니다.</p> <p>4. 3분 이내에 사진과 같이 오류없이 배포가 완료되어야 합니다.</p> </div> <div data-bbox="461 658 1062 981"> <p>The screenshot shows the AWS CodePipeline console for a pipeline named 'ws25-cd-green-pipeline'. It has two stages: 'Source' and 'Deploy'. Both stages are using 'Amazon S3' as the provider and show a successful status with a green checkmark. The Source stage ID is '8b424f4b-4629-4ba1-bf7b-6cb49fa6ee64' and the Deploy stage ID is '8b424f4b-4629-4ba1-bf7b-6cb49fa6ee64'. Both stages show '모든 작업이 성공했습니다.' (All jobs succeeded).</p> </div> <div data-bbox="461 981 1415 1070"> <p>5. <code>curl --silent http://\${NLB}/green?id=<기록한 id></code> 를 입력하여 <code>{"version": "1.0.1", "x": "codebuild", "y": 105}</code> 가 정상적으로 출력되는지 확인합니다.</p> </div>
11-7	<div data-bbox="293 1585 379 1617">11-7-A</div> <div data-bbox="461 1196 1426 1563"> <p>1. NLB=\${<본인의 ws25-hub-nlb 로드밸런서 엔드포인트>}' <code>curl --silent -X POST -H "Content-Type: application/json" -d '{"name": "github action"}' http://\${NLB}/red</code> 을 입력하여 출력된 ID를 기록합니다.</p> <p>2. <code>curl --silent http://\${NLB}/red?id=<기록한 id></code> 를 입력하여 <code>{"name": "github action", "version": "1.0.0"}</code> 이(가) 정상적으로 출력되는지 확인합니다.</p> <p>3. <code>/home/ec2-user/pipeline/</code> 경로에서 <code>"bash red.sh"</code> 명령을 실행합니다.</p> <p>4. 3분 이내에 사진과 같이 오류없이 배포가 완료되어야 합니다.</p> </div> <div data-bbox="461 1585 1062 1908"> <p>The screenshot shows the AWS CodePipeline console for a pipeline named 'ws25-cd-red-pipeline'. It has two stages: 'Source' and 'Deploy'. Both stages are using 'Amazon S3' as the provider and show a successful status with a green checkmark. The Source stage ID is '997ea1ae-0918-4eb7-a9b0-2b292b27c7d1' and the Deploy stage ID is '997ea1ae-0918-4eb7-a9b0-2b292b27c7d1'. Both stages show '모든 작업이 성공했습니다.' (All jobs succeeded).</p> </div> <div data-bbox="461 1908 1426 1998"> <p>5. <code>curl --silent http://\${NLB}/red?id=<기록한 id></code> 를 입력하여 <code>{"name": "github action", "version": "1.0.1"}</code> 이(가) 정상적으로 출력되는지 확인합니다.</p> </div>