

2024 전국기능경기대회 채점기준

1. 채점상의 유의사항	직 종 명	클라우드컴퓨팅
<p>※ 다음 사항을 유의하여 채점하십시오.</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) [] 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. 		

2. 채점기준표

1) 주요항목별 배점

1) 주요항목별 배점			직 종 명		클라우드컴퓨팅			
과제 번호	일련 번호	주요항목	배점	채점방법		채점시기		비고
				독립	합의	경기 진행중	경기 종료후	
제2과제	1	Building an CI/CD Pipeline using Argocd and CodeSeries	3.75		○		○	
합 계			3.75					

2) 채점방법 및 기준

과제 번호	일련 번호	주요항목	일련 번호	세부항목(채점방법)	배점
2과제	1	Building an CI/CD Pipeline using ArgoCD and CodeSeries	1	기본 구성 확인	1
			2	CI/CD Pipeline	1.5
			3	Blue/Green Deploy	1.25
	총점				3.75

3) 채점내용

순번	채점 항목	
1-1	1-1 (명령어 입력)	Bastion EC2에 접근 후 실행 <code>kubectl get crd grep argoproj.io awk {'print \$1'} ₩</code> <code>; curl \$(kubectl get ingress -n app grep blue-green-ingress awk {'print \$4'})</code>
	1-1 (예상 출력) <u>정확히 일치</u>	<code>analysisruns.argoproj.io</code> <code>analysistemplates.argoproj.io</code> <code>applications.argoproj.io</code> <code>applicationsets.argoproj.io</code> <code>appprojects.argoproj.io</code> <code>clusteranalysistemplates.argoproj.io</code> <code>experiments.argoproj.io</code> <code>rollouts.argoproj.io</code> This is Blue Page
1-2	1-2 (명령어 입력 및 작업 수행)	CodeCommit 콘솔 접근 후, gwangju-application-repo의 main.py 클릭

Developer Tools
CodeCommit

Developer Tools > CodeCommit > Repositories > application-repo

application-repo

Reference

Notify
main
Create pull request

Clone URL

application-repo
Info
Add file

Name
buildspec.yml
Dockerfile
main.py
requirements.txt

Edit 클릭

application-repo / main.py
Info
Edit

```

1 from fastapi import FastAPI, Response
2 import uvicorn
3
4 app = FastAPI()
5
6
7 @app.get("/")
8 def return_version():
9     return Response(content="This is Blue Page", status_code=200)
10
11
12 @app.get("/healthcheck")
13 def return_healthz():
14     return Response(content="healthy", status_code=200)
15
16
17 if __name__ == "__main__":
18     uvicorn.run(app, host="0.0.0.0", port=8080)
19

```

Blue를 Green으로 수정

application-repo / main.py Info

The code editor uses the Tab key to control indentation. To navigate away from the code editor, use Escape plus Tab keys.

```
1 from fastapi import FastAPI, Response
2 import uvicorn
3
4 app = FastAPI()
5
6
7 @app.get("/")
8 def return_version():
9     return Response(content="This is Green Page", status_code=200)
10
11
12 @app.get("/healthcheck")
13 def return_healthz():
14     return Response(content="healthy", status_code=200)
15
16
17 if __name__ == "__main__":
18     uvicorn.run(app, host="0.0.0.0", port=8080)
19 |
```

Author name, Email address 두 값 모두 test로 하고 Commit changes 클릭

Commit changes to main

File: application-repo/main.py

Author name

test

Email address

test

Commit message - optional

A default commit message will be used if you do not provide one.

Cancel

Commit changes

커밋 후 3분 정도 지난후에 ALB 주소에 접근하여 Blue가 아닌 Green으로 나오는 지 확인

```
sleep 180; curl $(kubectl get ingress -n app | grep blue-green-ingress | awk
{'print $4'})
```

	1-2 (예상 출력) <u>정확히 일치</u>	This is Green Page
1-3	1-3 (명령어 입력)	kubectl argo rollouts get rollout blue-green-app -n app egrep "Strategy" ₩ ; kubectl argo rollouts get rollout blue-green-app -n app egrep "stable" grep "Healthy" awk {'print \$6,\$8}
	1-3 (예상 출력) <u>정확히 일치</u>	Strategy: BlueGreen Healthy stable,active