

[← Return to Classroom](#)

Give Your Application Auto-Deploy Superpowers

[REVIEW](#)[CODE REVIEW 1](#)[HISTORY](#)

Requires Changes

2 specifications require changes

Hi Paul !

You did an amazing job considering it was your first attempt. Some of the points that I liked about your submission :

- *Excellent presentation for business stakeholders* 🙌
- *You used Slack Notification instead of Email Notification for the job alert notifications* 🙌
- *Build, Test and Analyze job for Frontend and Backend were done perfectly* 🙌
- *You were able to successfully deploy a web application using CI/CD pipeline* 🙌

You are very close to the appropriate project, however, you need to correct couple of small points before you're all set to go.

- ⚠️ *Provide Screenshot/Video (URL should be visible) OR Make sure that we are able to add the employees using the Frontend Application accessed via the S3 bucket endpoint*
- ⚠️ *Provide Screenshot/Video (URL should be visible) OR Make sure that we are able to add the employees using the Frontend Application accessed via the CloudFront domain name URL*
- ⚠️ *Make sure that the DNS name/IP in the Alert Notification sent by Alert Manager matches with any 1 of the Prometheus targets*

Hope you make the required changes soon. We look forward to receiving your future project submissions soon .

Additional Resources💡 :

- If you face challenge while working on Adding the employees part, you can go through the following Knowledge posts to get some idea about potential causes :
 - [Having an error in adding employee using CloudFront now](#)
 - [Can't add employee on CloudFront url](#)
 - [Can't add an employee](#)
- The best resource to learn anything is to read the documentation. If you developed an interest for configuration management tool like Ansible, you can go through the official documentation of Ansible - [Ansible Documentation](#)

PS: If you have any doubts regarding any of the concept, feel free to search or post a question on [Knowledge Portal](#) where many of the fellow students and mentors may have faced the same situation before and would have provided the appropriate steps to resolve it.

Have a Good Day and Stay Safe 🤘

Keep Learning and Stay Udacious



Section 1: Selling CI/CD to your Team/Organization

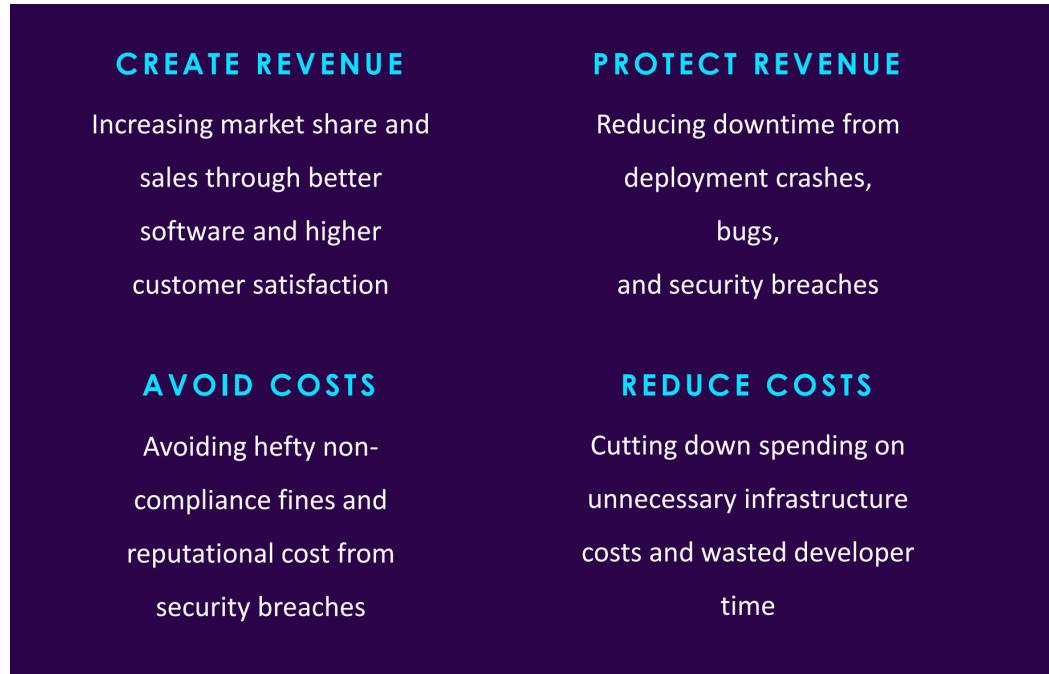
The CI/CD benefits proposal contains essential benefits of CI/CD, and describes the business context that will benefit from the automation tools. Explanation should include benefits that translate to revenue and cost for the business.

Great presentation. ✓

It includes all of the following points:

- Benefits of CI/CD. ✓
- Business context that will benefit from the automation tools ✓

- Translations in terms of revenue and cost for the business. 



Additional Reading Resource

You can go through the following LinkedIn Post which talks about - [5 Tips for Giving Effective Technical Presentations](#)

Section 2: Deploying Working, Trustworthy Software

A public git repository with your project code. [URL01]

Evidence of code-based CI/CD configuration in the form of yaml files in your git repository.

Console output of various pre-deploy job failure scenarios:

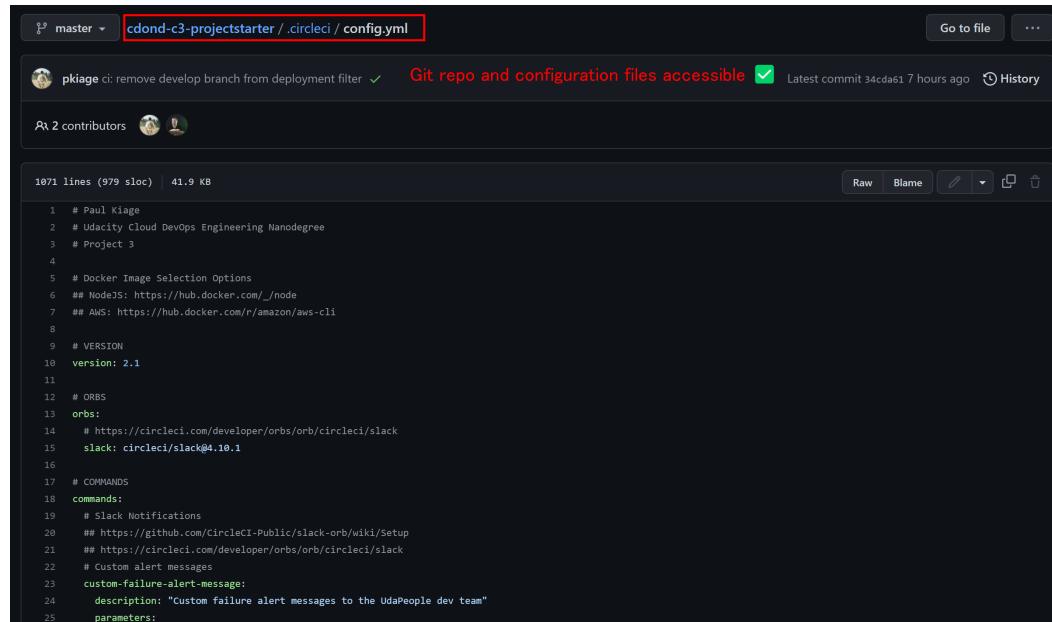
- Build Jobs that failed because of compile errors. [SCREENSHOT01]
- Failed unit tests. [SCREENSHOT02]
- Failure because of vulnerable packages. [SCREENSHOT03]
- An alert from one of your failed builds. [SCREENSHOT04]

Evidence in your code that:

- Compile errors have been fixed.
- Unit tests have been fixed.
- All critical security vulnerabilities caught by the "Analyze" job have been fixed.

1. [URL01] - A public git repository with your project code. 

- Evidence of code-based CI/CD configuration in the form of yaml files can be found in your git repository.

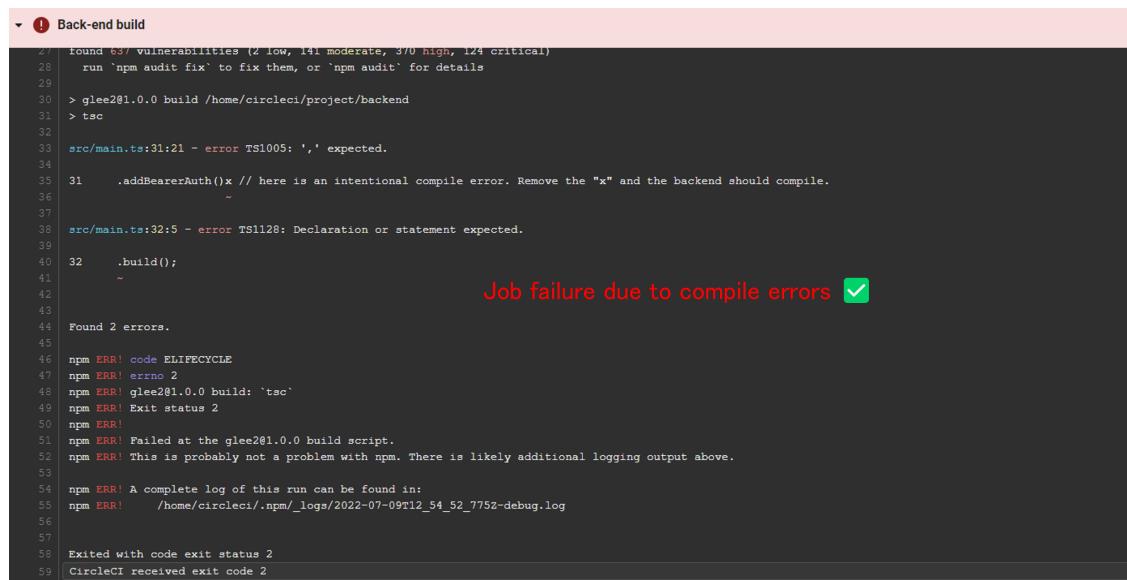


```

1 # Paul Kiage
2 # Udacity Cloud DevOps Engineering Nanodegree
3 # Project 3
4
5 # Docker Image Selection Options
6 ## NodeJS: https://hub.docker.com/_/node
7 ## AWS: https://hub.docker.com/r/amazon/aws-cli
8
9 # VERSION
10 version: 2.1
11
12 # ORBS
13 orbs:
14   # https://circleci.com/developer/orbs/orb/circleci/slack
15   slack: circleci/slack@4.10.1
16
17 # COMMANDS
18 commands:
19   # Slack Notifications
20   ## https://github.com/CircleCI-Public/slack-orb/wiki/Setup
21   ## https://circleci.com/developer/orbs/orb/circleci/slack
22   # Custom alert messages
23   custom-failure-alert-message:
24     description: "Custom failure alert messages to the UdaPeople dev team"
25     parameters:

```

2. [SCREENSHOT01] - Console output of build jobs that failed because of compile errors.



```

27 found 637 vulnerabilities (2 low, 141 moderate, 370 high, 124 critical)
28   run 'npm audit fix' to fix them, or 'npm audit' for details
29
30 > glee2@1.0.0 build /home/circleci/project/backend
31 > tsc
32
33 src/main.ts:31:21 - error TS1005: ',' expected.
34
35 31     .addBearerAuth()x // here is an intentional compile error. Remove the "x" and the backend should compile.
36   ~
37
38 src/main.ts:32:5 - error TS1128: Declaration or statement expected.
39
40 32     .build();
41   ~
42
43 Found 2 errors.
44
45 npm ERR! code ELIFECIRCLE
46 npm ERR! errno 2
47 npm ERR! glee2@1.0.0 build: `tsc`
48 npm ERR! Exit status 2
49 npm ERR!
50 npm ERR! Failed at the glee2@1.0.0 build script.
51 npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
52
53 npm ERR! A complete log of this run can be found in:
54 npm ERR!     /home/circleci/.npm/_logs/2022-07-09T12_54_52_775Z-debug.log
55
56
57 Exited with code exit status 2
58 CircleCI received exit code 2

```

Job failure due to compile errors 

3. [SCREENSHOT02] - Console output of failed unit tests

```

    279 |     Number of calls: 1
280 |
281 |     33 |         // Assert
282 |     > 34 |             expect(employeeRepository.findById).toBeCalledWith(100);
283 |     |
284 |     36 |                 expect(employeeRepository.save).toBeCalled();
285 |     37 |             });
286 |     });
287 |
288 Test Suites: 1 failed, 50 passed, 51 total
289 Tests:       1 failed, 78 passed, 79 total
290 Snapshots:   0 total
291 Time:        34.286s
292 Ran all test suites.
293 npm ERR! code ELIFECYCLE
294 npm ERR! errno 1
295 npm ERR! glee2@1.0.0 test: `jest --runInBand`
296 npm ERR! Exit status 1
297 npm ERR!
298 npm ERR! Failed at the glee2@1.0.0 test script.
299 npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
300
301 npm ERR! A complete log of this run can be found in:
302 npm ERR!     /home/circleci/.npm/_logs/2022-07-09T13_21_08_879Z-debug.log
303
304
305 Exited with code exit status 1
306 CircleCI received exit code 1

```

Unit test failure ✓

4. [SCREENSHOT03] - Console output of failure because of vulnerable packages. ✓

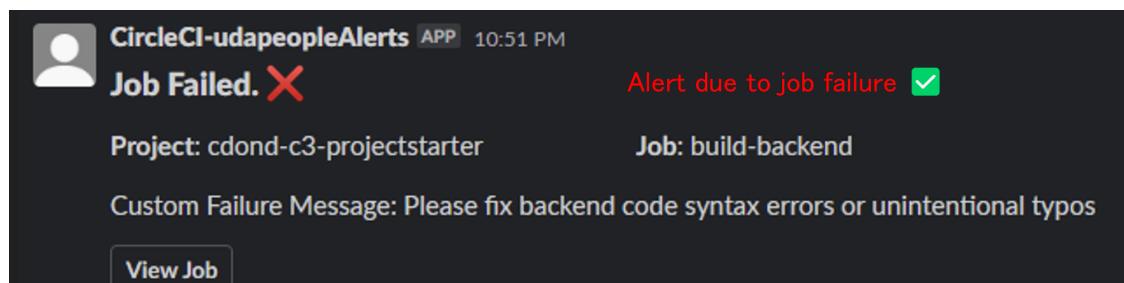
```

    5563 | Moderate | Prototype Pollution in Ajv
5564 | Package  | ajv
5565 |
5566 | Patched in | >=6.12.3
5567 |
5568 | Dependency of | winston-loggly-bulk
5569 |
5570 | Path | winston-loggly-bulk > node-loggly-bulk > request >
5571 |           har-validator > ajv
5572 |
5573 | More info | https://github.com/advisories/GHSA-v88g-cgmmr-v5xw
5574 |
5575
5576 | High | Crash in HeaderParser in dicer
5577 | Package  | dicer
5578 |
5579 | Patched in | No patch available
5580 |
5581 | Dependency of | @nestjs/platform-express
5582 |
5583 | Path | @nestjs/platform-express > multer > busboy > dicer
5584 |
5585 | More info | https://github.com/advisories/GHSA-wm7h-9275-46v2
5586 |
5587
5588 Found 637 vulnerabilities (2 low, 141 moderate, 370 high, 124 critical) in 1389 scanned packages
5589 run `npm audit fix` to fix 496 of them.
5590 134 vulnerabilities require semver-major dependency updates.
5591 7 vulnerabilities require manual review. See the full report for details.
5592
5593
5594 Exited with code exit status 1
5595 CircleCI received exit code 1

```

Job failure due to vulnerable packages ✓

5. [SCREENSHOT04] - An alert from one of your failed build ✓



Console output of appropriate failure for infrastructure creation job (using CloudFormation). [SCREENSHOT05]

Console output of a smoke test job that is failing appropriately. [SCREENSHOT06]

Console output of a successful rollback after a failed smoke test. [SCREENSHOT07]

Console output of successful promotion of new version to production in CloudFront. [SCREENSHOT08]

Console output of successful cleanup job that removes old S3 bucket and EC2 instance. [SCREENSHOT09]

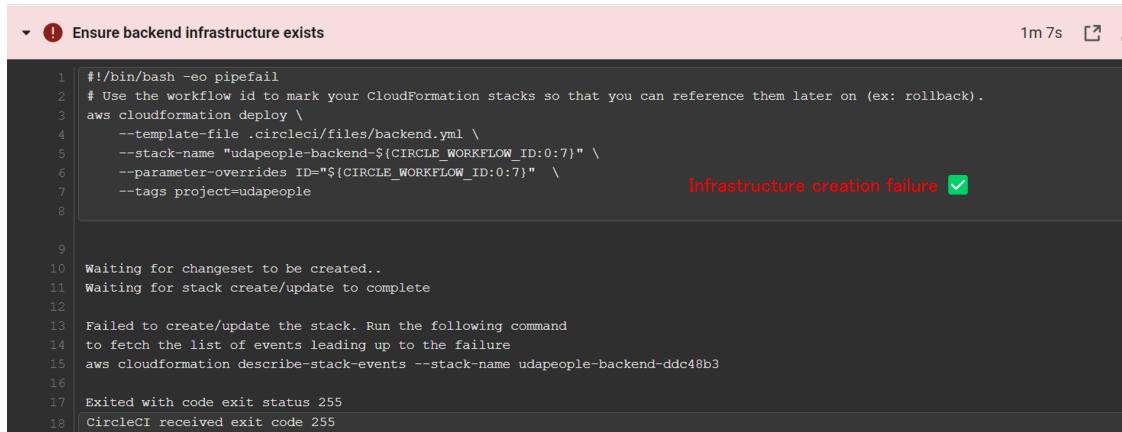
Evidence that the deploy jobs only happen on the `master` branch. [SCREENSHOT10]

Evidence of deployed and functioning front-end application in an S3 bucket [URL02].

Evidence of deployed and functioning front-end application in CloudFront. [URL03_SCREENSHOT]

Evidence of healthy back-end application. [URL04_SCREENSHOT]

1. [SCREENSHOT05] - Console output of appropriate failure for infrastructure creation job (using CloudFormation). 



The screenshot shows a terminal window with a pink header bar. The header bar contains a red exclamation mark icon and the text "Ensure backend infrastructure exists". In the top right corner, there is a timestamp "1m 7s" and a copy/paste icon. The main terminal area displays a bash script execution. Line 1 starts with "#!/bin/bash -eo pipefail". Lines 2-7 show the command "aws cloudformation deploy" with various parameters: --template-file .circleci/files/backend.yml, --stack-name "udapeople-backend-\${CIRCLE_WORKFLOW_ID:0:7}", --parameter-overrides ID="\${CIRCLE_WORKFLOW_ID:0:7}", and --tags project=udapeople. A red error message "Infrastructure creation failure" with a checkmark icon appears at the end of the log. The log continues with "Waiting for changeset to be created.." and "Waiting for stack create/update to complete". It then states "Failed to create/update the stack. Run the following command to fetch the list of events leading up to the failure" followed by the command "aws cloudformation describe-stack-events --stack-name udapeople-backend-ddc48b3". The final lines show "Exited with code exit status 255" and "CircleCI received exit code 255".

2. [SCREENSHOT06] - Console output of a smoke test job that is failing appropriately. 

Frontend smoke test.

```

1 #!/bin/sh -eo pipefail
2 # your code here
3 # Form the front-end url using the workflow id and your AWS region.
4 URL="http://udapeople-${CIRCLE_WORKFLOW_ID:0:7}.s3-website-us-west-2.amazonaws.com/#/employees"
5 echo ${URL}
6 # Check the front-end to make sure it includes a word or two that proves it is working properly.
7 # No errors mean a successful test
8 if curl -s ${URL} | grep "Welcome"
9 then
10   # Change this to 0 after the job fails (SCREENSHOT06)
11   return 1
12 else
13   return 1
14 fi
15

16 http://udapeople-b800653.s3-website-*****.amazonaws.com/#/employees
17   Welcome
18
19 Exited with code exit status 1
20 CircleCI received exit code 1

```

Failed Smoke Test ✓

3. [SCREENSHOT07] - Console output of a successful rollback after a failed smoke test. ✓

Destroy environments

```

1 #!/bin/sh -eo pipefail
2 echo "${CIRCLE_WORKFLOW_ID}"
3 # Your code here
4 aws cfninformation delete-stack --stack-name udapeople-backend-${CIRCLE_WORKFLOW_ID:0:7}
5 aws cloudformation delete-stack --stack-name udapeople-frontend-${CIRCLE_WORKFLOW_ID:0:7}
6 aws s3 rm s3://udapeople-${CIRCLE_WORKFLOW_ID:0:7} --recursive
7

8 f96808c9-ee6a-4737-a54e-1590425a13a7
9 delete: s3://udapeople-f96808c/1cefff9123b66b50a7d3cb5b221160855.eot
10 delete: s3://udapeople-f96808c/224b6d9d16679dab02826a2a7ea705eb.ttf
11 delete: s3://udapeople-f96808c/508d91b741cdebd5d476.js
12 delete: s3://udapeople-f96808c/images/6ab15f4239b7d187935cf6f4ec3313bf-kiwi.svg
13 delete: s3://udapeople-f96808c/bundle.js
14 delete: s3://udapeople-f96808c/2fa692cd5bfc83c228c0ab271e4388c.woff
15 delete: s3://udapeople-f96808c/index.html
16 delete: s3://udapeople-f96808c/7d234c755f55c47d2c5a.js
17 CircleCI received exit code 0

```

Successful Rollback ✓

Revert migrations

```

1 #!/bin/sh -eo pipefail
2 # Curl command here to see if there was a successful migration associated with the workflow id, store result in SUCCESS variable
3 # Using kvdb.io
4 SUCCESS=$(curl --insecure https://kvdb.io/HmnnciQG9qtfnJNaBpePv3/migration_${CIRCLE_WORKFLOW_ID:0:7})
5 # Logic for reverting the database state
6 if(( $SUCCESS==1 ));
7 then
8   cd ~/project/backend
9   npm install
10  npm run migration:revert
11 fi
12

13   % Total    % Received % Xferd  Average Speed   Time   Time     Current
14                               Dload  Upload Total Spent   Left Speed
15 100    9 100    9    0      0  81      0  --:-- --:-- --:--  81
16 /bin/sh: Not: not found
17 CircleCI received exit code 0

```

4. [SCREENSHOT08] - Console output of successful promotion of new version to production in CloudFront. ✓

3m 9s ↗ ↘

▼ ✓ Update cloudfont distribution

```
1 #!/bin/sh -eo pipefail
2 # your code here
3 # Change the initial stack name, as applicable to you
4 aws cloudformation deploy \
5     --template-file .circleci/files/cloudfront.yml \
6     --stack-name InitialStack \
7     --parameter-overrides WorkflowID="udapeople-${CIRCLE_WORKFLOW_ID:0:7}" \
8     --tags project=udapeople
9
10
11 Waiting for changeset to be created..
12 Waiting for stack create/update to complete
13 Successfully created/updated stack - InitialStack
14 CircleCI received exit code 0
```

Cloudfront promotion ✓

5. [SCREENSHOT09] - Console output of successful cleanup job that removes old S3 bucket and EC2 instance. ✓

▼ ✓ Remove old stacks and files: Remove dead-end production stacks from Blue Green Deployment 0s ↗ ↘

```
1 #!/bin/bash -eo pipefail
2 if [[ "${CIRCLE_WORKFLOW_ID:0:7}" =~ "${PriorWorkflowID}" ]]
3 then
4     echo "Already clean/nothing to clean"
5 else
6     echo "Cleaning up"                                Successful cleanup job ✓
7
8     echo "Emptying bucket created by .circleci/files/frontend.yml"
9     # need to empty bucket before deleting it when deleting frontend stack
10    aws s3 rm "s3://udapeople-${PriorWorkflowID}" --recursive
11
12    echo "Deleting frontend stack (S3 bucket and policy) created by .circleci/files/frontend.yml"
13    aws cloudformation delete-stack --stack-name "udapeople-frontend-${PriorWorkflowID}"
14
15    echo "Deleting backend stack (EC2 instance and rules) created by .circleci/files/backend.yml"
16    aws cloudformation delete-stack --stack-name "udapeople-backend-${PriorWorkflowID}"
17 fi
18
19 Already clean/nothing to clean
20 CircleCI received exit code 0
```

6. [SCREENSHOT10] - Evidence that the deploy jobs only happen on the master branch. ✓

Dashboard Project Branch Workflow

All Pipelines > cdond-c3-projectstarter > feature-customNotifications > default

default Success

Duration / Finished	Branch	Commit	Author & Message
1m 51s / 34m ago	feature-customNotifications	023ae4a	attempt(ci): custom slack notifications

Non-master branches not running complete workflow X

```

graph LR
    buildFrontend[build-frontend 32s] --> scanFrontend[scan-frontend 52s]
    buildFrontend --> testFrontend[test-frontend 1m 10s]
    buildBackend[build-backend 40s] --> scanBackend[scan-backend 1m 2s]
    buildBackend --> testBackend[test-backend 1m 7s]
  
```

7. [URL02] - Evidence of deployed and functioning front-end application in an S3 bucket. X

- Great work deploying the front-end application in the S3 bucket and providing the URL X. However, I was not able to access the application X

Not secure udapeople-ae1a57cs3-website-us-west-2.amazonaws.com

404 Not Found

- Code: NoSuchBucket
- Message: The specified bucket does not exist
- BucketName: udapeople-ae1a57c
- RequestId: QFNDCM074RSQ436C
- HostId: RXJzxZnchAVXpdHjDSa+8u04GSlaP7u/cRXEDXy/D1VvZxmg4Tlr47LqNqKXuOkzINWw1E2K5o=

Please ensure that you are able to add employees using the S3 link for the application. Here is a sample screenshot for your reference X

Not secure udapeople-edc305e.s3-website-us-west-2.amazonaws.com/#/employees

This shows that application is accessible

View & Manage Employees

Add New Employee

Display Name	Last Name	Email	Tags	Birthdate	Start Date	Action
Ayman	Samir	[REDACTED]		6/15/2022	6/15/2022	X
aa	Samirdsad	a.[REDACTED]@gmail....		6/18/2022	6/18/2022	X

This shows that application is functional

8. [URL03_SCREENSHOT] - Evidence of deployed and functioning front-end application in CloudFront. ✗

- Again, great work providing the CloudFront Domain name URL screenshot. I was able to access the application 🙌 However, the specification requires that the application should be functional . As can be seen below, I was not able to add the employee using the current front-end application 🤦

The screenshot shows a web browser window with the URL <https://d2zeyy0duyfl8.cloudfront.net/#/employees/new>. The page title is "Add a New Employee". On the left, there's a sidebar with "Employees", "View & Manage", and "Add New". The main area has a "TAGS" section with a placeholder "Press tab to create tags." Below it is a "Display Name" field with a placeholder "T", followed by "First Name" and "Middle name" fields. Then "Last Name" and "Second Last Name" fields. A "Gender" dropdown is set to "Male". Under "SALARY", "Yearly" is selected over "Hourly". "Set amount" is \$0, and "Effective Date" is 07/17/2022. At the bottom right is a "Save" button labeled "Add Employee". A large red arrow points from the text above to the "Display Name" field.

- Just like, previous point, this point requires that you are able to add the employees using the application that is accessed via the CloudFront domain name URL. Therefore, please ensure that you are able to add employees using the CloudFront domain name URL for the application. Here is a sample screenshot for your reference

The screenshot shows a web browser window with the URL <https://d3segrpg3jdzr.cloudfront.net/#/employees>. The page title is "View & Manage Employees". On the left, there's a sidebar with "Employees", "View & Manage", and "Add New". The main area has a table with columns: Display Name, Last Name, Email, Tags, Birthdate, Start Date, and Action. Three rows are listed: "Test User" (Last Name test, Email test@gmail.com), "test2" (Last Name test2, Email test2@gmail.com), and "testCF" (Last Name testCF, Email test@gmail.com). A red box highlights the entire table. A green box at the top right says "All changes have been saved successfully!". A red arrow points from the text above to the "Display Name" column of the first row.

9. [URL04_SCREENSHOT] Evidence of healthy back-end application. ✓

http://18.236.77.136:3030/api/status

GET http://18.236.77.136:3030/api/status

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Query Params	Key	Value	Description	...	Bulk Edit
			Description		

Body

Pretty Raw Preview Visualize JSON

```

1 "status": "ok",
2 "version": "1.0.0",
3 "environment": "local"
4
5

```

200 OK 15.74 s 281 B Save Response

Healthy Backend

For Point 7 and Point 8, please make sure you do one of the below mentioned 2 things:

- 1. You add the employees using the application (using both S3 link and CloudFront link) and take the screenshot for it, so that we know that the application was functional. This is the preferred option
- 2. If we don't find the screenshots for step 1 in your submission, then we will try to access the website and add the employees.

Note: It may happen that your backend is restarted before your review. In this case the reviewer will fail these 2 points because we will not be able to verify that application was functional. Therefore, option 1 is preferred over option 2.

Section 3: Turn Errors into Sirens

Evidence of Prometheus Server. [URL05_SCREENSHOT].

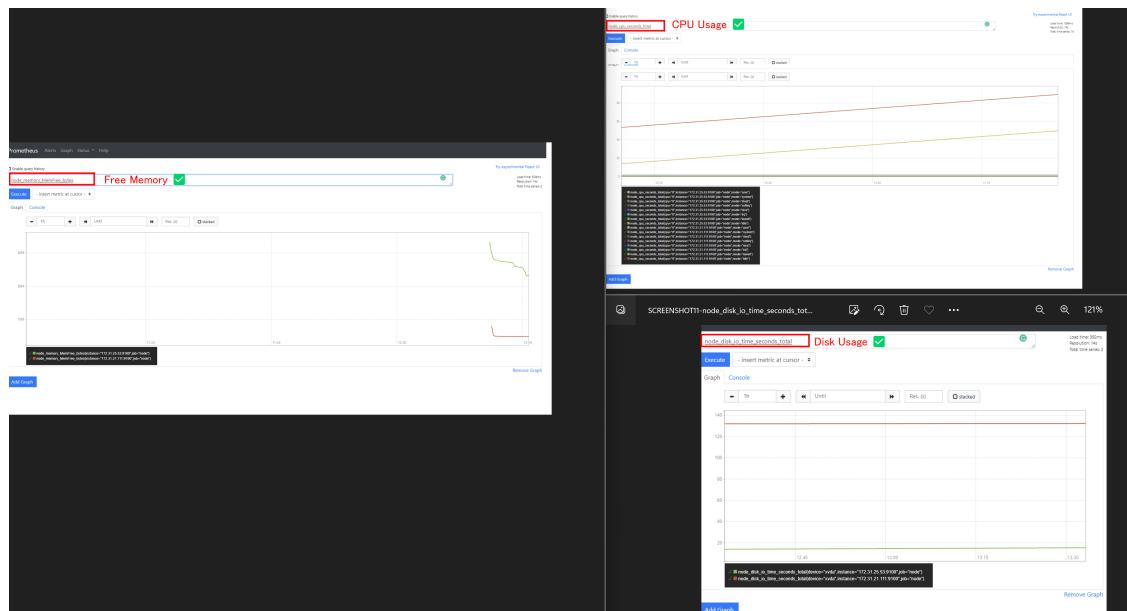
Evidence that Prometheus is monitoring memory, cpu and disk usage of EC2 instances. [SCREENSHOT11]

Evidence that Prometheus and AlertManager send alerts when certain conditions exist in the EC2 instance. [SCREENSHOT12]

1. [URL05_SCREENSHOT] - Evidence of Prometheus Server.

The screenshot shows the Prometheus Targets page. At the top, there are tabs for 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below the tabs, a section titled 'Targets' is displayed. Under 'Targets', there are two sections: 'node (2/2 up)' and 'prometheus (1/1 up)'. Each section has a 'show less' link. The 'node' section contains two entries, both of which are highlighted with a red box. The first entry is 'http://172.31.21.111:9100/metrics' with state 'UP', labels 'instance="172.31.21.111:9100"', and job 'node'. The second entry is 'http://172.31.25.53:9100/metrics' with state 'UP', labels 'instance="172.31.25.53:9100"', and job 'node'. The 'prometheus' section contains one entry, 'http://localhost:9090/metrics', with state 'UP', labels 'instance="localhost:9090"', and job 'prometheus'.

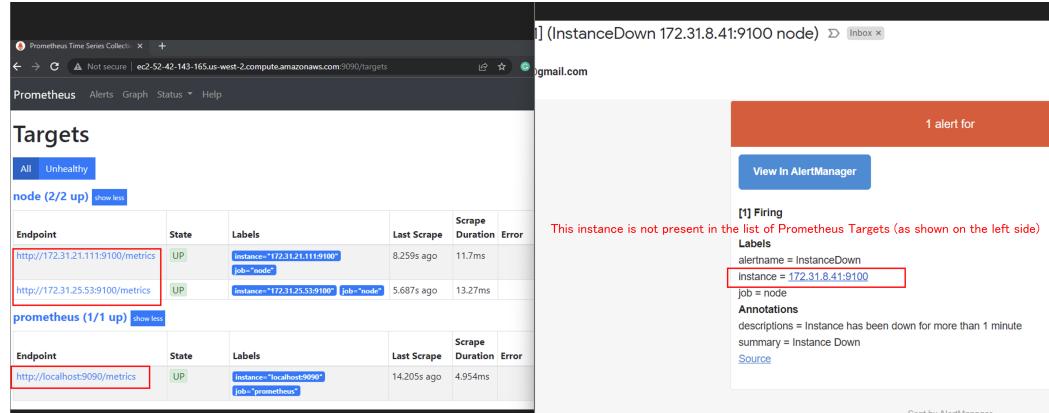
2. [SCREENSHOT11] - Evidence that Prometheus is monitoring memory, CPU and disk usage of EC2 instances. ✓



3. [SCREENSHOT12] - Evidence that Prometheus and Alert Manager send alerts when certain conditions exist in the EC2 instance. ✗

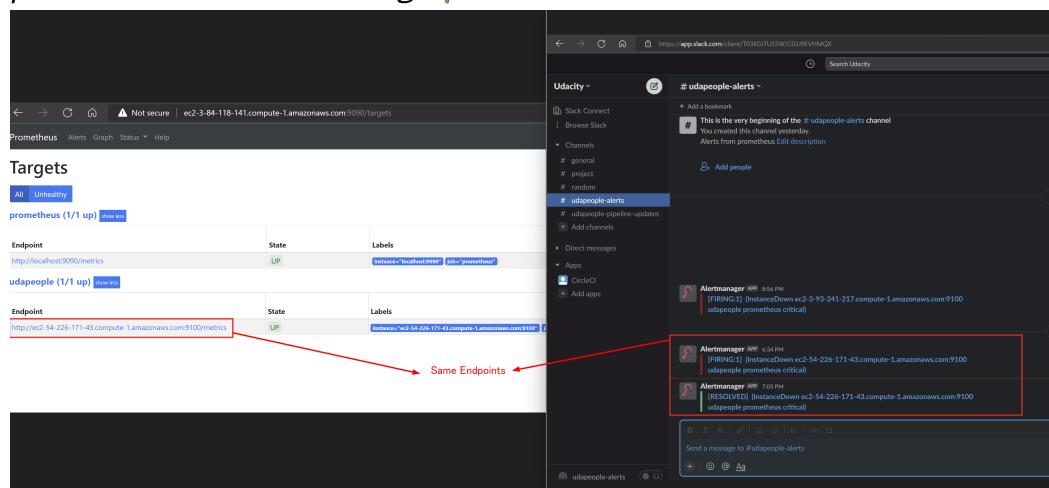
- Great work setting up the AlertManager with Prometheus. It's good to see that it is working 👏
- One small thing that you need to make sure is that
Target in Alert Message should be the one that Prometheus is monitoring .

As of now, I see that both are different 



The screenshot shows two side-by-side windows. On the left is the Prometheus 'Targets' page with two healthy instances: 'http://172.31.21.111:9100/metrics' and 'http://172.31.25.53:9100/metrics'. On the right is a Slack message from 'InstanceDown 172.31.8.41:9100 node' to the '#udapeople-alerts' channel. The message states: 'This instance is not present in the list of Prometheus Targets (as shown on the left side)'. It includes details like 'alertname = InstanceDown', 'Instance = 172.31.8.41:9100', 'job = node', and 'Annotations: summary = Instance Down'.

Please ensure that the Instance in the alert message is amongst the instances that Prometheus is monitoring. For example., if you look at the screenshot shared below, it shows that the DNS name in the Alert received on Slack is same as the DNS endpoint that prometheus is monitoring 



This screenshot illustrates a comparison between Prometheus monitoring and Slack alerts. On the left, the Prometheus 'Targets' page shows one healthy instance: 'http://localhost:9090/metrics'. On the right, a Slack message from 'Alertmanager' in the '#udapeople-alerts' channel indicates an alert for 'ec2-54-226-171-43.compute-1.amazonaws.com:9100'. A red arrow points from the highlighted endpoint in the Prometheus screenshot to the highlighted endpoint in the Slack message, with the text 'Same Endpoints' written below the arrow.

RESUBMIT

 DOWNLOAD PROJECT

1 CODE REVIEW COMMENTS 



Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

[⌚ Watch Video \(3:01\)](#)

[RETURN TO PATH](#)

[Rate this review](#)

[START](#)