

## **FSE\_C5\_Cohort1\_Assignment**

### **Objective:**

To understand and implement CICD pipeline using Jenkins.

Implement CICD pipeline for Front End and Back End Code Base

### **Group Assignment Group 9**

SHIVA KRISHNA K

AVIJIT NANDI

SACHIN GAURAV SRIVASTAVA

## ASSIGNMENT - CICD IMPLEMENTATION (PIPELINE)

### Description

- As part of this assignment, applying CICD on the project which was made as part of Assignment 2 (Microservices based application with at least 3 microservices) in the Cloud Native Development Course.
- Frontend is made on vanilla JS(static HTML files), and backend on NodeJS that includes three microservices each running on different ports. The Front end allows us to select food items to order while the back end has three services – Products,CartItems and Orders.
- As the frontend includes static html pages, so no monitoring is happening.
- For monitoring backend, one monitoring dashboard shows metrics and logs for all three microservices by using PW2 library.
- Execution instructions/configs and deployment script can be seen in the snapshots.

### **GitHub Code Links:**

<https://github.com/sachingaurav/frontend>  
<https://github.com/sachingaurav/backend-products>  
<https://github.com/sachingaurav/backend-cartItems>  
<https://github.com/sachingaurav/backend-orders>

### **CICD for Front-End Code BASE**

Steps:

1. Go to Jenkins and click on New Item.
2. Give item name and select project as Freestyle Project.
3. In Source Code Management, select Git and provide your github repository.
4. In build environment, select the Prepare SonarQube Scanner environment in order to integrate with Sonar scanner.
5. Select Provide Node & npm bin/ folder to PATH in order to run tests.
6. In build steps, under execute shell, run the following statements:  
npm install  
npm test
7. Next, add Execute SonarQube Scanner and add additional arguments.
8. Execute shell in order to run selenium test.
9. Again, select execute shell option in order to transfer the artifacts to local system which is staging env in our case.
10. Again, select execute shell option in order to transfer the artifacts to AWS S3 bucket.

## Jenkins UI of Frontend

Screenshot of the Jenkins UI for the 'frontend' project.

The top navigation bar includes the Jenkins logo, a search bar with placeholder "Search (⌘+K)", and user status indicators (2 notifications, 2 failed builds, admin logged in).

The left sidebar contains links: Status, Changes, Workspace, Build Now, Configure, Delete Project, Git Polling Log, SonarQube, and Rename.

The main content area shows the project name "Project frontend". It features a "SonarQube" icon with a blue checkmark and the text "SonarQube Last Successful Artifacts". A "SonarQube Quality Gate" section indicates "frontend Passed" and "server-side processing: Success".

The "Build History" section lists recent builds:

#	Build Number	Date	Status
64	#64	29 Jun 2023, 12:18	Passed
63	#63	29 Jun 2023, 12:07	Failed
62	#62	29 Jun 2023, 10:08	Passed
61	#61	29 Jun 2023, 10:05	Passed
60	#60	29 Jun 2023, 10:04	Failed
59	#59	28 Jun 2023, 22:40	Passed
58	#58	28 Jun 2023, 22:37	Failed
57	#57	28 Jun 2023, 22:34	Passed

## Configs - GitHub repository integration

Configure

Source Code Management

None

Git [?](#)

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Repositories [?](#)

Repository URL [?](#)

https://github.com/sachingaurav/frontend.git [X](#)

Credentials [?](#)

sachingaurav@gmail.com/\*\*\*\*\* [▼](#)

Add [▼](#)

Advanced [▼](#)

Add Repository

Branches to build [?](#)

Branch Specifier (blank for 'any') [?](#)

\*/main [X](#)

Add Branch

[Save](#) [Apply](#)

The screenshot shows a configuration interface for integrating a GitHub repository. The 'Source Code Management' section is set to 'Git'. The 'Repository URL' is set to 'https://github.com/sachingaurav/frontend.git'. The 'Credentials' field contains 'sachingaurav@gmail.com/\*\*\*\*\*'. Under 'Branches to build', the 'Branch Specifier' is set to '\*/main'. At the bottom, there are 'Save' and 'Apply' buttons.

## Configs – SonarQube and NodeJS integration

Dashboard > frontend > Configuration

### Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Send files or execute commands over SSH after the build runs [?](#)

Add timestamps to the Console Output

Execute shell script on remote host using ssh [?](#)

Prepare SonarQube Scanner environment [?](#)

#### Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled. Will default to the one defined in the SonarQube installation.

- none -

Add ▾

Provide Node & npm bin/ folder to PATH

#### NodeJS Installation

Specify needed nodejs installation where npm installed packages will be provided to the PATH

nodejs

#### npmrc file

- use system default -

#### Cache location

Default (~/.npm or %APP\_DATA%\npm-cache)

Terminate a build if it's stuck

With Ant [?](#)

Save

Apply

## Configs – Installing dependencies and running tests

Dashboard > frontend > Configuration

### Build Steps

## Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

#### Execute shell ?

Command

See the list of available environment variables

```
npm install
```

Advanced ▾

#### Execute shell ?

Command

See the list of available environment variables

```
npm test
```

Save

Apply

## Configs – SonarQube Scanner configs

Dashboard > frontend > Configuration

### Configure

-  General
-  Source Code Management
-  Build Triggers
-  Build Environment
-  **Build Steps**
-  Post-build Actions

Advanced ▾

**Execute SonarQube Scanner**

Task to run ?

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties ?

./

Analysis properties ?

-Dsonar.projectKey=frontend -Dsonar.exclusions=./node\_modules/\*\*/\*,test.js,products.html,cart.html,selenium-test.py

Additional arguments ?

JVM Options ?

Save Apply

## Configs – Running functional tests using selenium and delivering artefacts to staging environment ( local )

Dashboard > frontend > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions**

**Execute shell** ?

Command

See [the list of available environment variables](#)

```
python3 -m pip install selenium
python3 selenium-test.py
```

Advanced ▾

**Execute shell** ?

Command

See [the list of available environment variables](#)

```
cp ./*.html ~/frontend_output/
```

Advanced ▾

Save

Apply

## Configs - Code deliver to production environment(AWS S3)

Dashboard > frontend > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

≡ Publish artifacts to S3 Bucket ?

S3 profile

assignment-frontend

Files to upload

Source ?

\*.html

Exclude

Destination bucket ?

assignment-frontend

Storage class

STANDARD

Bucket Region ?

ap-southeast-1

No upload on build failure ?

Publish from Slave ?

**Save**   **Apply**

## Build output -Fetching code from GitHub, Installing dependencies

Jenkins

Dashboard > frontend > #64 > Console Output

Status Changes Console Output View as plain text Edit Build Information Delete build '#64' Git Build Data Previous Build

### Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /Users/s0s0noj/.jenkins/workspace/frontend
The recommended git tool is: NONE
using credential c7745485-6162-4612-94e3-02e172039844
> git rev-parse --resolve-git-dir /Users/s0s0noj/.jenkins/workspace/frontend/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/sachingaurav/frontend.git # timeout=10
Fetching upstream changes from https://github.com/sachingaurav/frontend.git
> git --version # timeout=10
> git --version # 'git version 2.39.2 (Apple Git-143)'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/sachingaurav/frontend.git +refs/heads/*:refs/remotes/origin/*
# timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 7e841de82d4facb723e441d634c32b9b71fc4198 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 7e841de82d4facb723e441d634c32b9b71fc4198 # timeout=10
Commit message: "Create sonar-project.properties"
First time build. Skipping changelog.
Injecting SonarQube environment variables using the configuration: server-sonar
[frontend] $ /bin/sh -xe /var/folders/01/_2nq8f0j251_w_g9fnyfnjv40000gg/T/jenkins12646224882033244627.sh
+ npm install

up to date, audited 130 packages in 1s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
[frontend] $ /bin/sh -xe /var/folders/01/_2nq8f0j251_w_g9fnyfnjv40000gg/T/jenkins8620569833572847766.sh
```

## Build output – Running unit tests

```
+ npm test

> test
> mocha test.js --exit || exit 1


Orders Page
  ✓ should display the "Products" button

[frontend] $
/Users/s0s0noj/.jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonarqubescanner/b
in/sonar-scanner -Dsonar.projectKey=frontend -
Dsonar.exclusions=./node_modules/**/*.* ,test.js,products.html,cart.html,selenium-test.py -
Dsonar.host.url=http://localhost:9000 ***** -
Dproject.settings=/Users/s0s0noj/.jenkins/workspace/frontend -
Dsonar.projectBaseDir=/Users/s0s0noj/.jenkins/workspace/frontend
INFO: Scanner configuration file:
/Users/s0s0noj/.jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonarqubescanner/c
onf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.8.0.2856
INFO: Java 19.0.1 Oracle Corporation (64-bit)
INFO: Mac OS X 13.4 x86_64
INFO: User cache: /Users/s0s0noj/.sonar/cache
INFO: Analyzing on SonarQube server 10.0.0.68432
INFO: Default locale: "en_GB", source code encoding: "UTF-8" (analysis is platform dependent)
INFO: Load global settings
INFO: Load global settings (done) | time=188ms
INFO: Server id: 147B411E-AYj-J1CwVZ7NFXS4nXYZ
INFO: User cache: /Users/s0s0noj/.sonar/cache
TINFO: Load/download plugins
```

## **Build output – Generating sonar report and running functional tests**

Dashboard > frontend > #64 > Console Output

```
INFO: Analysis report generated in 142ms, dir size=131.5 kB
INFO: Analysis report compressed in 151ms, zip size=19.7 kB
INFO: Analysis report uploaded in 55ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=frontend
INFO: Note that you will be able to access the updated dashboard once the server has
processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AYkF6jSudI7bkBHX3RM8
INFO: Analysis total time: 1:24.861 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 1:28.099s
INFO: Final Memory: 21M/80M
INFO: -----
[frontend] $ /bin/sh -xe
/var/folders/01/_2ng8f0j251_w_q9fnyfnjv40000qq/T/jenkins1748833843373965924.sh
+ python3 -m pip install selenium
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: selenium in /Users/s0s0noj/Library/Python/3.9/lib/python/site-
packages (4.10.0)
Requirement already satisfied: certifi>=2021.10.8 in
/Users/s0s0noj/Library/Python/3.9/lib/python/site-packages (from selenium) (2023.5.7)
Requirement already satisfied: trio~=0.17 in
/Users/s0s0noj/Library/Python/3.9/lib/python/site-packages (from selenium) (0.22.0)
Requirement already satisfied: trio-websocket~=0.9 in
/Users/s0s0noj/Library/Python/3.9/lib/python/site-packages (from selenium) (0.10.3)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in
/Users/s0s0noj/Library/Python/3.9/lib/python/site-packages (from selenium) (2.0.3)
Requirement already satisfied: outcome in /Users/s0s0noj/Library/Python/3.9/lib/python/site-
packages (from trio~=0.17 & trio-socketium) (1.2.0)
```

## Build output – Publishing artifacts to AWS S3

Dashboard > frontend > #64 > Console Output

```
Requirement already satisfied: wsproto>=0.14 in
/Users/s0s0noj/Library/Python/3.9/lib/python/site-packages (from trio-websocket~=0.9-
>selenium) (1.2.0)
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in
/Users/s0s0noj/Library/Python/3.9/lib/python/site-packages (from urllib3[socks]<3,>=1.26-
>selenium) (1.7.1)
Requirement already satisfied: h11<1,>=0.9.0 in
/Users/s0s0noj/Library/Python/3.9/lib/python/site-packages (from wsproto>=0.14->trio-
websocket~=0.9->selenium) (0.14.0)
WARNING: You are using pip version 21.2.4; however, version 23.1.2 is available.
You should consider upgrading via the '/Library/Developer/CommandLineTools/usr/bin/python3 -m
pip install --upgrade pip' command.
+ python3 selenium-test.py
/Users/s0s0noj/Library/Python/3.9/lib/python/site-packages/urllib3/__init__.py:34:
NotOpenSSLWarning: urllib3 v2.0 only supports OpenSSL 1.1.1+, currently the 'ssl' module is
compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
    warnings.warn(
[frontend] $ /bin/sh -xe
/var/folders/01/_2nq8f0j251_w_q9fnyfnjv40000gq/T/jenkins12047614145000466483.sh
+ cp ./cart.html ./order.html ./products.html /Users/s0s0noj/frontend_output/
Archiving artifacts
Publish artifacts to S3 Bucket Build is still running
Publish artifacts to S3 Bucket Using S3 profile: assignment-frontend
Publish artifacts to S3 Bucket bucket=assignment-frontend, file=cart.html region=ap-
southeast-1, will be uploaded from slave=false managed=false , server encryption false
Publish artifacts to S3 Bucket bucket=assignment-frontend, file=order.html region=ap-
southeast-1, will be uploaded from slave=false managed=false , server encryption false
Publish artifacts to S3 Bucket bucket=assignment-frontend, file=products.html region=ap-
southeast-1, will be uploaded from slave=false managed=false , server encryption false
Finished: SUCCESS
```

## AWS S3 bucket containing frontend artefacts

The screenshot shows the AWS S3 console interface. On the left, a sidebar titled "Amazon S3" contains sections for Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, AWS Organizations settings, Feature spotlight (3), and AWS Marketplace for S3. The main content area shows the "assignment-frontend" bucket under "Buckets". The bucket is "Publicly accessible". The "Objects" tab is selected, showing three HTML files: cart.html, order.html, and products.html, all uploaded on June 29, 2023, at 12:20:25 UTC+05:30, with sizes of 3.7 KB, 3.4 KB, and 5.5 KB respectively, and stored in Standard storage class. Below the object list, a progress bar indicates 0.0% completion for a task, with a note about 21 new lines.

Amazon S3 < Services Search [Option+S] Global sachingaurav

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight (3)

AWS Marketplace for S3

Amazon S3 > Buckets > assignment-frontend

assignment-frontend [Info](#)

Publicly accessible

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	cart.html	html	June 29, 2023, 12:20:25 (UTC+05:30)	3.7 KB	Standard
<input type="checkbox"/>	order.html	html	June 29, 2023, 12:20:25 (UTC+05:30)	3.4 KB	Standard
<input type="checkbox"/>	products.html	html	June 29, 2023, 12:20:25 (UTC+05:30)	5.5 KB	Standard

0.0%

Duplications on 21 New Lines

## Homepage of frontend

The screenshot shows a web browser window with the URL [assignment-frontend.s3.ap-southeast-1.amazonaws.com/products.html](http://assignment-frontend.s3.ap-southeast-1.amazonaws.com/products.html). The page title is "Products". Below the title is a table with four columns: "ID", "Name", "Price", and "Description". Each row in the table includes a "Count" column with a minus sign, a zero, and a plus sign. A green "Add to Cart" button is located at the bottom left of the table.

ID	Name	Price	Description	Count
103	Burger	89	Veg Crispy Burger	- 0 +
104	Onion Pizza	159	Onion and tomato cheesy pizza	- 0 +
101	Pav Bhaji	99	Pav Bhaji with Butter	- 0 +
102	Panner Paratha	119	Panner Stuffed Paratha	- 0 +

Add to Cart

### Frontend page showing items in the cart

Products

## Cart Items

ID	Name	Price	Description	Count
104	Onion Pizza	159	Onion and tomato cheesy pizza	2

Total Price: 318

**Confirm My Order**

### Frontend page showing order history

Products

## Orders

Order Date: 18/05/2023, 13:42:58

ID	Name	Price	Description	Count
101	Pav Bhaji	99	Pav Bhaji with Butter	1
102	Panner Paratha	119	Panner Stuffed Paratha	3

Total Price: 456

Order Date: 19/05/2023, 00:11:11

ID	Name	Price	Description	Count
104	Onion Pizza	159	Onion and tomato cheesy pizza	1
101	Pav Bhaji	99	Pav Bhaji with Butter	2
102	Panner Paratha	119	Panner Stuffed Paratha	1

Total Price: 476

### CICD for Backend Code BASE

Steps:

1. Open Jenkins and click on New Item.
2. Give item name and select project as Freestyle Project.
3. In Source Code Management, select Git and provide your github repository.
4. In build environment, select the Prepare SonarQube Scanner environment in order to integrate with Sonar scanner.
5. Select Provide Node & npm bin/ folder to PATH in order to run tests.
6. In build steps, under execute shell as given in the snapshots to install the dependencies and running tests.
7. Next, using Send files or execute commands over SSH option, transfer the artifacts to AWS EC2.
8. Add Execute shell script on remote host using ssh and write the deployment scripts in order to deploy to EC2.

## Jenkins UI of backend

The screenshot shows the Jenkins interface for the project 'backend-products'. The top navigation bar includes a logo, a search bar with placeholder text 'Search (⌘+K)', a help icon, a notifications icon with '2' notifications, a shield icon with '2' alerts, and a user account for 'admin' with a log out option.

The main header 'Project backend-products' is displayed above a sidebar on the left. The sidebar contains links for 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. It also features a 'Build History' section with a 'trend' dropdown set to 'trend' and a 'Filter builds...' input field. The build history lists builds #44 through #39, each with a timestamp: #44 (29 Jun 2023, 12:25), #43 (29 Jun 2023, 12:25), #42 (28 Jun 2023, 19:40), #41 (28 Jun 2023, 19:25), #40 (28 Jun 2023, 19:15), and #39 (28 Jun 2023, 18:21). Build #39 is marked with a crossed-out icon.

The right side of the screen displays the 'Permalinks' section, which lists the last seven builds with their respective build numbers and timestamps:

- Last build (#44), 1 day 9 hr ago
- Last stable build (#44), 1 day 9 hr ago
- Last successful build (#44), 1 day 9 hr ago
- Last failed build (#37), 2 days 3 hr ago
- Last unstable build (#31), 3 days 0 hr ago
- Last unsuccessful build (#39), 2 days 3 hr ago
- Last completed build (#44), 1 day 9 hr ago

On the far right, there are buttons for 'Add description' and 'Disable Project'.

## Build configs – Fetching code from GitHub repository

### Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Git ?

Repositories ?

Repository URL ? X

Credentials ? X

▼

Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ? X

Save Apply

## Build configs – NodeJS integration

### Configure

 General

 Source Code Management

 Build Triggers

 Build Environment

 Build Steps

 Post-build Actions

Send files or execute commands over SSH before the build starts [?](#)

Send files or execute commands over SSH after the build runs [?](#)

Add timestamps to the Console Output

Execute shell script on remote host using ssh [?](#)

Prepare SonarQube Scanner environment [?](#)

Provide Node & npm bin/ folder to PATH

#### NodeJS Installation

Specify needed nodejs installation where npm installed packages will be provided to the PATH

nodejs

#### npmrc file

- use system default -

#### Cache location

Default (~/.npm or %APP\_DATA%\npm-cache)

Terminate a build if it's stuck

With Ant [?](#)

Save

Apply

## Build configs – Installing dependencies and running tests

### Configure

#### Build Steps

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

##### Execute shell ?

###### Command

See [the list of available environment variables](#)

```
npm install
```

Advanced ▾

##### Execute shell ?

###### Command

See [the list of available environment variables](#)

```
npm test
```

Save

Apply

## Build configs - Delivering artefacts to AWS EC2

Dashboard > backend-products > Configuration

### Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

#### Send files or execute commands over SSH ?

##### SSH Publishers

###### SSH Server

Name ?

ec2-13-49-244-239 [No Exec]

Advanced ▾

##### Transfers

###### Transfer Set

Source files ?

app.js package.json

Remove prefix ?

/

Remote directory ?

/products

Save

Apply

## Build configs – Deployment scripts

### Configure

 General

 Source Code Management

 Build Triggers

 Build Environment

 Build Steps

 Post-build Actions

#### ≡ Execute shell script on remote host using ssh

##### SSH site

ubuntu@ec2-13-49-244-239.eu-north-1.compute.amazonaws.com:22

##### Command

```
pwd  
cd /home/ubuntu/products  
npm install  
sudo npm install -g pm2  
pm2 stop products  
pm2 start app.js --name "products"
```

Execute each line ?

Hide command from console output

Add build step ▾

Save

Apply

## Build output – Fetching code from GitHub repository

Dashboard > backend-products > #44 > Console Output

 Delete build '#44'

 Git Build Data

 Previous Build

```
products/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/sachingaurav/backend-products.git #
timeout=10
Fetching upstream changes from https://github.com/sachingaurav/backend-products.git
> git --version # timeout=10
> git --version # 'git version 2.39.2 (Apple Git-143)'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/sachingaurav/backend-
products.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 161ac4e1cf26e96e189c9ad749c497fb4674142d
(refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 161ac4e1cf26e96e189c9ad749c497fb4674142d # timeout=10
Commit message: "init"
> git rev-list --no-walk 161ac4e1cf26e96e189c9ad749c497fb4674142d # timeout=10
[backend-products] $ /bin/sh -xe
/var/folders/01/_2nq8f0j251_w_q9fnynjv40000qq/T/jenkins18102324079393902984.sh
+ npm install

up to date, audited 199 packages in 1s

30 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

## Build output – Running tests and delivering code to AWS EC2

Dashboard > backend-products > #44 > Console Output

```
+ npm test

> test
> mocha test.js --exit || exit 1

Server is running at PORT:3000


MongoDB Connection Test
Connected to MongoDB
  ✓ should connect to MongoDB successfully

SSH: Connecting from host [m-c02g4184md6n]
SSH: Connecting with configuration [ec2-13-49-244-239] ...
SSH: Disconnecting configuration [ec2-13-49-244-239] ...
SSH: Transferred 2 file(s)
Build step 'Send files or execute commands over SSH' changed build result to SUCCESS
[SSH] script:

pwd
cd /home/ubuntu/products
npm install
sudo npm install -g pm2
pm2 stop products
pm2 start app.js --name "products"

[SSH] executing...
```

## Build output – Running deployment scripts and monitoring tool

```
run `npm fund` for details
[PM2] Applying action stopProcessId on app [products](ids: [ 0 ])
[PM2] [products](0) ✓
⚡ PM2+ activated | Instance Name: ip-172-31-46-13-6edf | Dash: https://app.pm2.io/#/r/qqlq10e6vftyv4w
```

id	name	namespace	version	mode	pid	uptime	s	status	cpu	mem	user	watching
2	cartItems	default	N/A	fork	1789	16h	0	online	0%	60.4mb	ubuntu	disabled
1	orders	default	N/A	fork	1629	16h	0	online	0%	56.8mb	ubuntu	disabled
0	products	default	N/A	fork	0	0	0	stopped	0%	0b	ubuntu	disabled

```
[PM2] Applying action restartProcessId on app [products](ids: [ 0 ])
[PM2] [products](0) ✓
[PM2] Process successfully started
⚡ PM2+ activated | Instance Name: ip-172-31-46-13-6edf | Dash: https://app.pm2.io/#/r/qqlq10e6vftyv4w
```

id	name	namespace	version	mode	pid	uptime	s	status	cpu	mem	user	watching
2	cartItems	default	N/A	fork	1789	16h	0	online	0%	60.4mb	ubuntu	disabled
1	orders	default	N/A	fork	1629	16h	0	online	0%	56.8mb	ubuntu	disabled
0	products	default	N/A	fork	10004	0s	0	online	0%	19.5mb	ubuntu	disabled

```
[SSH] completed
[SSH] exit-status: 0
```

```
Finished: SUCCESS
```

## AWS EC2 instance output

Sachin | assign | Data | ec2- | Insta... | ec2- | Con... | EC2 | McA... | EC2 | PM2 | Chai... | SSH | Addi... | githu... | vanill... | Assi... | cent... | + eu-north-1.console.aws.amazon.com/ec2/v2/home?region=eu-north-1#InstanceDetails:instanceId=i-0e050e53111fef7fb Search [Option+S] Stockholm sachingaurav

aws Services Search [Option+S] EC2 Instances i-0e050e53111fef7fb

**Instance summary for i-0e050e53111fef7fb (backend-products)** [Info](#) [Copy](#) [Connect](#) [Instance state](#) [Actions](#)

Updated less than a minute ago

Instance ID <a href="#">i-0e050e53111fef7fb (backend-products)</a>	Public IPv4 address <a href="#">13.49.244.239</a>   <a href="#">open address</a>	Private IPv4 addresses <a href="#">172.31.46.13</a>
IPv6 address -	Instance state <a href="#">Running</a>	Public IPv4 DNS <a href="#">ec2-13-49-244-239.eu-north-1.compute.amazonaws.com</a>   <a href="#">open address</a>
Hostname type IP name: ip-172-31-46-13.eu-north-1.compute.internal	Private IP DNS name (IPv4 only) <a href="#">ip-172-31-46-13.eu-north-1.compute.internal</a>	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t3.micro	AWS Compute Optimizer finding <a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>   <a href="#">Learn more</a>
Auto-assigned IP address <a href="#">13.49.244.239</a> [Public IP]	VPC ID <a href="#">vpc-00099d47a3b0709e3</a>	Auto Scaling Group name -
IAM Role -	Subnet ID <a href="#">subnet-0ca7b2623675d1e01</a>	
IMDSv2 Optional		

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

## PW2 monitoring tool - Showing dashboard of microservices performance

app.pm2.io/bucket/649c1f2e4e0cba42972fbeda/backend/overview/servers

3 / 4 proc. free\_v4 CONNECT + ? ⚙️ 🌐

**PM2**

backer ip-1/2-31-46-13-beat 3 N/A V18.16.1 5.3.0 13.49.244.239 / 172.31.46.13 / ip-1/2-31-46-13 X DELETE SERVER

Glob... App ...

**cartItems**

1 process NODE\_ENV=N/A online for 18 hours

	Active handles	12	Active requests	0	Event Loop Latency	0.3ms	Event Loop Latency p95	1ms
Metrics	Heap Size	28.3MiB	Heap Usage	84%	HTTP	0req/min	HTTP Mean Latency	173ms
	HTTP P95 Latency	1s	Used Heap Size		23.7MiB			
f Metrics	Add your metric ⓘ							

**orders**

1 process NODE\_ENV=N/A online for 18 hours

	Active handles	12	Active requests	0	Event Loop Latency	0.3ms	Event Loop Latency p95	1ms
Metrics	Heap Size	28.3MiB	Heap Usage	82%	HTTP	0req/min	HTTP Mean Latency	395ms
	HTTP P95 Latency	1.8s	Used Heap Size		23.2MiB			
f Metrics	Add your metric ⓘ							

**products**

1 process NODE\_ENV=N/A online for 44 minutes

	Active handles	12	Active requests	0	Event Loop Latency	0.3ms	Event Loop Latency p95	1ms
Metrics	Heap Size	25.3MiB	Heap Usage	92%	HTTP	0req/min	HTTP Mean Latency	257ms
	HTTP P95 Latency	411ms	Used Heap Size		23.3MiB			
f Metrics	Add your metric ⓘ							

## PW2 monitoring tool - Showing logs of microservices

← → C 🔒 app.pm2.io/bucket/649c1f2e4e0cba42972fbeda/backend/overview/servers

### PM2

backer Glob... App ... PM2 ...

cartItems

1 process NODE\_ENV=N/A online for 20 hours

Metrics Active handles Heap Size HTTP P95 Latency

f Metrics Add your metric

Log View

```
15:41:53 ip-172-31-46-13-6edf 2 [ { itemId: '103', count: 0 }, { itemId: '104', count: 2 }, { itemId: '101', count: 0 }, { itemId: '102', count: 0 } ] 15:41:53 ip-172-31-46-13-6edf 2 1 new cart items inserted into MongoDB
```

Searchable Mini Map

orders

1 process NODE\_ENV=N/A online for 20 hours

Metrics Active handles Heap Size HTTP P95 Latency

f Metrics Add your metric

products

1 process NODE\_ENV=N/A online for 3 hours

Metrics Active handles Heap Size HTTP P95 Latency

f Metrics Add your metric

Events

HTTP

g

Events

HTTP

g

Events

HTTP

g

The screenshot shows the PM2 monitoring interface for a microservice named 'cartItems'. The service has been online for 20 hours. The log view displays the following entries:

```
15:41:53 ip-172-31-46-13-6edf 2 [ { itemId: '103', count: 0 }, { itemId: '104', count: 2 }, { itemId: '101', count: 0 }, { itemId: '102', count: 0 } ] 15:41:53 ip-172-31-46-13-6edf 2 1 new cart items inserted into MongoDB
```

The log entries indicate that four new items have been inserted into MongoDB, with item IDs 103, 104, 101, and 102, each having a count of 0 or 2.