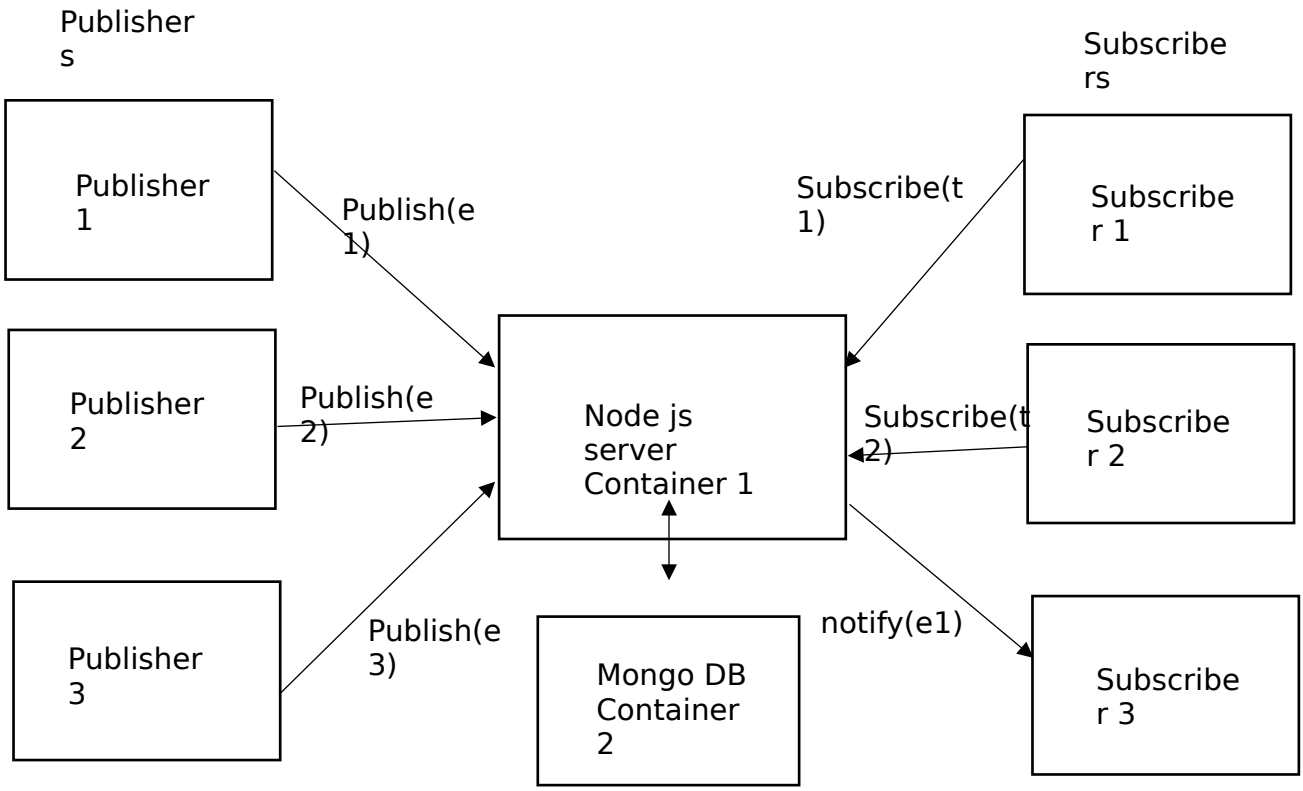
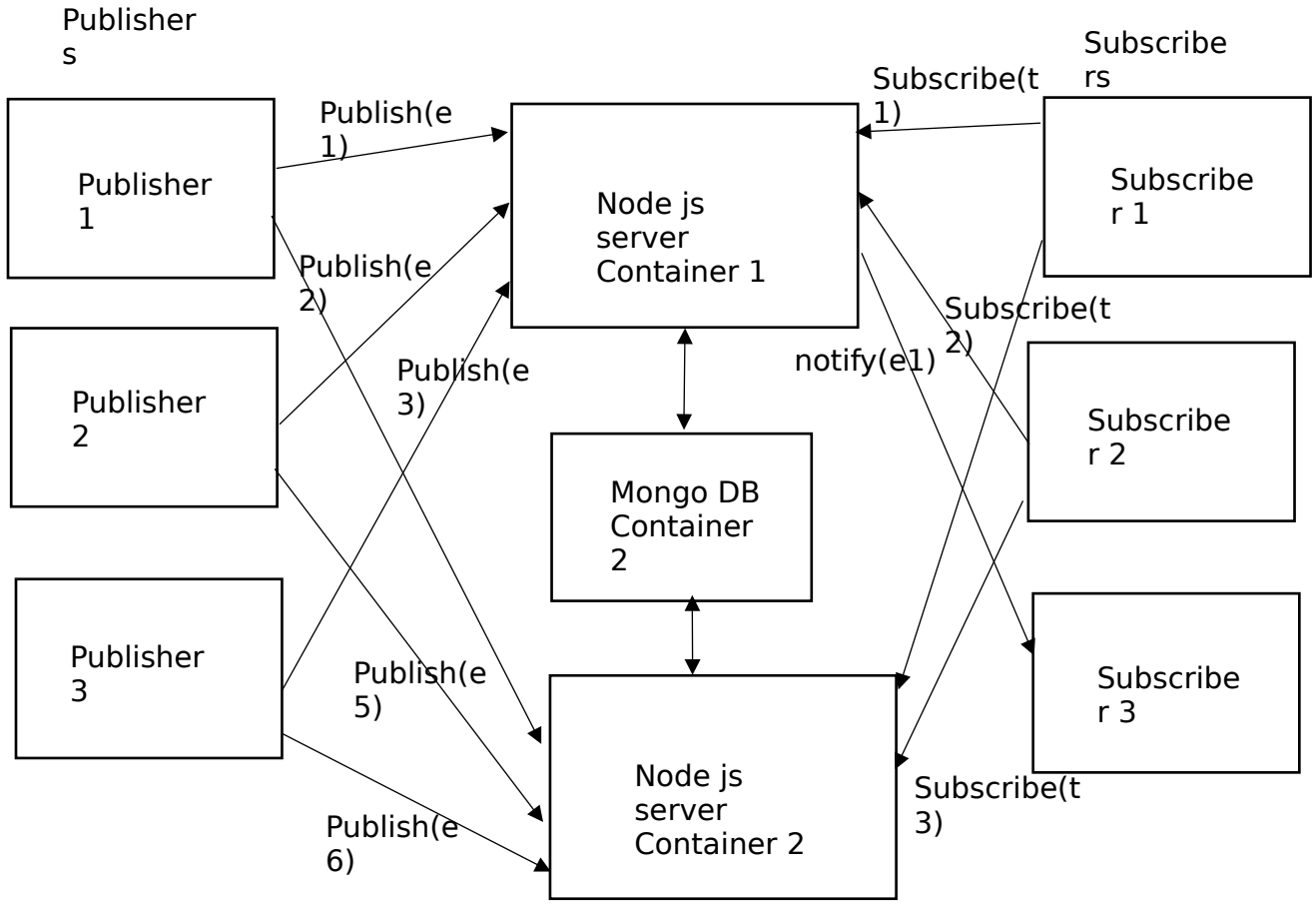

Publisher
s
Subscribe
rs
Publisher
1
Publish(e
1)
Subscribe(t
1)
Subscribe
r 1
Publisher
2
Publish(e
2)
Node js
server
Container 1
Subscribe(t
2)
Subscribe
r 2
Publisher
3
Publish(e
3)
Mongo DB
Container
2
notify(e1)
Subscribe
r 3


Publisher
s
Subscribe
rs
Subscribe(t
1)
Publish(e
1)
Publisher
1
Node js
server
Container 1
Subscribe
r 1
Publish(e
2)
Subscribe(t
2)
Publisher
2
Publish(e
3)
notify(e1)
Subscribe
r 2
Mongo DB
Container
2
Publisher
3
Publish(e
5)
Subscribe
r 3
Node js
server
Container 2
Subscribe(t
3)
Publish(e
6)

Phase 1

-A simple online python compiler is created using docker and node js.

-The docker only has one container and one image. The base image for the docker is node 8 which was downloaded from the docker hub.

-The docker port is mapped to the host system port and accessed on the web browser.

-All the executions such as storing data in python file,executing python file is carried out from inside the container.

Phase 2

-The phase 2 of the project is a simple publisher subscriber model.

- The model is implemented using node js and mongoDB.

-The image of the model is created and is stored inside a docker container.

-Two containers are being used

1) To store the image of node 8 as base image and pub sub system.

2) To store the image of mongoDb which is used as a database.

-docker compose file is used to link the two containers inside the docker.

-There are two servers present in the pub sub system.

1) To allow the publishers to publish the article based on their ID and the topic on which they want to publish their article.

2)To allow the subscribers to subscribe on particular topic using subscribe button and view the articles based on their subscriptions using the view subscriptions button.

-There can be multiple publishers and subscribers present on the nodes.

-All the published articles,subscribers ID,publishers ID,Topics are stored in the mongoDB database.

Phase 3

-The phase 3 of the project is a distributed implementation of the publisher subscriber model implemented in the phase 2.

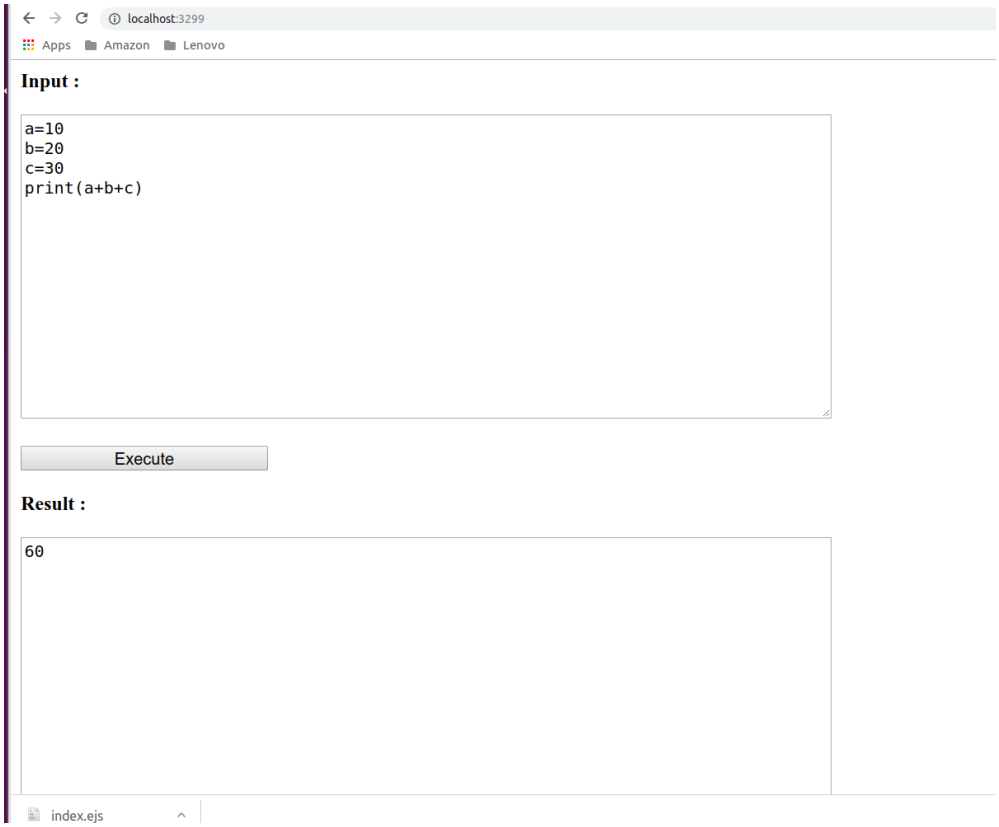
- In phase 3 instead of only one container containing publishers and subscribers there are 2 containers having different publishers and subscribers. There is also a third container which has the mongo image which is used for database and linked with both the other containers.

-In phase 3 if the article is published by a publisher in one container then it gets updated in all the containers ,similarly subscribers in all the containers are notified simultaneously about the newly published article.

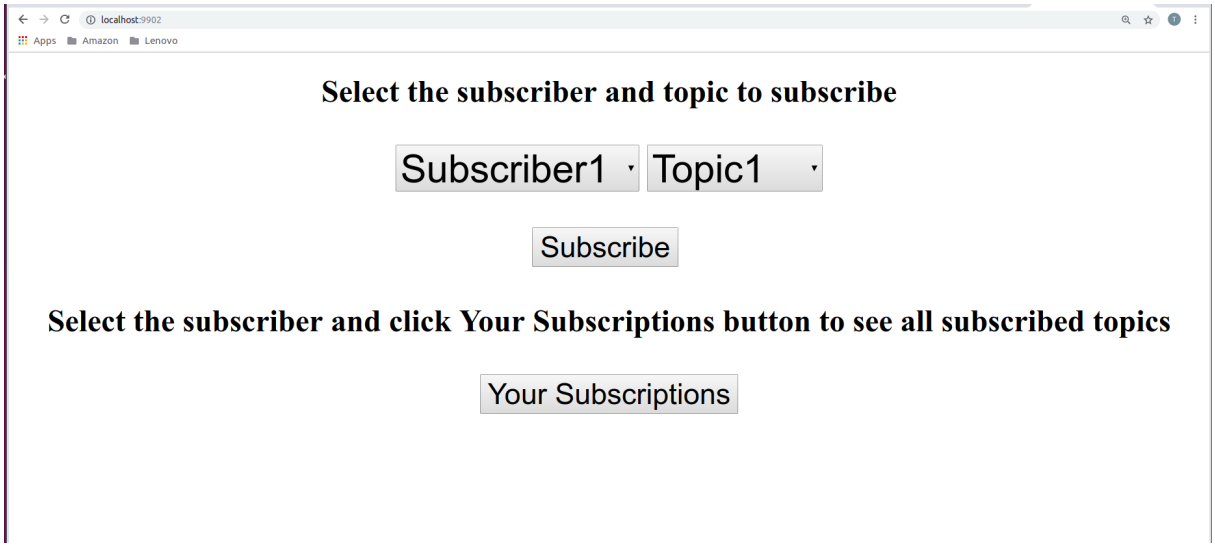
- The docker compose file is used to specify the ports,links,network and container names.

Outputs

Phase 1



Phase 2


localhost:9902
Apps
Amazon
Lenovo
Select the subscriber and topic to subscribe
Subscriber1
Topic1
Subscribe
Select the subscriber and click Your Subscriptions button to see all subscribed topics
Your Subscriptions

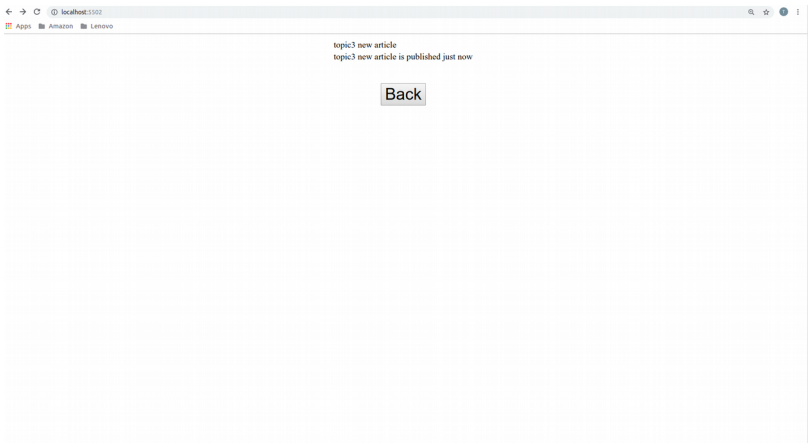

10.5.0.6:9900
Apps
Amazon
Lenovo
Select the publisher and today's topic
Publisher1
Topic4
hi this is new article
Submit Article


localhost:5502
Apps
Amazon
Lenovo
topic3 new article
topic3 new article is published just now
Back

Phase 3

node 1

node 2


localhost:5502
Apps
Amazon
Lenovo
topic3 new article
topic3 new article is published just now
Back