Phase 5

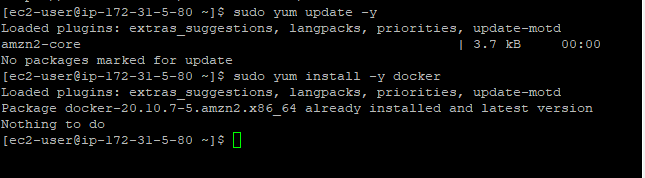
Lesson 5 project

Deploying ELK Stack on Docker Container

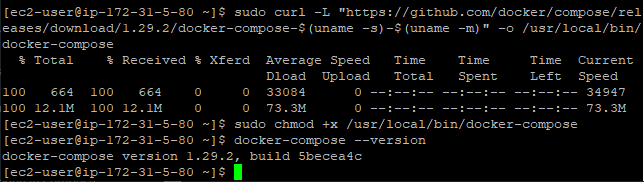
Steps:

Step 1:

* To deploy the ELK stack on the docker container
* First install the docker



* The install the docker-compose inside the docker machine

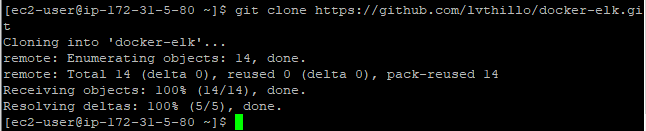


Step 2: create a **docker-compose.yml** and add this code

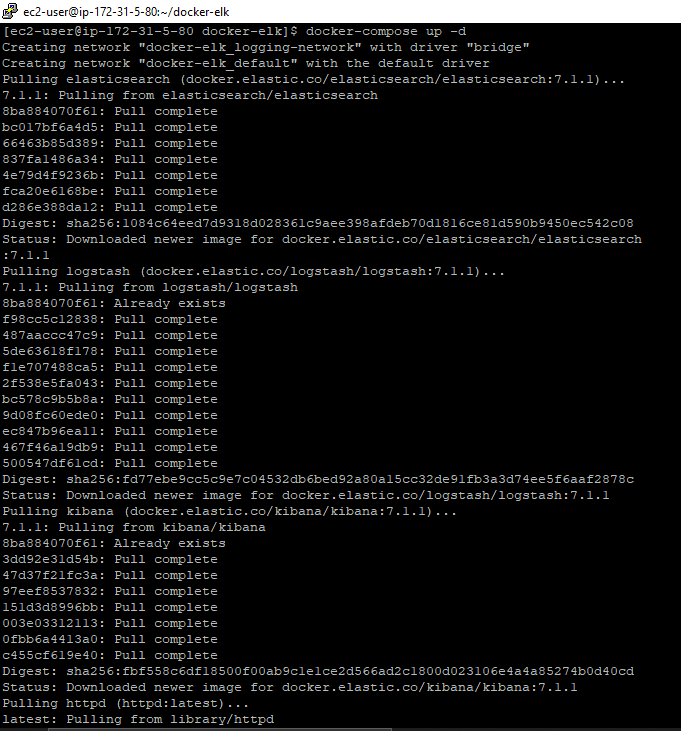
|  |
| --- |
| version: "3" |
|  |  |
|  | services: |
|  | elasticsearch: |
|  | image: docker.elastic.co/elasticsearch/elasticsearch:7.1.1 |
|  | environment: |
|  | - discovery.type=single-node |
|  | - xpack.security.enabled=false |
|  | networks: |
|  | - logging-network |
|  |  |
|  | logstash: |
|  | image: docker.elastic.co/logstash/logstash:7.1.1 |
|  | depends\_on: |
|  | - elasticsearch |
|  | ports: |
|  | - 12201:12201/udp |
|  | volumes: |
|  | - ./logstash.conf:/usr/share/logstash/pipeline/logstash.conf:ro |
|  | networks: |
|  | - logging-network |
|  |  |
|  | kibana: |
|  | image: docker.elastic.co/kibana/kibana:7.1.1 |
|  | depends\_on: |
|  | - logstash |
|  | ports: |
|  | - 5601:5601 |
|  | networks: |
|  | - logging-network |
|  |  |
|  | httpd: |
|  | image: httpd:latest |
|  | depends\_on: |
|  | - logstash |
|  | ports: |
|  | - 80:80 |
|  | logging: |
|  | driver: gelf |
|  | options: |
|  |  |
|  |  |
|  |  |
|  | gelf-address: "udp://localhost:12201" |
|  |  |
|  | networks: |
|  | logging-network: |
| Step3: | driver: bridge  Create a **logstash.conf** file and add this code  Input {  gelf {}  }  Output {  elasticsearch {  hosts => “elasticsearch:9200”  }  } |
|  |  |
|  |  |

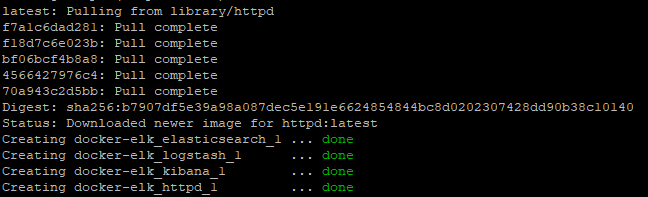
Step 4: push that files to the github

* Clone that github file using
* git clone <https://github.com/lvthillo/docker-elk.git>



* do cd docker-elk
* do docker-compose up –d to deploy the ELK stack to docker container

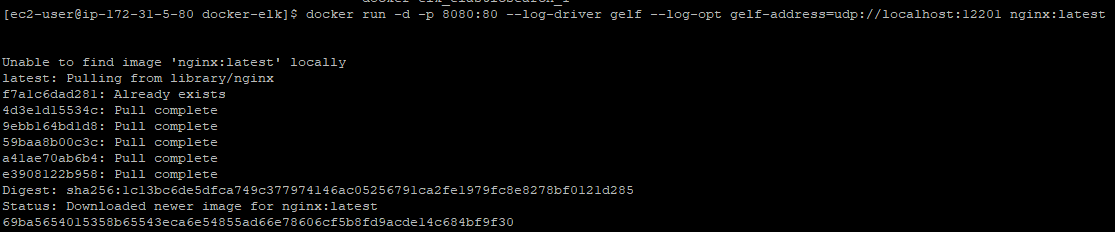




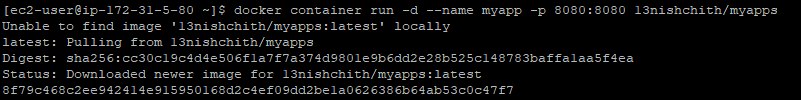


Step 4:  additional Nginx container using the Docker CLI

* docker run -d -p 8080:80 --log-driver gelf --log-opt gelf-address=udp://localhost:12201 nginx:latest



* docker container run -d --name myapp -p 8080:8080 13nishchith/myapps



* Browse your localhost ip address to see output