

DOCKER

Docker is a containerization platform that packages your application and all its dependencies together in the form of Containers to ensure that your application works seamlessly in any environment.

Benefits of Docker

Now, the QA team need not install all the dependent software and applications to test the code and this helps them save lots of time and energy. This also ensures that the working environment is consistent across all the individuals involved in the process, starting from development to deployment. The number of systems can be scaled up easily and the code can be deployed on them effortlessly.

CONTAINERIZATION

Containerization is the technique of bringing virtualization to the operating system level. While Virtualization brings abstraction to the hardware, Containerization brings abstraction to the operating system. Do note that Containerization is also a type of Virtualization. Containerization is however more efficient because there is no guest OS here and utilizes a host's operating system, share relevant libraries & resources as and when needed unlike virtual machines. Application specific binaries and libraries of containers run on the host kernel, which makes processing and execution very fast. Even booting-up a container takes only a fraction of a second. Because all the containers share, host operating system and holds only the application related binaries & libraries. They are lightweight and faster than Virtual Machines.

Advantages of Containerization over Virtualization:

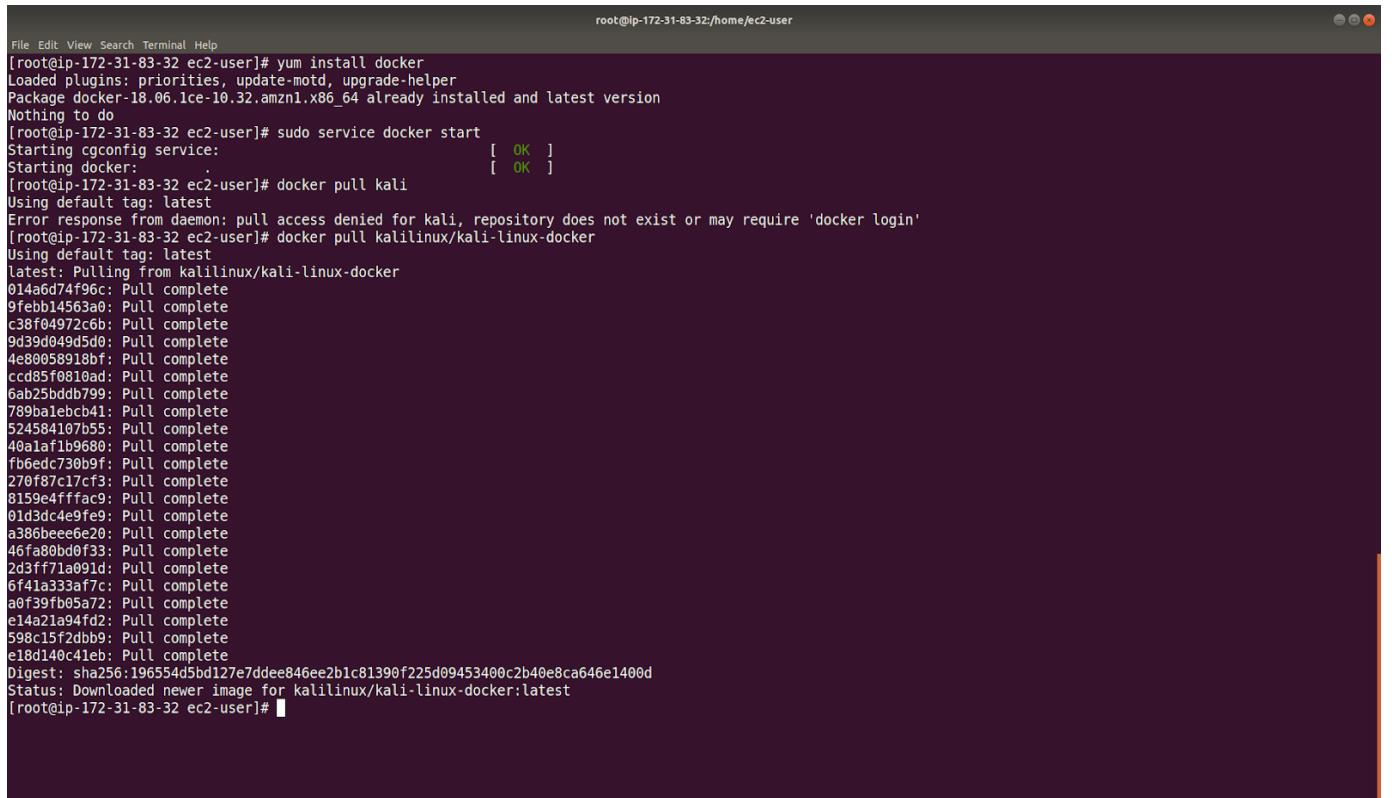
- Containers on the same OS kernel are lighter and smaller
- Better resource utilization compared to VMs
- Boot-up process is short and takes a few seconds

Installing Docker in EC2 Instance

> First, We need to launch an EC2 Instance

> Afterwards, Switch to root using sudo su

> Use Command yum install docker



```
root@ip-172-31-83-32:~# yum install docker
Loaded plugins: priorities, update-motd, upgrade-helper
Package docker-18.06.1ce-10.32.amzn1.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-83-32 ~]# sudo service docker start
Starting cgroup config service: [ OK ]
Starting docker: [ OK ]
[root@ip-172-31-83-32 ~]# docker pull kali
Using default tag: latest
Error response from daemon: pull access denied for kali, repository does not exist or may require 'docker login'
[root@ip-172-31-83-32 ~]# docker pull kalilinux/kali-linux-docker
Using default tag: latest
latest: Pulling from kalilinux/kali-linux-docker
014a6d74f96c: Pull complete
9febb14563a0: Pull complete
c38f04972c6b: Pull complete
9d39d049d5d0: Pull complete
4e80058918bf: Pull complete
cc85f0810ad: Pull complete
6ab25bdb799: Pull complete
799ba1ebcb41: Pull complete
524584107b55: Pull complete
40a1af1b9680: Pull complete
fb6edc730b9f: Pull complete
270f87c17cf3: Pull complete
8159e4fffae9: Pull complete
01d3dc4e9fe9: Pull complete
a386bee6e620: Pull complete
46fa80b0df33: Pull complete
2d3fff71a091d: Pull complete
6f41a333af7c: Pull complete
a0f39fb05a72: Pull complete
e14a21a94fd2: Pull complete
598c15f2dbb9: Pull complete
e18d140c41eb: Pull complete
Digest: sha256:196554d5bd127e7ddee846ee2b1c81390f225d09453400c2b40e8ca646e1400d
Status: Downloaded newer image for kalilinux/kali-linux-docker:latest
[root@ip-172-31-83-32 ~]#
```

> Use command **sudo service docker start** to check the status of the docker - engine

> Use command **docker pull <image name>** to pull the docker image.

> Use command **sudo docker run -it <image name>**

```
[root@ip-172-31-83-32 ec2-user]# sudo docker run -it kalilinux/kali-linux-docker
root@062ceb36c313:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
```

>Use command **docker ps** to see the running containers. To see the all containers use -a (or) -all flag

>Use Command **sudo docker exec -it <container id> bash** to run the docker image using the container id.

```
[root@ip-172-31-83-32 ec2-user]# sudo docker ps
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS
37062750b317        kalilinux/kali-linux-docker   "bash"        2 hours ago    Up 2 hours
[root@ip-172-31-83-32 ec2-user]# sudo docker exec -it 37062750b317 bash
root@37062750b317:/# ls
bin  dev  home  lib64  mnt  proc  run  srv  tmp  var
boot  etc  lib  media  opt  root  sbin  sys  usr
```

Docker Compose

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.

Compose works in all environments: production, staging, development, testing, as well as CI workflows.

Using Compose is basically a three-step process:

1. Define your app's environment with a Dockerfile so it can be reproduced anywhere.

2. Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.

3. Run docker-compose up and Compose starts and runs your entire app.

>Install Docker-Compose

```
root@ip-172-31-3-230:/home/ec2-user# sudo pip install docker-compose
Collecting docker-compose
  Downloading https://files.pythonhosted.org/packages/dd/e6/1521d1df9c0dal1863b18e592d91c3df222e55f258b9876fa1e59bc4b5/docker_compose-1.24.1-py2.py3-none-any.whl (134kB)
    100% |██████████| 143KB 6.1MB/s
Requirement already satisfied: PyYAML<4.3,>=3.10 in /usr/lib64/python2.7/dist-packages (from docker-compose)
Collecting backports.ssl-match-hostname>=3.5; python_version < "3.5" (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/ff/2b/8265224812912bc5b7a607c44bf7b027554e1b9775e9ee0de8032e3de4b2/backports.ssl_match_hostname-3.7.0.1.tar.gz
Collecting ipaddress>=1.0.16; python_version < "3.3" (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/fc/d0/7fc3a81e011d4b388be48a0e381db8d990042df54aa4ef4599a31d39853/ipaddress-1.0.22-py2.py3-none-any.whl
Collecting websocket-client<1.0,>=0.32.0 (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/29/19/44753eb1fd50770ac69605527e8859468f3c0fd7dc5a76dd9c4dbd7906/websocket_client-0.56.0-py2.py3-none-any.whl (200kB)
    100% |██████████| 204KB 4.6MB/s
Collecting docker[ssh]<4.0,>=3.7.0 (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/09/da/7cc7ecdd01145e9924a8ccbe9c1baf3a362fc75d4cb150676eb5231ea60/docker-3.7.3-py2.py3-none-any.whl (134kB)
    100% |██████████| 143KB 6.4MB/s
Collecting docopt<0.7,>=0.6.1 (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/a2/55/f8fcab2af404cf578136ef2cc5dfb50baa1761b68c9da1fb1e4eed343c9/docopt-0.6.2.tar.gz
Collecting jsonschema>3,>=2.5.1 (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/77/de/47e35a97b2b05c2fadbec67d44fcfcdd09b8086951b331d82d90d2912da/jsonschema-2.6.0-py2.py3-none-any.whl
Collecting enum34<2,>=1.0.4; python_version < "3.4" (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/c5/db/e56e6b4bbac7c4a06de1c50de6fe1ef3810018ae11732a50f15f62c7d050/enum34-1.1.6-py2-none-any.whl
Collecting cached-property<2,>=1.2.0 (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/3b/86/85c1be2e8db9e13ef9a350aec6dea292bd612fa288c2f40d035bb750ded/cached_property-1.5.1-py2.py3-none-any.whl
Requirement already satisfied: six<2,>=1.3.0 in /usr/lib/python2.7/dist-packages (from docker-compose)
Collecting requests!=2.11.0,>=2.12.2,>=2.18.0,>=2.21,>=2.6.1 (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/ff/17/5ccb02005115301a8fb2f9b0e3e8d32313142fe8b617070e7baad20554f/requests-2.20.1-py2.py3-none-any.whl (57kB)
    100% |██████████| 61KB 8.4MB/s
Collecting texttable<0.10,>=0.9.0 (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/02/e1/2565e6b842d7945af0555167d33acfca8a615584ef7abd30d1ae00a4d80/texttable-0.9.1.tar.gz
Collecting dockerpty>0.5,>=0.4.1 (from docker-compose)
  Downloading https://files.pythonhosted.org/packages/8d/ee/e9ecce4c32204a6738e0a5d5883d3413794d7498fe8b06f44becc028d3ba/dockerpty-0.4.1.tar.gz
Collecting docker-pycreds>=0.4.0 (from docker[ssh]<4.0,>=3.7.0->docker-compose)
  Downloading https://files.pythonhosted.org/packages/f5/e8/f6bd1ee09314e7e6dee49cbe2c5e22314cdb38db16c9fc72d2fa80d054/docker_pycreds-0.4.0-py2.py3-none-any.whl
Collecting paramiko>=2.4.2; extra == 'ssh' (from docker[ssh]<4.0,>=3.7.0->docker-compose)
  Downloading https://files.pythonhosted.org/packages/4b/80/74dace9e48b0ef923633dfb5e48798f58a168e4734bca8ecfaf839ba051a/paramiko-2.6.0-py2.py3-none-any.whl (199kB)
    100% |██████████| 204KB 4.9MB/s
Collecting functools32; python_version == "2.7" (from jsonschema<3,>=2.5.1->docker-compose)
  Downloading https://files.pythonhosted.org/packages/c5/60/6ac26ad05857c601308d8fb9e87fa36d0ebf889423f47c3502ef034365db/functools32-3.2.3-2.tar.gz
Collecting chardet<3.1.0,>=3.0.2 (from requests!=2.11.0,>=2.12.2,>=2.18.0,>=2.21,>=2.6.1->docker-compose)
  Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddc7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% |██████████| 143KB 6.4MB/s
Collecting certifi>=2017.4.17 (from requests!=2.11.0,>=2.12.2,>=2.18.0,>=2.21,>=2.6.1->docker-compose)
  Downloading https://files.pythonhosted.org/packages/18/b0/8146a4f8dd402f60744fa380bc73ca47303ccf8b9190fd16a827281eac2/certifi-2019.9.11-py2.py3-none-any.whl (154kB)
    100% |██████████| 163KB 6.3MB/s
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/lib/python2.7/dist-packages (from requests!=2.11.0,>=2.12.2,>=2.18.0,>=2.21,>=2.6.1->docker-compose)
```

Wordpress and phpmyadmin using Docker Compose

>here, I created a file named docker-compose.yml in **Cinephilia** Directory

>here, In below yml code, First I defined a container Named **Wordpress** and it's built from an image **wordpress** that was present in the docker hub. But, this wordpress image does not have a database. For that, I defined another container named **wordpress_db** and it's actually built from an image **mariadb** that was present in the docker hub.

>Afterwards, I need to link my **wordpress_db** with my **wordpress** container for that I written **wordpress_db:mysql** under **wordpress** links section.

>In, Ports section I linked port 80 of docker container with port 5000 of my Instance.

>After, I defined another container named **phpmyadmin** and it was built from an image

corbinu/docker-phpmyadmin that is present in the docker hub. I linked port 80 of this

Container **phpmyadmin** with port 5001 of my instance.

>Again, I need to link **phpmyadmin** container with my **wordpress_db** container. For that I written **wordpress_db:mysql** under links section of **phpmyadmin** container. Finally I given a user name as Root and password as Reddy.

>here, we will type **sudo docker-compose -up -d**. This command will actually pull all the images from the docker hub and built all the containers.

File Edit View Search Terminal Help

GNU nano 2.5.3

File: docker-compose.yml

```
wordpress:
  image: wordpress
  links:
    - wordpress_db:mysql
  ports:
    - 5000:80
wordpress_db:
  image: mariadb
  environment:
    MYSQL_ROOT_PASSWORD: root
phpmyadmin:
  image: corbinu/docker-phpmyadmin
  links:
    - wordpress_db:mysql
  ports:
    - 5001:80
  environment:
    MYSQL_USERNAME: root
    MYSQL_ROOT_PASSWORD: Reddy
```

[Read 20 lines]

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify **^C** Cur Pos **^Y** Prev Page
^X Exit **^R** Read File **^** Replace **^U** Uncut Text **^T** To Spell **^** Go To Line **^V** Next Page

```
[root@ip-172-31-3-230 Cinephilia]# sudo docker-compose up -d
Pulling wordpress_db (mariadb):...
latest: Pulling from library/mariadb
5667fdb72017: Pull complete
d83811f270d5: Pull complete
ee671aafb583: Pull complete
7fc152dfb3a6: Pull complete
9fe69c535a8b: Pull complete
a6de1092ee4e: Pull complete
ee37a2c88d9: Pull complete
d927a3dd356c: Pull complete
d83c9d39c64f: Pull complete
1b6644883413: Pull complete
09a38adc2558: Pull complete
3c853415b952: Pull complete
2690cf0bfab9: Pull complete
3c68d64f060f: Pull complete
Digest: sha256:a32daf0281803fd96e86daf6b0293b4d476ced1b5ce80b18452dfa1405360ff
Status: Downloaded newer image for mariadb:latest
Pulling wordpress (wordpress:...
latest: Pulling from library/wordpress
b8f262c62ec6: Pull complete
a98660e7def6: Pull complete
4d75689ceb37: Pull complete
639eb0368afa: Pull complete
99e337926e9c: Pull complete
431d44b3ce98: Pull complete
beb665ea0e0e: Pull complete
1914f5ed0362: Pull complete
3bb6538c14677: Pull complete
6a4699b1063e: Pull complete
d23f6accef3d: Pull complete
3814846efc9c: Pull complete
e14c865e4394: Pull complete
2133ee9f21fd: Pull complete
c54f25ec0676: Pull complete
b2f1c6cffd7b: Pull complete
c26c7edcc26f: Pull complete
119b6e1da171: Pull complete
555b521641eb: Pull complete
e2864c97d80f: Pull complete
dfb00b06eb36: Pull complete
Digest: sha256:661e9b06de5e9a2b9b44461206fe05362afbc66d4cc9e4f2135da083a709de9e
Status: Downloaded newer image for wordpress:latest
Pulling phpmyadmin (corbinu/docker-phpmyadmin:...
latest: Pulling from corbinu/docker-phpmyadmin
04c460fac791: Pull complete
```

```
[root@ip-172-31-3-230 Cinephilia]# nano docker-compose.yml
[root@ip-172-31-3-230 Cinephilia]# sudo docker-compose up -d
Recreating cinephilia_wordpress_db_1 ... done
cinephilia_phpmyadmin_1 is up-to-date
Recreating cinephilia_wordpress_1 ... done
[root@ip-172-31-3-230 Cinephilia]# █
```

54.183.135.167:5000 opens up wordpress installation page

54.183.135.167:5000/wp-admin/install.php?step=1 | Search



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

| | |
|---|--|
| Site Title | <input type="text" value="Honest Take On Saaho"/> |
| Username | <input type="text" value="Rk_Reddy"/> |
| Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol. | |
| Password | <input type="password" value="go2work#"/> Weak  Hide |
| Important: You will need this password to log in. Please store it in a secure location. | |
| Confirm Password | <input checked="" type="checkbox"/> Confirm use of weak password |
| Your Email | <input type="text" value="rkreddi01@gmail.com"/> |
| Double-check your email address before continuing. | |
| Search Engine Visibility | <input type="checkbox"/> Discourage search engines from indexing this site It is up to search engines to honor this request. |

Enter the credentials those were given in the yml code snippet

54.183.135.167:5001 opens up phpmyadmin page



The screenshot shows the phpMyAdmin configuration interface. On the left, there's a sidebar with database icons for New, Information_schema, mysql, performance_schema, and wordpress. The main area has several tabs: General settings, Appearance settings, Database server, Web server, and phpMyAdmin.

- General settings:** Includes a 'Change password' link and a dropdown for 'Server connection collation' set to 'utf8mb4_unicode_ci'.
- Appearance settings:** Shows 'Theme' set to 'pmahomme' and 'Font size' at '82%'. There's also a 'More settings' link.
- Database server:** Lists the server as '172.17.0.2 via TCP/IP', 'Server type: MariaDB', 'Server version: 10.4.8-MariaDB-1:10.4.8+maria~bionic~maradb.org binary distribution', 'Protocol version: 10', 'User: root@172.17.0.3', and 'Server charset: UTF-8 Unicode (utf8)'.
- Web server:** Lists 'nginx/1.7.12', 'Database client version: libmysql - 5.6.24', 'PHP extension: mysqli', and 'PHP version: 5.6.9-1+deb.sury.org~trusty+2'.
- phpMyAdmin:** Lists 'Version information: 4.5.0.2, latest stable version: 4.9.1', links to 'Documentation', 'Wiki', 'Official Homepage', 'Contribute', 'Get support', and 'List of changes'.

At the bottom, there are three error messages in red boxes:

- A newer version of phpMyAdmin is available and you should consider upgrading. The newest version is 4.9.1, released on 2019-09-21.
- The phpMyAdmin configuration storage is not completely configured, some extended features have been deactivated. [Find out why.](#) Or alternately go to 'Operations' tab of any database to set it up there.
- Connection for controluser as defined in your configuration failed.

docker-compose.yml code snippet

```
wordpress:  
  image: wordpress  
  links:  
    - wordpress_db:mysql  
  ports:  
    - 5000:80  
wordpress_db:  
  image: mariadb  
  environment:  
    MYSQL_ROOT_PASSWORD: Reddy  
phpmyadmin:  
  image : corbinu/docker-phpmyadmin  
  links:  
    - wordpress_db:mysql  
  ports:  
    - 5001:80  
  environment:  
    MYSQL_USERNAME: root  
    MYSQL_ROOT_PASSWORD: Reddy
```

Amazon Ecs

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, high-performance container orchestration service that supports Docker containers and allows you to easily run and scale containerized applications on AWS. Amazon ECS eliminates the need for you to install and operate your own container orchestration software, manage and scale a cluster of virtual machines, or schedule containers on those virtual machines.

With simple API calls, you can launch and stop Docker-enabled applications, query the complete state of your application, and access many familiar features such as IAM roles, security groups, load balancers, Amazon CloudWatch Events, AWS CloudFormation templates, and AWS CloudTrail logs

AWS FARGATE

AWS Fargate is a compute engine for Amazon ECS that allows you to run containers without having to manage servers or clusters. With AWS Fargate, you no longer have to provision, configure, and scale clusters of virtual machines to run containers. This removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing. AWS Fargate removes the need for you to interact with or think about servers or clusters. Fargate lets you focus on designing and building your applications instead of managing the infrastructure that runs them.

Amazon ECS has two modes: Fargate launch type and EC2 launch type. With Fargate launch type, all you have to do is package your application in containers, specify the CPU and memory requirements,

define networking and IAM policies, and launch the application. EC2 launch type allows you to have server-level, more granular control over the infrastructure that runs your container applications. With EC2 launch type, you can use Amazon ECS to manage a cluster of servers and schedule placement of containers on the servers. Amazon ECS keeps track of all the CPU, memory and other resources in your cluster, and also finds the best server for a container to run on based on your specified resource requirements. You are responsible for provisioning, patching, and scaling clusters of servers. You can decide which type of server to use, which applications and how many containers to run in a cluster to optimize utilization, and when you should add or remove servers from a cluster. EC2 launch type gives you more control of your server clusters and provides a broader range of customization options, which might be required to support some specific applications or possible compliance and government requirements.

>Go to My Security Credentials to create and download Access Key. Later, That will be helpful in configuring Awscli.

Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for Amazon CloudWatch Metrics, see the [Amazon CloudWatch Metrics User Guide](#).

To learn more about the types of AWS credentials and how they're used, see [AWS Security Credentials](#).

- [▲ Password](#)
- [▲ Multi-factor authentication \(MFA\)](#)
- [▼ Access keys \(access key ID and secret access key\)](#)

[My Account](#)
[My Organization](#)
[My Service Quotas](#)
[My Billing Dashboard](#)
[Orders and Invoices](#)
[My Security Credentials](#)

[Sign Out](#)

You use access keys to sign programmatic requests to AWS services. To learn how to sign requests using your access keys, see the [signing documentation](#). For your protection, store your access keys securely and do not share them. In addition, AWS recommends that you rotate your access keys every 90 days.

Note: You can have a maximum of two access keys (active or inactive) at a time.

| Created | Deleted | Access Key ID | Last Used | Last Used Region | Last Used Service | Status | Actions |
|---------------------------------------|---------|----------------------|-----------|------------------|-------------------|--------|--|
| Oct 18th 2019 | | AKIAJ3GRQK3H3DKCAPNQ | N/A | N/A | N/A | Active | Make Inactive Delete |
| Create New Access Key | | | | | | | |

>In order to install awscli first, we need to install python pip by using commands

>**curl -0 <https://bootstrap.pypa.io/get-pip.py> and python get-pip.py --user**

```
root@ip-172-31-43-72:~/docker-on-ecs
File Edit View Search Terminal Help
delta@delta-Lenovo-ideapad-320-15IKB:~$ cd Downloads/
delta@delta-Lenovo-ideapad-320-15IKB:~/Downloads$ ssh -i a.pem ec2-user@3.89.184.21
Last login: Fri Oct 18 06:57:05 2019 from 14.98.185.26
              _|_(_|_) /   Amazon Linux AMI
              __|_\_|__|_
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
4 package(s) needed for security, out of 10 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-43-72 ~]$ sudo su
[root@ip-172-31-43-72 ec2-user]# curl -0 https://bootstrap.pypa.io/get-pip.py
  % Total    % Received % Xferd  Average Speed   Time   Time   Current
     0      0      0      0      0      0      0      0      0      0
  100 1734k  100 1734k    0      0  19.6M      0 --:--:-- --:--:-- 19.9M
[root@ip-172-31-43-72 ec2-user]# python get-pip.py --user
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade
your Python as Python 2.7 won't be maintained after that date. A future version of pip will
drop support for Python 2.7. More details about Python 2 support in pip, can be found at
https://pip.pypa.io/en/latest/development/release-process/#python-2-support
Collecting pip
  Downloading https://files.pythonhosted.org/packages/00/b6/9cfa56b4081ad13874b0c6f96af8ce1
6cfbc1ch06hedef8e9164ce5551ec1/nin-19.3.1-nv2_nv3-none-any.whl (1.4MR)
```

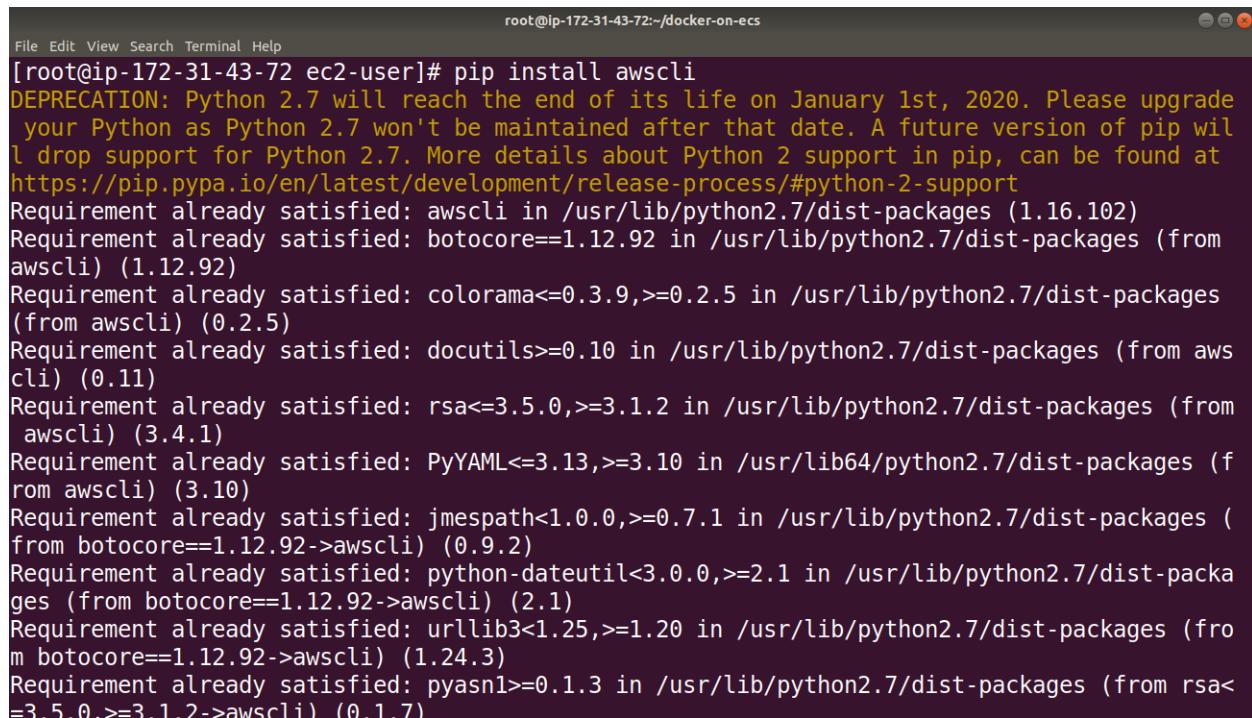
>Install awscli by command **pip install awscli**

>after installation configure your awscli by using command **aws configure**

>you need to have your Access Key and Secret Access Key, if you don't have one follow the below steps otherwise skip that part.

>Create Access Key Login to AWS Console Under your IAM name choose 'My Security Credentials' My Security Credentials Click on Access Keys, create a new one like I've shown in the first screenshot and save the Secret Access Key as you no longer going to have access to it anymore.

>Create Access Key Go to your EC2 again and run 'aws configure' Put both Access and Secret Key Choose your region or leave it empty for default



```
root@ip-172-31-43-72:~# pip install awscli
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of pip will drop support for Python 2.7. More details about Python 2 support in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
Requirement already satisfied: awscli in /usr/lib/python2.7/dist-packages (1.16.102)
Requirement already satisfied: botocore==1.12.92 in /usr/lib/python2.7/dist-packages (from awscli) (1.12.92)
Requirement already satisfied: colorama<=0.3.9,>=0.2.5 in /usr/lib/python2.7/dist-packages (from awscli) (0.2.5)
Requirement already satisfied: docutils>=0.10 in /usr/lib/python2.7/dist-packages (from awscli) (0.11)
Requirement already satisfied: rsa<=3.5.0,>=3.1.2 in /usr/lib/python2.7/dist-packages (from awscli) (3.4.1)
Requirement already satisfied: PyYAML<=3.13,>=3.10 in /usr/lib64/python2.7/dist-packages (from awscli) (3.10)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/lib/python2.7/dist-packages (from botocore==1.12.92->awscli) (0.9.2)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/lib/python2.7/dist-packages (from botocore==1.12.92->awscli) (2.1)
Requirement already satisfied: urllib3<1.25,>=1.20 in /usr/lib/python2.7/dist-packages (from botocore==1.12.92->awscli) (1.24.3)
Requirement already satisfied: pyasn1>=0.1.3 in /usr/lib/python2.7/dist-packages (from rsa<=3.5.0.>=3.1.2->awscli) (0.1.7)
```

>Install git for cloning and other activities of repositories.

```
root@ip-172-31-43-72:~/docker-on-ecs
File Edit View Search Terminal Help
[root@ip-172-31-43-72 ~]# yum install git
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                         | 2.1 kB  00:00:00
amzn-updates                                      | 2.5 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.14.5-1.60.amzn1 will be installed
--> Processing Dependency: perl-Git = 2.14.5-1.60.amzn1 for package: git-2.14.5-1.60.amzn1.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.14.5-1.60.amzn1.x86_64
--> Processing Dependency: perl(Git::I18N) for package: git-2.14.5-1.60.amzn1.x86_64
--> Processing Dependency: perl(Git) for package: git-2.14.5-1.60.amzn1.x86_64
--> Processing Dependency: perl(Error) for package: git-2.14.5-1.60.amzn1.x86_64
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.9.amzn1 will be installed
--> Package perl-Git.noarch 0:2.14.5-1.60.amzn1 will be installed
--> Package perl-TermReadKey.x86_64 0:2.30-20.9.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch    Version        Repository      Size
=====
```

>here,We are going to perform creating and running containerized application on Amazon Elastic Container Service (ECS) with AWS Fargate using AWS CloudFormation and AWS CLI.

Following are the steps to walk through,

1. Containerize a simple REST API application - Build Docker Image
2. Using AWS CLI, create repository on Amazon Elastic Container Registry (Docker Registry)
3. Push the docker image to repository on Amazon ECR
4. Define and run CloudFormation stack to create VPC and all network components
5. Define and run CloudFormation stack to create IAM roles
6. Define and run CloudFormation stack to create ECS Cluster, Load balancer, CloudWatch Log Group and Security Groups

7. Define and run CloudFormation stack to deploy docker application - Create Task for container, Service to schedule the task, Load balancer target group for container services and Load balancer listener rule for container service to run docker container on ECS with Fargate.

>First, Initialize a Git repository in root.then clone the repository using command

```
git clone https://github.com/devteds/e9-cloudformation-docker-ecs.git  
docker-on-ecs
```

>this repository contains the yaml files those will be helpful for creating stacks on cloudformation and creating the simple docker application.

Dockerize a simple app

```
# Run on local
```

```
docker build -t books ./app/
```

```
docker run -it -p 4567:4567 --rm books:latest
```

```
open http://localhost:4567/
```

```
open http://localhost:4567/stat
```

```
open http://localhost:4567/api/books
```

```
root@ip-172-31-43-72:~/docker-on-ecs
[File Edit View Search Terminal Help]
[root@ip-172-31-43-72 ~]# git init
Initialized empty Git repository in /root/.git/
[root@ip-172-31-43-72 ~]# git clone https://github.com/devteds/e9-cloudformation-docker-ecs.git docker-on-ecs
Cloning into 'docker-on-ecs'...
remote: Enumerating objects: 28, done.
remote: Total 28 (delta 0), reused 0 (delta 0), pack-reused 28
Unpacking objects: 100% (28/28), done.
[root@ip-172-31-43-72 ~]# cd docker-
docker-compose.yml docker-on-ecs/
[root@ip-172-31-43-72 docker-on-ecs]# cd docker-on-ecs/
[root@ip-172-31-43-72 docker-on-ecs]# ls
app deploy app.sh infra README.md
[root@ip-172-31-43-72 docker-on-ecs]# docker build -t books ./app/
Sending build context to Docker daemon 5.12kB
Step 1/9 : FROM ruby:2.5.1
2.5.1: Pulling from library/ruby
bc9ab73e5b14: Pull complete
193a6306c92a: Pull complete
e5c3f8c317dc: Pull complete
a587a86c9dcb: Pull complete
72744d0a318b: Pull complete
31d57ef7a684: Pull complete
a2a726425592: Pull complete
```

```
root@ip-172-31-43-72:~/docker-on-ecs
[File Edit View Search Terminal Help]
[root@ip-172-31-43-72 docker-on-ecs]# docker run -it -p 4567:4567 --rm books:latest
[2019-10-18 09:55:27] INFO WEBrick 1.4.2
[2019-10-18 09:55:27] INFO ruby 2.5.1 (2018-03-29) [x86_64-linux]
== Sinatra (v2.0.7) has taken the stage on 4567 for development with backup from WEBrick
[2019-10-18 09:55:27] INFO WEBrick::HTTPServer#start: pid=1 port=4567
^X^Z^C== Sinatra has ended his set (crowd applauds)
[2019-10-18 09:57:49] INFO going to shutdown ...
[2019-10-18 09:57:49] INFO WEBrick::HTTPServer#start done.
[root@ip-172-31-43-72 docker-on-ecs]# service httpd start
Starting httpd:
[root@ip-172-31-43-72 docker-on-ecs]# service httpd restart
Stopping httpd: [OK]
Starting httpd: [OK]
[root@ip-172-31-43-72 docker-on-ecs]# docker run -it -p 4567:4567 --rm books:latest
[2019-10-18 09:59:40] INFO WEBrick 1.4.2
[2019-10-18 09:59:40] INFO ruby 2.5.1 (2018-03-29) [x86_64-linux]
== Sinatra (v2.0.7) has taken the stage on 4567 for development with backup from WEBrick
[2019-10-18 09:59:40] INFO WEBrick::HTTPServer#start: pid=1 port=4567
14.98.185.26 - - [18/Oct/2019:09:59:50 +0000] "GET / HTTP/1.1" 200 10 0.0043
14.98.185.26 - - [18/Oct/2019:09:59:50 UTC] "GET / HTTP/1.1" 200 10
- -> /
14.98.185.26 - - [18/Oct/2019:09:59:51 +0000] "GET /favicon.ico HTTP/1.1" 404 471 0.0005
14.98.185.26 - - [18/Oct/2019:09:59:51 UTC] "GET /favicon.ico HTTP/1.1" 404 471
- -> /favicon.ico
```

root@ip-172-31-43-72:~/HermoinI

File Edit View Search Terminal Help

GNU nano 2.5.3 File: sample.rb

```
require 'sinatra'
require "sinatra/namespace"

class Book
  @@books = [
    { id: "123", name: "Docker for Beginners" },
    { id: "124", name: "Docker with Kubernetes" },
    { id: "125", name: "Docker on ECS" }
  ]

  def self.all
    @@books
  end

  def self.find(book_id)
    @@books.select{ |p| p[:id] == book_id }.first
  end
end
```

[Read 54 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^^ Go To Line

root@ip-172-31-43-72:~/HermoinI

File Edit View Search Terminal Help

GNU nano 2.5.3 File: gemfile

```
source 'https://rubygems.org'

gem 'sinatra', '~> 2.0.1'
gem 'sinatra-contrib', '~> 2.0.1'
```

[Read 4 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^^ Go To Line

```
root@ip-172-31-43-72:~/Hermoini
File Edit View Search Terminal Help
GNU nano 2.5.3                               File: docker

FROM ruby:2.5.1
RUN apt-get update -qq && apt-get install -y build-essential libpq-dev
RUN mkdir /sapp
WORKDIR /sapp

ADD Gemfile /sapp/Gemfile
ADD Gemfile.lock /sapp/Gemfile.lock

RUN bundle install

ADD . /sapp

CMD ["ruby", "service.rb", "-o", "0.0.0.0"]

[ Read 16 lines ]
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos
^X Exit          ^R Read File     ^E Replace       ^U Uncut Text    ^T To Spell      ^L Go To Line
```

Security groups | EC2 Management Console - Mozilla Firefox (Private Browsing)

Google Docs | (2) Delta Scho | Ramakrishna | Assignment_ | Docker on Am | e9-cloudform | e9-cloudform | Docker: unabl | Security gro X + ☰

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#SecurityGroups:search= ... 🌐 ⚡

Most Visited Getting Started Rain-Typing Games - ...

aws Services Resource Groups Rk_Reddy N. Virginia Support

Edit inbound rules

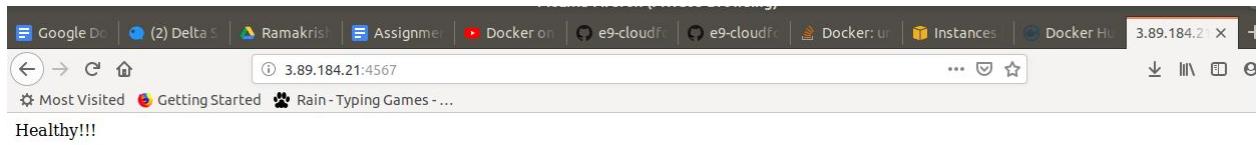
| Type | Protocol | Port Range | Source | Description | Actions |
|------------|----------|------------|--------|-------------|----------------------------|
| HTTP | TCP | 80 | Custom | 0.0.0.0/0 | e.g. SSH for Admin Desktop |
| HTTP | TCP | 80 | Custom | ::/0 | e.g. SSH for Admin Desktop |
| SSH | TCP | 22 | Custom | 0.0.0.0/0 | e.g. SSH for Admin Desktop |
| Custom TCF | TCP | 4567 | Custom | 0.0.0.0/0 | e.g. SSH for Admin Desktop |
| Custom TCF | TCP | 4567 | Custom | ::/0 | e.g. SSH for Admin Desktop |
| HTTPS | TCP | 443 | Custom | 0.0.0.0/0 | e.g. SSH for Admin Desktop |
| HTTPS | TCP | 443 | Custom | ::/0 | e.g. SSH for Admin Desktop |

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

Feedback English (US) © 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use



A screenshot of a JSON viewer interface. The URL in the address bar is `3.89.184.21:4567/api/books`. The JSON data is displayed as follows:

```
[{"id": "123", "name": "Docker for Beginners"}, {"id": "124", "name": "Docker with Kubernetes"}, {"id": "125", "name": "Docker on ECS"}]
```

The JSON viewer includes tabs for "JSON", "Raw Data", and "Headers", and provides options to "Save", "Copy", "Collapse All", "Expand All", and "Filter JSON".

Push Docker Image to ECR

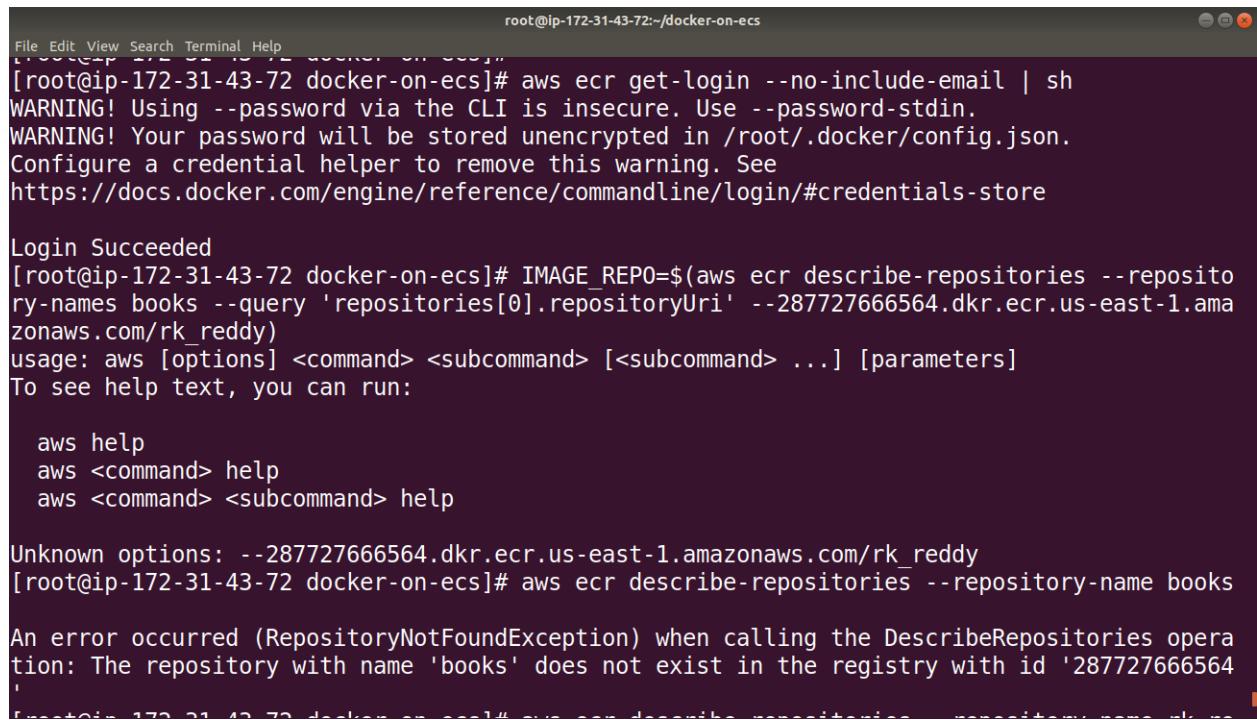
```
aws ecr create-repository --repository-name rk_reddy
```

```
aws ecr get-login --no-include-email | sh
```

```
IMAGE_REPO=$(aws ecr describe-repositories --repository-name rk_reddy --query 'repositories[0].repositoryUri' --output text)
```

```
docker tag books-api:latest $IMAGE_REPO:v1
```

```
docker push $IMAGE_REPO:v1
```



The screenshot shows a terminal window with a dark background and white text. The title bar says "root@ip-172-31-43-72:~/docker-on-ecs". The terminal output is as follows:

```
[root@ip-172-31-43-72 docker-on-ecs]# aws ecr get-login --no-include-email | sh
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[root@ip-172-31-43-72 docker-on-ecs]# IMAGE_REPO=$(aws ecr describe-repositories --repository-names books --query 'repositories[0].repositoryUri' --287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy)
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help

Unknown options: --287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy
[root@ip-172-31-43-72 docker-on-ecs]# aws ecr describe-repositories --repository-name books

An error occurred (RepositoryNotFoundException) when calling the DescribeRepositories operation: The repository with name 'books' does not exist in the registry with id '287727666564'
```

```
root@ip-172-31-43-72:~/docker-on-ecs# aws ecr create-repository --repository-name RkReddy
An error occurred (InvalidParameterException) when calling the CreateRepository operation:
Invalid parameter at 'repositoryName' failed to satisfy constraint: 'must satisfy regular expression '(?:[a-zA-Z0-9]+(?:[-_][a-zA-Z0-9]+)*|[a-zA-Z0-9]+(?:[-_][a-zA-Z0-9]+)*)''
[root@ip-172-31-43-72:~/docker-on-ecs# aws ecr create-repository --repository-name Rk_Reddy
An error occurred (InvalidParameterException) when calling the CreateRepository operation:
Invalid parameter at 'repositoryName' failed to satisfy constraint: 'must satisfy regular expression '(?:[a-zA-Z0-9]+(?:[-_][a-zA-Z0-9]+)*|[a-zA-Z0-9]+(?:[-_][a-zA-Z0-9]+)*)''
[root@ip-172-31-43-72:~/docker-on-ecs# aws ecr create-repository --repository-name rk_reddy
{
    "repository": {
        "registryId": "287727666564",
        "repositoryName": "rk_reddy",
        "repositoryArn": "arn:aws:ecr:us-east-1:287727666564:repository/rk_reddy",
        "createdAt": 1571393167.0,
        "repositoryUri": "287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy"
    }
}
[root@ip-172-31-43-72:~/docker-on-ecs#]
[root@ip-172-31-43-72:~/docker-on-ecs#]
[root@ip-172-31-43-72:~/docker-on-ecs#]
```

```
[root@ip-172-31-43-72:~/docker-on-ecs# aws ecr describe-repositories --repository-name rk_reddy
{
    "repositories": [
        {
            "registryId": "287727666564",
            "repositoryName": "rk_reddy",
            "repositoryArn": "arn:aws:ecr:us-east-1:287727666564:repository/rk_reddy",
            "createdAt": 1571393167.0,
            "repositoryUri": "287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy"
        }
    ]
}
```

```
[root@ip-172-31-91-76 docker-on-ecs]# docker tag books 287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy:v1
[root@ip-172-31-91-76 docker-on-ecs]# docker push 287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy:v1
The push refers to repository [287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy]
ca879a8e1586: Pushed
bf6eeead2fb04: Pushed
c0833ae30f4e: Pushed
bd6f83d12e0a: Pushed
46c1fe072307: Pushed
55de4a218619: Pushed
3696bfaafb5ed: Pushed
dda5ada3fa53: Pushed
f5e615773459: Pushed
9978d084fd77: Pushed
1191b3f5862a: Pushed
08a01612ffca: Pushed
8bb25f9cdc41: Pushed
f715ed19c28b: Pushed
v1: digest: sha256:c9d58863d1461557d7279757eae7737a624b3a3eadf532e33529d66b344dd5cb size: 3255
[root@ip-172-31-91-76 docker-on-ecs]#
```

Amazon ECR - Mozilla Firefox

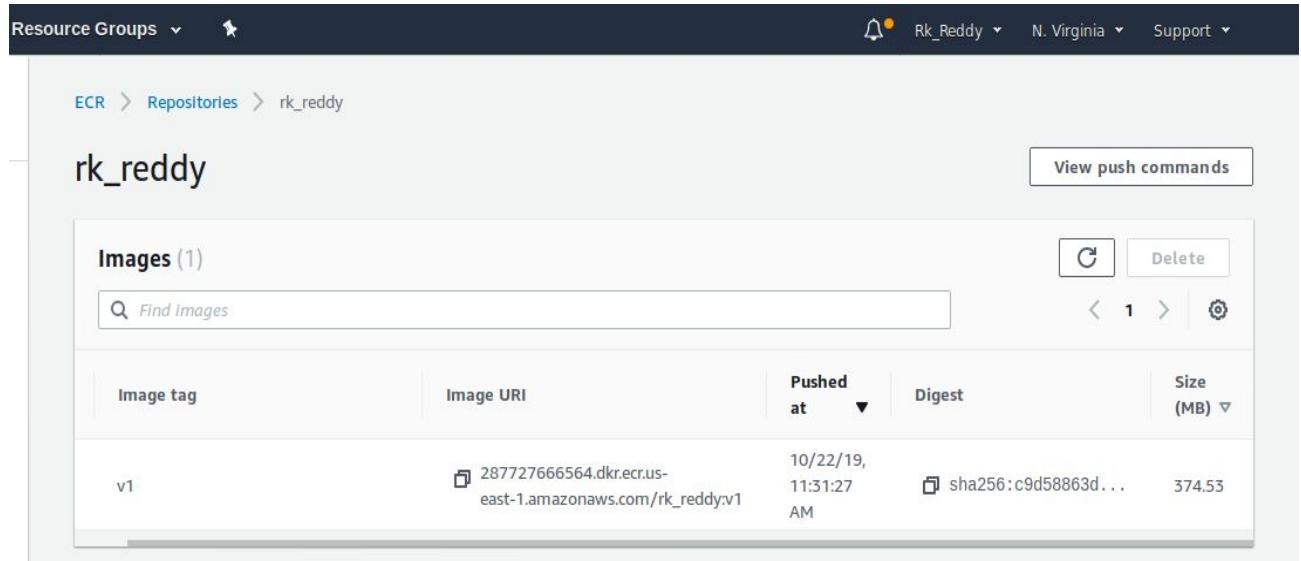
The screenshot shows the AWS ECR console in a Mozilla Firefox browser. The left sidebar has a collapsed navigation bar with the following items:

- Amazon Container Services
- Amazon ECS
 - Clusters
 - Task definitions
- Amazon EKS
 - Clusters
- Amazon ECR
 - Repositories** (selected)

The main content area is titled "Repositories" and shows a table with one item:

| Repository name | URI | Created at | Tag immutability |
|-----------------|---|----------------------|------------------|
| rk_reddy | 287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy | 10/18/19, 3:36:07 PM | Disabled |

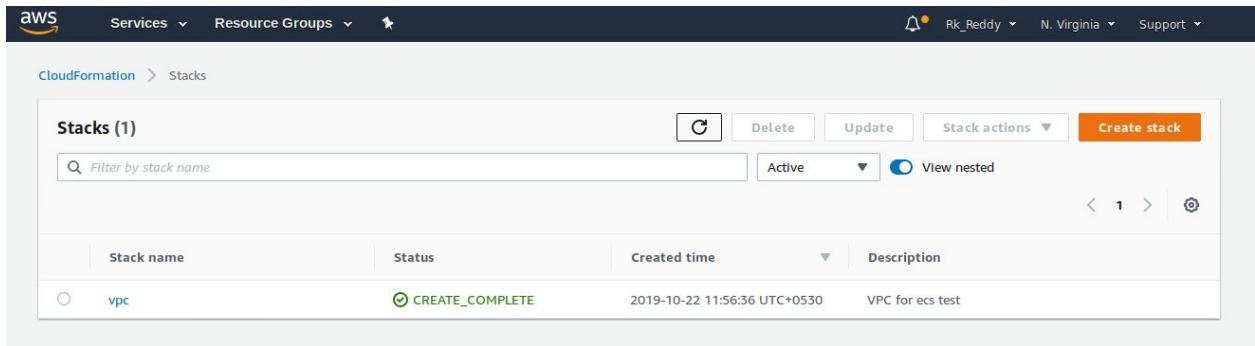
At the bottom of the page, there are links for Feedback, English (US), Copyright notice (© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.



Creation of VPC stack using yaml scripting

```
aws cloudformation create-stack --template-body file://$PWD/infra/vpc.yml  
--stack-name vpc
```

```
[root@ip-172-31-91-76 docker-on-ecs]# aws cloudformation create-stack --template-body file://$PWD/infra/vpc.yml --stack-name vpc  
An error occurred (AlreadyExistsException) when calling the CreateStack operation: Stack [vpc] already exists  
[root@ip-172-31-91-76 docker-on-ecs]#
```



vpc.yml

AWSTemplateFormatVersion: '2010-09-09'

Description: VPC for ecs test

Resources:

VPC:

Type: 'AWS::EC2::VPC'

Properties:

CidrBlock: '10.0.0.0/16'

EnableDnsSupport: true

EnableDnsHostnames: true

InstanceTenancy: default

Subnet1:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [0, !GetAZs "]

CidrBlock: !Sub '10.0.0.0/20'

MapPublicIpOnLaunch: true

Subnet2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [1, !GetAZs "]

CidrBlock: !Sub '10.0.32.0/20'

MapPublicIpOnLaunch: true

InternetGateway:

Type: 'AWS::EC2::InternetGateway'

VPCGatewayAttachment:

Type: 'AWS::EC2::VPCGatewayAttachment'

Properties:

InternetGatewayId: !Ref InternetGateway

VpcId: !Ref VPC

RouteTable:

Type: 'AWS::EC2::RouteTable'

Properties:

VpcId: !Ref VPC

RouteTableAssociation1:

Type: 'AWS::EC2::SubnetRouteTableAssociation'

Properties:

SubnetId: !Ref Subnet1

RouteTableId: !Ref RouteTable

RouteTableAssociation2:

Type: 'AWS::EC2::SubnetRouteTableAssociation'

Properties:

SubnetId: !Ref Subnet2

RouteTableId: !Ref RouteTable

InternetRoute:

Type: 'AWS::EC2::Route'

DependsOn: VPCGatewayAttachment

Properties:

GatewayId: !Ref InternetGateway

RouteTableId: !Ref RouteTable

DestinationCidrBlock: '0.0.0.0/0'

Outputs:

VPC:

Description: VPC

Value: !Ref VPC

Export:

Name: 'VPC'

Subnet1:

Description: 'Subnet 1 on AZ1'

Value: !Ref Subnet1

Export:

Name: 'Subnet1'

Subnet2:

Description: 'Subnet 2 on AZ2'

Value: !Ref Subnet2

Export:

Name: 'Subnet2'

The screenshot shows the AWS CloudFormation console interface. On the left, there's a sidebar with 'Resource Groups' and a search bar. The main area shows a 'CloudFormation > Stacks > vpc' navigation path. The 'vpc' stack is selected. The top right has buttons for 'Delete', 'Update', 'Stack actions', and 'Create stack'. Below that is a navigation bar with tabs: 'Stack info', 'Events', 'Resources', 'Outputs' (which is highlighted in orange), 'Parameters', and 'Template'. Under the 'Outputs' tab, there's a section titled 'Outputs (3)' with a search bar. A table lists the outputs:

| Key | Value | Description | Export name |
|---------|--------------------------|-----------------|-------------|
| Subnet1 | subnet-0a4efd81498fc9284 | Subnet 1 on AZ1 | Subnet1 |
| Subnet2 | subnet-0a7343092ce6f4b6e | Subnet 2 on AZ2 | Subnet2 |
| VPC | vpc-0358e320e29748b2c | VPC | VPC |

Creation of IAM stack using yml

```
aws cloudformation create-stack --template-body file://$PWD/infra/iam.yml  
--stack-name iam --capabilities CAPABILITY_IAM
```

```
[root@ip-172-31-91-76 docker-on-ecs]# aws cloudformation create-stack --template-body file://$PWD/infra/  
iam.yml --stack-name iam --capabilities CAPABILITY_IAM  
{  
    "StackId": "arn:aws:cloudformation:us-east-1:287727666564:stack/iam/c5781fc0-f4b6-11e9-b9f0-0a933e79  
26f6"  
}  
[root@ip-172-31-91-76 docker-on-ecs]#
```

The screenshot shows the AWS CloudFormation console with the 'iam' stack selected. The 'Outputs' tab is active, showing one output named 'ECSTaskExecutionRole' with the value 'arn:aws:iam::287727666564:role/iam-ECSTaskExecutionRole-1QBUKJXVYBJA1'. The 'Description' column indicates it is an 'ECS Task Execution Role' and the 'Export name' is 'ECSTaskExecutionRole'.

| Key | Value | Description | Export name |
|----------------------|---|-------------------------|----------------------|
| ECSTaskExecutionRole | arn:aws:iam::287727666564:role/iam-ECSTaskExecutionRole-1QBUKJXVYBJA1 | ECS Task Execution Role | ECSTaskExecutionRole |

iam.yml

AWSTemplateFormatVersion: '2010-09-09'

Description: roles and policies

Resources:

ECSTaskExecutionRole:

Type: AWS::IAM::Role

Properties:

AssumeRolePolicyDocument:

Statement:

- Effect: Allow

Principal:

Service: [ecs-tasks.amazonaws.com]

Action: ['sts:AssumeRole']

Path: /

Policies:

- PolicyName: AmazonECSTaskExecutionRolePolicy

PolicyDocument:

Statement:

- Effect: Allow

Action:

ECS Tasks to download images from ECR

- 'ecr:GetAuthorizationToken'
- 'ecr:BatchCheckLayerAvailability'
- 'ecr:GetDownloadUrlForLayer'
- 'ecr:BatchGetImage'

ECS tasks to upload logs to CloudWatch

- 'logs>CreateLogStream'
- 'logs:PutLogEvents'

Resource: '*'

Outputs:

ECSTaskExecutionRole:

Description: ECS Task Execution Role

Value: !GetAtt 'ECSTaskExecutionRole.Arn'

Export:

Name: 'ECSTaskExecutionRole'

Creation of app-cluster stack using yml

```
aws cloudformation create-stack --template-body  
file://$PWD/infra/app-cluster.yaml --stack-name app-cluster
```

```
[root@ip-172-31-91-76 docker-on-ecs]# aws cloudformation create-stack --template-body file://$PWD/infra/  
app-cluster.yaml --stack-name app-cluster  
{  
    "StackId": "arn:aws:cloudformation:us-east-1:287727666564:stack/app-cluster/3e954300-f4b8-11e9-b986-  
0a56ba5f6f9e"  
}  
[root@ip-172-31-91-76 docker-on-ecs]# █
```

| CloudFormation > Stacks | | | | |
|---|-----------------|------------------------------|-------------|---|
| Stacks (3) <input type="button" value="C"/> Delete Update Stack actions ▾ <input type="button" value="Create stack"/> | | | | |
| <input type="text"/> Filter by stack name | | Active | View nested | ◀ 1 ▶ ⚙ |
| <hr/> | | | | |
| Stack name | Status | Created time | ▼ | Description |
| app-cluster | CREATE_COMPLETE | 2019-10-22 16:09:37 UTC+0530 | | container cluster on ECS, loadbalancer, security group... |
| iam | CREATE_COMPLETE | 2019-10-22 15:59:04 UTC+0530 | | roles and policies |
| vpc | CREATE_COMPLETE | 2019-10-22 11:56:36 UTC+0530 | | VPC for ecs test |

| CloudFormation > Stacks > app-cluster | | | | |
|---|-----------------|--|---|---|
| Stacks (3) <input type="button" value="C"/> | | Outputs (4) <input type="button" value="C"/> | | |
| <input type="text"/> Filter by stack name | | <input type="text"/> Search outputs | | |
| <input checked="" type="radio"/> View nested ◀ 1 ▶ ⚙ | | | | |
| app-cluster | CREATE_COMPLETE | Cluster | bookstore | - ECSCluster |
| iam | CREATE_COMPLETE | ContainerSecurityGroup | sg-0959bf2db29e09995 | container security group ContainerSecurityGroup |
| vpc | CREATE_COMPLETE | Listener | arn:aws:elasticloadbalancing:us-east-1:287727666564:listener/app/ecs-services/31053d6d2fb8df2c/262d8deed349d127 | listener port 80 Listener |
| | | LoadBalancerDNS | ecs-services-1097299462.us-east-1.elb.amazonaws.com | Domain name for the loadbalancer DomainName |

app-cluster.yml

AWSTemplateFormatVersion: '2010-09-09'

Description: container cluster on ECS, load balancer, security groups and cloudwatch

Resources:

ECSCluster:

Type: AWS::ECS::Cluster

Properties:

ClusterName: 'bookstore'

LoadBalancer:

Type: AWS::ElasticLoadBalancingV2::LoadBalancer

Properties:

Name: ecs-services

Subnets:

- !ImportValue 'Subnet1'

- !ImportValue 'Subnet2'

SecurityGroups:

- !Ref LoadBalancerSecurityGroup

LoadBalancerListener:

Type: AWS::ElasticLoadBalancingV2::Listener

Properties:

LoadBalancerArn: !Ref LoadBalancer

Protocol: HTTP

Port: 80

DefaultActions:

- Type: forward

TargetGroupArn: !Ref DefaultTargetGroup

LoadBalancerSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: Security group for load balancer to services on ECS

VpcId: !ImportValue 'VPC'

SecurityGroupIngress:

- CidrIp: 0.0.0.0/0

IpProtocol: -1

DefaultTargetGroup:

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

Name: default

VpcId: !ImportValue 'VPC'

Protocol: 'HTTP'

Port: '80'

CloudWatchLogsGroup:

Type: AWS::Logs::LogGroup

Properties:

LogGroupName: 'apis'

RetentionInDays: 1

ContainerSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

VpcId: !ImportValue 'VPC'

GroupDescription: for ecs containers

SecurityGroupIngress:

- SourceSecurityGroupId: !Ref 'LoadBalancerSecurityGroup'

IpProtocol: -1

Outputs:

Cluster:

Value: !Ref ECSCluster

Export:

Name: 'ECSCluster'

Listener:

Description: listener port 80

Value: !Ref LoadBalancerListener

Export:

Name: 'Listener'

ContainerSecurityGroup:

Description: container security group

Value: !Ref ContainerSecurityGroup

Export:

Name: 'ContainerSecurityGroup'

LoadBalancerDNS:

Description: Domain name for the load balancer

Value: !GetAtt LoadBalancer.DNSName

Export:

Name: 'DomainName'

Creation of api stack using yml

```
aws cloudformation create-stack --template-body file://$PWD/infra/api.yaml  
--stack-name api
```

```
[root@ip-172-31-91-76 docker-on-ecs]# aws cloudformation create-stack --template-body file://$PWD/infra/  
api.yaml --stack-name api  
{  
    "StackId": "arn:aws:cloudformation:us-east-1:287727666564:stack/api/f3cddc80-f4ba-11e9-a2f7-1271b8f3  
5690"  
}  
[root@ip-172-31-91-76 docker-on-ecs]#
```

| Stack name | Status | Created time | Description |
|-------------|-----------------|------------------------------|-----------------------------|
| api | CREATE_COMPLETE | 2019-10-22 16:29:00 UTC+0530 | container on ecs cluster |
| app-cluster | CREATE_COMPLETE | 2019-10-22 16:09:37 UTC+0530 | container cluster on ECS, I |
| iam | CREATE_COMPLETE | 2019-10-22 15:59:04 UTC+0530 | roles and policies |
| vpc | CREATE_COMPLETE | 2019-10-22 11:56:36 UTC+0530 | VPC for ecs test |

We created four cloudformation stacks api,app-cluster,iam and vpc using yml scripting.

The screenshot shows the AWS CloudFormation console interface. On the left, there's a sidebar titled "Resource Groups" with a star icon. Below it, the navigation path is "CloudFormation > Stacks > api". The main area has a title "api" and a toolbar with "Delete", "Update", "Stack actions ▾", and "Create stack". A navigation bar below the toolbar includes "Stack info", "Events", "Resources", "Outputs" (which is highlighted in orange), "Parameters", and "Template".

Stacks (4)

- api** (Selected) - Created on 2019-10-22 16:29:00 UTC+0530, status: CREATE_COMPLETE
- app-cluster** - Created on 2019-10-22 16:09:37 UTC+0530, status: CREATE_COMPLETE
- iam** - Created on 2019-10-22 15:59:04 UTC+0530, status: CREATE_COMPLETE
- vpc** - Created on 2019-10-22 15:59:04 UTC+0530, status: CREATE_IN_PROGRESS

Outputs (1)

| Key | Value | Description | Export name |
|-------------|--|--------------------|------------------|
| ApiEndpoint | http://ecs-services-1097299462.us-east-1.elb.amazonaws.com/api/books | Books API Endpoint | BooksApiEndpoint |

api.yml

AWSTemplateFormatVersion: '2010-09-09'

Description: container on ecs cluster

Resources:

Task:

Type: AWS::ECS::TaskDefinition

Properties:

Family: apis

Cpu: 256

Memory: 512

NetworkMode: aws vpc

RequiresCompatibilities:

- FARGATE

ExecutionRoleArn: !ImportValue ECSTaskExecutionRole

ContainerDefinitions:

- Name: books

Image: 287727666564.dkr.ecr.us-east-1.amazonaws.com/rk_reddy

Cpu: 256

Memory: 512

PortMappings:

- ContainerPort: 4567

Protocol: tcp

LogConfiguration:

LogDriver: awslogs

Options:

awslogs-group: 'apis'

awslogs-region: !Ref AWS::Region

awslogs-stream-prefix: 'books'

Service:

Type: AWS::ECS::Service

DependsOn: ListenerRule

Properties:

ServiceName: books-service

TaskDefinition: !Ref Task

Cluster: !ImportValue 'ECSCluster'

LaunchType: FARGATE

DesiredCount: 2

DeploymentConfiguration:

MaximumPercent: 200

MinimumHealthyPercent: 70

NetworkConfiguration:

AwsVpcConfiguration:

AssignPublicIp: ENABLED

Subnets:

- !ImportValue Subnet1

- !ImportValue Subnet2

SecurityGroups:

- !ImportValue ContainerSecurityGroup

LoadBalancers:

- ContainerName: books

ContainerPort: 4567

TargetGroupArn: !Ref TargetGroup

TargetGroup:

Type: AWS::ElasticLoadBalancingV2::TargetGroup

Properties:

Name: books-tg

VpcId: !ImportValue VPC

Port: 80

Protocol: HTTP

Matcher:

HttpCode: 200-299

HealthCheckIntervalSeconds: 10

HealthCheckPath: /stat

HealthCheckProtocol: HTTP

HealthCheckTimeoutSeconds: 5

HealthyThresholdCount: 10

TargetType: ip

ListenerRule:

Type: AWS::ElasticLoadBalancingV2::ListenerRule

Properties:

ListenerArn: !ImportValue Listener

Priority: 2

Conditions:

- Field: path-pattern

Values:

- /api/book*

Actions:

- TargetGroupArn: !Ref TargetGroup

Type: forward

Outputs:

ApiEndpoint:

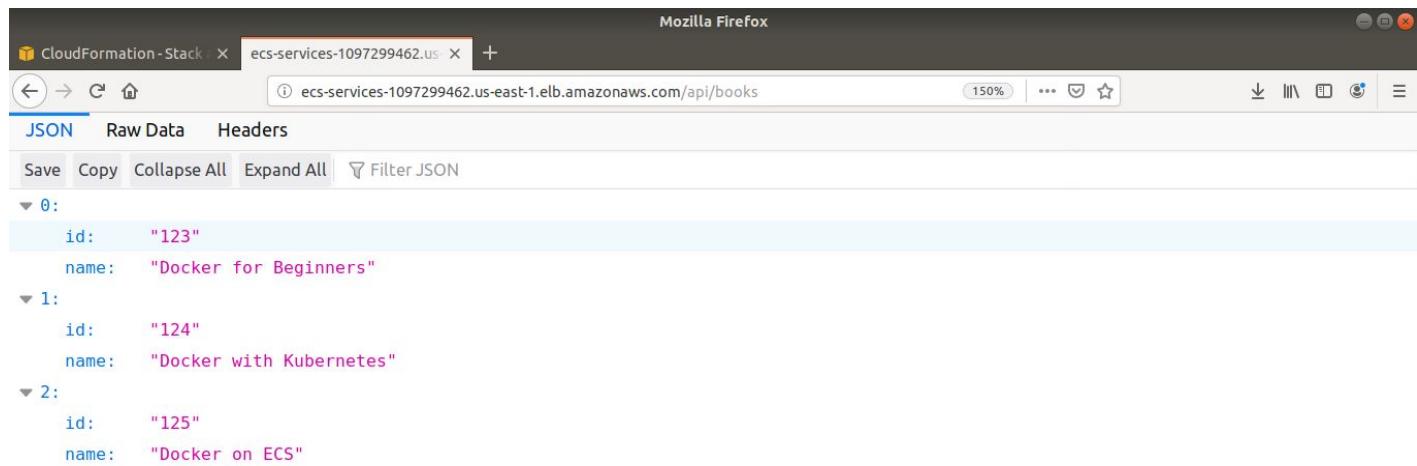
Description: Books API Endpoint

Value: !Join [", [http://", !ImportValue DomainName, '/api/books']]

Export:

Name: 'Books Api Endpoint'

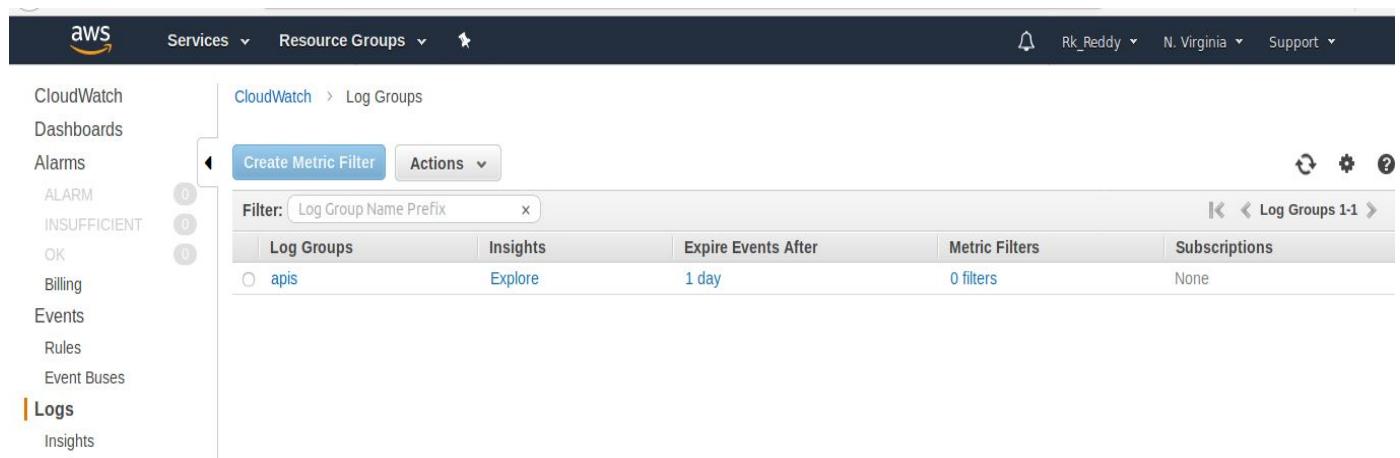
Copy the url under api stack and search in the browser.



The screenshot shows a Mozilla Firefox browser window. The address bar displays the URL: `ecs-services-1097299462.us-east-1.elb.amazonaws.com/api/books`. The main content area shows a JSON array with three items:

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
▼ 0:
  id: "123"
  name: "Docker for Beginners"
▼ 1:
  id: "124"
  name: "Docker with Kubernetes"
▼ 2:
  id: "125"
  name: "Docker on ECS"
```

Below picture shows the Log Group under Logs of cloudwatch. We created a group named **apis**.



The screenshot shows the AWS CloudWatch Logs interface. The left sidebar is collapsed, showing options like CloudWatch, Dashboards, Alarms, Events, Rules, Event Buses, and Logs. The Logs option is selected and highlighted in orange. The main pane shows the 'Log Groups' section for the 'CloudWatch' region. A filter bar at the top has the text 'Log Group Name Prefix' with a dropdown set to 'apis'. Below the filter is a table with the following columns: Log Groups, Insights, Expire Events After, Metric Filters, and Subscriptions. There is one entry in the table:

| Log Groups | Insights | Expire Events After | Metric Filters | Subscriptions |
|------------|----------|---------------------|----------------|---------------|
| apis | Explore | 1 day | 0 filters | None |

The screenshot shows the AWS CloudWatch Log Groups interface. At the top, there are navigation links for 'Services' and 'Resource Groups'. On the right, there are user details: 'Rk Reddy', 'N. Virginia', and 'Support'. Below the navigation, the path 'CloudWatch > Log Groups > Streams for apis' is shown. There are three buttons at the top: 'Search Log Group', 'Create Log Stream', and 'Delete Log Stream'. A search bar labeled 'Filter: Log Stream Name Prefix' is present. To the left, there are three circular icons with the numbers 0, 0, and 0. The main area displays a table with two rows of log streams. The columns are 'Log Streams' and 'Last Event Time'. The first row has a link 'books/books/c361605f-5d89-4e73-8780-53b95afeddc0' with a timestamp '2019-10-22 16:36 UTC+5:30'. The second row has a link 'books/books/d3425d3c-ae57-4420-adb1-bab4cb8a3652' with a timestamp '2019-10-22 16:31 UTC+5:30'. Navigation arrows and a help icon are at the top right.

These are the log streams for log group **apis**

The screenshot shows the AWS ECS Dashboard. On the left, there is a sidebar with links: 'Account Settings', 'Amazon EKS', 'Clusters', 'Amazon ECR', 'Repositories', 'AWS Marketplace', 'Discover software', and 'Subscriptions'. The main area has a heading 'For more information, see the ECS documentation.' and two buttons: 'Create Cluster' and 'Get Started'. Below this, there is a section titled 'bookstore > FARGATE'. It shows the status of the 'bookstore' service: 'CloudWatch monitoring' and 'Default Monitoring' are checked. Under the 'FARGATE' section, it shows '1 Services', '2 Running tasks', and '0 Pending tasks'. Below this, under 'EC2', it shows '0 Services', '0 Running tasks', '0 Pending tasks', 'No data' for CPUUtilization, 'No data' for MemoryUtilization, and '0 Container instances'. Navigation arrows and a 'view all' button are at the top right.

Amazon ECS - Mozilla Firefox

Amazon ECS Services Resource Groups

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

Create Cluster Get Started

View list card

bookstore > CloudWatch monitoring
Default Monitoring

FARGATE

| 1 | 2 | 0 |
|----------|---------------|---------------|
| Services | Running tasks | Pending tasks |

EC2

1 - 1 of 1

Finally, We successfully created the ECS cluster **bookstore**.

Services Tasks ECS Instances Metrics Scheduled Tasks Tags

Run new Task Stop Stop All Actions Last updated on October 22, 2019 4:51:22 PM (0m ago) [Edit](#) [?](#)

Desired task status: [Running](#) [Stopped](#)

Filter in this page Launch type ALL Page size 50

| <input type="checkbox"/> | Task | Task definit... | Container i... | Last status | Desired sta... | Started By | Group | Launch type | Platform ve... |
|--------------------------|----------------|-----------------|----------------|-------------|----------------|----------------|-----------------|-------------|----------------|
| <input type="checkbox"/> | c361605f-5d... | apis:1 | -- | RUNNING | RUNNING | ecs-svc/922... | service:book... | FARGATE | 1.3.0 |
| <input type="checkbox"/> | d3425d3c-a... | apis:1 | -- | RUNNING | RUNNING | ecs-svc/922... | service:book... | FARGATE | 1.3.0 |

Mysql

Mysql remote access of host from another host using **docker container**.

> Launch an EC2 instance and install **docker** and **docker-compose** in it.

| Services | | Resource Groups | | | | Rk Reddy | N. Virginia | Support |
|---|--|---|--|---------------------------|--|----------|-------------|---------|
| | | | | | | | | |
| | | | | | | | | |
| Launch Instance | | Connect | | Actions ▾ | | | | |
| <input type="text"/> Filter by tags and attributes or search by keyword | | ? K < 1 to 2 of 2 > X | | | | | | |
| | | | | | | | | |

```
root@ip-172-31-94-124:/home/ec2-user
File Edit View Search Terminal Help
[root@ip-172-31-94-124 ec2-user]# yum install docker
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:18.06.1ce-10.32.amzn1 will be installed
--> Processing Dependency: xfsprogs for package: docker-18.06.1ce-10.32.amzn1.x86_64
--> Processing Dependency: pigz for package: docker-18.06.1ce-10.32.amzn1.x86_64
--> Processing Dependency: libseccomp.so.2()(64bit) for package: docker-18.06.1ce-10.32.amzn1.x86_64
--> Processing Dependency: libltdl.so.7()(64bit) for package: docker-18.06.1ce-10.32.amzn1.x86_64
--> Running transaction check
--> Package libseccomp.x86_64 0:2.3.1-2.4.amzn1 will be installed
--> Package libtool-ltdl.x86_64 0:2.4.2-20.4.8.5.32.amzn1 will be installed
--> Package pigz.x86_64 0:2.3.3-1.6.amzn1 will be installed
--> Package xfsprogs.x86_64 0:4.5.0-18.23.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch    Version            Repository      Size
=====
Installing:
=====
```

```
root@ip-172-31-94-124:/home/ec2-user
File Edit View Search Terminal Help
[root@ip-172-31-94-124 ec2-user]# sudo curl -L "https://github.com/docker/compose/releases/download/1.22.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload  Upload Total Spent   Left  Speed
100  617     0  617     0      0  6170      0 --::-- --::-- --::--  6170
100 11.2M  100 11.2M     0      0  39.8M      0 --::-- --::-- --::--  39.8M
[root@ip-172-31-94-124 ec2-user]# sudo mv /usr/local/bin/docker-compose /usr/bin/docker-compose
[root@ip-172-31-94-124 ec2-user]# sudo chmod +x /usr/bin/docker-compose
[root@ip-172-31-94-124 ec2-user]# yum install docker-compose
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                         | 2.1 kB  00:00:00
amzn-updates                                      | 2.5 kB  00:00:00
No package docker-compose available.
Error: Nothing to do
[root@ip-172-31-94-124 ec2-user]# sudo apt-get install docker-compose
sudo: apt-get: command not found
[root@ip-172-31-94-124 ec2-user]# sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
ln: failed to create symbolic link '/usr/bin/docker-compose': File exists
[root@ip-172-31-94-124 ec2-user]# docker-co
docker-compose          docker-containerd      docker-containerd-shim
[root@ip-172-31-94-124 ec2-user]# docker-co
docker-compose          docker-containerd      docker-containerd-shim
```

```
root@ip-172-31-94-124:/home/ec2-user
File Edit View Search Terminal Help
[root@ip-172-31-94-124 ec2-user]# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
80369df48736: Pull complete
e8f52315cb10: Pull complete
cf2189b391fc: Pull complete
cc98f645c682: Pull complete
27a27ac83f74: Pull complete
fa1f04453414: Pull complete
d45bf7d22d33: Pull complete
3dbac26e409c: Pull complete
9017140fb8c1: Pull complete
b76dda2673ae: Pull complete
bea9eb46d12a: Pull complete
elf050a38d0f: Pull complete
Digest: sha256:7345ce4ce6f0c1771d01fa333b8edb2c606ca59d385f69575f8e3e2ec6695eee
Status: Downloaded newer image for mysql:latest
[root@ip-172-31-94-124 ec2-user]#
```

> Allow mysql service in security group of the instance.

The screenshot shows the AWS EC2 Security Groups interface with the 'Edit inbound rules' dialog box open. The dialog box contains four inbound rule entries:

| Type | Protocol | Port Range | Source | Description |
|------------|----------|------------|--------------------------|----------------------------|
| HTTP | TCP | 80 | Custom 0.0.0.0/0 | e.g. SSH for Admin Desktop |
| HTTP | TCP | 80 | Custom ::/0 | e.g. SSH for Admin Desktop |
| SSH | TCP | 22 | Custom 0.0.0.0/0 | e.g. SSH for Admin Desktop |
| Custom TCF | TCP | 3306 | Anywhere 0.0.0.0/0, ::/0 | e.g. SSH for Admin Desktop |

Below the table is a note: "NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created." At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

> Pull mysql image from the docker hub.

```
delta@delta-Lenovo-ideapad-320-15IKB: ~/Downloads
File Edit View Terminal Tabs Help
delta@delta-Lenovo-ideapad-320-15IKB: ~/Downloads x delta@delta-Lenovo-ideapad-320-15IKB: ~/Downloads x
[root@ip-172-31-94-124 ec2-user]# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
80369df48736: Pull complete
e8f52315cb10: Pull complete
cf2189b391fc: Pull complete
cc98f645c682: Pull complete
27a27ac83f74: Pull complete
fa1f04453414: Pull complete
d45bf7d22d33: Pull complete
3dbac26e409c: Pull complete
9017140fb8c1: Pull complete
b76dda2673ae: Pull complete
bea9eb46d12a: Pull complete
elf050a38d0f: Pull complete
Digest: sha256:7345ce4ce6f0c1771d01fa333b8edb2c606ca59d385f69575f8e3e2ec6695eee
Status: Downloaded newer image for mysql:latest
[root@ip-172-31-94-124 ec2-user]# yum install mysql
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main
amzn-updates
Resolving Dependencies
--> Running transaction check
```

```
root@ip-172-31-94-124:/home/ec2-user
File Edit View Search Terminal Tabs Help
root@ip-172-31-94-124:/home/ec2-user x delta@delta-Lenovo-ideapad-320-15IKB:~/Downloads x
[root@ip-172-31-94-124 ec2-user]# yum install mysql-server
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
--> Package mysql-server.noarch 0:5.5-1.6.amzn1 will be installed
--> Processing Dependency: mysql55-server >= 5.5 for package: mysql-server-5.5-1.6.amzn1.noarch
--> Running transaction check
--> Package mysql55-server.x86_64 0:5.5.62-1.23.amzn1 will be installed
--> Processing Dependency: perl-DBD-MySQL(mysql55) for package: mysql55-server-5.5.62-1.23.amzn1.x86_64
--> Processing Dependency: perl(Data::Dumper) for package: mysql55-server-5.5.62-1.23.amzn1.x86_64
--> Processing Dependency: perl(DBI) for package: mysql55-server-5.5.62-1.23.amzn1.x86_64
--> Processing Dependency: perl(DBI) for package: mysql55-server-5.5.62-1.23.amzn1.x86_64
--> Running transaction check
--> Package perl-DBD-MySQL55.x86_64 0:4.023-5.23.amzn1 will be installed
--> Package perl-DBI.x86_64 0:1.627-4.8.amzn1 will be installed
--> Processing Dependency: perl(RPC::PlServer) >= 0.2001 for package: perl-DBI-1.627-4.8.amzn1.x86_64
--> Processing Dependency: perl(RPC::PlClient) >= 0.2000 for package: perl-DBI-1.627-4.8.amzn1.x86_64
--> Package perl-Data-Dumper.x86_64 0:2.145-3.5.amzn1 will be installed
```

```
root@ip-172-31-94-124:/home/ec2-user
File Edit View Search Terminal Tabs Help
root@ip-172-31-94-124:/home/ec2-user x delta@delta-Lenovo-ideapad-320-15IKB:~/Downloads x
[root@ip-172-31-94-124 ec2-user]# service mysqld start
Initializing MySQL database: Installing MySQL system tables...
191017  9:53:00 [Note] Ignoring --secure-file-priv value as server is running with --bootst
rap.
191017  9:53:00 [Note] /usr/libexec/mysql55/mysqld (mysqld 5.5.62) starting as process 3328
...
OK
Filling help tables...
191017  9:53:00 [Note] Ignoring --secure-file-priv value as server is running with --bootst
rap.
191017  9:53:00 [Note] /usr/libexec/mysql55/mysqld (mysqld 5.5.62) starting as process 3335
...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

/usr/libexec/mysql55/mysqladmin -u root password 'new-password'
/usr/libexec/mysql55/mysqladmin -u root -h ip-172-31-94-124 password 'new-password'
```

> Create the credentials of Root user and login to mysql using those credentials

```
root@ip-172-31-94-124:/home/ec2-user
File Edit View Search Terminal Tabs Help
root@ip-172-31-94-124:/home/ec2-user x delta@delta-Lenovo-ideapad-320-15IKB: ~/Downloads x
[root@ip-172-31-94-124 ~]# mysqladmin -u root password 5689
[root@ip-172-31-94-124 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
Bye
[root@ip-172-31-94-124 ~]# mysqladmin -u root -p create database1
Enter password:
[root@ip-172-31-94-124 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 7
```

> DataBase's

```
mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| database1      |
| mysql          |
| performance_schema |
| test           |
+-----+
5 rows in set (0.00 sec)
```

> Create one database and login through root user.

```
root@ip-172-31-94-124:/home/ec2-user
File Edit View Search Terminal Tabs Help
root@ip-172-31-94-124:/home/ec2-user x delta@delta-Lenovo-ideapad-320-15IKB: ~/Downloads x
bye
[root@ip-172-31-94-124 ~]# mysqladmin -u root -p create database1
Enter password:
[root@ip-172-31-94-124 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show dbs
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
to your MySQL server version for the right syntax to use near 'dbs' at line 1
mysql> SHOW DATABASES;
+-----+
```

> Create one table and add some arguments. Here, I created RkReddy.

```

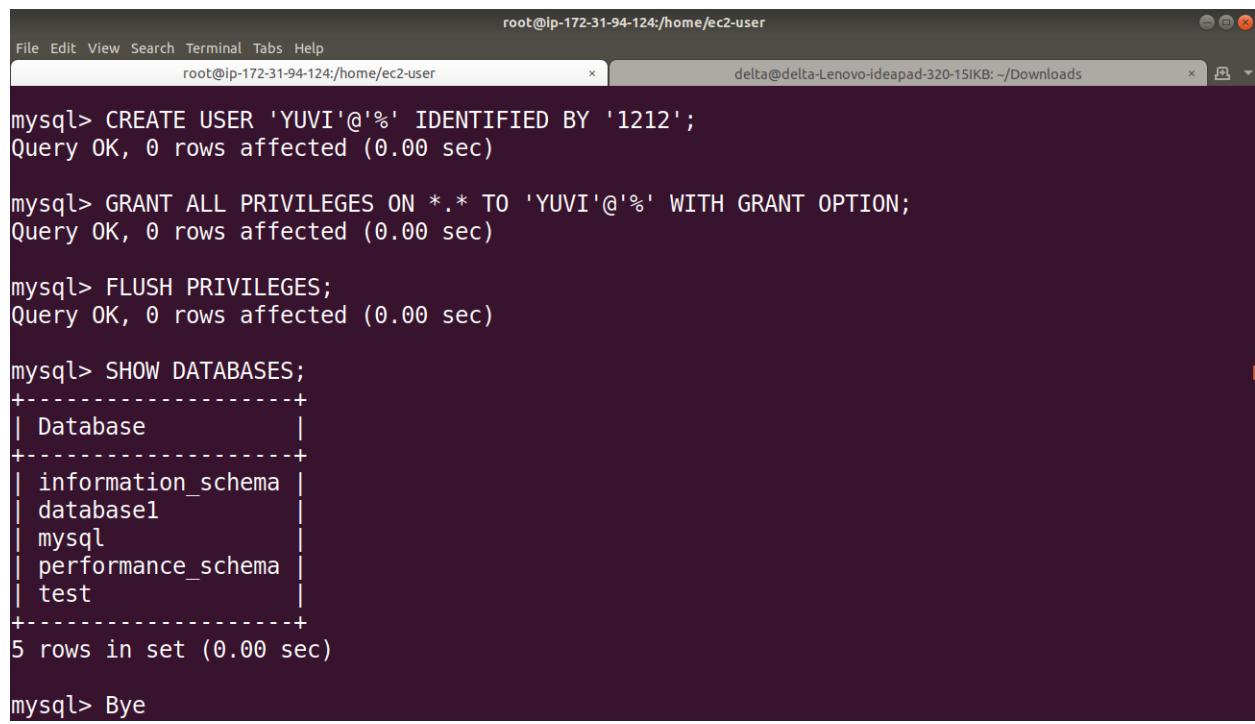
mysql> use database1
Database changed
mysql> CREATE TABLE RkReddy (name VARCHAR(100), roll_no INT);
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO RkReddy(name,roll_no) values("RkR",12);
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * from RkReddy;
+-----+-----+
| name | roll_no |
+-----+-----+
| RkR  |      12 |
+-----+-----+
1 row in set (0.00 sec)

```

> Assign privileges to the user to get access through another user.



The screenshot shows a terminal window with two tabs. The active tab is titled 'root@ip-172-31-94-124:/home/ec2-user' and contains the following MySQL session:

```

root@ip-172-31-94-124:/home/ec2-user
File Edit View Search Terminal Tabs Help
root@ip-172-31-94-124:/home/ec2-user x delta@delta-Lenovo-ideapad-320-15IKB: ~/Downloads x
mysql> CREATE USER 'YUVI'@'%' IDENTIFIED BY '1212';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'YUVI'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| database1      |
| mysql          |
| performance_schema |
| test           |
+-----+
5 rows in set (0.00 sec)

mysql> Bye

```

> Launch another instance

> Use **mysql -h host ip -u username -p** and enter password to login

```
delta@delta-Lenovo-ideapad-320-15IKB:~/Downloads$ ssh -i a.pem ec2-user@52.55.86.42
The authenticity of host '52.55.86.42 (52.55.86.42)' can't be established.
ECDSA key fingerprint is SHA256:1MdXW1dNdz1Uz6rV678gFaQA0SSLpw030gS4/HcCrNo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.55.86.42' (ECDSA) to the list of known hosts.

 _|_(_|_) /   Amazon Linux AMI
 __|\_\_|_\_|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
4 package(s) needed for security, out of 10 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-83-253 ~]$ sudo su
[root@ip-172-31-83-253 ec2-user]# yum install mysql-server
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                         | 2.1 kB  00:00:00
amzn-updates                                     | 2.5 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package mysql-server.noarch 0:5.5-1.6.amzn1 will be installed
--> Processing Dependency: mysql55-server >= 5.5 for package: mysql-server-5.5-1.6.amzn1.no
arch
```

```
[root@ip-172-31-83-253 ec2-user]# service mysqld start
Initializing MySQL database:  Installing MySQL system tables...
191017 10:06:42 [Note] Ignoring --secure-file-priv value as server is running with --bootst
rap.
191017 10:06:42 [Note] /usr/libexec/mysql55/mysqld (mysqld 5.5.62) starting as process 2976
...
OK
Filling help tables
```

```
[root@ip-172-31-83-253 ec2-user]# mysql -h 54.227.77.195 -u YUVI -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.5.62 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES
```

> here, We are inside sql monitor. Next Type **SHOW DATABASES;** to check DB's.

```
mysql> SHOW DATABASES
-> ;
+-----+
| Database      |
+-----+
| information_schema |
| database1      |
| mysql          |
| performance_schema |
| test           |
+-----+
5 rows in set (0.00 sec)

mysql> Bye
```

> Here, We can check the all DB's resides inside the main host using the command.

```
mysqldump -u root -p --all-databases > all.sql
```