

# CLOUD LAB

## What is Docker?

Docker is a platform that allows you to develop, ship, and run applications in isolated environments called **containers**. Containers package applications and their dependencies together, ensuring that they run consistently across different environments. It eliminates the "works on my machine" problem.

## How Does Docker Work?

Docker uses **containerization** to isolate applications. It works by:

1. Using **Docker Engine**, which runs on the host system to manage containers.
2. Utilizing **Docker Images**, which are pre-packaged applications with all dependencies.
3. Running **Docker Containers**, which are instances of Docker images.
4. Using **Docker Hub**, a repository of pre-built images that you can pull and run.

---

## Steps to Use Docker on Kali Linux

### 1. Install Docker on Kali Linux

Run the following commands to install Docker:

```
sudo apt update  
sudo apt install docker.io -y
```

Enable and start the Docker service:

```
sudo systemctl enable --now docker
```

Verify installation:

```
docker --version
```

## 2. Verify Docker Installation

Check if Docker is working properly:

```
docker run hello-world
```

This downloads a small test image and runs it in a container.

---

# Running a Container in Docker

## 1. Pull an Image from Docker Hub

You can pull images from [Docker Hub](#). For example, to pull an **Ubuntu** image:

```
docker pull ubuntu
```

## 2. Run a Container

Start a container with an interactive shell:

```
docker run -it ubuntu bash
```

This starts a new Ubuntu container and opens a bash shell inside it.

## 3. List Running Containers

To see running containers:

```
docker ps
```

To see all containers (including stopped ones):

```
docker ps -a
```

## 4. Stop and Remove Containers

To stop a running container:

```
docker stop <container_id>
```

To remove a container:

```
docker rm <container_id>
```

## 5. Remove an Image

To remove a downloaded image:

```
docker rmi <image_name>
```

## Example: Running Nginx on Kali Linux

If you want to run a web server using **Nginx**, follow these steps:

1. Pull the Nginx image:

```
docker pull nginx
```

2. Run the Nginx container:

```
docker run -d -p 8080:80 nginx
```

3. Open a browser and visit:

```
http://localhost:8080
```

You should see the default Nginx welcome page.

4. Stop the Nginx container:

```
docker stop <container_id>
```

## Running DVWA (Damn Vulnerable Web Application) Using Docker on Kali Linux

DVWA is a deliberately vulnerable web application for security testing. You can quickly set it up using Docker.

## Step 1: Pull the DVWA Docker Image

Run the following command to download the DVWA image:

```
docker pull vulnerables/web-dvwa
```

## Step 2: Run DVWA Container

Now, start a new DVWA container using the command below:

```
docker run -d -p 8080:80 vulnerables/web-dvwa
```

- `-d` : Runs the container in detached mode (in the background).
- `-p 8080:80` : Maps port 8080 on your Kali Linux machine to port 80 inside the container.

## Step 3: Access DVWA

Open your web browser and go to:

```
http://localhost:8080
```

If running on a remote server, use:

```
http://<your-ip>:8080
```

## Step 4: Login to DVWA

Use the default login credentials:

- **Username:** `admin`
- **Password:** `password`

Click **Login**, and you should now be inside DVWA.

## Step 5: Configure DVWA

Once logged in:

1. Go to **DVWA Security** settings.

2. Set **Security Level** to **Low** to enable testing.
  3. Click **Create/Reset Database** under **Database Setup**.
- 

## Step 6: Managing the DVWA Container

### Check Running Containers

```
docker ps
```

### Stop the Container

```
docker stop <container_id>
```

### Restart the Container

```
docker start <container_id>
```

### Remove the Container

```
docker rm <container_id>
```

### Remove the Image (if needed)

```
docker rmi vulnerables/web-dvwa
```

## DC4 using docker

### Step 1: Extract the OVA File

First, extract the **DC-4.ova** file to get the virtual disk (**.vmdk**) inside it.

```
tar -xvf DC-4.ova
```

This will extract files like:

- **DC-4.ovf** (Open Virtualization Format descriptor)
- **DC-4.vmdk** (Virtual disk file)

The terminal window shows the following command sequence:

```
(kali㉿kali)-[~]
$ cd Downloads
(kali㉿kali)-[~/Downloads]
$ ls
DC-4.ova  DC-4.zip
(kali㉿kali)-[~/Downloads]
$ tar -xvf DC-4.ova
(kali㉿kali)-[~/Downloads]
$ sudo apt update && sudo apt install -y qemu-utils
[sudo] password for kali:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1543 packages can be upgraded. Run 'apt list --upgradable' to see them.
Installing:
qemu-utils

Installing dependencies:
ibverbs-providers  libboost-iostreams1.83.0  libfmt10  libgfprpc0  libglusterfs0  libiscsi7  librbd1      qemu-block-extra
libblkio1          libboost-thread1.83.0   libgfapi0  libgffd0  libibverbs1  librados2  librdmacm1t64

Summary:
Upgrading: 0, Installing: 16, Removing: 0, Not Upgrading: 1543
Download size: 22.6 MB
Space needed: 58.4 MB / 58.1 GB available
```

## Step 2: Convert VMDK to a Raw Image

Use qemu-img to convert the VMDK file to a raw image:

```
qemu-img convert -f vmdk -O raw DC-4.vmdk dc4.raw
```

The terminal window shows the following command sequence:

```
kali@kali: ~/Downloads
$ ls
dc4.raw  DC-4.ova  DC-4.ovf  DC-4.vmdk  DC-4.zip
(kali㉿kali)-[~/Downloads]
$ qemu-img convert -f vmdk -O raw DC-4.vmdk dc4.raw
qemu-img: Could not open 'DC-4.vmdk': Could not open 'DC-4.vmdk': No such file or directory
(kali㉿kali)-[~/Downloads]
$ qemu-img convert -f vmdk -O raw DC-4.vmdk dc4.raw
(kali㉿kali)-[~/Downloads]
$ ls
DC-4.ova  DC-4.ovf  dc4.raw  DC-4.vmdk  DC-4.zip
(kali㉿kali)-[~/Downloads]
$
```

### Step 3: Create a Docker Image from the Raw Disk

Now, create a Docker image using the raw disk file.

1. Create a directory for the image: **mkdir dc4-docker && cd dc4-docker**
2. Move the raw image into the directory: **mv ..dc4.raw .**
3. Create a Dockerfile in the dc4-docker directory: **nano Dockerfile**

```
(kali㉿kali)-[~/Downloads]
└─$ mkdir dc4-docker && cd dc4-docker
Network
(kali㉿kali)-[~/Downloads/dc4-docker]
└─$ mv ..dc4.raw .

(kali㉿kali)-[~/Downloads/dc4-docker]
└─$ ls
dc4.raw
```

##on terminal

```
sudo apt update && sudo apt install -y qemu-system-x86 qemu-utils
```

```
RUN apt update && apt install -y qemu-system-x86 wget
```

```
sudo apt update && sudo apt install -y docker.io qemu-system-x86\n
```

```
sudo systemctl start docker
```

```
sudo systemctl status docker
```

```
sudo systemctl enable --now docker
```

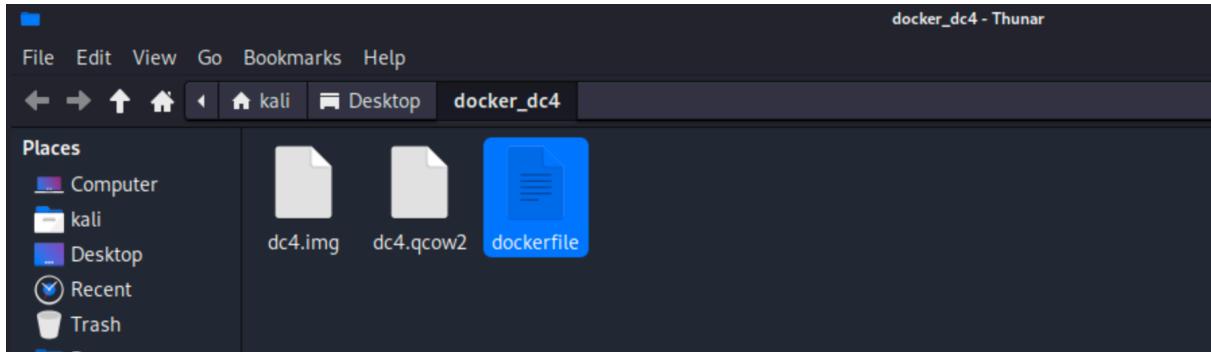
```
mkdir docker_dc4
```

```
#keep a copy of dc4.img and dc4.q2cow
```

```
#steps to extract .img
```

```
qemu-img convert -O raw DC4-disk001.vmdk dc4.img
```

```
sudo qemu-img convert -f raw -O qcow2 dc4.img dc4.qcow2
```



```
touch dockerfile
```

```
##paste this inside dockerfile
```

```
FROM ubuntu:latest
```

```
RUN apt update && apt install -y qemu-system-x86
```

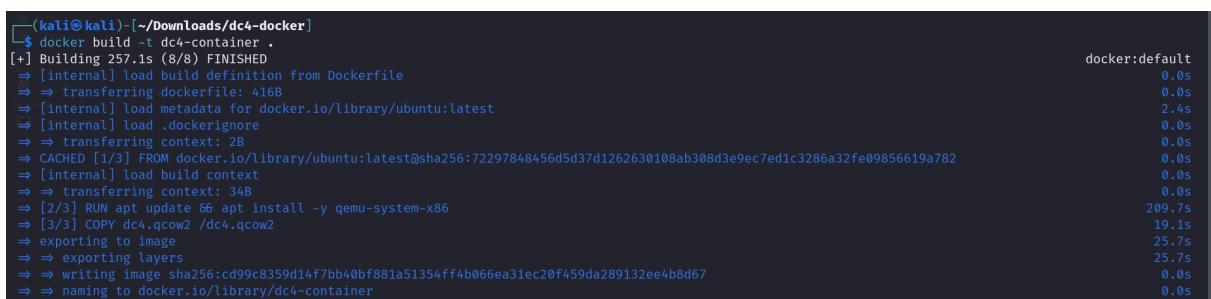
```
COPY dc4.qcow2 /dc4.qcow2
```

```
EXPOSE 80
```

```
CMD ["qemu-system-x86_64", "-m", "2048", "-hda", "/dc4.qcow2", "-net", "nic", "-net", "user,hostfwd=tcp::80::80", "-nographic"]
```

4. Build the Docker image:

```
docker build -t dc4-container .
```



## Step 4: Run the DC-4 Container

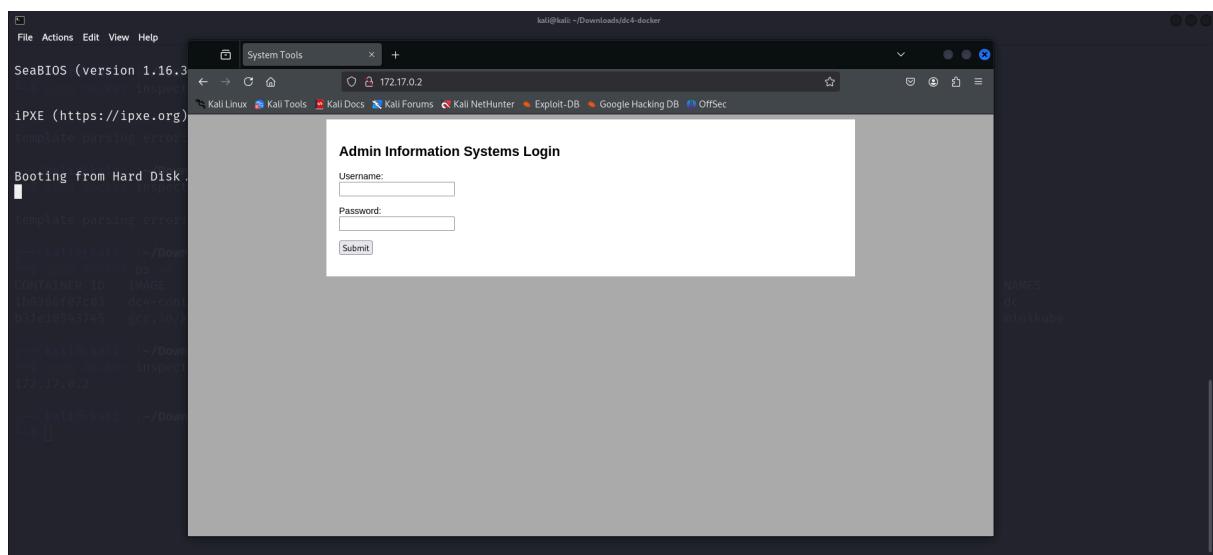
```
sudo docker run --name dc -it dc4-container
```

##it will start to boot

NEW TERMINAL;

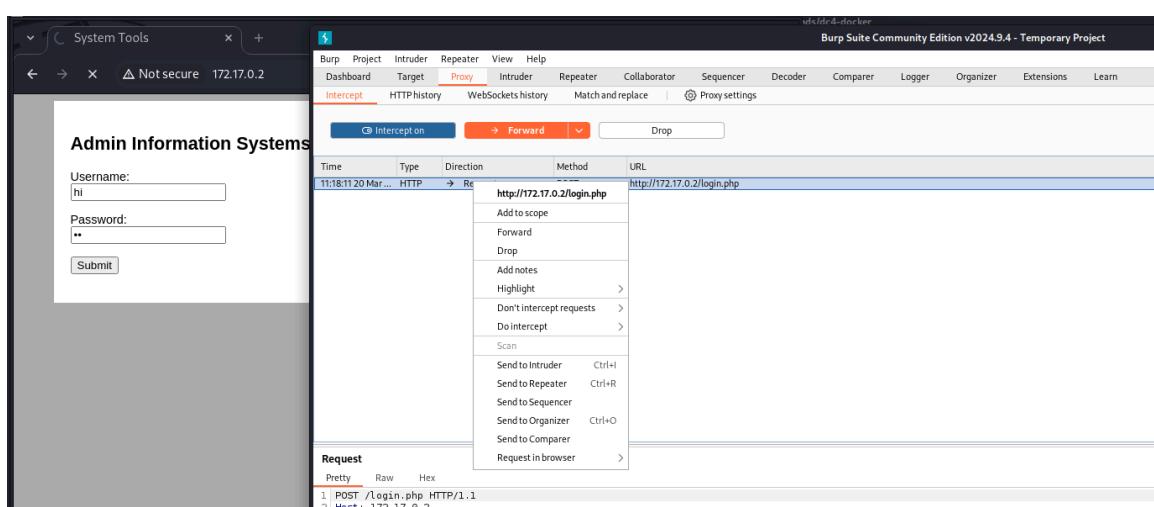
```
sudo docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}' dc4-container
```

copy ip address and paste it in mozilla



## Start burpsuit bruteforce attack

1. Send the login request to intruder



## 2. Add the section to add payload

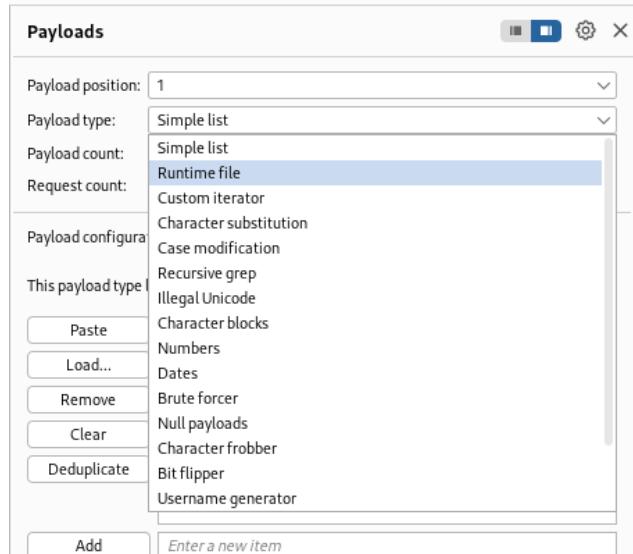
```
Add § Clear § Auto §  
1 POST /login.php HTTP/1.1  
2 Host: 172.17.0.2  
3 Content-Length: 23  
4 Cache-Control: max-age=0  
5 Accept-Language: en-US,en;q=0.9  
6 Origin: http://172.17.0.2  
7 Content-Type: application/x-www-form-ur  
8 Upgrade-Insecure-Requests: 1  
9 User-Agent: Mozilla/5.0 (Windows NT 10.  
10 Accept: text/html,application/xhtml+xml  
11 Referer: http://172.17.0.2/  
12 Accept-Encoding: gzip, deflate, br  
13 Cookie: PHPSESSID=j2blakm4rh5uq22jb5j04  
14 Connection: keep-alive  
15  
16 username=$hi$&password=$www$
```

## 3. Use the required type of brute force attack and payload type

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. A 'Sniper attack' is chosen from the dropdown menu. The payload configuration panel on the right is open, showing 'Payloads' with a 'Simple list' type set to 'All payload positions'. The payload list contains the value '\$hi\$&password=\$www\$'. The main window displays the original request and the modified payload.

```
Burp Suite Community Edition v2024.9.4 - Temporary Project  
Burp Project Intruder Repeater View Help  
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn  
4 x 5 x +  
① Sniper attack  
Sniper attack  
Inserts each payload into each position one at a time, using a single payload set.  
header to match target  
Target  
② Battering ram attack  
Battering ram attack  
Simultaneously places the same payload into all positions, using a single payload set.  
③ Pitchfork attack  
Pitchfork attack  
1. pos Allocates a payload set to each position. Intruder iterates through each set in parallel.  
2. Header Header  
3. Content-Type Content-Type  
4. Cache-Control Cache-Control  
5. Accept Accept  
6. Origin Origin  
7. Content-Type Content-Type: application/x-www-form-urlencoded  
8. Upgrade-Insecure-Requests Upgrade-Insecure-Requests: 1  
9. User-Agent User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36  
10. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7  
11. Referer: http://172.17.0.2/  
12. Accept-Encoding: gzip, deflate, br  
13. Cookie: PHPSESSID=j2blakm4rh5uq22jb5j045suq5  
14. Connection: keep-alive  
15  
16 username=$hi$&password=$www$
```

**Payloads**  
Payload position: All payload positions  
Payload type: Simple list  
Payload count: 0  
Request count: 0  
Payload configuration  
This payload type lets you configure a simple list of strings that are used as payloads.  
Paste  
Load...  
Remove  
Clear  
Duplicate  
Add Enter a new item  
Add from list... [Pro version only]



#### 4. Add wordlist

#### 5. Start attack check the payload for where status code is 200 you have the password Congrats 😎

Screenshot of the Burp Suite interface showing an intruder attack results table and a captured response message.

**Intruder attack results table:**

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
1 POST	72 happy	toor	302	4	5	504	6	641
2 Host	73 hello	happy	302	5	12	504	8	641
3 Content	75 password	happy	302	8	7	606	7	641
4 Content	76 admin	happy	302	7	6	665	6	641
5 Access	77 user	happy	200	6	6	641	6	641
6 Access	78 root	happy	200	6	8	641	8	641
7 Content	79 pass	happy	200	8	8	641	8	641
8 Content	80 toor	happy	200	5	5	641	5	641
9 User	81 happy	happy	200	5	5	641	5	641

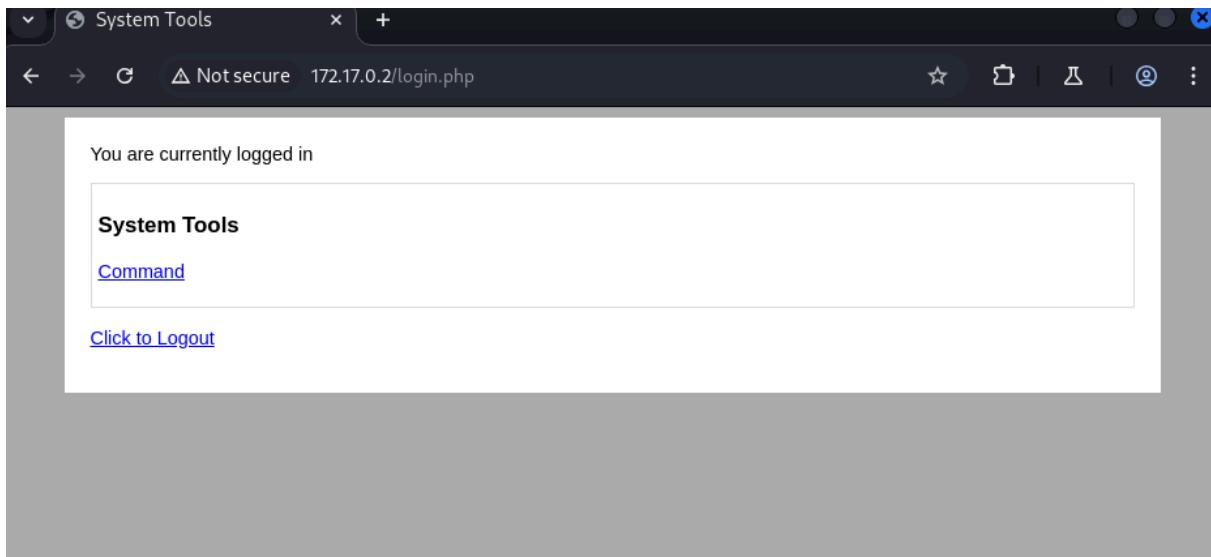
**Captured Response Message:**

```

HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 641
Connection: close
Date: Mon, 11 Mar 2024 11:13:42 GMT
Server: Apache/2.4.41 (Ubuntu)

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <div class="container">
        <div class="inner">
            You are currently logged in<br>
            <a href="menu-system">System Tools</a>
            <a href="command.php">Command</a>
            <a href="logout.php">Click to Logout</a>
        </div>
    </div>
</body>
</html>

```



## How to push image to docker Hub?

1. Create an account in docker hub
2. Create a repository
3. Login to the docker hub account from local using **docker login**
4. If your image is not already tagged correctly, tag it using: **docker tag <local-image-id> kanusharao/cloud-lab:tagname**
5. Find your **local image ID** using: **docker images**
6. Run the following command to push the image: **docker push kanusharao/cloud-lab:tagname**
7. Repo Link: <https://hub.docker.com/repository/docker/kanusharao/cloud-lab/tags>

```
(kali㉿kali)-[~/Downloads/dc4-docker]
└─$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username: anusharao4520@gmail.com
Password:
WARNING! Your password will be stored unencrypted in /home/kali/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

(kali㉿kali)-[~/Downloads/dc4-docker]
└─$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
<none>              <none>   48a0db00d404  2 hours ago  2.42GB
dc4-container        latest   c5256f14b4ce  2 hours ago  2.42GB
<none>              <none>   cd99c8359d14  2 hours ago  2.42GB
<none>              <none>   147a9431e4e5  2 hours ago  2.42GB
dc4                 latest   e050eb2e5992  22 hours ago 5.44GB
<none>              <none>   2a3e68654697  22 hours ago 5.37GB
gcr.io/k8s-minikube/kicbase  v0.0.46  e72c4cbe9b29  2 months ago 1.31GB
```

```
(kali㉿kali)-[~/Downloads/dc4-docker]
└─$ docker tag c5256f14b4ce kanusharao/cloud-lab:dc4

(kali㉿kali)-[~/Downloads/dc4-docker]
└─$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
<none>              <none>   147a9431e4e5  2 hours ago  2.42GB
<none>              <none>   48a0db00d404  2 hours ago  2.42GB
kanusharao/cloud-lab        dc4     c5256f14b4ce  2 hours ago  2.42GB
dc4-container        latest   c5256f14b4ce  2 hours ago  2.42GB
<none>              <none>   cd99c8359d14  2 hours ago  2.42GB
dc4                 latest   e050eb2e5992  22 hours ago 5.44GB
<none>              <none>   2a3e68654697  22 hours ago 5.37GB
gcr.io/k8s-minikube/kicbase  v0.0.46  e72c4cbe9b29  2 months ago 1.31GB

(kali㉿kali)-[~/Downloads/dc4-docker]
└─$ docker push kanusharao/cloud-lab:dc4

The push refers to repository [docker.io/kanusharao/cloud-lab]
058270631f00: Pushed
738005c3537e: Pushed
4b7c01ed0534: Mounted from library/ubuntu
dc4: digest: sha256:9dd245707e91835fa2b36d1a043b77d83479fc16c9ebd3a0bacd9b6668b518e3 size: 955
```

The screenshot shows the Docker Hub interface for the 'cloud-lab' repository owned by 'kanusharao'. The left sidebar includes sections for 'Repositories', 'Settings', 'Billing', 'Usage', 'Pulls', and 'Storage'. The main content area displays repository details: 'kanusharao/cloud-lab' was last pushed 1 minute ago, with a repository size of 884.9 MB. It features tabs for 'General', 'Tags', 'Image Management (BETA)', 'Collaborators', 'Webhooks', and 'Settings'. Under the 'Tags' tab, there is a table showing tag information:

TAG	Digest	OS/ARCH	Last pull	Compressed size
<a href="#">dc4</a>	<a href="#">9dd245707e91</a>	linux/amd64	less than 1 day	884.9 MB

On the right, there are 'Docker commands' and a 'Public view' button. A command input field contains 'docker push kanusharao/cloud-lab :tagname'.

# AWS

After signup console:

The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' links: Billing and Cost Management, Service Quotas, IAM, and IAM Identity Center. The main area has a heading 'Applications (0)' with a 'Create application' button. Below it, there's a search bar for 'Find applications' and a table header for 'Name', 'Description', 'Region', and 'Origin'. A message says 'No applications' and 'Get started by creating an application.' with a 'Create application' button. At the bottom, there are links for 'View all services', 'CloudShell', 'Feedback', and copyright information.

NAVIGATE TO IAM:

The screenshot shows the AWS navigation sidebar. The 'Identity Management' (IAM) link is highlighted in blue. Other visible links include Analytics, Application Integration, Blockchain, Business Applications, Cloud Financial Management, Compute, Containers, Customer Enablement, Database, Developer Tools, End User Computing, Front-end Web & Mobile, Game Development, Internet of Things, Machine Learning, Management & Governance, Media Services, Migration & Transfer, Networking & Content Delivery, Quantum Technologies, Favorites, All applications, All services, and various access management sections like User Groups, Users, Roles, Policies, Identifiers, Accounts, Access keys, Access policies, External Identities, Unlinked identities, and Access grants.

add users , grant him certain permission/Policies

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group	Last activity	MFA	Password age	Console last sign-in
group_user	/	1	-	-	-	-
sid	/	1	-	-	-	-
user1	/	1	-	-	-	-
user3	/	0	-	-	-	-

create group with few policies

Identity and Access Management (IAM)

User groups (1) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Group name	Users	Permissions	Creation time
Group1	3	Defined	1 month ago

permission override for user in group

## Using IAM to assign a user to access S3 bucket

→ Go to IAM [root user]

→ create users

Screenshot of the AWS IAM 'Create user' wizard - Step 1: Specify user details.

The 'User name' field is set to 'test\_user1'. A note below it states: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +, . @ \_ - (hyphen)'. An optional checkbox 'Provide user access to the AWS Management Console - optional' is unchecked. A callout box notes: 'If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user.' Buttons at the bottom are 'Cancel' and 'Next'.

★ARN

Screenshot of the AWS IAM 'Users' page showing the 'test\_user1' user profile.

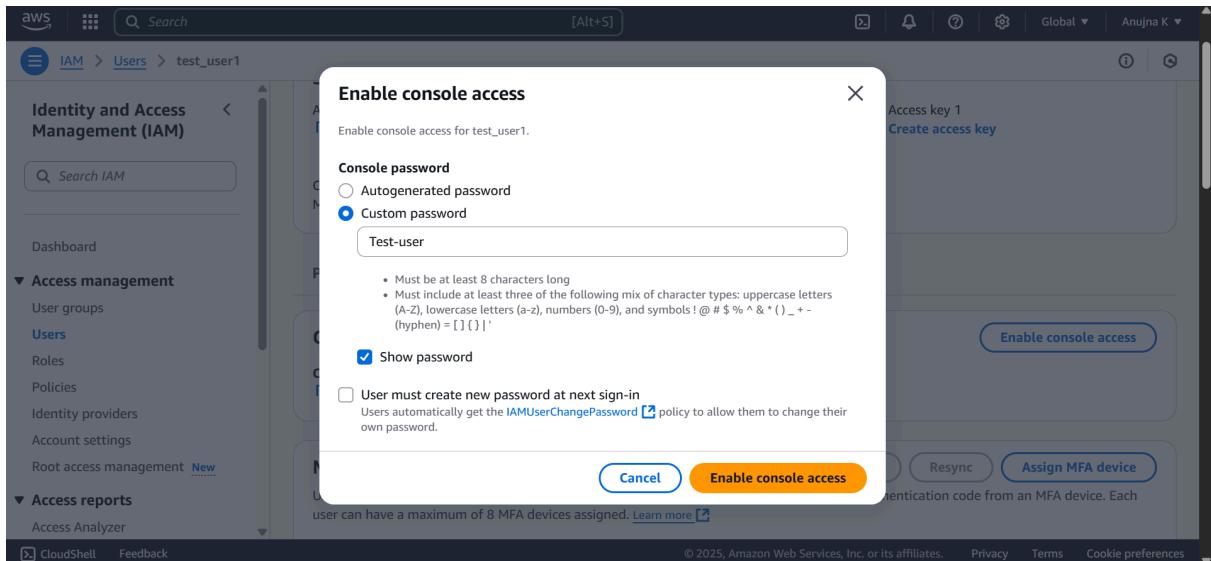
The 'Summary' section shows the ARN (arn:aws:iam::331409392843:user/test\_user1), console access status (Disabled), and access key information (Access key 1, Create access key). The 'Permissions' tab is selected, showing 'Permissions policies (0)' and a 'Permissions policies' table with a search bar and filter.

→ to enable console access to that user [test-user1]

Screenshot of the 'Security credentials' tab for the 'test\_user1' user.

The 'Console sign-in' section contains a 'Console sign-in link' (https://331409392843.signin.aws.amazon.com/console) and a 'Enable console access' button. The 'Console password' section indicates 'Not enabled'.

password → Test-user



## Console password

You have successfully enabled the user's new password.

This is the only time you can view this password. After you close this window, if the password is lost, you must create a new one.

### Console sign-in URL

<https://331409392843.signin.aws.amazon.com/console>

### User name

test\_user1

### Console password

Test-user [Hide](#)

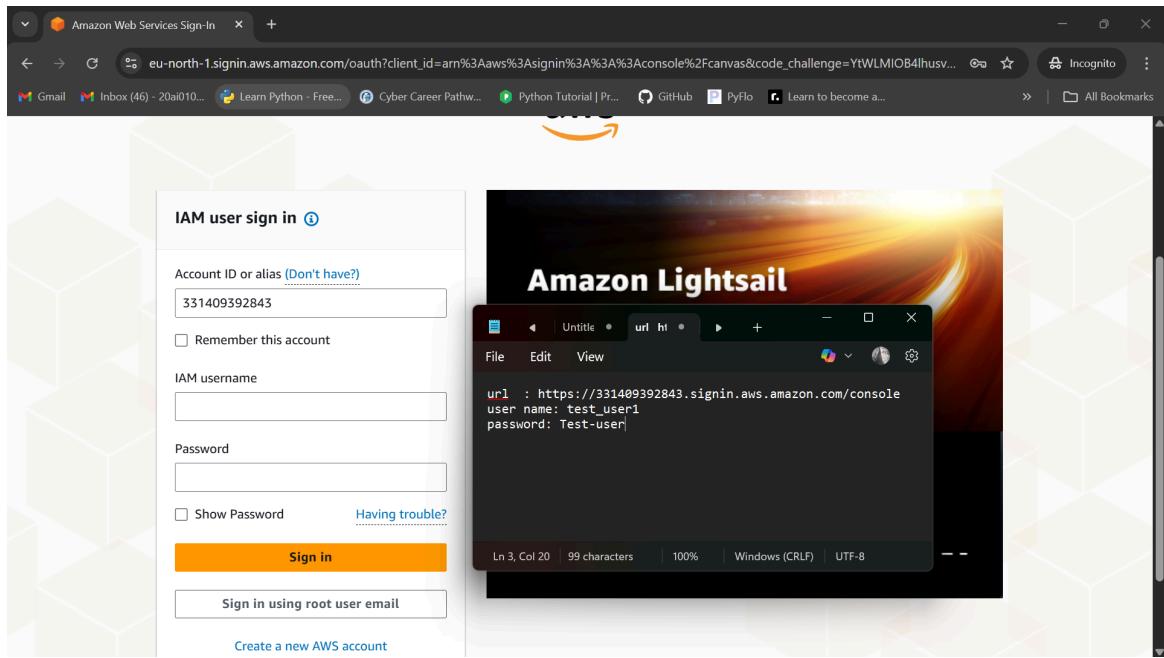
[Download .csv file](#)

[Close](#)

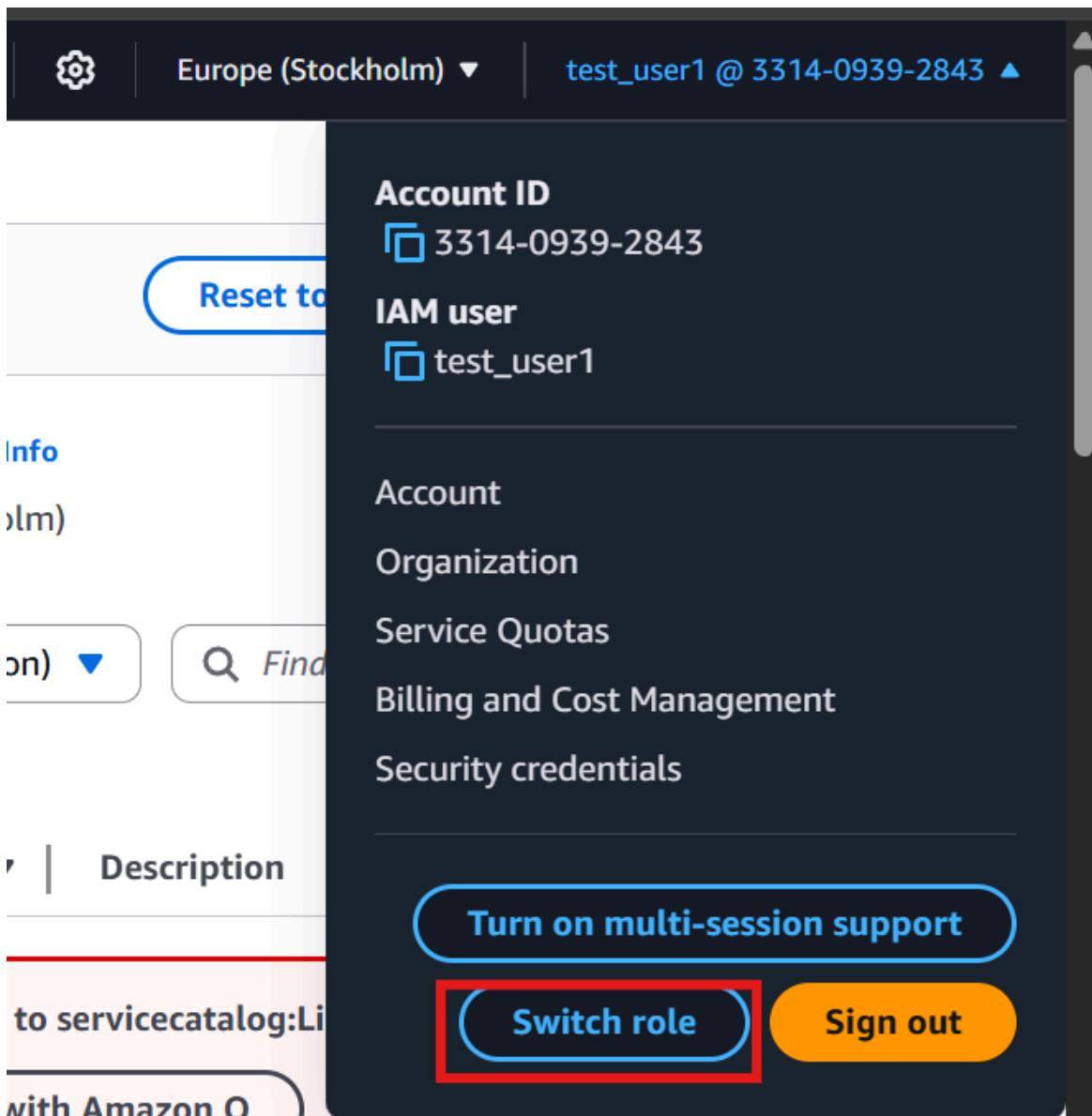
→ to perform login for this test user :

→ open a new window and copy the console sign-in url [can use incognito mode]

→ copy and paste username and password



The screenshot shows the 'Console Home' at [eu-north-1.console.aws.amazon.com/console/home?region=eu-north-1#](https://eu-north-1.console.aws.amazon.com/console/home?region=eu-north-1#). A modal window titled 'Service menu' is open, stating 'You can access all AWS services here. There are sections for recently visited and you can save your favorite services too.' Below the modal, there's a large cube icon and a message 'No recently visited services'. At the bottom, it says 'Explore one of these commonly visited AWS services.' with links to EC2, S3, Aurora and RDS, and Lambda. To the right, the 'Applications' section shows '(0)' applications with a 'Create application' button. A red box highlights an error message: 'Access denied to servicatalog>ListApplications' with a link to 'Diagnose with Amazon Q'.



→ switch role means its not a root user, is an IAM user

→ click on S3 on dashboard:

The screenshot shows the Amazon S3 landing page. On the left, there's a sidebar with options like 'General purpose buckets', 'Directory buckets', etc. The main area features the 'Amazon S3' logo and the tagline 'Store and retrieve any amount of data from anywhere'. A prominent orange 'Create a bucket' button is visible. Below it, there's a section about pricing: 'With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.'

The screenshot shows the 'Buckets' page under the 'Amazon S3' service. The sidebar includes 'General purpose buckets', 'Directory buckets', etc. The main content area shows a table for 'General purpose buckets (0)'. A red box highlights an 'Error' message: 'Access Denied'. There's also a 'Diagnose with Amazon Q' button.

This user doesn't have the permission to access S3 buckets. The root user have the permission to access S3 buckets.

- To create a policy for accessing S3 bucket for that user:
  - go to root user → then IAM → Policies → Create policy
  - we can then use the built-in policies or create a default policy(as in our case)

Policies (1348) <small>Info</small>					
A policy is an object in AWS that defines permissions.					
<a href="#">Actions</a> <a href="#">Delete</a> <a href="#" style="border: 2px solid orange; color: orange;">Create policy</a>					
<b>Filter by Type</b>					
Policy name	Type	Used as	Description		
<a href="#">AccessAnalyzerSer...</a>	AWS managed	None	-		
<a href="#">AdministratorAccess</a>	AWS managed - job fu...	None	Provides full access to AWS services an		
<a href="#">AdministratorAcce...</a>	AWS managed	None	Grants account administrative permis		
<a href="#">AdministratorAcce...</a>	AWS managed	None	Grants account administrative permis		
<a href="#">AIOpsAssistantPolicy</a>	AWS managed	None	Provides ReadOnly permissions requir.		
<a href="#">AIOpsConsoleAdmi...</a>	AWS managed	None	Grants full access to Amazon AI Opera		
<a href="#">AIOpsOperatorAcc...</a>	AWS managed	None	Grants access to the Amazon AI Opera		

aws IAM Policies Create policy

Step 1 **Specify permissions**

Step 2 Review and create

**Specify permissions** Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

**Policy editor**

Visual [JSON](#) Actions

```

1▼ {
2    "Version": "2012-10-17",
3▼   "Statement": [
4▼     {
5       "Sid": "statement1",
6       "Effect": "Allow",
7       "Action": [],
8       "Resource": []
9     }
10  ]
11 }
```

Edit statement Statement1 Remove

Add actions

Choose a service  Filter services

Available

- AI Operations
- AMP
- API Gateway
- API Gateway V2

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```

1▼ {
2    "Version": "2012-10-17",
3▼   "Statement": [
4▼     {
5       "Sid": "AllowS3AccessToTestUser",
6       "Effect": "Allow",
7       "Action": "s3:*",
8       "Resource": "*"
9     }
10  ]
11 }
```

© 2025, Amazon Web Services, Inc.

The screenshot shows the 'Create policy' wizard in the AWS IAM console. The current step is 'Review and create'. The left sidebar shows 'Step 1: Specify permissions' and 'Step 2: Review and create' (which is selected). The main area contains 'Policy details' with a 'Policy name' field set to 'test-user-allow-s3'. It also includes a 'Description - optional' field which is empty. Below this is a section titled 'Permissions defined in this policy' with an 'Edit' button. At the bottom of the page are standard AWS navigation links: CloudShell, Feedback, © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

→ Assigning the policy to the user:

IAM → users → test\_user → Add Permission

The screenshot shows the 'Users' page in the AWS IAM console. On the left, there's a sidebar with 'Identity and Access Management (IAM)' and several navigation items like Dashboard, Access management, and Access reports. The main area displays the 'test\_user1' profile. The 'Summary' section shows the ARN as 'arn:aws:iam::331409392843:user/test\_user1', 'Console access' status as 'Enabled without MFA', and 'Access key 1' status as 'Create access key'. Below the summary is a tab bar with 'Permissions' (which is selected), Groups, Tags, Security credentials, and Last Accessed. Under the 'Permissions' tab, there's a section for 'Permissions policies (0)' with a 'Remove' button and an 'Add permissions' button (which is highlighted with a red box). There are also search and filter options for the list.

The screenshot shows the 'Add permissions' step in the AWS IAM console. The 'Permissions options' section has 'Attach policies directly' selected. Below it, the 'Permissions policies' list shows two policies: 'AWSIAMIdentityCenterAllowAll' (AWS managed) and 'test-user-allow-s3' (Customer managed), with the latter being selected. The bottom navigation bar includes 'Cancel', 'Next', 'CloudShell', 'Feedback', and links for '© 2025, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the 'Review' step in the AWS IAM console. It displays the 'User details' (User name: test\_user1) and 'Permissions summary' (1 policy selected: 'test-user-allow-s3' - Customer managed, Used as Permissions policy). The navigation bar at the bottom includes 'Cancel', 'Previous', 'Add permissions', 'CloudShell', 'Feedback', and links for '© 2025, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

After adding the policy, now the test user has the access to S3 buckets -

The screenshot shows the AWS S3 console interface. The left sidebar has a 'General purpose buckets' section with links for Directory buckets, Table buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. Below this is a 'Block Public Access settings for this account' section. Under 'Storage Lens', there are links for Dashboards and Storage Lens groups. At the bottom of the sidebar are CloudShell and Feedback links. The main content area displays an 'Account Snapshot' with a note about storage usage and activity trends. It shows a 'General purpose buckets (0)' section with a 'Create bucket' button. A message states 'No buckets' and 'You don't have any buckets.' The bottom right of the screen shows copyright information: © 2025, Amazon Web Services, Inc. or its affiliates.

→ Creating a group with multiple users