

Docker Commands Cheat Sheet

Overview

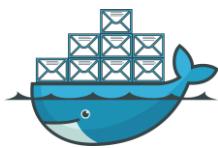
Docker works on Microservices Architecture, which helps developers to package and deploy applications in lightweight containers that run consistently across different environments.

Installation & Service Management

Purpose	Command
Install Docker (on RHEL/CentOS)	sudo yum install docker -y
Start Docker Service	sudo systemctl start docker
Enable Docker to Start on Boot	sudo systemctl enable docker
Check Docker Service Status	sudo systemctl status docker
Check Docker Version	docker --version
View Docker Info	docker info

Docker Hub — Popular Images

Image Name	Description	Pull Command
nginx	Web server	docker pull nginx
httpd	Apache HTTP server	docker pull httpd

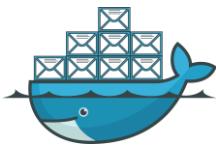


Docker Commands Cheat Sheet

Image Name	Description	Pull Command
mysql	MySQL Database	<code>docker pull mysql</code>
postgres	PostgreSQL Database	<code>docker pull postgres</code>
mongo	MongoDB Database	<code>docker pull mongo</code>
redis	In-memory database	<code>docker pull redis</code>
node	Node.js runtime	<code>docker pull node</code>
python	Python environment	<code>docker pull python</code>
ubuntu	Base OS image	<code>docker pull ubuntu</code>
alpine	Lightweight Linux base image	<code>docker pull alpine</code>
jenkins/jenkins	Jenkins CI/CD server	<code>docker pull jenkins/jenkins</code>
wordpress	WordPress CMS	<code>docker pull wordpress</code>
tomcat	Java Application Server	<code>docker pull tomcat</code>

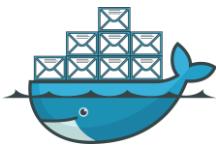
Container Management

Purpose	Command
Run a container (foreground)	<code>docker run <image_name></code>



Docker Commands Cheat Sheet

Purpose	Command
Run container in background (detached mode)	docker run -d <image_name>
Run container and expose port	docker run -d -p 80:80 nginx
Run container with a custom name	docker run -d -p 80:80 --name web nginx
Run container on a different host port	docker run -d -p 90:80 --name webserver nginx
List running containers	docker ps
List all containers (including stopped)	docker ps -a
Start a container	docker start <container_id>
Stop a container	docker stop <container_id>
Restart a container	docker restart <container_id>
Stop multiple containers	docker stop \$(docker ps -aq)
Remove a container	docker rm <container_id>
Remove all containers	docker rm \$(docker ps -aq)
Create container in stopped state	docker create <image_name>
Access container terminal	docker exec -it <container_id> /bin/bash
View container logs	docker logs <container_id>

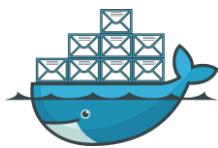


Docker Commands Cheat Sheet

Purpose	Command
Rename a container	docker rename <old_name> <new_name>
Inspect container details	docker inspect <container_id>

Image Management

Purpose	Command
Pull an image from Docker Hub	docker pull <image_name>
List all images	docker images
Build image from Dockerfile	docker build -t <image_name> .
Tag an image	docker tag <image_id> <username>/<repo_name>:<tag>
Remove an image	docker rmi <image_id>
Remove all unused images	docker image prune
Inspect image details	docker inspect <image_id>
Search image on Docker Hub	docker search <image_name>



Docker Commands Cheat Sheet

Docker Hub Login & Push Commands

Purpose	Command
Login to Docker Hub	docker login
Tag local image before pushing	docker tag <image_id> <dockerhub_username>/<repo_name>:<tag>
Push image to Docker Hub	docker push <dockerhub_username>/<repo_name>:<tag>
Pull image from Docker Hub	docker pull <dockerhub_username>/<repo_name>:<tag>
Logout from Docker Hub	docker logout

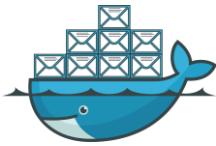
Example:

```
docker login
```

```
docker tag myapp:v1 shrikantkalange/myapp:v1
```

```
docker push shrikantkalange/myapp:v1
```

```
docker pull shrikantkalange/myapp:v1
```

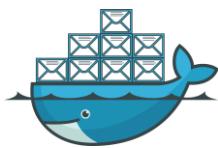


Docker Commands Cheat Sheet

Networking & Ports

Purpose	Command
Run container with port mapping (host:container)	docker run -d -p 8080:80 nginx
List all networks	docker network ls
Create a new network	docker network create <network_name>
Inspect a network	docker network inspect <network_name>
Connect a container to a network	docker network connect <network_name> <container_name>
Disconnect a container from a network	docker network disconnect <network_name> <container_name>
Remove a network	docker network rm <network_name>

Volumes (Storage Management)



Docker Commands Cheat Sheet

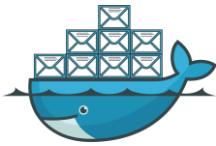
Purpose	Command
List all volumes	docker volume ls
Create a volume	docker volume create <volume_name>
Inspect a volume	docker volume inspect <volume_name>
Remove a volume	docker volume rm <volume_name>
Mount a volume while running a container	docker run -d -p 80:80 -v <volume_name>:/usr/share/nginx/html nginx

File Operations

Purpose	Command
Copy file from host to container	docker cp <file> <container_id>:/path
Copy file from container to host	docker cp <container_id>:/path <host_path>

Monitoring & Troubleshooting

Purpose	Command
View resource usage (CPU, Memory, IO)	docker stats



Docker Commands Cheat Sheet

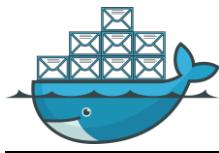
Purpose	Command
View all Docker processes	docker top <container_id>
Check container events/logs	docker events

Cleanup Commands

Purpose	Command
Remove all stopped containers	docker container prune
Remove all unused images	docker image prune
Remove all unused networks	docker network prune
Remove all unused data	docker system prune -a

Extra Notes

- Default Docker storage location: /var/lib/docker/
- Docker uses client-server architecture (Docker CLI → Docker Daemon).



Docker Commands Cheat Sheet

- You can view your web container (like nginx/httpd) in browser using:
👉 `http://localhost:80` or the mapped port.
 - To ensure Docker runs after every reboot:
`sudo systemctl enable docker.service`
 - Default Docker registry: <https://hub.docker.com>
-