Docker Command Cheatsheet

Basic Docker Info

```
docker --version                # Check Docker version
docker info                     # Display system-wide Docker info
```

Images

```
docker pull nginx               # Download image from Docker Hub
docker images                   # List downloaded images
docker rmi nginx                # Remove an image
docker build -t myapp:1.0 .     # Build image from Dockerfile in current dir
```

Containers

```
docker run nginx                        # Run container (nginx image)
docker run -it ubuntu /bin/bash         # Run in interactive terminal mode
docker run -d -p 80:80 nginx            # Run detached & map port 80
docker run --name web nginx             # Run container with custom name
docker ps                               # List running containers
docker ps -a                            # List all containers (including stopped)
docker stop <container_id>              # Stop a running container
docker start <container_id>             # Start a stopped container
docker restart <container_id>           # Restart a container
docker rm <container_id>                # Remove a container
```

Volumes

```
docker volume create myvol              # Create volume
docker volume ls                        # List volumes
docker run -v myvol:/data nginx         # Mount volume inside container
```

Container Management

```
docker exec -it <container> bash        # Run command inside running container
docker logs <container>                 # View logs of a container
docker inspect <container_or_image>     # Low-level details in JSON format
```

Dockerfile

```
FROM python:3.10
```

```
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
docker build -t myapp .              # Build image from Dockerfile
```

Docker Compose

```
version: "3"
services:
  web:
    image: nginx
    ports:
      - "8080:80"
docker-compose up                    # Start all services
docker-compose down                  # Stop and remove services
```

Clean-Up

```
docker system prune                  # Remove unused containers, networks, images
docker image prune                   # Remove unused images
docker volume prune                  # Remove unused volumes
```

Run with Environment Variables

```
docker run -e ENV=prod myapp         # Pass environment variable to container
```

Networking

```
docker network ls                    # List networks
docker network create mynet          # Create custom network
docker run --network=mynet myapp     # Attach container to custom network
```