### 1. ****Simple Python Application****

**Description:** A basic Python application with a single script.

Dockerfile

# Dockerfile

# Use the official Python image from the Docker Hub

FROM python:3.9-slim

# Set the working directory in the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . /app

# Install any needed packages specified in requirements.txt

RUN pip install --no-cache-dir -r requirements.txt

# Make port 80 available to the world outside this container

EXPOSE 80

# Run app.py when the container launches

CMD ["python", "app.py"]

### 2. ****Node.js Application****

**Description:** A simple Node.js application using Express.

Dockerfile

# Dockerfile

# Use the official Node.js image from the Docker Hub

FROM node:14

# Create app directory

WORKDIR /usr/src/app

# Install app dependencies

COPY package\*.json ./

RUN npm install

# Bundle app source

COPY . .

# Make port 8080 available to the world outside this container

EXPOSE 8080

# Run the app

CMD ["node", "server.js"]

### 3. ****React Application****

**Description:** A Dockerfile for a React application.

Dockerfile

# Dockerfile

# Use the official Node.js image from the Docker Hub

FROM node:14

# Create app directory

WORKDIR /usr/src/app

# Install app dependencies

COPY package\*.json ./

RUN npm install

# Bundle app source

COPY . .

# Build the React app

RUN npm run build

# Use nginx to serve the built app

FROM nginx:alpine

COPY --from=0 /usr/src/app/build /usr/share/nginx/html

# Expose port 80

EXPOSE 80

# Start nginx server

CMD ["nginx", "-g", "daemon off;"]

### 4. ****Django Application****

**Description:** A Dockerfile for a Django application.

Dockerfile

# Dockerfile

# Use the official Python image from the Docker Hub

FROM python:3.9-slim

# Set environment variables

ENV PYTHONDONTWRITEBYTECODE 1

ENV PYTHONUNBUFFERED 1

# Set the working directory in the container

WORKDIR /app

# Install dependencies

COPY requirements.txt /app/

RUN pip install --no-cache-dir -r requirements.txt

# Copy project

COPY . /app/

# Expose port 8000

EXPOSE 8000

# Run the Django development server

CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]

### 5. ****Flask Application****

**Description:** A Dockerfile for a Flask application.

Dockerfile

# Dockerfile

# Use the official Python image from the Docker Hub

FROM python:3.9-slim

# Set the working directory in the container

WORKDIR /app

# Copy requirements.txt

COPY requirements.txt /app/

# Install any needed packages specified in requirements.txt

RUN pip install --no-cache-dir -r requirements.txt

# Copy the rest of the application

COPY . /app/

# Make port 5000 available to the world outside this container

EXPOSE 5000

# Define environment variable

ENV FLASK\_APP=app.py

# Run the Flask app

CMD ["flask", "run", "--host=0.0.0.0"]

### 6. ****Spring Boot Application****

**Description:** A Dockerfile for a Spring Boot application.

Dockerfile

# Dockerfile

# Use Maven to build the app

FROM maven:3.6.3-jdk-11 AS build

WORKDIR /app

COPY . /app

RUN mvn clean package

# Use OpenJDK for the runtime

FROM openjdk:11-jre-slim

WORKDIR /app

COPY --from=build /app/target/\*.jar app.jar

# Make port 8080 available to the world outside this container

EXPOSE 8080

# Run the Spring Boot application

CMD ["java", "-jar", "app.jar"]

### 7. ****Go Application****

**Description:** A Dockerfile for a Go application.

Dockerfile

# Dockerfile

# Build the Go app

FROM golang:1.16 AS build

WORKDIR /src

COPY . .

RUN go mod tidy

RUN go build -o app .

# Run the Go app

FROM debian:buster

COPY --from=build /src/app /app

EXPOSE 8080

CMD ["/app"]

### 8. ****Rust Application****

**Description:** A Dockerfile for a Rust application.

Dockerfile

# Dockerfile

# Build the Rust app

FROM rust:1.52 as builder

WORKDIR /usr/src/myapp

COPY . .

RUN cargo build --release

# Run the Rust app

FROM debian:buster-slim

COPY --from=builder /usr/src/myapp/target/release/myapp /usr/local/bin/myapp

EXPOSE 8080

CMD ["myapp"]

### 9. ****Ruby on Rails Application****

**Description:** A Dockerfile for a Ruby on Rails application.

Dockerfile

# Dockerfile

# Use the official Ruby image from the Docker Hub

FROM ruby:2.7

# Set the working directory

WORKDIR /app

# Install dependencies

COPY Gemfile Gemfile.lock ./

RUN bundle install

# Copy the rest of the application

COPY . .

# Precompile assets

RUN bundle exec rake assets:precompile

# Expose port 3000

EXPOSE 3000

# Start the Rails server

CMD ["rails", "server", "-b", "0.0.0.0"]

### 10. ****PHP Application****

**Description:** A Dockerfile for a PHP application with Apache.

Dockerfile

# Dockerfile

# Use the official PHP image with Apache

FROM php:7.4-apache

# Copy the PHP application to the Apache document root

COPY src/ /var/www/html/

# Expose port 80

EXPOSE 80

# Start Apache server

CMD ["apache2-foreground"]

### 11. ****Laravel Application****

**Description:** A Dockerfile for a Laravel application.

Dockerfile

# Dockerfile

# Use the official PHP image with Apache

FROM php:7.4-apache

# Install system dependencies

RUN apt-get update && apt-get install -y \

libpng-dev \

libjpeg62-turbo-dev \

libfreetype6-dev \

&& docker-php-ext-configure gd --with-freetype --with-jpeg \

&& docker-php-ext-install gd

# Install PHP extensions

RUN docker-php-ext-install pdo pdo\_mysql

# Enable Apache mod\_rewrite

RUN a2enmod rewrite

# Set the working directory

WORKDIR /var/www/html

# Copy the Laravel application

COPY . .

# Set permissions

RUN chown -R www-data:www-data /var/www/html/storage /var/www/html/bootstrap/cache

# Expose port 80

EXPOSE 80

# Start Apache server

CMD ["apache2-foreground"]

### 12. ****Nginx Server****

**Description:** A basic Nginx web server.

Dockerfile

# Dockerfile

# Use the official Nginx image

FROM nginx:alpine

# Copy custom configuration file from the current directory

COPY nginx.conf /etc/nginx/nginx.conf

# Copy static website files

COPY html /usr/share/nginx/html

# Expose port 80

EXPOSE 80

# Start Nginx server

CMD ["nginx", "-g", "daemon off;"]

### 13. ****Apache Server****

**Description:** A basic Apache web server.

Dockerfile

# Dockerfile

# Use the official Apache image

FROM httpd:2.4

# Copy custom configuration file from the current directory

COPY httpd.conf /usr/local/apache2/conf/httpd.conf

# Copy static website files

COPY html /usr/local/apache2/htdocs/

# Expose port 80

EXPOSE 80

# Start Apache server

CMD ["httpd-foreground"]

### 14. ****MongoDB****

**Description:** A basic MongoDB server.

Dockerfile

# Dockerfile

# Use the official MongoDB image

FROM mongo:4.4

# Expose port 27017

EXPOSE 27017

# Start MongoDB

CMD ["mongod"]

### 15. ****PostgreSQL****

**Description:** A basic PostgreSQL server.

Dockerfile

# Dockerfile

# Use the official PostgreSQL image

FROM postgres:13

# Expose port 5432

EXPOSE 5432

# Start PostgreSQL

CMD ["postgres"]

### 16. ****MySQL****

**Description:** A basic MySQL server.

Dockerfile

# Dockerfile

# Use the official MySQL image

FROM mysql:8.0

# Set environment variables for MySQL

ENV MYSQL\_ROOT\_PASSWORD=root

ENV MYSQL\_DATABASE=mydb

# Expose port 3306

EXPOSE 3306

# Start MySQL

CMD ["mysqld"]

### 17. ****SQLite****

**Description:** A basic setup for SQLite.

Dockerfile

# Dockerfile

# Use the official Python image

FROM python:3.9-slim

# Set the working directory

WORKDIR /app

# Install SQLite

RUN apt-get update && apt-get install -y sqlite3

# Copy application files

COPY . .

# Expose port 5000

EXPOSE 5000

# Run the app

CMD ["python", "app.py"]

### 18. ****Redis****

**Description:** A basic Redis server.

Dockerfile

# Dockerfile

# Use the official Redis image

FROM redis:6.2

# Expose port 6379

EXPOSE 6379

# Start Redis

CMD ["redis-server"]

### 19. ****Elasticsearch****

**Description:** A basic Elasticsearch server.

Dockerfile

# Dockerfile

# Use the official Elasticsearch image

FROM elasticsearch:7.10.2

# Expose port 9200

EXPOSE 9200

# Expose port 9300

EXPOSE 9300

# Start Elasticsearch

CMD ["elasticsearch"]

### 20. ****Kibana****

**Description:** A basic Kibana server.

Dockerfile

# Dockerfile

# Use the official Kibana image

FROM kibana:7.10.2

# Expose port 5601

EXPOSE 5601

# Start Kibana

CMD ["kibana"]

### 21. ****Logstash****

**Description:** A basic Logstash server.

Dockerfile

# Dockerfile

# Use the official Logstash image

FROM logstash:7.10.2

# Copy the Logstash configuration file

COPY logstash.conf /usr/share/logstash/pipeline/logstash.conf

# Expose port 5044

EXPOSE 5044

# Start Logstash

CMD ["logstash", "-f", "/usr/share/logstash/pipeline/logstash.conf"]

### 22. ****Zookeeper****

**Description:** A basic Zookeeper server.

Dockerfile

# Dockerfile

# Use the official Zookeeper image

FROM zookeeper:3.6

# Expose port 2181

EXPOSE 2181

# Start Zookeeper

CMD ["zkServer.sh", "start-foreground"]

### 23. ****Kafka****

**Description:** A basic Kafka server.

Dockerfile

# Dockerfile

# Use the official Kafka image

FROM wurstmeister/kafka:2.13-2.6.0

# Expose port 9092

EXPOSE 9092

# Start Kafka

CMD ["/bin/sh", "-c", "start-kafka.sh"]

### 24. ****RabbitMQ****

**Description:** A basic RabbitMQ server.

Dockerfile

# Dockerfile

# Use the official RabbitMQ image

FROM rabbitmq:3.8

# Expose port 5672

EXPOSE 5672

# Start RabbitMQ

CMD ["rabbitmq-server"]

### 25. ****WordPress****

**Description:** A basic WordPress setup.

Dockerfile

# Dockerfile

# Use the official WordPress image

FROM wordpress:latest

# Expose port 80

EXPOSE 80

# Start WordPress

CMD ["apache2-foreground"]

### 26. ****Jenkins****

**Description:** A basic Jenkins setup.

Dockerfile

# Dockerfile

# Use the official Jenkins image

FROM jenkins/jenkins:lts

# Expose port 8080

EXPOSE 8080

# Start Jenkins

CMD ["jenkins"]

### 27. ****Grafana****

**Description:** A basic Grafana server.

Dockerfile

# Dockerfile

# Use the official Grafana image

FROM grafana/grafana:7.5.0

# Expose port 3000

EXPOSE 3000

# Start Grafana

CMD ["grafana-server"]

### 28. ****Prometheus****

**Description:** A basic Prometheus server.

Dockerfile

# Dockerfile

# Use the official Prometheus image

FROM prom/prometheus:v2.26.0

# Expose port 9090

EXPOSE 9090

# Start Prometheus

CMD ["prometheus"]

### 29. ****InfluxDB****

**Description:** A basic InfluxDB server.

Dockerfile

# Dockerfile

# Use the official InfluxDB image

FROM influxdb:2.0

# Expose port 8086

EXPOSE 8086

# Start InfluxDB

CMD ["influxd"]

### 30. ****Telegraf****

**Description:** A basic Telegraf setup.

Dockerfile

# Dockerfile

# Use the official Telegraf image

FROM telegraf:1.18

# Copy the Telegraf configuration file

COPY telegraf.conf /etc/telegraf/telegraf.conf

# Expose port 8125

EXPOSE 8125

# Start Telegraf

CMD ["telegraf"]

### 31. ****Chronograf****

**Description:** A basic Chronograf server.

Dockerfile

# Dockerfile

# Use the official Chronograf image

FROM chronograf:1.8

# Expose port 8888

EXPOSE 8888

# Start Chronograf

CMD ["chronograf"]

### 32. ****Hadoop****

**Description:** A basic Hadoop setup.

Dockerfile

# Dockerfile

# Use the official Hadoop image

FROM sequenceiq/hadoop-docker:2.7.1

# Expose port 50070

EXPOSE 50070

# Expose port 8088

EXPOSE 8088

# Start Hadoop

CMD ["/etc/bootstrap.sh", "-d"]

### 33. ****Spark****

**Description:** A basic Spark setup.

Dockerfile

# Dockerfile

# Use the official Spark image

FROM bitnami/spark:3.1.1

# Expose port 8080

EXPOSE 8080

# Start Spark master

CMD ["/opt/bitnami/scripts/spark/entrypoint.sh", "/opt/bitnami/scripts/spark/run.sh", "master"]

### 34. ****Airflow****

**Description:** A basic Airflow setup.

Dockerfile

# Dockerfile

# Use the official Airflow image

FROM apache/airflow:2.1.0

# Expose port 8080

EXPOSE 8080

# Initialize the database

RUN airflow db init

# Start Airflow web server

CMD ["airflow", "webserver"]

### 35. ****Consul****

**Description:** A basic Consul server.

Dockerfile

# Dockerfile

# Use the official Consul image

FROM consul:1.9

# Expose port 8500

EXPOSE 8500

# Start Consul

CMD ["consul", "agent", "-dev"]

### 36. ****Vault****

**Description:** A basic Vault server.

Dockerfile

# Dockerfile

# Use the official Vault image

FROM vault:1.6

# Expose port 8200

EXPOSE 8200

# Start Vault

CMD ["vault", "server", "-dev"]

### 37. ****Etcd****

**Description:** A basic Etcd server.

Dockerfile

# Dockerfile

# Use the official Etcd image

FROM quay.io/coreos/etcd:v3.4.15

# Expose port 2379

EXPOSE 2379

# Expose port 2380

EXPOSE 2380

# Start Etcd

CMD ["etcd"]

### 38. ****MinIO****

**Description:** A basic MinIO server.

Dockerfile

# Dockerfile

# Use the official MinIO image

FROM minio/minio:RELEASE.2021-02-14T04-01-33Z

# Expose port 9000

EXPOSE 9000

# Start MinIO

CMD ["server", "/data"]

### 39. ****OpenLDAP****

**Description:** A basic OpenLDAP server.

Dockerfile

# Dockerfile

# Use the official OpenLDAP image

FROM osixia/openldap:1.4.0

# Expose port 389

EXPOSE 389

# Expose port 636

EXPOSE 636

# Start OpenLDAP

CMD ["/usr/sbin/slapd", "-d", "32768"]

### 40. ****Keycloak****

**Description:** A basic Keycloak server.

Dockerfile

# Dockerfile

# Use the official Keycloak image

FROM jboss/keycloak:12.0.4

# Expose port 8080

EXPOSE 8080

# Start Keycloak

CMD ["-b", "0.0.0.0"]

### 41. ****Tomcat****

**Description:** A basic Tomcat server.

Dockerfile

# Dockerfile

# Use the official Tomcat image

FROM tomcat:9.0

# Expose port 8080

EXPOSE 8080

# Start Tomcat

CMD ["catalina.sh", "run"]

### 42. ****WildFly****

**Description:** A basic WildFly server.

Dockerfile

# Dockerfile

# Use the official WildFly image

FROM jboss/wildfly:23.0.2.Final

# Expose port 8080

EXPOSE 8080

# Start WildFly

CMD ["/opt/jboss/wildfly/bin/standalone.sh", "-b", "0.0.0.0"]

### 43. ****GlassFish****

**Description:** A basic GlassFish server.

Dockerfile

# Dockerfile

# Use the official GlassFish image

FROM glassfish:5.0

# Expose port 8080

EXPOSE 8080

# Expose port 4848

EXPOSE 4848

# Start GlassFish

CMD ["asadmin", "start-domain", "-v"]

### 44. ****Payara****

**Description:** A basic Payara server.

Dockerfile

# Dockerfile

# Use the official Payara image

FROM payara/server-full:5.2020.7

# Expose port 8080

EXPOSE 8080

# Expose port 4848

EXPOSE 4848

# Start Payara

CMD ["asadmin", "start-domain", "-v"]

### 45. ****JBoss EAP****

**Description:** A basic JBoss EAP server.

Dockerfile

# Dockerfile

# Use the official JBoss EAP image

FROM jboss-eap-7/eap74-openjdk8-openshift

# Expose port 8080

EXPOSE 8080

# Start JBoss EAP

CMD ["standalone.sh", "-b", "0.0.0.0"]

### 46. ****NATS****

**Description:** A basic NATS server.

Dockerfile

# Dockerfile

# Use the official NATS image

FROM nats:2.1

# Expose port 4222

EXPOSE 4222

# Start NATS

CMD ["nats-server"]

### 47. ****NSQ****

**Description:** A basic NSQ server.

Dockerfile

# Dockerfile

# Use the official NSQ image

FROM nsqio/nsq

# Expose port 4150

EXPOSE 4150

# Expose port 4151

EXPOSE 4151

# Start NSQ

CMD ["nsqd"]

### 48. ****Mosquitto****

**Description:** A basic Mosquitto MQTT server.

Dockerfile

# Dockerfile

# Use the official Mosquitto image

FROM eclipse-mosquitto:1.6

# Expose port 1883

EXPOSE 1883

# Expose port 9001

EXPOSE 9001

# Start Mosquitto

CMD ["mosquitto", "-c", "/mosquitto/config/mosquitto.conf"]

### 49. ****EMQX****

**Description:** A basic EMQX MQTT server.

Dockerfile

# Dockerfile

# Use the official EMQX image

FROM emqx/emqx:4.2

# Expose port 1883

EXPOSE 1883

# Expose port 8083

EXPOSE 8083

# Expose port 8084

EXPOSE 8084

# Expose port 18083

EXPOSE 18083

# Start EMQX

CMD ["emqx", "start"]

### 50. ****RabbitMQ MQTT****

**Description:** A basic RabbitMQ MQTT server.

Dockerfile

# Dockerfile

# Use the official RabbitMQ image

FROM rabbitmq:3.8-management

# Enable the MQTT plugin

RUN rabbitmq-plugins enable rabbitmq\_mqtt

# Expose port 1883

EXPOSE 1883

# Expose port 15675

EXPOSE 15675

# Start RabbitMQ

CMD ["rabbitmq-server"]

### 51. ****Celery****

**Description:** A basic Celery worker.

Dockerfile

# Dockerfile

# Use the official Python image

FROM python:3.9-slim

# Set the working directory

WORKDIR /app

# Install dependencies

COPY requirements.txt /app/

RUN pip install --no-cache-dir -r requirements.txt

# Copy the Celery worker code

COPY . /app/

# Start Celery worker

CMD ["celery", "-A", "tasks", "worker", "--loglevel=info"]

### 52. ****RabbitMQ Celery****

**Description:** A basic RabbitMQ Celery setup.

Dockerfile

# Dockerfile

# Use the official RabbitMQ image

FROM rabbitmq:3.8

# Expose port 5672

EXPOSE 5672

# Start RabbitMQ

CMD ["rabbitmq-server"]

### 53. ****Postfix****

**Description:** A basic Postfix mail server.

Dockerfile

# Dockerfile

# Use the official Postfix image

FROM catatnight/postfix:latest

# Set environment variables

ENV maildomain example.com

ENV smtp\_user user:password

# Expose port 25

EXPOSE 25

# Start Postfix

CMD ["supervisord", "-c", "/etc/supervisor/conf.d/supervisord.conf"]

### 54. ****OpenSMTPD****

**Description:** A basic OpenSMTPD mail server.

Dockerfile

# Dockerfile

# Use the official OpenSMTPD image

FROM opensmtpd/smtpd

# Expose port 25

EXPOSE 25

# Start OpenSMTPD

CMD ["smtpd", "-d"]

### 55. ****Mailhog****

**Description:** A basic Mailhog server.

Dockerfile

# Dockerfile

# Use the official Mailhog image

FROM mailhog/mailhog

# Expose port 8025

EXPOSE 8025

# Start Mailhog

CMD ["MailHog"]

### 56. ****Postfix + Dovecot****

**Description:** A basic Postfix + Dovecot mail server.

Dockerfile

# Dockerfile

# Use the official Dovecot image

FROM tvial/docker-mailserver:latest

# Expose ports

EXPOSE 25 143 587 993

# Start Postfix + Dovecot

CMD ["supervisord", "-c", "/etc/supervisor/supervisord.conf"]

### 57. ****ProFTPD****

**Description:** A basic ProFTPD server.

Dockerfile

# Dockerfile

# Use the official ProFTPD image

FROM stilliard/pure-ftpd:hardened

# Expose port 21

EXPOSE 21

# Start ProFTPD

CMD ["/run.sh"]

### 58. ****Vsftpd****

**Description:** A basic Vsftpd server.

Dockerfile

# Dockerfile

# Use the official Vsftpd image

FROM fauria/vsftpd

# Expose port 21

EXPOSE 21

# Start Vsftpd

CMD ["/usr/sbin/vsftpd", "/etc/vsftpd/vsftpd.conf"]

### 59. ****OpenVPN****

**Description:** A basic OpenVPN server.

Dockerfile

# Dockerfile

# Use the official OpenVPN image

FROM kylemanna/openvpn

# Expose port 1194

EXPOSE 1194/udp

# Start OpenVPN

CMD ["ovpn\_run"]

### 60. ****WireGuard****

**Description:** A basic WireGuard server.

Dockerfile

# Dockerfile

# Use the official WireGuard image

FROM linuxserver/wireguard

# Expose port 51820

EXPOSE 51820/udp

# Start WireGuard

CMD ["/init"]

### 61. ****Squid****

**Description:** A basic Squid proxy server.

Dockerfile

# Dockerfile

# Use the official Squid image

FROM sameersbn/squid:3.5.27-2

# Expose port 3128

EXPOSE 3128

# Start Squid

CMD ["/usr/sbin/squid3", "-N"]

### 62. ****HAProxy****

**Description:** A basic HAProxy load balancer.

Dockerfile

# Dockerfile

# Use the official HAProxy image

FROM haproxy:2.3

# Expose port 80

EXPOSE 80

# Start HAProxy

CMD ["haproxy", "-f", "/usr/local/etc/haproxy/haproxy.cfg"]

### 63. ****Traefik****

**Description:** A basic Traefik reverse proxy.

Dockerfile

# Dockerfile

# Use the official Traefik image

FROM traefik:v2.4

# Expose ports

EXPOSE 80

EXPOSE 443

# Start Traefik

CMD ["traefik"]

### 64. ****Envoy****

**Description:** A basic Envoy proxy.

Dockerfile

# Dockerfile

# Use the official Envoy image

FROM envoyproxy/envoy:v1.18.2

# Copy the Envoy configuration file

COPY envoy.yaml /etc/envoy/envoy.yaml

# Expose port 10000

EXPOSE 10000

# Start Envoy

CMD ["envoy", "-c", "/etc/envoy/envoy.yaml"]

### 65. ****Nginx Reverse Proxy****

**Description:** A basic Nginx reverse proxy.

Dockerfile

# Dockerfile

# Use the official Nginx image

FROM nginx:alpine

# Copy the Nginx configuration file

COPY nginx.conf /etc/nginx/nginx.conf

# Expose port 80

EXPOSE 80

# Start Nginx

CMD ["nginx", "-g", "daemon off;"]

### 66. ****Apache Reverse Proxy****

**Description:** A basic Apache reverse proxy.

Dockerfile

# Dockerfile

# Use the official Apache image

FROM httpd:2.4

# Copy the Apache configuration file

COPY httpd.conf /usr/local/apache2/conf/httpd.conf

# Expose port 80

EXPOSE 80

# Start Apache

CMD ["httpd-foreground"]

### 67. ****Lighttpd Reverse Proxy****

**Description:** A basic Lighttpd reverse proxy.

Dockerfile

# Dockerfile

# Use the official Lighttpd image

FROM seccubus/lighttpd:latest

# Copy the Lighttpd configuration file

COPY lighttpd.conf /etc/lighttpd/lighttpd.conf

# Expose port 80

EXPOSE 80

# Start Lighttpd

CMD ["lighttpd", "-D", "-f", "/etc/lighttpd/lighttpd.conf"]

### 68. ****Caddy Reverse Proxy****

**Description:** A basic Caddy reverse proxy.

Dockerfile

# Dockerfile

# Use the official Caddy image

FROM caddy:2.3

# Copy the Caddy configuration file

COPY Caddyfile /etc/caddy/Caddyfile

# Expose port 80

EXPOSE 80

# Start Caddy

CMD ["caddy", "run", "--config", "/etc/caddy/Caddyfile"]

### 69. ****Ghost****

**Description:** A basic Ghost blog.

Dockerfile

# Dockerfile

# Use the official Ghost image

FROM ghost:4-alpine

# Expose port 2368

EXPOSE 2368

# Start Ghost

CMD ["npm", "start", "--production"]

### 70. ****Joomla****

**Description:** A basic Joomla CMS.

Dockerfile

# Dockerfile

# Use the official Joomla image

FROM joomla:latest

# Expose port 80

EXPOSE 80

# Start Joomla

CMD ["apache2-foreground"]

### 71. ****Magento****

**Description:** A basic Magento e-commerce platform.

Dockerfile

# Dockerfile

# Use the official Magento image

FROM magento/magento2

# Expose port 80

EXPOSE 80

# Start Magento

CMD ["apache2-foreground"]

### 72. ****PrestaShop****

**Description:** A basic PrestaShop e-commerce platform.

Dockerfile

# Dockerfile

# Use the official PrestaShop image

FROM prestashop/prestashop:latest

# Expose port 80

EXPOSE 80

# Start PrestaShop

CMD ["apache2-foreground"]

### 73. ****OpenCart****

**Description:** A basic OpenCart e-commerce platform.

Dockerfile

# Dockerfile

# Use the official OpenCart image

FROM opencart/opencart

# Expose port 80

EXPOSE 80

# Start OpenCart

CMD ["apache2-foreground"]

### 74. ****MediaWiki****

**Description:** A basic MediaWiki server.

Dockerfile

# Dockerfile

# Use the official MediaWiki image

FROM mediawiki:latest

# Expose port 80

EXPOSE 80

# Start MediaWiki

CMD ["apache2-foreground"]

### 75. ****Matomo****

**Description:** A basic Matomo analytics server.

Dockerfile

# Dockerfile

# Use the official Matomo image

FROM matomo:latest

# Expose port 80

EXPOSE 80

# Start Matomo

CMD ["apache2-foreground"]

### 76. ****Odoo****

**Description:** A basic Odoo ERP server.

Dockerfile

# Dockerfile

# Use the official Odoo image

FROM odoo:14

# Expose port 8069

EXPOSE 8069

# Start Odoo

CMD ["odoo"]

### 77. ****ERPNext****

**Description:** A basic ERPNext server.

Dockerfile

# Dockerfile

# Use the official ERPNext image

FROM frappe/erpnext

# Expose port 8000

EXPOSE 8000

# Start ERPNext

CMD ["bench", "start"]

### 78. ****SuiteCRM****

**Description:** A basic SuiteCRM server.

Dockerfile

# Dockerfile

# Use the official SuiteCRM image

FROM bitnami/suitecrm:latest

# Expose port 80

EXPOSE 80

# Start SuiteCRM

CMD ["apache2-foreground"]

### 79. ****SugarCRM****

**Description:** A basic SugarCRM server.

Dockerfile

# Dockerfile

# Use the official SugarCRM image

FROM sugarcrm/sugarcrm

# Expose port 80

EXPOSE 80

# Start SugarCRM

CMD ["apache2-foreground"]

### 80. ****EspoCRM****

**Description:** A basic EspoCRM server.

Dockerfile

# Dockerfile

# Use the official EspoCRM image

FROM espo/espo

# Expose port 80

EXPOSE 80

# Start EspoCRM

CMD ["apache2-foreground"]

### 81. ****Dolibarr****

**Description:** A basic Dolibarr ERP/CRM server.

Dockerfile

# Dockerfile

# Use the official Dolibarr image

FROM bitnami/dolibarr:latest

# Expose port 80

EXPOSE 80

# Start Dolibarr

CMD ["apache2-foreground"]

### 82. ****Redmine****

**Description:** A basic Redmine project management server.

Dockerfile

# Dockerfile

# Use the official Redmine image

FROM redmine:latest

# Expose port 3000

EXPOSE 3000

# Start Redmine

CMD ["rails", "server", "-b", "0.0.0.0"]

### 83. ****Taiga****

**Description:** A basic Taiga project management server.

Dockerfile

# Dockerfile

# Use the official Taiga image

FROM taigaio/taiga-back

# Expose port 8000

EXPOSE 8000

# Start Taiga

CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]

### 84. ****Kanboard****

**Description:** A basic Kanboard project management server.

Dockerfile

# Dockerfile

# Use the official Kanboard image

FROM kanboard/kanboard:latest

# Expose port 80

EXPOSE 80

# Start Kanboard

CMD ["apache2-foreground"]

### 85. ****Wekan****

**Description:** A basic Wekan project management server.

Dockerfile

# Dockerfile

# Use the official Wekan image

FROM wekan/wekan

# Expose port 8080

EXPOSE 8080

# Start Wekan

CMD ["node", "main.js"]

### 86. ****Phabricator****

**Description:** A basic Phabricator server.

Dockerfile

# Dockerfile

# Use the official Phabricator image

FROM phacility/phabricator

# Expose port 80

EXPOSE 80

# Start Phabricator

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

### 87. ****Jira****

**Description:** A basic Jira server.

Dockerfile

# Dockerfile

# Use the official Jira image

FROM atlassian/jira-software

# Expose port 8080

EXPOSE 8080

# Start Jira

CMD ["/entrypoint.sh", "-fg"]

### 88. ****Confluence****

**Description:** A basic Confluence server.

Dockerfile

# Dockerfile

# Use the official Confluence image

FROM atlassian/confluence-server

# Expose port 8090

EXPOSE 8090

# Start Confluence

CMD ["/entrypoint.sh", "-fg"]

### 89. ****Bitbucket****

**Description:** A basic Bitbucket server.

Dockerfile

# Dockerfile

# Use the official Bitbucket image

FROM atlassian/bitbucket-server

# Expose port 7990

EXPOSE 7990

# Expose port 7992

EXPOSE 7992

# Start Bitbucket

CMD ["/entrypoint.sh", "-fg"]

### 90. ****GitLab****

**Description:** A basic GitLab server.

Dockerfile

# Dockerfile

# Use the official GitLab image

FROM gitlab/gitlab-ce

# Expose port 80

EXPOSE 80

# Expose port 443

EXPOSE 443

# Start GitLab

CMD ["/assets/wrapper"]

### 91. ****Gogs****

**Description:** A basic Gogs Git server.

Dockerfile

# Dockerfile

# Use the official Gogs image

FROM gogs/gogs

# Expose port 3000

EXPOSE 3000

# Expose port 22

EXPOSE 22

# Start Gogs

CMD ["/app/gogs/docker/start.sh"]

### 92. ****Gitea****

**Description:** A basic Gitea Git server.

Dockerfile

# Dockerfile

# Use the official Gitea image

FROM gitea/gitea

# Expose port 3000

EXPOSE 3000

# Expose port 22

EXPOSE 22

# Start Gitea

CMD ["/usr/local/bin/gitea", "web"]

### 93. ****Drone****

**Description:** A basic Drone CI/CD server.

Dockerfile

# Dockerfile

# Use the official Drone image

FROM drone/drone

# Expose port 80

EXPOSE 80

# Start Drone

CMD ["/bin/drone-server"]

### 94. ****Jenkins****

**Description:** A basic Jenkins CI/CD server.

Dockerfile

# Dockerfile

# Use the official Jenkins image

FROM jenkins/jenkins:lts

# Expose port 8080

EXPOSE 8080

# Start Jenkins

CMD ["/usr/bin/tini", "--", "/usr/local/bin/jenkins.sh"]

### 95. ****TeamCity****

**Description:** A basic TeamCity CI/CD server.

Dockerfile

# Dockerfile

# Use the official TeamCity image

FROM jetbrains/teamcity-server

# Expose port 8111

EXPOSE 8111

# Start TeamCity

CMD ["/run-services.sh"]

### 96. ****SonarQube****

**Description:** A basic SonarQube server.

Dockerfile

# Dockerfile

# Use the official SonarQube image

FROM sonarqube:latest

# Expose port 9000

EXPOSE 9000

# Start SonarQube

CMD ["bin/run.sh"]

### 97. ****Nexus****

**Description:** A basic Nexus Repository Manager server.

Dockerfile

# Dockerfile

# Use the official Nexus image

FROM sonatype/nexus3

# Expose port 8081

EXPOSE 8081

# Start Nexus

CMD ["bin/nexus", "run"]

### 98. ****Artifactory****

**Description:** A basic Artifactory server.

Dockerfile

# Dockerfile

# Use the official Artifactory image

FROM docker.bintray.io/jfrog/artifactory-oss:latest

# Expose port 8081

EXPOSE 8081

# Start Artifactory

CMD ["/entrypoint-artifactory.sh"]

### 99. ****Portainer****

**Description:** A basic Portainer server.

Dockerfile

# Dockerfile

# Use the official Portainer image

FROM portainer/portainer

# Expose port 9000

EXPOSE 9000

# Start Portainer

CMD ["portainer"]

### 100. ****Rancher****

**Description:** A basic Rancher server.

Dockerfile

# Dockerfile

# Use the official Rancher image

FROM rancher/rancher

# Expose port 80

EXPOSE 80

# Expose port 443

EXPOSE 443

# Start Rancher

CMD ["entrypoint.sh"]