Absolutely, Harshith! Here's your complete command reference for Docker 📆 , Maven 🔅 , and Git 👺 , with syntax, purpose, and usage context. This covers everything we've used so far across your Smart Assessment Hub, SMMA, and MJP projects.



MAVEN COMMANDS

Command	Description	When to Use
mvn clean	Deletes target/ directory and cleans previous builds	Before fresh builds
mvn compile	Compiles Java source code	After writing/modifying code
mvn test	Runs unit tests	During development or CI
mvn package	Packages code into .jar or .war	To generate deployable artifact
mvn install	Installs package into local .m2 repo	For reuse in other local projects
mvn clean install	Cleans and builds project fully	Most common full build
mvn dependency:tree	Shows dependency hierarchy	To debug conflicts or verify dependencies



MAVEN CONFIGURATION SNIPPETS

Add Servlet API

```
<dependency>
 <groupId>javax.servlet
 <artifactId>javax.servlet-api</artifactId>
 <version>4.0.1
 <scope>provided</scope>
</dependency>
```

Add MySQL Connector/J

```
<dependency>
 <groupId>mysql</groupId>
 <artifactId>mysql-connector-java</artifactId>
 <version>8.0.33
</dependency>
```

Change Output Directory

```
<build>
  <directory>build_output</directory>
</build>
```

GIT COMMANDS

Command	Description	When to Use
git configglobal user.name	Set global username	Initial setup
git configglobal user.email "email@example.com"	Set global email	Initial setup
git init	Initialize Git repo	Start version control
git clone <repo-url></repo-url>	Clone remote repo	Copy project locally
git status	Show staged/unstaged changes	Before committing
git add .	Stage all changes	Before commit
git add <file></file>	Stage specific file	Selective commit
git reset HEAD <file></file>	Unstage file	Undo staging

git commit -m "message"	Commit staged changes	Save snapshot
git commitamend -m "new message"	Modify last commit message	Before push
git log	View commit history	Audit changes
git logoneline	Compact commit view	Quick history check
git logpretty=fuller	Full commit details	Author, date, etc.
git show	Show last commit diff	Inspect changes
git diff	Compare working directory vs last commit	See unstaged changes
git checkout -b branch>	Create and switch to new branch	Feature development
git checkout <branch></branch>	Switch branches	Move between features
git merge <branch></branch>	Merge branch into current	Combine changes
git mergeff-only <branch></branch>	Fast-forward merge	Clean history
git stash	Save changes temporarily	Switch tasks
git stash pop	Restore stashed changes	Resume work
git fetch	Download changes from remote	Preview updates
git pull	Fetch + merge	Sync with remote
git revert <commit></commit>	Undo commit without deleting history	Safe rollback

git diff origin/main	Compare local vs remote	Check divergence
git apply <patch></patch>	Apply patch file	Bug fixes
git push origin <branch></branch>	Push branch to GitHub	Share work
git remote add origin <url></url>	Link local repo to GitHub	Setup remote
git fork (via GitHub UI)	Copy repo to your account	Contribute externally
git tag <name></name>	Tag a commit	Mark releases

OCKER COMMANDS

Command	Description	When to Use
dockerversion	Check Docker version	Initial setup
docker images	List local images	Audit builds
docker ps	List running containers	Monitor runtime
docker ps -a	List all containers	View stopped too
docker pull <image/>	Download image	Use official base
docker run <image/>	Run container	Quick test
docker run -it <image/> bash	Interactive shell	Debug or explore
docker runname <name> -p host:container <image/></name>	Run with port mapping	Web apps

docker stop <container></container>	Stop running container	Graceful shutdown
docker start <container></container>	Start stopped container	Resume work
docker rm <container></container>	Remove container	Cleanup
docker rmi <image/>	Remove image	Free space
docker exec -it <container> bash</container>	Enter running container	Inspect or debug
docker exec -it <container> <command/></container>	Run command inside container	DB ops, etc.
docker commit <container> <new-image></new-image></container>	Save container as image	Preserve changes
docker tag <image/> <repo image=""></repo>	Tag image for Docker Hub	Before push
docker login	Authenticate to Docker Hub	Push images
docker push <repo image=""></repo>	Upload image to Docker Hub	Share builds
docker logs <container></container>	View container logs	Troubleshooting



DOCKERFILE SYNTAX

For Maven Web App with Tomcat

FROM tomcat:9.0 RUN rm -rf /usr/local/tomcat/webapps/* COPY target/Smart-Assessment-Hub.war /usr/local/tomcat/webapps/Smart-Assessment-Hub.war EXPOSE 8080 CMD ["catalina.sh", "run"]



DOCKER COMPOSE COMMANDS

Command	Description	When to Use
docker compose up -d	Start all services in background	Launch app stack
docker compose ps	List running services	Verify status
docker compose logs <service></service>	View logs of a service	Debug issues
docker compose down	Stop and remove containers	Cleanup

DOCKER COMPOSE FILE (Multi-Service)

```
version: '3.8'
services:
  tomcat:
    image: harshith/smarthub-image
    container_name: sah-tomcat
    ports:
      - "8080:8080"
    depends_on:
      - mysql
 mysql:
    image: mysql:8.0
    container_name: sah-mysql
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: rootpass
      MYSQL_DATABASE: sahdb
    volumes:
      - mysql_data:/var/lib/mysql
volumes:
  mysql_data:
```

This cheat sheet is now your command arsenal *. Want me to format this into a printable guide or turn it into a markdown doc for GitHub README? 1

References (1)

1 https://github.com/smy-

I/Semicircle/tree/46cfd502dd5d7181eacf33a0f06765564ecd4988/javaee(new)%2Fnote%2Fmaven%2FMaven文稿.md