\*\*برای استفاده از قابلیت ، Gitlab CICD باید تغییرات زیر را در پروژه خود اعمال کنید: به نکات پایانی حتما دقت نمایید\*\*

\*\*نکته :100: اگر قبلا تغییرات را در پروژه اعمال کرده اید تنها فایل جدید .gitlab-ci.yml را به branch خود اضافه نمایید\*\*

\*\*نکته :100: تنها در برنچ مستر، Gitlab-CICD اجرا خواهد شد\*\*

\*\*نکته :100: در فایل جدید تنها استیج build بصورت خودکار انجام می شود، اگر به استیج های دیگر (build image , deploy) نیاز دارید باید به صورت دستی آن ها را اجرا کنید\*\*

\*\*نکته :100: برای دریافت WAR فایل پروژه خود، با اتمام موفق استیج build، استیج را باز کرده و در سمت راست صفحه aritifact-download آن را دانلود نمایید\*\*

\*\*نکته :100: برای اجرای دستی استیج ها و یا دانلود War فایل پروژه، تنها 60 دقیقه مهلت دارید و بعد از آن cache پاک خواهد شد\*\*

1. \*\*ابتدا باید Maven Wrapper را به پروژه اضافه کرد که با اضافه کردن فایل های زیر قابل انجام است:\*\*

\* اضافه کردن فایل های mvnw و mvnw.cmd به root directory

\* اضافه کردن دایرکتوری mvn. به root directory

\* اضافه کردن دایرکتوری wrapper به دایرکتوری mvn.

\* اضافه کردن فایلهای MavenWrapperDownloader.java و maven-wrapper.properties به دایرکتوری mvn/wrapper.

\* اضافه کردن پکیج maven-wrapper.jar به دایرکتوری mvn/wrapper.

2. \*\*اضافه کردن اطلاعات Nexus Repository Management-URL,username,password\*\*

\* اضافه کردن دایرکتوری m2. به root directory

\* اضافه کردن فایل settings.xml به دایرکتوری m2.

\* تغییر فایل pom.xml در root directory (اضافه کردن DistributionManagement)

3. \*\*اضافه کردن فایل مربوط به Docker\*\*

\* اضافه کردن app-compose.yml به src/main/docker

4. \*\*اضافه کردن فایل Gitlab-CICD\*\*

\* اضافه کردن gitlab-ci.yml. به root directory

\*\*نکته1: تمامی فایلهایی که ذکر شدند در همین پروژه و در دایرکتوری هایی که در بالا ذکر شده اند وجود دارد

لطفا از آنها استفاده کنید\*\*

\*\*نکته2: در فایل Gitlab-ci.yml. حتما دقت کنید در ابتدای کد در متغیر "App\_Name" کلمه "test" را به اسم پروژه خود تغییر دهید\*\*

\*\*نکته3: در فایل app-compose.yml حتما دقت نمایید کلمه "test" را به اسم پروژه خود تغییر دهید (در خطوط 6و7و8) \*\*

\*\*نکته4: در فایل app-compose.yml شماره پورت 8080 را به پورت دلخواه خود تغییر دهید (ممکن است برنامه دیگری همزمان از پورت انتخابی شما استفاده نماید که چرخه دچار خطا خواهد شد در نتیجه باید شماره پورت را تغییر دهید . دقت کنید که در این فایل، در 2 جا باید این شماره پورت را تغییر دهید\*\*

\*\*نکته 5: دقت کنید که مانند سابق حتما فایل های Dockerfile , Docker.properties به درستی در دایرکتوری مناسب موجود باشند\*\*

\*\*نکته 6: استیج deploy در pipeline بصورت دستی گذاشته شده است تا فقط در زمانهایی که نیاز داریم اپلیکیشن Deploy شود آن را اجرا کنیم. بنابراین اگر می خواهیم برنامه Deploy شود، باید در قسمت Pipline اجرای آن را Play نماییم\*\*

\*\*نکته7: بعد از انجام دقیق مراحل فوق برنامه شما بصورت خودکار ساخته و در ماشین 192.168.253.75 با پورت انتخابی خودتان اجرا می شود\*\*

\*\*نکته 8: برای دیدن چرخه CICD می توانید از قسمت CICD پروژه خود، گزینه Pipelne را انتخاب نمایید و مراحل اجرا را ببینید\*\*

\* نکته9: با هربار Push/commit در پروژه این چرخه اجرا خواهد شد

\*\*\* نکته 10: فایل war پروژه در nexus شرکت به آدرس 192.168.253.10:8081 آپلود می شود. برای دسترسی به آن به repository-maven-releases مراجعه کرده و مسیر پکیج خود را دنبال نمایید.\*\* یا maven-snapshots

\* نکته 11: در فایل pom.xml حتما پکیجینگ را war بگذارید و اسم اپلیکیشن نباید شامل حروف بزرگ باشد

برای استفاده از قابلیت ، Gitlab CICD باید تغییرات زیر را در پروژه خود اعمال کنید: به نکات پایانی حتما دقت نمایید

نکته 💯 اگر قبلا تغییرات را در پروژه اعمال کرده اید تنها فایل جدید .gitlab-ci.yml را به branch خود اضافه نمایید

نکته 💯 تنها در برنچ مستر، Gitlab-CICD اجرا خواهد شد

نکته 💯 در فایل جدید تنها استیج build بصورت خودکار انجام می شود، اگر به استیج های دیگر (build image , deploy) نیاز دارید باید به صورت دستی آن ها را اجرا کنید

نکته 💯 برای دریافت WAR فایل پروژه خود، با اتمام موفق استیج build، استیج را باز کرده و در سمت راست صفحه aritifact-download آن را دانلود نمایید

نکته 💯 برای اجرای دستی استیج ها و یا دانلود War فایل پروژه، تنها 60 دقیقه مهلت دارید و بعد از آن cache پاک خواهد شد

ابتدا باید Maven Wrapper را به پروژه اضافه کرد که با اضافه کردن فایل های زیر قابل انجام است:

اضافه کردن فایل های mvnw و mvnw.cmd به root directory

اضافه کردن دایرکتوری mvn. به root directory

اضافه کردن دایرکتوری wrapper به دایرکتوری mvn.

اضافه کردن فایلهای MavenWrapperDownloader.java و maven-wrapper.properties به دایرکتوری mvn/wrapper.

اضافه کردن پکیج maven-wrapper.jar به دایرکتوری mvn/wrapper.

اضافه کردن اطلاعات Nexus Repository Management-URL,username,password

اضافه کردن دایرکتوری m2. به root directory

اضافه کردن فایل settings.xml به دایرکتوری m2.

تغییر فایل pom.xml در root directory (اضافه کردن DistributionManagement)

اضافه کردن فایل مربوط به Docker

اضافه کردن app-compose.yml به src/main/docker

اضافه کردن فایل Gitlab-CICD

اضافه کردن gitlab-ci.yml. به root directory

نکته1: تمامی فایلهایی که ذکر شدند در همین پروژه و در دایرکتوری هایی که در بالا ذکر شده اند وجود دارد

لطفا از آنها استفاده کنید

نکته2: در فایل Gitlab-ci.yml. حتما دقت کنید در ابتدای کد در متغیر "App\_Name" کلمه "test" را به اسم پروژه خود تغییر دهید

\*\*نکته3: در فایل app-compose.yml حتما دقت نمایید کلمه "test" را به اسم پروژه خود تغییر دهید (در خطوط 6و7و8) \*\*

نکته4: در فایل app-compose.yml شماره پورت 8080 را به پورت دلخواه خود تغییر دهید (ممکن است برنامه دیگری همزمان از پورت انتخابی شما استفاده نماید که چرخه دچار خطا خواهد شد در نتیجه باید شماره پورت را تغییر دهید . دقت کنید که در این فایل، در 2 جا باید این شماره پورت را تغییر دهید

نکته 5: دقت کنید که مانند سابق حتما فایل های Dockerfile , Docker.properties به درستی در دایرکتوری مناسب موجود باشند

نکته 6: استیج deploy در pipeline بصورت دستی گذاشته شده است تا فقط در زمانهایی که نیاز داریم اپلیکیشن Deploy شود آن را اجرا کنیم. بنابراین اگر می خواهیم برنامه Deploy شود، باید در قسمت Pipline اجرای آن را Play نماییم

نکته7: بعد از انجام دقیق مراحل فوق برنامه شما بصورت خودکار ساخته و در ماشین 192.168.253.75 با پورت انتخابی خودتان اجرا می شود

نکته 8: برای دیدن چرخه CICD می توانید از قسمت CICD پروژه خود، گزینه Pipelne را انتخاب نمایید و مراحل اجرا را ببینید

نکته9: با هربار Push/commit در پروژه این چرخه اجرا خواهد شد

\*\*\* نکته 10: فایل war پروژه در nexus شرکت به آدرس 192.168.253.10:8081 آپلود می شود. برای دسترسی به آن به repository-maven-releases مراجعه کرده و مسیر پکیج خود را دنبال نمایید.\*\* یا maven-snapshots

نکته 11: در فایل pom.xml حتما پکیجینگ را war بگذارید و اسم اپلیکیشن نباید شامل حروف بزرگ باشد