MAVEN

~Maven is a project management and comprehension tool. Maven provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

In case of multiple development teams environment, Maven can set-up the way to work as per standards in a very short time. As most of the project setups are simple and reusable, Maven makes life of developer easy while creating reports, checks, build and testing automation setups.

10.What is maven life-cycle?

Ans:

Validate - validate that the project is correct and all necessary information is available

compile - compile the source code of the project

test - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed

package - take the compiled code and package it in its distributable format, such as a JAR.

integration-test - process and deploy the package if necessary into an environment where integration tests can be run

verify - run any checks to verify the package is valid and meets quality criteria

install - install the package into the local repository, for use as a dependency in other projects locally

deploy - done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects.

validate - validate the project is correct and all necessary information is available

* + - * + compile - compile the source code of the project
        + test - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
        + package - take the compiled code and package it in its distributable format, such as a JAR.
        + verify - run any checks on results of integration tests to ensure quality criteria are met
        + install - install the package into the local repository, for use as a dependency in other projects locally
        + deploy - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

~What are the advantages of Maven?

No need to add jar file when sharing the code in different machine.

Creates right directory structure

Builds and deploys the project

-The maven plugins are reusable.

~What is POM?

-In Maven, POM stands for Project Object Model. It is fundamental Unit of Work in Maven. It is an XML file.

~Why Maven Plugins are used?

Maven plugins are used to

• Create a jar file

• Create war file

• Compile code files

• Unit testing of code

• Documenting projects

• Reporting

~What is archetype?

Archetype is the maven plugin. It creates the project structure.

~Explain what is Maven artifact?

Usually, an artifact is a JAR file which gets arrayed to a Maven repository. One or more artifacts a maven build produces such as compiled JAR and a sources JAR.

Each artifact includes a group ID, an artifact ID and a version string.

~Explain how to run test classes in Maven?

-To run test classes in Maven, you need surefire plugin, check and configure your settings in setting.xml and pom.xml for a property named “test.”

~List out what are the Maven’s order of inheritance?

The maven’s order of inheritance is

• Parent Pom

• Project Pom

• Settings

• CLI parameters

~Explain what is Maven Repository? What are their types?

-A Maven repository is a location where all the project jars, library jars, plugins or any other particular project related artifacts are stored and can be easily used by Maven.

Their types are local, central and remote

Mention how profiles are specified in Maven?

-Profiles are specified in Maven by using a subset of the elements existing in the POM itself.

~For POM what are the minimum required elements?

-The minimum required elements for POM are project root, modelVersion, groupID, artifactID and version

~Explain how you can produce execution debug output or error messages?

-To produce execution debug output you could call Maven with X parameter or e parameter

What are the commend you used in maven?

Answer: Mvn test, mvn clean test, mvn test -DBrowser=BrowserName

How to run a single method from a command line?

-mvn Dtest=ProectName#MethodName test.

What is the benefit of using maven rather than java project?

-Maven has its own server where we can add all the dependencies directly in my project, so if I share my project with anybody they would not have to add the dependencies.

We can run the test from command line even if eclipse is closed.

Which is the main file used in maven to run the project?

-pom.xml file.

1. What is Jenkins?
2. CI & CD Tool
3. CI = continuous integration (integration process keep going >> Github has my old code and push new code >> then Jenkins will take both old & new code >> after that Jenkins will check the build)
4. Continuous = keep going / endless
5. Integration / integration testing = testing two or more unit / function together
6. Testing one unit/function = unit testing
7. CD = continuous delivery/development (When Jenkins build passed >> then it will deploy the code from another environment/ production)
8. Why do you need Jenkins in Automation?

CI purpose

1. Why do not use Jenkins as CD in Automation?

I am doing Automation in QA/dev environment & never do automation in production.

1. Why do you not run Automation in production?
2. Production means real application.
3. If I run automation, then application can break/down.
4. That will be problem for my client
5. Where do you run your Jenkins in your current project?
6. Server
7. Physical server (own house)

* Windows server >> safe answer
* Linux server

1. Cloud server (rent house)

* AWS
* Google web service
* HP

1. Container(Tent >> temporary)

* Docker container (Linux based) >> my current project.
* S3 bucket container (AWS) >> my current project.

1. Localhost(learning purpose) >> local or my laptop.
2. Who setup Docker/S3 container(AWS)?

I worked with my DevOPPs team and setup the contaiers.

***Q: How to configure Jenkins ?***

 Click manage jenkin–>System configure–>give maven home path–>jdk homepath–>git.exe file ar path

 if y want any plugin

 click manage jenkins -->manage plugin–>search let say github for git plugin-->install

***Q:How to create a job?***

 Click new item–>job name–>source for project (if from git then path for repo)–>build trigger(click any one let say build periodicly)–>build post(select

 command batch line and write command for run the project let say mvn test)—>u can set up email notification–> save

1. Mention what is Jenkins?

Jenkins is an open source tool with plugin built for continuous integration purpose. The principle functionality of Jenkins is to keep a track of version control system and to initiate and monitor a build system if changes occur. It monitors the whole process and provides reports and notifications to alert.

1. Explain what is continuous integration?

In software development, when multiple developers or teams are working on different segments of same web application, we need to perform integration test by integrating all modules. In order to do that an automated process for each piece of code is performed on daily bases so that all your code get tested.

1. What is the requirement for using Jenkins?

To use Jenkins you require

• A source code repository which is accessible, for instance, a Git repository

• A working build script, e.g., a Maven script, checked into the repository

1. Mention what are the advantages of Jenkins?

Advantage of Jenkins include

• At integration stage, build failures are cached

• For each code commit changes an automatic build report notification generates

• To notify developers about build report success or failure, it is integrated with LDAP mail server

• Achieves continuous integration agile development and test driven development

• With simple steps, maven release project is automated

• Easy tracking of bugs at early stage in development environment than production

1. Explain how you can move or copy Jenkins from one server to another?

• Slide a job from one installation of Jenkins to another by copying the related job directory

• Make a copy of an already existing job by making clone of a job directory by a different name

• Renaming an existing job by renaming a directory. https://www.guru99.com/ ------

1. **Move** a job from **one** installation of **Jenkins** to **another** by simply **copying the** corresponding job directory.
2. Make a **copy** of **an** existing job by making a **clone** of a job directory by a **different** name.
3. Rename **an** existing job by renaming a directory.
4. Mention what are the commands you can use to start Jenkins manually?

To start Jenkins manually, you can use either of the following

• (Jenkins\_url)/restart: Forces a restart without waiting for builds to complete

• (Jenkin\_url)/safeRestart: Allows all running builds to complete

1. Mention some of the useful plugins in Jenkin?

Some of the important plugins in Jenkin includes

• Maven 2 project

• Amazon EC2

• HTML publisher

• Copy artifact

• Join

• Green Balls

1. Explain how you can deploy a custom build of a core plugin?

To deploy a custom field of a core plugin, you have to do following things

• Stop Jenkins

• Copy the custom HPI to $Jenkins\_Home/plugins

• Delete the previously expanded plugin directory

• Make an empty file called .hpi.pinned

• Start Jenkins

1. Explain how can create a backup and copy files in Jenkins?

Jenkins saves all the setting, build artifacts and logs in its home directory, to create a back-up of your Jenkins setup, just copy this directory. You can also copy a job directory to clone or replicate a job or rename the directory. https://www.guru99.com/ -------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1. Explain how you can clone a Git repository via Jenkins?

To clone a Git repository via Jenkins, you have to enter the e-mail and user name for your Jenkins system. For that, you have to switch into your job directory and execute the “git config” command.

1. Explain how you can set up Jenkins job?

To create a project that is handled via jobs in Jenkins. Select New item from the menu, once this done enter a name for the job and select free-style job. Then click OK to create new job in Jenkins. The next page enables you to configure your job.

1. Mention what are the two components Jenkins is mainly integrated with?

Jenkin is mainly integrated with two components

• Version Control system like GIT, SVN

• And build tools like Apache Maven.

**What is a CI/CD pipeline?**

* A CI/CD pipeline automates the process of software delivery.

A CI/CD pipeline automates the process of software delivery. It builds code, runs tests, and helps you to safely deploy a new version of the software. CI/CD pipeline reduces manual errors, provides feedback to developers, and allows fast product iterations.

CI/CD pipeline introduces automation and continuous monitoring throughout the lifecycle of a software product. It involves from the integration and testing phase to delivery and deployment. These connected practices are referred as CI/CD pipeline.

**What is Continuous Integration, Continuous Delivery, and Continuous Deployment?**

* **Continuous integration** is a software development method where members of the team can integrate their work at least once a day. In this method, every integration is checked by an automated build to search the error.
* **Continuous delivery** is a software engineering method in which a team develops software products in a short cycle. It ensures that software can be easily released at any time.
* **Continuous deployment** is a software engineering process in which product functionalities are delivered using automatic deployment. It helps testers to validate whether the codebase changes are correct, and it is stable or not.

2.1M

SSH,In this CI/CD Pipeline tutorial, you will learn:

* [Stages of a CI/CD pipeline](https://www.guru99.com/ci-cd-pipeline.html#1)
* [Example of CI/CD Pipeline](https://www.guru99.com/ci-cd-pipeline.html#2)
* [CI/CD pipeline Best Practices](https://www.guru99.com/ci-cd-pipeline.html#3)
* [Advantages of CI/CD pipelines](https://www.guru99.com/ci-cd-pipeline.html#4)
* [Important CI/CD tools](https://www.guru99.com/ci-cd-pipeline.html#5)
* [Why Does the CI/CD Pipeline Matter for IT Leaders?](https://www.guru99.com/ci-cd-pipeline.html#6)
* [Ci/CD Pipeline KPI](https://www.guru99.com/ci-cd-pipeline.html#7)

**Stages of a CI/CD pipeline**

A CI/CD pipeline is a runnable specification of the steps that any developer should perform to deliver a new version of any software. Failure in each and every stage triggers a notification via email, Slack, or other communication platforms. It enables responsible developers to know about the important issues.

Here are the important Stages of CI/CD pipeline:

[Diagram

Description automatically generated](https://www.guru99.com/images/2/100820_0534_CICDPipelin1.png)Stages of CI/CD pipeline

**Source Stage**

In the source stage, CI/CD pipeline is triggered by a code repository. Any change in the program triggers a notification to the CI/CD tool that runs an equivalent pipeline. Other common triggers include user-initiated workflows, automated schedules, and the results of other pipelines.

**Build Stage**

This is the second stage of the CI/CD Pipeline in which you merge the source code and its dependencies. It is done mainly to build a runnable instance of software that you can potentially ship to the end-user.

Programs that are written in languages like C++, Java, C, or Go language should be compiled. On the other hand, JavaScript, Python, and Ruby programs can work without the build stage.

Failure to pass the build stage means there is a fundamental project misconfiguration, so it is better that you address such issue immediately.

**Test Stage**

Test Stage includes the execution of automated tests to validate the correctness of code and the behaviour of the software. This stage prevents easily reproducible bugs from reaching the clients. It is the responsibility of developers to write automated tests.

**Deploy Stage**

This is the last stage where your product goes live. Once the build has successfully passed through all the required test scenarios, it is ready to deploy to live server.

**Example of CI/CD Pipeline**

Here is example of CI/CD pipeline:

* **Source Code Control:** Host code on GitHub as a private repository. This will help you to integrate your application with major services and software.
* **Continuous integration:** Use continuous integration and delivery platform CircleCI and commit every code. When the changes notify, this tool will pull the code available in GitHub and process to build and run the test.
* **Deploy code to UAT:**Configure CitcleCI to deploy your code to AWS UAT server.
* **Deploy to production:**You have to reuse continuous integration steps for deploying code to UAT.

**CI/CD pipeline Best Practices**

Here is a CI/CD pipeline best practices:

* Write up the current development process therefore, you can know the procedures that require to change and one that can be easily automated.
* Start off with a small proof of project before going ahead and complete whole development process at once.
* Set up a pipeline with more than one stage in which fast fundamental tests run first.
* Start each workflow from the same, clean, and isolated environment.
* Run open source tools that cover everything from code style to security scanning.
* Setup a better code hub to continuously check the quality of your code by running the standard set of tests against every branch.
* Peer code review each pull request to solve a problem in a collaborative manner.
* You have to define success metrics before you start the transition to CD automation. This will help you to consistently analyze your software, developing progress help refining where needed.

**Advantages of CI/CD pipelines**

Here are the pros/ benefits of CI/CD Pipeline:

* Builds and testing can be easily performed manually.
* It can improve the consistency and quality of code.
* Improves flexibility and has the ability to ship new functionalities.
* CI/CD pipeline can streamline communication.
* It can automate the process of software delivery.
* Helps you to achieve faster customer feedback.
* CI/CD pipeline helps you to increase your product visibility.
* It enables you to remove manual errors.
* Reduces costs and labour.
* CI/CD pipelines can make the software development lifecycle faster.
* It has automated pipeline deployment.
* A CD pipeline gives a rapid feedback loop starting from developer to client.
* Improves communications between organization employees.
* It enables developers to know which changes in the build can turn to the brokerage and to avoid them in the future.
* The automated tests, along with few manual test runs, help to fix any issues that may arise.

**Important CI/CD tools**

Here are the important CI/CD tools:

**Jenkins**

Jenkins is an open-source Continuous Integration server that helps to achieve the Continuous Integration process (and not only) in an automated fashion. Jenkins is free and is entirely written in Java.

Jenkins is a widely used application around the world that has around 300k installations and growing day by day.

**Features:**

* Jenkin will build and test code many times during the day.
* Automated build and test process, saving timing, and reducing defects.
* The code is deployed after every successful build and test.
* The development cycle is fast.

**Link:** <https://www.jenkins.io/download/>

**Bambo**

[Bamboo](https://www.atlassian.com/software/bamboo) is a continuous integration build server that performs - automatic build, test, and releases in a single place. It works seamlessly with JIRA software and Bitbucket.

**Features:**

* Run parallel batch tests
* Setting up Bamboo is pretty simple
* Per-environment permissions feature allows developers and QA to deploy to their environments
* Built-in Git branching and workflows. It automatically merges the branches.

**Link:** <https://www.atlassian.com/software/bamboo>

**CircleCI**

[Circle CI](https://circleci.com/) is a flexible CI tool that runs in any environment like a cross-platform mobile app, Python API server, or Docker cluster. This tool reduces bugs and improves the quality of the application.

**Features:**

* Allows to select Build Environment
* Supports many languages including C++, JavaScript, NET, PHP, Python, and Ruby
* Support for Docker lets you configure a customized environment.
* Automatically cancel any queued or running builds when a newer build is triggered.

**Link:** <https://circleci.com/>

**Why Does the CI/CD Pipeline Matter for IT Leaders?**

* CI/CD pipeline can improve reliability.
* It makes IT team more attractive to developers.
* CI/CD pipeline helps IT leaders, to pull code from version control and execute software build.
* Helps to move code to target computing environment.
* Enables project leaders to easily manage environment variables and configure for the target environment.
* Project managers can publish push application components to services like web services, database services, API services, etc.
* Providing log data and alerts on the delivery state.
* It enables programmers to verify code changes before they move forward, reducing the chances of defects ending up in production.

**Ci/CD Pipeline KPI**

* **Cycle or Deployment Time:**Cycle time is the time taken to go from the build stage to production. You can obtain average life cycle time by measuring the development process phases. This metric will give insight into bottlenecks in your process and the overall speed of development time.
* **Development Frequency:**Development frequency allows you to analyse bottlenecks you find during automation. The more frequent smaller releases reduce the risk of defects and fix them when found. Such a metric is an overall measure of your team efficiency.
* **Change Lead Time:**Itmeasures the start time of the development phase to deployment. This metric is an indicator of the entire development process and how well the team works together.
* **Change Failure Rate:**It focuses on the number of times development get succeeds vs. the number of times it fails.
* **MTTR vs. MTTF:**MTTR (Mean Time to Recovery) is the amount of time required by your team to recover from failure. MTTF (Mean Time to Failure) measures the amount of time between fixes and outages. These metrics are a reflection of the team's ability to respond and fix issues.

**Summary**

* A CI/CD pipeline automates the process of software delivery.
* CI/CD pipeline introduces automation and continuous monitoring throughout the lifecycle of a software product.
* Continuous integration is a software development method where members of the team can integrate their work at least once a day.
* Continuous delivery is a software engineering method in which a team develops software products in a short cycle.
* Continuous deployment is a software engineering process in which product functionalities are delivered using automatic deployment.
* There are four stages of a CI/CD pipeline 1) Source Stage, 2) Build Stage, 3) Test Stage, 4) Deploy Stage.
* Important CI/CD tools are Jenkins, Bambo, and Circle CI.
* CI/CD pipeline can improve reliability.
* CI/CD pipeline makes IT team more attractive to developers.
* Cycle time is the time taken to go from the build stage to production.
* Development frequency allows you to analyse bottlenecks you find during automation.
* Change Lead Time measures the start time of the development phase to deployment.
* Change Failure Ratefocuses on the number of times development get succeeds vs. the number of times it fails.
* MTTR (Mean Time to Recovery) is the amount of time required by your team to recover from failure.
* MTTF (Mean Time to Failure) measures the amount of time between fixes and outages.

|  |  |
| --- | --- |
| ***Candidate Profile*** |  |
|  | MD DIN ISLAM |
| **Contact Information***(Phone & Email)* | (646)644-6922 & [mddinislamqa@gmail.com](mailto:mddinislamqa@gmail.com) |
| **LinkedIn URL***(if available)* | Sorry, I don't maintain social media yet. |
| **Current Location***(City & State/Country)* | New York City, New York |
| **Available Work Locations** | Yes |
| **Estimated Daily Commute** | 45 min |
| **Re-location Details***(if applicable)* | Yes |
| **Employment Type** | W2 |
| **Layers?** | - |
| **Pay Rate** | $45/hr. |
| **Notice Period Required** | 2 Week |
| **Additional Notes**:  Level:      Years of Experience: Upcoming Vacation: | |
|  | |

|  |  |  |
| --- | --- | --- |
| ***Work Eligibility*** |  | |
| **Work Authorization** | | US Citizen |
| **Expiration Date (if Visa/EAD)** | |  |
| **Extension Available (if Visa/EAD)** | |  |
| **Additional Notes**: | | |

|  |  |
| --- | --- |
| DOB (MM/DD) | 03-04-1988 |
| Last 5 Digits of SSN | 95022 |