DevOps

* A method to help engineering teams build products continuosly getting user feedback

Git

* Git is used for cloud based code change collaboration

Developer – develops software

* Programmers
* Database developers

IT operations – manages databases, manage applications

* Build engineer
* Production engineer
* Application support
* DBAs

DevOps – Developers and IT operators come together

* Source code management
* Build automation
* Test automation
* Infrastructure and provisioning
* Deployment and monitoring

DevOps engineering pillars

* Pull request (merge request) automation: Developer request to merge new code
* Deployment automation
* Application performance management

DevOps managers – Principles and practices

DevOps engineers – Continuous build, continuous integration, automated provisioning, incremented testing

DevOps culture – DevOps principles, DevOps practices and DevOps tools

DevOps principles – 2 main principles (Automated and incremental)

* Iterative – Final product is not going to be achieved in the first try, repeat the process (Process – Designing → coding → building → testing → configuration provisioning → deployment → monitoring)
* Incremental – Start small and keep building
* Automated – Ultimate goal of DevOps is NoOps
* Continuous
* Collaborative - Discussion
* Self-service - Independency
* Holistic – Overall knowledge about the process

Continuous testing – Need build automation framework and test automation framework to continuously build newly added codes and test them automatically

Continuous integration – Integration of source code management (Coding), build automation (Building), test automation framework (Testing), static code review tool, code coverage tool, etc happening automatically

Continuous delivery – Release integrated with continuous integration automatically

Continuous deployments – Deployment integrated with continuous delivery automatically

Software development life cycle (SDLC)

* Waterfall model – Rigid and one-way method
  + Planning
  + Designing
  + Coding
  + Compiling and testing
  + Configuration and provisioning
  + Deployment and monitoring
* Iterative model – Waterfall model continuous repeatedly
* DevOps pipeline – Automated process for copying files, configuration changes, mailing, monitoring servers

DevOps engineer’s knowledge area – Need DevOps principles, DevOps practices and DevOps tools

* Source code management tools
* Build automation tools
* Test automation tools
* Infrastructure automation tools
* Continuous integration
* Continuous delivery
* Continuous deployment
* Intelligent monitoring

Source code management – Developers, QA and Ops gets involved

Version control system – Tracks who modified the code, when, by who and which feature

git

* Distributed system (Not centralized - has local repository)
* Open source
* Has a feature called Webhook which works with Jenkins and Bamboo

Commit - Fundamental building block of git

git commands

* pwd → Print working directory
* cd → Change directory
  + cd devops/ → Open devops folder
  + cd .. → Go back to previous folder
* ls → Show files inside
  + ls -a → Show hidden files too
  + ls -l → Show files in a list
  + ls -la → Both of above
* git log → Shows information about changes done, all commits
* git show (put commit id here) → Show information about changes done in a commit
* git log --help → Shows all codes
* clear → Clear the terminal
* git log --pretty=short → Gives short description of the commits
* git log --pretty=oneline → Gives short description in one line
* git log --decorate --oneline –graph → Gives more decorated description in one line
* vim file1.c → Create file1.c file or access file if it’s already created
* :wq! → Save and exit file (In read mode, by pressing Esc)
* git status → Show the status of git
* git add file1.c → Add untracked file to index area
* git commit -m “Adding file1.c” → Committing file1.c to local repository
* git commit -am “Adding and committing file1.c” → Adding and committing file1
* git push → Send file from local repository to GitHub (Remote repository)
* git push --set-upstream origin master → Send file from local repository while setting upstream
* git show → Shows details on last push
* git show (Put commit id here) → Shows details on the push with the commit id

Command need to add file to GitHub → add, commit and push

git branch → Linear series of commit objects

* git branch → Show the current branch
* git branch (Add new branch name here) → Create new branch
* git checkout (Put branch name here) → Move to different branch

Adding and committing a new file while in a different branch is same as before (While in master branch)

* git push --set-upstream origin devbranch → Push all files in working dir to remote branch in GitHub

When you create a file in master branch it also creates the file in all other branches but when you create a file in a different branch such as ‘devbranch’ here it only creates a file in that particular branch

Generally, master branch only contains main versions so when you add a new feature create a new branch and test out the new feature and then merge it

If the merged feature is working fine then it will be sent to release branch and then master branch

Likewise, master, release, develop and feature branches can be created

* git diff master…devbranch → Show the difference of master branch and a particular branch (devbranch)

Changing a file in a branch and committing it won’t change those changes in the same file in other branches

* git merge devbranch → Merge devbranch to master when you are in master branch. Content on devbranch will be merged to master branch but devbranch will still exist.
* git branch -d devbranch → Delete devbranch

For continuous integration and continuous delivery → CI/CD tools → Jenkins

* Jenkins is open source, easy to use, extensible, instant report, distributed build, robust email notification system
* Need Java to run

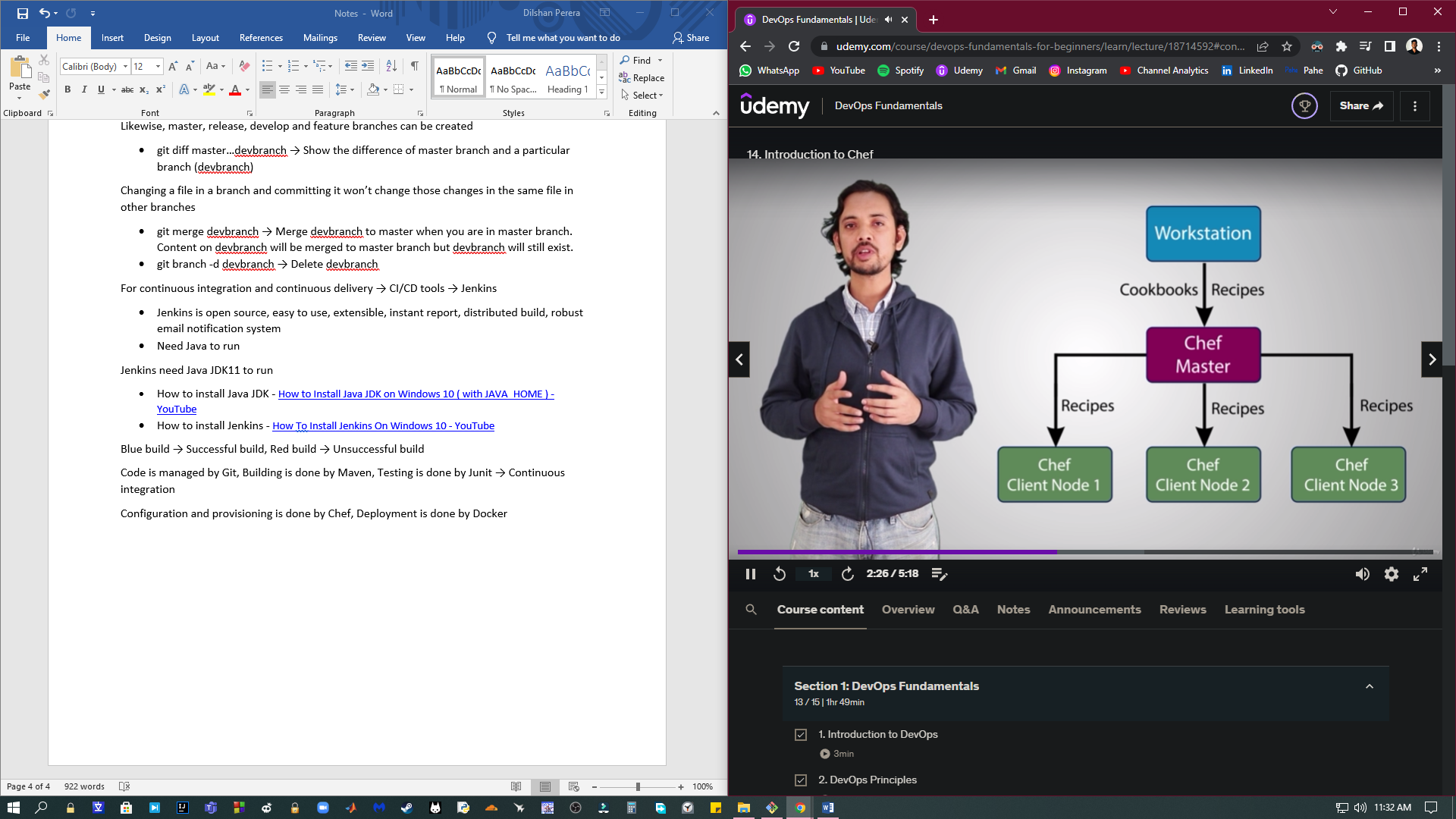
Jenkins need Java JDK11 to run

* How to install Java JDK - [How to Install Java JDK on Windows 10 ( with JAVA\_HOME ) - YouTube](https://www.youtube.com/watch?v=IJ-PJbvJBGs&ab_channel=ProgrammingKnowledge)
* How to install Jenkins - [How To Install Jenkins On Windows 10 - YouTube](https://www.youtube.com/watch?v=1_Zs0gQq1Yc&ab_channel=Mukeshotwani)

Blue build → Successful build, Red build → Unsuccessful build

Code is managed by Git, Building is done by Maven, Testing is done by Junit → Continuous integration

Configuration and provisioning is done by Chef, Deployment is done by Docker



Nodes can be application server, database server, web server, etc

Virtual machines are like houses while containers are like apartments

Advantages of containers are,

* Scalability
* Isolation
* Accurate testing
* Replicable environment
* Resource optimization
* Performance
* High availability