* **WEEK1.md**

Info

View source[**Edit**](https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MLP151/files/edit/refs/heads/master/--/WEEK1.md?region=us-east-2)

Classroom material

**Structure of the program**

* Team
* Agile
* Learn by doing - Project
* Cloud basics(AWS)

**Goals of the program**

* "Learning to learn"
  + Operating Systems & Computer Networks, Problem Solving and Data Structure, Software Development Life Cycle, Agile, Git, TRELLO (Project Management Tool), MySQL, JDBI, Java, Junit, Jmockit, Maven, Javascript/Typescript, HTML5, CSS3, Angular 6, AWS Basics
* Productivity hacks
  + google
  + command line (+ simple IDE)
  + cheat sheets
  + bookmarks
  + check lists
  + REPL
  + tutorials
  + ...
* Communication, teamwork, ...

**Software Development**

* + Tools : source code control, ticket, IDE, build, test, execute
  + Iteration:
    - Assign to yourself
    - Checkout
    - branch
    - change
    - test
    - deploy/test
    - checkin
    - review
    - close
    - delete branch

**Concept Learning**

* Operating Systems & Computer Networks
* Problem Solving and Data Structure
* Unix Commands and Version Control System(GIT)
* Maven
* Software Development Life Cycle
* Agile,TRELLO (Project Management Tool)

**Git**

* workspace, staging, local, remote

**Agile basics**

* Sprint
* Product Backlog, Sprint Backlog
  + Epics, User Stories, Tasks, Bug
  + To Do, In Progress, Resolved, Accepted
* Sprint Rituals
  + Planning
  + Standup
  + Review
  + Retrospective

Workshop material - **Day #1** - AWS CodeCommit/Visual Studio (VS) Code

**Setting up AWS CodeCommit**

* Open: <https://635681506020.signin.aws.amazon.com/console> in your browser; bookmark this page in your browser
* Type your IAMS user name as {hexaware e-mail Id}-Hexaware, e.g, KrishnaKumar-Hexaware and "hexaware123" as your initial sign-in password
* After you sign in for the first time, you must change your password.

**Setting Up the Public and Private Keys for Git and AWS CodeCommit**

* + Step 1: Open the Git Bash emulator; From the emulator, run the ssh-keygen command, and follow the directions to save the file to the .ssh directory for your profile.
  + For example:
    - $ ssh-keygen
  + Generating public/private rsa key pair.
  + Enter file in which to save the key (/drive/Users/user-name/.ssh/id\_rsa): Pree Enter [to save the key in C:\users\Hvuser.ssh\id\_rsa)
  + Enter passphrase (empty for no passphrase): Pree Enter [do not use passphrase]
  + Enter same passphrase again: Press Enter
  + Your identification has been saved in drive/Users/user-name/.ssh/id\_rsa.
  + Your public key has been saved in drive/Users/user-name/.ssh/id\_rsa.pub.
  + The key fingerprint is:
    - 45:63:d5:99:0e:99:73:50:5e:d4:b3:2d:86:4a:2c:14 user-name@client-name
  + The key's randomart image is:

+--[ RSA 2048]----+

| E.+.o\*.++|

| .o .=.=o.|

| . .. \*. +|

| ..o . +..|

| So . . . |

| . |

| |

| |

| |

+-----------------+

* + This generates: • The id\_rsa file, which is the private key file. • The id\_rsa.pub file, which is the public key file.
  + Step 2: Run the following commands to display the value of the public key file (id\_rsa.pub) cd .ssh notepad id\_rsa.pub
  + Copy the contents of the file, and then close Notepad without saving. The contents of the file will look similar to the following:

ssh-rsa EXAMPLE-AfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJB

gNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb2

5zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhc

NMTEwNDI1MjA0NTIxWhcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAw

DgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDAS=EXAMPLE user-name@computer-name

* + Step 3: Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
  + Step 4: In the IAM console, in the navigation pane, choose Users, and from the list of users, choose your IAM user
  + Step 5: On the user details page, choose the Security Credentials tab, and then choose Upload SSH public key.
  + Step 6: Paste the contents of your SSH public key into the field, and then choose Upload SSH public key
  + Step 7: Copy or save the information in SSH Key ID (for example, APKAEIBAERJR2EXAMPLE).
  + Step 8: In the Bash emulator, type the following commands to create a config file in the ~/.ssh directory, or edit it if one already exists: cd .. [if you are in .ssh directory already] vi ~/.ssh/config
  + Step 9: Add the following lines to the file, where the value for User is the SSH key ID you copied earlier, and the value for [User shift+ p to copy the following lines in vi editor]

Host git-codecommit.\*.amazonaws.com

User APKAEIBAERJR2EXAMPLE [user i(insertmode)to edit the user id]

IdentityFile ~/.ssh/id\_rsa

* + - Use :wq to save the file
  + Step 10: Run the following command to test your SSH configuration:
  + ssh git-codecommit.us-east-2.amazonaws.com
  + You will be asked to confirm the connection because git-codecommit.us-east-2.amazonaws.com is not yet included in your known hosts file.
  + The AWS CodeCommit server fingerprint is displayed as part of the verification (a9:6d:03:ed:08:42:21:be:06:e1:e0:2a:d1:75:31:5e for MD5 or
  + 3lBlW2g5xn/NA2Ck6dyeJIrQOWvn7n8UEs56fG6ZIzQ for SHA256).
  + Command Line/Gitbash -- Open Gitbash
  + pwd -- Check and verify that the current working directory is C:\users\Hvuser
  + cd workspace -- change current working to workspace
  + pwd -- Check and verify that the current working directory is C:\users\Hvuser/workspace
  + Git (setup/checkout/clone/pull/push)
  + git --version // should be atleast 2.14+
  + git config --global -l // should throw an error
  + git config --global user.name "<your name>"
  + git config --global user.email <your email>
  + git config --global -l

**Clone repository at Gitbash**

* + Open <https://console.aws.amazon.com/codecommit>
  + Use the following URL to access your repository
  + [https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MLP151/browse?region=us-east-2#](https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MLP151/browse?region=us-east-2)
  + Choose Clone URL, and then copy the SSH URL.
  + Go back to Gitbash and clone the project
  + pwd -- Check and verify that the current working directory is C:\users\Hvuser/workspace
  + git clone ssh://git-codecommit.us-east-2.amazonaws.com/v1/repos/MLP151
  + cd MLP151
  + git status
  + open Visual Studio (VS) Code
  + Open folder c:\users\Hvuser\workspace\MLP151
  + Browse the directories to understand the repository structure

Workshop material - Day #2 - MySQL Workbench/Cli

* Open MySQL Workbench
* CREATE DATABASE MLP151; and click the lightning button
* CREATE USER 'MLP151'@'localhost' IDENTIFIED BY 'MLP151';
* GRANT ALL ON MLP151.\* TO 'MLP151'@'localhost';
* Open database/database.ddl in VS code
* Copy the entire contents to MySQL Workbench
* Execute the ddl
* Click the table icon against the MENU table in the right-hand side schemas section
* Open database/database.dml in VS code
* Copy the contents to MySQL Workbench
* Execute the dml
* Click the table icon against the MENU table in the right-hand side schemas section. You should see the data just inserted.
* Play around with
  + SELECT with predicates
  + UPDATE statements with predicates
  + DELETE statements with predicates
  + At the end of all the playing around, leave the database with 5 records with ids (101,102..
* Next, we will build and run the java code
* go to gitbash, ensure you are in workspace/MLP151
* cd restservice/canteenmanagement
* mvn compile
* mvn exec:java -Dexec.mainClass=com.hexaware.MLP151.util.CliMain
  + As expected, the cli displays only the food id for the MENU; we need to do some code changes before the other attributes
  + such as name will start appearing in the cli.

Workshop material - Day #3 - Git workflow

In this workshop, we will get familiar with

* The git development cycle - see CHEATSHEET.md, git/development cycle
* Branching, Reviewing, Merging
* Resolving conflicts

We will learn to this by playing tic-tac-toe within each group. Play as many rounds as needed to get the hang of how we use git for branching and merging, by letting different members play different roles in each round.

In this, let's use TTT-99 for the Trello Project Board Ticket number which is needed for branch naming.

**Playing without conflicts**

Split the group into two teams of two each and one reviewer, say team-X and team-O. Choose who goes first, perhaps by coin toss.

In this, team-X use one desktop in a shared manner and play 'X's while team-O share another desktop and play 'O's.

The team that goes first follows the git process to checkout and change tictactoe.html to add 'X' or 'O' in a cell. The reviewer merges the change - after verifying that the game is not over and the team has added only one cell.

Then the second team plays in a similar manner. This continues until the reviewer decides that the game has been won, or is a draw.

The reviewer then cleans up tictactoe.html. One of the others review and merge this change.

Play as many rounds as needed till everyone is comfortable with git operations. The git commands should become 'muscle' memory - the ability to run these commands without referring to the cheatsheet.

Workshop material - Day #4

**Playing with conflicts**

Here, the team members play their turns individually on their desktops; say team-X has two members X1 and X2. Both start their turns simultaneously, by pulling master and checking out branches.

The reviewer will merge one, but the second pull request can not be auto-merged. So the player who goes second needs to resolve the conflict. S/he looks at the change as made by the first player and her own change and chooses the better of the two. If hers is better, this is another pull request for the reviewer to merge.

Play as many rounds as needed till everyone is comfortable with git conflict resolution.

Workshop material - Day #5 - Configure Project Board in Trello for Managing and Tracking Your Project

* Login to Trello Project Board with {Your Hexaware e-mail id}.hexaware as user name; e.g., [KrishnaKumar.hexaware@gmail.com](mailto:KrishnaKumar.hexaware@gmail.com);
* Please follow the next step to create user name if it is not created
* Create a free e-mail id at gmail with user name {your Hexaware e-mail id}.hexaware. e.g., [KrishnaKumar.hexaware@gmail.com](mailto:KrishnaKumar.hexaware@gmail.com) and
* use this id for Trello login
* Create a board for your project as "MLP151"
* [Team member #1] Create a "Product Backlog" [List] to describe a user story : "Demonstrate team familiarity with git workflow, both as a developer and as a reviewer"
* Using Add Members option, please include others team members in this board
* [Team member #2] Use "Sprint Backlog" [list] to include (as cards) to create the following stories"
  + Summary: "Clean up tictactoe.html before the demonstrations"
  + Summary: "Delete tictactoe.html after the demonstrations"
* [All team members] In the "Sprint Backlog" and create two user stories (through card)
  + Summary: "As a git user, I, [full name], will demo my ability to use git as a developer by making one move on the tictactoe board"
  + Summary: "As a git user, I, [full name], will demo my ability to use git as a review by reviewing and merging another team member's pull request of a tictactoe board change"
  + Assign these user stories to you (develoepr/reviewer) and your neighbour (reviwer/develoepr)
* [Team member #3]
  + Create "Current Sprint" with "MLP151 Sprint 1" as label
  + Drag all stories /task into “To do” list
    - Due Date: start date being today and end date being tomorrow
* [Team member #4]
  + Drag the "Clean up tictactoe.html before the demonstrations" story to "IN PROGRESS" swimlane
  + Assign the user story to yourself using Add Members
  + With a facilitator overlooking your work, follow the git development cycle workflow to clean up the tictactoe.html to produce a clean fresh board
    - Self-merge the pull request
  + Drag the story to "DONE" list
* [Team members, one-by-one]
  + Drag the "develop demo" story to IN PROGRESS
  + With a facilitator overlooking your work, follow the git development cycle workflow to make one move on the tictactoe.html
    - Use your left neighbour in the team as the reviewer. S/he will drag the "reviewer demo" story to IN PROGRESS to do your review and then move that story to DONE.
  + Drag the story to "DONE" after review/merge done, and local git cleaned up
* [Team member #5]
  + Drag the "Delete tictactoe.html after the demonstrations" story to "IN PROGRESS" swimlane
  + Assign the user story to yourself using Add members
  + With a facilitator overlooking your work, follow the git development cycle workflow to delete the tictactoe.html file
    - Self-merge the pull request
  + Drag the story to "DONE" list
  + "Mark as Done" the Product Catalog by assigning "complete" sticker
* *MANDATORY*
  + Each trainee to suggest one online material for the Must-Read, Nice-To-Read and Go-Deep for any topics covered in the classroom sessions or workshop sessions

Reading material

**Must-Read**

Multi-Tier

* <https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture>

Unix commandline

* <http://oliverelliott.org/article/computing/ref_unix/>

Agile

* <https://www.mountaingoatsoftware.com/agile/scrum>
* <https://www.atlassian.com/agile/developer>
* <https://www.atlassian.com/agile/scrum>
* <https://www.atlassian.com/agile/ceremonies>

Trello

* <https://help.trello.com/>
* <https://trello.com/b/6otslC4i/template-board-agile-with-trello>
* <https://trello.com/b/ZqN99gGN/agile-sprint-board>

Git

* <https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow> (We use this, w/ the rebase variant)

AWS CodeCommit

* <https://docs.aws.amazon.com/codecommit/latest/userguide/setting-up-ssh-windows.html>

**Nice-To-Read**

* <https://git-scm.com/docs>

**Go-Deep**

* <https://git-scm.com/book/en/v2>

**Go to resource**

Find top-level resources in Developer Tools in US East - (Ohio)

Cancel

**[Feedback](https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MLP151/browse/refs/heads/master/--/WEEK1.md)**

**[English (US)](https://us-east-2.console.aws.amazon.com/codesuite/codecommit/repositories/MLP151/browse/refs/heads/master/--/WEEK1.md)**