**Introduction to GitHub**

Welcome to Introduction to GitHub After watching this video, you will be able to: Describe the purpose of source repositories and explain how GitHub satisfies the needs of a source repository.

Linux development in the early 2000’s was managed under a free-to-use system known as BitKeeper. In 2005, BitKeeper changed to a for-fee system which was problematic for Linux developers for many reasons.

Linus Torvalds led a team to develop a replacement source-version control system. The project ran in a short a timeframe and the key characteristics were defined by a small group. These include:

* Strong support for non-linear development. (Linux patches were then arriving at a rate of 6.7 patches per second) Distributed development.
* Each developer can have a local copy of the full development history.
* Compatibility with existing systems and protocols. This was necessary to acknowledge the diversity of the Linux community.
* Efficient handling of large projects.
* Cryptographic authentication of history. This makes certain that distributed systems all have identical code updates.
* Pluggable merge strategies. Many pathways of development can lead to complex integration decisions that might require explicit integration strategies.

What is special about the Git Repository model?

Git is designed as a distributed version-control system. Primarily focused on tracking source code during development.

Contains elements to coordinate among programmers, track changes, and support non-linear workflows.

Created in 2005 by Linus Torvalds for distribution of Linux kernels.

Git is a distributed version-control system that is used to track changes to content. It serves as a central point for collaboration with a particular focus on agile development methodologies.

In a central version control system, every developer needs to check out code from the central system and commit back into it.

As Git is a distributed version control, each developer has a local copy of the full development history, and changes are copied from one such repository to another. Each developer can act as a hub.

When Git is used correctly, there is a main branch that corresponds to the deployable code. Teams can continuously integrate changes that are ready to be released and can simultaneously work on separate branches in between releases.

Git also allows centralized administration of tasks with access-level controls for each team. Git can co-exist locally such as through the GitHub Desktop client or it can be used directly through a browser connected to the GitHub web interface.

IBM Cloud is based on sound and established open-source tools including Git repositories, often called repos.

GitHub is an online hosting service for Git repositories. GitHub hosted by a subsidiary of Microsoft.

GitHub offers free, professional and enterprise accounts. As of August 2019, GitHub had over 100M repositories.

A Repository is: A data structure for storing documents including application source code. A repository can track and maintain version-control.

GitLab is a complete DevOps platform, delivered as a single application. GitLab provides access to Git repositories, controlled by source code management.

With GitLab, developers can:

* Collaborate, reviewing code, making comments and helping to improve each other’s code.
* Work from their own local copy of the code.
* Branch and merge code when required.
* Streamline testing and delivery with Built-in Continuous Integration (CI) and Continuous Delivery (CD).