**Git Hooks:**

Like many other Version Control Systems, Git has a ***way*** to ***fire off custom scripts*** ***when certain important actions occur***.

There are **two groups** of these hooks:

1. **Client-side hooks** - are **triggered** by **operations** such as **committing** and **merging**.
2. **Server-side hooks** - **run** on **network operations** such as **receiving pushed commits.**

We can use these hooks for all sorts of reasons.

**Installing a Hook:**

The **hooks are all stored in the hooks subdirectory** of the **Git directory**. In most projects, that’s .git/hooks.

When we **initialize** a **new repository** with **git init**, **Git populates** the **hooks directory** with a bunch of example scripts, many of which are useful by themselves; but they also document the input values of each script. All the examples are written as shell scripts, with some Perl thrown in, but any properly named executable scripts will work fine – you can write them in Ruby or Python or whatever language you are familiar with. If you want to use the bundled hook scripts, you’ll have to rename them; their file names all end with .sample.

**To enable a hook script**, **put a file in the hooks subdirectory of your .git directory** that is named appropriately (without any extension) and is executable. From that point forward, it should be called. We’ll cover most of the major hook filenames here.