



# Last Lesson

In the previous lesson we looked at git and what we as developers can use version control for.



***Knowledge check: What are the 3 stages?***

# GitHub & SSH Keys

GitHub can use SSH keys to authenticate push and pull requests to or from a repository. Let's use our terminal to create a new SSH key on our local machine and then look at how we can add our keys to our GitHub enterprise accounts.

If you've not yet created an account in the General Assembly GitHub, simply go to

<https://git-invite.generalassembly.ly/invite>





# Making And Cloning Repositories

1. Go to your Github Account

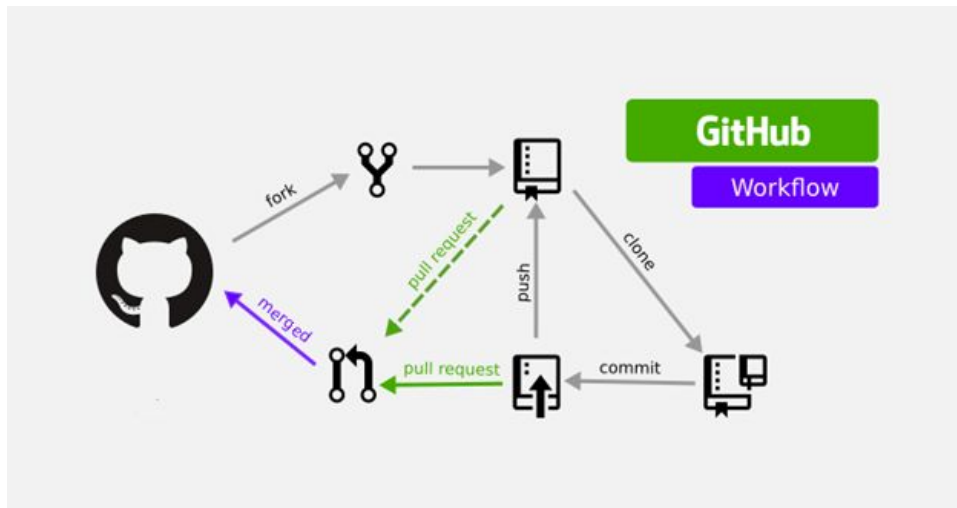
**Continued: `week1_lesson2/01-github.md`**



# What is Forking

The `fork` and `pull` model lets anyone fork an existing repository and push changes to their personal fork without requiring access be granted to the source repository.

Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.





Discussion:

# Forking and Cloning



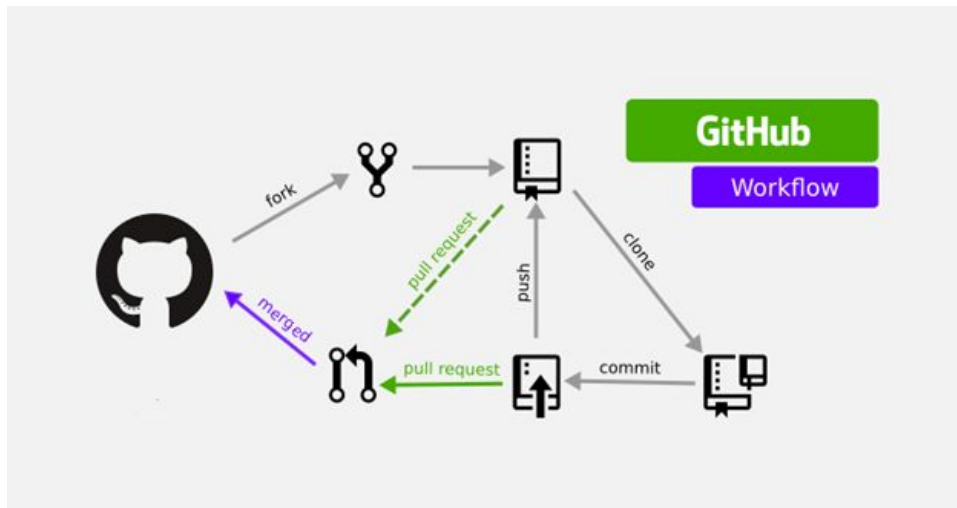
What is the difference between forking and cloning?



# Cloning vs Forking

When you fork a repository, you make a new remote repository that is exactly the same as the original, except you are the owner. You can then `clone` your new fork and `push` and `pull` to it without needing any special permissions.

When you clone a repository, unless you have been added as a contributor, you will not be able to push your changes to the original remote repository because it's not your GitHub repository!

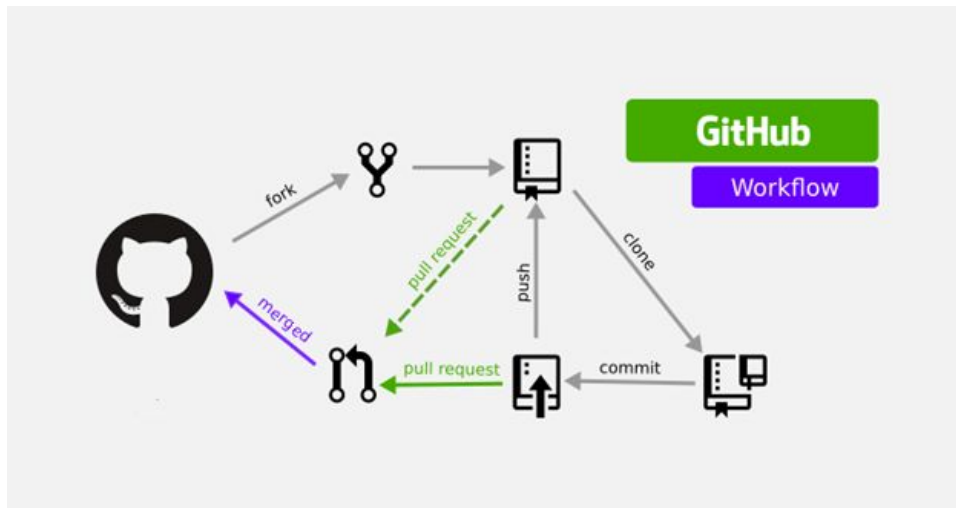


# Pulling

When you want to propose a change to a repository (the original project) that you have forked, you can issue a pull request. This basically is you saying:

*"I've made some changes to your repository, if you want to include them in your original one then you can pull them from my fork!"*

We'll give this a shot in the next section.







## Group Exercise: Review

Use the internet and what you've learned today to answer the following questions with a partner:

- How do I send changes to the staging area?
- How do I check what is going to be committed?
- How do I send the commits to GitHub?
- How do I go back to the previous commit?
- How do I check the configuration on a specific machine?
- How does GitHub know that I am allowed to push to a specific repo?

# Git Ignore

When you create a new project, most integrated development environments usually generate files specific to your computer (i.e. setup files, temporary files, compiled code, etc). These kind of files should not be pushed to the remote Git repository, as they are specific to you alone and might affect other peoples' ability to use the project.

This is where the .gitignore file comes in.

The .gitignore file lists the type of files that should not be uploaded to your remote Git repo (i.e., what files to *ignore*).

You can put .gitignore files in your repo, so whoever clones your project will ignore unnecessary files. You can also set up your computer so you always ignore certain files for all of your projects - a "global gitignore". Let's do the latter now.



# Git Ignore

Go to [gitignore.io](https://gitignore.io), a website that generates .gitignore files. Type in the types of projects you'll be working with (Rails, Android, OSX, Windows), and press *Generate*. Copy all of the generated text.

Okay, now open Terminal and create the .gitignore file wherever you want; I tend to run `touch ~/.gitignore`. Then, open it and paste the generated text into the file. Make sure to save it!

Now, you have to register the file with Git. In Terminal, run:

```
git config --global core.excludesfile ~/.gitignore
```

All of your future projects will ignore the files listed.

*Note:* For local .gitignore files, you don't have to register them with Git. Just put them in the root folder of your Git project.





**Solo Exercise:**

# Switch and Restore

Take a few minutes to read about these new git commands that can replace some instances of the checkout command

<https://www.banterly.net/2021/07/31/new-in-git-switch-and-restore/>



# Conclusion

As a developer, you'll have to use Git pretty much everyday - the learning curve is steep and all the principles of version control can be a bit blurry sometimes, so we ask students to push their homework everyday and to commit regularly during project time.

Don't be frustrated by all the new commands because we will definitely have the time to practice during this course.

- Explain the difference between forking and cloning.
- Describe the steps to initialize a Git repository and link your local repository to a GitHub remote location.



# — Cloud Providers

# Our Objectives

- Define cloud computing.
- Compare and contrast virtualization and containerization and how they relate to cloud computing.
- Explain the benefits of cloud computing.
- Differentiate between the main service models (SaaS, IaaS, PaaS).





Group Exercise:

## Business Drivers



There are many reasons a business may want to migrate to the cloud. Undoubtedly one of the more common reasons for a business making the move is to reduce IT costs but when we scratch the surface it's not hard to find other reasons to migrate.

**Get your thinking cap on:** What reasons might encourage a business to migrate their infrastructure to the cloud?





# The Drivers

Let's have a look at some potential drivers:

- Reduce IT costs
- Increased agility
- Improved security
- No more end-of-life concerns
- Consolidation of data centers
- Scalability
- Environmental benefits



## Group Exercise:

# Business Drivers



You're the CEO of a travel agency that has branch offices in Taiwan, China, Japan, and the United States.

The vast majority of your company's business — \$30 billion TWD's (\$974 million USD's) worth — is conducted via the website. Customers visit the website, review travel destinations, and book online.

Your IT administrators need to make hardware and software updates that take the website offline periodically. This isn't ideal for a travel agency that services customers around the world in various time zones, but you deal with it.

But then, during an effort to upgrade the security settings for the database that stores personal information about your customers (payment details, passport numbers, home addresses, etc.), your website's performance is disrupted for two days. The two days down equates to around \$35 million TWD (\$1.13 million USD) in lost website revenues.

**How do you prevent this from EVER happening again?**





## Group Exercise:

# What is The Cloud?

10 min



In your breakout room teams, define "the cloud."

In addition to covering what it is at a high level, can you answer:

- How does it work?
- How is it purchased?
- What functionality does it provide?

How did it go? Post your answers in Teams



A close-up of Morpheus from the movie The Matrix, wearing his signature black sunglasses and a blue leather jacket. He has a serious, questioning expression. The background is slightly out of focus, showing a red, fleshy, organic structure.

**WHAT IF I TOLD YOU**

**CLOUD COMPUTING IS JUST USING  
SOMEONE ELSE'S COMPUTER?**



Discussion:

# What are the Blockers



What are the potential blockers a business might find when considering moving to the cloud.



# Blockers

- Legacy systems – is anything holding you back?
- Do you have the skills required?
  - Does your team have the skills and the new mindset required?
- Regulations
  - will you experience problems with regulatory requirements, such as where your clients require their data to be stored?
- Cloud governance and change control
  - how will you find a balance of business agility and control?
- Ongoing management
  - how will you support your day-to-day operations? Internal/Outsource
- Service optimisation and monitoring
  - how will you monitor your cloud spend and optimise it?