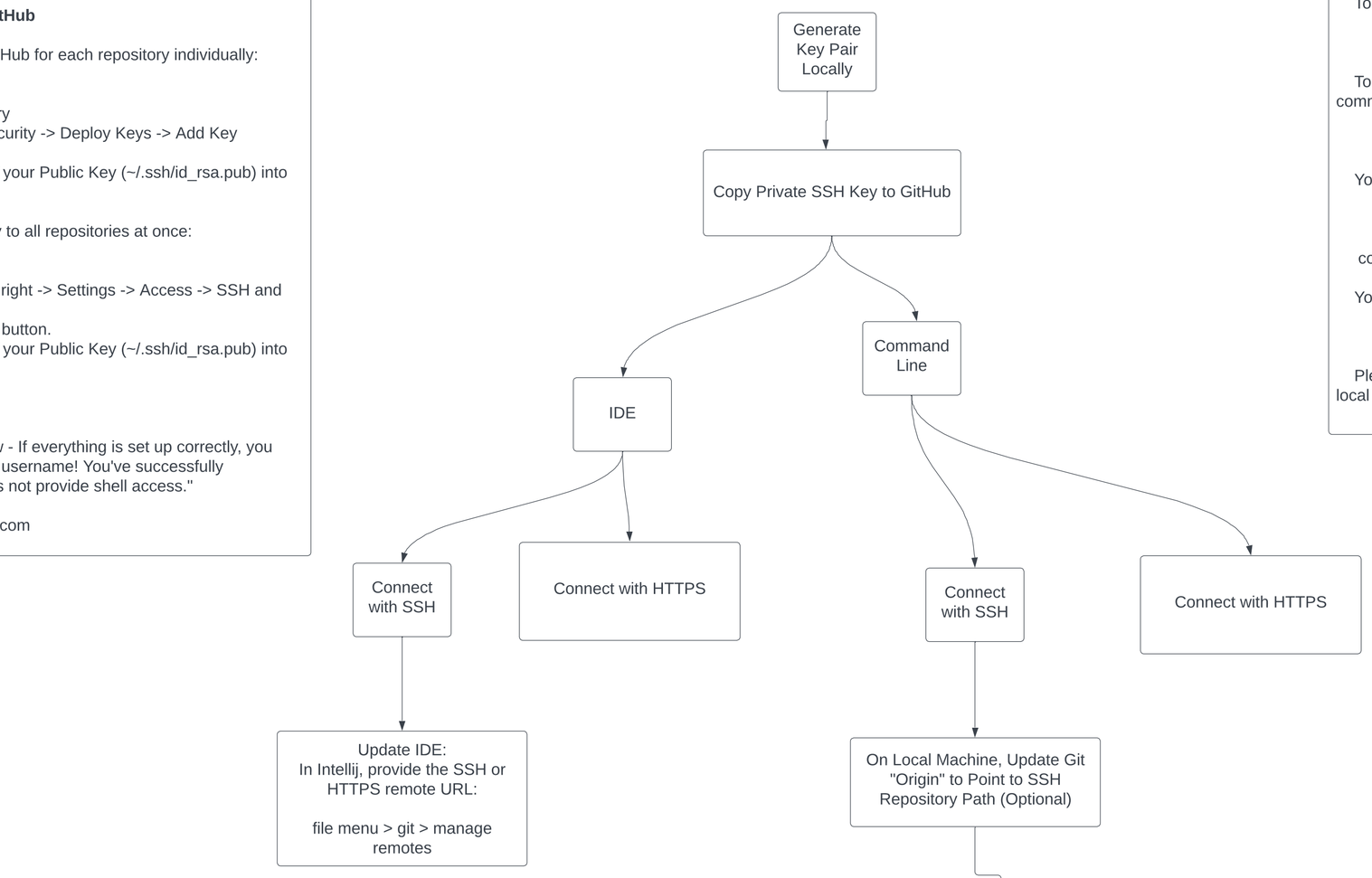
GitHub connect with SSH and HTTPS - Sean Corzo 8/12/2023



COPY PRIVATE SSH KEY TO GITHUB

1. **Create SSH key pair:**

   ssh-keygen -f ~/.ssh/my_key

   # creates private key
   ~/.ssh/id_rsa

   # and public key
   ~/.ssh/id_rsa.pub

2. **Add your SSH key to GitHub**

   Add your SSH key to GitHub for each repository individually:

   Go to Github
   Navigate to Repository
   Under Settings -> Security -> Deploy Keys -> Add Key
   Give a title to the key
   Paste the contents of your Public Key (~/.ssh/id_rsa.pub) into the Key Field

   Or add a single SSH key to all repositories at once:

   Go to your GitHub
   Go profile picture top right -> Settings -> Access -> SSH and GPG keys
   Click "New SSH key" button.
   Paste the contents of your Public Key (~/.ssh/id_rsa.pub) into the Key Field

3. **Test the connection**

   Run the command below - If everything is set up correctly, you should see a message like "Hi username! You've successfully authenticated, but GitHub does not provide shell access."

   ssh -T git@github.com

LISTING GIT CONFIG SETTINGS LIKE USERNAME AND ORIGIN FROM THE COMMAND LINE

To list the Git config settings for your current repository, you can use the git config command.

Here are a few examples of how you can use the git config command to view specific settings:

To view your username, you can use the command

```
git config user.name
```

To view your email, you can use the command

```
git config user.email
```

To view the URL of the remote repository named "origin", you can use the command

```
git config --get remote.origin.url
```

You can also use the

```
git config --list
```

command to list all the config settings for your repository.

You can also list all the global configuration settings by using the command

```
git config --global --list
```

Please note that you should run the command from within the folder of the local repository you want to check.

What is my "Origin" set to?

Use Git Config Command

CONNECT WITH SSH or HTTPS?

The difference between the two remote URLs below lies in the way they authenticate and establish the connection to the remote repository on GitHub. These two URL formats correspond to different authentication methods and protocols.

`git@github.com:username/repository.git`

and

`https://github.com/username/repository.git`

UPDATING THE ORIGIN PATH WITH A NEW ONE

The error message "remote origin already exists" means that you already have a remote repository named "origin" configured for your local repository. To update the origin value with a new one, you can use the git remote set-url command.

The syntax for this command is:

```
git remote set-url origin new_url
```

Where "origin" is the name of the remote repository that you want to update and "new_url" is the new URL for the remote repository.

In your case, you can use the following command:

```
git remote set-url origin git@github.com:username/gcp_endpoints.git
```

This will update the URL for the remote repository named "origin" to the new URL you specified.

After updating the origin URL, you should be able to push and pull changes to and from the new repository without any issues.

CONNECTING TO GITHUB WITH SSH PROTOCOL

git@github.com:username/repository.git

This is an SSH-based remote URL. When you use this format, you're using the SSH protocol for communication with the remote repository. SSH uses public-key cryptography for authentication.

Here's a breakdown of the URL components:

git: Shorthand notation used in Git URLs to indicate that you're using the SSH protocol for accessing the Git repository hosted on GitHub. SSH is handling the secure communication and authentication aspects of the connection.
github.com: The hostname of the remote Git server.
username: Your GitHub username.
repository.git: The name of the repository ending with .git.

With SSH, you need to have an SSH key pair set up on your machine and added to your GitHub account. This key pair consists of a public key stored on GitHub and a corresponding private key stored on your local machine. This method is more secure and efficient, as it doesn't require entering your GitHub credentials every time you interact with the repository.

MANAGING MANY PRIVATE KEYS (OPTIONAL)

In order to ssh to a host without having to specify the private key location every time on the command line, you can set up a config file locally as follows:

config file
~/.ssh/config

When working with Git and SSH, you can set up multiple keys using the ~/.ssh/config file. Here's an example configuration:

```
Host github-repo1
HostName github.com
User git
IdentityFile ~/.ssh/repo1_private_key

Host github-repo2
HostName github.com
User git
IdentityFile ~/.ssh/repo2_private_key
```

CONNECTING TO GITHUB WITH HTTPS

https://github.com/username/repository.git (HTTPS protocol):

This is an HTTPS-based remote URL. When you use this format, you're using the HTTPS protocol for communication with the remote repository. HTTPS relies on username and password or token-based authentication. Here's a breakdown of the URL components:

https: This indicates the protocol to be used.
github.com: The hostname of the remote Git server.
username: Your GitHub username.
repository.git: The name of the repository ending with .git.