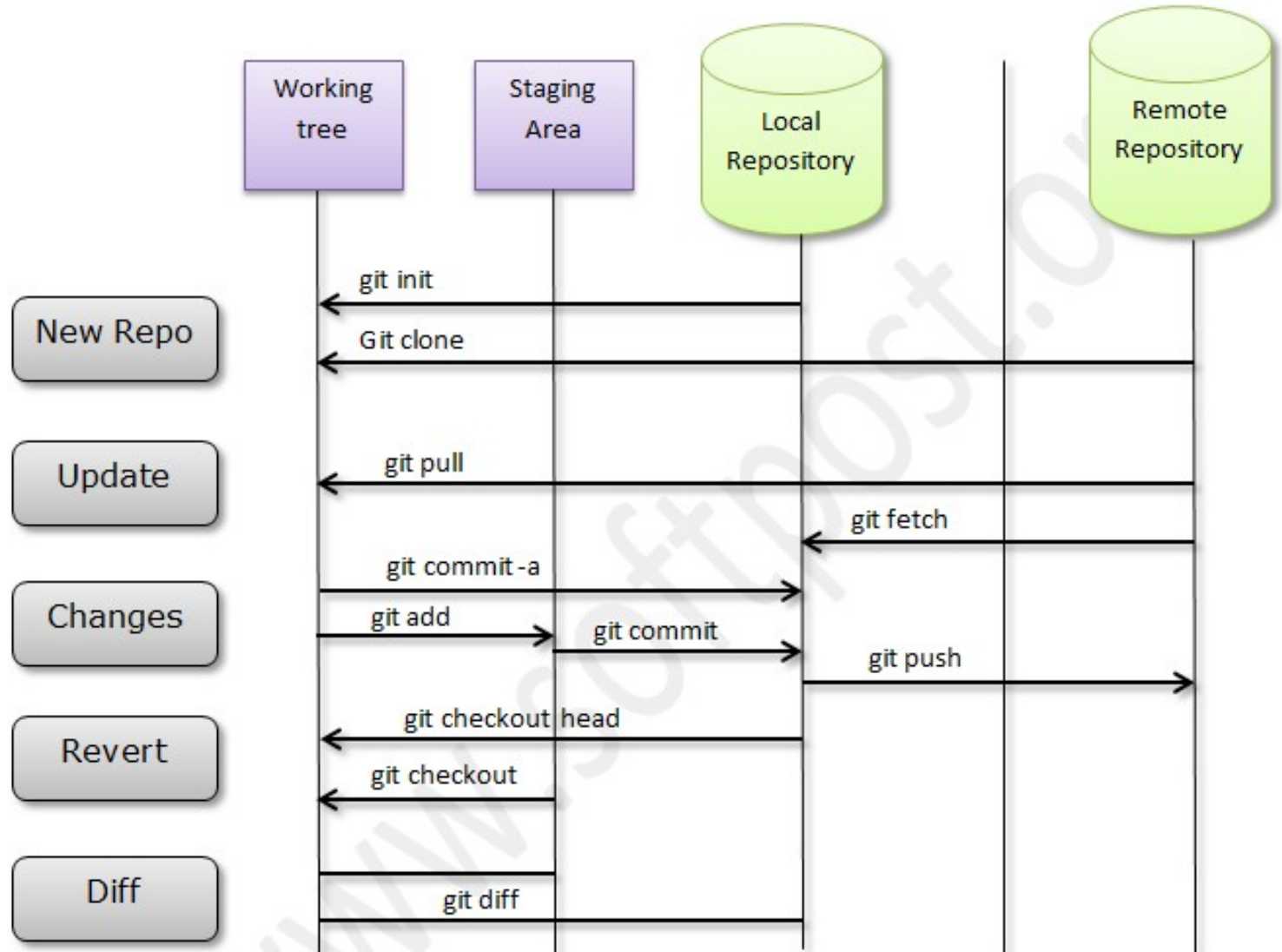




GIT

GIT



```
aureliano@aureliano-N141CU ~$ cat ~/.gitconfig
[user]
    email = aureliano.buendias@outlook.com
    name = Aureliano Sinatra
[includeIf "gitdir:~/home/aureliano/Documents/Aureliano/projects/mydatamodels"]
    path = ~/home/aureliano/Documents/Aureliano/projects/mydatamodels/.gitconfig
```

GIT config file: always configure user /email



GIT BASH

Git is a combination of command line programs that were built to execute on a Unix environment. Operating systems based on Unix, like Linux or macOS, come with Bash (Bourne Again SHell), which ensures that Git runs smoothly across these systems. Basic bash commands

pwd — displays the path of your current working directory;

cd <directory> — navigates to specified directory;

cd .. — navigates to parent directory;

ls — lists directory contents;

mkdir <directory> — creates a new directory;

touch <file> create file

nano <file> file editor



GIT BASH

Git is a combination of command line programs that were built to execute on a Unix environment. Operating systems based on Unix, like Linux or macOS, come with Bash (Bourne Again SHell), which ensures that Git runs smoothly across these systems. Basic git commands

git init — creates a new local repository;

git clone <repository> — clones an existing repository;

git status — lists changed files in your working directory;

git add . — adds all current changes to the next commit;

git commit — commits previously staged files;

git branch <new-branch> — creates a new branch based on your current HEAD;

git checkout <branch> — switches to a different HEAD branch;

git push <remote> <branch> — publishes local changes on a remote repository;

git pull <remote> <branch> — downloads changes from a remote repository and integrates them into HEAD;

git merge <branch> — merges the specific branch into your current HEAD.

SHELL VS BASH SCRIPTING

Using GIT BASH, Create simple shell script example.sh

```
#!/bin/sh
if [[ -z $1 ]];
then
    echo "No parameter passed."
else
    echo "Parameter passed = $1"
fi
```

Make it executable: `chmod a+x example.sh`

Run it: `./example.sh Ynov`

Expected output: "Parameter passed = Ynov"

```
docker run -it -v /home/user/yourscript.sh:/usr/local/src ubuntu-22.04 bash
/usr/local/src/example.sh Ynov
```

EXECUTE SHELL SCRIPTING VIA PIPELINE

Choose a trigger rule

```
on:
  push:
    paths:
      - 'scripts/*'
      - '.github/workflows/*.yaml'
```

Define a job and a runner type

```
jobs:
  example-job:
    runs-on: ubuntu-22.04
```

Checkout the repository and execute script

```
steps:
  - uses: actions/checkout@v2
  - name: List
    run: ls -a
  - name: where I am
    run: pwd
  - name: Run example script
    run: /bin/sh scripts/example.sh Ynov
```

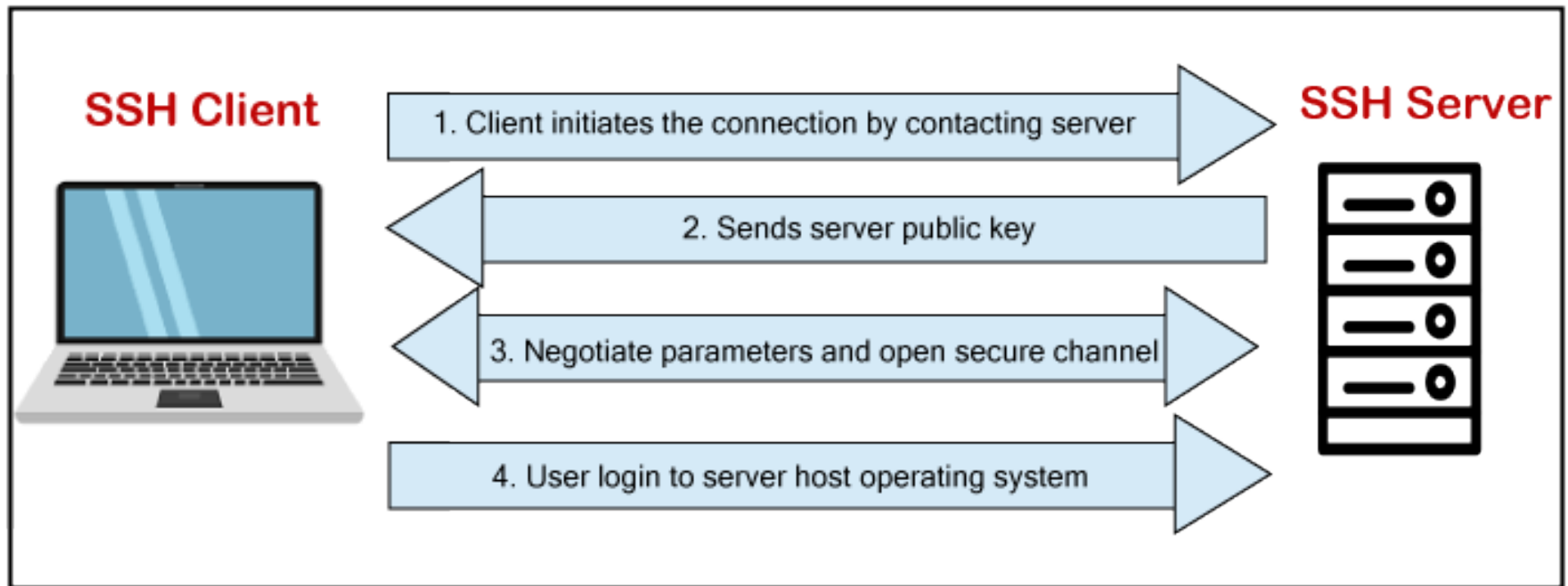
INFRA / CYBERSEC CONFIGURATIONS

- SSH config

<https://linuxhint.com/generate-ssh-key-ubuntu/>

Upload the .pub key to github to enable the ssh connection instead of https

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account>





INFRA / CYBERSEC CONFIGURATIONS

SSH

SSH (Secure Shell) is a public-key cryptography protocol that ensures no one can intercept or change the data during the transfer. Since it is more difficult to set up, it is not as widespread as HTTPS, but it offers greater data integrity and security.

However, **firewalls** on some systems refuse to allow SSH connections on the default port, which can further complicate the setup. Additionally, some operating systems don't have SSH clients installed by default.

HTTPS

HTTPS (Hyper Text Transfer Protocol Secure) is a more widespread network protocol that uses **SSL/TLS** data encryption. Since it is easier to configure than SSH, HTTPS is more common but provides a lower data security level since it doesn't use public-key cryptography.

Git with HTTPS uses token-based authentication to establish connections on port 443 via the Public/Private Pair authentication mode. Port 443 is open in almost every firewall, which isn't always the case for SSH.

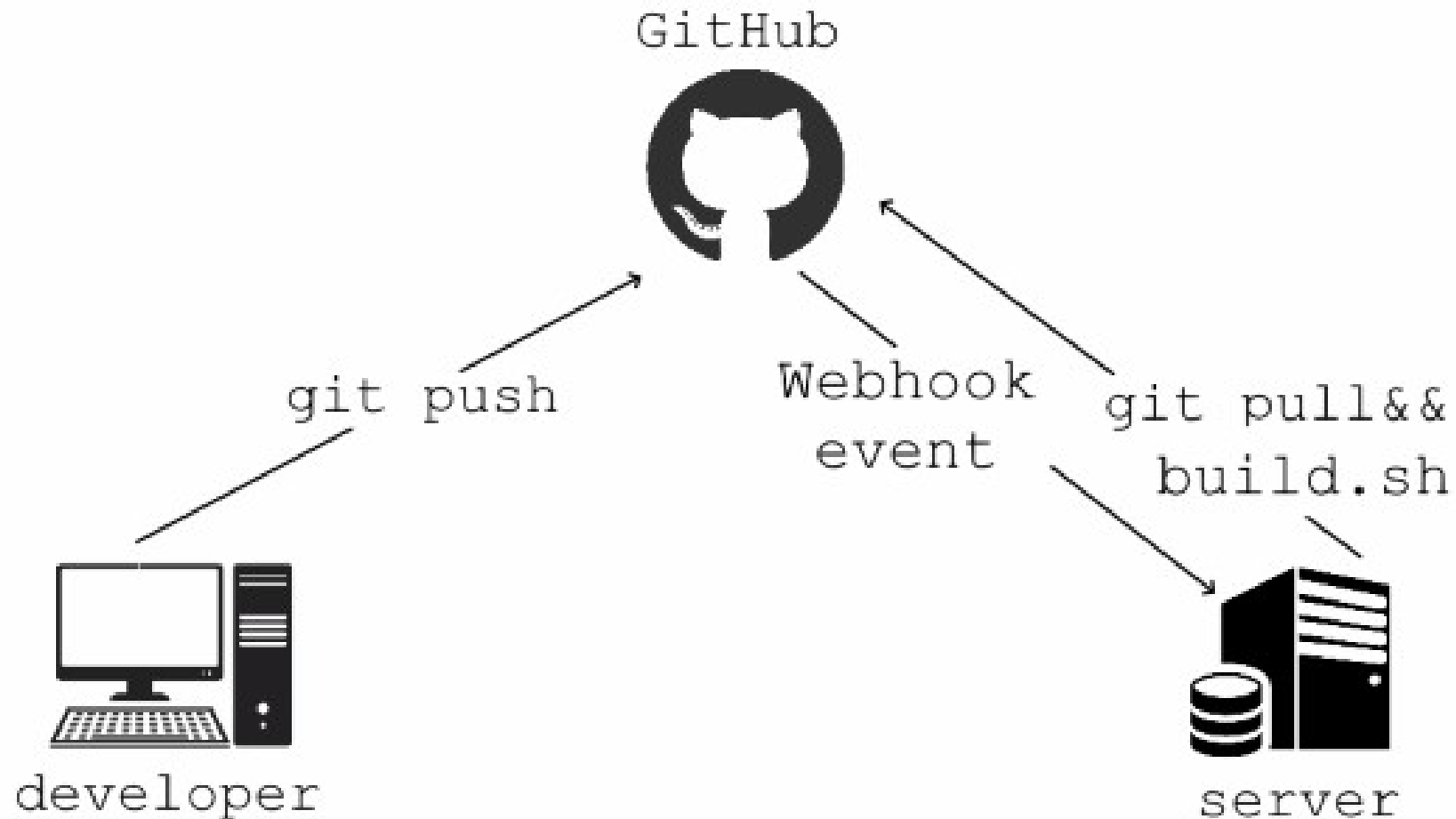
The downside of using HTTPS is that every action, such as **git fetch**, **git pull**, or **git push** asks for your username and password.



Requirements

- Create SSH key:
<https://linuxhint.com/generate-ssh-key-ubuntu/>
- Send public key to aureliano.sinatra@ynov.com
- Git clone ssh repo
`git@github.com:ynov-campus-sophia/devops-B3-2023.git`

GITHUB / GITHUB RUNNER





GITOPS

GitOps est une méthodologie et une pratique qui utilisent les dépôts Git comme une source unique de vérité pour fournir l'infrastructure sous forme de code (**laC**). Elle s'appuie sur les principes et les approches de la culture DevOps et fournit un cadre pour commencer à concrétiser les résultats. GitOps est devenu le choix populaire pour mettre en œuvre et améliorer DevOps, l'ingénierie de plateforme et les opérations de fiabilité des services (SRE).

GitOps est une approche agnostique, et un cadre GitOps peut être construit à l'aide d'outils tels que Git, Kubernetes et des solutions CI/CD. Certains principes de GitOps sont les suivants :

- Git est la source de vérité unique
- Traitez tout comme du code
- Les opérations sont effectuées via les flux de travail Git



GITOPS

- Privilégiez une approche déclarative plutôt qu'une approche programmatique
- Versionné et immuable
- L'état souhaité est stocké de manière à garantir l'immuabilité et la version, tout en conservant un historique complet des versions.
- Récupéré automatiquement : les agents logiciels récupèrent automatiquement les déclarations d'état souhaité depuis la source.
- Réconciliation continue : les agents logiciels observent en continu l'état réel du système et tentent d'appliquer l'état souhaité.