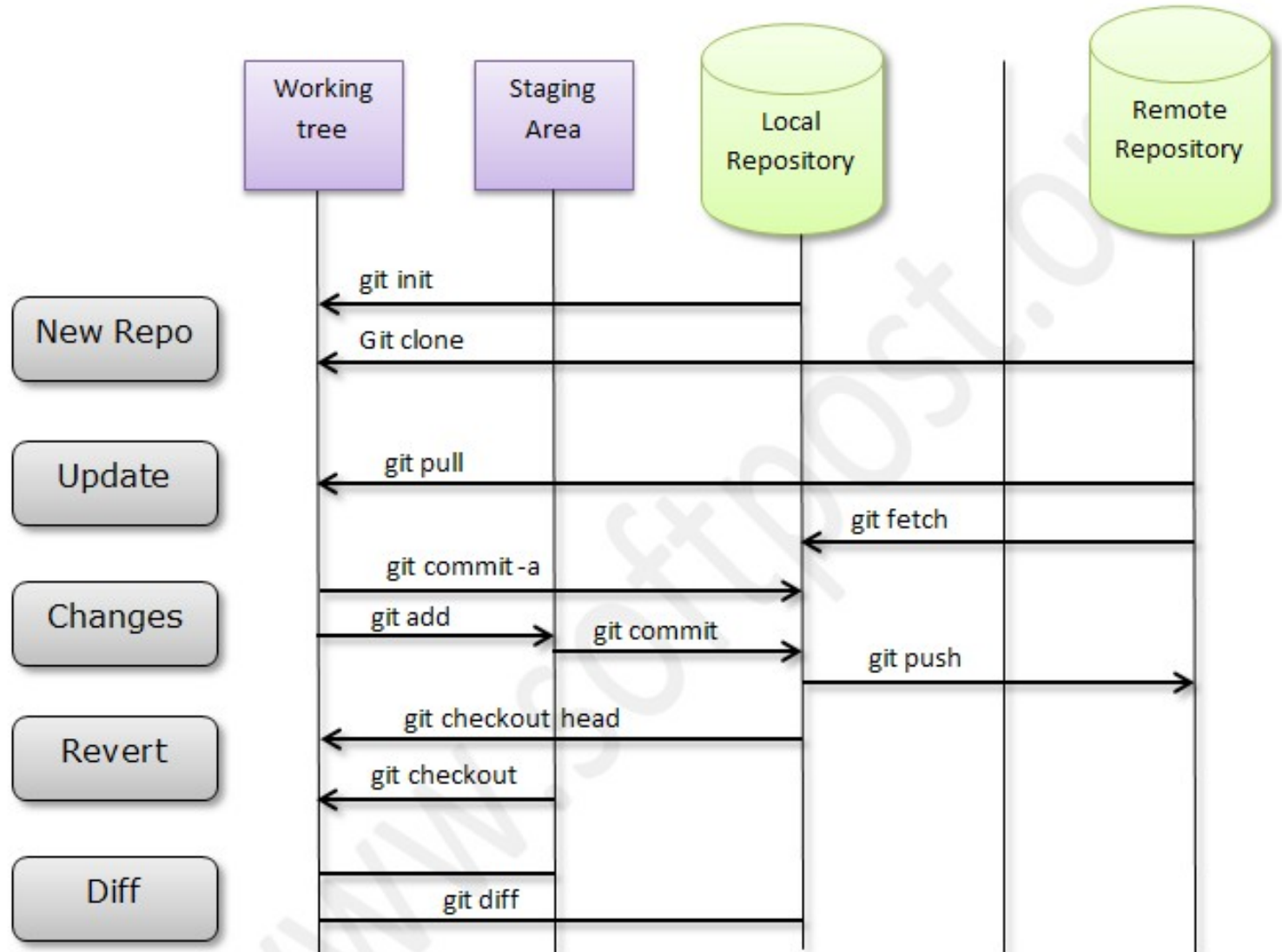




WORKSHOPS

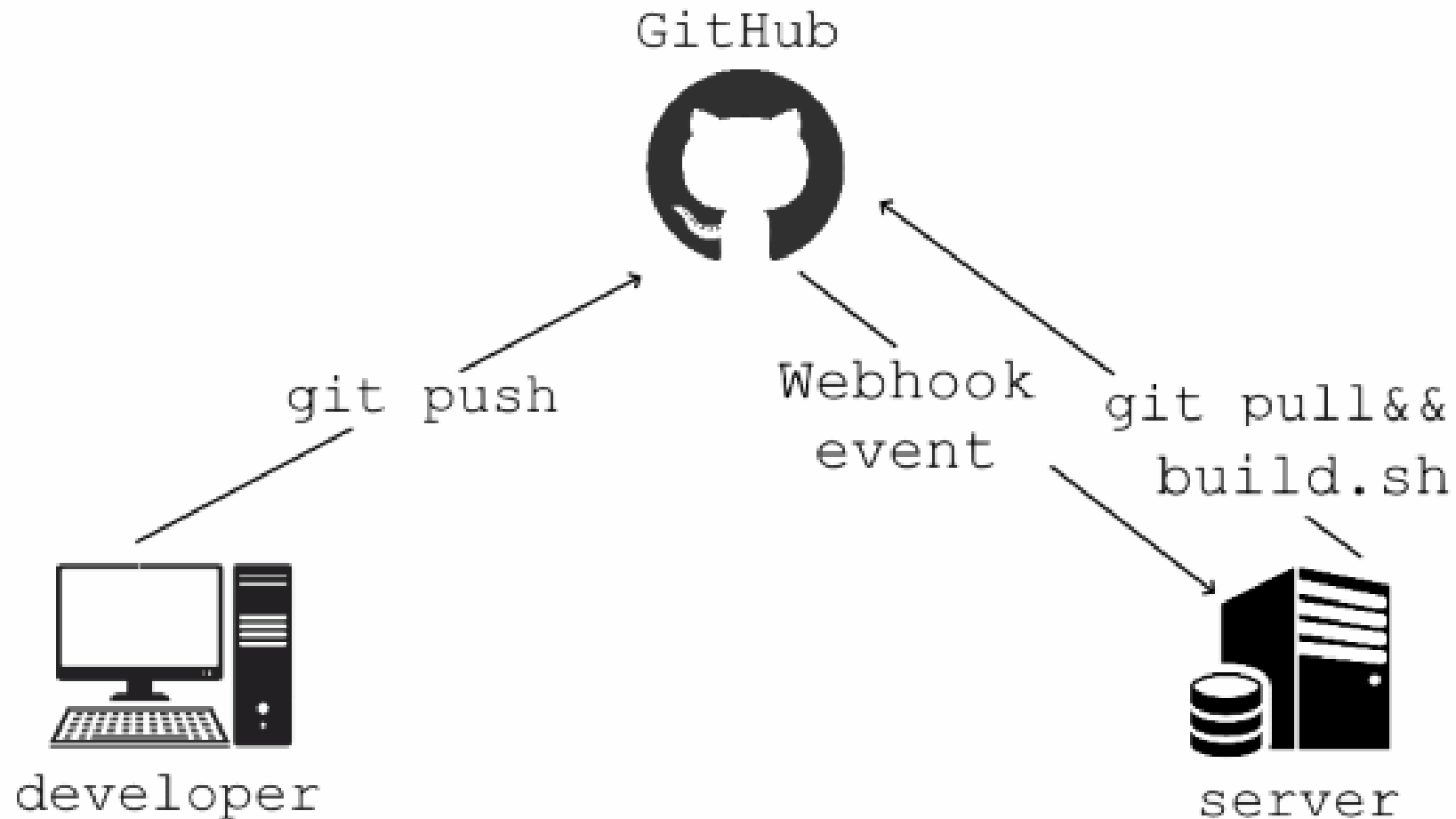
GIT



```
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

GITHUB / GITHUB RUNNER



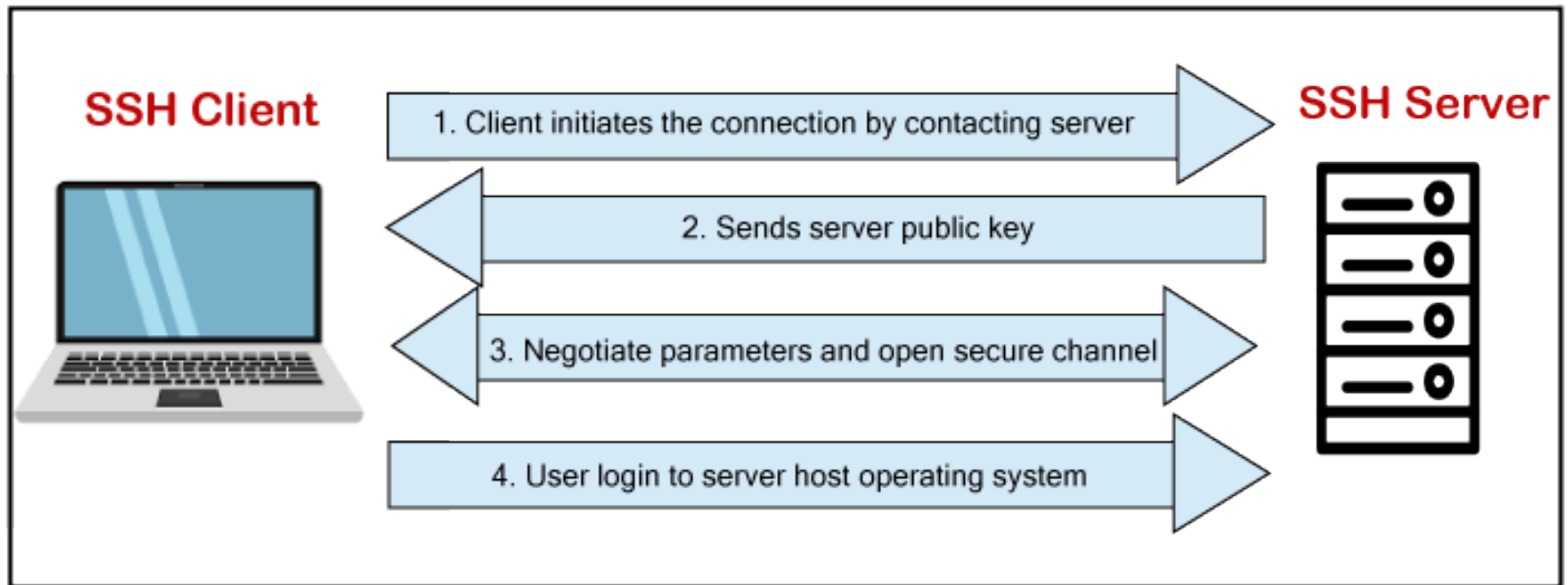
INFRA / CYBERSEC CONFIGURATIONS

- 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

- DOCKER login to a registry : `docker login registry.hub.docker.com -u user -p password`
- DOCKER config file

```
aureliano@aureliano-N141CU ~$ cat .docker/config.json
{
  "auths": {
    "https://index.docker.io/v1/": {
      "auth": "YXNpbmF0cmE6cDBydHVnYWw="
    }
  }
}
```

- DOCKER in a VM
<https://medium.com/@peorth/using-docker-with-virtualbox-and-windows-10-b351e7a34ad>
<https://armstar.medium.com/install-virtualbox-ubuntu-docker-on-windows-10-a16765a09b>
- BASHRC, ALIASES, SHELL SCRIPTING
open the file `~/.bashrc` and create aliases and functions often needed in command line
- DOCTL <https://github.com/digitalocean/doctl> install using apt if possible



AUTOMATION JS

- PACKAGE.JSON

Your project's package.json is the central place to configure and describe how to interact with and run your application. It is used by the npm CLI (and yarn) to identify your project and understand how to handle the project's dependencies.

Name / version / scripts / dependencies / author / licence

- NPM

is a package manager for Node.js packages.

It make possible operations on source code (es. build , package, release, and publish to a registry)

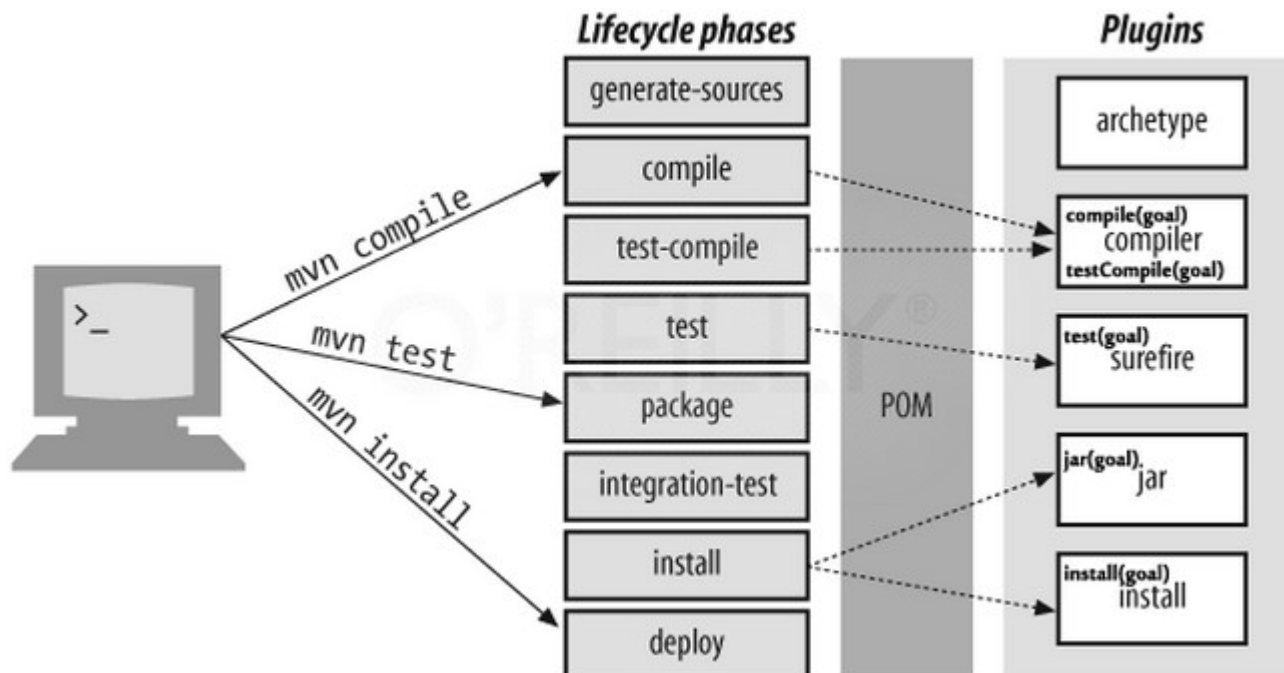
<https://docs.npmjs.com/cli/v6/commands>

<https://docs.npmjs.com/creating-and-publishing-scoped-public-packages>

AUTOMATION JAVA

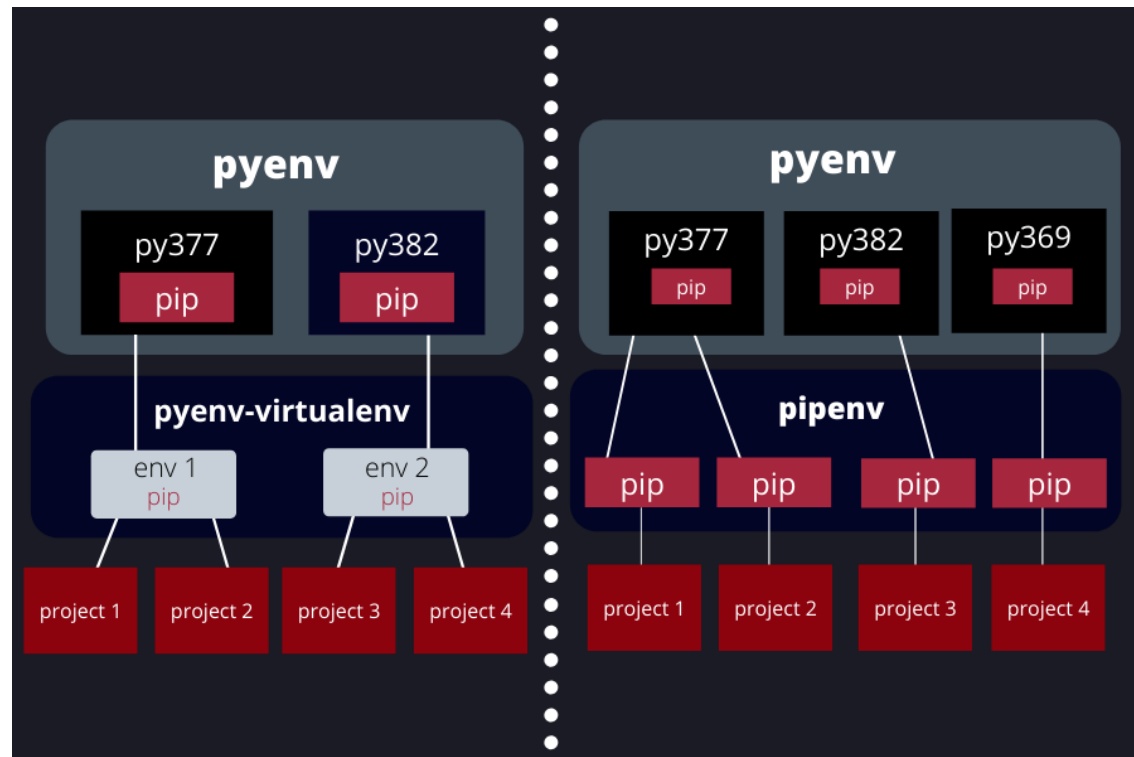
- MAVEN: is a popular open-source build tool developed by the Apache Group to build, publish, and deploy several projects at once for better project management. The tool provides allows developers to build and document the lifecycle framework

<https://www.digitalocean.com/community/tutorials/install-maven-linux-ubuntu>



AUTOMATION PYTHON

- PIP : one of the most famous and widely used package management system to install and manage software packages written in Python and found in Python Package Index (PyPI). Pip is a recursive acronym that can stand for either "Pip Installs Packages"
- Pipfile.lock: is intended to specify, based on the packages present in Pipfile, which specific version of those should be used, avoiding the risks of automatically upgrading packages that depend upon each other and breaking your project dependency tree
- PIPENV manages virtual environments.
- VIRTUALENV is a tool to create isolated Python environments to let possible to work with different python versions.





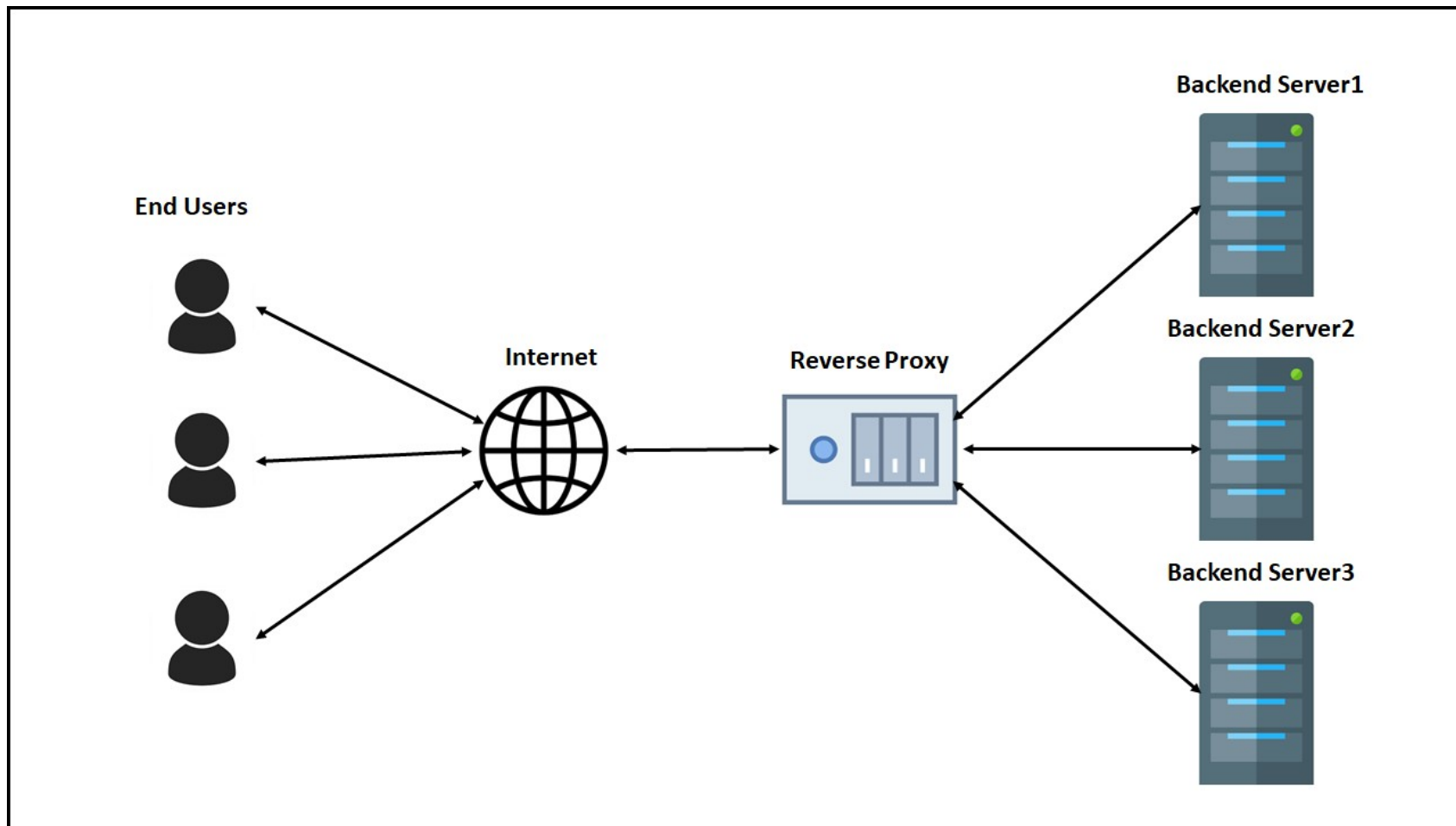
DOCKER COMPOSE

- Docker Compose is a tool that was developed to help define and share **multi-container applications**. With Compose, we can create a YAML file to define the services and with a single command, can spin everything up or tear it all down.
- **Docker-compose up / down / build**
- Is possible to define bootstrap dependencies between container via **depends_on** keyword

<https://github.com/ynov-campus-sophia/devops/tree/master/docker/compose>

ARCHITECTURE

- MICROSERVICE ARCHITECTURE (NGINX + MICROSERVICES)





FRAMEWORKS

- <https://create-react-app.dev/docs/adding-typescript> REACTJS, JS, FRONT
- <https://docs.nestjs.com/> NESTJS, JS, API
- <https://flask.palletsprojects.com/en/2.2.x/> FLASK , PYTHON , API
- <https://spring.io/guides/gs/spring-boot/> SPRING BOOT, JAVA, API



Examples with python pip

- Hands on pipenv, lockfile generation
- Reduce size of image using a slim base layer (from ubuntu to debian slim OS)
- Security hardening (from debian to alpine) + hands on USER keyword to avoid run as superuser in a container
- 2 stages builds