

Continuous everything

Integration, Delivery, Deployment



Agenda

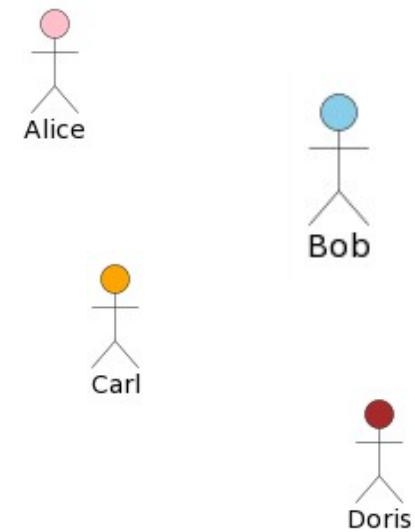
- First pieces of software product lines
 - Dev. Machine
 - Share for collaboration
 - Share for dependencies
 - SemVer
 - Build versus Run dependencies
 - Remote Build
- Virtualisation
 - Hypervisors
 - Containers
 - Dockerfile
- Continuous Integration
 - Pipelines
 - Push versus Merge Request
 - Stages
 - Gitlab Pages
- Conclusion

Software product lines

First pieces of ...

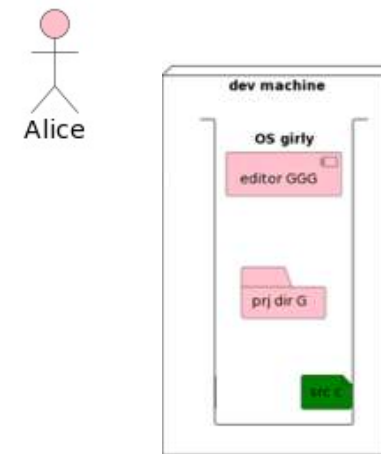
Story telling

- Let's see the major tools of a software factory (in agile context) with the needs of a growing team
 - SCM (Source Code manager)
 - Dependency Manager
 - Continuous X Engine

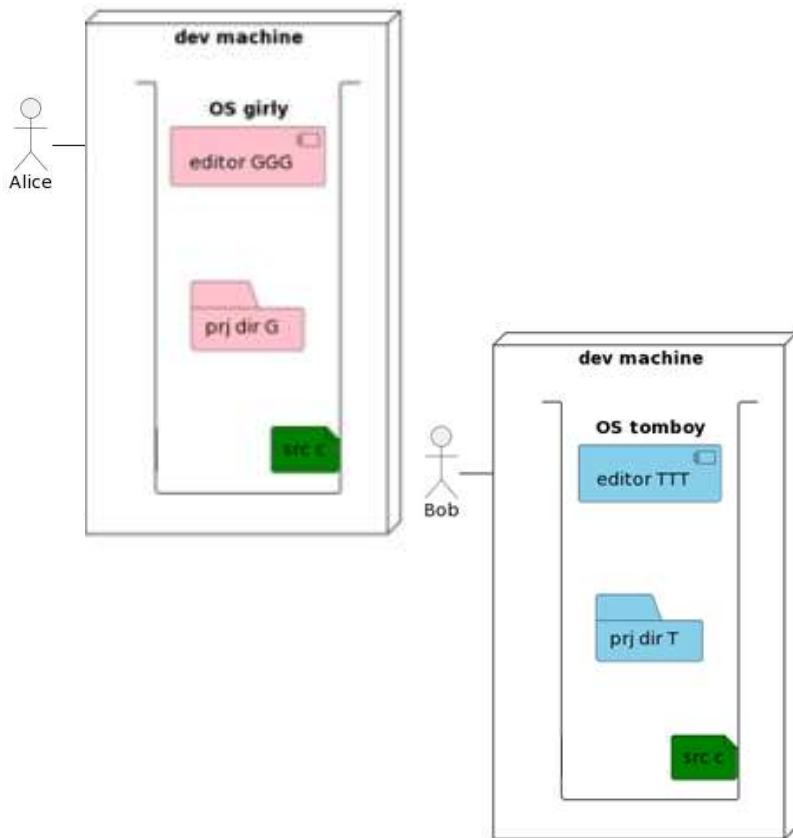


One dev : intro. To Dev. Machine

- Dev. Machine embeds
 - Directories with source code
 - Editor(s) of source code
 - Possibly build tools (or tasks runner(s))

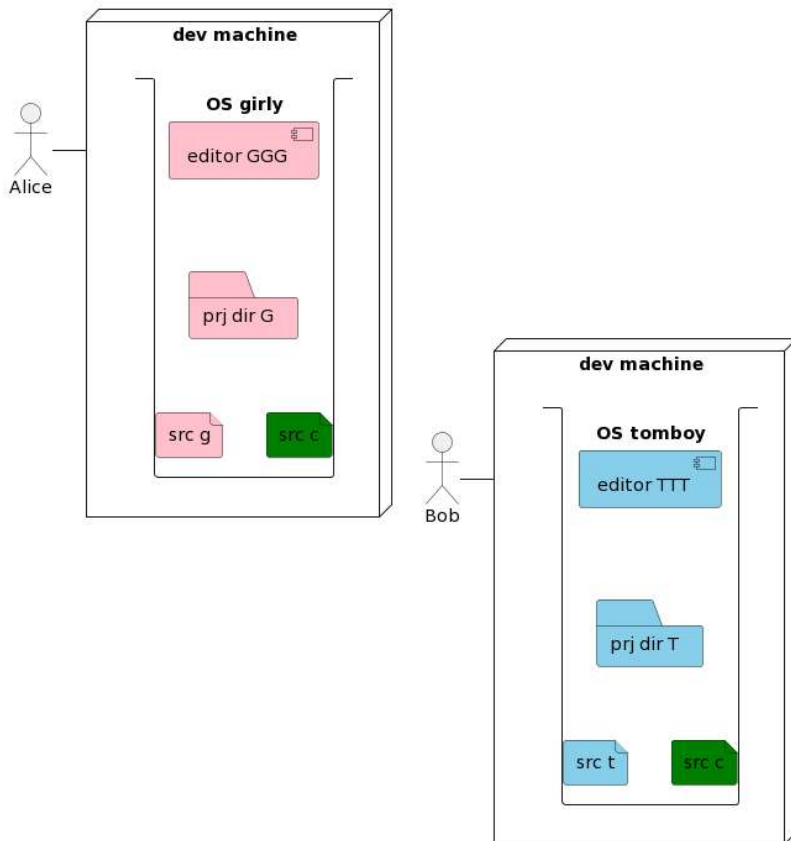


Second developer - Share for collaboration



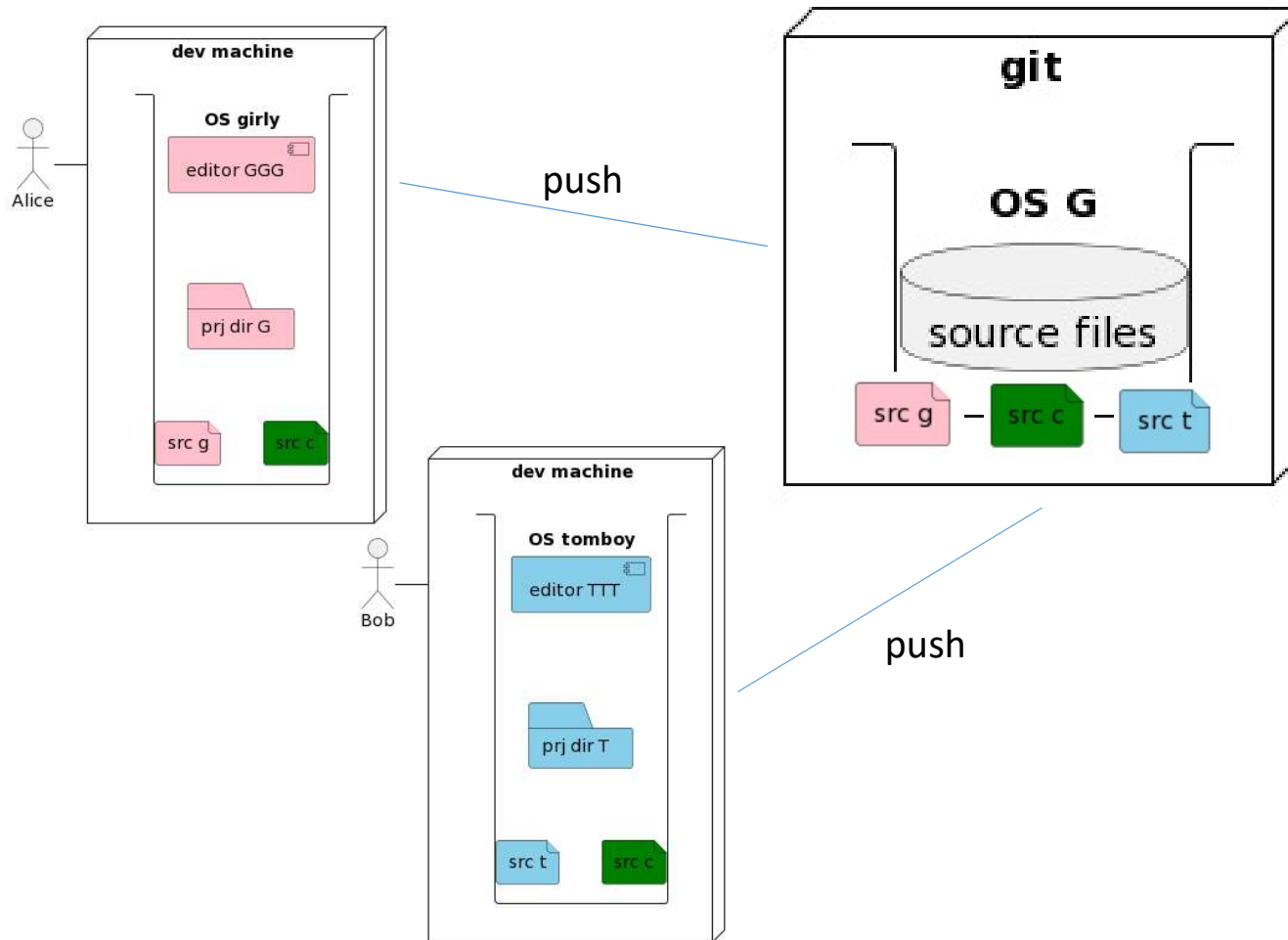
- A second dev. is joining (Bob)
 - He needs the existing file (src c)
 - He can copy it from Alice project's directory

Share for collaboration



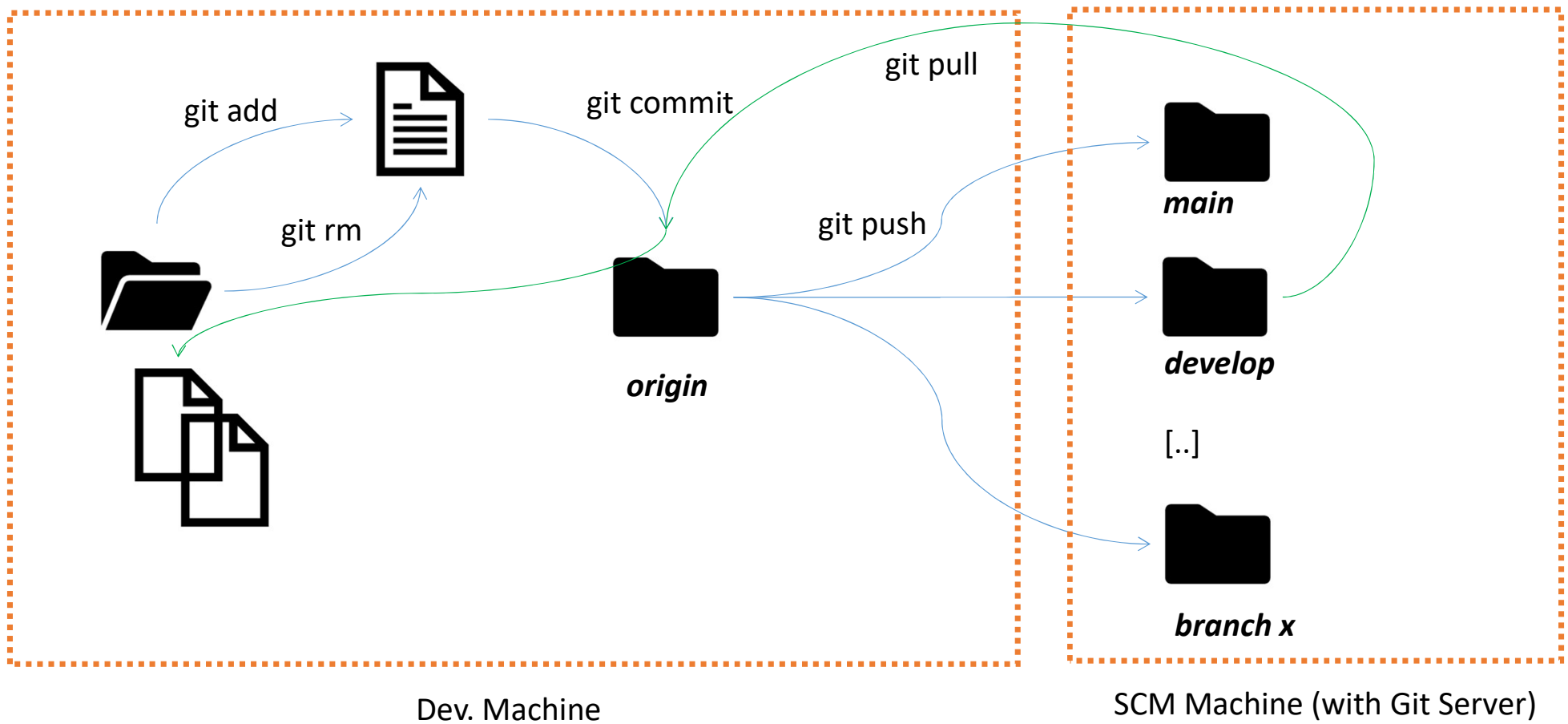
- A second dev. is joining (Bob)
 - He needs the existing file (src c)
 - He can copy it from Alice project's directory
- Life goes on
 - Alice creates src g
 - Bob creates src t
 - Maybe one of them change src c
- HOW TO SYNC !?

Share for collaboration

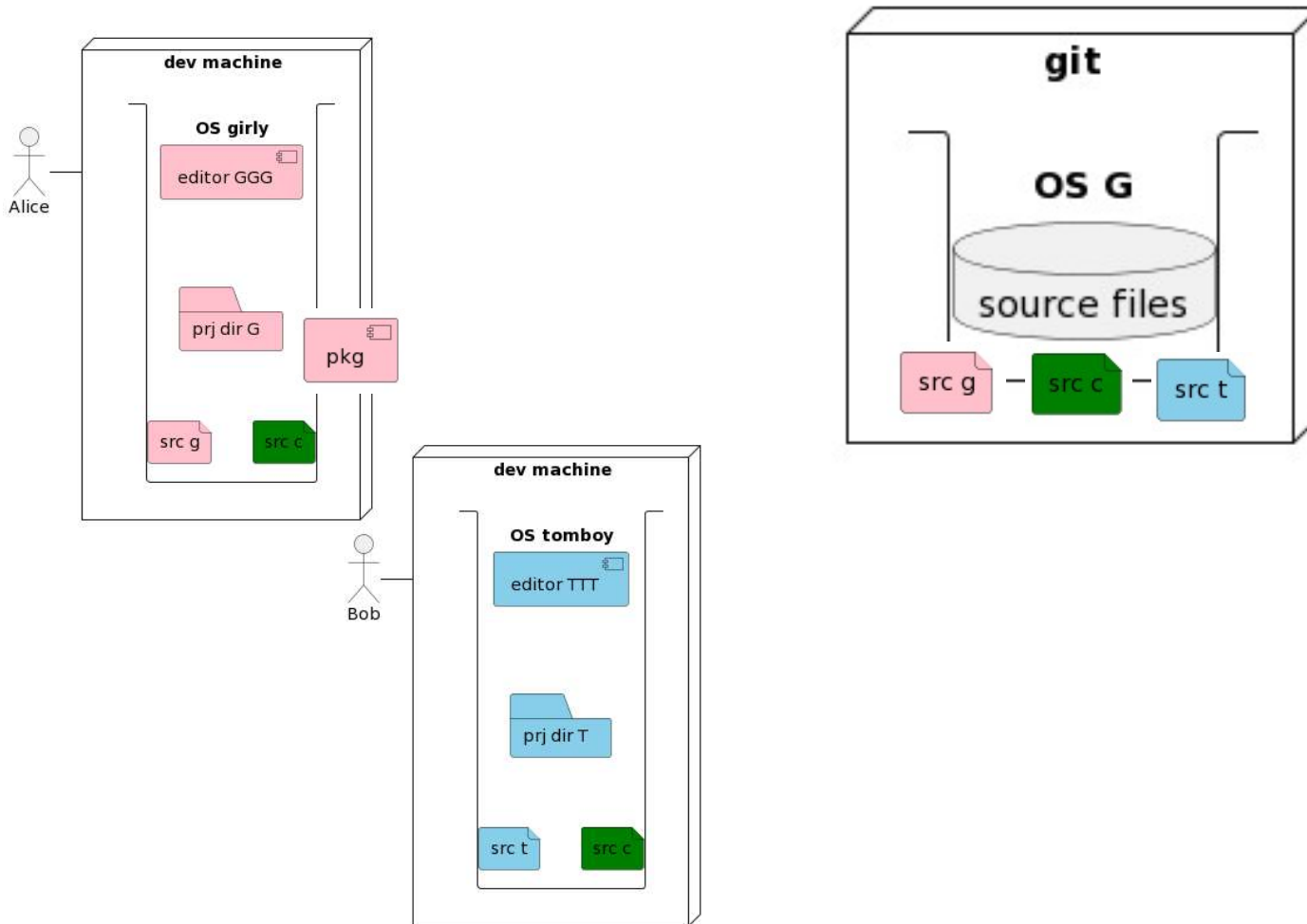


- Let SCM come into play
 - Source Code Manager
 - State of the art
 - Git
 - Older
 - SVN, CVS

Share – zoom on git

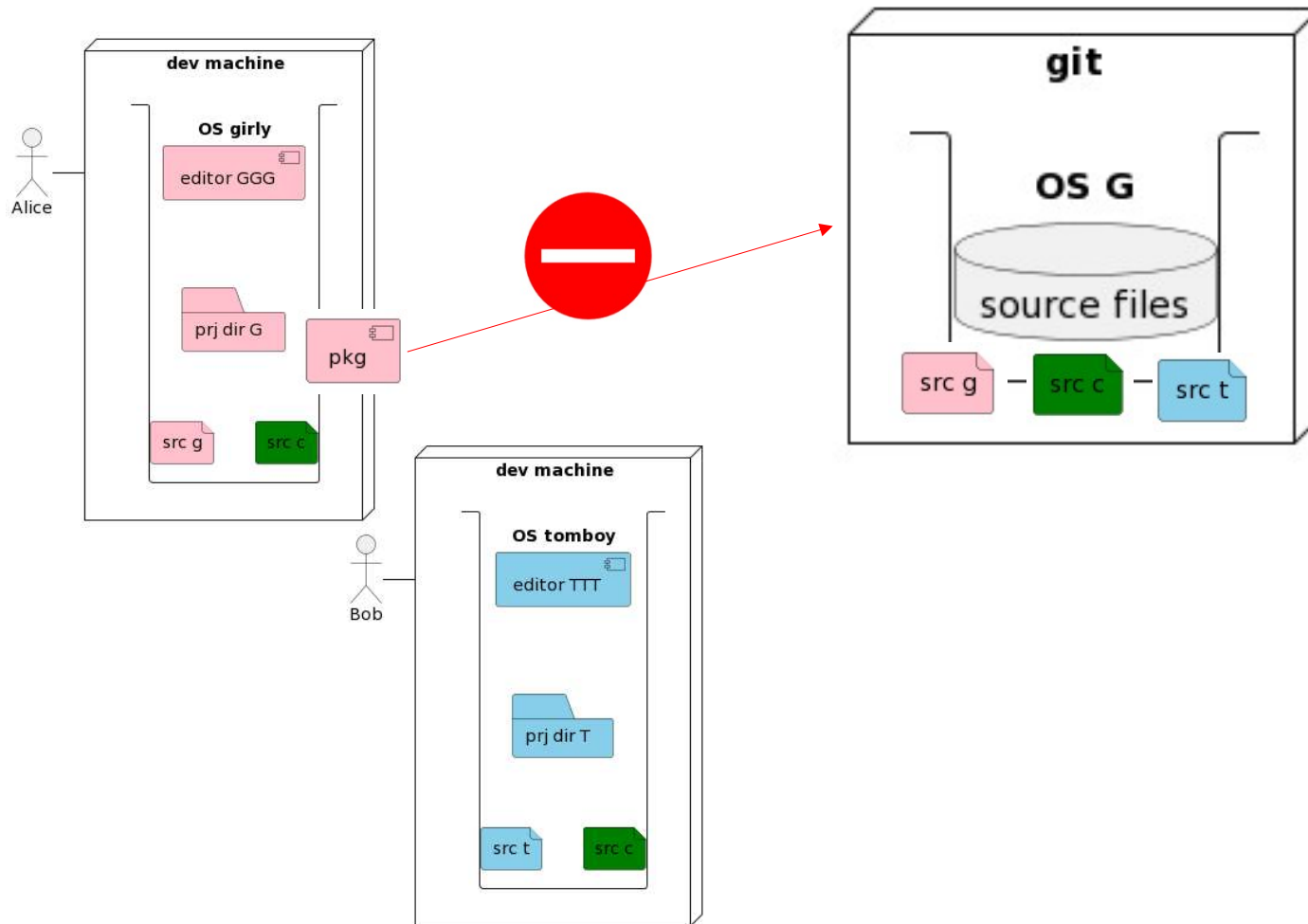


Share dependencies



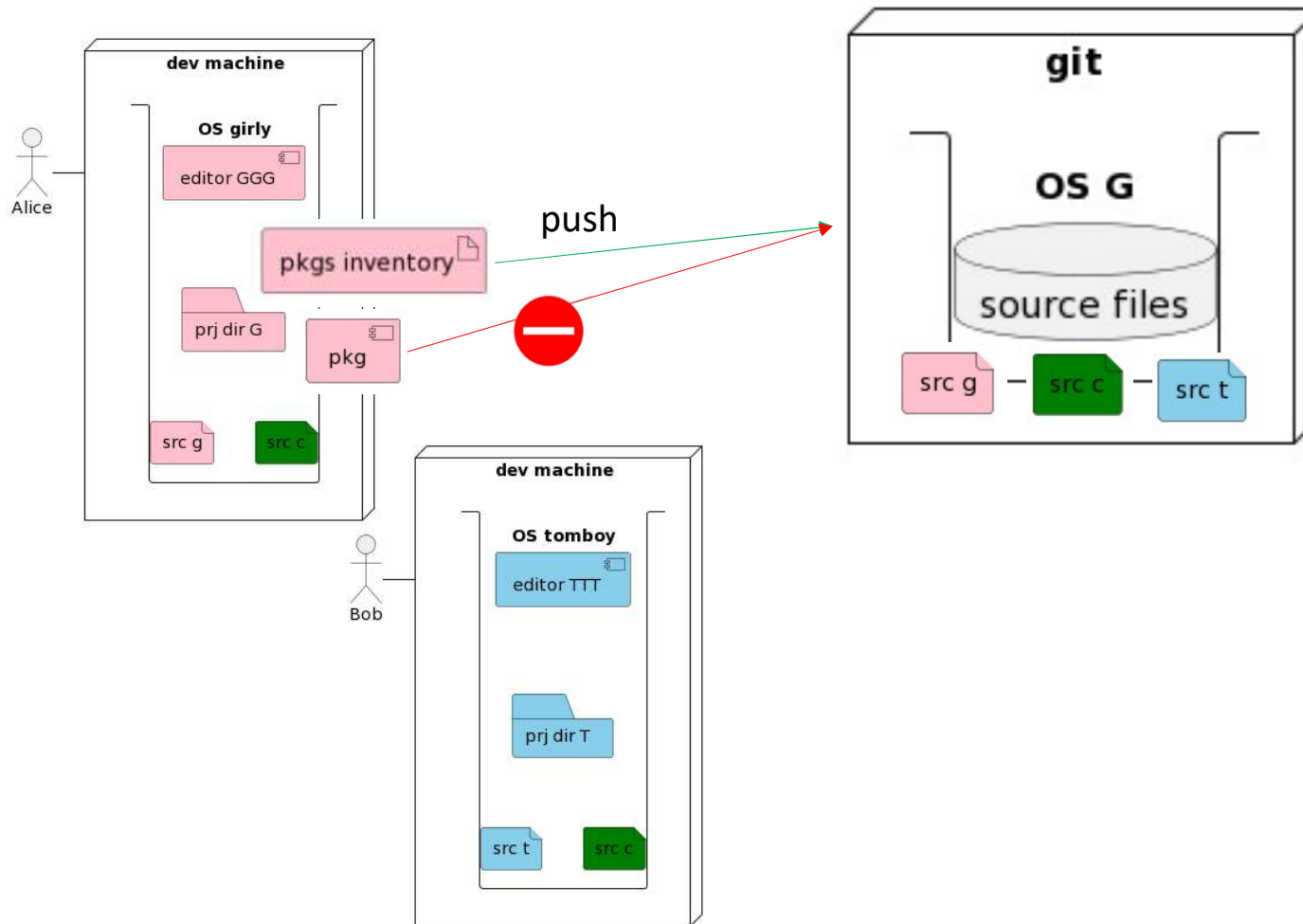
- Alice a new dependency (an external package : ie JavaFX libraries)

Share dependencies



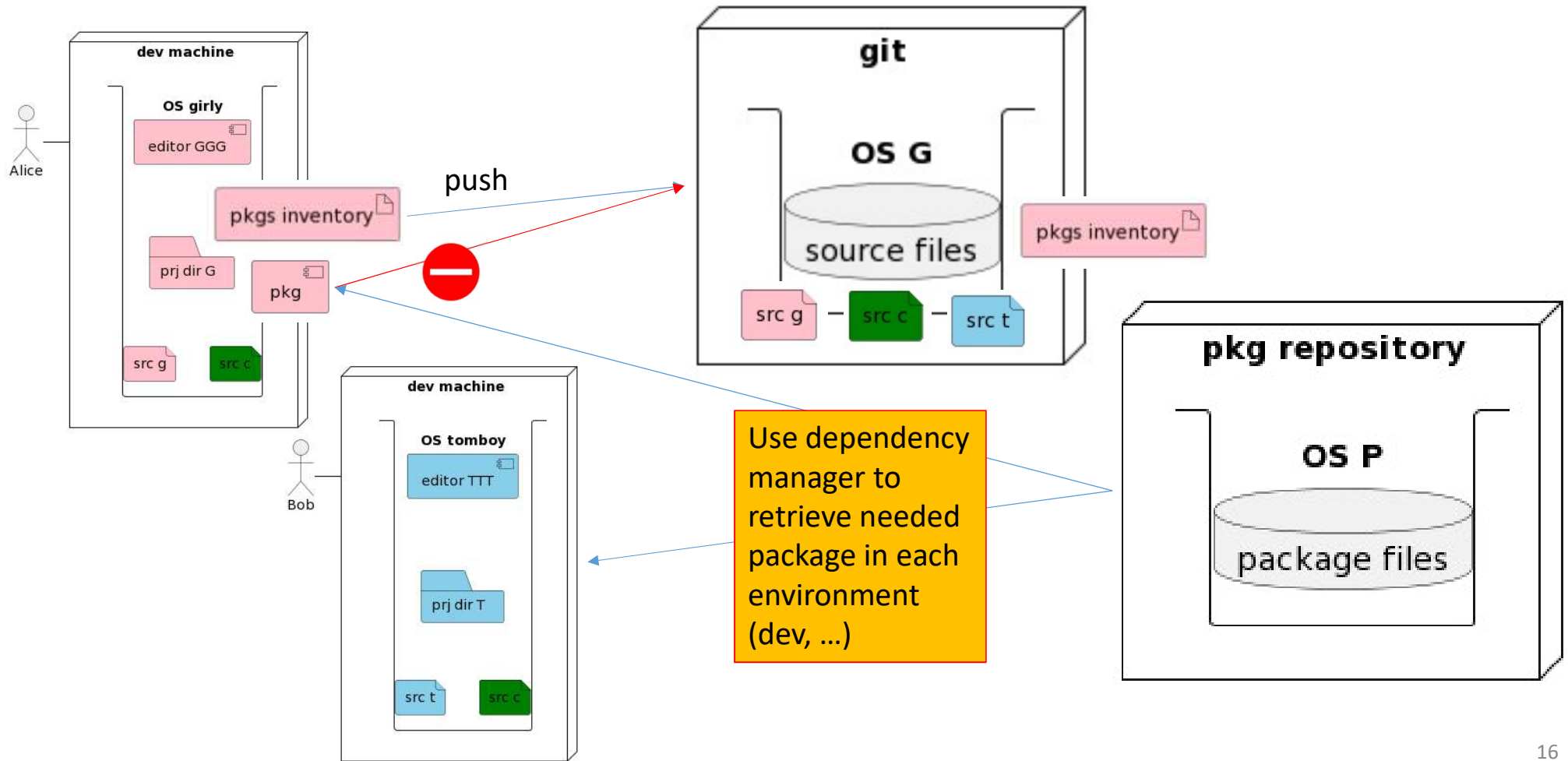
- Not in the scope of git that hosts only the content produced by the team
- NO

Share dependencies



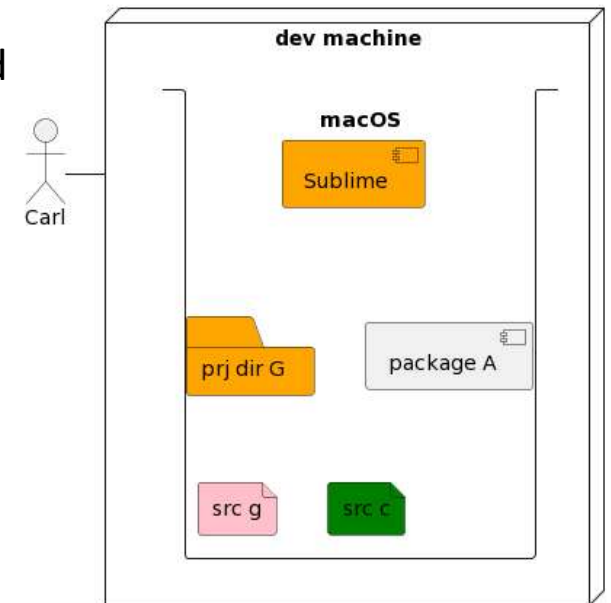
- But the list of package is a production of the team => do put it in git.

Share dependencies



Share dependencies – Python example

- When joining a dev. Project, two operations
 - Get the source code
 - Example command
 - > `git clone https://gitlab.insa-rouen.fr/delestre/mgpi-tp-cicd`
 - Get the dependencies
 - Example command
 - > `pip install -r requirements.txt`

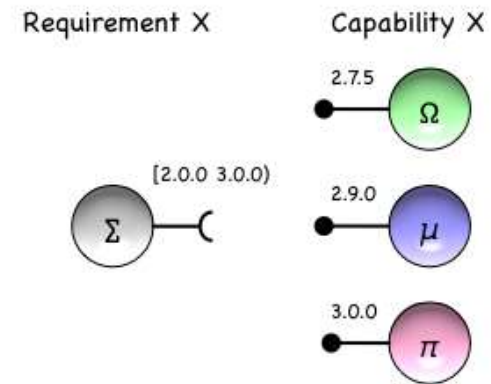
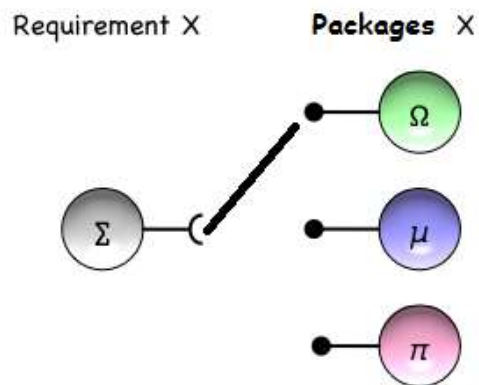


Share dependencies - Package Managers

	Tool	Dependencies file	Elements	
Java	maven	pom.xml	package	https://repo.maven.apache.org/maven2/
JavaScript (NodeJS)	npm	package.json	package	https://registry.npmjs.org/
Ruby	bundle	.Gemspec	gem	https://rubygems.org/
Python	pip	requirements.txt	module, package	

Share dependencies - semver

MAJOR.MINOR.PATCH
BREAKING.FEATURE.FIX



1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards-compatible manner, and
3. PATCH version when you make backwards-compatible bug fixes.

Image-credit : <https://www.osgi.org/developer/modularity/>

Share dependencies – semver – EXA7124B



<http://prodageo.insa-rouen.fr/casi4publi/fboodle/EXA7124B.enonce.pdf>

Share dependencies – prod vs dev

- Some dependencies are for machines **running** the deliverables
 - Utilities packages
 - Frameworks
 - Application Servers
- Others are for machines **building** the deliverables
 - Tools
 - Linters : **static code** analysis tool used to flag programming errors, bugs, stylistic errors and suspicious constructs.
 - Source generators : generate code based on a higher-level language specification (REST server from Swagger spec, ...)

Share dependencies – lint example

Explicit dependency

```
"pylint": {  
  "hashes": [  
    "sha256:ea82cd6ale11062dc86d555d07c021b0fb65afe39becbe6fe692efd6c4a67443",  
    "sha256:ec4a87c33da054ab86a6c79afa6771dc8765cb5631620053e727fcf3ef8cbcd7"  
  ],  
  "index": "pypi",  
  "version": "==2.15.8"  
},
```

Dev. Console

```
$ pylint --rcfile=.pylintrc huffman  
huffman/compteur.py:15:22: C0201: Consider iterating the dictionary  
directly instead of calling .keys() (consider-iterating-dictionary)
```

Source code not compliant

```
def incremter(self, element):  
    """ méthode qui incrémente le nom d'un élément """  
    if element in self._stat.keys():  
        self._stat[element] = self._stat[element] + 1  
huffman/compteur.py 15,22 17
```

consider-iterating-dictionary / C0201

📄 ©

Pylint doc

Message emitted:

Consider iterating the dictionary directly instead of calling `.keys()`

Description:

Emitted when the keys of a dictionary are iterated through the ``.keys()``` method or when ``.keys()``` is used for a membership check. It is enough to iterate through the dictionary itself, ```for key in dictionary```. For membership checks, ```if key in dictionary``` is faster.

Problematic code:

```
FRUITS = {"apple": 1, "pear": 5, "peach": 10}  
  
for fruit in FRUITS.keys(): # [consider-iterating-dictionary]  
    print(fruit)
```

Correct code:

```
FRUITS = {"apple": 1, "pear": 5, "peach": 10}  
  
for fruit in FRUITS:  
    print(fruit)
```

Source : <https://pylint.pycqa.org/>

Task Runners

	Tool	Default sc
Java	maven	pom.xml
JavaScript (NodeJS)	grunt	Gruntfile.js
Ruby	rake	Rakefile
Python		

arey Bump to jib-maven-plugin v3.3.1 for Java 17 docker base image 3b1d0fe last week 653 commits

.github/workflows	Change master branch	2 months ago
.mvn/wrapper	Upgrade Maven wrapper	3 years ago
src	Readability improvement	2 months ago
.editorconfig	#96 change EditorConfig in order to impact other files than Java and ...	7 years ago
.gitignore	ignore .vscode	2 months ago
LICENSE.txt	Create LICENSE.txt	2 years ago
mvnw	Upgrade Maven wrapper to 3.5.4	3 years ago
mvnw.cmd	Upgrade Maven wrapper to 3.5.4	3 years ago
pom.xml	Bump to jib-maven-plugin v3.3.1 for Java 17 docker base image	last week
readme.md	Configure CSS generation from SCSS with the libsass-maven-plugi	2 months ago

Java project

markgoodyear Initial ac6af0c on Jul 30, 2013 1 commit

css	Initial	9 years ago
js	Initial	9 years ago
.editorconfig	Initial	9 years ago
.gitignore	Initial	9 years ago
.jshintrc	Initial	9 years ago
Gruntfile.js	Initial	9 years ago
README.md	Initial	9 years ago
index.html	Initial	9 years ago
package.json	Initial	9 years ago

Javascript project

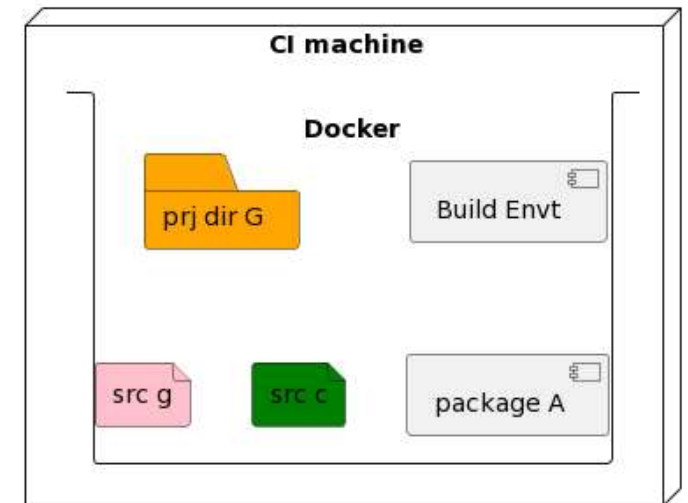
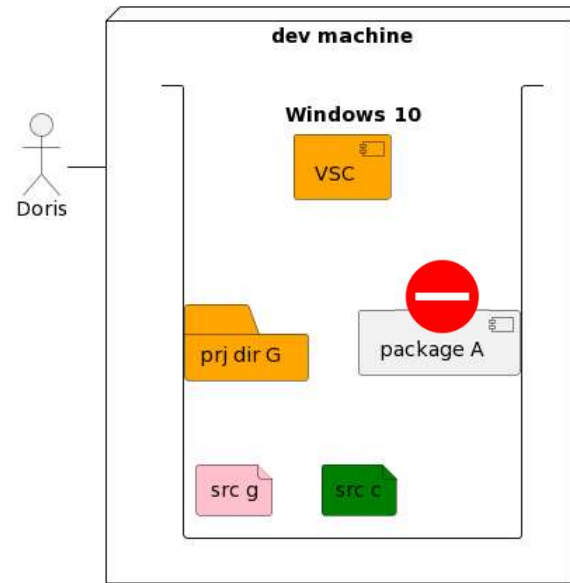
Ilcleandro Update SearchHelper.rb ...

config	commit project	
features	Update SearchHelper.rb	
Gemfile	update path rubygems	4 years ago
Gemfile.lock	update path rubygems	4 years ago
README.md	first commit	4 years ago
Rakefile	commit project	4 years ago

Help in :
 - Running tests
 - Init. DB

Remote build

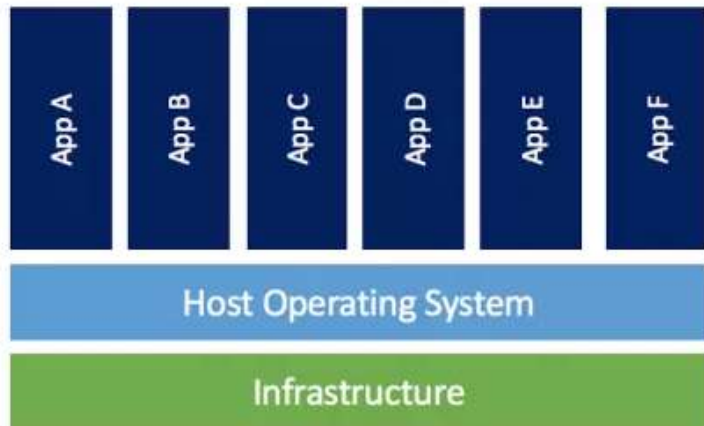
- Doris is joining the team with a Windows OS.
- Some of the dev dependencies are not available on this OS !



Virtualisation

Piece of knowledge to understand the software factory line

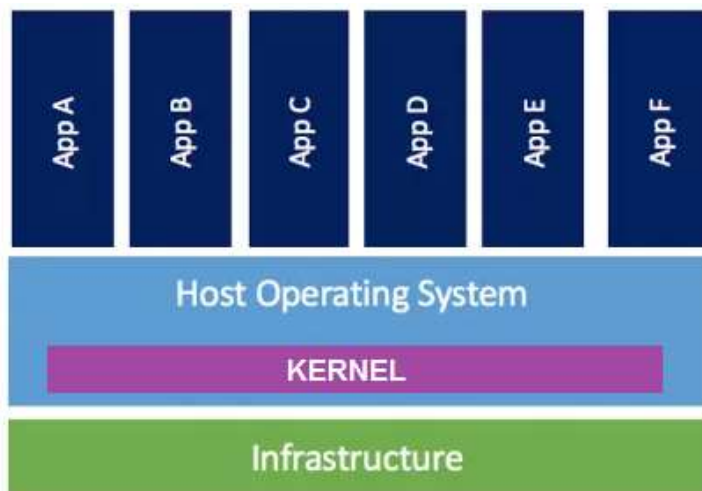
Without virtualisation



- Applications are setup in a « Host Operating System » that gives access to resources
 - File System
 - Network
 - ...
- Host Operating System adapt a physical infrastructure.

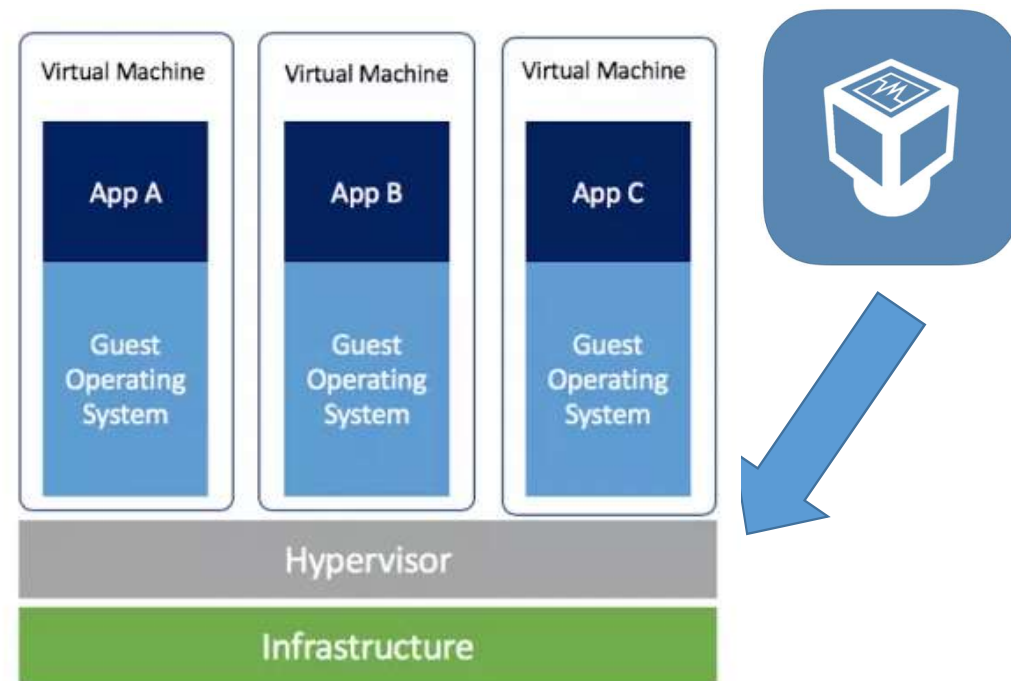
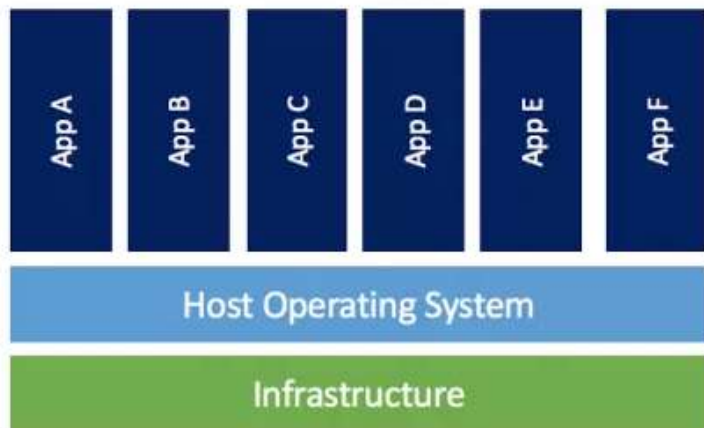
Kernel concept

- The kernel is the important program in the operating system

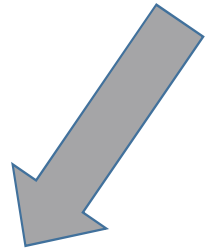
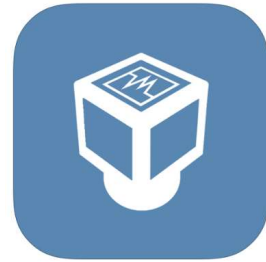
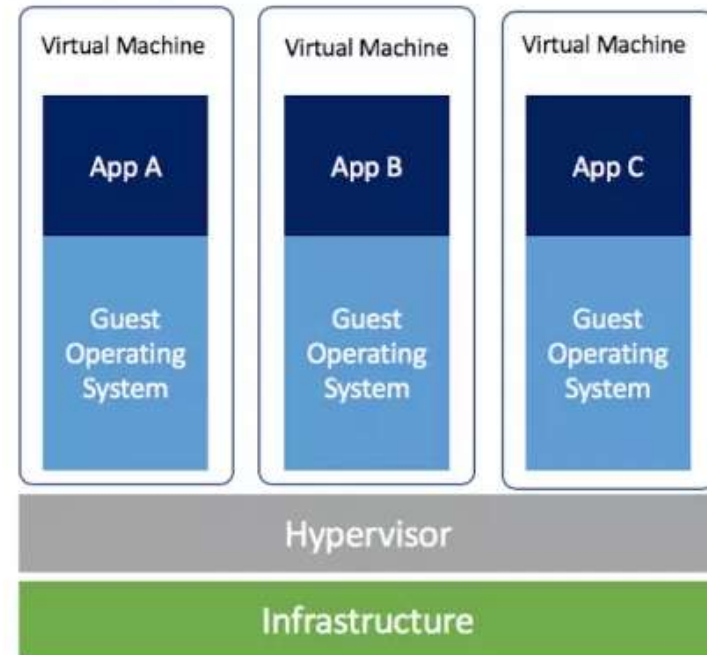
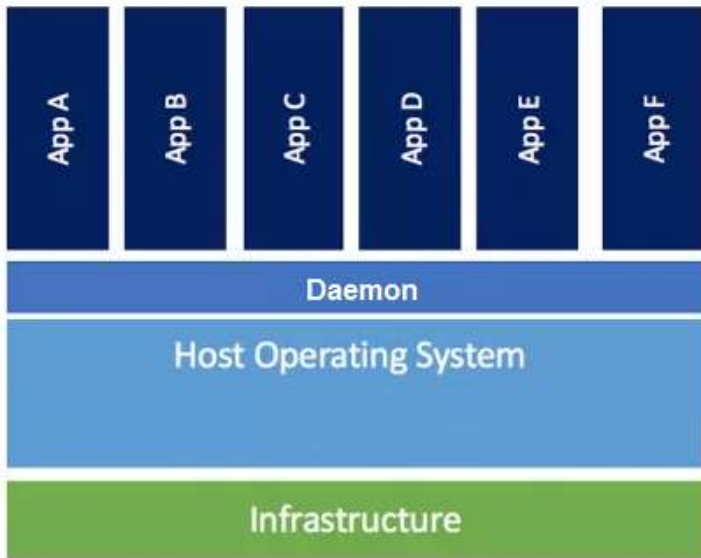
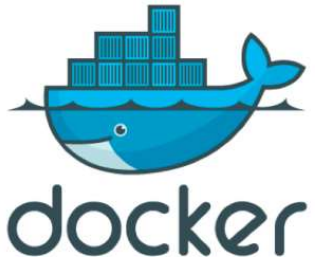


BASIS FOR COMPARISON	KERNEL	OPERATING SYSTEM
Basic	Kernel is an important part of the operating system.	Operating System is a system program.
Interface	Kernel is an interface between software and hardware of the computer.	Operating System is an interface between user and hardware of the computer.

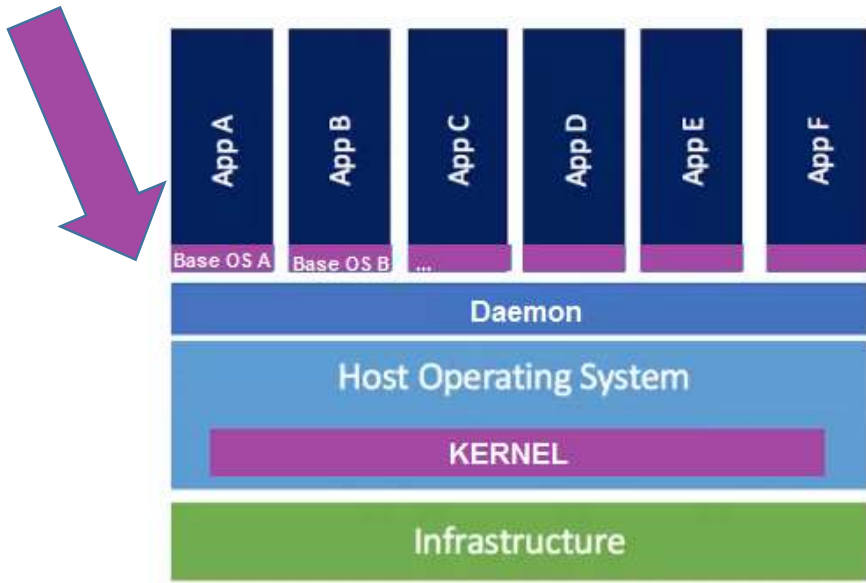
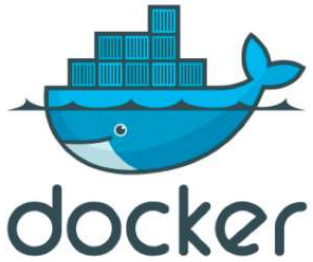
Physical Machine vs Hypervisor



Containers versus Hypervisor

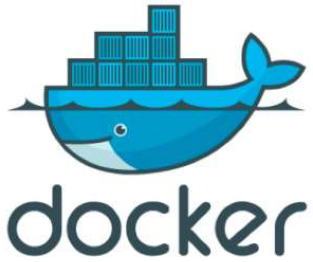


Containers : « Base OS », « kernel »

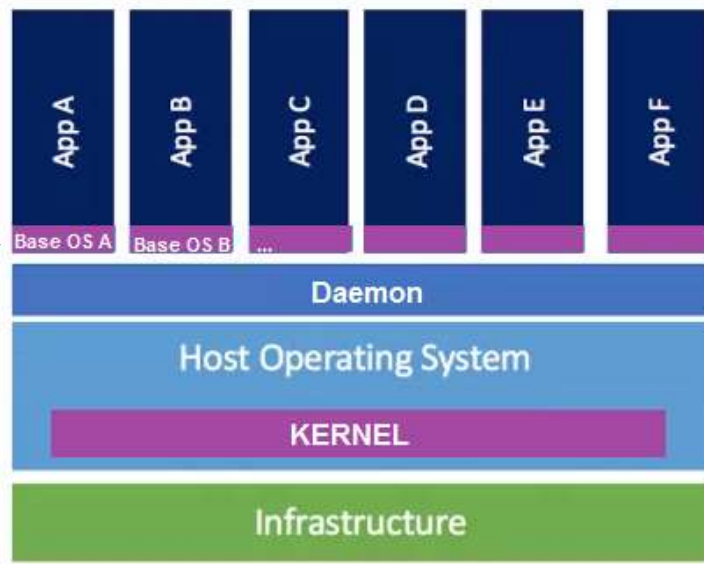


- Container embeds
 - One or several application(s)
 - A *Base OS*
- « *Base OS* » interact with « *kernel* » of « *Host Operating System* »
- Note : in specific cases, some apps are allowed to interact directly with kernel (by-passing « *Base OS* »).

Containers : « daemon »



- « daemon » is in charge of managing container
 - Starting
 - Stopping
 - ...

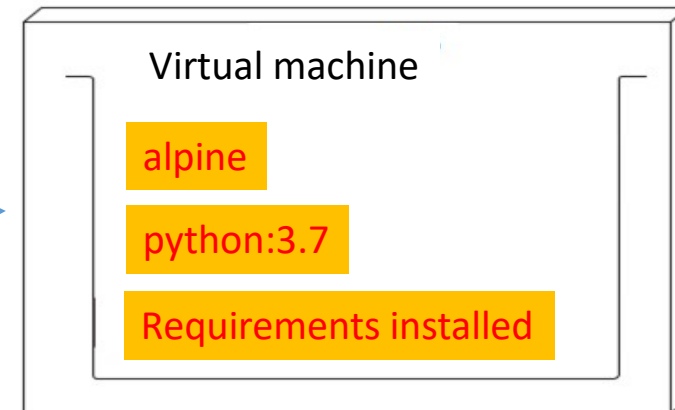
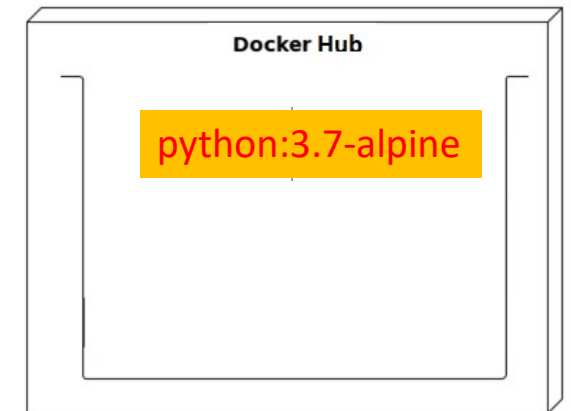


Dockerfile : installation fully scripted

Dockerfile

```
# syntax=docker/dockerfile:1
FROM python:3.7-alpine
WORKDIR /code
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
COPY . .
CMD ["flask", "run"]
```

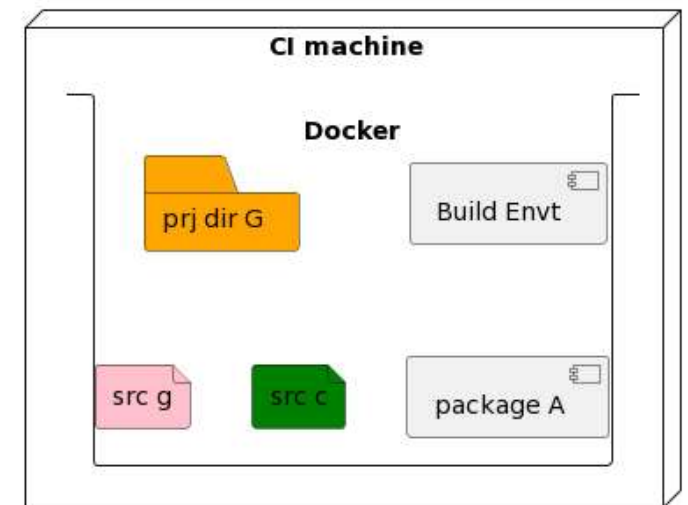
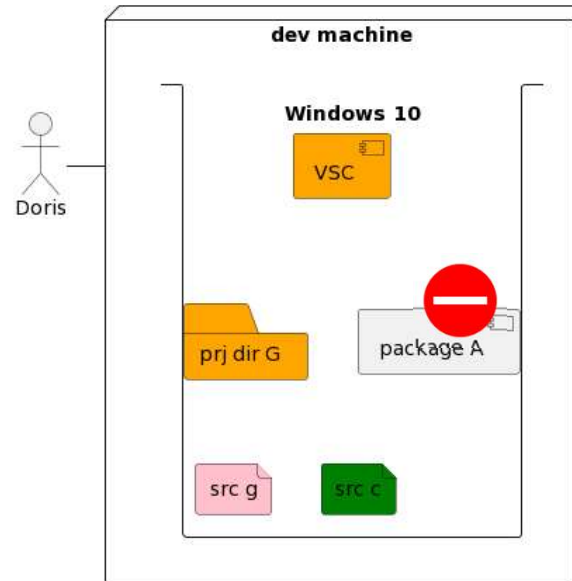
> docker compose up



Continuous Integration

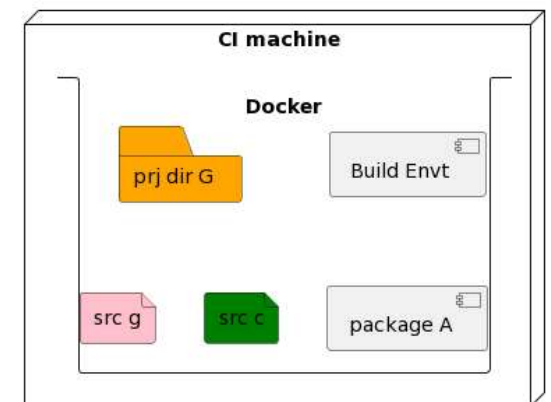
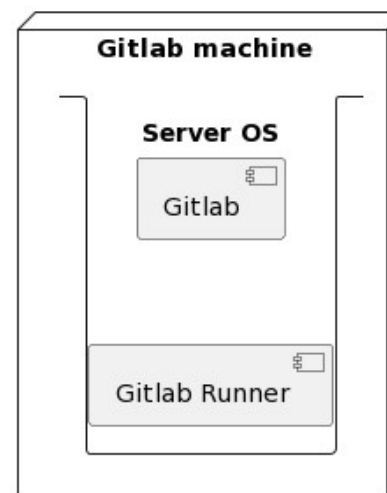
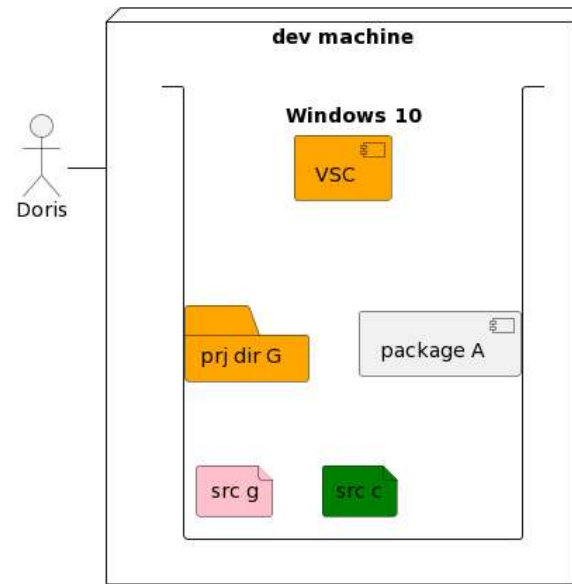
Remote build

- Doris is joining the team with a Windows OS.
- Some of the dev dependencies are not available on this OS !



Pipeline

- Based on an event, Gitlab Runner will trigger the **CREATION** of a virtual machine to run the job(s) of the CI platform !
- Event (CI_PIPELINE_SOURCE)
 - push
 - merge_request_event
 - scheduled



First pipeline : failed

```
Edit Visualize Validate NEW View merged YAML
Browse templates Help
1 default:
2   image: python:3.10.5
3
4   before_script:
5     - pip install -r requirements.txt
6
7   pylint:
8     script:
9     - pylint --rcfile=.pylintrc huffman
```

failed Update .gitlab-ci.yml file
#18414 main 03e049a9
00:00:28
3 minutes ago

```
compteur' inherits from object, can be safely removed from bases in python3 (useless-object-inheritance)
er iterating the dictionary directly instead of calling .keys() (consider-iterating-dictionary)
er iterating the dictionary directly instead of calling .keys() (consider-iterating-dictionary)
89 huffman/compteur.py:40:0: C0206: Consider iterating with .items() (consider-using-dict-items)
90 huffman/compteur.py:44:54: W0108: Lambda may not be necessary (unnecessary-lambda)
91 huffman/compteur.py:48:54: W0108: Lambda may not be necessary (unnecessary-lambda)
92 huffman/compteur.py:53:29: R1718: Consider using a set comprehension (consider-using-set-comprehension)
93 huffman/compteur.py:55:16: R1718: Consider using a set comprehension (consider-using-set-comprehension)
94 huffman/compteur.py:59:15: C0209: Formatting a regular string which could be a f-string (consider-using-f-string)
95 -----
96 Your code has been rated at 7.63/10
98 Cleaning up project directory and file based variables
100 ERROR: Job failed: exit code 1
```

First pipeline : success

The image shows the GitLab CI/CD interface. On the left, the `.gitlab-ci.yml` configuration is displayed in a code editor. The configuration includes a `default` section with `image: python:3.10.5` and a `before_script` section with `- pip install -r requirements.txt`. The `pylint` job is defined with a `script` section containing `- pylint --rcfile=.pylintrc --fail-under=7 huffman`. The `--fail-under=7` part is highlighted with a red box. Below the code editor, a sidebar menu is visible with `CI/CD` selected and highlighted with a red box. The main area shows a table of pipeline runs. The top row shows a successful pipeline run with the status `passed`, duration `00:00:19`, and trigger `just now`. The pipeline name is `Update .gitlab-ci.yml file` with ID `#18415` on the `main` branch. The bottom row shows a failed pipeline run with the status `failed`, duration `00:00:28`, and trigger `8 minutes ago`. The pipeline name is `Update .gitlab-ci.yml file` with ID `#18414` on the `main` branch.

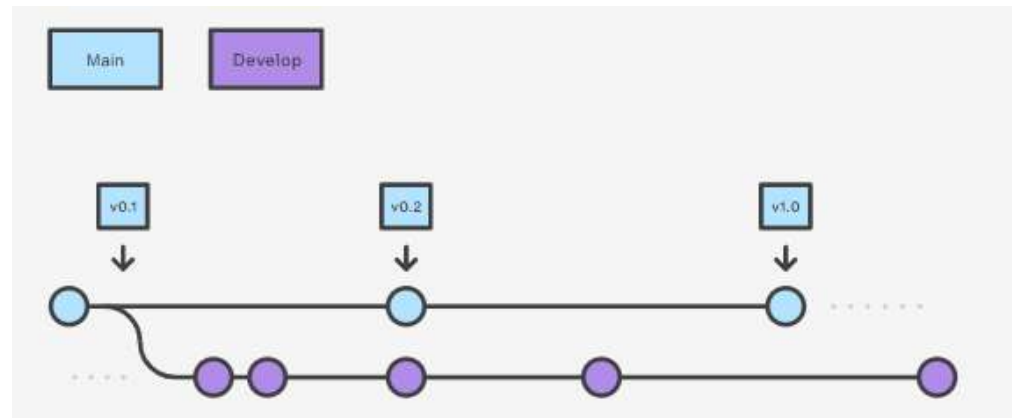
```
1 default:
2   image: python:3.10.5
3
4   before_script:
5     - pip install -r requirements.txt
6
7   pylint:
8     script:
9       - pylint --rcfile=.pylintrc --fail-under=7 huffman
10
```

Status	Pipeline	Triggerer	Stages
passed 00:00:19 just now	Update .gitlab-ci.yml file #18415 main fbac4036 latest		
failed 00:00:28 8 minutes ago	Update .gitlab-ci.yml file #18414 main 03e049a9		

- Issues 0
- Merge requests 0
- CI/CD**
- Pipelines
- Editor
- Jobs
- Schedules
- Test Cases

Some gouvernance on the pipeline

- With 4 persons working on the project, Alice wants to have control on the source code introduced in the **main** branch
 - => she set a new branch : **develop**
- By default, a repository is created with **one branch** named **main**
 - Note : **master** has been eradicated



Merge Request versus Push

- Push
 - The developer **decides** to add the source code to a branch
- Merge Request
 - Other designated people (Alice) **do decide**
 - Possible to insertion additional control operation

Gitlab – restrict access to a branch

- Settings
 - General
 - Integrations
 - Webhooks
 - Access Tokens
 - Repository**
 - Merge requests
 - CI/CD
 - Packages and registries
 - Pages
 - Monitor
 - Usage Quotas

Branch	Allowed to merge	Allowed to push	Allowed to force push ?	Code owner approval ?	
main default	Maintainers	No one	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Unprotect

```
C:\Apps\laragon\www\cicd (main)
λ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 6 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 555 bytes | 555.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Gitlab: You are not allowed to push code to protected branches on this project.
To https://gitlab.insa-rouen.fr/prodageo/cicd.git
 | [remote rejected] main -> main (pre-receive hook declined)
error: failed to push some refs to 'https://gitlab.insa-rouen.fr/prodageo/cicd.git'
```

Sync with the develop branch

- Bob can continue to work
- He switched to the develop branch
- And now, push is possible.

```
C:\Apps\laragon\www\cicd (main)
λ git checkout develop
Switched to a new branch 'develop'
Branch 'develop' set up to track remote branch 'develop' from 'origin'.

C:\Apps\laragon\www\cicd (develop)
λ git push origin develop
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 307 bytes | 307.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for develop, visit:
remote:   https://gitlab.insa-rouen.fr/prodageo/cicd/-/merge_requests/new?merge_request[source_branch]=develop
remote:
To https://gitlab.insa-rouen.fr/prodageo/cicd.git
   fbac403..855db7d  develop -> develop
```


Gitlab – approval permissions rules

- Settings
- General
- Integrations
- Webhooks
- Access Tokens
- Repository
- Merge requests**
- CI/CD
- Packages and registries
- Pages
- Monitor
- Usage Quotas

Merge request approvals

Define approval rules and settings to ensure [separation of duties](#) for new merge requests. [Learn more.](#)

Approval rules

Name	Approvers	Target branch	Approvals required	
All eligible users ?		All branches	<input type="text" value="0"/>	
Product Owner	 Alice	main	<input type="text" value="1"/>	Edit Delete
License-Check ?				Enable
Requires approval for Denied licenses. Learn more.				
Coverage-Check ?				Enable
Requires approval for decreases in test coverage. Learn more.				
Add approval rule				

Security Approvals

Create more robust vulnerability rules and apply them to all your projects. [Learn more](#)

Name	Target branch	Approvals required
You don't have any security policies yet		
Create security policy		

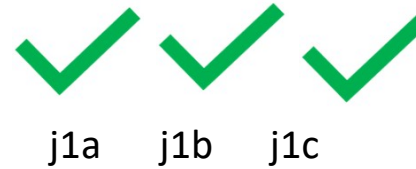
Gitlab – operations specific to Merge

```
  Browse templates  Help
1  default:
2  |   image: python:3.10.5
3  |
4  |   before_script:
5  |   | - pip install -r requirements.txt
6  |
7  |   pylint:
8  |   |   script:
9  |   |   | - pylint --rcfile=.pylintrc --fail-under=7 huffman
10 |
11 |   job1:
12 |   |   script:
13 |   |   | - echo "This job runs in merge request pipelines"
14 |   |   | rules:
15 |   |   | - if: $CI_PIPELINE_SOURCE == 'merge_request_event'
```

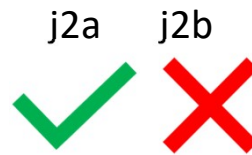

Pipelines - Stages

```
stages:  
- tests  
- static analysis  
- deploy
```

Stage tests



Stage static analysis



Stage deploy

j3a j3b j3c

Pipelines - Gitlab Pages

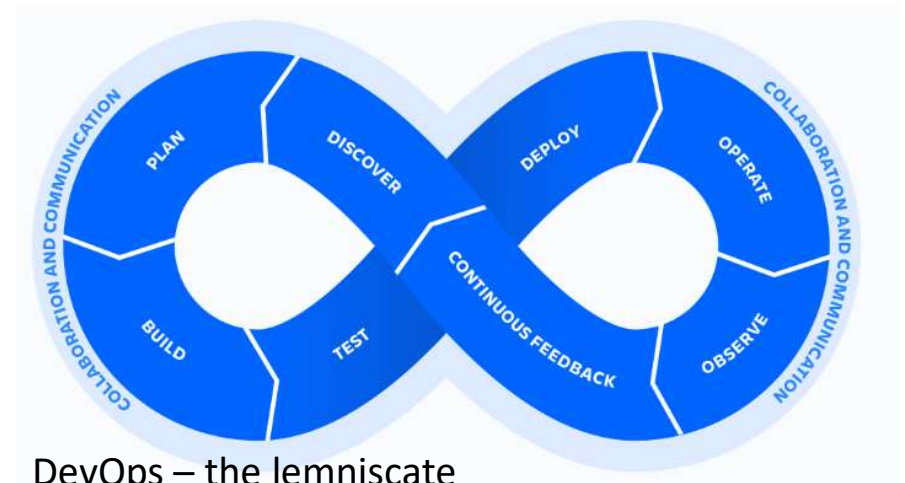
```
image: ruby:2.7

pages:
  script:
    - gem install bundler
    - bundle install
    - bundle exec jekyll build -d public
  artifacts:
    paths:
      - public
```

- A special job (pages:deploy) is triggered after the pages job.
- Every files (HTML, ...) generated within **public** directory are published on the URL indicated in Settings > Pages
 - <https://account.pages.insa-rouen.fr/reponame>
 - **account** : name of the person or group
 - **Reponame** : name of the repository (project)

Conclusion : Continuous Everything and DevOps

- Continuous
 - Integration : unit test, lint, ...
 - Prepare and test the build
 - Delivery
 - Package the build in artefacts (deliverables like files, ...)
 - Deployment
 - Copy and install the artefacts of the target environment (be production, pre-production, demo, ...)



DevOps – the lemniscate

Source : <https://www.atlassian.com/devops>

