

# Deploying at scale with a single click



**Wim Godden**  
**Cu.be Solutions**

# Who am I ?

- Wim Godden (@wimgtr)

# Where I'm from



# Where I'm from



**Where I'm from**



**Where I'm from**





**Where I'm from**



**Where I'm from**





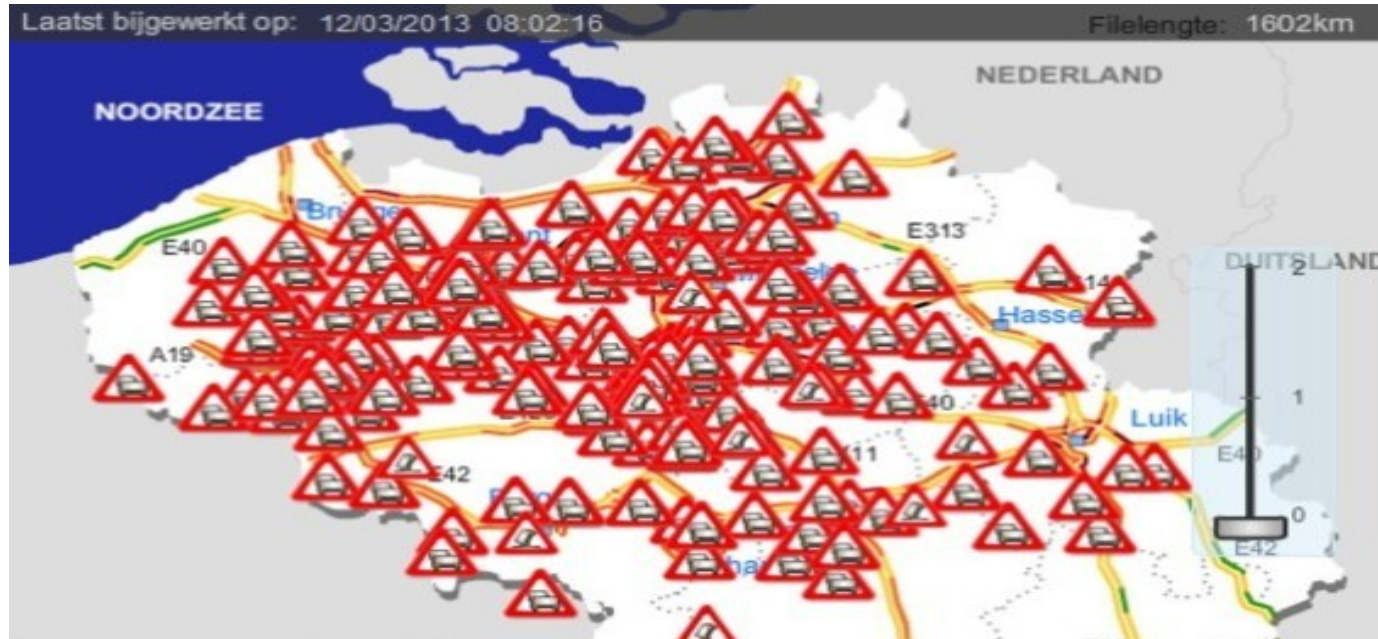
# My town



# My town



# Belgium – the traffic



# Who am I ?

- Wim Godden (@wimgtr)
- Founder of Cu.be Solutions (<https://cu.be>)
- Open Source developer since 1997
- Developer of PHPCompatibility, ...
- Speaker at PHP and Open Source conferences

# Who are you ?

- Developers ?
- Devops ?
- Project with many servers or external machines (cloud or native) ?



# Deploying at scale with a single click

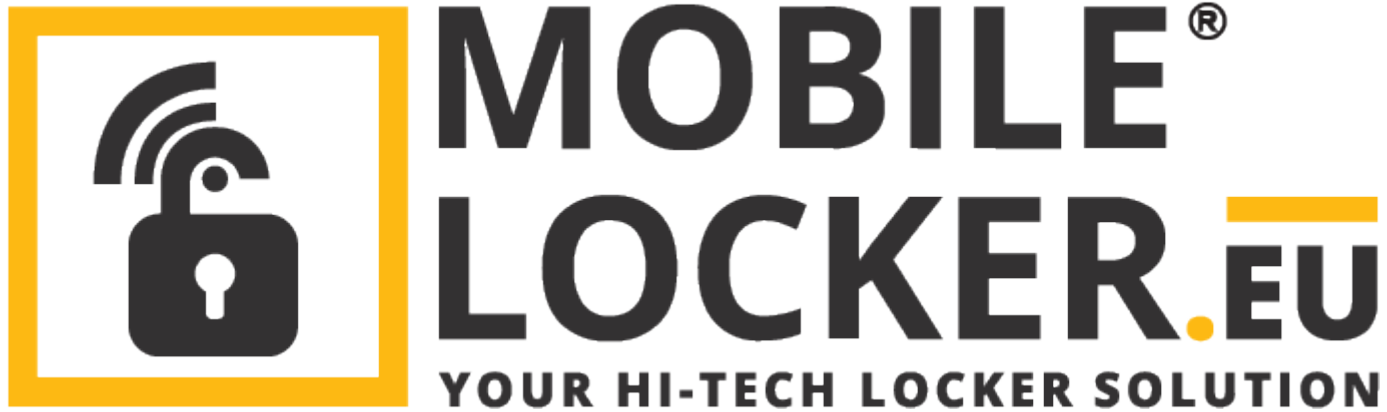
= Orchestration

- Deployment
- Configuration
- Development
- Management
- Coordination
- ...

# Short disclaimers

- Many strategies
  - Let's look at one
  - But add notes on alternatives along the way
- Works for different types of devices
  - Servers
  - Network of clients
  - IOT
  - Kiosks
  - Payment terminals
  - ...

**Our initial goal**



# Our initial goal





**These devices are in the field...**





## Previous software

- Windows-based
- Written in Delphi
- Manual setup
- Deployment method :  **TeamViewer**
- Tedious
- Doesn't scale...

# The challenge

- Rebuild the software (**not** in Delphi !)
- Make it scalable to 100 locker units over time

Meaning :

- Automated Linux installation
- Automated updating of :
  - Linux packages
  - Custom software
  - Docker containers
  - Configuration files
  - ...

# Installing linux

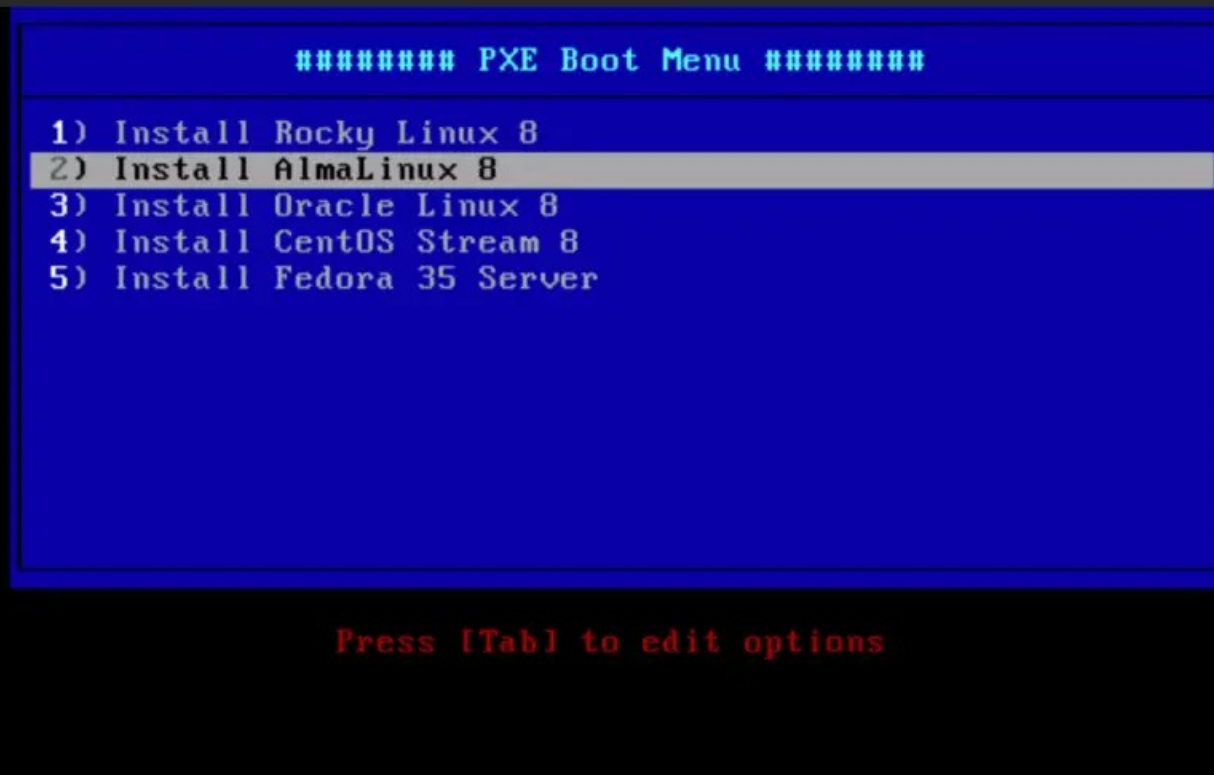
- Manual install (insert USB stick, select all options, ...)
- Kickstart via USB
  - Pre/post scripting
- Kickstart over network via PXE boot
- Cloud-Init

## Scaling to...

- 2017 : “Up to 100 over time”
- 2020 : Easily surpassed that...
- 2024 : “Should be ready for thousands”

# Scaling the installation

- Installing with USB stick
  - Fine for a few at a time
  - No-go for hundreds/thousands
- PXE boot to the rescue





# Scaling the installation

All Temporary Units ▾ [Options](#) - Edited

Name: All ▾

Last online at: later than 01 Dec 2024, 00:00 ✕ ▾

Page: < 1 > of 1 Total of 5 records

<input type="checkbox"/> ▾	NAME	STATUS	LAST RESPONSE TIME	LAST ONLINE AT	LAST UPDATED AT
<input type="checkbox"/>	temp-8PFSV5	installed	Offline	12 days ago	05 Feb 2025, 12:01
<input type="checkbox"/>	temp-QSPQZQ	installed	Offline	12 days ago	05 Feb 2025, 11:54
<input type="checkbox"/>	temp-LDALPS	installed	Offline	a month ago	09 Jan 2025, 17:35
<input type="checkbox"/>	temp-72D876	installed	Offline	2 months ago	30 Dec 2024, 04:01
<input type="checkbox"/>	temp-LF7563	installed	Offline	2 months ago	27 Dec 2024, 04:02

# Scaling the installation

- How do you link the device with the name ?
  - Do you have a screen → display it
  - No screen → network MAC address → port on managed switch



**Next step : orche... software installation & configuration**

## **Next step : software installation & configuration**

- Ansible (Alternatives : Chef, Puppet, Salt, Terraform, ...)
- No agent software on target machine
- Straightforward
- Connects to target over SSH
  - All you need is public key installed
  - Could be done in kickstart installation

# Ansible high-level

- Playbook (install-software.yml / update-release.yml / ...)
  - Roles (packages, vpn, users, api, ...)
    - Tasks

Example tasks :

```
- name: Install docker
  ansible.builtin.package:
    name:
      - docker-ce
      - docker-ce-cli
      - containerd.io
      - docker-compose-plugin
    state: present
```

```
- name: Create docker directory
  ansible.builtin.file:
    path: /etc/docker
    state: directory
    recurse: true
```

```
- name: Add the nginx configs
  ansible.builtin.template:
    src: nginx.conf
    dest: /etc/nginx/nginx.conf
```

Modules :

```
- name: Create a proxy network
  community.docker.docker_network:
    name: proxy
```



# Playbook example

```
---  
- name: Webserver install  
  hosts: webserver_group  
  roles:  
    - { role: hostname }  
    - { role: user }  
    - { role: ssh }  
    - { role: sysctl }  
    - { role: packages }  
    - { role: postfix, when: "'mailcatcher' not in group_names" }  
    - { role: firewall, tags: firewall }  
    - { role: nginx }  
    - { role: ntp }  
    - { role: snmp }  
    - { role: backup, tags: backup }  
  
- import_playbook: install-mailcatcher.yml
```

## What does a run look like ?

```
2025-02-01 23:00:43,956 | PLAY [software_update] *****
2025-02-01 23:00:43,986 | TASK [Gathering Facts] *****
2025-02-01 23:00:43,987 | Monday 1 February 2025 23:00:43 +0000 (0:00:00.032) 0:00:00.032 *****
2025-02-01 23:00:48,819 | ok: [DemoMachine]
2025-02-01 23:00:48,826 | TASK [Check ALL hosts are reachable before doing the release] *****
2025-02-01 23:00:48,871 | ok: [DemoMachine] => changed=false
2025-02-01 23:01:25,470 | TASK [applications/api-config : Api config] *****
2025-02-01 23:01:30,495 | changed: [DemoMachine]
2025-02-01 23:01:30,501 | TASK [applications/api : Run composer install] *****
2025-02-01 23:02:28,024 | changed: [DemoMachine]
2025-02-01 23:02:28,027 | TASK [applications/api : Warmup the cache] *****
2025-02-01 23:02:32,088 | changed: [DemoMachine]
2025-02-01 23:02:51,661 | TASK [applications/healthcheck : Run healthcheck] *****
2025-02-01 23:02:55,735 | changed: [DemoMachine]
2025-02-01 23:03:31,894 | TASK [applications/sync-system-info : Sync system info with the OMS] *****
2025-02-01 23:03:36,543 | changed: [DemoMachine]
2025-02-01 23:03:55,201 | PLAY RECAP *****
2025-02-01 23:03:55,201 | DemoMachine : ok=7 changed=5 unreachable=0 failed=0 skipped=0
rescued=0 ignored=0
2025-02-01 23:03:55,201 | Monday 1 February 2025 23:02:55 +0000 (0:00:01.304) 0:05:11.247 *****
```

## **What if a task times out or fails along the way ?**

- Try to make everything as atomic as possible
- So just run it again
- Ansible will skip tasks that don't need action

# Where do you run it ?

Units / DEMO-P001 3 ms

- Mark EOL
- Config
- Edit
- Push users
- Deploy new software
- Update system packages
- Go to admin

Created At: 01 Mar 2024, 15:11 | Updated At: 18 Feb 2025, 10:03 | Active event:

Business unit: Mobile locker | Change History

- Unit
- Software version
- Payment terminals
- Placements
- Event log

Name

DEMO-P001

Display Name

DEMO-P001

Disabled

No

Last Response Time

3 ms

Last Online At

a few seconds ago

Last Config Pushed

2 days ago

Last Users Config Pushed

never

Last Users Failed Config Push Reason

Default Locker Units

DEMO-P001

Internal IP

VPN IP

Screen Orientation

horizontal

Devices

Barcode reader front

- Rotation: 180

UPS

Software version

Software Version

11.23.0 (Latest: 11.25.0 ) Compare

Commit

e736d0c1163998ad34c371f0549535b44af42b8c

Commit Date

December 10, 2024 18:07 ( 2 months ago )

Last Updated Software


a month ago

Last Updated System Packages

19 days ago

# Where do you run it ?

 Deploy new software

 Update system packages

## Software version

Software Version [11.23.0](#) (Latest: [11.25.0](#) ) [Compare](#)

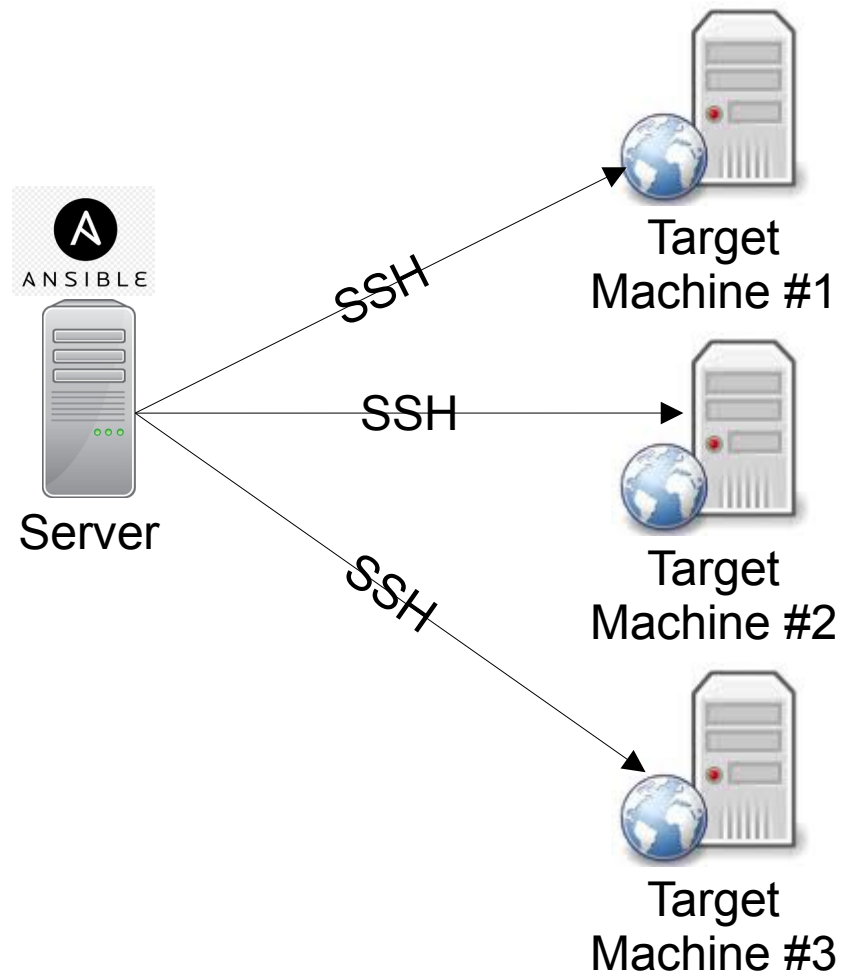
Commit [e736d0c1163998ad34c371f0549535b44af42b8c](#)

Commit Date December 10, 2024 18:07 ( 2 months ago )

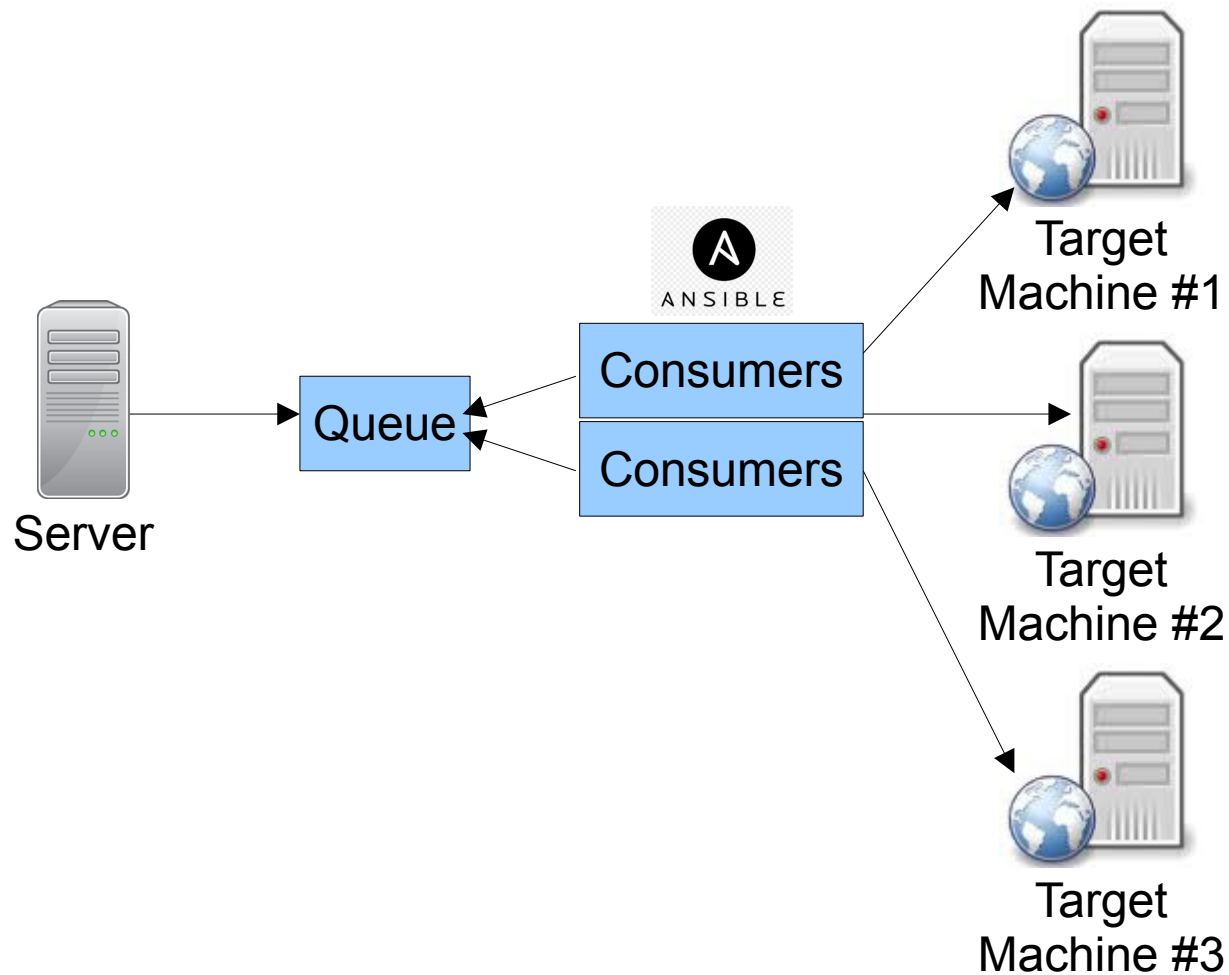
Last Updated Software a month ago

Last Updated System Packages 19 days ago

# Where do you run it ?

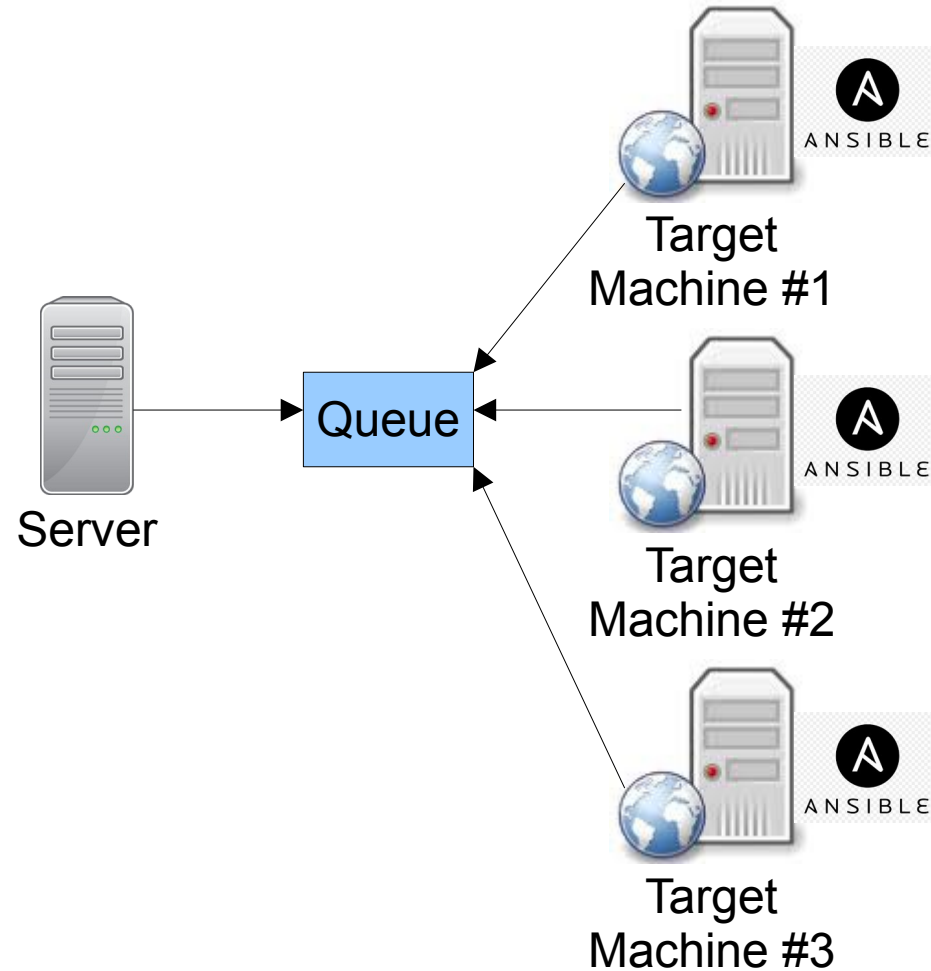


# Where do you run it ?





# Where do you run it ?



# Grouping hosts

- Ansible has inventory folder
- Contains
  - Hosts to connect to (target machines)
  - Variables
    - Global
      - or
    - Host-specific

# Variables

## ■ In tasks/main.yml :

- name: Verify that the necessary packages are installed  
  ansible.builtin.package:  
    name: "{{ packages | join(',') }}"  
    state: present

## ■ In vars/main.yml :

- packages:
- bash-completion
  - git
  - wget
  - net-tools

# Variables

## ■ In users/vars/main.yml

user\_create:

- ajohnson
- bjackson

user\_deleted\_users:

- smcleod
- pgabriel

## ■ Iteration over a list in users/task/main.yml

- name: Remove old users

ansible.builtin.user:

state: absent

name: "{{ item }}"

remove: true

with\_items: "{{ user\_deleted\_users }}"

# Variables - replication

- name: Start replication.  
community.mysql.mysql\_replication:  
  mode: startreplica  
  login\_user: root  
  login\_password: "{{ db\_root\_password }}"  
  login\_host: 127.0.0.1  
  fail\_on\_error: true  
when: "'cluster\_secondary' in group\_names"

# Parallel processing

```
2025-02-01 23:00:43,956 | PLAY [remotemachine] *****
2025-02-01 23:00:43,986 | TASK [Gathering Facts] *****
2025-02-01 23:00:43,987 | Monday 1 February 2025 23:00:43 +0000 (0:00:00.032) 0:00:00.032 ****
2025-02-01 23:00:44,819 | ok: [DemoMachine1]
2025-02-01 23:00:45,219 | ok: [DemoMachine2]
2025-02-01 23:00:48,819 | ok: [DemoMachine3]
2025-02-01 23:00:48,826 | TASK [Check ALL hosts are reachable before doing the release] *****
2025-02-01 23:00:48,871 | ok: [DemoMachine1] => changed=false
2025-02-01 23:00:48,891 | ok: [DemoMachine2] => changed=false
2025-02-01 23:00:48,899 | ok: [DemoMachine3] => changed=false
2025-02-01 23:01:25,470 | TASK [applications/api-config : Api config] *****
2025-02-01 23:01:30,475 | changed: [DemoMachine1]
2025-02-01 23:01:30,477 | changed: [DemoMachine2]
2025-02-01 23:01:30,495 | changed: [DemoMachine3]
2025-02-01 23:01:30,501 | TASK [applications/api : Run composer install] *****
2025-02-01 23:02:28,024 | changed: [DemoMachine1]
2025-02-01 23:03:52,024 | changed: [DemoMachine2]
2025-02-01 23:08:51,024 | changed: [DemoMachine3]
2025-02-01 23:08:51,027 | TASK [applications/api : Warmup the cache] *****
```

# Folder structure

```
production          # Inventory file for production servers
staging             # Inventory file for staging servers
group_vars/         # Variables for specific groups
    group1.yml
    group2.yml
host_vars/          # Variables for specific targets
    hostname1.yml
    hostname2.yml
roles/
    common/          # This is a "role"
        tasks/
            main.yml  # The main task file
    nginx/
        tasks/
            main.yml
        templates/
            nginx.conf.j2
someplaybook.yml     # A playbook file describing which roles to run
```



## But I don't have hundreds of devices, just servers...

- That's what tools like Ansible were originally built for :
  - Deploying in a consistent way on every server
  - Deploying to test...
  - ... then staging ...
  - ... then production
  - in the exact same way
- If a server crashes
  - Reinstall Linux
  - Run Ansible
    - Which should contain your configuration files
- If you want to scale up
  - Add another host to the group in the inventory
  - Run Ansible

# Logging and monitoring

- Final task : call the central platform and report in
- If it's a software update : report the version
- Small networks can grow quickly

# Questions ?



**Questions ?**

## Contact

- Mastodon [@wimg@phpc.social](https://wimg@phpc.social)
- Slides <http://www.slideshare.net/wimg>
- E-mail [wim@cu.be](mailto:wim@cu.be)

Thanks !