ICA0002: IT Infrastructure Services
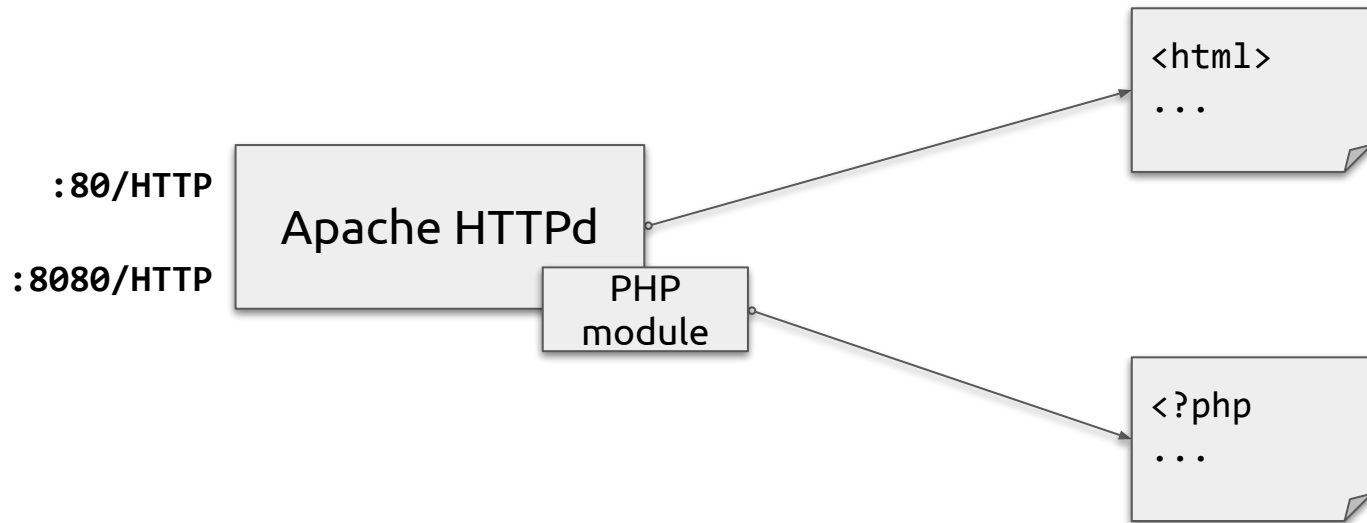
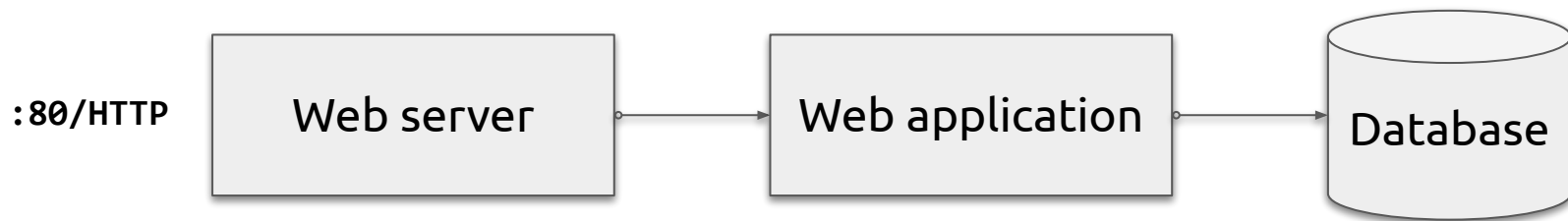# #4
Web applications
Ansible variables

Roman Kuchin
Juri Hudolejev
2019

Previous Lab

Next Lab

:80/HTTP   Web server → Web application → Database

WordPress

Started as a simple blogging software in 2003

Most widely used content management system (CMS) today:
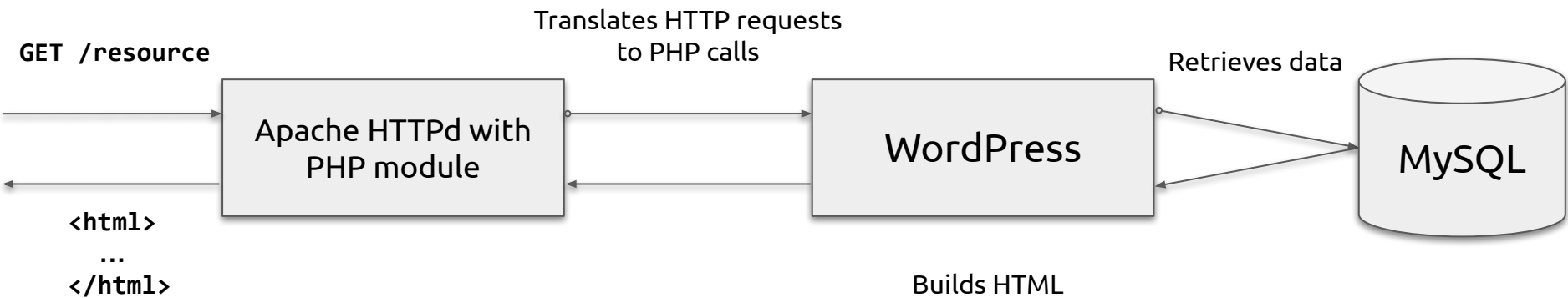
    ~34% of all websites, ~61% of all CMS

PHP application, requires web server and database (MySQL)

Modular architecture: API and lots of plugins available

More info: https://wordpress.org

How WordPress Works

```
GET /resource
```

Apache HTTPd with
PHP module

Translates HTTP requests
to PHP calls

WordPress

Retrieves data

MySQL

```
<html>
...
</html>
```

Builds HTML

Note: this is just one of possible setups...

MySQL

Relational database management system (RDBMS)

Multiple editions (including open-source community edition)

Used in Google, Facebook, Wikipedia, Twitter, Ebay, PayPal, Netflix, GitHub etc.

Notable forks: MariaDB, Percona

More info: https://www.mysql.com/products/community

See also: https://mariadb.org, https://www.percona.com

General Database Topics

Not discussed in detail but somewhat covered in labs as needed:

- Access control, authentication, authorization
- Backups
- Non-relational databases

Not discussed in this course:

- Replication mechanisms
- Storage engines
- SQL

Questions?

Previous Labs

What we did before:

```
# playbook.yml

…
  - copy:
      src: some-file.conf
      dest: /path/to/file.conf
…
```

```
# ports.conf (local)

Listen 80
Listen 8080


...
```

```
# sites.conf (local)

<VirtualHost *:8080>
  …
</VirtualHost>
```

```
# README.md

…
Use port 8080…
…
```

Problems

1.

What if we need to change the custom port?

How to ensure that __all__ needed files are changed?

Problems

1.

What if we need to change the custom port?

How to ensure that <u>all</u> needed files are changed?

2.

How to make this code support similar but <u>different</u> installations?

Server A: port 8080, server B: port 8180 etc.

Ansible Variables

```
# playbook.yml

...
  - template:
      src: some-file.conf
      dest: /path/to/file.conf
...
```

```
# ports.conf (local)

Listen 80
Listen {{ port }}


...
```

```
# sites.conf (local)

<VirtualHost *:{{ port }}>
  ...
</VirtualHost>
```

```
# README.md

...
Use port {{ port }}...
...
```

Ansible Variables in a Play

```
- name: My play
  hosts: ...

  vars:
    port: 8080

  tasks:
    - name: My task
      template:
        src: ...
        dest: ...
```

Ansible Variables in an Inventory File

```
...

[webservers]
despina    port=8080
proteus    port=8180

...
```

```
- name: My play
  hosts: ...

  tasks:
    - name: My task
      template:
        src: ...
        dest: ...
```

Ansible Variables

Can be defined in multiple places:

- Command line
- Inventory file
- Separate `group_vars` files
- Separate `host_vars` files
- Plays
- Tasks
- Etc.

More: docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html

Ansible Variables in Command Line

```
ansible all -m shell -a 'ps -aux' -u elvis
```

```
{{ ansible_user }} -> elvis
```

Ansible Variables in Command Line

`ansible-playbook webserver.yml -a `**`port=8080`**

`{{ port }} -> 8080`

Ansible Variables in Inventory File

```
[my_servers]

ubuntu   ansible_host=localhost ansible_port=2222 ...


         {{ ansible_host }} -> localhost

         {{ ansible_port }} -> 2222
```

Ansible Variables in a Play

```
tasks:

  - command: "netstat -tlnp"

    register: result

  - debug:

      var: result.stdout_lines
```

Ansible Variables

```
# my.conf (local)

x.user = {{ username }}
x.pass = {{ password }}
```

```
vars:
  username: elvis
  password: p4ssw0rd
```

```
# playbook.yml
...

- template:
    src: my.conf
    dest: /path/to/my.conf

...
```

```
# /path/to/my.conf

x.user = elvis
x.pass = p4ssw0rd
```
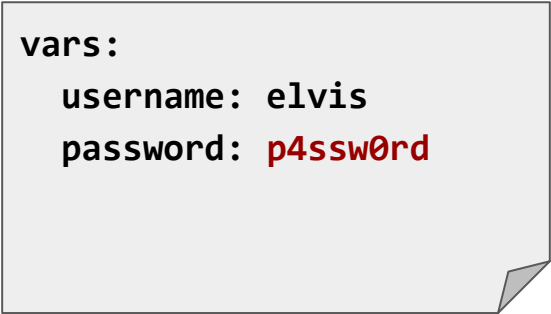
Problems

3.

We cannot store plain text secrets in Git!

```
vars:
  username: elvis
  password: p4ssw0rd
```

Ansible Vault

Command line tool to encrypt and decrypt sensitive data

Can encrypt files or separate variables

More: https://docs.ansible.com/ansible/latest/user_guide/vault.html

Ansible Vault: Encryption

```
$ ansible-vault encrypt_string p4ssw0rd

New Vault password:
Confirm New Vault password:

!vault |
    $ANSIBLE_VAULT;1.1;AES256
    66313066386566346263353134373763333466303863656133343232623162303562643366383562
    62363733646535343761646330636639373863393333765320a316538386363663733383939626333
    66343661343564353934373239316538346666656436303035623635363661636631393261353966
    31666132343136396406134313062356433333731646436666664373563303061663064313666662
    6161

Encryption successful
```

Ansible Vault: Decryption

```
vars:
  password: !vault |
    $ANSIBLE_VAULT;1.1;AES256
    66313066386566346263353134373763333466303863656133343232623162303562643366383562
    62363733646535343761646330636639373863393333765320a31653838636366373338393962633
    66343661343564353934373239316538346666656436303035623635363661363631393261353966
    31666132343136396640a61343130623564333337316464366666643735633030616306431366662
    6161

...

ansible-playbook my_playbook.yml --ask-vault-pass

...

  {{ password }} -> p4ssw0rd
```

Demo time!

Questions?