

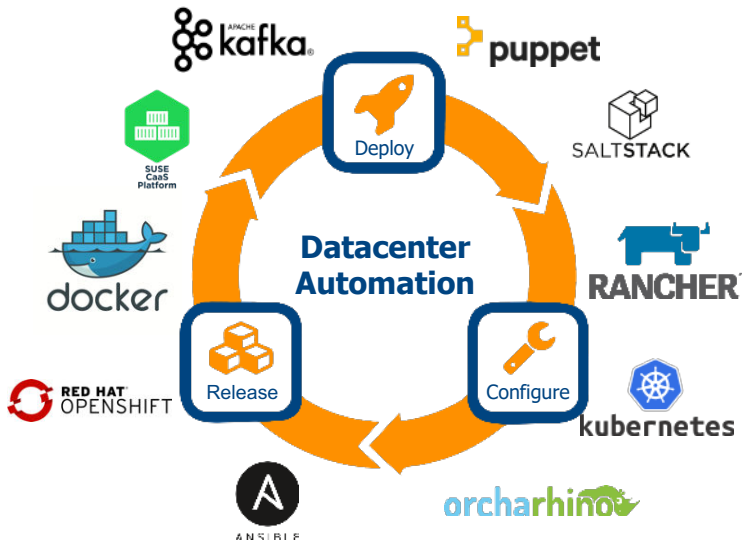
Workshops Environments

Automatic Configuration and Deployment



David Sardari

11.08.2019








- ▶ **ANSIBLE** by Red Hat (since 2015):
 - ▶ Release: 2012
 - ▶ Basis: Python



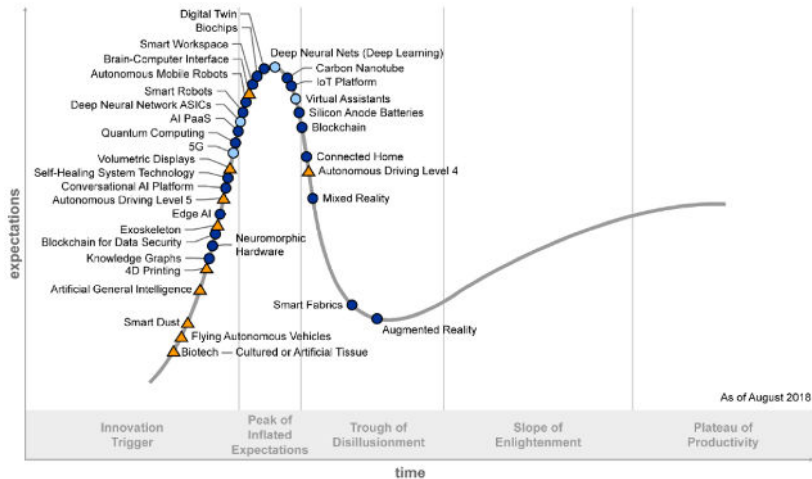
- ▶ **puppet** by puppetlabs:
 - ▶ Release: 2005
 - ▶ Basis: Ruby



- ▶ **SALTSTACK** by Saltstack:
 - ▶ Release: 2011
 - ▶ Basis: Python

- ▶  **docker** by Docker Inc.:
 - ▶ Release: 2013
 - ▶ SaaS and PaaS for OS-level virtualisation
- ▶  **kubernetes** by CNCF (since 2015):
 - ▶ Release: 2014
 - ▶ Automatic application deployment, scaling and management
- ▶  **orcharhino** by ATIX AG:
 - ▶ Release: 2015
 - ▶ Server deployment, configuration, release & patch management

Why? Limited time & technological change



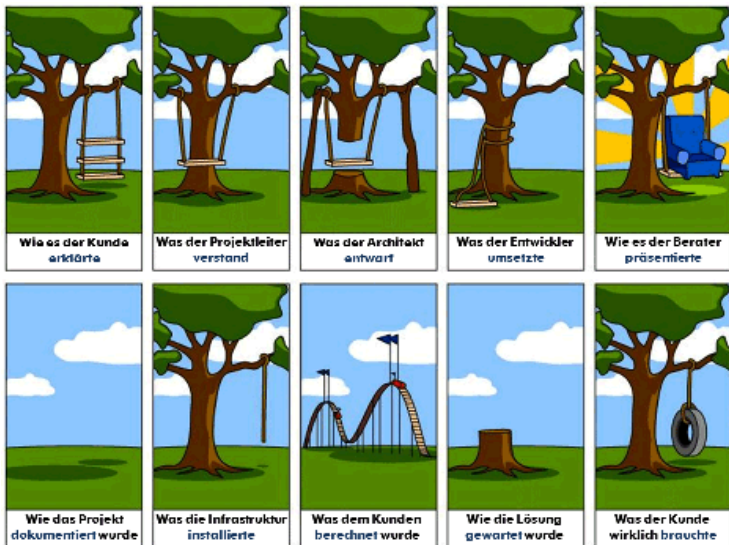
Plateau will be reached:

- less than 2 years
- 2 to 5 years
- 5 to 10 years
- ▲ more than 10 years
- ⊗ obsolete before plateau

© 2016 Gartner, Inc.

Source: Gartner Inc.

Why? Don't know requirements & solutions

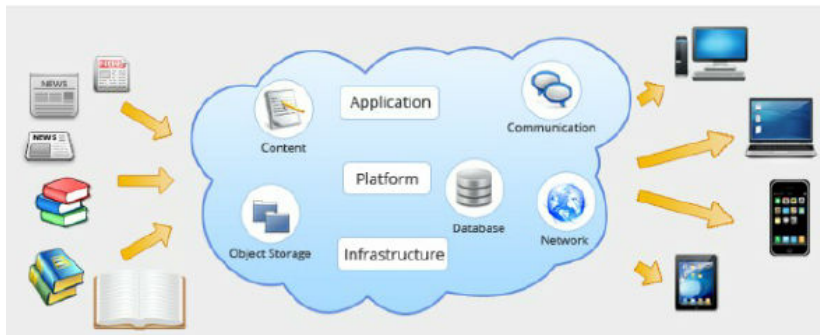


Source: Unknown

The dark age of holding workshops



Source: ATIX AG

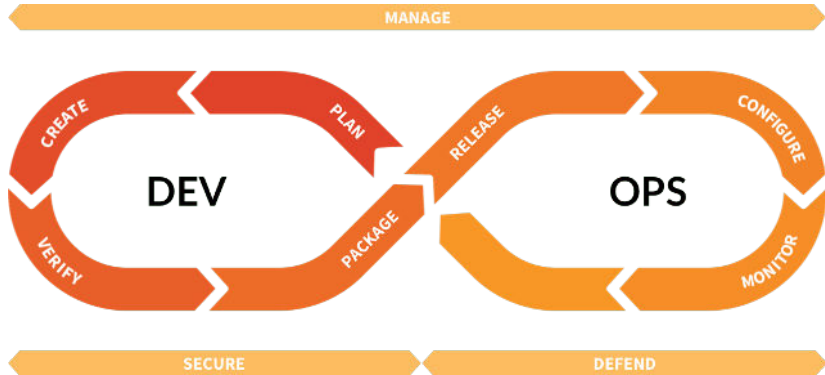


Source: Khaddage, Ferial, and Jesús Hernández Cosío

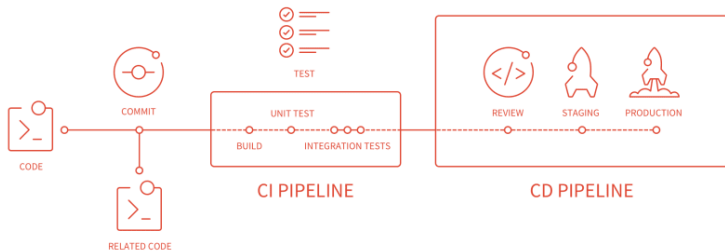
Web application for the management of Git repositories and more...

- ▶ Start as a private project by Dmitri Saporoschez in 2011
- ▶ GitLab Inc. (2013) with products:
 - ▶ GitLab.com: SaaS offering hosted by GitLab
 - ▶ Self-Managed: GitLab instances in cloud or on-premise
- ▶ Plans from private projects up to enterprise customers:





Source: GitLab Inc.



Source: GitLab Inc.

Pipeline Jobs 5



Source: Own graphic

.gitlab-ci.yml:

```
1 ---
2 stages:
3   - command
4   - only_after_setup
5
6 cache:
```

```
7  .command: &command
8  stage: command
9
10 tags:
11   - docker
12 image:
13   name: example.org/myimage:latest
14   entrypoint: ["sh", "-c"]
15
16 script:
17   - ansible localhost -m template
18     ↪ -a "src=req.yaml.j2 dest=req.yaml"
19   - ansible-galaxy install -r req.yaml
20
21   - ansible-playbook --tags "${CI_JOB_NAME}"
22     ↪ basic.yaml
23
24   - [ "${CI_JOB_NAME}" == "setup" ] &&
25     ↪ tar -cf artifact.tar myconfig
```

1

2

3

```
23 # ansible-playbook --tags "${CI_JOB_NAME}" basic.yaml }
24 cleanup: } 1
25   <<: *command
26   when: manual
27
28 gc: } 2
29   <<: *command
30   only:
31     - schedules
32
33 setup: } 3
34   <<: *command
35   artifacts:
36     paths:
37       - artifact.tar
38     expire_in: 1 week
39   when: manual
```

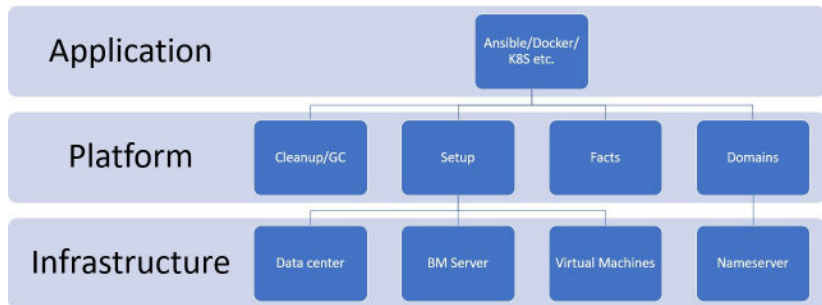
```
40 workshop: _____ 1
41   stage: only_after_setup
42
43   etc. _____ 2
44
45   dependencies: _____ 3
46     - setup
47
48   script:
49     - tar -xf artifact.tar
50
51     - ansible localhost -m template
52     ↪ -a "src=req.yaml.j2 dest=req.yaml"
53     - ansible-galaxy install -r req.yaml
54
55     - ansible-playbook -i hcloud.py workshop.yaml
56
57   when: manual
58   ...
```

1

2

3

4

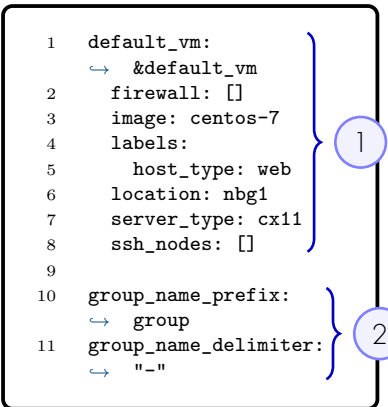


Source: Own graphic

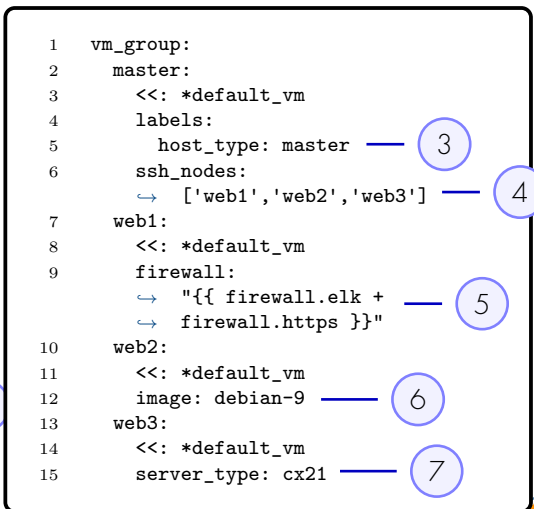
What kind of VMs?

wksp/default/main.yaml:

```
1  default_vm:
   ↳ &default_vm
2  firewall: []
3  image: centos-7
4  labels:
5     host_type: web
6     location: nbg1
7     server_type: cx11
8     ssh_nodes: []
9
10 group_name_prefix:
   ↳ group
11 group_name_delimiter:
   ↳ "-"
```

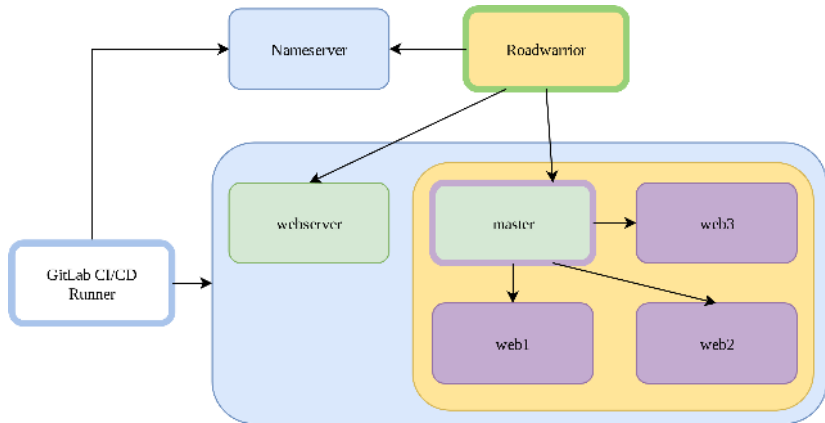


```
1  vm_group:
2     master:
3         <<: *default_vm
4         labels:
5             host_type: master — 3
6             ssh_nodes:
7                 ↳ ['web1', 'web2', 'web3'] — 4
8     web1:
9         <<: *default_vm
10        firewall:
11            ↳ "{{ firewall.elk + — 5
12                ↳ firewall.https }}"
13    web2:
14        <<: *default_vm
15        image: debian-9 — 6
16    web3:
17        <<: *default_vm
18        server_type: cx21 — 7
```



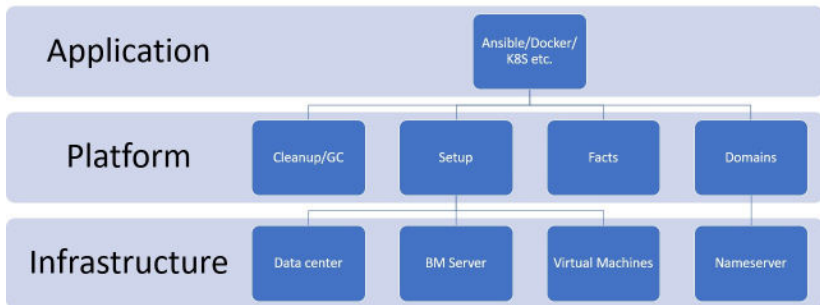
How many? And, further properties

```
1  number_of_vm_groups:
   ↪  "{{ lookup('env', 'NUMBER_OF_VM_GROUPS') }}" } ①
2  roadwarrior_ssh_nodes: ['master']
3
4  firewall:
5    elk:
6      - { chain: INPUT, ctstate: NEW, destination_port:
   ↪  '9200', jump: ACCEPT, protocol: tcp, source:
   ↪  ['master'] } } ②
7      - { chain: INPUT, ctstate: NEW, destination_port:
   ↪  '9300', jump: ACCEPT, protocol: tcp, source:
   ↪  ['master'] }
8    https:
9      - { chain: INPUT, ctstate: NEW, destination_port: '443',
   ↪  jump: ACCEPT, protocol: tcp, source: ['web1',
   ↪  'web3', '213.155.95.23'] } } ③
```



Source: Own graphic

Definition of VMs \Rightarrow workshop's playbook



Source: Own graphic

.gitlab-ci.yml:

```
1  ansible-playbook -i hcloud.py workshop.yaml
```

wksp/default/main.yaml:

```
1  default_vm: &default_vm
2  firewall: []
3  image: centos-7
4  labels:
5    host_type: web — 1
6  location: nbg1
7  server_type: cx11
8  ssh_nodes: []
```

```
1  vm_group:
2    master:
3      <<: *default_vm
4      labels:
5        host_type: master — 2
6      ssh_nodes:
7        ↪ ['web1', 'web2', 'web3']
```

Name	Description
all	contains all hosts
hcloud	contains all hosts in Hetzner Cloud
fsn1_dc8	contains all hosts in datacenter Falkenstein
nbg1_dc3	contains all hosts in datacenter Nürnberg
label1_value1	contains all hosts have label "label1"="value1"
label1_value2	contains all hosts have label "label1"="value2"

Source: <https://github.com/hg8496/ansible-hcloud-inventory>

Name	Description
ansible_host	Public IPv4 Adress
hcloud_server_type	Seryertype eg. CX11
hcloud_image	Name of the used image
hcloud_datacenter	Datacenter the server is running in
hcloud_labels	Instance labels

Source: <https://github.com/hg8496/ansible-hcloud-inventory>

.gitlab-ci.yml:

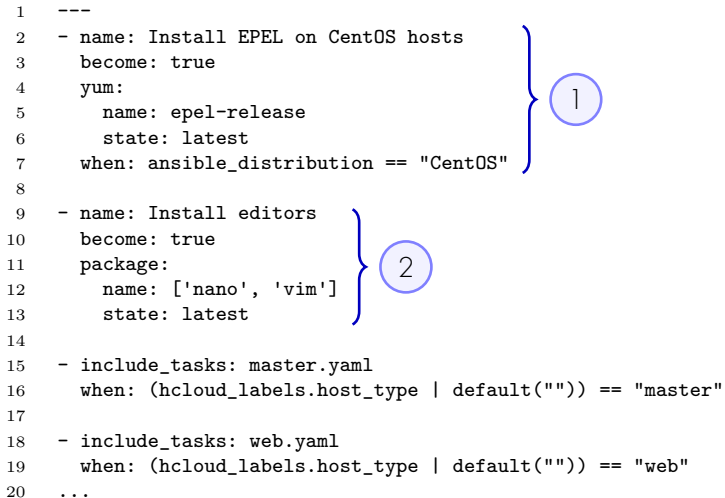
```
1  ansible-playbook -i hcloud.py workshop.yaml
```

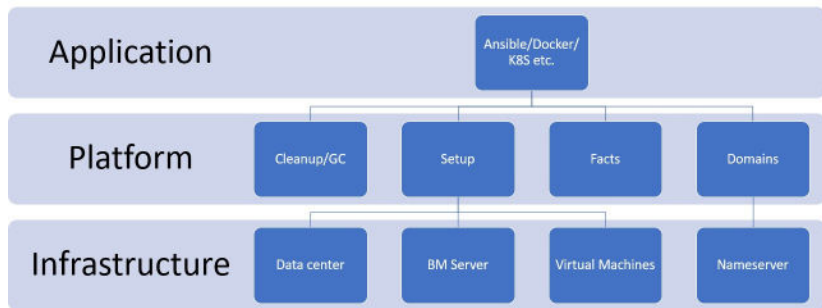
workshop.yaml:

```
1  ---
2  - hosts: hcloud
3    remote_user: "{{ non_root }}"
4    roles:
5      - { role: "{{ lookup('env', 'WORKSHOP_TYPE') }}" }
6  ...
```



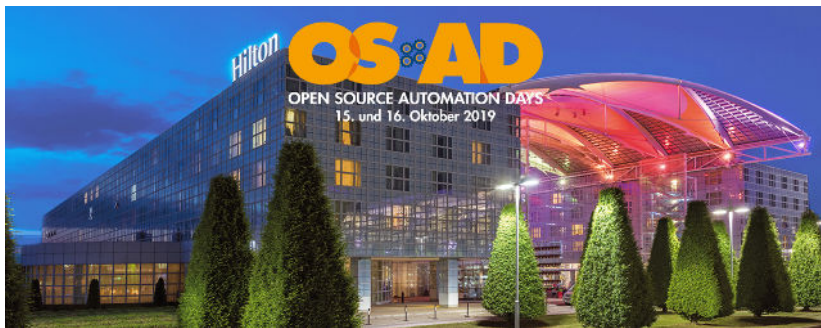
```
1  ---
2  - name: Install EPEL on CentOS hosts
3    become: true
4    yum:
5      name: epel-release
6      state: latest
7    when: ansible_distribution == "CentOS"
8
9  - name: Install editors
10   become: true
11   package:
12     name: ['nano', 'vim']
13     state: latest
14
15  - include_tasks: master.yaml
16    when: (hcloud_labels.host_type | default("")) == "master"
17
18  - include_tasks: web.yaml
19    when: (hcloud_labels.host_type | default("")) == "web"
20  ...
```





Source: Own graphic

Thanks



David Sardari
IT Consultant @ ATIX AG
sardari@atix.de

<https://github.com/ATIX-AG>
<https://www.facebook.com/ATIXDE/>
Twitter: @ATIXAG