

# Automation of malleable CI-Infrastructure using Ansible

Jakob Fritz, Jülich Supercomputing Centre, Forschungszentrum Jülich

## What you need to do

Set configuration:

- Copy OpenStack-login from `cclouds.yml`
- `gitlab.yml`

```
gitlab_url: https://jugit.fz-juelich.de/
runner_descr: Gitlab runner via ansible
runner_reg_token: !vault |
  $ANSIBLE_VAULT;1.2;AES256;jugit
  3731[...]3836
```

- `gitlab_openstack.yml`

```
instance_names: "TestRunner*"
openstack_key_name: "ansible_key"
openstack_key_file: "ansible_key"
name_jumphost: "Jump_host"
```

```
default_runner_config:
  config:
    run_untagged: true
    tags: []
```

```
runners:
  # min example
  - amount: 1
    flavor: "s4"
  # Partly defined config
  - flavor: "m2.large-disk"
    config:
      run_untagged: false
      amount: 1
  # larger example
  - flavor: "m1.large-disk"
    config:
      tags:
        - "tag_with_underscore"
        - "2 numbered tag 1"
      run_untagged: false
      amount: 2
```

Start the ansible-playbook

```
ansible-playbook
--vault-password-file passwd
gitlab_openstack.yml
```

Sit back and watch ansible work ☕

## For whom is this relevant?

You want ...

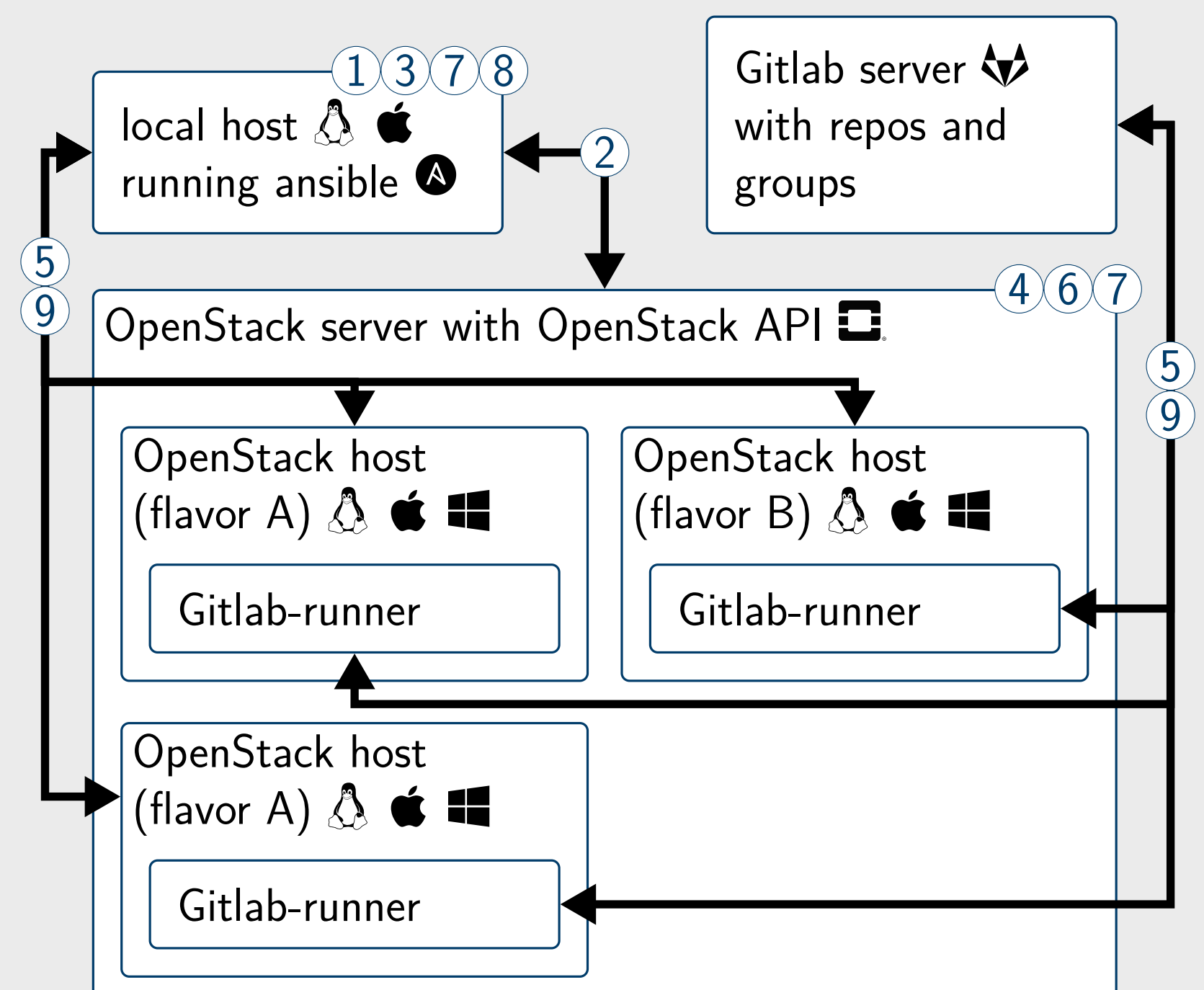
- ... to use Gitlab-CI, but no runners are available
- ... more Gitlab-runners (that you can control)
- ... an easy way to adapt your number of runners

Then, this may be a solution for you

## What prerequisites are needed?

- A system with Linux or Mac-OS to install ansible on
- An OpenStack instance to provide necessary hosts for runners
- A Gitlab-repo/-group to assign the runners to

## Architecture



## What Ansible does for you

