Cloudal

A tool for managing experiment campaigns on cloud platforms

Thuy Linh Nguyen

RainbowFS Final Workshop
Paris, France
28th March 2022

RainbowFS

Outline

- Objective
- Architecture
- elmerfs experiment

Outline

- Objective
- Architecture
- elmerfs experiment

Objective

Experiment-driven research plays an important role in many sciences to **verify new hypotheses**, or to **unveil hidden interactions between many factors** of a phenomenon.

Objective

It is fundamental to run **large-scale experiments** on ***heterogeneous cloud platforms*** to have a better understanding of the system.

How to perform experiments on cloud platform *efficiently* ?

A cloud experiment workflow

1. **provisioning** some hosts (physical machines or virtual machines)
2. **configuring** the experiment environment (installing software or deploying services)
3. performing an **experiment workflow**.

Existing Solutions

Terraform/Pulumi

pros: support public cloud platforms

cons:

- mainly for provisioning
- no support for managing experiments

Existing Solutions

Ansible/Chef/Puppet

pros: support public cloud platforms

cons:

- mainly for configuring hosts
- no support for managing experiments

Existing Solutions

Execo:

pros: support provisioning + managing experiments

cons:

- only support Grid5000
- no support for configuring

Existing Solutions

EnOSlib

pros: support cloud experiment workflow

cons:

- no support for public cloud platforms

so we built

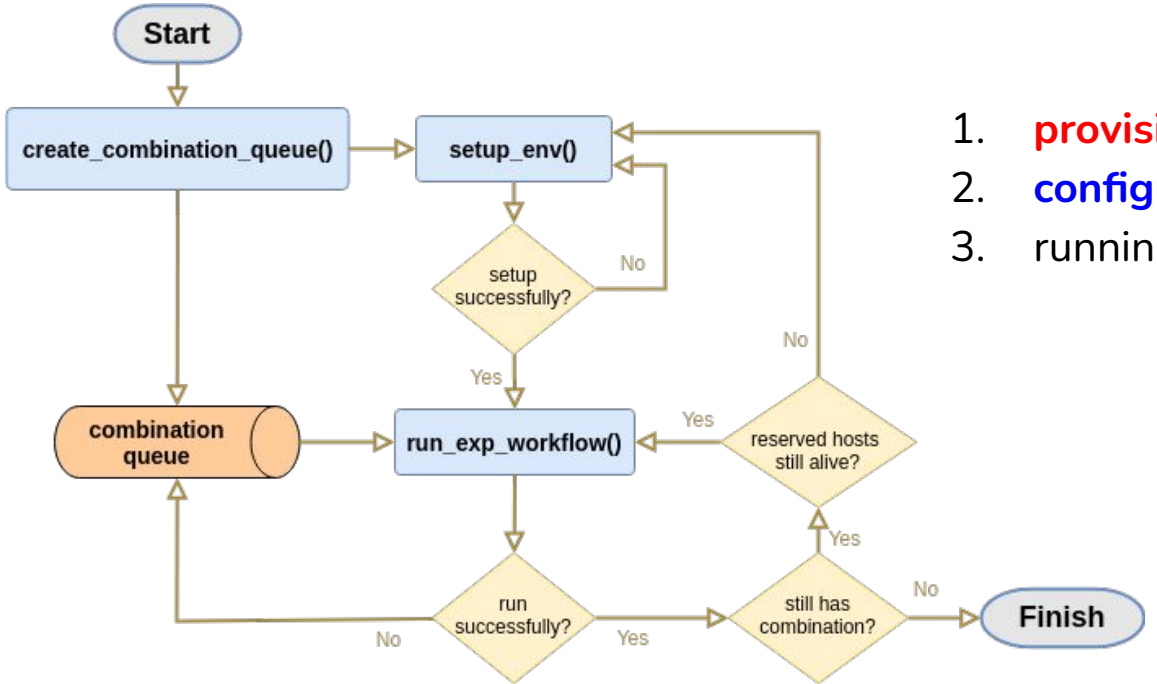
/ˈklaʊdəl/

github.com/ntlinh16/cloudal

Outline

- Objective
- Architecture
- elmerfs experiment

Cloudal experiment workflow

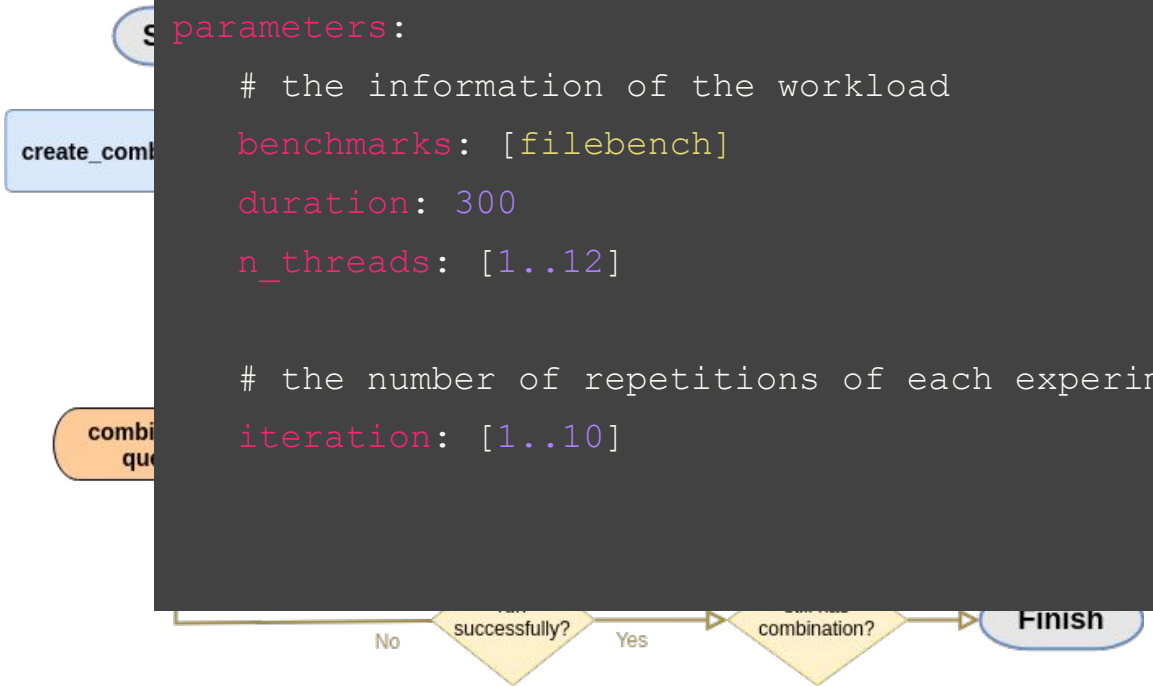


1. **provisioning** some hosts
2. **configuring** hosts
3. running **experiment workflow**

Cloudal experiment workflow

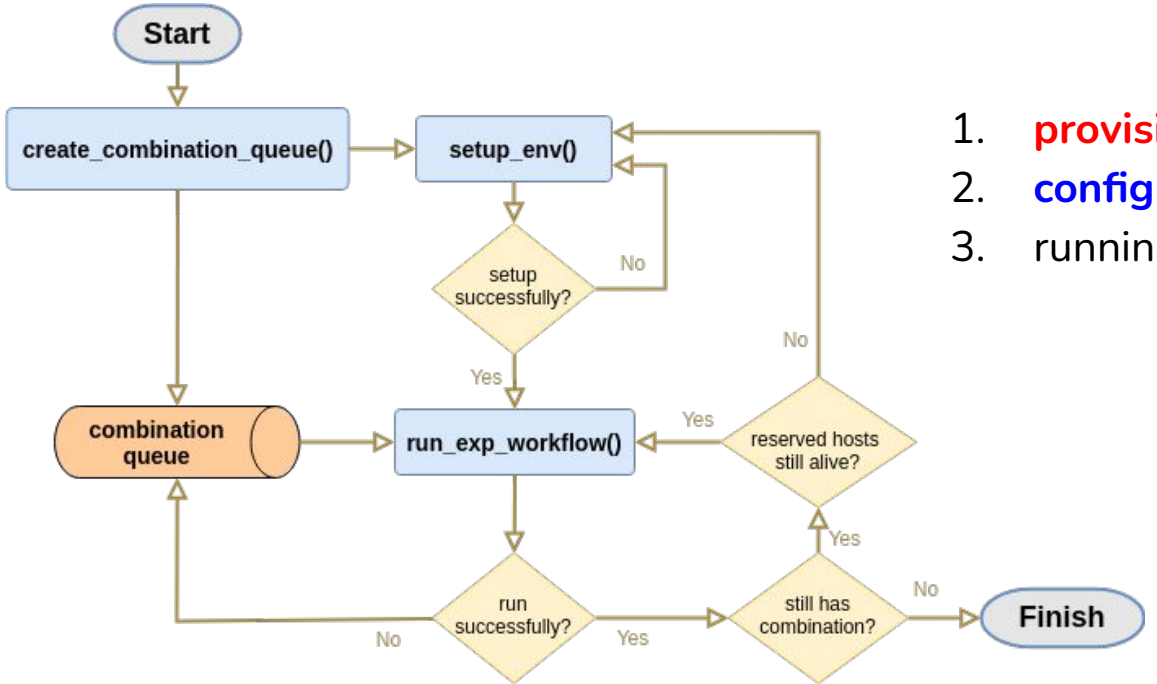
```
parameters:
    # the information of the workload
    benchmarks: [filebench]
    duration: 300
    n_threads: [1..12]

    # the number of repetitions of each experiment
    iteration: [1..10]
```



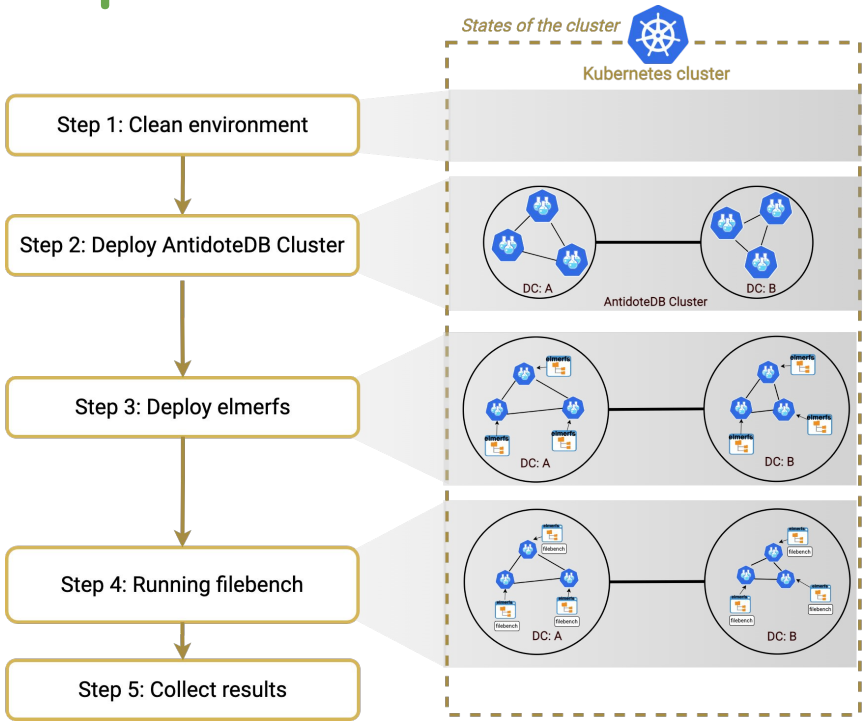
some hosts
hosts
riment workflow

Cloudal experiment workflow

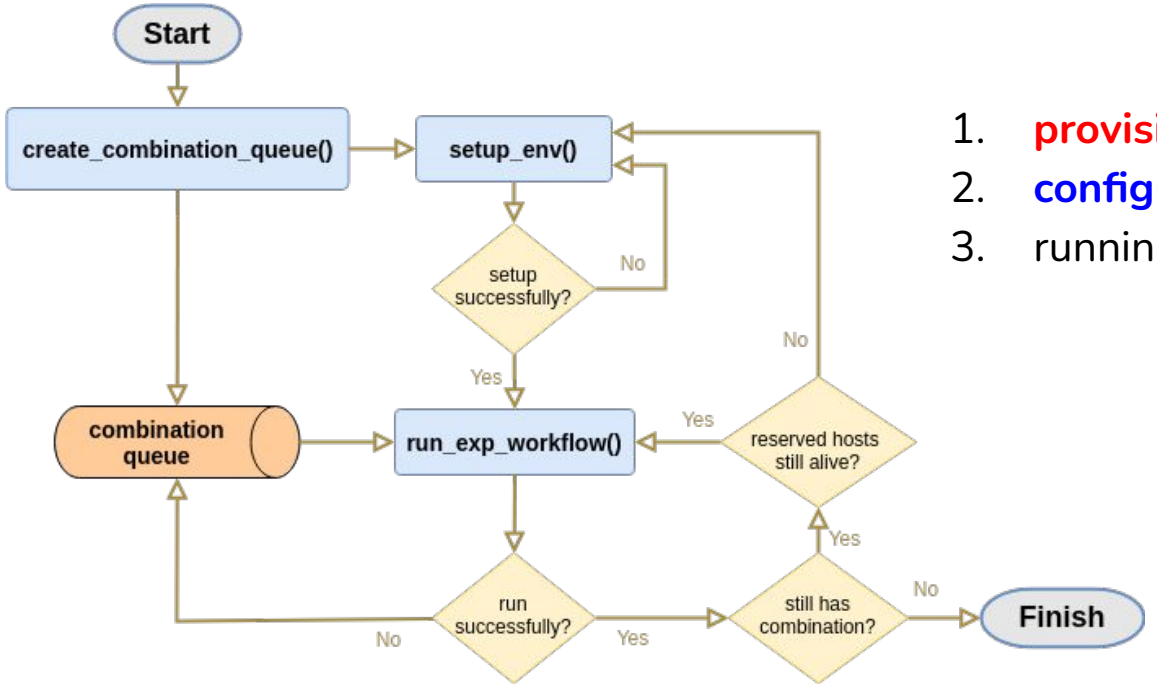


1. **provisioning** some hosts
2. **configuring** hosts
3. running **experiment workflow**

elmerfs experiment workflow

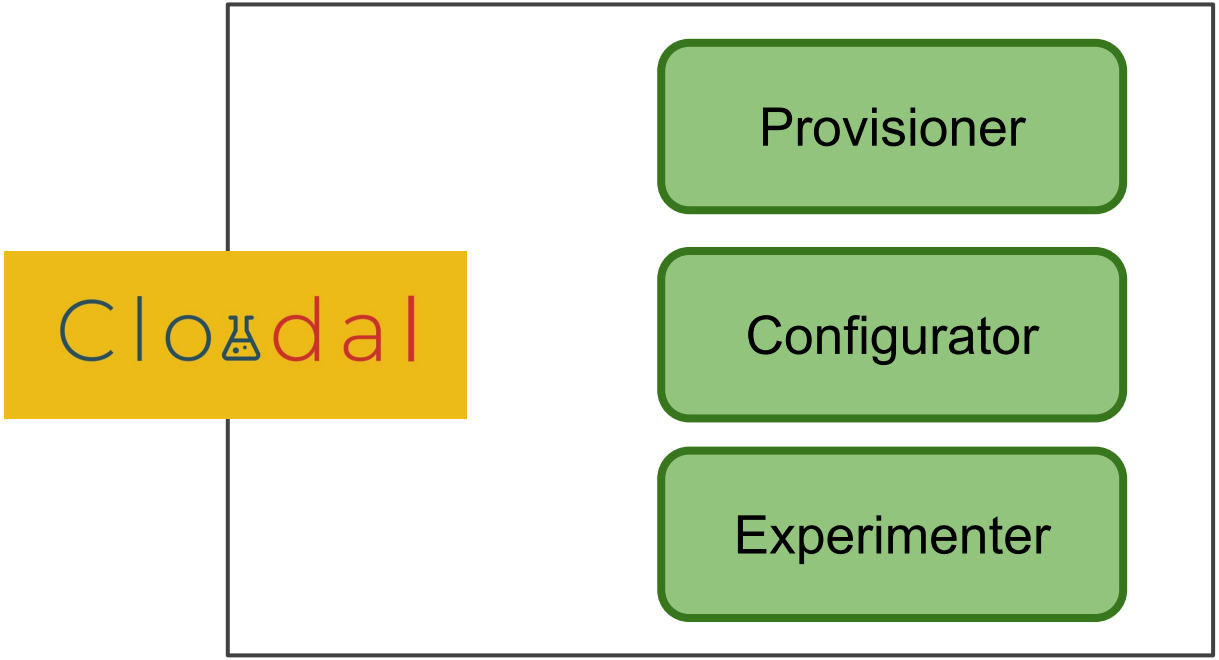


Cloudal experiment workflow

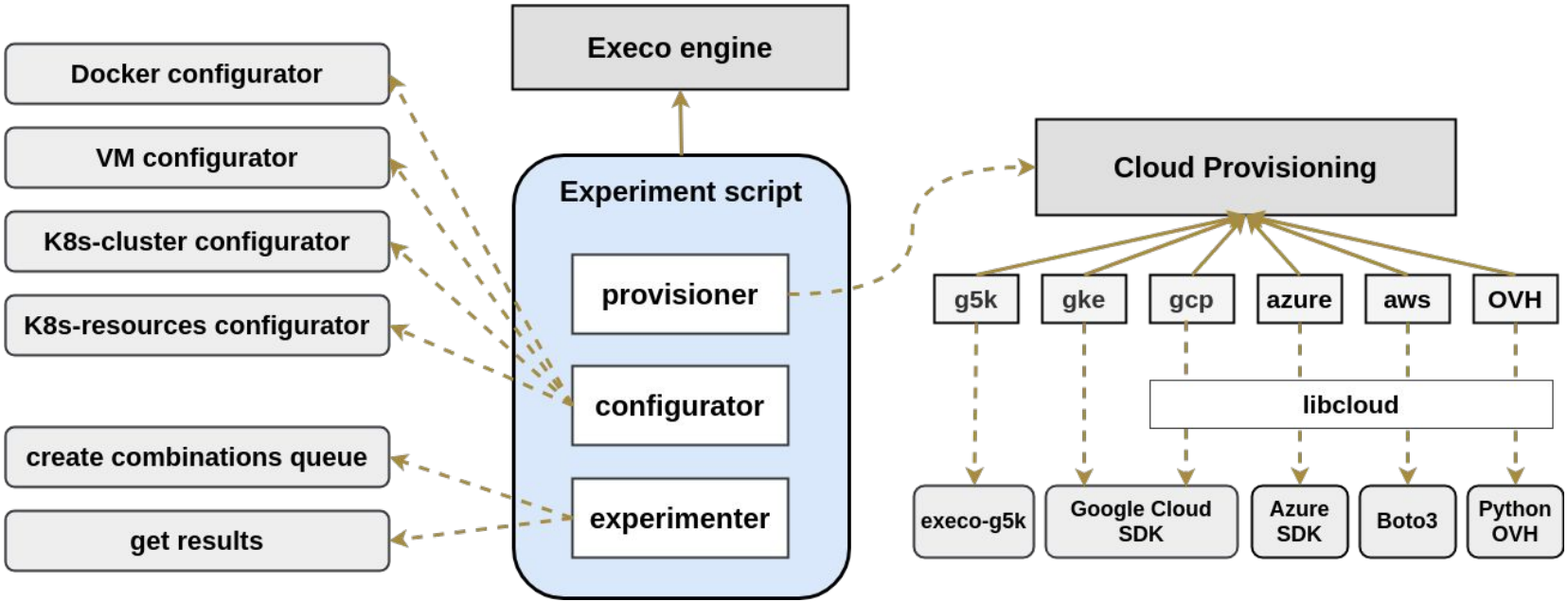


1. **provisioning** some hosts
2. **configuring** hosts
3. running **experiment workflow**

Architecture



Architecture



Outline

- Objective
- Architecture
- elmerfs experiment

Getting Started

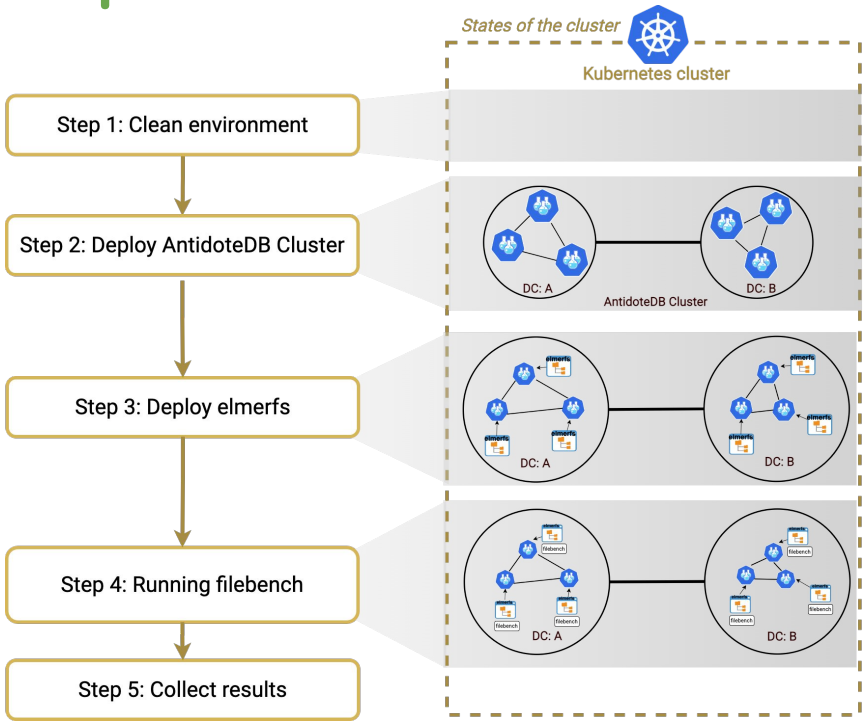
- 3 AntidoteDB DCs
- 6 nodes per DC
- elmerfs is deployed on each AntidoteDB node

- 1 filebench process runs on each elmerfs node
- 1 filebench process generates multiple threads

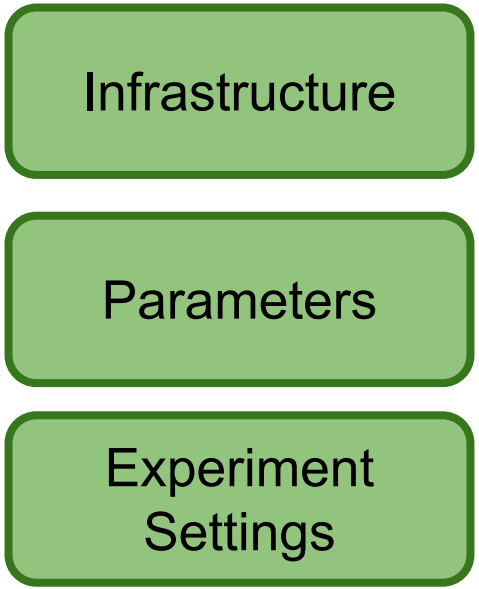
Getting Started

1. **provisioning** 19 hosts
2. **configuring** experiment environment:
   a. deploying a Kubernetes cluster
   b. installing dependencies of elmerfs and filebench
3. running **experiment workflow**.

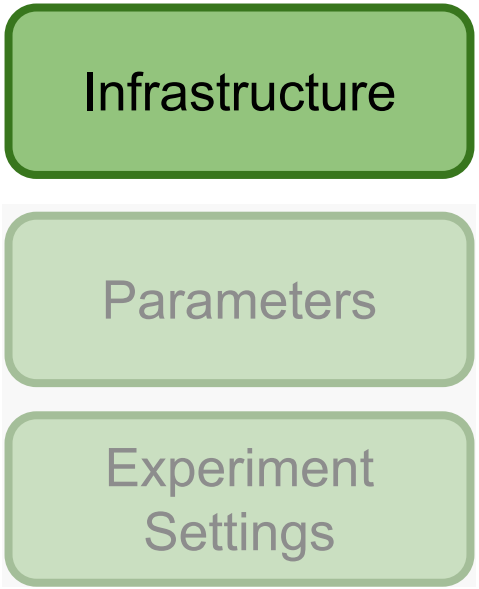
elmerfs experiment workflow



configuration file



configuration file




```
# your authorization information
endpoint: <your_endpoint>
application_key: <your_application_key>
application_secret: <your_application_secret>
consumer_key: <your_consumer_key>
project_id: <your_project_id>

instance_type: b2-60
image: Debian 11
flexible_instance: False
kube_master_site: SBG5
clusters: [SBG5, SGP1, BHS3]
n_nodes_per_dc: [6]
```

configuration file

Infrastructure

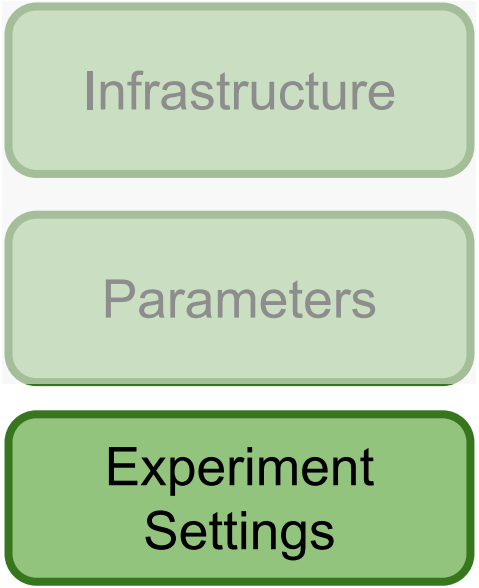
Parameters

Experiment Settings

```
parameters:
    # the information of the workload
    benchmarks: [filebench]
    duration: 300
    n_threads: [1..12]

    # the number of repetitions of each experiment
    iteration: [1..10]
```

configuration file




```
exp_env:
  results_dir: elmerfs-eval/results
  antidotedb_yaml: elmerfs-eval/antidotedb_yaml
  monitoring_yaml: elmerfs-eval/monitoring_yaml
  elmerfs_repo: https://github.com/scality/elmerfs
  elmerfs_version: latest
```

Provisoning

cloudal

```
1 from cloudal.provisioner import ovh_provisioner
2
3 # provision hosts on OVHCloud with requirement from the config file
4 provisioner = ovh_provisioner(config_file_path)
5 provisioner.provisioning()
6
7 # get all the provisioned hosts' IP
8 hosts = provisioner.hosts
9
```

Configuring

cloudal

```
1  from cloudal.configurator import (
2      packages_configurator,
3      docker_configurator,
4      kubernetes_configurator,
5  )
6
7  # deploy a Kubernetes cluster with all hosts
8  configurator = kubernetes_configurator(hosts, kube_master)
9  kube_master, kube_workers = configurator.deploy_kubernetes_cluster()
10
11 # install dependencies of elmerfs and filebench on all hosts
12 configurator = packages_configurator()
13 configurator.install_packages(['fuse3', 'libfuse2', 'jq', 'build-essential', 'bison', 'flex', 'libtool'], hosts)
14
```

running experiment workflow

cloudal

```python
def run_exp_workflow(kube_namespace, comb, kube_master, sweeper):
    comb_ok = False
    try:
        clean_exp_env(kube_namespace)
        deploy_antidote(kube_namespace, comb)

        deploy_elmerfs(kube_master, kube_namespace, elmerfs_hosts)
        if is_elmerfs:
            is_finished, hosts = run_benchmark(comb, elmerfs_hosts)
            if is_finished:
                save_results(comb, hosts)
                comb_ok = True
            else:
                comb_ok = False
    finally:
        if comb_ok:
            sweeper.done(comb)
        else:
            sweeper.cancel(comb)
    return sweeper
```

elmerfs experiments

https://github.com/ntlinh16/elmerfs-eval

cloudal demo

https://asciinema.org/a/481162

Q&A