BonFIRE

Building service testbeds on FIRE

D5.2.5 Virtual Cluster over Federated Cloud Demonstration Kit

August 2012

Version 1.0

Objective:

To show the BonFIRE programming capabilities for developing <u>Experiment Agents</u> using as an example a Virtual Cluster developed by VCOC experiment.

Audience:

Cloud experimenters, programmers, and researchers who would like to know more about how to develop Experiment Agents for BonFIRE

Scenario 3
(complex network)
GÉANT AutoBAHN
and FEDERICA

EPCC (Edinburgh)

PSNC (Poznan)

Scenario 1
(normal internet)

Scenario 2
(emulated network)

Emulab

HP (Bristol)

IBBT Virtual Wall
(Ghent)

INRIA (Rennes)

HLRS (Stuttgart)
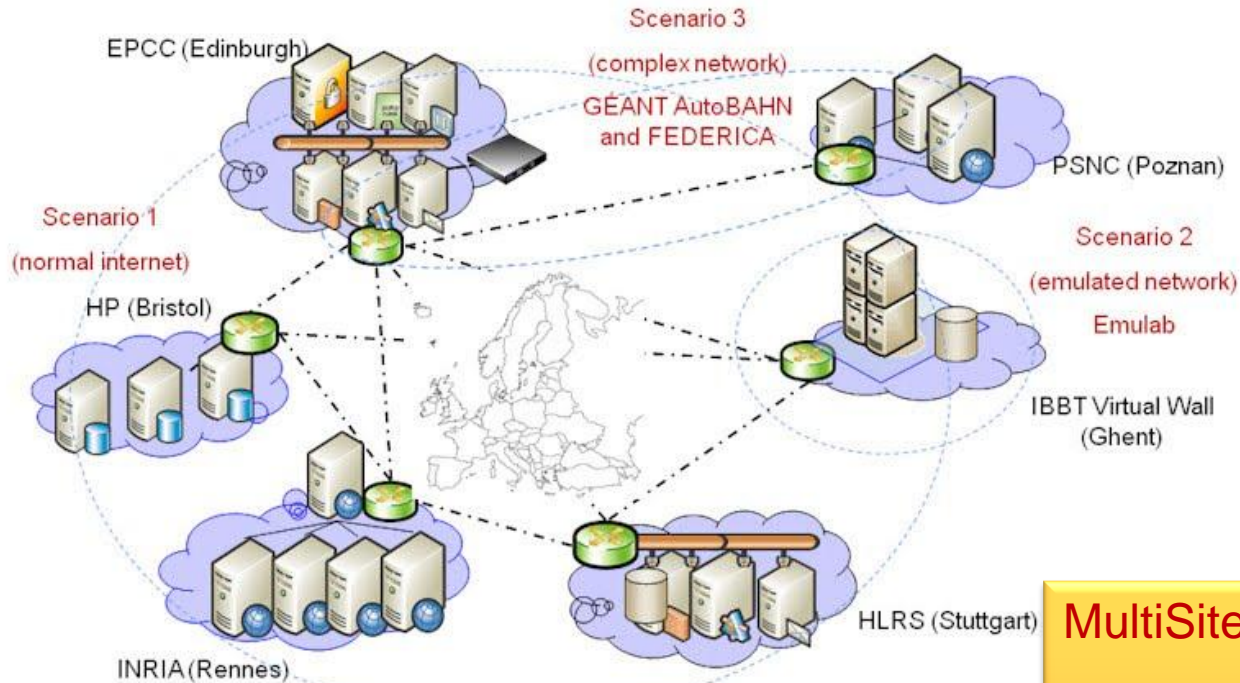
Permanent (~350cores / 30TB) & On-Request (theoretically 3000+ cores) infrastructures

Note: network links indicative only

Source: http://www.bonfire-project.eu/infrastructure (August 2012)

MultiSite Cloud

More information:
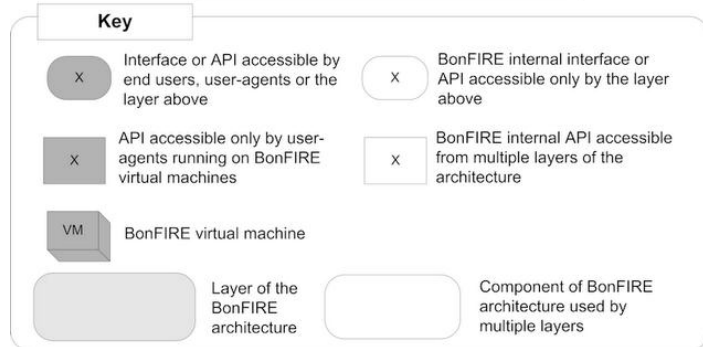
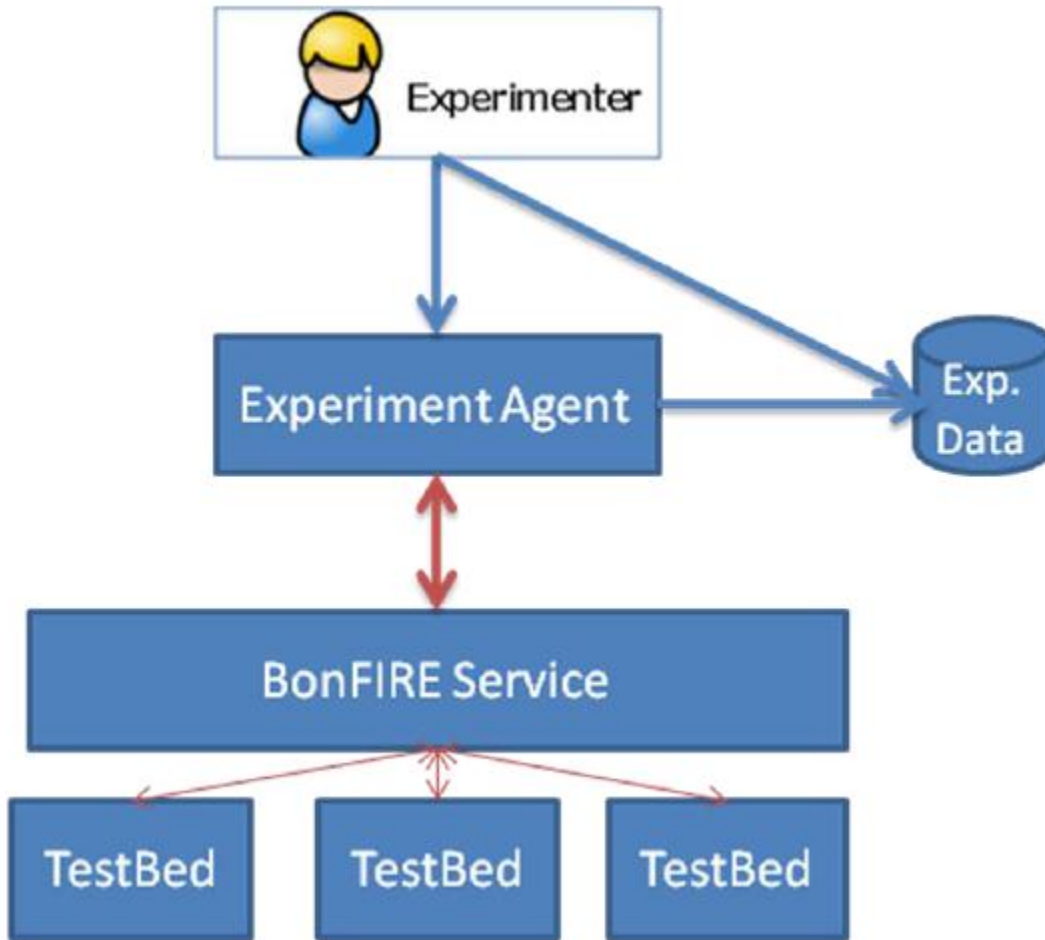http://www.bonfire-project.eu/infrastructure
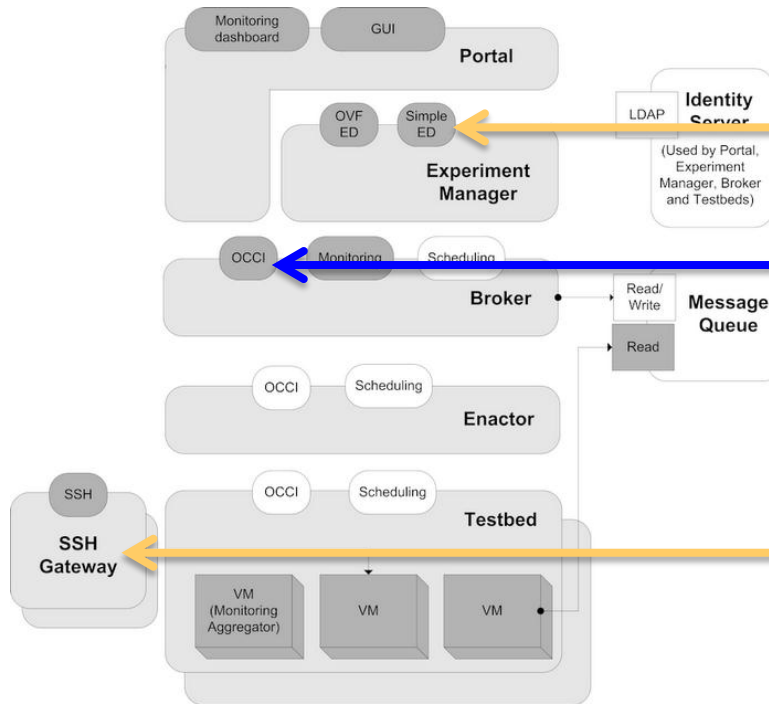
VCOC Experiment Agent uses this interface

Appliance and storage are created using Resource Manager (Broker)

To login to VMs, the SSH gateway is used

Source: http://www.bonfire-project.eu/infrastructure/bonfire-architecture
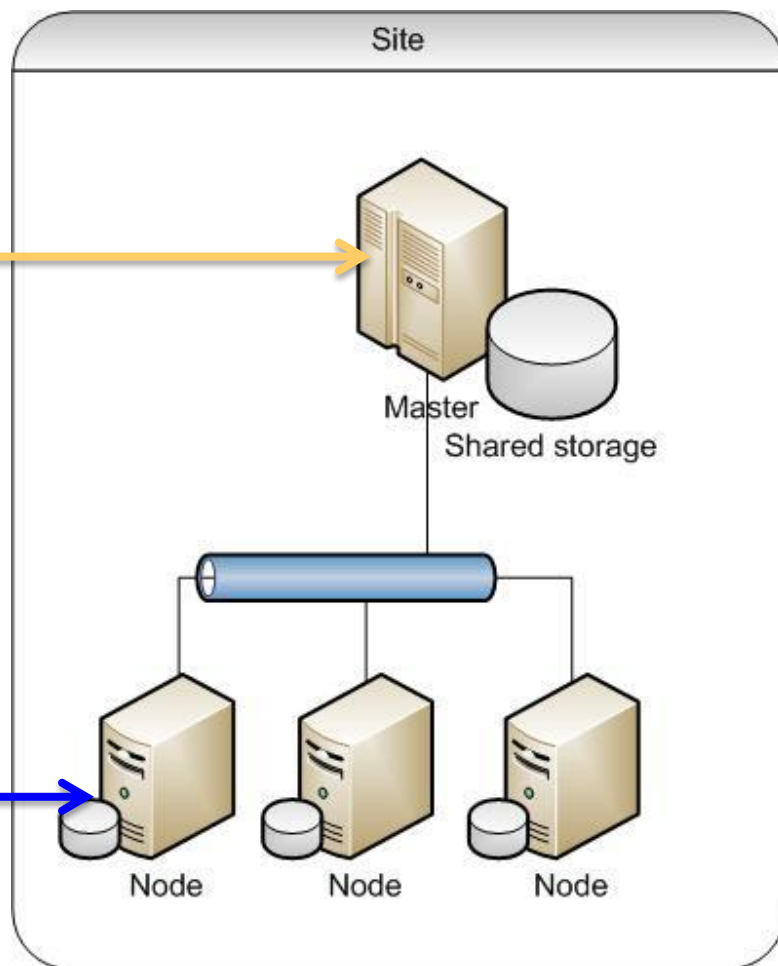
❑ Manages multiple experiment executions

❑ Works with the Experiment Manager

❑ Controls deployed resources

❑ Stores Experiment data

Virtual Cluster Architecture

- Manages Open Grid Scheduler
- Shares filesystem using NFS
- Controls setup and executions

- Executes Jobs
- Open Grid Scheduler Compute Element

Site

Master   Shared storage

Node   Node   Node

1. Create your own appliance:

Master, client, and additional disk storage resources will be created and configured contacting with BonFIRE Resource Manager using Ruby script based on BonFIRE *restfully*.

2. Create your experiment descriptor:

A Virtual Cluster Experiment Descriptor will be created that uses these private appliances. It utilizes *curl + BonFIRE API* to check that these storage resources are available.

3. Execute the experiment:

The Experiment Descriptor will be executed. The Experiment Agent will control the deployment of the experiment and the status of the Virtual Cluster, recording times for each step. Once the experiment is deployed, it is automatically deleted.

You must have:

- A BonFIRE account (request it from http://portal.bonfire-project.eu)

You need:

- LINUX machine, and
- openSSH
- curl
- python 2.7 with modules httplib2, multiprocessing, etree, and libxml2
- ruby (version 1.8.7)
- restfully (see install instructions from BonFIRE at http://doc.bonfire-project.eu)
- subversion

Or download the Demo Kit Appliance:

- From: https://upload.cesga.es/?vid=67f214a3-95e3-8f68-827e-00003b0134c1
- OVA format.
- Import it into your VM manager. It has been tested only on VirtualBox
- Password *bonfire2012*

Using Subversion:

– If your own LINUX, execute (**password is** anonsvn):

svn checkout –username anonsvn http://cvs.forge.cesga.es/svn/bonfiredemokit

– In the distributed appliance, upgrade to last version

svn update –username anonsvn

Configure SSH:

- Generate a key using *ssh-keygen* command
- Upload public key to your BonFIRE portal user details.
- Configure SSH Gateway parameters to access deployed VM (they will use private BonFIRE WAN network without public IPs).
- Templates at *bonfiredemokit/security/ssh*

Steps

- Go to directory ~/.ssh
- Execute (DO NOT SET PASSWORD FOR KEY)

    ssh-keygen –t rsa

- Copy bonfiredemokit/security/ssh/config
- Edit *config* file and substitute *[login@]* by yourUsername@
- Copy public key and upload it to your BonFIRE portal user details

Configure restfully:

- Add your user and password to .restfully/api.bonfire-project.eu.yml
- Templates at *bonfiredemokit/security/restfully*

Steps

- Create directory ~/.restfully
- Copy bonfiredemokit/security/restfully/api.bonfire-project.eu.yml
- Protect it with *chmod 700 api.bonfire-project.eu.yml*
- Edit it to add your BonFIRE username and password

Step 1. Generate your Master private appliance

Create Master VM appliance:

– Restfully-script using BonFIRE Resource Manager: *build-master.rb*

– The script creates an experiment

– It starts a BonFIRE-provided VM appliance

– Remotely installs needed software

– Uploads OGS software (it will be deployed on startup)

– Uploads VCOC control scripts

– Saves new machine as VirtualClusterMaster-<your username>-v0.1

Steps

– Go to directory bonfiredemokit/virt-cluster

– Execute:

restfully –c ~/.restfully/api.bonfire-project.eu.yml build-master.rb

Create Client VM appliance:

- Restfully-script using BonFIRE Resource Manager: *build-client.rb*
- It creates an experiment
- Starts a BonFIRE-provided VM appliance
- Remotely installs needed software
- Saves new machine as VirtualClusterClient-<your username>-v0.1

Steps

- Go to directory bonfiredemokit/virt-cluster
- Execute:

restfully –c ~/.restfully/api.bonfire-project.eu.yml build-client.rb

BonFIRE

Create Additional Storage for Master:

– Restfully-script using BonFIRE Resource Manager: *build-storages.rb*

– It creates an experiment

– Creates a new private permanent storage resource

  • <your username>_volume-1GB-empty

– Starts a BonFIRE-provided VM appliance and attaches the new storage resource

– Initialises it to 0's

– Saves it and changes its persistent property to FALSE.

Steps

– Go to directory *bonfiredemokit/virt-cluster*

– Execute:

restfully –c ~/.restfully/api.bonfire-project.eu.yml build-storages.rb

BonFIRE

Create JSON Experiment Description:

- Bash script: create_vcoc.sh
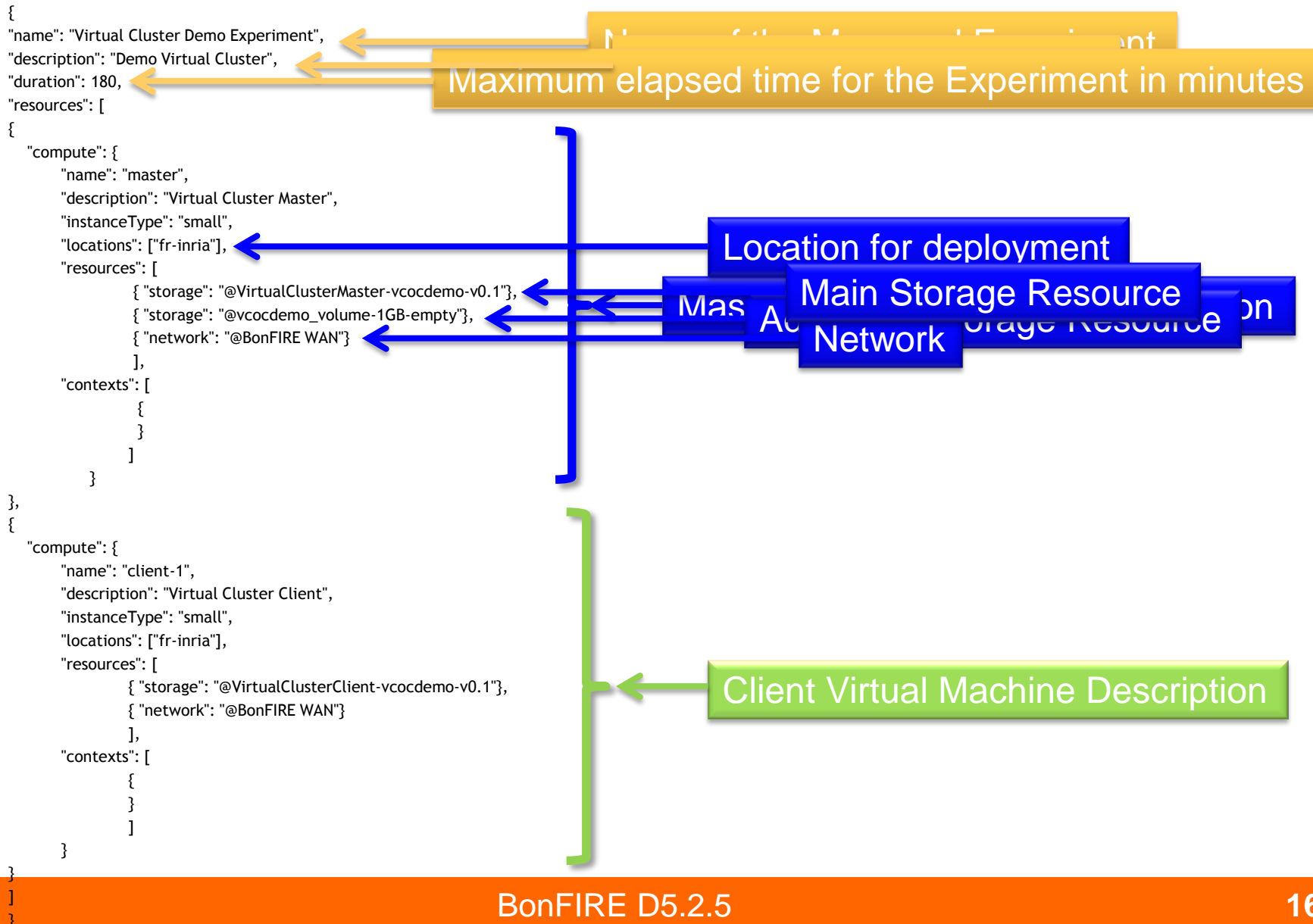- It checks that needed appliances and storage resources are available using *curl* + *BonFIRE API* interface
- Creates a Virtual Cluster JSON descriptor on ../examples
- Virtual Cluster is composed of one master machine with additional storage resource plus one computing element.

Steps

- Go to directory *bonfiredemokit/virt-cluster*
- Execute:

    bash create_vcoc.sh -n 1

BonFIRE

```json
{
"name": "Virtual Cluster Demo Experiment",
"description": "Demo Virtual Cluster",
"duration": 180,
"resources": [
{
    "compute": {
        "name": "master",
        "description": "Virtual Cluster Master",
        "instanceType": "small",
        "locations": ["fr-inria"],
        "resources": [
                { "storage": "@VirtualClusterMaster-vcocdemo-v0.1"},
                { "storage": "@vcocdemo_volume-1GB-empty"},
                { "network": "@BonFIRE WAN"}
                ],
        "contexts": [
                {
                }
                ]
            }
},
{
    "compute": {
        "name": "client-1",
        "description": "Virtual Cluster Client",
        "instanceType": "small",
        "locations": ["fr-inria"],
        "resources": [
                { "storage": "@VirtualClusterClient-vcocdemo-v0.1"},
                { "network": "@BonFIRE WAN"}
                ],
        "contexts": [
                {
                }
                ]
        }
}
]
}
```

Name of the Managed Experiment

Maximum elapsed time for the Experiment in minutes

Location for deployment

Main Storage Resource

Master ... Additional Storage Resource ... Description

Network

Client Virtual Machine Description

Execute the Managed Experiment with Experiment Agent:

- Python program
- It deploys Experiment using BonFIRE Experiment Manager
- Checks status of the deployment
- Checks VMs are ready
- Checks OGS is running
- Records time measurements
- Deletes experiment
- Saves data

Steps

- Go to directory *bonfiredemokit/ea*
- Execute:

python submit_file.py vcoc.json

CONGRATULATIOS!!!!

You have executed your first experiment!!!

Now, explore the scripts to create your own

Demo Kit Provided by

CESGA

Building service testbeds on FIRE

Thank you for your attention