



ANSIBLE

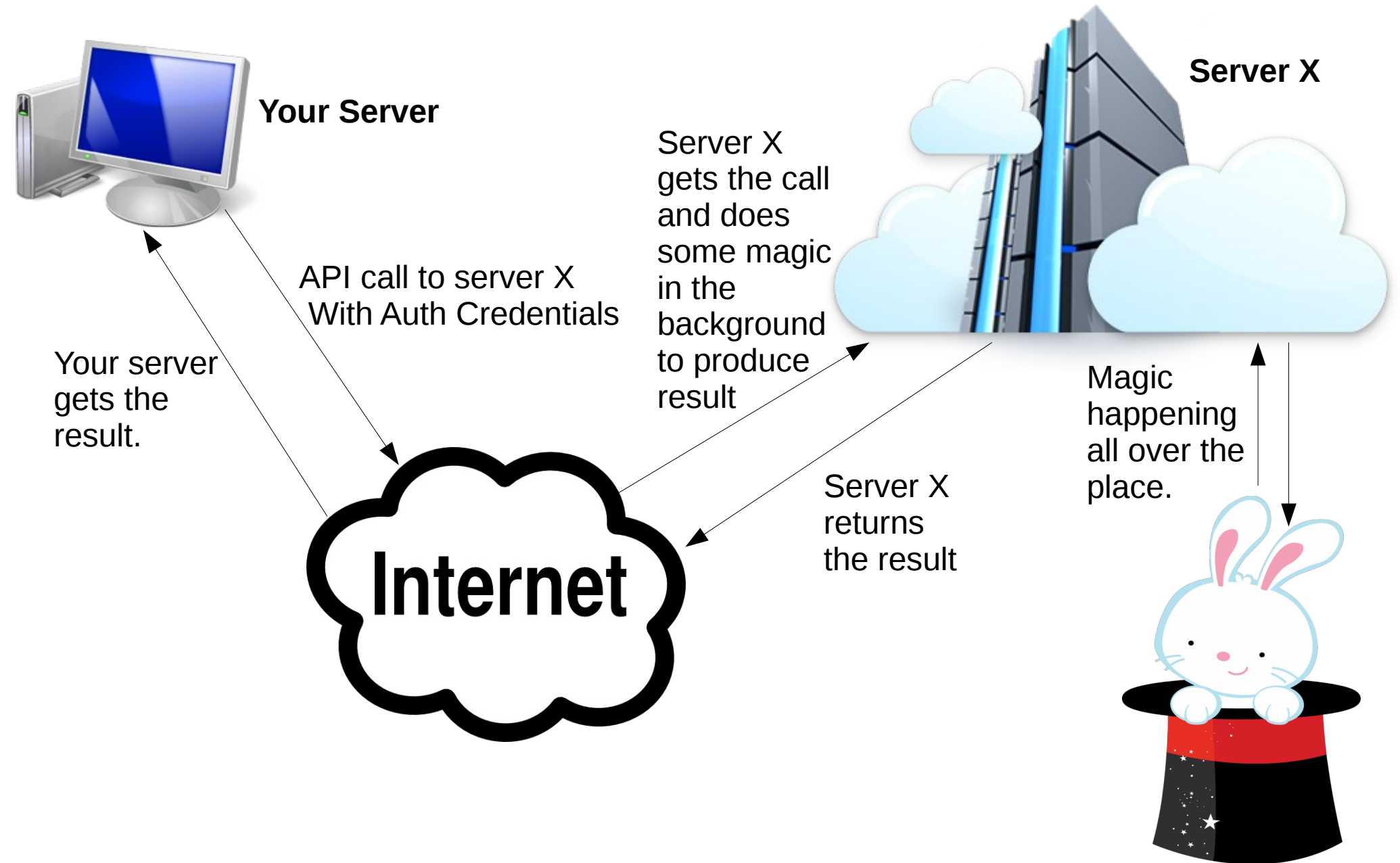
# Running Ansible via API

By: Hussain Nagri

Email: [hussain.nagri@knowlarity.com](mailto:hussain.nagri@knowlarity.com)

Follow me [@imnagri](#) please.....

# How API works with Web Services ?



# Ansible playbook

- Create a playbook

---

- host: controller

tasks:

- name: ensure ntp package is installed

action: apt pkg=ntp

- name: ensure ntp service is restarted

action: service name=ntp state=restarted

# How playbook runs

```
root@vm102:~# ansible-playbook first-playbook.yml -k
SSH password:

PLAY [all] *****

GATHERING FACTS *****
ok: [127.0.0.1]

TASK: [install apache2] *****
changed: [127.0.0.1]

TASK: [Make sure index.html is present for the default virtual host] *****
changed: [127.0.0.1]

TASK: [Make sure index.html is owned by www-data] *****
changed: [127.0.0.1]

NOTIFIED: [start apache2] *****
ok: [127.0.0.1]

PLAY RECAP *****
127.0.0.1           : ok=5    changed=3    unreachable=0    failed=0
```

# Inventory file

[vagrant]

192.168.33.10 ansible\_ssh\_port=22 ansible\_ssh\_user=vagrant

[cluster-1]

192.168.33.12

192.168.33.13

[cluster-2]

192.168.33.14

192.168.33.15

[cluster-3]

192.168.33.16

192.168.33.17

[cluster-4]

192.168.33.18

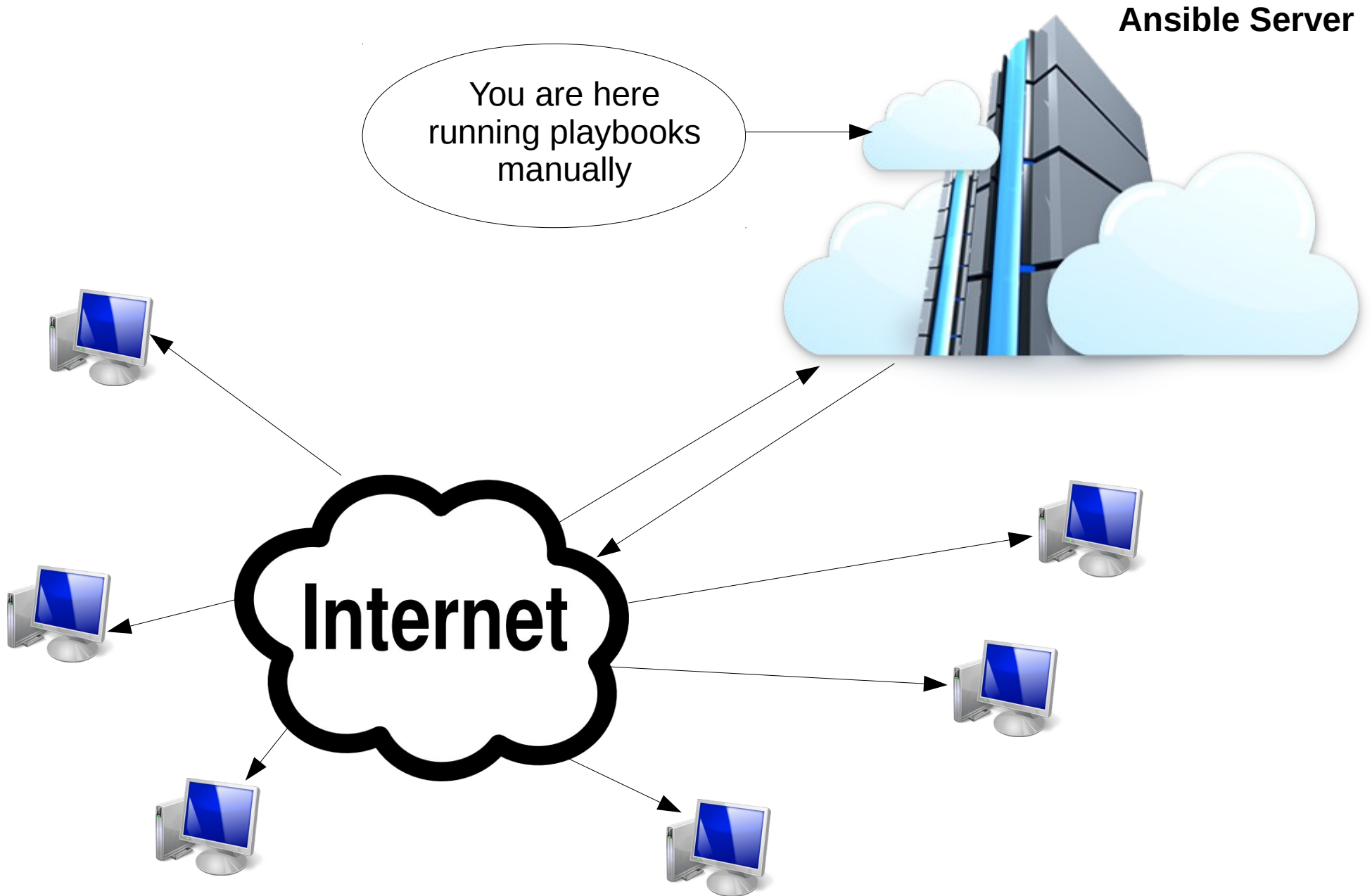
192.168.33.19

[cluster-5]

192.168.33.20

192.168.33.21

# How Ansible works



# How Ansible works

- Create a playbook
- Run the plabook for perticular server or group of servers in a console
- See your playbook run and get bored to death

But

- We mange 100+ servers via a single ansible instance.
- Working on console is boring as well as error prone.
- Though ansible automates things for you sometimes you may want to automate running of ansible.



Who are we ?



DevOps



What do we want ?



To Automate  
Everything



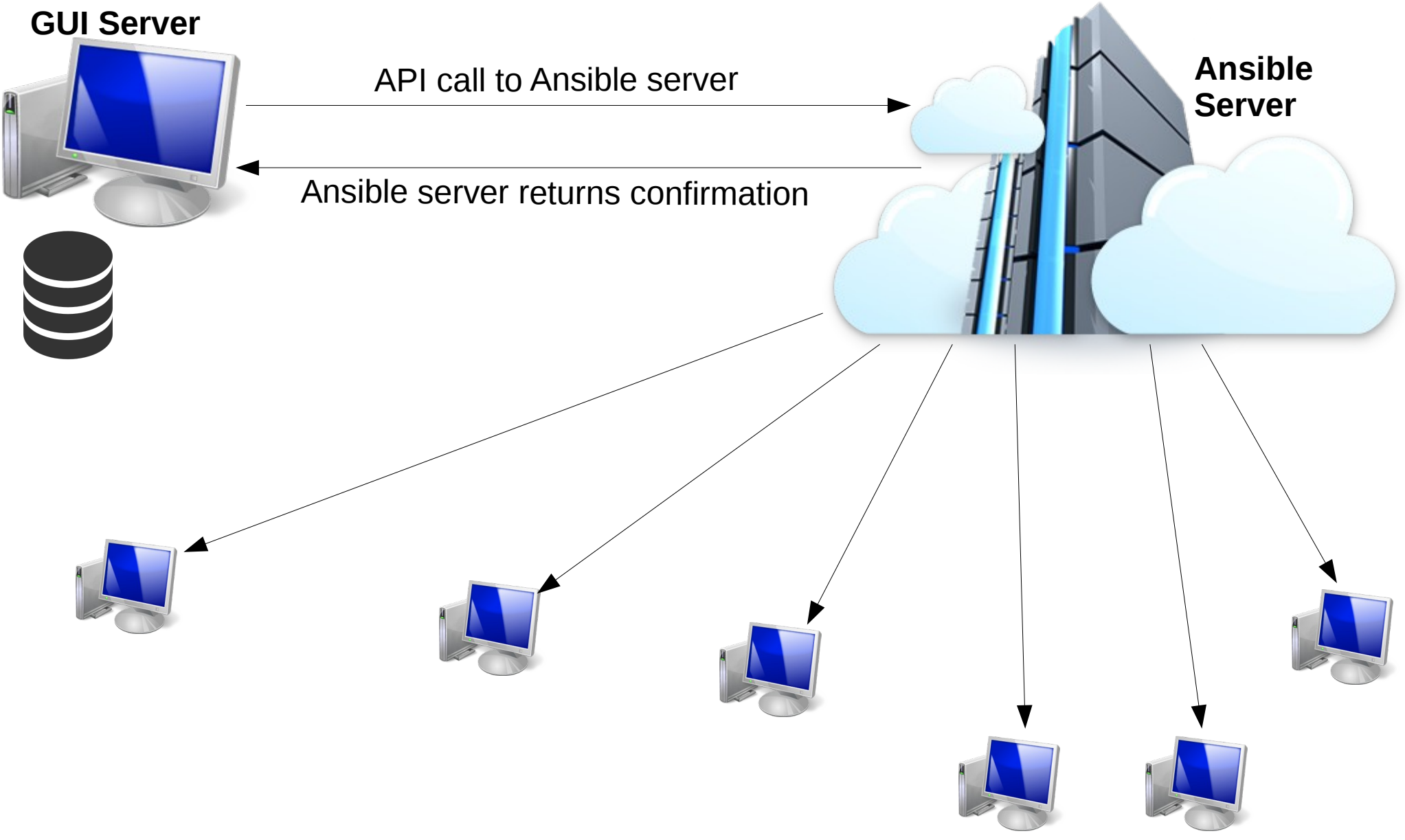
# How our Ansible works ?

GUI Server

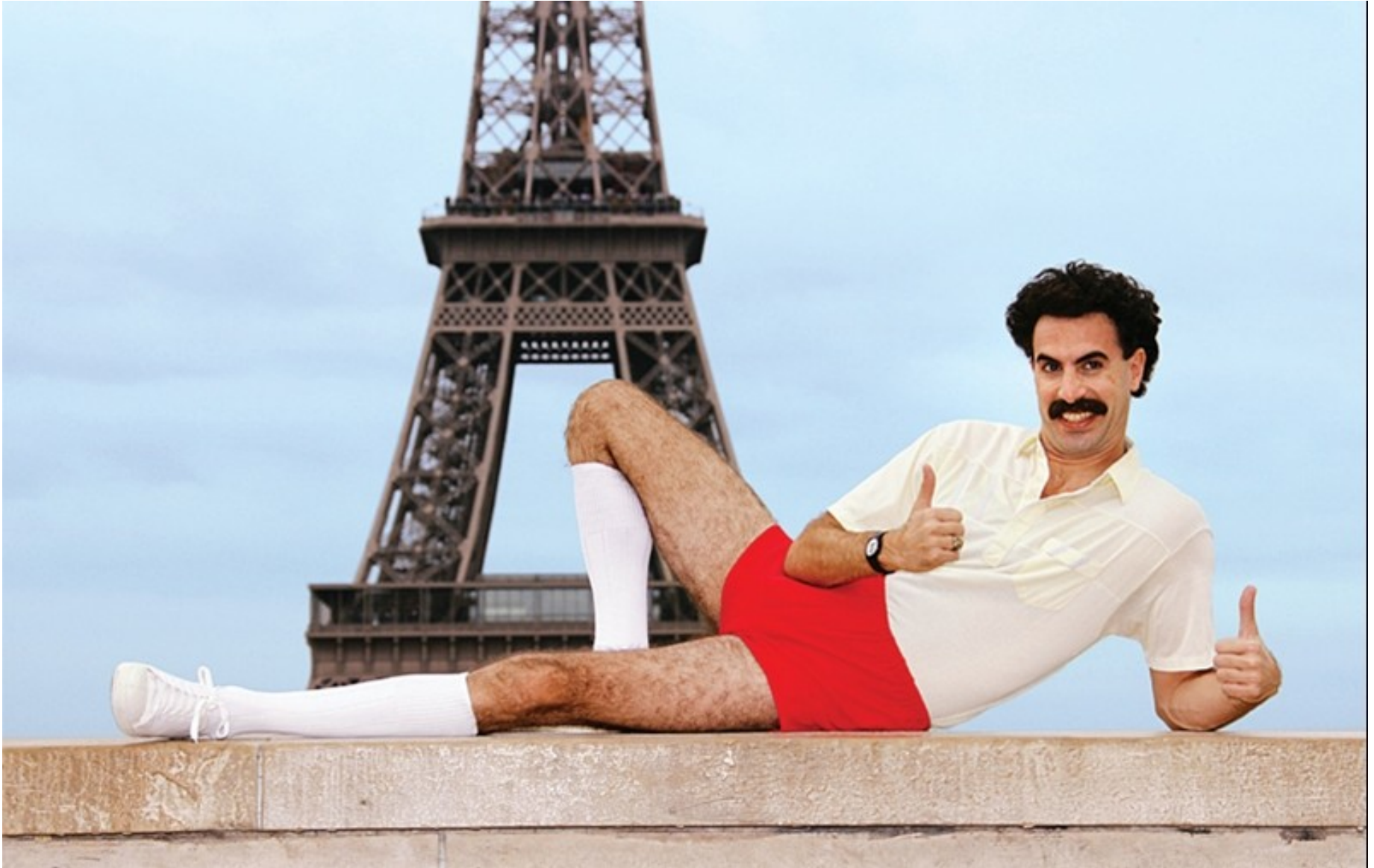
API call to Ansible server

Ansible server returns confirmation

Ansible Server



# Happy DevOps guy



# Python Code

```
import ansible.runner
runner = ansible.runner.Runner(
    module_name='ping', # ping is the module
    module_args="",
    pattern='web*',
    forks=10
)
datastructure = runner.run()
```

# Where

```
# Datastructure >> {'dark': {}, 'contacted': {'192.168.33.10':
# {'invocation': {'module_name': 'ping', 'module_args': ""},
# u'changed': False, u'ping': u'pong'}}}
```

# Another way

```
import ansible.playbook
from ansible import callbacks
from ansible import utils

stats = callbacks.AggregateStats()
playbook_cb = callbacks.PlaybookCallbacks(verbose=utils.VERBOSITY)
runner_cb = callbacks.PlaybookRunnerCallbacks(stats,
verbose=utils.VERBOSITY)

ansible.playbook.PlayBook(
    Playbook='/path/to/playbook.yml',
    callbacks=playbook_cb,
    runner_callbacks=runner_cb,
    stats=stats,
    Inventory='path/to/inventory/file',
    extra_vars={

    }
).run()
```

A terrible demo of such an awesome  
thing

git clone

[https://nagri@bitbucket.org/nagri/ansible-python-api-  
demo.git](https://nagri@bitbucket.org/nagri/ansible-python-api-demo.git)

# How is this working ?

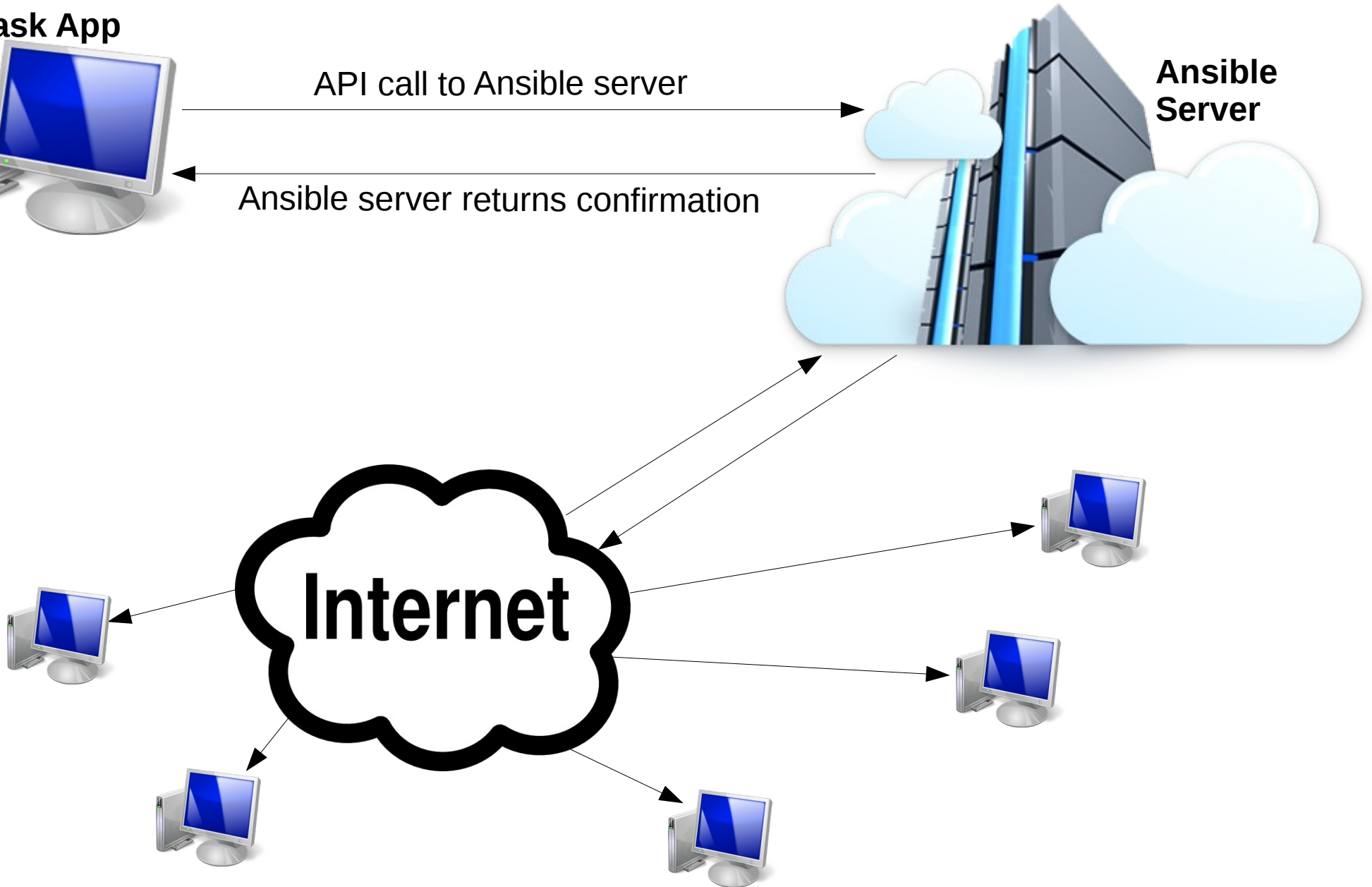
Flask App

API call to Ansible server

Ansible server returns confirmation

Ansible Server

Internet







**Any Questions ?**