Collabora

LibreOffice
The Document Foundation

SUSE

Deploying Uyuni with Sumaform

How to use IaC to deploy test environments for Uyuni project

MySQL

LINUX MAGAZINE

datto

ADMIN Network & Security

fedora

TUXEDO COMPUTERS

.riessapplications

Required Magic
advanced technology

About me

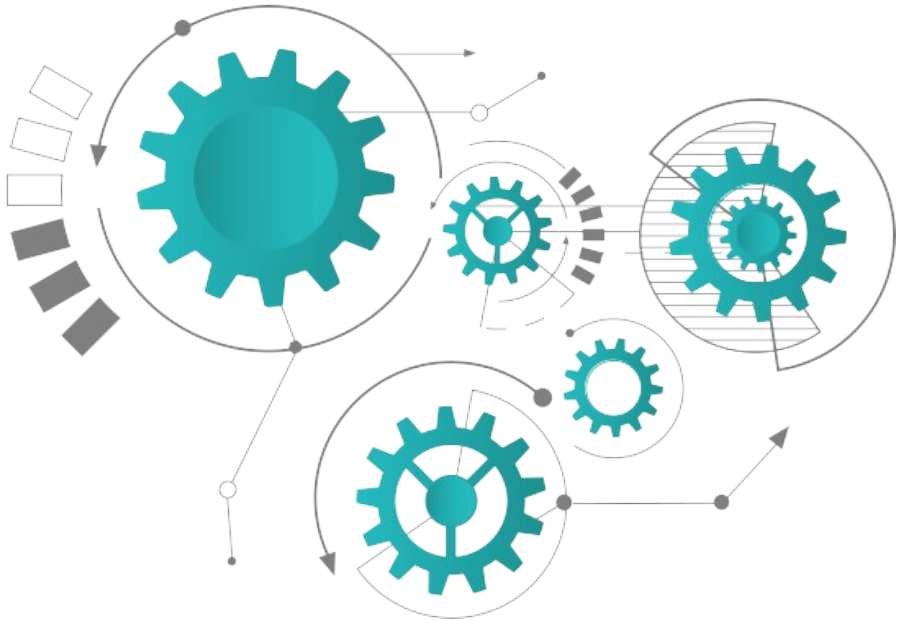
Ricardo Mateus

Software Engineer
SUSE Manager Team
UYUNI Community

rjmateus @ gitter.im
rmateus@suse.com

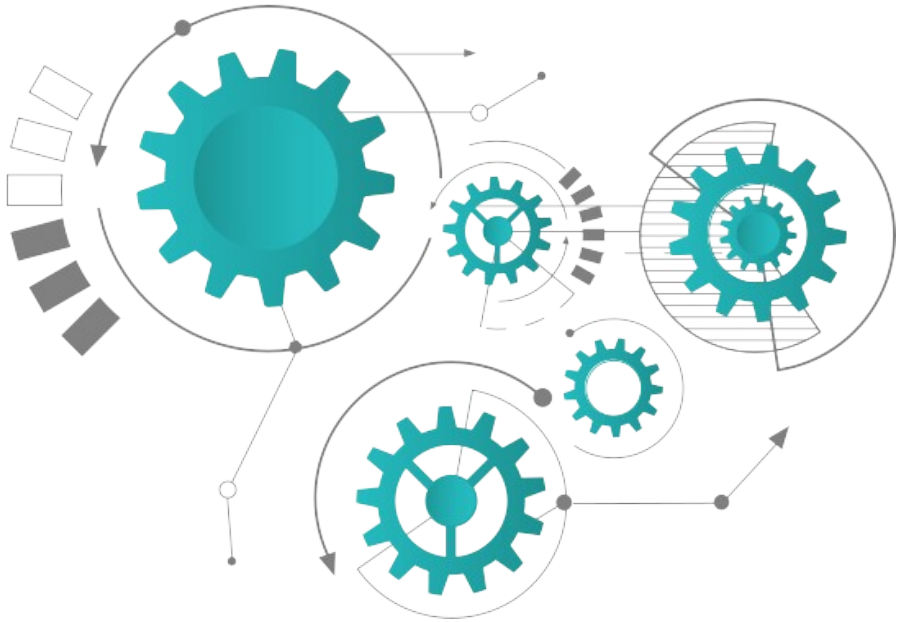
Agenda

- Uyuni Project
- Uyuni deployment
- Sumaform
  - Architecture
  - Modules
  - Demo



Agenda

- Uyuni Project
- Uyuni deployment
- Sumaform
  - Architecture
  - Modules
  - Demo



What?

- Systems management
  - System deployment
  - Patch management
  - Service Pack migration
  - Configuration management

*https://www.uyuni-project.org/uyuni-docs/

What?

- Systems management
  - System deployment
  - Patch management
  - Service Pack migration
  - Configuration management
- Automate audit and reporting capabilities
- Hardware and software inventories

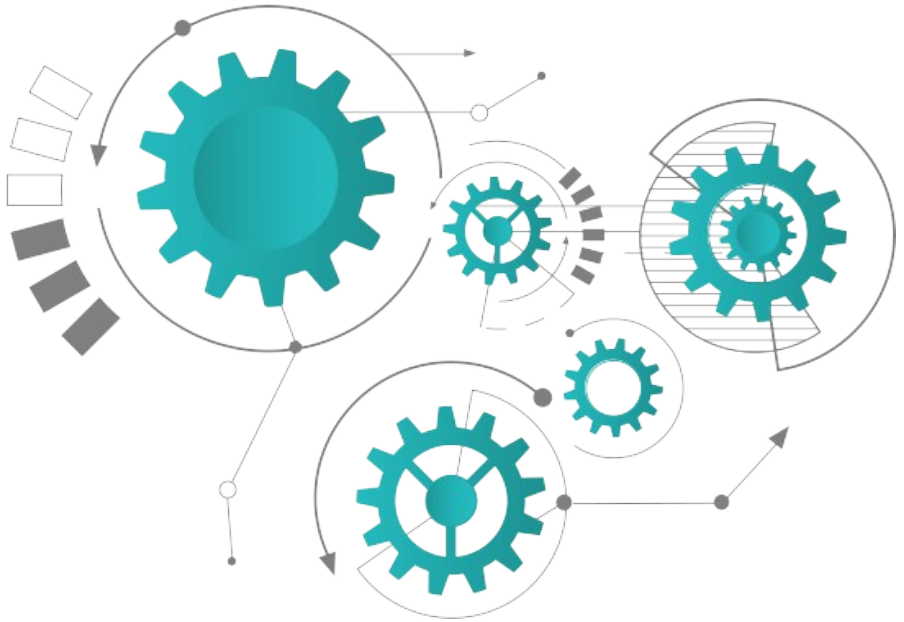
*https://www.uyuni-project.org/uyuni-docs/

Supported Client OS

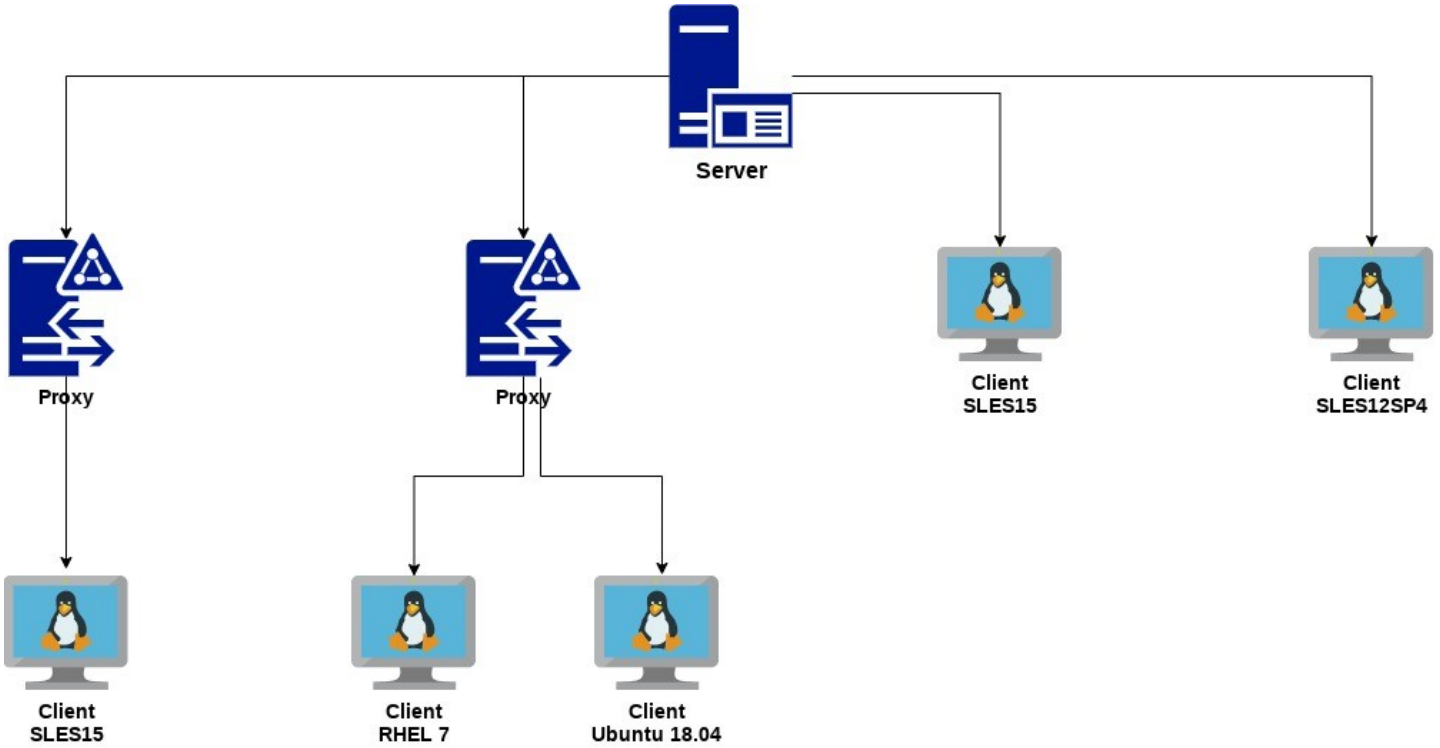
- SLE11/12/15
- RHEL6/7/8
- openSUSE Leap 15
- Ubuntu 16.04/18.04/20.04
- CentOS6/7/8
- Debian 9/10
- Oracle Linux 6/7/8

Agenda

- Uyuni Project
- Uyuni deployment
- Sumaform
  - Architecture
  - Modules
  - Demo



Deployment Architecture



Deployment Needs

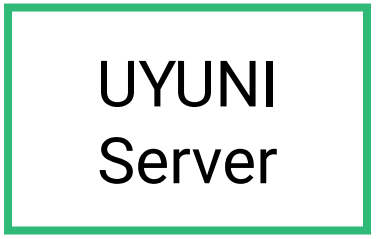
- Development team
  - Bug Fixing: replicate bug environment
  - Test new feature development

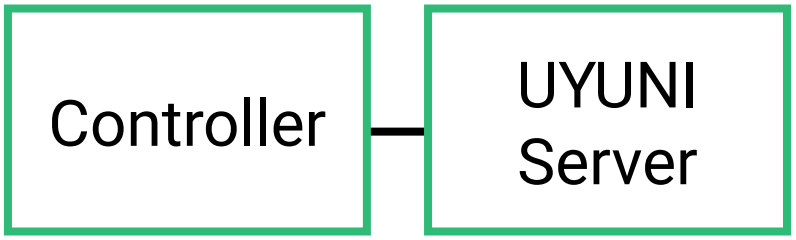
Deployment Needs

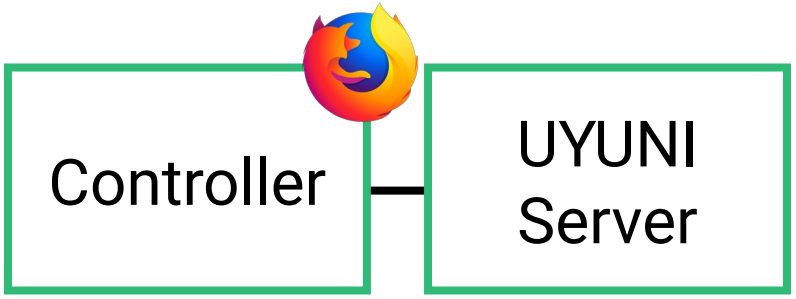
- Development team
  - Bug Fixing: replicate bug environment
  - Test new feature development

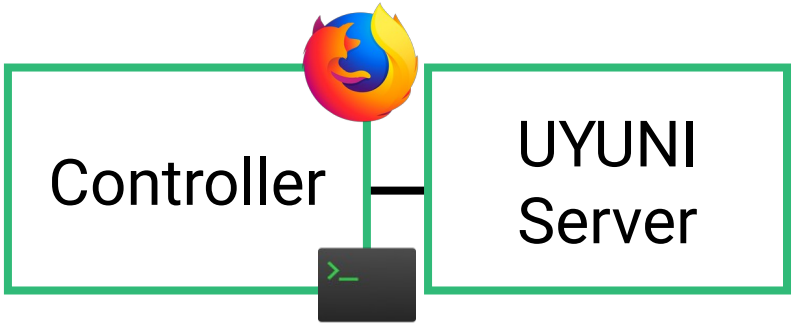
- QA team
  - Manual testing
  - Run cucumber base test-suite

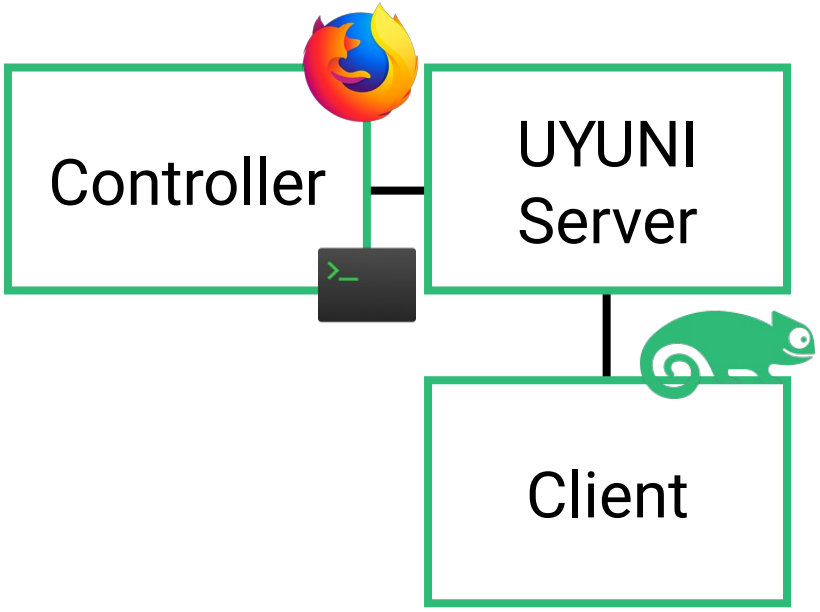
What these architectural needs really mean?

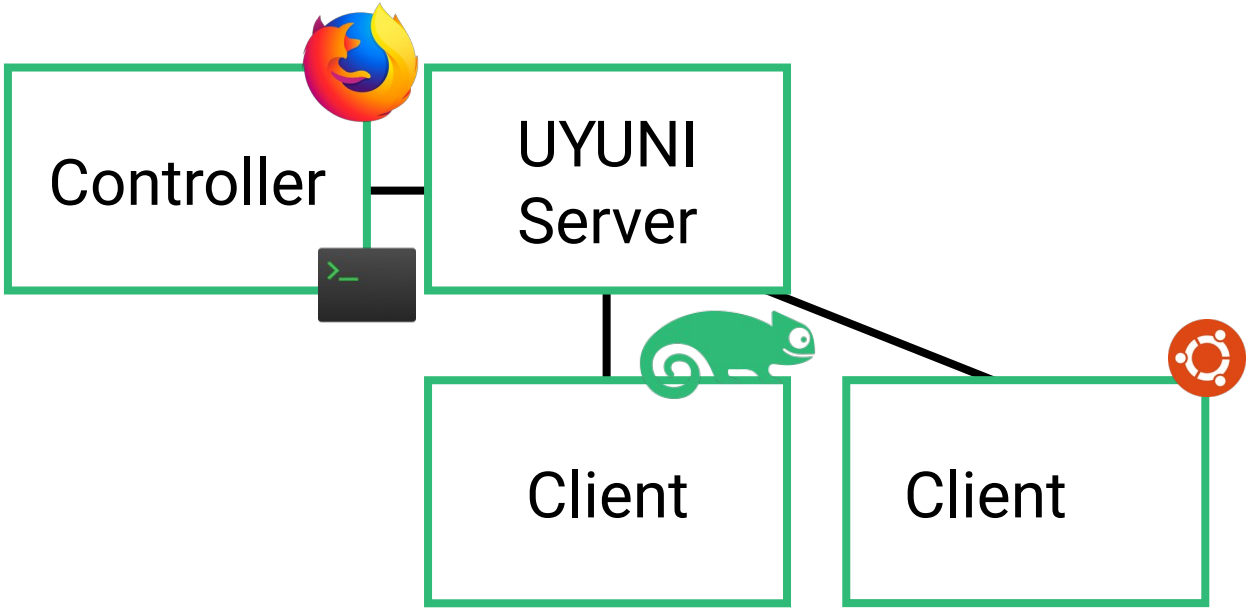

UYUNI
Server

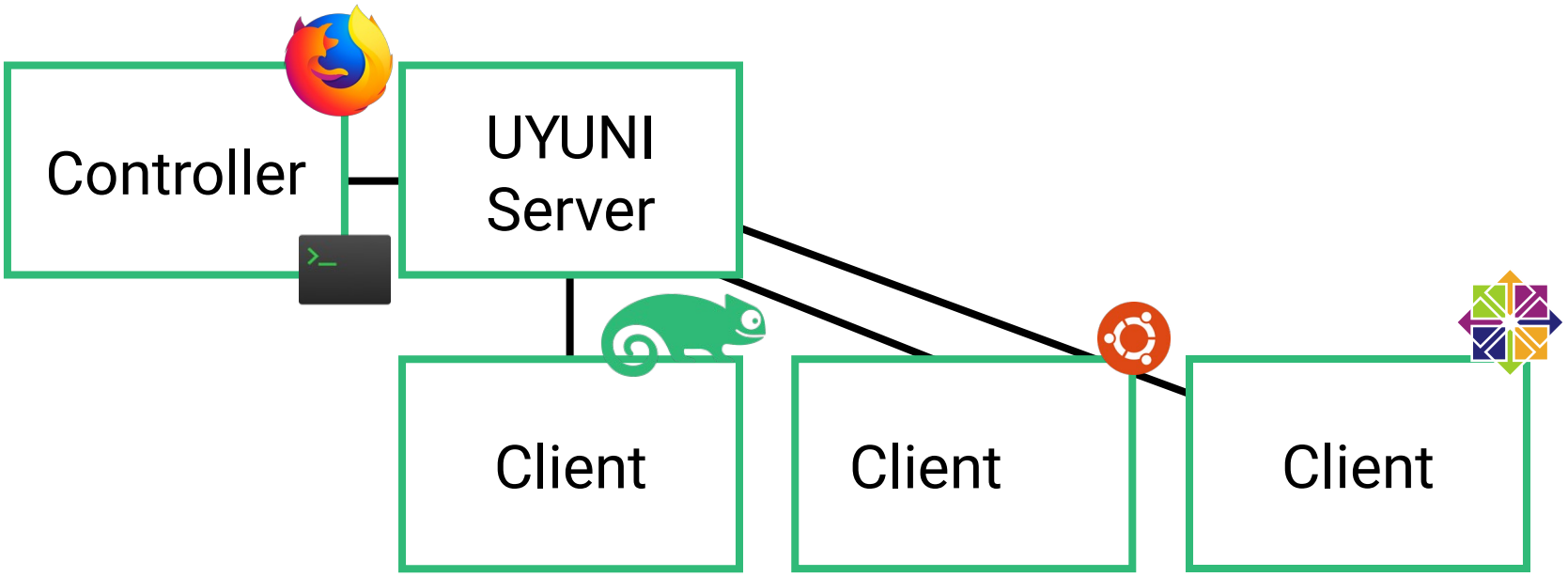

Controller
UYUNI
Server

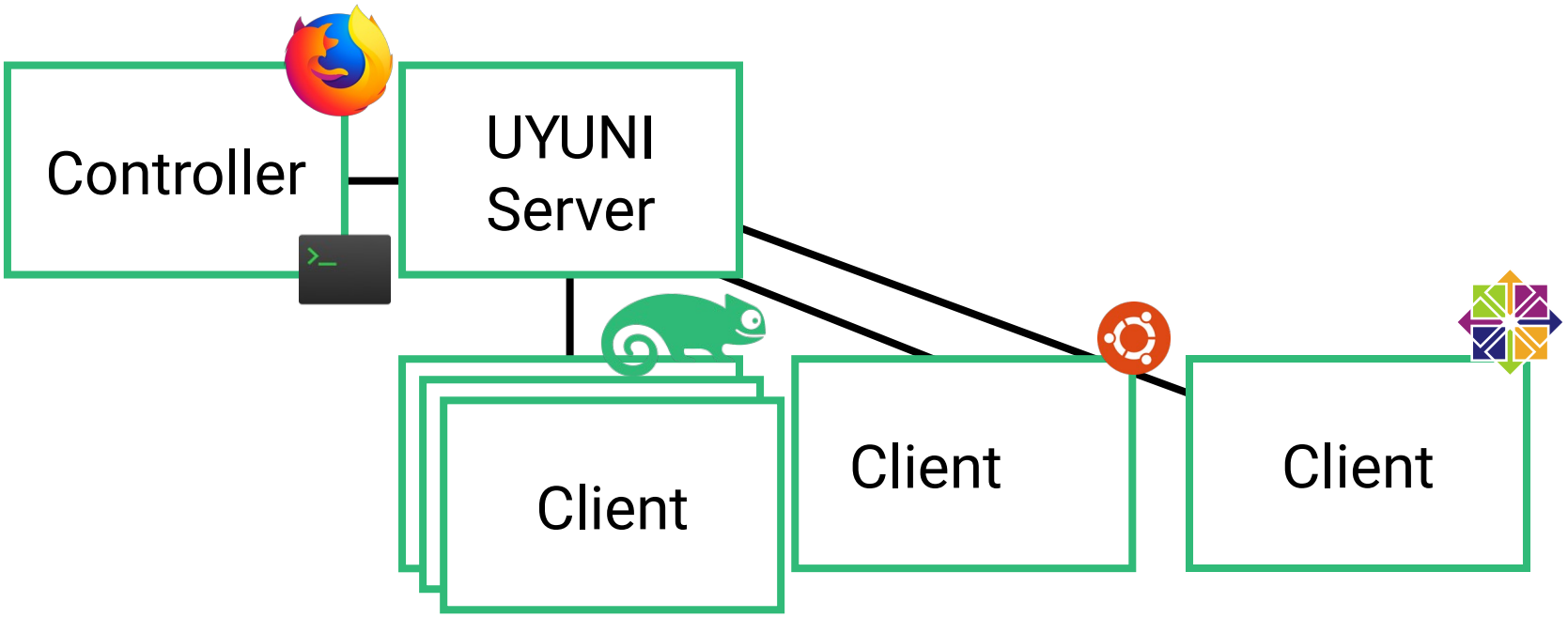

Controller
UYUNI
Server

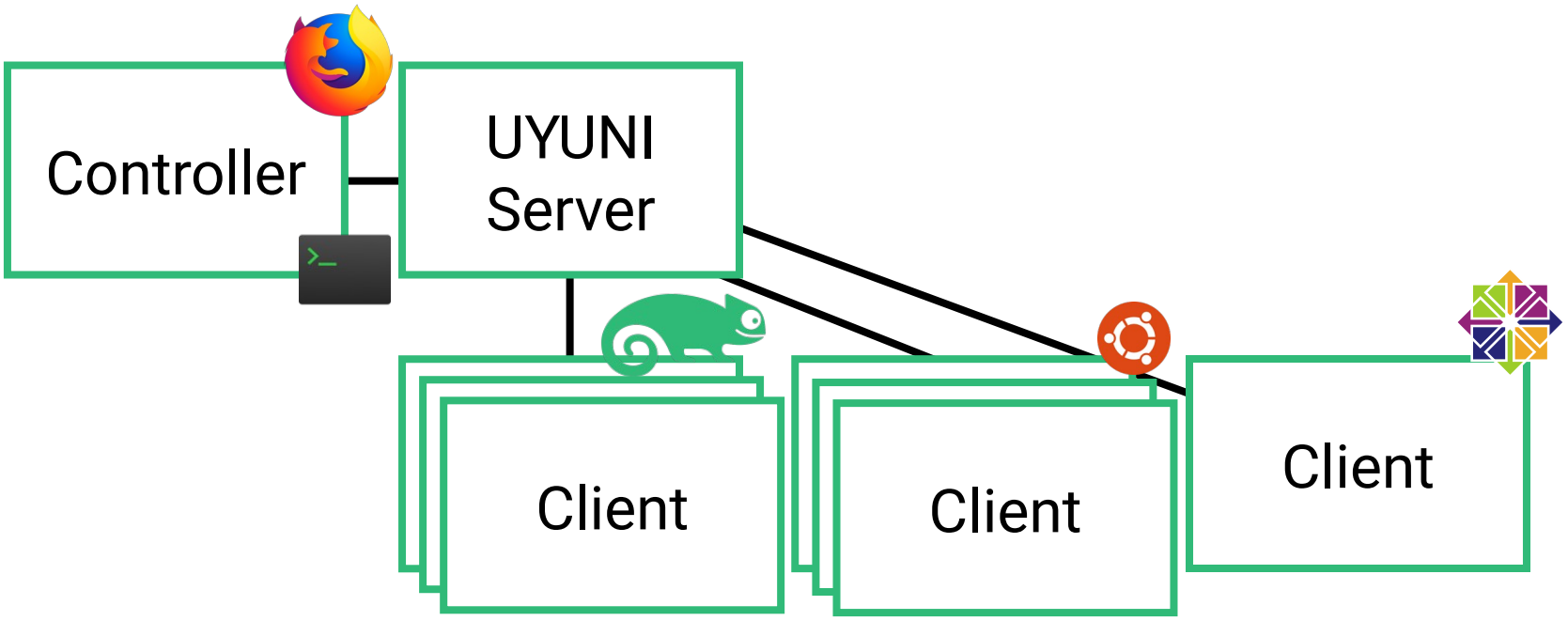

Controller
UYUNI
Server

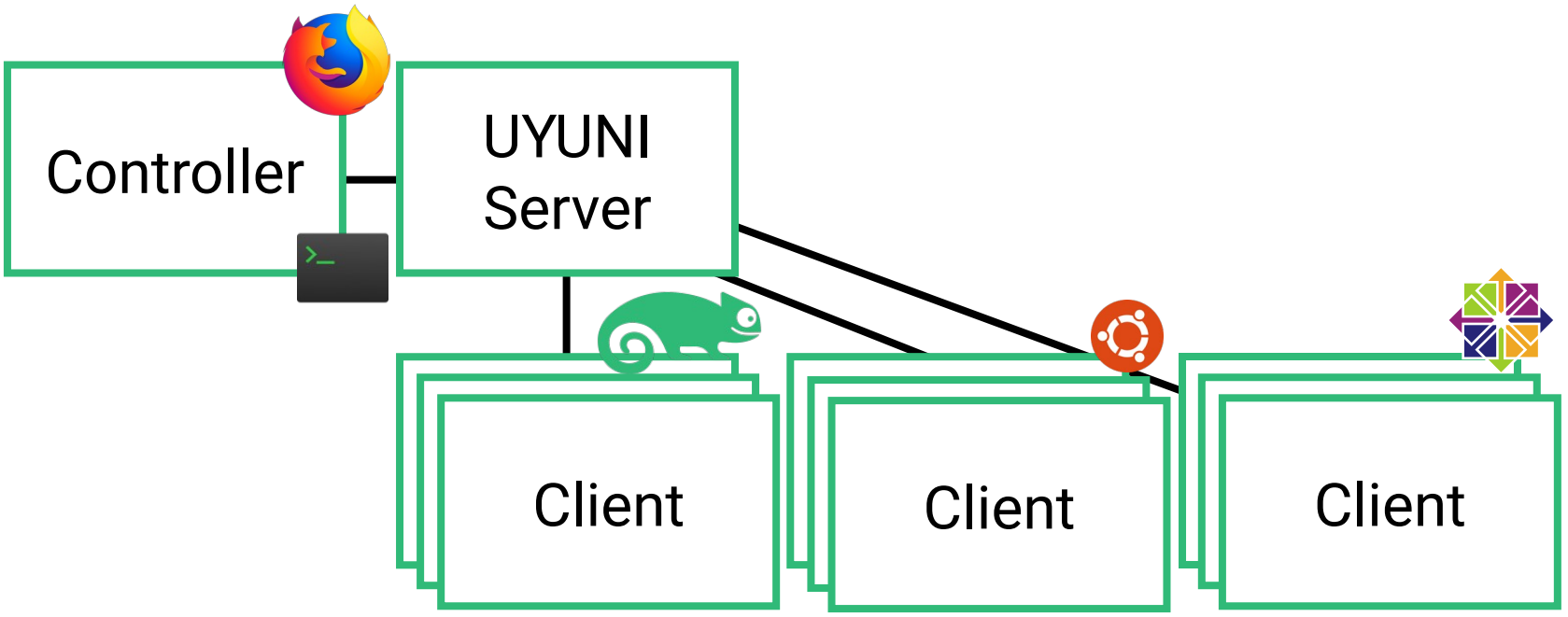

Controller
UYUNI
Server
Client

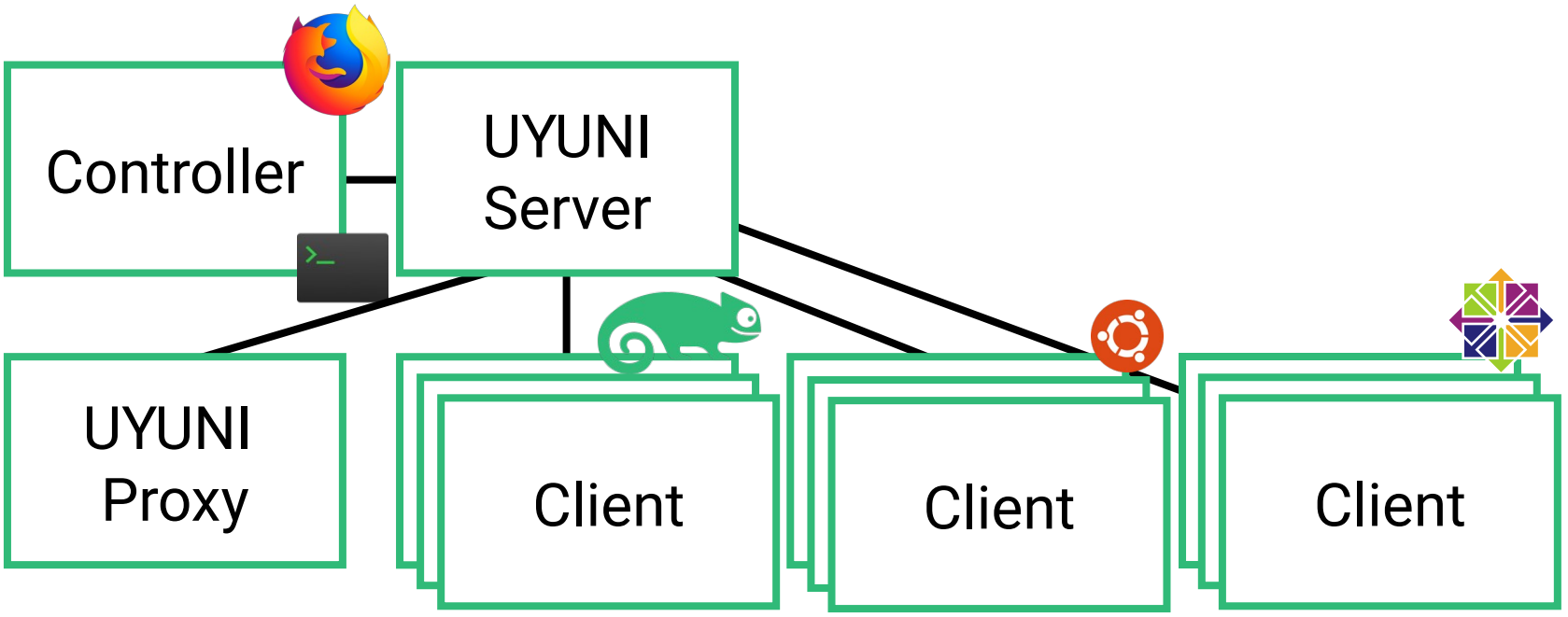

Controller
UYUNI
Server
Client
Client

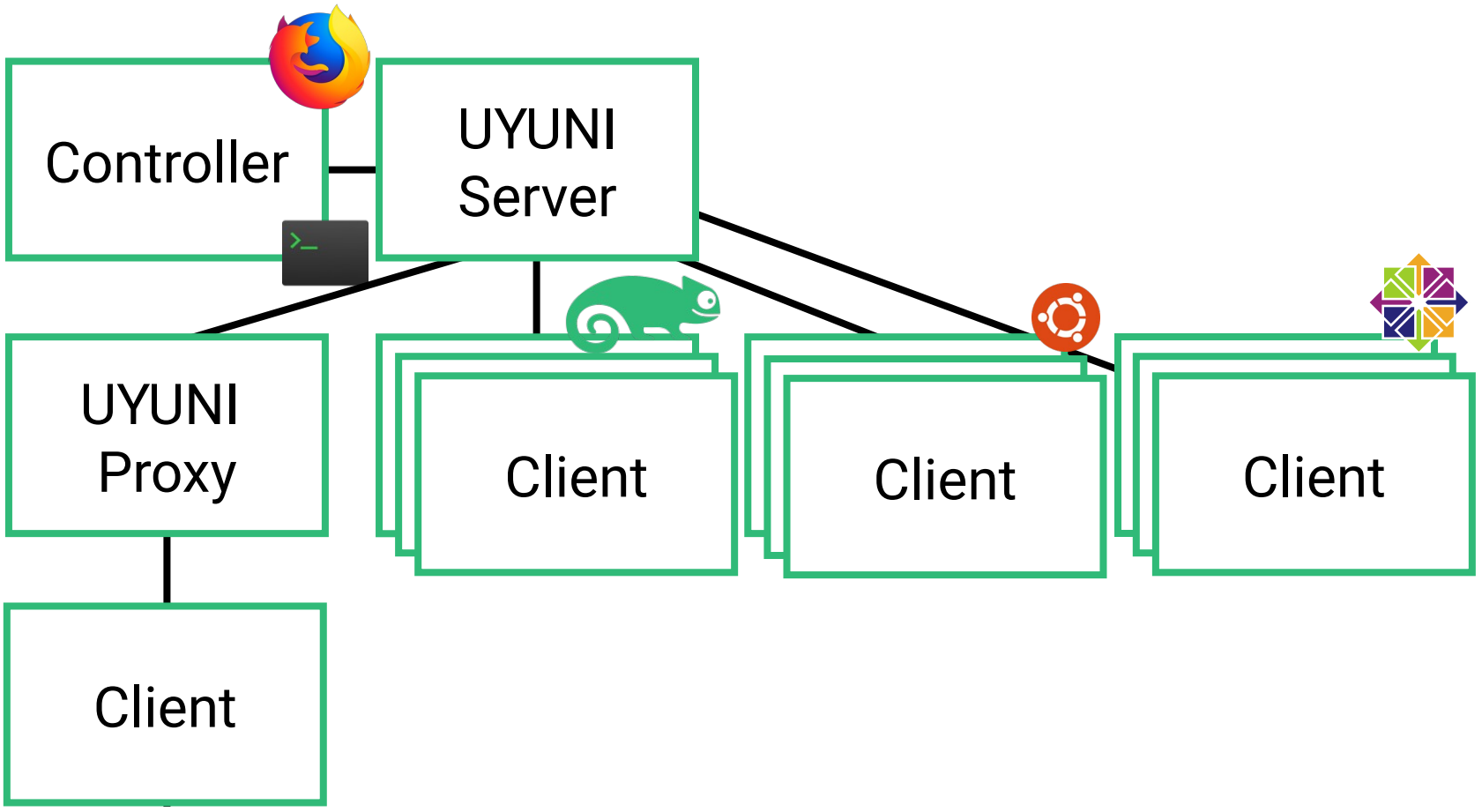

Controller
UYUNI
Server
Client
Client
Client

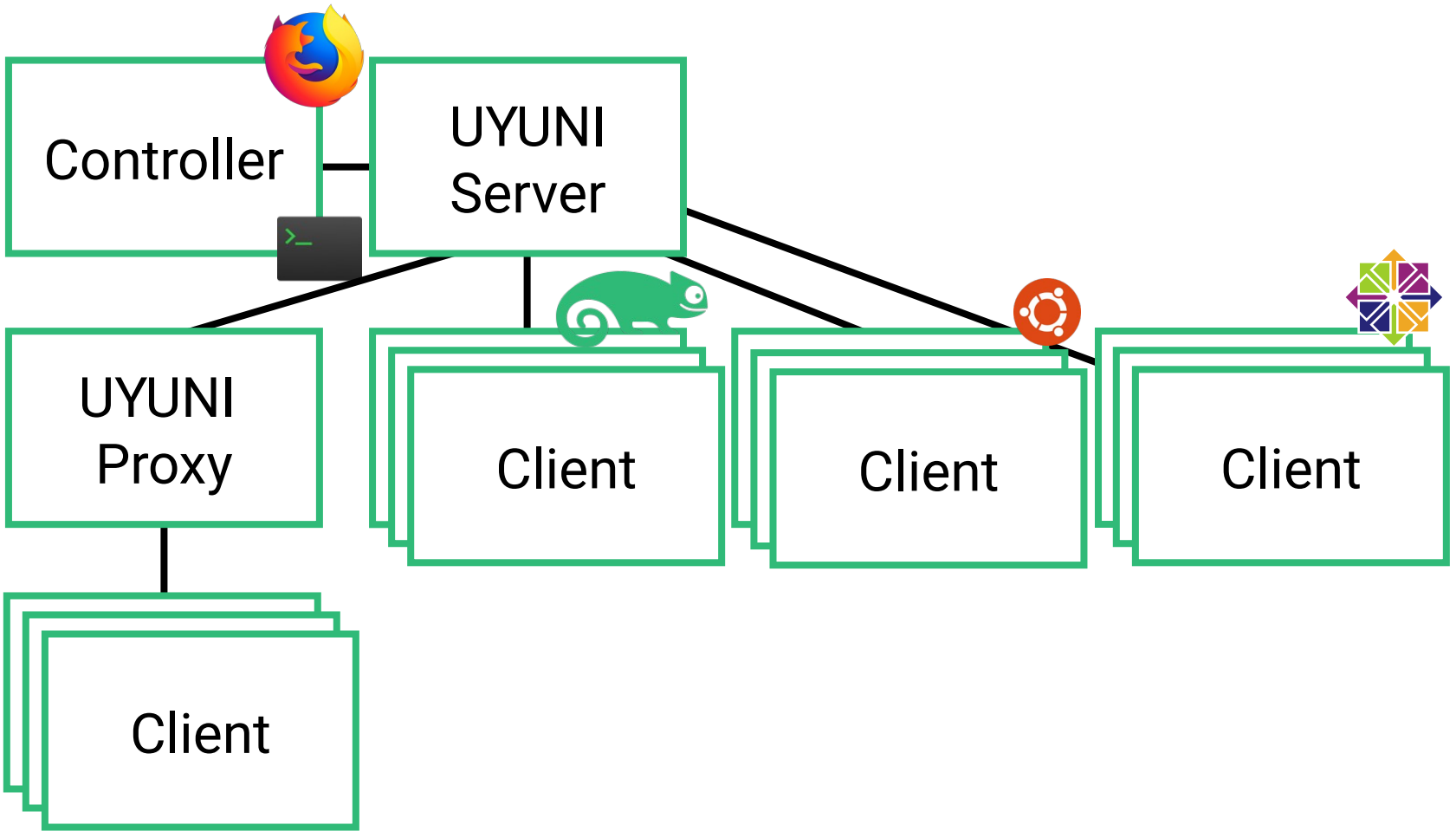

Controller
UYUNI
Server
Client
Client
Client

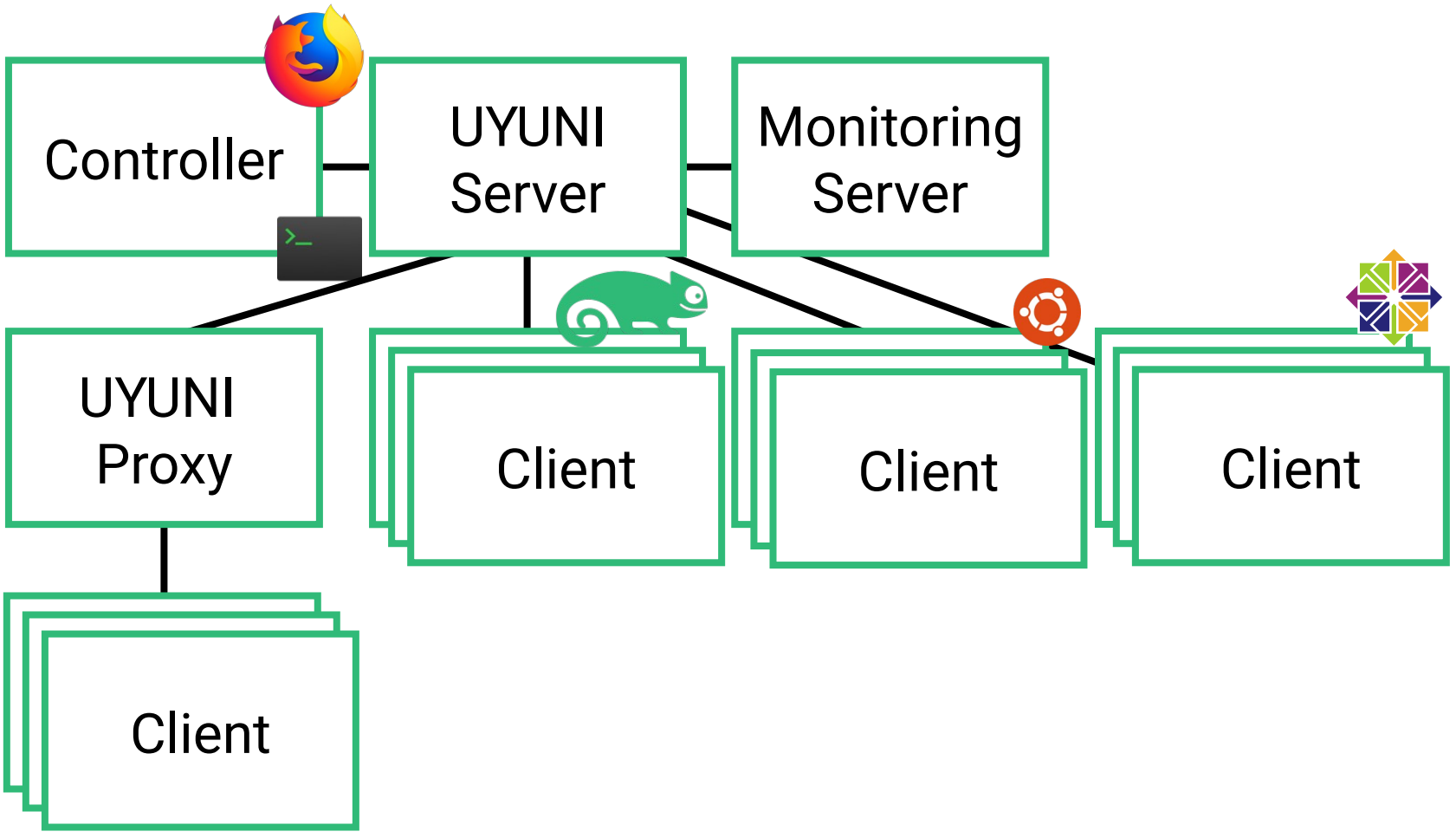

Controller
UYUNI
Server
Client
Client
Client

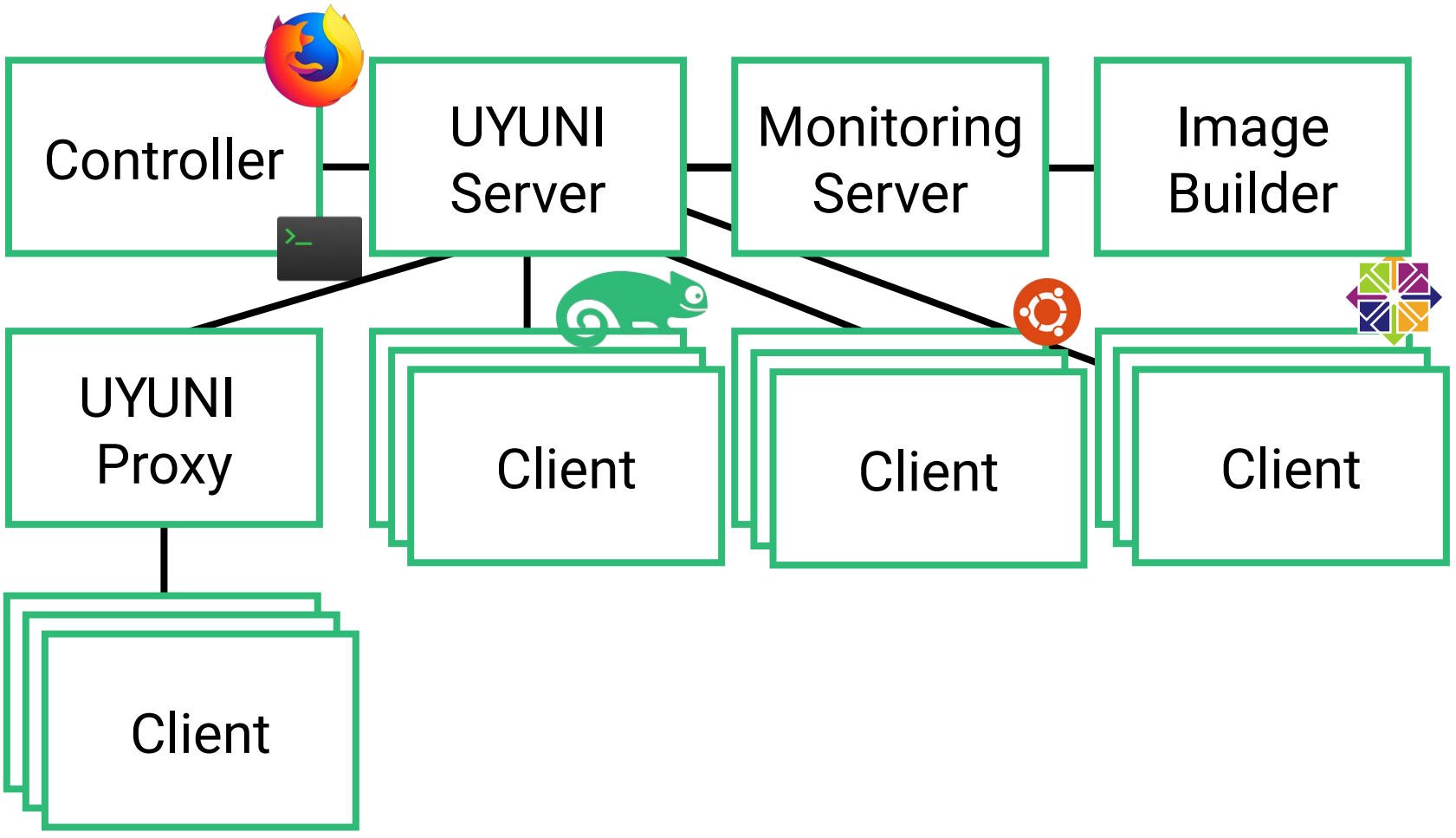

Controller
UYUNI
Server
Client
Client
Client

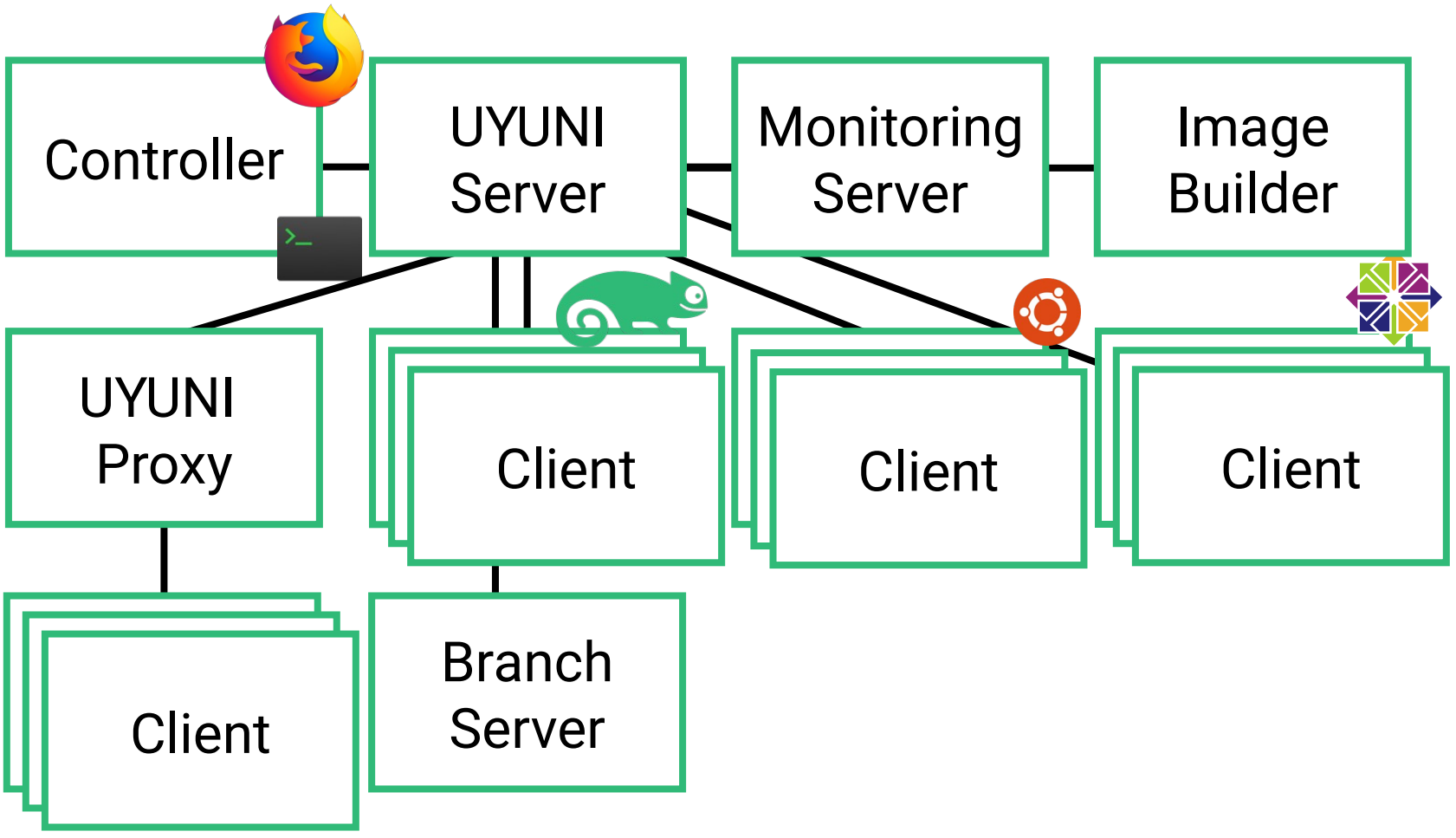

Controller
UYUNI
Server
UYUNI
Proxy
Client
Client
Client

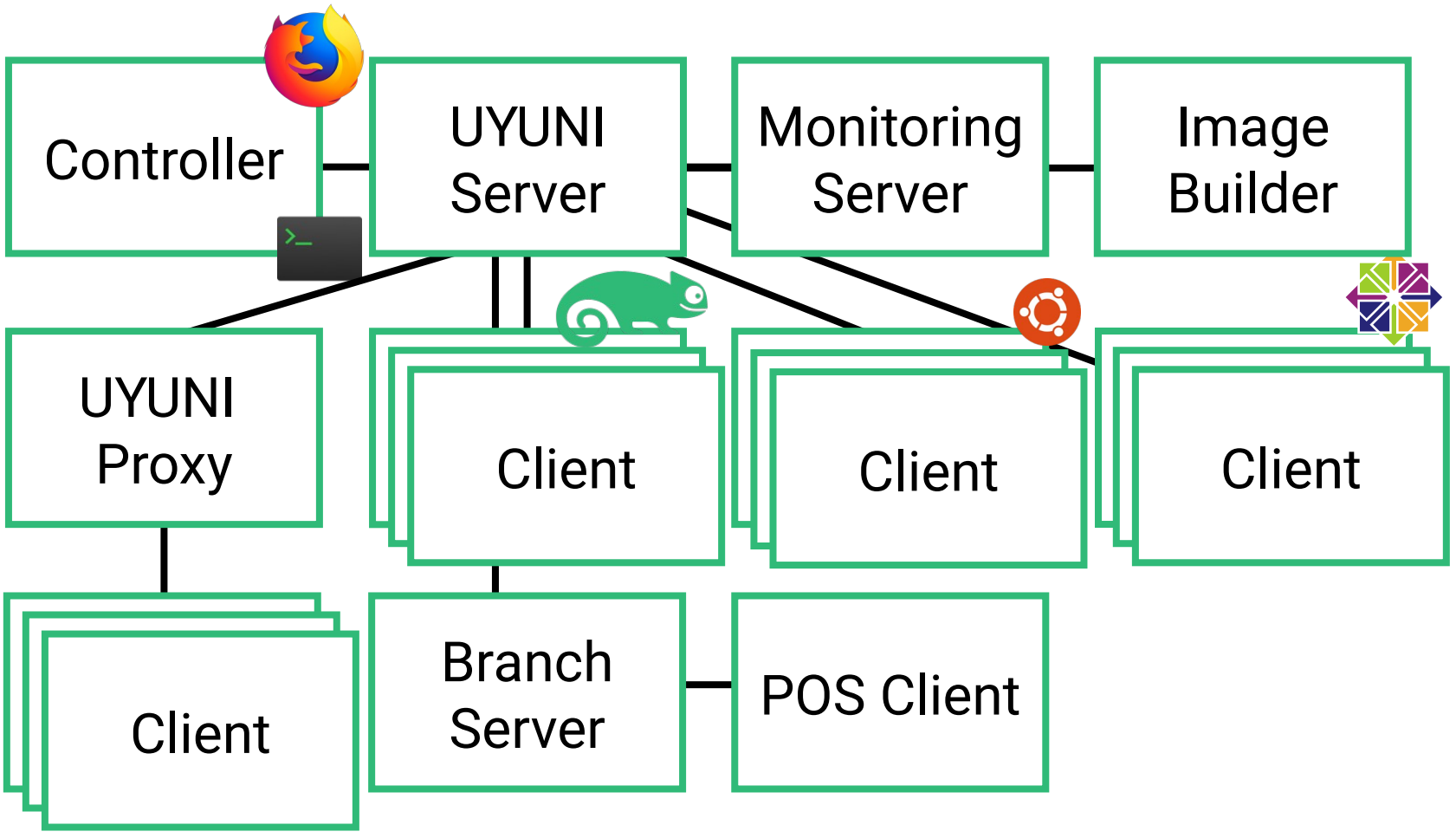

Controller
UYUNI
Server
UYUNI
Proxy
Client
Client
Client
Client

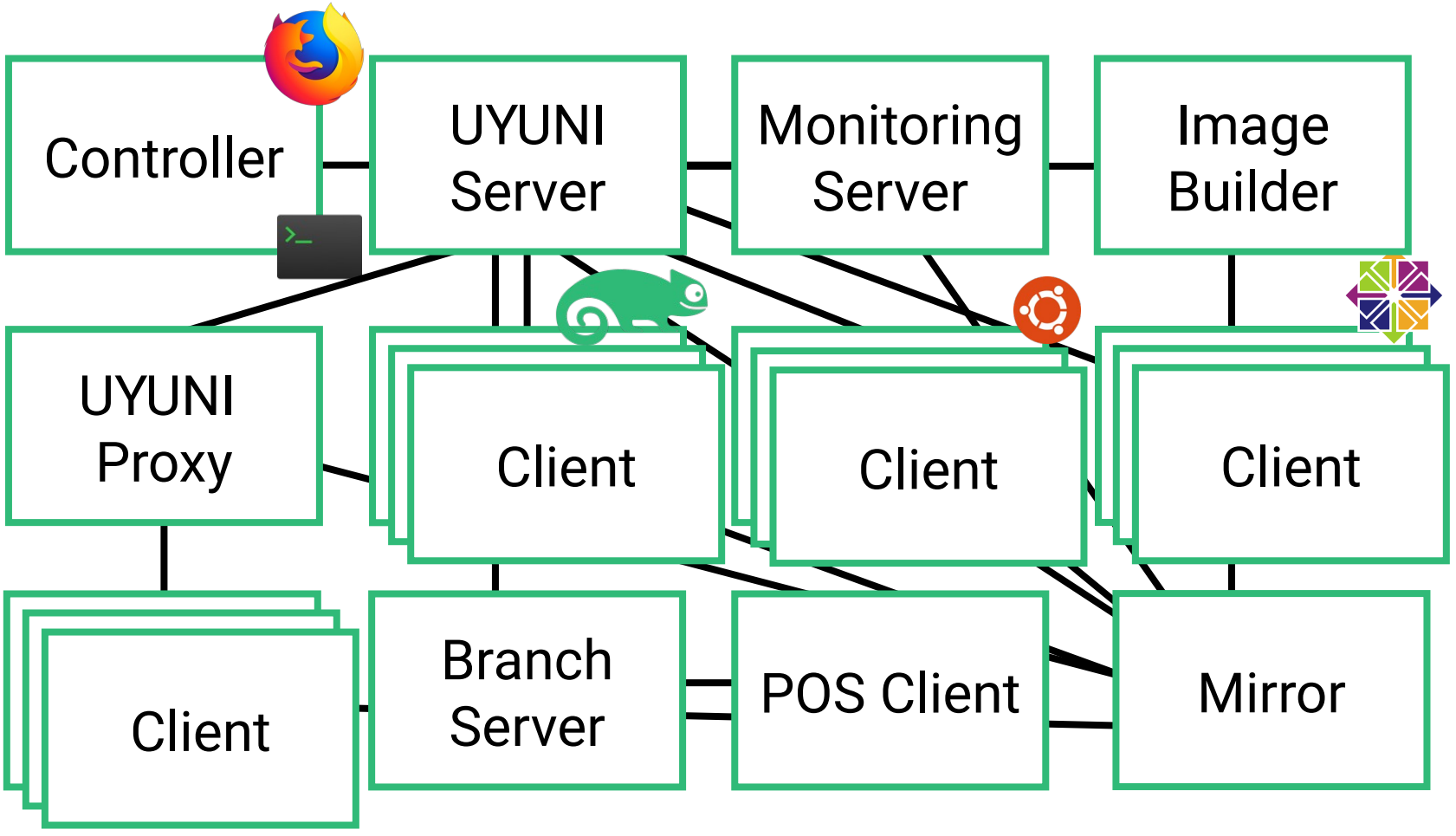

Controller
UYUNI
Server
UYUNI
Proxy
Client
Client
Client
Client

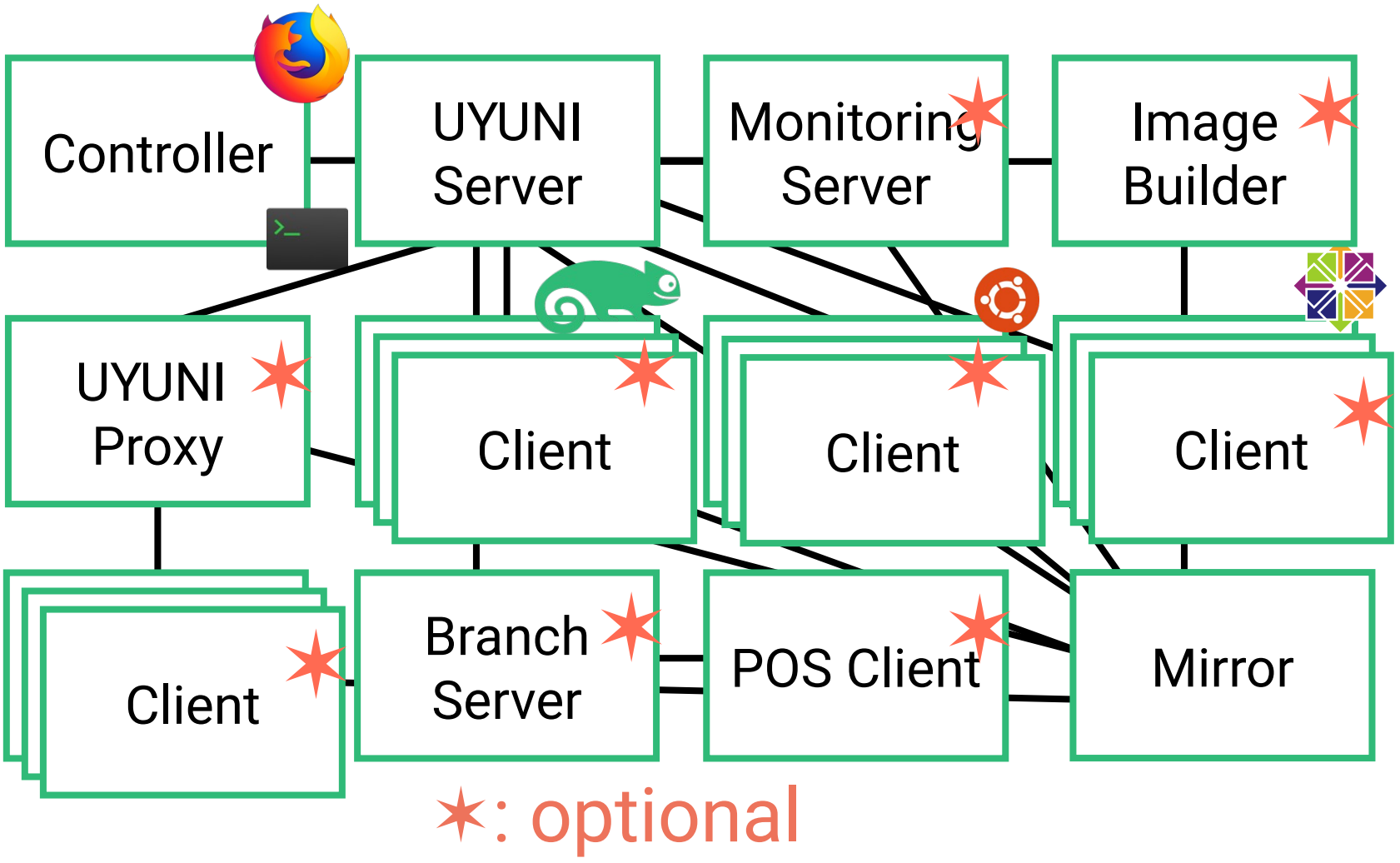

Controller
UYUNI Server
Monitoring Server
UYUNI Proxy
Client
Client
Client
Client

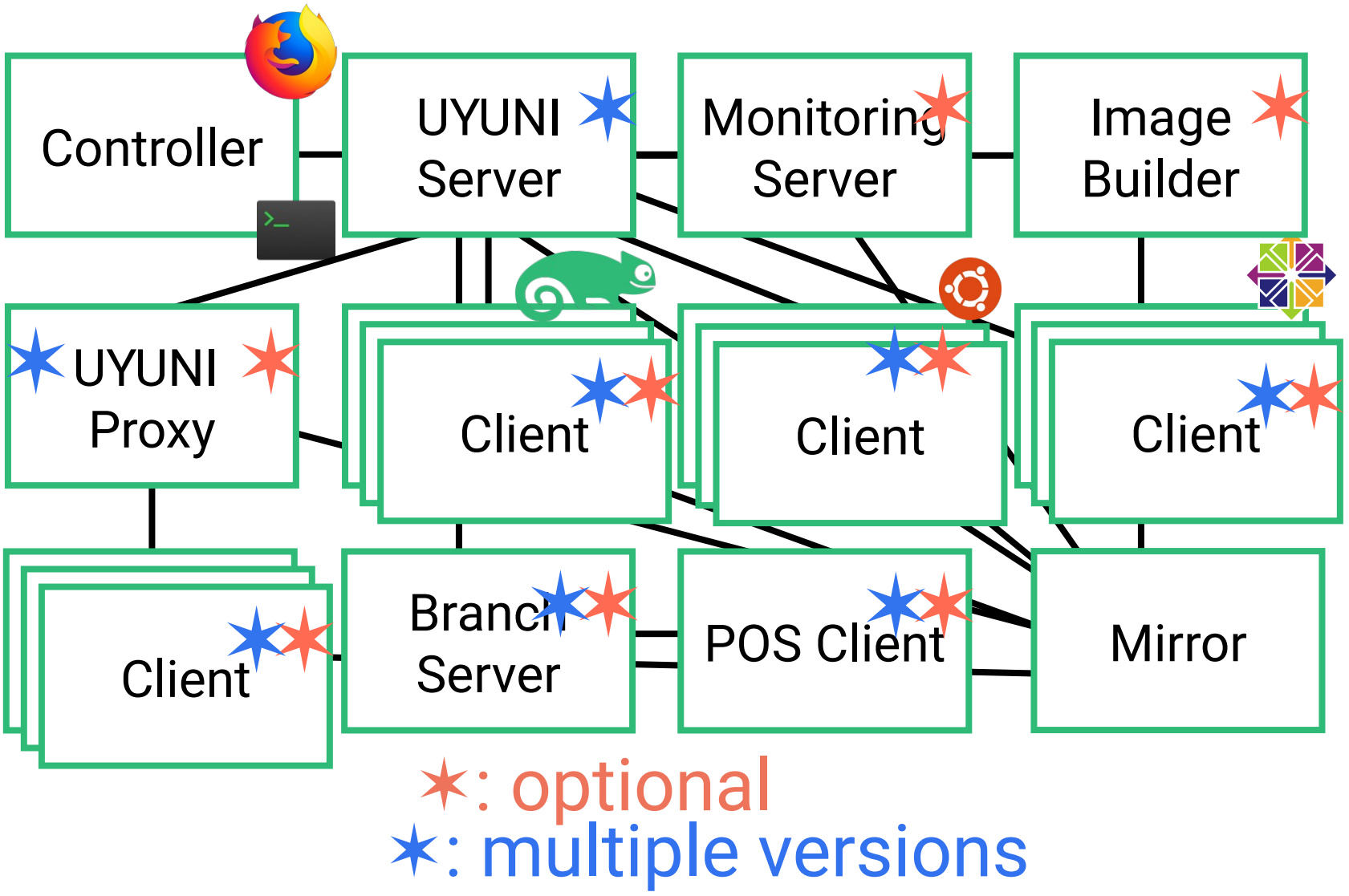

Controller
UYUNI Server
Monitoring Server
Image Builder
UYUNI Proxy
Client
Client
Client
Client

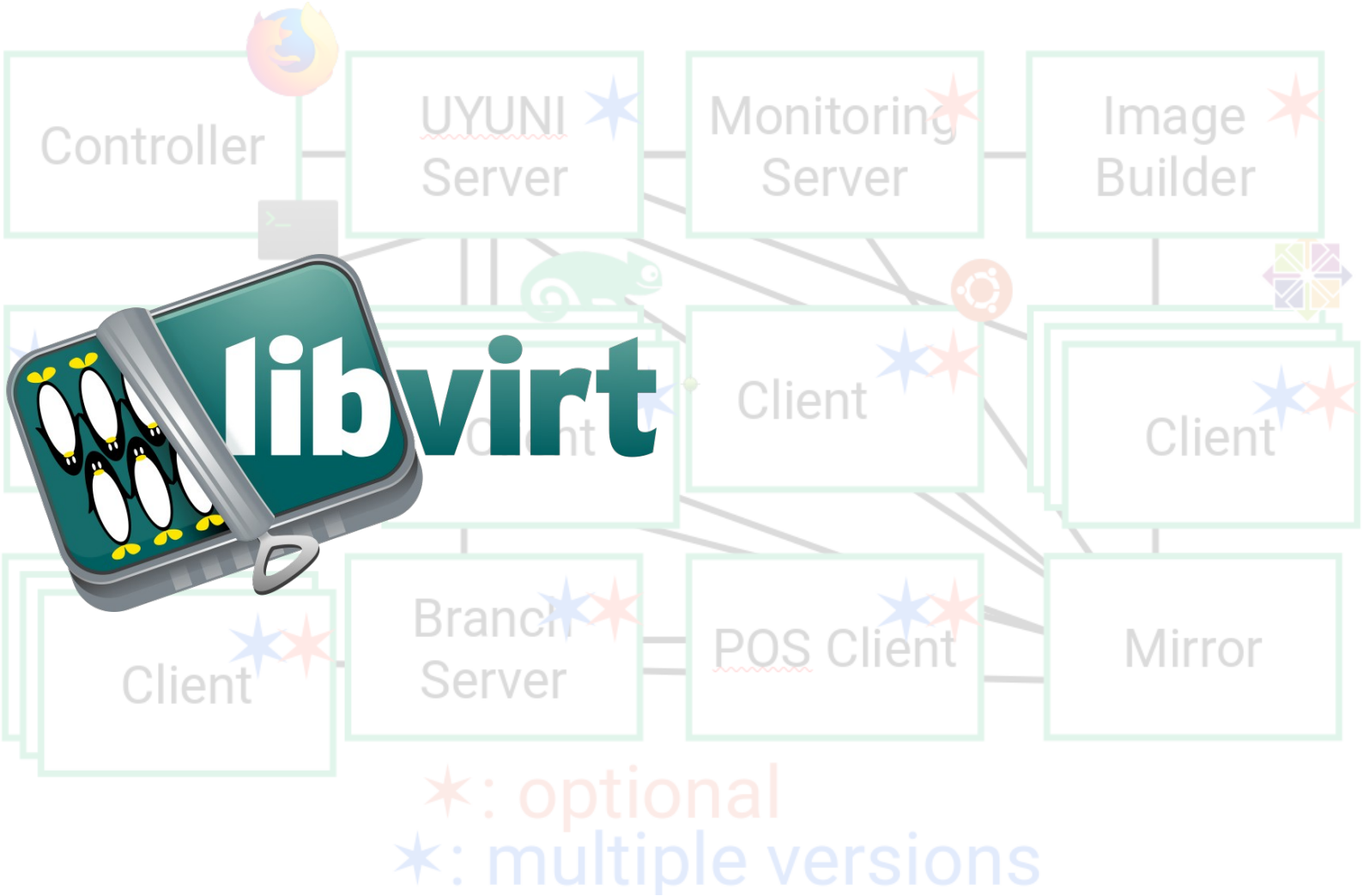

Controller
UYUNI Server
Monitoring Server
Image Builder
UYUNI Proxy
Client
Client
Client
Client
Branch Server


Controller
UYUNI Server
Monitoring Server
Image Builder
UYUNI Proxy
Client
Client
Client
Client
Branch Server
POS Client


Controller
UYUNI Server
Monitoring Server
Image Builder
UYUNI Proxy
Client
Client
Client
Client
Branch Server
POS Client
Mirror


Controller
UYUNI Server
Monitoring Server
Image Builder
UYUNI Proxy
Client
Client
Client
Client
Branch Server
POS Client
Mirror
✶: optional


Controller
UYUNI Server
Monitoring Server
Image Builder
UYUNI Proxy
Client
Client
Client
Client
Branch Server
POS Client
Mirror
: optional
: multiple versions


Controller
UYUNI Server
Monitoring Server
Image Builder
libvirt
Client
Client
Client
Client
Branch Server
POS Client
Mirror
✶: optional
✶: multiple versions


Controller
UYUNI Server
Monitoring Server
Image Builder
Client
Client
Client
Branch Server
POS Client
Mirror
libvirt
aws
✶: optional
✶: multiple versions

Test-suite Deployments

- >4 versions, 1000 tests, tens of runs per day

Test-suite Deployments

- >4 versions, 1000 tests, tens of runs per day
- >240 VM deployments per day
- 2 physical locations + AWS

Manual deployment is not an option.

- Infrastructure as Code (IaC)

- Infrastructure as Code (IaC)

- Configuration as Code (CaC)

UYUNI

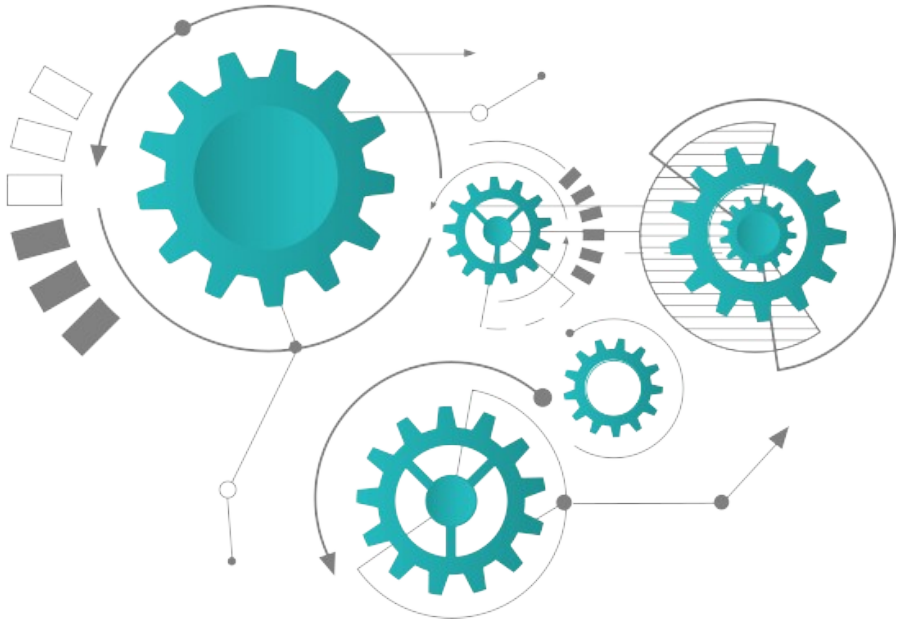
- Infrastructure as Code (IaC)

+

- Configuration as Code (CaC)

Agenda

- Uyuni Project
- Uyuni deployment
- Sumaform
  - Architecture
  - Modules
  - Demo

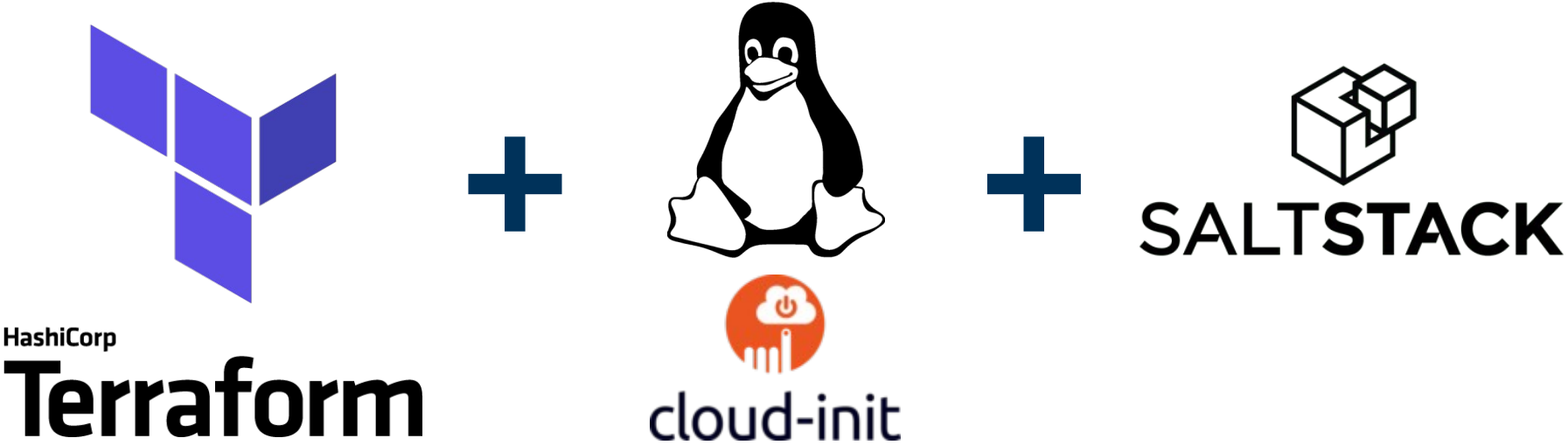


What is?

“Sumaform provides UYUNI-specific Terraform modules that leverage OS base images and Salt states to deliver a complete solution: from VM creation to installation of the product to configuration.”

** Silvio Moioli, SUSE Manager Development Team*

Architecture



Terraform

- Infrastructure as Code (IaC)
  - Deploy virtual machines
  - Manage resources life-cycle
- Domain specific modules
- Provisioning
  - Copy salt resources and apply state

Terraform Modules

Provider Independent Modules

- Notion of backend independent modules
  - server
  - proxy
  - minon
  - ...

Provider Specific Modules

- Two modules for each supported provider
  - Base
  - Host

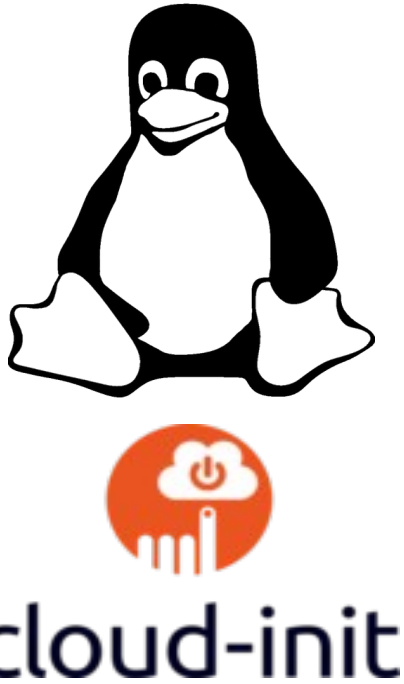
Supported providers (backends)

* run provisioning on existing machines (null provider)

OS image + cloud init

- Public JeOS images

- Cloud init: bring machine to “known base”
  - Install "salt-minion" package
  - Allow ssh login



Salt (CaC)

- Salt states for each machine role

- Parameters for customization
  - auto-register (on server)
  - additional repositories

Code example

```
provider "libvirt" {
  uri = "qemu:///system"
}

module "base" {
  source          = "./modules/base"

  cc_username = "..."
  cc_password  = "..."

  name_prefix = "uyuni-"
  domain         = "tf.local"
  images         = ["opensuse152"]
  use_avahi     = true
}
```

```
module "server" {
  source                = "./modules/server"
  base_configuration = module.base.configuration
  product_version      = "uyuni-released"
  name                    = "server"
}

module "min-opensuse" {
  source                   = "./modules/minion"
  base_configuration   = module.base.configuration
  product_version        = "uyuni-released"
  name                     = "opensuse"
  image                    = "opensuse152"
  server_configuration = module.server.configuration
}
```

Code example

```
provider "libvirt" {
  uri = "qemu:///system"
}

module "base" {
  source          = "./modules/base"

  cc_username = "..."
  cc_password  = "..."

  name_prefix = "uyuni-"
  domain         = "tf.local"
  images         = ["opensuse152"]
  use_avahi     = true
}
```

```
module "server" {
  source                = "./modules/server"
  base_configuration = module.base.configuration
  product_version       = "uyuni-released"
  name                  = "server"
}

module "min-opensuse" {
  source                = "./modules/minion"
  base_configuration    = module.base.configuration
  product_version       = "uyuni-released"
  name                  = "opensuse"
  image                 = "opensuse152"
  server_configuration = module.server.configuration
}
```


Demo time..


Q&A

/uyuni-project

/uyuni-project

/UyuniProject

Join Us at uyuni-project.org

Thank You