INFRASTRUCTURE AS CODE (alias IAC)

par Fabien Combernous

Présentation de l'auteur

ADULLACT

En quelques chiffres :

- 17 années d'expérience professionnelle dans le logiciel libre, l'architecture à base de libre.
- 1ère utilisation d'IAC en 2003.
- Gestion par IAC de 10 à 1000 machines
- Utilisation de Cfengine 2, Puppet 2, 3, 4, 5

De l’âge de pierre à IAC

A l’âge de pierre :

- $t=t_0$: machine requise
- $t=t_0+1j$: devis fournisseurs reçus
- $t=t_0+3j$: commande effectuée
- $t=t_0+10j$: machine réceptionnée
- $t=t_0+12j$: machine allumée
- $t=t_0+13j$: système configuré
- $t=t_0+14j$: ici commence la dérive

De l’âge de pierre à IAC

A l’arrivée de la virtualisation :

- $t=t_0$: machine requise
- $t=t_0+1j$: machine allumée
- $t=t_0+1j$: système configuré
- $t=t_0+2j$: ici commence la dérive

De l’âge de pierre à IAC

A l’arrivée de la virtualisation :

- $t=t_0$: machine requise
- $t=t_0+1j$: machine allumée
- $t=t_0+1j$: système configuré
- $t=t_0+2j$: ici commence la dérive

Multiplication des machines virtuelles.

Avant IAC, c’est quoi ?

Face à cette multiplication des machines (virtuelles), pour gérer une infrastructure j’ai vu et je vois :

- Beaucoup de configuration manuelle.
- Pas toujours des scripts.
- Souvent des images systèmes de référence.

Avant IAC, c’est quoi ?

Face à cette multiplication des machines (virtuelles), pour gérer une infrastructure j’ai vu et je vois:

- Beaucoup de configuration manuelle.
- Pas toujours des scripts.
- Souvent des images systèmes de référence

Il est difficile pour les administrateurs :

- a. D’effectuer une administration efficiente.
- b. De reproduire les environnements à l’identique.
- c. De suivre les modifications sur le parc de machines.
- b+c. De proposer sereinement un plan de reprise d’activité.

Et IAC, c’est quoi ?

Parfois IAC est vue, ou présentée, comme une méthode de déploiement de scripts.

Mais c’est bien plus !

Infrastructure as code

Une définition :

Utilisation des méthodes de génie logiciel pour décrire la configuration désirée d'une infrastructure.

Infrastructure as code

Une définition :

Utilisation des méthodes de génie logiciel pour décrire la configuration désirée d'une infrastructure.

Par exemple :

```
$services = [ 'sshd', 'ntpd',]

$services.each | String $s | {
  service { $s :
    ensure => running,
    enable => true,
  }
}
```

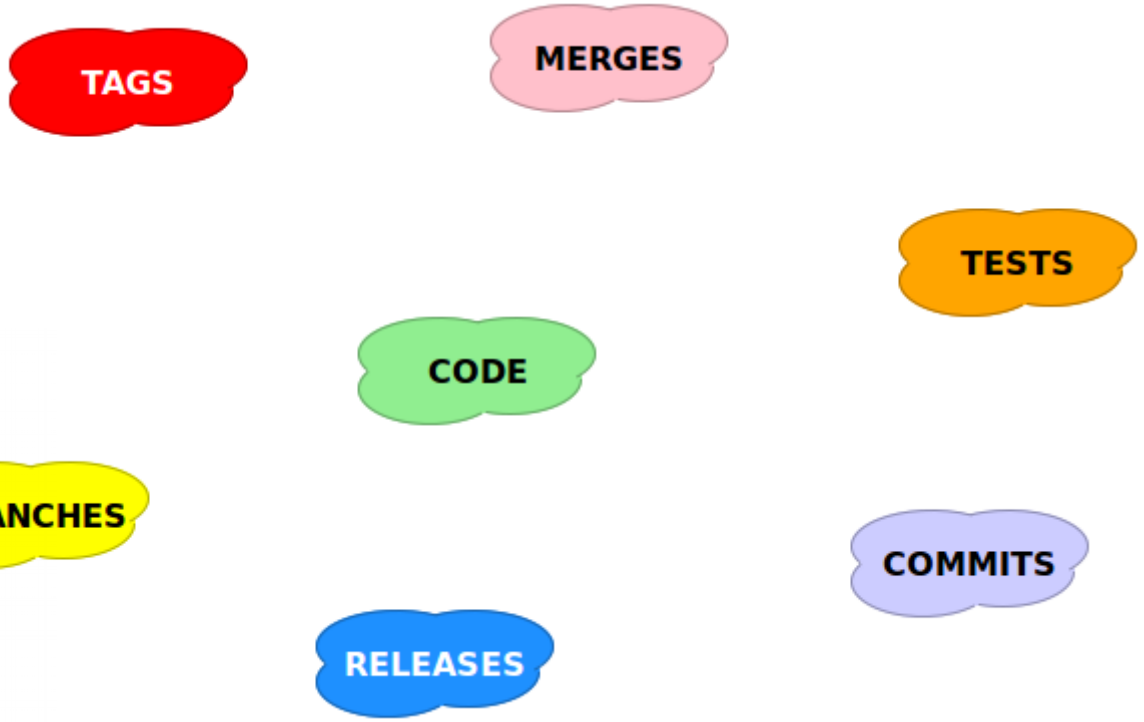
Et IAC, c’est quoi ?

IAC nous disons que c’est bien plus. Mais c’est quoi en plus :

1. Du code versionné.
2. De l’idempotence.
3. Des suites de tests.
4. De la vérification continue.
5. De l’abstraction.
6. De la documentation à jour.
7. De la traçabilité.

IAC, c’est du code versionné.

Nous allons donc décrire l’infrastructure au travers de ce que les développeurs affectionnent :



Territoire Numérique Libre

Et IAC, c’est quoi ?

IAC nous disons que c’est bien plus. Mais c’est quoi en plus :

1. Du code versionné. ✓
2. De l’idempotence.
3. Des suites de tests.
4. De la vérification continue.
5. De l’abstraction.
6. De la documentation à jour.
7. De la traçabilité.

IAC, idempotence

L’idempotence.

Propriété qui permet de mettre l’environnement cible toujours dans la même configuration.

Autrement dit, appliquer plusieurs fois une même procédure doit permettre d’arriver au même résultat.

IAC, idempotence

En bash, simple mais pas idempotent :

```
echo "127.0.0.1 localhost.localdomain" >> /etc/hosts
```

En bash, idempotent mais compliqué :

```
if (!grep -qP '^127\.0\.0\.1\s+localhost\.localdomain' /etc/hosts); then
  echo "127.0.0.1 localhost.localdomain" >> /etc/hosts
fi
```

En IAC ?

IAC, idempotence

En bash :

```
if (!grep -qP '^127\.0\.0\.1\s+localhost\.localdomain' /etc/hosts); then
  echo "127.0.0.1 localhost.localdomain" >> /etc/hosts
fi
```

En IAC ?

IAC, idempotence

En bash, idempotent mais compliqué :

```
if (!grep -qP '^127\.0\.0\.1\s+localhost\.localdomain' /etc/hosts); then
  echo "127.0.0.1 localhost.localdomain" >> /etc/hosts
fi
```

En IAC, je décris l'état final et non le chemin. Exemple d'opérateur convergeant avec puppet :

```
host { 'localhost.localdomain':
    ip => '127.0.0.1',
}
```

Et IAC, c’est quoi ?

IAC nous disons que c’est bien plus. Mais c’est quoi en plus :

1. Du code versionné. √
2. De l’idempotence. √
3. Des suites de tests.
4. De la vérification continue.
5. De l’abstraction.
6. De la documentation à jour.
7. De la traçabilité.

IAC, des tests et des tests

Le test du code produit.

Tests pour voir si le code produit suit les recommandations de bonne pratique.

```
$services = [ 'sshd', 'ntpd',]

$services.each | String $s | {
  service { $s :
    ensure => running,
    enable => true,
  }
}
```

IAC, des tests et des tests

Le test unitaire du code produit.

Test pour vérifier la logique du code, s'assurer qu'une fonctionnalité ne soit pas supprimée par inadvertance.

IAC, des tests et des tests

Le test unitaire du code produit.

Test pour vérifier la logique du code, s'assurer qu'une fonctionnalité ne soit pas supprimée par inadvertance.

```
it { is_expected.to contain_service('sshd').with ({
  :ensure => 'running',
  :enable => true,
}) }
```

IAC, des tests et des tests

Le test fonctionnel du résultat obtenu.

Test pour s’assurer qu’un code produise le résultat escompté.

IAC, des tests et des tests

Le test fonctionnel du résultat obtenu.

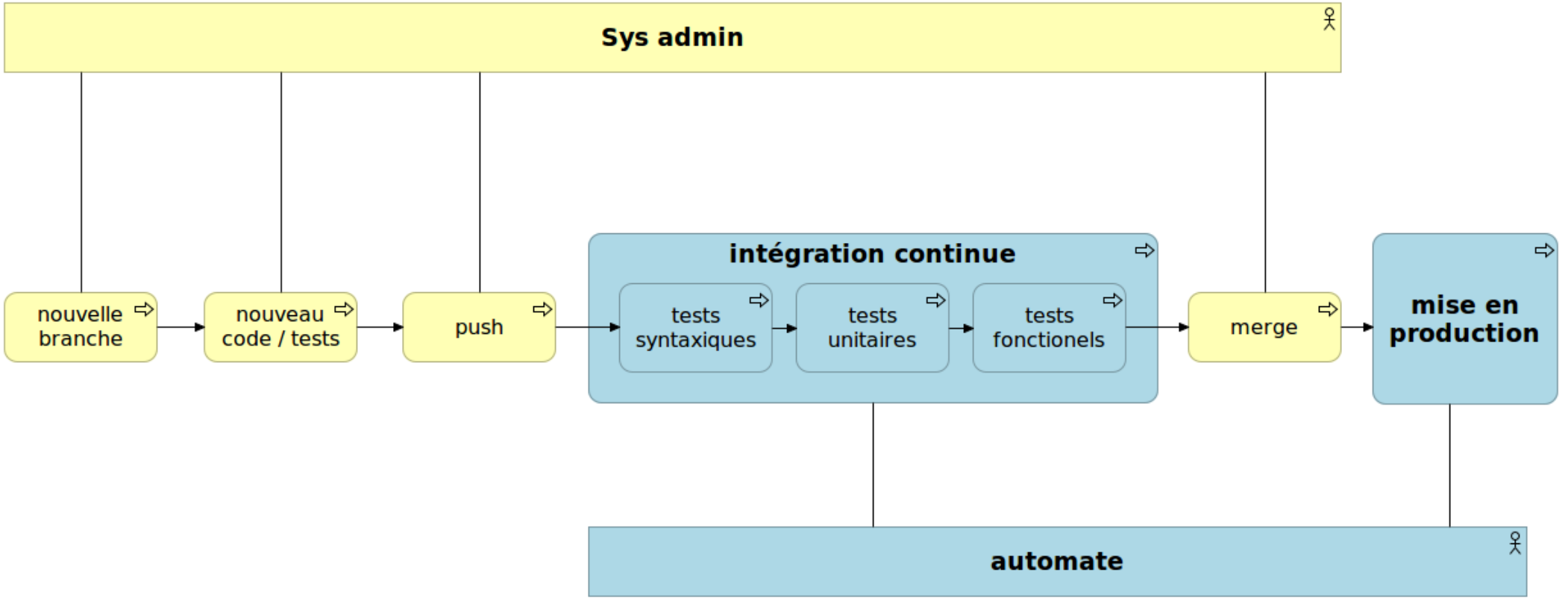
Test pour s'assurer qu'un code produise le résultat escompté.

```
describe file("/etc/ssh/sshd_config") do
  it { should be_file }
end

describe service("ssh") do
  it { should be_running }
  it { should be_enabled }
end
```

IAC, des tests et des tests

En résumé (formalisme TOGAF), que pouvons nous dire ?



Et IAC, c’est quoi ?

IAC nous disons que c’est bien plus. Mais c’est quoi en plus :

1. Du code versionné. √
2. De l’idempotence. √
3. Des suites de tests. √
4. De la vérification continue.
5. De l’abstraction
6. De la documentation à jour.
7. De la traçabilité.

IAC, vérification continue

Grâce à l'idempotence, il est possible de rejouer le code encore et encore.

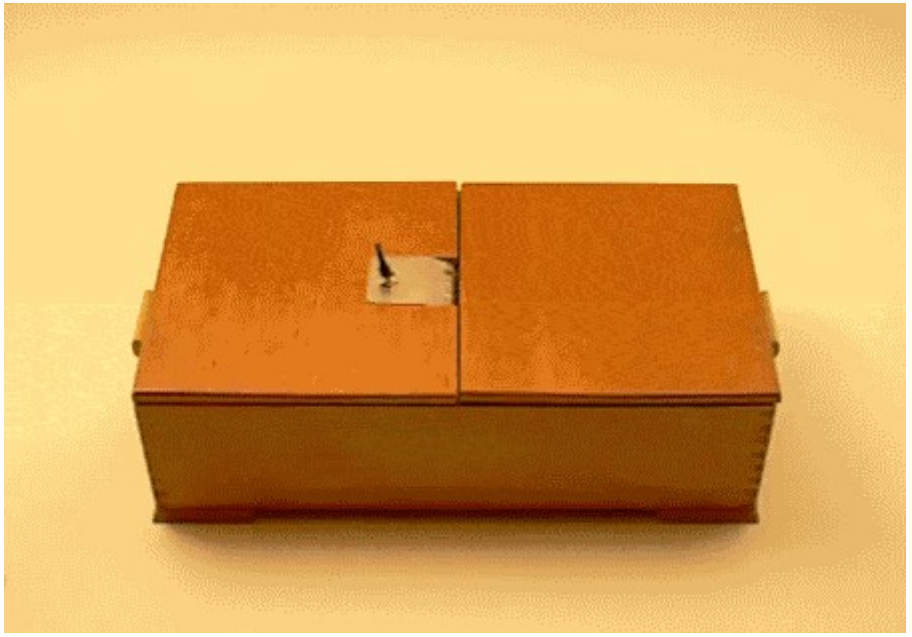
Périodiquement, il est vérifié que le système est dans l'état désiré.

Si ce n'est pas le cas :

- Produit un rapport : état avant / état après
- Essaie de remettre dans l'état désiré.

IAC, vérification continue

Grâce à l'idempotence, il est possible de rejouer le code encore et encore.



Et IAC, c’est quoi ?

IAC nous disons que c’est bien plus. Mais c’est quoi en plus :

1. Du code versionné. √
2. De l’idempotence. √
3. Des suites de tests. √
4. De la vérification continue. √
5. De l’abstraction
6. De la documentation à jour.
7. De la traçabilité.

IAC, une couche d'abstraction

Exemple pour configurer un reverse proxy :

```
app_reverseproxy::vhosts:
  http_listes_adullact_org:
    servername: listes.adullact.org
    port: 80
    ip: 5.39.26.92
    redirect_status: permanent
    redirect_dest: 'https://listes.adullact.org/'
  https_listes_adullact_org:
    servername: listes.adullact.org
    ip: 5.39.26.92
    port: 443
    ssl: true
    ssl_cert: '/etc/letsencrypt/live/listes.adullact.org/cert.pem'
    ssl_key: '/etc/letsencrypt/live/listes.adullact.org/privkey.pem'
    ssl_proxyengine: true
    ssl_proxy_verify: none
    proxy_preserve_host: false
    proxy_pass:
      - { path: '/', url: 'http://listes.lan100/' }
```

Et si le reverse proxy doit servir une autre machine ? C'est compliqué ?

IAC, une couche d'abstraction

```
app_reverseproxy::vhosts:
  http_listes_adullact_org:
    servername: listes.adullact.org
    port: 80
    ip: 5.39.26.92
    redirect_status: permanent
    redirect_dest: 'https://listes.adullact.org/'
  https_listes_adullact_org:
    servername: listes.adullact.org
    ip: 5.39.26.92
    port: 443
    ssl: true
    ssl_cert: '/etc/letsencrypt/live/listes.adullact.org/cert.pem'
    ssl_key: '/etc/letsencrypt/live/listes.adullact.org/privkey.pem'
    ssl_proxyengine: true
    ssl_proxy_verify: none
    proxy_preserve_host: false
    proxy_pass:
      - { path: '/', url: 'http://listes.lan100/' }
  http_sugar_adullact_org:
    servername: sugar.adullact.org
    port: 80
    ip: 5.39.26.92
    redirect_status: permanent
    redirect_dest: 'https://sugar.adullact.org/'
  https_sugar_adullact_org:
    servername: sugar.adullact.org
    ip: 5.39.26.92
    port: 443
    ssl: true
    ssl_cert: '/etc/letsencrypt/live/sugar.adullact.org/cert.pem'
    ssl_key: '/etc/letsencrypt/live/sugar.adullact.org/privkey.pem'
    ssl_proxyengine: true
    ssl_proxy_verify: none
    proxy_preserve_host: false
    proxy_pass:
      - { path: '/', url: 'http://sugar.lan200/', params: { Keepalive: 'On', timeout: '60', }, }
```

Et IAC, c’est quoi ?

IAC nous disons que c’est bien plus. Mais c’est quoi en plus :

1. Du code versionné. √
2. De l’idempotence. √
3. Des suites de tests. √
4. De la vérification continue. √
5. De l’abstraction. √
6. De la documentation à jour.
7. De la traçabilité.

IAC, de la documentation à jour.

Ce qui est configuré sur les machines est décrit dans des documents.

Ces documents sont actifs puisque produisant les actions sur l'infrastructure, donc forcément à jour.

IAC, de la documentation à jour.

Ce qui est configuré sur les machines est décrit dans des documents.

Ces documents sont actifs puisque produisant les actions sur l'infrastructure, donc forcément à jour.

Q : Ai-je bien un redirect http vers https pour les listes.adullact.org ?

IAC, de la documentation à jour.

Q : Ai-je bien un redirect http vers https pour les listes.adullact.org ?

R

```
app_reverseproxy::vhosts:
  http_listes_adullact_org:
    servername: listes.adullact.org
    port: 80
    ip: 5.39.26.92
    redirect_status: permanent
    redirect_dest: 'https://listes.adullact.org/'
  https_listes_adullact_org:
    servername: listes.adullact.org
    ip: 5.39.26.92
    port: 443
    ssl: true
    ssl_cert: '/etc/letsencrypt/live/listes.adullact.org/cert.pem'
    ssl_key: '/etc/letsencrypt/live/listes.adullact.org/privkey.pem'
    ssl_proxyengine: true
    ssl_proxy_verify: none
    proxy_preserve_host: false
    proxy_pass:
      - { path: '/', url: 'http://listes.lan100/' }
```

Et IAC, c’est quoi ?

IAC nous disons que c’est bien plus. Mais c’est quoi en plus :

1. Du code versionné. √
2. De l’idempotence. √
3. Des suites de tests. √
4. De la vérification continue. √
5. De l’abstraction. √
6. De la documentation à jour. √
7. De la traçabilité.

IAC, traçabilité.

Traçabilité, auditabilité, informatique cachée réduite.

A tout moment, il est possible de répondre aux questions :

- Qui ?
- Quoi ?
- Où ?
- Quand ?
- Comment ?
- Pourquoi ?

Et IAC, c’est quoi ?

IAC nous disons que c’est bien plus. Mais c’est quoi en plus :

1. Du code versionné. √
2. De l’idempotence. √
3. Des suites de tests. √
4. De la vérification continue. √
5. De l’abstraction. √
6. De la documentation à jour. √
7. De la traçabilité. √

IAC, compagnon de devops

IAC est nécessaire à une mise en œuvre de devops efficace.

- Provisionner l'infrastructure (serveurs, réseaux, stockage).
- Configurer l'infrastructure (IAC, parfois DevOps)
- Déployer l'application (DevOps)

IAC, les outils.

Cfengine premier logiciel (libre) de la famille des gestionnaires de configuration.

Initié par Mark Burgess pendant son postdoc dans l'une des Universités d'Oslo.

1993 cfengine version 1

1995 Opérateur convergeant

1998 Immunologie des systèmes

2003 Théorie de la promesse

IAC, les outils libres

Outil	Langage	Naissance	Charge	Sécurité	Date w/x
cfengine	C	1993	1	3	2009
puppet	Ruby/java	2005	2	2	2012
chef	Ruby/erlang	2009	2	2	2014
ansible	python	2012	2	1	2014
saltstack	python	2011	2	?	?
rudder	Scala/C	« 2010 »	2	2	« 2012 »
powershell	.net	« 2006 »	3	?	« 2016 »

Charge système induite : de plus faible à plus forte 1 2 3 (avis personnel argumenté)
Sécurité : de mois sécurisé à plus sécurisé 1 2 3 (avis personnel argumenté)
Date w/x : année à partir de laquelle l'outil adresse les systèmes Unix, Linux et Windoze

Un seul outil dans la colonne « Outils », mais il faut penser à une chaîne d'outils derrière la tête de pont indiquée.

Que peut-on gérer avec IAC ?

Quels types d'équipements et services est-il possible de gérer la configuration avec IAC ?

- Des serveurs physiques ou virtuels.
- Des serveurs de base de données
- Des contrôleurs de domaine active directory
- Des hyperviseurs
- Des switchs
- Du stockage
- Etc ...

IAC, des freins humains.

Habitudes modifiées.

Le passage à IAC représente un changement profond qui va se heurter aux habitudes. Parfois même, IAC vient en conflit de l'idée même que le sys admin se fait de son métier.

Réaction possible :

« Mais ce n'est pas mon métier ! »

IAC, des freins humains.

Partage de l’information obligée.

Avec IAC tout est décrit dans des documents. Les personnes qui s’appuient sur le secret pour avoir de l’importance perdent leur pouvoir.

Réaction possible :

« Tout le monde peut me remplacer ?! »

IAC, des freins humains.

Courbe d’apprentissage.

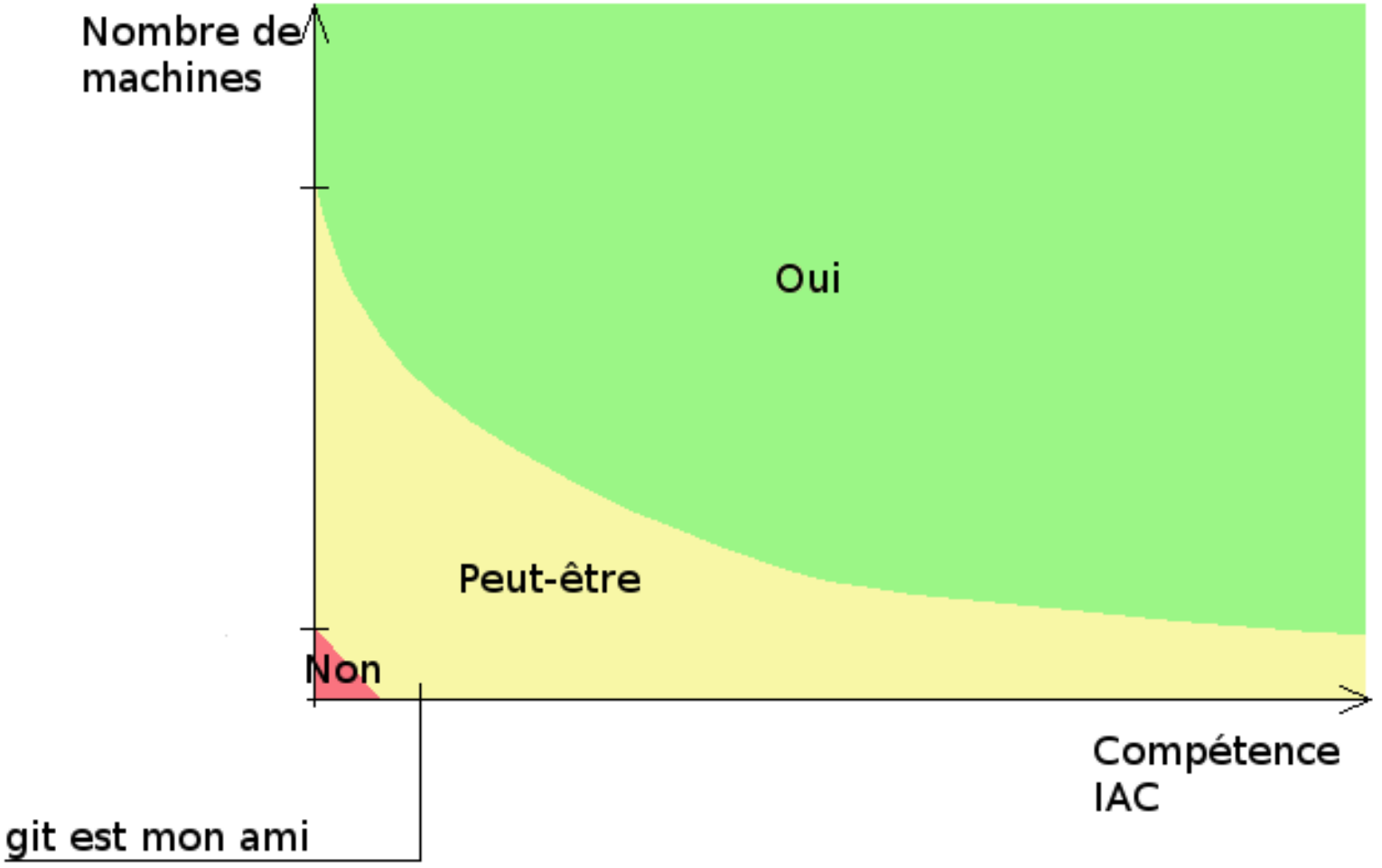
Corollaire du point précédent, il va y avoir une courbe d’apprentissage qui varie grandement selon le profil des personnes.

Ne pas hésiter à utiliser de la formation.

Les ressources ne sont pas pléthoriques en France.

IAC, des freins humains.

Évaluation de réponse à la question : « Est-ce que j'investis dans IAC ? »



IAC réussie ?

Si vous avez réalisé un glissement progressif vers une stratégie où la continuité du service est dans le code de l'infrastructure, plus que dans l'infrastructure elle-même,

Alors, vous êtes sur la bonne voie.

IAC et supervision

Quand tout est automatisé, testé et supervisé, tout devient magique.

- Exemple 1 :
  - La supervision détecte un système de fichiers saturé.
  - IAC étend automatiquement le système de fichiers.
- Exemple 2 :
  - La supervision détecte un cluster surchargé.
  - IAC met en production un nouveau nœud.

Questions / Réponses

Me contacter ?
Par mail : fabien.combernous@adullact.org
Par IRC : freenode.net #puppet Daneel