

Test automatique de la plate-forme Grid'5000

Arthur Garnier

INRIA - AIGorille - Grid'5000

June 12, 2015



Test automatique de la plate-forme Grid'5000

Arthur Garnier

INRIA - AIGorille - Grid'5000

June 12, 2015



[Contexte](#)[Description du problème](#)[Travail réalisé](#)[Conclusion](#)

Contexte

Description du problème

Travail réalisé

Conclusion

○○○
○○○
○○○
○○

Contexte

Description du problème

Travail réalisé

Conclusion

1. Peu de contrainte : Ruby + Jenkins

Contexte

- Centre de recherche

1. Peu de contrainte : Ruby + Jenkins

Contexte

- Centre de recherche
- Grid'5000

Contexte

Description du problème

Travail réalisé

Conclusion

○○○
○○○
○○○
○○○

Contexte

- Centre de recherche
- Grid'5000

- Centre de recherche
- Grid'5000
 - Expérimentation à échelle réelle

1. Peu de contrainte : Ruby + Jenkins



Contexte

- Centre de recherche
- Grid'5000
 - Expérimentation à échelle réelle

- Centre de recherche
- Grid'5000
 - Expérimentation à échelle réelle
 - 1000 noeuds → 1900+ CPUs pour 9000+ cœurs

1. Peu de contrainte : Ruby + Jenkins



Contexte

- Centre de recherche
- Grid'5000
 - Expérimentation à échelle réelle
 - 1000 noeuds → 1900+ CPUs pour 9000+ cœurs

- Centre de recherche
- Grid'5000
 - Expérimentation à échelle réelle
 - 1000 noeuds → 1900+ CPUs pour 9000+ cœurs
 - P2P, Cloud, HPC

1. Peu de contrainte : Ruby + Jenkins



Contexte

- Centre de recherche
- Grid'5000
 - Expérimentation à échelle réelle
 - 1000 noeuds → 1900+ CPUs pour 9000+ cœurs
 - P2P, Cloud, HPC

- Centre de recherche
- Grid'5000
 - Expérimentation à échelle réelle
 - 1000 noeuds → 1900+ CPUs pour 9000+ cœurs
 - P2P, Cloud, HPC
 - **Réservation, VLAN, déploiements**

1. Peu de contrainte : Ruby + Jenkins



Contexte

- ▶ Centre de recherche
- ▶ Grid'5000
 - ▶ Expérimentation à échelle réelle
 - ▶ 1000 noeuds → 1900+ CPUs pour 9000+ cœurs
 - ▶ P2P, Cloud, HPC
 - ▶ **Réservation, VLAN, déploiements**

└─ Description du problème

└─ Description du problème

1. Grenoble, Lille, Nancy, Lyon, Nantes, Rennes, Sophia-Antipolis, Toulouse
2. Conso énergétique, VLAN, Déploiement, ...

Description du problème

► Plusieurs sites

Contexte

Description du problème

Travail réalisé

Conclusion

○○○
○○○
○○○
○○

Description du problème

► Plusieurs sites

└─ Description du problème

└─ Description du problème

1. Grenoble, Lille, Nancy, Lyon, Nantes, Rennes, Sophia-Antipolis, Toulouse
2. Conso énergétique, VLAN, Déploiement, ...

Description du problème

- Plusieurs sites
- **Nombreux services**



Description du problème

- Plusieurs sites
- **Nombreux services**

1. Grenoble, Lille, Nancy, Lyon, Nantes, Rennes, Sophia-Antipolis, Toulouse
2. Conso énergétique, VLAN, Déploiement, ...

Description du problème

- Plusieurs sites
- Nombreux services
- Pannes matérielles et logicielles régulières



Description du problème

- ▶ Plusieurs sites
- ▶ Nombreux services
- ▶ Pannes matérielles et logicielles régulières

1. Grenoble, Lille, Nancy, Lyon, Nantes, Rennes, Sophia-Antipolis, Toulouse
2. Conso énergétique, VLAN, Déploiement, ...

Description du problème

- Plusieurs sites
- Nombreux services
- Pannes matérielles et logicielles régulières
- **Contrainte d'homogénéité**



Description du problème

- ▶ Plusieurs sites
- ▶ Nombreux services
- ▶ Pannes matérielles et logicielles régulières
- ▶ **Contrainte d'homogénéité**

1. Genant si un utilisateur nouveau...

Description du problème
Solutions en place

► Nagios

○○○
○○○
○○○
○○○

Description du problème

Solutions en place

► Nagios

1. Genant si un utilisateur nouveau...

Description du problème

Solutions en place

- Nagios
 - Test de l'état des services



Description du problème

Solutions en place

- Nagios
 - Test de l'état des services

1. Genant si un utilisateur nouveau...

Description du problème

Solutions en place

- Nagios
 - Test de l'état des service
 - Remontée rapide de la défaillance



Description du problème

Solutions en place

- Nagios
 - Test de l'état des service
 - Remontée rapide de la défaillance

1. Genant si un utilisateur nouveau...

Description du problème

Solutions en place

▸ Nagios

- Test de l'état des service
- Remontée rapide de la défaillance
- Pas de test des commandes



Description du problème

Solutions en place

- Nagios
 - Test de l'état des service
 - Remontée rapide de la défaillance
 - Pas de test des commandes

1. Genant si un utilisateur nouveau...

Description du problème

Solutions en place

- Nagios
 - Test de l'état des services
 - Remontée rapide de la défaillance
 - Pas de test des commandes
- Bugzilla



Description du problème

Solutions en place

- Nagios
 - Test de l'état des services
 - Remontée rapide de la défaillance
 - Pas de test des commandes
- Bugzilla

1. Genant si un utilisateur nouveau...

Description du problème

Solutions en place

- Nagios
 - Test de l'état des service
 - Remontée rapide de la défaillance
 - Pas de test des commandes
- Bugzilla
 - Report des bugs détectés



Description du problème

Solutions en place

- Nagios
 - ▶ Test de l'état des service
 - ▶ Remontée rapide de la défaillance
 - ▶ Pas de test des commandes
- Bugzilla
 - ▶ Report des bugs détectés

1. Genant si un utilisateur nouveau...

Description du problème

Solutions en place

- Nagios
 - Test de l'état des service
 - Remontée rapide de la défaillance
 - Pas de test des commandes
- Bugzilla
 - Report des bugs détectés
 - Bug rencontré par l'utilisateur



Description du problème

Solutions en place

- Nagios
 - ▶ Test de l'état des service
 - ▶ Remontée rapide de la défaillance
 - ▶ Pas de test des commandes
- Bugzilla
 - ▶ Report des bugs détectés
 - ▶ Bug rencontré par l'utilisateur

└─ Travail réalisé

└─ Travail réalisé

1. Large couverture + protège les new users

Travail réalisé

1. Automatisation de l'exécution des TP

Contexte

Description du problème

Travail réalisé

Conclusion



Travail réalisé

1. Automatisation de l'exécution des TP

└─ Travail réalisé

└─ Travail réalisé

1. Large couverture + protège les new users

Travail réalisé

1. Automatisation de l'exécution des TP
2. Exécution automatique via Jenkins



Travail réalisé

1. Automatisation de l'exécution des TP
2. Exécution automatique via Jenkins

└─ Travail réalisé

└─ Travail réalisé

1. Large couverture + protège les new users

Travail réalisé

1. Automatisation de l'exécution des TP
2. Exécution automatique via Jenkins
3. Génération automatique de la configuration Jenkins



Travail réalisé

1. Automatisation de l'exécution des TP
2. Exécution automatique via Jenkins
3. Génération automatique de la configuration Jenkins

└─ Travail réalisé

└─ Travail réalisé

1. Large couverture + protège les new users

Travail réalisé

1. Automatisation de l'exécution des TP
2. Exécution automatique via Jenkins
3. Génération automatique de la configuration Jenkins
4. Interface web simplifiée pour consulter les résultats des tests



Travail réalisé

1. Automatisation de l'exécution des TP
2. Exécution automatique via Jenkins
3. Génération automatique de la configuration Jenkins
4. Interface web simplifiée pour consulter les résultats des tests

Automatisation de l'exécution des TP

► Bibliothèque d'action

- Bibliothèque d'action
- « Codage » des scripts



Automatisation de l'exécution des TP

- Bibliothèque d'action
- « Codage » des scripts

Automatisation de l'exécution des TP



Automatisation de l'exécution des TP

Automatisation de l'exécution des TP

For this practical session we will use a routed vlan (6 per site).

```
frontend: oarsub -t deploy -l {"type='kavlan'}/vlan=1+/nodes=3,walltime=01:40:00 -I -n "kavlan_openstack" -t destructive
```

Like a regular deploy job, a shell is now opened on the frontend, you can get you assigned vlan number using the `kavlan` command (add `-j JOBID` if needed)

```
frontend: kavlan -V
```

The command will return an integer and we will pass it to kadeploy in a few moment.

The integer returned by `kavlan -V` is a VLAN number (e.g. 4 is VLAN 704). Each site has 3 unrouted VLAN (`kavlan-local [1..3]`), 6 routed VLAN (`kavlan [4..9]`), and on

Enable the DHCP server of the VLAN

During a deployment, a node gets its IP address at boot time, using DHCP. As our nodes are in a different VLAN than the **Grid5000 production VLAN** they will get their IP from The `kavlan` command therefore allows to enable or disable this DHCP server.

```
frontend: kavlan -e
```

(To disable: `kavlan -d`).

Deploy nodes and change VLAN in one step

So, using kadeploy, we will deploy an Ubuntu image in the VLAN we previously reserved.

```

1 tp = Tp.new(name)
2 job_id, nodes = tp.sub(site, "-l_u-t_deploy_l_u{type='kavlan'}/vlan=1+
  nodes=3,walltime=01:40:00")
3 kavlan_id = tp.exec("frontend.#{site}.grid5000.fr", user, "kavlan_v-
  j_#{@job_id}")[:stdout].strip
4 tp.exec("frontend.#{site}.grid5000.fr", user, "kavlan_e-j_#{@job_id}")
5 tp.deploy("ubuntu-x64-1204", kavlan_id)
6 #...

```



Automatisation de l'exécution des TP

```

1 tp = Tp.new(name)
2 job_id, nodes = tp.sub(site, "-l_u-t_deploy_l_u{type='kavlan'}/vlan=1+
  nodes=3,walltime=01:40:00")
3 kavlan_id = tp.exec("frontend.#{site}.grid5000.fr", user, "kavlan_v-
  j_#{@job_id}")[:stdout].strip
4 tp.exec("frontend.#{site}.grid5000.fr", user, "kavlan_e-j_#{@job_id}")
5 tp.deploy("ubuntu-x64-1204", kavlan_id)
6 #...

```

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─┬─ Exécution automatique via Jenkins

└─┬─┬─ Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- ▶ Plate-forme d'intégration continue
- ▶ OpenSource
- ▶ Très modulaire
- ▶ Vue d'ensemble rapide

1. Pourquoi Jenkins? Déjà utilisé pour des tests unitaires

Contexte

Description du problème

Travail réalisé

Conclusion



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- ▶ Plate-forme d'intégration continue
- ▶ OpenSource
- ▶ Très modulaire
- ▶ Vue d'ensemble rapide

Test automatique de la plate-forme Grid'5000

└ Travail réalisé

└─ Exécution automatique via Jenkins

└─ Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- Plate-forme d'intégration continue
- **OpenSource**
- Très modulaire
- Vue d'ensemble rapide

1. Pourquoi Jenkins? Déjà utilisé pour des tests unitaires

Contexte

Description du problème

Travail réalisé

Conclusion



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- Plate-forme d'intégration continue
- **OpenSource**
- Très modulaire
- Vue d'ensemble rapide

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Exécution automatique via Jenkins

└─ Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- Plate-forme d'intégration continue
- OpenSource
- **Très modulaire**
- Vue d'ensemble rapide

1. Pourquoi Jenkins? Déjà utilisé pour des tests unitaires

Contexte

Description du problème

Travail réalisé

Conclusion



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- Plate-forme d'intégration continue
- OpenSource
- **Très modulaire**
- Vue d'ensemble rapide

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─┬─ Exécution automatique via Jenkins

└─└─ Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- Plate-forme d'intégration continue
- OpenSource
- Très modulaire
- **Vue d'ensemble rapide**

1. Pourquoi Jenkins? Déjà utilisé pour des tests unitaires

Contexte

Description du problème

Travail réalisé

Conclusion



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- Plate-forme d'intégration continue
- OpenSource
- Très modulaire
- **Vue d'ensemble rapide**

- Plate-forme d'intégration continue
- OpenSource
- Très modulaire
- Vue d'ensemble rapide

1. Pourquoi Jenkins? Déjà utilisé pour des tests unitaires



Exécution automatique via Jenkins

Jenkins ?

- Plate-forme d'intégration continue
- OpenSource
- Très modulaire
- Vue d'ensemble rapide

All	Hebdo	Manual Checks	Production env	Reference env	TP	TP_Dev	+
S	W	Name	Dernier succès	Dernier échec	Dernière durée		
		all-environments-deploy-current-ui	1 mo, 0 j - #222	2 j 15 h - #251	5 mn 18 s		
		check-notes-state	10 j - #111	3 j 16 h - #111	1 h 53 mn		
		rdkit_kerbin	16 j - #22	2 j 4 h - #103	1 h 0 mn		
		rdkit_storage	1 mo, 14 j - #105	2 j 11 h - #131	9.8 s		
		katalonze-ui	2 j 15 h - #122	2 j 15 h - #125	12 mn		
		katalonze-ui	10 j - #99	3 j 16 h - #100	19 mn		

Idône:

Legende: RSS pour tout RSS de tous les échecs RSS juste pour les dernières corrélations

Test automatique de la plate-forme Grid'5000

Travail réalisé

Exécution automatique via Jenkins

Exécution automatique via Jenkins

Exécution automatique via Jenkins

Jenkins ?

- Plate-forme d'intégration continue
- OpenSource
- Très modulaire
- Vue d'ensemble rapide

Mais laborieux à utiliser

1. Pourquoi Jenkins? Déjà utilisé pour des tests unitaires



Exécution automatique via Jenkins

Exécution automatique via Jenkins
Jenkins?

- ▶ Plate-forme d'intégration continue
- ▶ OpenSource
- ▶ Très modulaire
- ▶ Vue d'ensemble rapide

All	Hebdo	Manual Checks	Production env	Reference env	TP	TP_Dev	+
S	W	Name	Dernier succès	Dernier échec	Dernière durée		
		all-environments-deploy-current-ui	1 mo, 0 j - #232	2 j 15 h - #251	5 mn 18 s		
		check-notes-state	10 j - #111	3 j 16 h - #111	1 h 53 mn		
		rdkit_karion	16 j - #22	2 j 4 h - #103	1 h 0 mn		
		rdkit_storage	1 mo, 14 j - #105	2 j 11 h - #131	9.8 s		
		kateloby-ui	2 j 15 h - #172	2 j 15 h - #175	12 mn		
		kateloby-ctl	10 j - #99	3 j 16 h - #100	19 mn		



▶ Mais laborieux à utiliser

Exécution automatique via Jenkins
Gestion des builds instables avec TextFinder

- ▶ « Reservation Timed Out » = Faux positif
- ▶ Détection par regex
- ▶ Etat instable



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Gestion des builds instables avec TextFinder

- ▶ « Reservation Timed Out » = Faux positif
- ▶ Détection par *regex*
- ▶ État instable

- ▶ « Reservation Timed Out » = Faux positif
- ▶ Détection par regex
- ▶ Etat instable



Exécution automatique via Jenkins

Gestion des builds instables avec TextFinder

- ▶ « Reservation Timed Out » = Faux positif
- ▶ Détection par regex
- ▶ État instable

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Exécution automatique via Jenkins

└─ Exécution automatique via Jenkins

Exécution automatique via Jenkins
Gestion des builds instables avec TextFinder

- « Reservation Timed Out » = Faux positif
- Détection par *regex*
- **État instable**

Contexte

Description du problème

Travail réalisé

Conclusion



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Gestion des builds instables avec TextFinder

- « Reservation Timed Out » = Faux positif
- Détection par *regex*
- **État instable**

- ▶ « Reservation Timed Out » = Faux positif
- ▶ Détection par *regex*
- ▶ État instable
- ▶ Cet état n'est pas exploitable



Exécution automatique via Jenkins

Gestion des builds instables avec TextFinder

- ▶ « Reservation Timed Out » = Faux positif
- ▶ Détection par *regex*
- ▶ État instable
- ▶ Cet état n'est pas exploitable

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─┬─ Exécution automatique via Jenkins

└─┬─ Exécution automatique via Jenkins

1. Autre outil, script
2. Licence MIT

Exécution automatique via Jenkins
Relancer les builds instables

- Plugin Naginator
- Alternative ?
- Modification du code

Contexte

Description du problème

Travail réalisé

Conclusion



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Relancer les builds instables

- ▶ Plugin Naginator
- ▶ Alternative ?
- ▶ Modification du code

Test automatique de la plate-forme Grid'5000

- └─ Travail réalisé
 - └─ Exécution automatique via Jenkins
 - └─ Exécution automatique via Jenkins

1. Autre outil, script
2. Licence MIT

Exécution automatique via Jenkins
Relancer les builds instables

- Plugin Naginator
- Alternative ?
- Modification du code

Contexte

Description du problème

Travail réalisé

Conclusion



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Relancer les builds instables

- ▶ Plugin Naginator
- ▶ **Alternative ?**
- ▶ Modification du code

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─┬─ Exécution automatique via Jenkins

└─└─ Exécution automatique via Jenkins

1. Autre outil, script
2. Licence MIT

Exécution automatique via Jenkins
Relancer les builds instables

- Plugin Naginator
- Alternative ?
- **Modification du code**

Contexte

Description du problème

Travail réalisé

Conclusion



Exécution automatique via Jenkins

Exécution automatique via Jenkins

Relancer les builds instables

- ▶ Plugin Naginator
- ▶ Alternative ?
- ▶ **Modification du code**

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- Modification de la structure de la plate-forme
- Puppet
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »

1. Obligation de modifier à la main dans jenkins

Contexte

Description du problème

Travail réalisé

Conclusion



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- Modification de la structure de la plate-forme
- Puppet
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- Modification de la structure de la plate-forme
- **Puppet**
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »

1. Obligation de modifier à la main dans jenkins



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- Modification de la structure de la plate-forme
- **Puppet**
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- Modification de la structure de la plate-forme
- Puppet
- **Jenkins Job Builder**
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »

1. Obligation de modifier à la main dans jenkins



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- Modification de la structure de la plate-forme
- Puppet
- **Jenkins Job Builder**
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »

- Modification de la structure de la plate-forme
- Puppet
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »



Génération automatique de la configuration Jenkins

1. Obligation de modifier à la main dans jenkins

- Modification de la structure de la plate-forme
- Puppet
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »

- Modification de la structure de la plate-forme
- Puppet
- Jenkins Job Builder
 - Fichier YAML
 - **Lisible**
 - Facilement « Parsable »

1. Obligation de modifier à la main dans jenkins



Génération automatique de la configuration Jenkins

- ▶ Modification de la structure de la plate-forme
- ▶ Puppet
- ▶ Jenkins Job Builder
 - ▶ Fichier YAML
 - ▶ **Lisible**
 - ▶ Facilement « Parsable »

- Modification de la structure de la plate-forme
- Puppet
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »

1. Obligation de modifier à la main dans jenkins



Génération automatique de la configuration Jenkins

- ▶ Modification de la structure de la plate-forme
- ▶ Puppet
- ▶ Jenkins Job Builder
 - ▶ Fichier YAML
 - ▶ Lisible
 - ▶ Facilement « Parsable »

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- Modification de la structure de la plate-forme
- Puppet
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »
- Tous les plugins ne sont pas gérés

1. Obligation de modifier à la main dans jenkins



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- ▶ Modification de la structure de la plate-forme
- ▶ Puppet
- ▶ Jenkins Job Builder
 - ▶ Fichier YAML
 - ▶ Lisible
 - ▶ Facilement « Parsable »
- ▶ Tous les plugins ne sont pas gérés

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- Modification de la structure de la plate-forme
- Puppet
- Jenkins Job Builder
 - Fichier YAML
 - Lisible
 - Facilement « Parsable »
- Tous les plugins ne sont pas gérés
- Fichier statique

1. Obligation de modifier à la main dans jenkins



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

- ▶ Modification de la structure de la plate-forme
- ▶ Puppet
- ▶ Jenkins Job Builder
 - ▶ Fichier YAML
 - ▶ Lisible
 - ▶ Facilement « Parsable »
- ▶ Tous les plugins ne sont pas gérés
- ▶ Fichier statique

Test automatique de la plate-forme Grid'5000

└ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins
Extension des possibilités de Jenkins Job Builder avec PostScriptJJB

- Configuration interne à Jenkins : XML
- Créer un parser XML
- Modification des fichiers XML
- Prévenir Jenkins de la modification avec son API

1. Commande simple : Fichier + XPath + texte à importer



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

Extension des possibilités de Jenkins Job Builder avec PostScriptJJB

- Configuration interne à Jenkins : XML
- Créer un parser XML
- Modification des fichiers XML
- Prévenir Jenkins de la modification avec son API

Test automatique de la plate-forme Grid'5000

└ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─┬ Génération automatique de la configuration
└─┬ JenkinsGénération automatique de la configuration Jenkins
Extension des possibilités de Jenkins Job Builder avec PostScriptJJB

- Configuration interne à Jenkins : XML
- Créer un parser XML
- Modification des fichiers XML
- Prévenir Jenkins de la modification avec son API

1. Commande simple : Fichier + XPath + texte à importer



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

Extension des possibilités de Jenkins Job Builder avec PostScriptJJB

- Configuration interne à Jenkins : XML
- Créer un parser XML
- Modification des fichiers XML
- Prévenir Jenkins de la modification avec son API

- Configuration interne à Jenkins : XML
- Créer un parser XML
- **Modification des fichiers XML**
- Prévenir Jenkins de la modification avec son API

1. Commande simple : Fichier + XPath + texte à importer



Génération automatique de la configuration Jenkins

Extension des possibilités de Jenkins Job Builder avec PostScriptJJB

- Configuration interne à Jenkins : XML
- Créer un parser XML
- **Modification des fichiers XML**
- Prévenir Jenkins de la modification avec son API

Test automatique de la plate-forme Grid'5000

└ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─┬ Génération automatique de la configuration
└─┬ JenkinsGénération automatique de la configuration Jenkins
Extension des possibilités de Jenkins Job Builder avec PostScriptJJB

- Configuration interne à Jenkins : XML
- Créer un parser XML
- Modification des fichiers XML
- Prévenir Jenkins de la modification avec son API

1. Commande simple : Fichier + XPath + texte à importer



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

Extension des possibilités de Jenkins Job Builder avec PostScriptJJB

- ▶ Configuration interne à Jenkins : XML
- ▶ Créer un parser XML
- ▶ Modification des fichiers XML
- ▶ Prévenir Jenkins de la modification avec son API

- Configuration interne à Jenkins : XML
- Créer un parser XML
- Modification des fichiers XML
- Prévenir Jenkins de la modification avec son API
- **YAML ⇒ JJB ⇒ XML ⇒ PostScriptJJB ⇒ Final XML**

1. Commande simple : Fichier + XPath + texte à importer



Génération automatique de la configuration Jenkins

Extension des possibilités de Jenkins Job Builder avec PostScriptJJB

- ▶ Configuration interne à Jenkins : XML
- ▶ Créer un parser XML
- ▶ Modification des fichiers XML
- ▶ Prévenir Jenkins de la modification avec son API

▶ **YAML ⇒ JJB ⇒ XML ⇒ PostScriptJJB ⇒ Final XML**

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins
Mise à jour automatique du fichier YAML avec AutoUpdateJJB

- Mise à jour des configurations des jobs
- Basé sur l'API
- Parser YAML



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

Mise à jour automatique du fichier YAML avec AutoUpdateJJB

- Mise à jour des configurations des jobs
- Basé sur l'API
- Parser YAML

Test automatique de la plate-forme Grid'5000

└ Travail réalisé

└└ Génération automatique de la configuration Jenkins

└└└ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins
Mise à jour automatique du fichier YAML avec AutoUpdateJJB

- Mise à jour des configurations des jobs
- Basé sur l'API
- Parser YAML



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

Mise à jour automatique du fichier YAML avec AutoUpdateJJB

- Mise à jour des configurations des jobs
- Basé sur l'API
- Parser YAML

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Génération automatique de la configuration Jenkins

└─ Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins
Mise à jour automatique du fichier YAML avec AutoUpdateJJB

- Mise à jour des configurations des jobs
- Basé sur l'API
- **Parser YAML**



Génération automatique de la configuration Jenkins

Génération automatique de la configuration Jenkins

Mise à jour automatique du fichier YAML avec AutoUpdateJJB

- Mise à jour des configurations des jobs
- Basé sur l'API
- **Parser YAML**

2015-06-12

Test automatique de la plate-forme Grid'5000

Travail réalisé

Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

- ▶ Réduction du nombre de clic dans Jenkins
- ▶ Meilleure lisibilité
- ▶ Résumé des sorties console

Contexte

Description du problème

Travail réalisé

Conclusion



Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

- ▶ Réduction du nombre de clic dans Jenkins
- ▶ Meilleure lisibilité
- ▶ Résumé des sorties console

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Interface web simplifiée pour consulter les résultats des tests

└─ Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

- Réduction du nombre de clic dans Jenkins
- **Meilleure lisibilité**
- Résumé des sorties console



Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

- Réduction du nombre de clic dans Jenkins
- **Meilleure lisibilité**
- Résumé des sorties console

Test automatique de la plate-forme Grid'5000

└─ Travail réalisé

└─ Interface web simplifiée pour consulter les résultats des tests

└─ Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

- Réduction du nombre de clic dans Jenkins
- Meilleure lisibilité
- Résumé des sorties console



Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

- ▶ Réduction du nombre de clic dans Jenkins
- ▶ Meilleure lisibilité
- ▶ Résumé des sorties console

Test automatique de la plate-forme Grid'5000

Travail réalisé

Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests



Interface web simplifiée pour consulter les résultats des tests

Interface web simplifiée pour consulter les résultats des tests

g5ktp_kavlan #104

g5ktp_mpi_multi #351

FAILURE : g5ktp_mpi_multi » toulouse-grenoble-lille #351

```

OAR_JOB_ID=429982 oarsh pastel-120.toulouse.grid5000.fr 'cd /home/ajenkins/src/mpi/ && mpirun -machinefile ~/gridnodes --mca plm_rsh_agent "oarsh" --mca opal_net_private_ipv4 "192.168.160.0/24;192.168.14.0/23" --mca btl_tcp_if_exclude ib0,lo,myri0 --mca btl self,sm,tcp tp' on [frontend.toulouse.grid5000.fr]
D, [2015-05-25T07:09:00.301543 #13399] DEBUG -- : oarsh: Cannot find cpuset file : /dev/cpuset/oar/ajenkins_429982/tasks
F, [2015-05-25T07:09:00.301764 #13399] FATAL -- : ***ERROR*** during : OAR_JOB_ID=429982 oarsh pastel-120.toulouse.grid5000.fr 'cd /home/ajenkins/src/mpi/ && mpirun -machinefile ~/gridnodes --mca plm_rsh_agent "oarsh" --mca opal_net_private_ipv4 "192.168.160.0/24;192.168.14.0/23" --mca btl_tcp_if_exclude ib0,lo,myri0 --mca btl self,sm,tcp tp'
I, [2015-05-25T07:09:00.301886 #13399] INFO -- : execute cat ~/oargrid.out on [frontend.toulouse.grid5000.fr]
D, [2015-05-25T07:09:03.013024 #13399] DEBUG -- : grenoble:rdef=nodes=2,lille:rdef=nodes=2,toulouse:rdef=nodes=2
[OAR_GRIDSUB] [toulouse] Date/TZ adjustment: 0 seconds
[OAR_GRIDSUB] [toulouse] Reservation success on toulouse : batchId = 429982
[OAR_GRIDSUB] [lille] Date/TZ adjustment: 0 seconds
[OAR_GRIDSUB] [lille] Reservation success on lille : batchId = 1499936
[OAR_GRIDSUB] [grenoble] Date/TZ adjustment: 1 seconds
[OAR_GRIDSUB] [grenoble]Your reservation was rejected when contacting grenoble
[OAR_GRIDSUB] I delete jobs already submitted
[OAR_GRIDSUB] Delete job 1499936 on the cluster lille
[OAR_GRIDSUB] Delete job 429982 on the cluster toulouse
I, [2015-05-25T07:09:03.014069 #13399] INFO -- : execute oardel 429982 2> /dev/null on [frontend.toulouse.grid5000.fr]
D, [2015-05-25T07:09:05.971207 #13399] DEBUG -- : Deleting the job = 429982 ..ERROR..
W, [2015-05-25T07:09:05.971389 #13399] WARN -- : **Permissible Error** during : oardel 429982 2> /dev/null
Build step 'Execute shell' marked build as failure
Checking console output
Checking console output
Finished: FAILURE
  
```

UNSTABLE : g5ktp_mpi_multi » grenoble-lille-luxembourg #351

UNSTABLE : g5ktp_mpi_multi » luxembourg-lyon-nancy #351

UNSTABLE : g5ktp_mpi_multi » lyon-nancy-nantes #351

UNSTABLE : g5ktp_mpi_multi » nancy-nantes-reims #351

UNSTABLE : g5ktp_mpi_multi » nantes-reims-rennes #351

Conclusion

- Large couverture des possibilités
- Réduction des interactions
- Plusieurs bugs détectés

Conclusion

- Large couverture des possibilités
- Réduction des interactions
- Plusieurs bugs détectés

Conclusion

- Large couverture des possibilités
- Réduction des interactions
- Plusieurs bugs détectés

Conclusion

- Large couverture des possibilités
- Réduction des interactions
- Plusieurs bugs détectés

└─ Conclusion

└─ Conclusion

Conclusion

- Large couverture des possibilités
- Réduction des interactions
- **Plusieurs bugs détectés**



Conclusion

- Large couverture des possibilités
- Réduction des interactions
- **Plusieurs bugs détectés**

Question time

Question time