

Virtualisation des réseaux Extension du laboratoire numéro 2 :

Vérification automatique du réseau par intégration continue

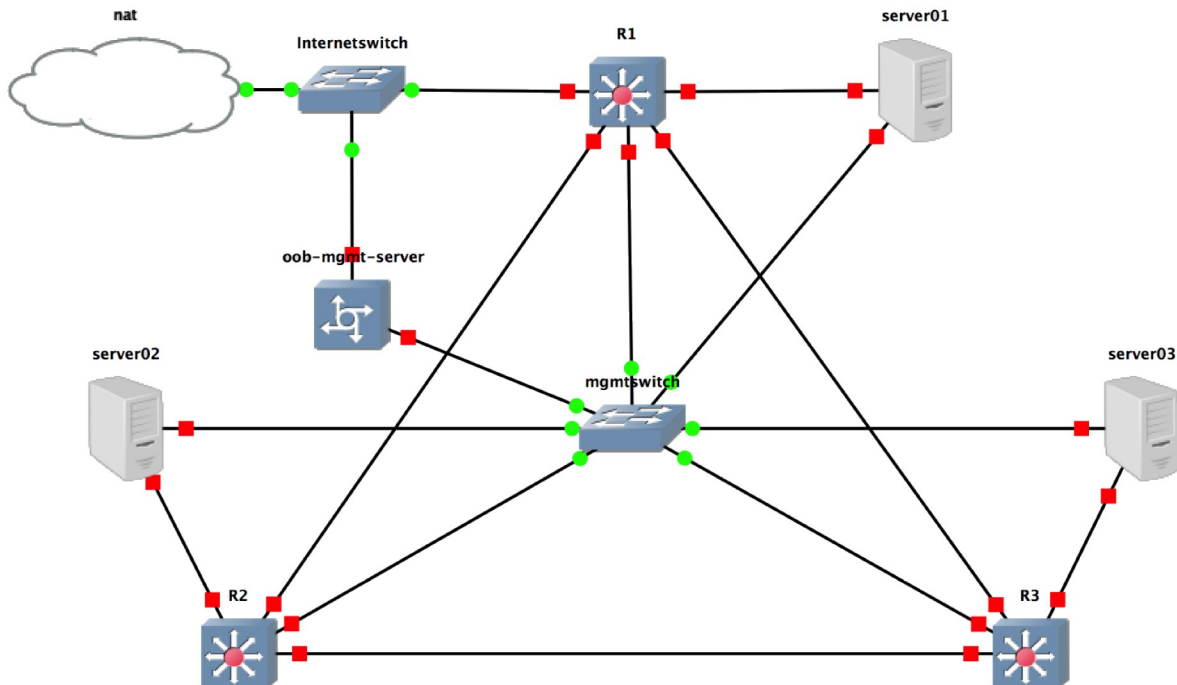


Figure 1 : Schéma du réseau avec à configurer par Ansible et ZTP depuis le serveur de management « oob-mgmt-server »

1. But et objectifs du laboratoire

L'un des avantages majeur de l'automatisation de déploiement de configurations réseaux via Linux/Ansible est de pouvoir vérifier automatiquement que la configuration satisfait toujours des critères après chaque évolution. On effectue alors de l'intégration continue¹, technique jusqu'alors réservée au monde du logiciel, mais appliquée à la configuration d'un réseau.

Le but du présent laboratoire est d'étendre le laboratoire précédent de tel sorte à ce que des changements dans le dépôt gitlab qui contiennent la configuration de votre réseau déclenchent un test automatisé de la bonne marche du réseau. Seuls le *playbook C* et le *playbook D* seront exécutés à travers le moteur d'intégration continue de gitlab. Pour rappel, l'objectifs de ces deux *playbook*, exécutés depuis le serveur de management, étaient les suivants :

- *Playbook C* : Configuration de l'adressage et routage L3 du réseau².
 - Les routeurs et serveurs devront être configurés dans des sous-réseaux du 192.168.0.0/24 en utilisant le moins d'adresses IP possible.
 - Le routage devra être statique³.
 - La bonne configuration devra vérifier que chaque serveur arrive à pinguer deux autres **autres**.

¹ https://fr.wikipedia.org/wiki/Int%C3%A9gration_continue

² <https://docs.cumulusnetworks.com/version/cumulus-linux-37/Layer-1-and-Switch-Ports/Interface-Configuration-and-Management/>

³ <https://docs.cumulusnetworks.com/version/cumulus-linux-37/Layer-3/Routing/>

- *Playbook D* : Configuration de VLANs⁴ sur le réseau
 - Le réseau devra contenir deux VLANs : L'un avec Serveur01 et Serveur02, l'autre avec Server03.
 - Un routage statique inter-vlan devra être installé sur R1.
 - **L'adressage L3 des VLANs devra être configurés dans des sous-réseaux du 192.168.0.0/24**

2. Travail à faire

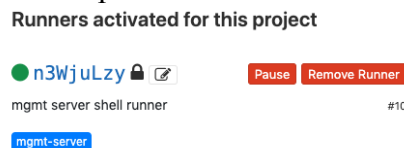
a) Installation d'un runner ssh sur oob-mgmt-server

On passera par la procédure d'installation d'un paquet debian décrite sur <https://docs.gitlab.com/runner/install/linux-manually.html>, l'architecture du serveur oob-mgmt-server étant amd64.

b) Enregistrement du runner comme runner spécifique sur votre dépôt gitlab

La procédure d'enregistrement du runner est décrite là : <https://docs.gitlab.com/runner/register/> . On trouve les informations d'enregistrement du runner gitlab sous Settings → CI/CD → Runners → Spécific Runners. Vous enregistrerez un runner de type « shell » afin de pouvoir exécuter vos scripts de vérification et votre playbook directement sur oob-mgmt-server.

Une fois votre runner enregistré il doit apparaître avec un **cercle vert** dans la liste de vos runners spécifiques, tel qu'indiqué sur la capture d'écran suivante :



c) Ecriture d'un fichier `.gitlab-ci.yml` contenant l'exécution des *playbook C* et *D* et la vérification du réseau.

A l'aide de la documentation suivante : <https://docs.gitlab.com/ee/ci/yaml/README.html> On installera au minimum deux étapes dans le pipeline : l'une qui installera et vérifiera le bon fonctionnement du *Playbook C*, l'autre qui installera et vérifiera le bon fonctionnement du *playbook D*. La vérification sera effectuée en vérifiant la bonne connectivité ICMP entre chaque équipement de la topologie (Est-ce que tout les équipement peuvent pinger tous les équipements du même LAN/VLAN) . On utilisera l'outil classique ping, qui retourne un statut d'erreur faux si l'IP destination n'est pas joignable.

3. Rendu

Vous fournirez l'url du dépôt git exécutant les playbook dans le contexte d'intégration continue. Les deux dernier pipelines exécuté par gitlab sur votre dépôt seront évalués. **Le fichier README.md de votre dépôt devra indiquer le numéro de laboratoire GNS3 utilisé**. Le premier pipeline devra présenter une erreur insérée volontairement tandis que le deuxième n'aura pas d'erreurs., La qualité des logs fournit par le pipeline sera de première importance, surtout en cas d'erreur.

⁴ <https://docs.cumulusnetworks.com/version/cumulus-linux-37/Layer-2/Ethernet-Bridging-VLANs/VLAN-aware-Bridge-Mode/>