Contenido

[1. Pre-Instalar 2](#_Toc67911231)

[2. Instalar 3](#_Toc67911232)

[a. Variables ansible playbook 3](#_Toc67911233)

[b. Variables ansible role nginx-ingress 4](#_Toc67911234)

[c. Ejecutar proyecto ansible role 5](#_Toc67911235)

[3. Post-Instalar 5](#_Toc67911236)

[a. Validar creación de secreto TLS 5](#_Toc67911237)

[b. Validar creación de Nginx Ingress Controller 5](#_Toc67911238)

[c. Validar creación de Default Backend Web 5](#_Toc67911239)

[d. Validar funcionalidad Nginx Ingress Controller 5](#_Toc67911240)

[4. Desinstalar 6](#_Toc67911241)

[Anexos 7](#_Toc67911242)

[1. Imagen de prueba de Nginx Ingress Controller. 7](#_Toc67911243)

**Instructivo Kubernetes Nginx Ingress Controller**

# Pre-Instalar

* El bastión ANSIBLE debe tener instalado HELM, y configurado el KUBECTL con el clúster Kubernetes a utilizar.
* El clúster KUBERNETES debe existir el NAMESPACE *"ingress-nginx".*
* Descargar y copiar el proyecto *“genoit-nginx-ingress-ansiblerole”*al bastión ANSIBLE.
* Descargar y copiar el proyecto *“genoit-nginx-ingress-helmchart”* al bastión ANSIBLE, dentro de la carpeta “helmchart” del proyecto *“genoit-nginx-ingress-ansiblerole”.*

Estructura:

|-- genoit-nginx-ingress-ansiblerole

|-- helmchart

|-- genoit-nginx-ingress-helmchart

* Copiar los certificados en la carpeta “files” del proyecto *“genoit-nginx-ingress-ansiblerole”.*

Estructura:

|-- genoit-nginx-ingress-ansiblerole

|-- files

|-- ca-cert.pem

|-- ca-key.pem

* La instalación no necesita internet ni proxy.
* El usuario “admcuc” de ejecución del ansible role, debe tener permisos sudo en el servidor bastión.

# Instalar

## Variables ansible playbook

Variables importantes en el archivo main.yaml, ubicado en la estructura:

|-- genoit-nginx-ingress-ansiblerole

|-- vars

|-- main.yaml

|  |  |  |
| --- | --- | --- |
| **Variable** | **Valor** | **Descripción** |
| namespace | ingress-nginx | Nombre de namespace |
| nombre\_secreto\_tls | nginx-ingress-tls | Nombre del secreto TLS. |
| state | present | Estado para crear recursos Kubernetes. |
| tls\_cert\_path | "{{ playbook\_dir }}/files/ca-cert.pem" | Ruta de archivo certificado TLS. |
| tls\_key\_path | "{{ playbook\_dir }}/files/ca-key.pem" | Ruta de archivo key TLS. |
| instalar\_enabled | true | Indicador para instalación de ansible role. |
| helmchart\_path | {{ playbook\_dir }}/helmchart/genoit-nginx-ingress-helmchart/ | Ruta de proyecto helmchart. |
| output\_path | /tmp/ansible\_tasks | Ruta temporal de creación de templates. |
| desired\_deployment\_name | nginx-ingress | Nombre de implementación helmchart. |
| controller\_name | controller | Nombre de controlador. |
| controller\_kind | DaemonSet | Tipo de recurso kubernetes. (Daemonset o Deployment) |
| controller\_image | repository: k8s.gcr.io/ingress-nginx/controller tag: "v0.44.0" digest: sha256:3dd0fac48073beaca2d67a78c746c7593f9c575168a17139a9955a82c63c4b9a pullPolicy: IfNotPresent runAsUser: 101 allowPrivilegeEscalation: true | Valores de la imagen Docker de Kubernetes Nginx ingress controller. |
| controller\_affinity | nodeAffinity:  requiredDuringSchedulingIgnoredDuringExecution:  nodeSelectorTerms:  - matchExpressions:  - key: node-role.kubernetes.io/node-loadbalancer  operator: In  values:  - "true" | Valores del node affinity. |
| controller\_nodeSelector | node-role.kubernetes.io/node-loadbalancer: "true" | Valores del node selector. |
| controller\_service | enabled: true annotations: {} labels: {} externalIPs: [] loadBalancerSourceRanges: [] enableHttp: true enableHttps: true ports:  http: 80  https: 443 targetPorts:  http: http  https: https type: NodePort nodePorts:  http: "30080"  https: "30443"  tcp: {}  udp: {} internal:  enabled: false  annotations: {}  loadBalancerSourceRanges: [] | Valores del recurso tipo Servicio. type: Valores posibles LoadBalancer y NodePort. nodePorts: Valores de puertos para protocolos.  http: "30080"  https: "30443" |
| controller\_extraArgs | default-ssl-certificate: "{{ namespace}}/{{ nombre\_secreto\_tls }}" enable-ssl-passthrough: true | Valores de argutementos extra del controlador. default-ssl-certificate: Ruta del secreto tipo tls creado en un namespace, el cual contiene los certificados TLS. enable-ssl-passthrough: Habilitar el paso a través de SSL. |
| defaultBackend\_enabled | true | Indicador para instalación de web default backend. |
| defaultBackend\_name | defaultbackend | Nombre de la web default backend. |
| defaultBackend\_image | repository: k8s.gcr.io/defaultbackend-amd64 tag: "1.5" pullPolicy: IfNotPresent runAsUser: 65534 runAsNonRoot: true readOnlyRootFilesystem: true allowPrivilegeEscalation: false | Valores de imagen Docker de la web default backend. |

## Variables ansible role nginx-ingress

Todas las variables del ansible role del helmchart en el archivo main.yaml, ubicado en la estructura:

|-- genoit-nginx-ingress-ansiblerole

|-- roles

|-- nginx-ingress

|-- vars

|-- main.yaml

## Ejecutar proyecto ansible role

* Dirigirse a la carpeta raíz del proyecto.

|-- genoit-nginx-ingress-ansiblerole

|-- main.yaml

* Ejecutar el comando, con un usuario con permisos sudo, en el bastión Ansible.

ansible-playbook main.yml

# Post-Instalar

## Validar creación de secreto TLS

Ejecutar el comando:

kubectl -n { namespace } | grep { nombre\_secreto\_tls }

Ejm:

kubectl -n ingress-nginx get secret | grep nginx-ingress-tls

## Validar creación de Nginx Ingress Controller

Ejecutar el comando:

kubectl -n { namespace } all | grep { controller\_name }

Ejm:

kubectl -n ingress-nginx get all | grep controller

## Validar creación de Default Backend Web

Ejecutar el comando:

kubectl -n { namespace } all | grep { defaultBackend\_name }

Ejm:

kubectl -n ingress-nginx get all | grep defaultbackend

## Validar funcionalidad Nginx Ingress Controller

* + Configurar los manifiestos de test.
  + Registrar en el host del cliente, una IP de los nodos Kubernetes con el nombre de host del manifiesto de prueba “ingress.yaml”.
  + Ejecutar los manifiestos de test, que crean recursos Kubernetes: 2 servicios y un ingress.
  + Abrir un navegador y dirigirse a los servicios creados. (ver anexo 1)

# Desinstalar

* Desinstalar proyecto *genoit-nginx-ingress-ansiblerole*, ejecutando los comandos:
  + *helm uninstall {{ desired\_deployment\_name }} -n {{ namespace }}*

Ejm:

helm uninstall nginx-ingress -n ingress-nginx

* + *kubectl -n {{ namespace }} delete secret {{ nombre\_secreto\_tls }} {{ desired\_deployment\_name }}-ingress-nginx-admission*

Ejm:

kubectl -n ingress-nginx delete secret nginx-ingress-tls nginx-ingress-ingress-nginx-admission

* Desinstalar antiguo Nginx Controller del clúster Kubernetes, ejecutando el comando:

Kubectl delete -f { ruta\_manifiestos\_antiguos }

Nota: la variable “ruta\_manifiestos\_antiguos” debe contener varios manifiestos.

# Anexos

## Imagen de prueba de Nginx Ingress Controller.

Captura de pantalla de un celular

Descripción generada automáticamente