Contenido

[1. Pre-Instalar 2](#_Toc69247554)

[2. Instalar 3](#_Toc69247555)

[a. Variables ansible playbook 3](#_Toc69247556)

[b. Variables ansible role istio 5](#_Toc69247557)

[c. Ejecutar proyecto ansible role 5](#_Toc69247558)

[3. Post-Instalar 6](#_Toc69247559)

[a. Validar creación de secreto TLS 6](#_Toc69247560)

[b. Validar creación de Istio 6](#_Toc69247561)

[c. Validar funcionalidad Istio 6](#_Toc69247562)

[4. Desinstalar 7](#_Toc69247563)

[Anexos 8](#_Toc69247564)

[1. Configuración Helmchart Istio Discovery 8](#_Toc69247565)

[2. Imagen de prueba de Istio seguro en navegador. 9](#_Toc69247566)

[3. Imagen de prueba de Istio seguro en consola. 9](#_Toc69247567)

**Instructivo Kubernetes Istio**

# 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 *"istio-system".*
* Descargar y copiar el proyecto *“genoit-istio-ansiblerole”*al bastión ANSIBLE.
* Descargar y copiar el proyecto *“genoit-istio-helmchart”* al bastión ANSIBLE, dentro de la carpeta “helmchart” del proyecto *“genoit-istio-ansiblerole”.*

Estructura:

|-- genoit-istio-ansiblerole

|-- helmchart

|-- genoit-istio-helmchart

* Verificar la configuración del “nodeSelector” en el helmchart de Istio Dicovery, ubicado en el archivo “*genoit-istio-ansiblerole\helmchart\genoit-istio-helmchart\manifests\charts\istio-control\istio-discovery\templates\deployment.yaml*”. (ver anexo 1)
* Copiar los certificados en la carpeta “files” del proyecto *“genoit-istio-ansiblerole”.*

Estructura:

|-- genoit-istio-ansiblerole

|-- files

|-- ca-cert.pem

|-- ca-key.pem

* La instalación necesita internet para descargar las imágenes del contenedor Docker.
* 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-istio-ansiblerole

|-- vars

|-- main.yaml

|  |  |  |
| --- | --- | --- |
| **Variable** | **Valor** | **Descripción** |
| namespace | istio-system | Nombre de namespace |
| nombre\_secreto\_tls | genoit-credential | 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-istio-helmchart/" | Ruta principal de proyectos helmchart |
| helmchart\_istio\_base\_path | "{{ helmchart\_path }}/manifests/charts/base/" | Ruta de proyecto helmchart Istio Base |
| helmchart\_istio\_cni\_path | "{{ helmchart\_path }}/manifests/charts/istio-cni/" | Ruta de proyecto helmchart Istio CNI |
| helmchart\_istio\_discovery\_path | "{{ helmchart\_path }}/manifests/charts/istio-control/istio-discovery/" | Ruta de proyecto helmchart Istio Discovery |
| helmchart\_istio\_ingress\_path | "{{ helmchart\_path }}/manifests/charts/gateways/istio-ingress/" | Ruta de proyecto helmchart Istio Ingress |
| helmchart\_istio\_egress\_path | "{{ helmchart\_path }}/manifests/charts/gateways/istio-egress/" | Ruta de proyecto helmchart Istio Egress |
| output\_path | /tmp/ansible\_tasks | Ruta temporal de creación de templates. |
| desired\_deployment\_name\_base | istio-base | Nombre de implementación helmchart Istio Base |
| desired\_deployment\_name\_cni | istio-cni | Nombre de implementación helmchart Istio CNI |
| desired\_deployment\_name\_discovery | istio-discovery | Nombre de implementación helmchart Istio Discovery |
| desired\_deployment\_name\_ingress | istio-ingress | Nombre de implementación helmchart Istio Ingress |
| desired\_deployment\_name\_egress | istio-egress | Nombre de implementación helmchart Istio Egress |
| jwt\_policy | "first-party-jwt" | Política para validar JWT.("third-party-jwt" o "first-party-jwt") |
| istio\_cni\_psp\_cluster\_role | "psp:privileged" | Nombre de ClusterRole PodSecurityPolicies preexistente para vincular a ServiceAccount istio/cni. |
| istio\_cni\_enabled | true | Activar la funcionalidad Istio CNI |
| istio\_cni\_global\_hub | "docker.io/istio" | Nombre de imagen docker de Istio CNI |
| istio\_cni\_global\_tag | 1.9.2 | Versión de imagen docker de Istio CNI |
| istio\_cni\_logLevel | "info" | Nivel de log de Istio CNI |
| istio\_cni\_excludeNamespaces | - istio-system - kube-system | Lista de exclusión de namespaces |
| istio\_discovery\_global\_hub | "docker.io/istio" | Nombre de imagen docker de Istio Discovery |
| istio\_discovery\_global\_tag | 1.9.2 | Versión de imagen docker de Istio Discovery |
| istio\_discovery\_global\_logging | level: "default:info" | Nivel de log de Istio Discovery |
| istio\_discovery\_pilot\_nodeSelector | node-role.kubernetes.io/node-loadbalancer: "true" | Label de nodo selector de pod Istio Discovery |
| istio\_all\_global\_jwtPolicy | "{{ jwt\_policy }}" | Política para validar JWT.("third-party-jwt" o "first-party-jwt") |
| istio\_all\_global\_istioNamespace | "{{ namespace }}" | Nombre de namespace |
| istio\_gateways\_istio\_ingressgateway\_name | "istio-ingressgateway" | Nombre de recurso Ingress Gateway |
| istio\_gateways\_istio\_ingressgateway\_labels | app: istio-ingressgateway istio: ingressgateway | Label del recurso Ingress Gateway |
| istio\_gateways\_istio\_ingressgateway\_ports | - port: 15021  targetPort: 15021  name: status-port  protocol: TCP - port: 80  targetPort: 8080  nodePort: 31080  name: http2  protocol: TCP - port: 443  targetPort: 8443  nodePort: 31443  name: https  protocol: TCP - port: 15012  targetPort: 15012  name: tcp-istiod  protocol: TCP - port: 15443  targetPort: 15443  name: tls  protocol: TCP | Puertos del servicio Ingress Gateway |
| istio\_gateways\_istio\_ingressgateway\_type | "NodePort" | Tipo de servicio de Ingress Gateway |
| istio\_gateways\_ingressgateway\_global\_defaultNodeSelector | node-role.kubernetes.io/node-loadbalancer: "true" | Label de nodo selector de pod Ingress Gateway |
| istio\_gateways\_istio\_ingressgateway\_runAsRoot | false | Indicador de ejecucion como root de pod Ingress Gateway |
| istio\_gateways\_ingressgateway\_global\_hub | "docker.io/istio" | Nombre de imagen docker de Ingress Gateway |
| istio\_gateways\_ingressgateway\_global\_tag | 1.9.2 | Versión de imagen docker de Ingress Gateway |
| istio\_gateways\_ingressgateway\_global\_logging | level: "default:info" | Nivel de log de Istio Ingress Gateway |
| istio\_gateways\_istio\_egressgateway\_name | "istio-egressgateway" | Nombre de recurso Egress Gateway |
| istio\_gateways\_istio\_egressgateway\_labels | app: istio-egressgateway istio: egressgateway | Label del recurso Egress Gateway |
| istio\_gateways\_istio\_egressgateway\_ports | - port: 80  targetPort: 8080  nodePort: 32080  name: http2  protocol: TCP - port: 443  name: https  targetPort: 8443  nodePort: 32443  protocol: TCP - port: 15443  targetPort: 15443  name: tls  protocol: TCP | Puertos del servicio Egress Gateway |
| istio\_gateways\_istio\_egressgateway\_type | "NodePort" | Tipo de servicio de Egress Gateway |
| istio\_gateways\_egressgateway\_global\_defaultNodeSelector | node-role.kubernetes.io/node-loadbalancer: "true" | Label de nodo selector de pod Egress Gateway |
| istio\_gateways\_istio\_egressgateway\_runAsRoot | false | Indicador de ejecucion como root de pod Egress Gateway |
| istio\_gateways\_egressgateway\_global\_hub | "docker.io/istio" | Nombre de imagen docker de Egress Gateway |
| istio\_gateways\_egressgateway\_global\_tag | 1.9.2 | Versión de imagen docker de Egress Gateway |
| istio\_gateways\_egressgateway\_global\_logging | level: "default:info" | Nivel de log de Istio Egress Gateway |

## Variables ansible role istio

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

|-- genoit-istio-ansiblerole

|-- roles

|-- istio

|-- vars

|-- main.yaml

## Ejecutar proyecto ansible role

* Dirigirse a la carpeta raíz del proyecto.

|-- genoit-istio-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 } get secret | grep { nombre\_secreto\_tls }

Ejm:

kubectl -n istio-system get secret | grep genoit-credential

## Validar creación de Istio

Ejecutar el comando:

kubectl -n { namespace } get all | grep istio

Ejm:

kubectl -n istio-system get all | grep istio

## Validar funcionalidad Istio

* + Configurar y ejecutar los archivos ubicados en la carpeta “manifiestos\_test”.
  + Registrar en el host del cliente, una IP de los nodos Kubernetes con el nombre de host del manifiesto de prueba “3-gateways.yaml” y “4-virtual\_services.yaml”.
  + Ejecutar los manifiestos test, los cuales crean recursos Kubernetes: 1 servicio, 1 Deployment, 1 Gateway , y un VirtualService.
  + Abrir un navegador y dirigirse a la ruta:

[https://helloworld-v1.micro-ingress.genoit.int:31443/hello](https://helloworld-v1.micro-ingress.sunat-cuc.int:31443/hello)

Los resultados deben ser una respuesta correcta. (ver anexo 2)

* + Abrir una consola, ubicar el certificado, y ejecutar los siguientes comandos:

export INGRESS\_HOST=$(kubectl get po -l istio=ingressgateway -n istio-system -o jsonpath='{.items[0].status.hostIP}')

export SECURE\_INGRESS\_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="https")].nodePort}')

curl -v -HHost:helloworld-v1.micro-ingress.genoit.int \

--resolve " helloworld-v1.micro-ingress.genoit.int:$SECURE\_INGRESS\_PORT:$INGRESS\_HOST" \

--cacert ca-cert.pem "https://helloworld-v1.micro-ingress.genoit.int:$SECURE\_INGRESS\_PORT/hello"

Los resultados deben ser una respuesta correcta. (ver anexo 3)

# Desinstalar

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

Ejm:

helm uninstall istio-egress -n istio-system

helm uninstall istio-ingress -n istio-system

helm uninstall istio-discovery -n istio-system

helm uninstall istio-cni -n istio-system

helm uninstall istio-base -n istio-system

* + *kubectl -n {{ namespace }} delete secret {{ nombre\_secreto\_tls }} istio-ca-secret*

Ejm:

kubectl -n istio-system delete secret istio-ca-secret genoit-credential

* + *kubectl -n {{ namespace }} delete configmap --all*

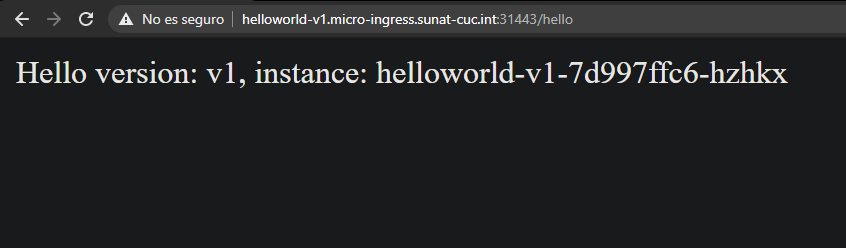
Ejm:

kubectl -n istio-system delete configmap --all

# Anexos

## Configuración Helmchart Istio DiscoveryTexto Descripción generada automáticamente

## Imagen de prueba de Istio seguro en navegador.



## Imagen de prueba de Istio seguro en consola.

