***Note:***

* *Negli yaml sono presenti****placeholder****e comunque verificate nomenclatura e correttezza prima di applicarli*
* *In allegato il file in MarkDown originale più leggibile.*

# Additional Ingress Gateway per i servizi di backend

## ServiceMeshControlPlane configuration

Aggiunta ingressGateway dedicato ai servizi di backend

Al pari del ingress Gateway di default (istio-ingressgateway) quello dedicato ai servizi di backend è stato installato sui nodi infra.

Visto che viene utilizzata l’esposizione su NodePort è necessario predisporre altre porte che non vadano in conflitto con l’Ingress Gateway di default.

Si sono scelte le porte: - 32082 - status-port - 30011 - http - 30012 - https

Di seguito la sezione additionalIngress aggiunta nel ServiceMeshControlPlane (basic)

oc -n istio-system edit smcp basic

---

apiVersion: [maistra.io/v2](https://urldefense.com/v3/__http:/maistra.io/v2__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOeDXKWwvQ$)

kind: ServiceMeshControlPlane

metadata:

  name: basic

  namespace: istio-system

spec:

  ...

  gateways:

    ...

    additionalIngress:

      backend-ingressgateway:

        enabled: true

        runtime:

          pod:

            nodeSelector:

[node-role.kubernetes.io/infra](https://urldefense.com/v3/__http:/node-role.kubernetes.io/infra__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOcCfBFL3g$): ""

            tolerations:

            - effect: NoSchedule

              key: [node-role.kubernetes.io/infra](https://urldefense.com/v3/__http:/node-role.kubernetes.io/infra__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOcCfBFL3g$)

            - effect: NoExecute

              key: [node-role.kubernetes.io/infra](https://urldefense.com/v3/__http:/node-role.kubernetes.io/infra__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOcCfBFL3g$)

            - effect: NoSchedule

              key: [node.ocs.openshift.io/storage](https://urldefense.com/v3/__http:/node.ocs.openshift.io/storage__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOfkq49awA$)

        service:

          metadata:

            labels:

              istio: backend-ingressgateway

          ports:

          - name: status-port

            nodePort: 32082

            port: 15021

            protocol: TCP

            targetPort: 15021

          - name: http

            nodePort: 30011

            port: 80

            protocol: TCP

            targetPort: 8080

          - name: https

            nodePort: 30012

            port: 443

            protocol: TCP

            targetPort: 8443

          type: NodePort

...

Verificare che venga creato il pod backend-ingressgateway e il relativo service NodePort sulle sopraindicate porte.

oc -n istio-system get pod backend-ingressgateway

oc -n istio-system get svc backend-ingressgateway

Verificare inoltre le labels (app e istio) e il serviceAccount

IG\_NAME=$(oc -n istio-system get po -l app=backend-ingressgateway -o jsonpath='{.items[0].[metadata.name](https://urldefense.com/v3/__http:/metadata.name__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOfOT1DAyA$)}')

oc -n istio-system get po $IG\_NAME -o jsonpath='{.metadata.labels.istio}'

oc -n istio-system get po $IG\_NAME -o jsonpath='{.[metadata.labels.app](https://urldefense.com/v3/__http:/metadata.labels.app__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOet8o7--Q$)}'

oc -n istio-system get po $IG\_NAME -o jsonpath='{.spec.serviceAccount}'

Dovrebbero essere:

istio          = backend-ingressgateway

app            = backend-ingressgateway

serviceAccount = backend-ingressgateway-service-account

Questa attività va eseguita per cluster (eseguita oggi sul cluster di dev/collaudo)

## Applicazioni

Per ogni servizio invece va adeguato o aggiunto il Gateway in modo che faccia riferimento al ingressGateway di backend (istio: backend-ingressgateway)

### Gateway

---

apiVersion: [networking.istio.io/v1beta1](https://urldefense.com/v3/__http:/networking.istio.io/v1beta1__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOc02A91lA$)

kind: Gateway

metadata:

  name: $GW\_NAME

  namespace: $NAMESPACE

spec:

  selector:

    istio: backend-ingressgateway

  servers:

  - hosts:

    - $APP\_NAME.$DOMAIN

    port:

      name: https

      number: 443

      protocol: HTTPS

    tls:

      credentialName: $TLS\_SECRET

      mode: SIMPLE

Il VirtualService non dovrebbe essere modificato se già presente perché fa già rifermento al gateway che abbiamo appena modificato

### AuthorizationPolicy

Per abilitare infine il traffico in ingresso tramite Authorization Policies, vanno aggiunte le seguenti Authorization Policy

#### allow-istio-ingressgateway

---

apiVersion: [security.istio.io/v1beta1](https://urldefense.com/v3/__http:/security.istio.io/v1beta1__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOecP8Ja6Q$)

kind: AuthorizationPolicy

metadata:

  name: allow-istio-ingressgateway

  namespace: $NAMESPACE

spec:

  action: ALLOW

  rules:

  - to:

    - operation:

        paths:

        - '\*'

  selector:

    matchLabels:

      app: backend-ingressgateway

La label *app* deve essere uguale a quella del pod *backend-ingressgateway* vista prima #### allow-gateway-sa

apiVersion: [security.istio.io/v1beta1](https://urldefense.com/v3/__http:/security.istio.io/v1beta1__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOecP8Ja6Q$)

kind: AuthorizationPolicy

metadata:

  name: allow-gateway-sa

  namespace: $NAMESPACE

spec:

  action: ALLOW

  rules:

  - from:

    - source:

        principals:

        - cluster.local/ns/istio-system/sa/backend-ingressgateway-service-account

Il service account deve essere uguale a quello utilizzato nel pod *backend-ingressgateway* visto prima

Documentazione Openshift:

* - [SMCP reference](https://urldefense.com/v3/__https:/docs.openshift.com/container-platform/4.15/service_mesh/v2x/ossm-reference-smcp.html__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOeciUaiMA$)
* - [Traffic management](https://urldefense.com/v3/__https:/docs.openshift.com/container-platform/4.15/service_mesh/v2x/ossm-traffic-manage.html__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOeYxQxXVA$)
* - [Security](https://urldefense.com/v3/__https:/docs.openshift.com/container-platform/4.15/service_mesh/v2x/ossm-security.html__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOdC_Nstgg$)

Documentazione Istio;

* - [IngressGateway](https://urldefense.com/v3/__https:/istio.io/latest/docs/tasks/traffic-management/ingress/ingress-control/__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOdKKCAeHA$)
* - [Managing Gateways](https://urldefense.com/v3/__https:/istio.io/latest/docs/setup/additional-setup/gateway/*managing-gateways__;Iw!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOeURU3R6w$)
* - [Gateway](https://urldefense.com/v3/__https:/istio.io/latest/docs/reference/config/networking/gateway/__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOde7TLIpw$)
* - [VirtualService](https://urldefense.com/v3/__https:/istio.io/latest/docs/reference/config/networking/virtual-service/__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOfW2V6-Qw$)
* - [AuthorizationPolicy](https://urldefense.com/v3/__https:/istio.io/latest/docs/reference/config/security/authorization-policy/__;!!OrxsNty6D4my!6Z-wxclz1oiBjNPvgUqDL1-FKKP31MinhMM8085kWH3FQIp7CjHS71v2411iSMzYh3AqznCJcu5IKOcaUVxu9g$)