# SNAT Traffic for Kubernetes with Cisco ACI CNI

Revised: March 28, 2023

# New and Changed Information

The following table lists the significant changes up to this current release. The table does not provide an exhaustive list of all changes or of the new features up to this release.

*Table 1: New Features and Changed Information for SNAT for Kubernetes Environments*

| Cisco APIC Release Version | Feature | Description |
|---|---|---|
| 5.0(1) | Destination-based egress Source Network Address Translation (SNAT) | You can associate multiple SNAT policies with a pod so hat the SNAT IP is allocated based on the destination. You can also completely suppress SNAT to known destinations on the same fabric. |
| 4.2(1) | SNAT for Kubernetes environments | You can now configure SNAT for Kubernetes environments within the Cisco Application Centric Infrastructure (ACI) fabric.<br>The feature enables you to control SNAT traffic and apply granular SNAT policy. |

# SNAT for Kubernetes

Kubernetes pods need external access (outside their cluster) to download packages, clone repositories, and access services through published APIs. However, pod IP addresses are local to the cluster, and any traffic going outside the cluster must go through a source IP address translation.

In earlier releases, the translation was done outside the Cisco Application Centric Infrastructure (ACI) fabric on an independently configured and managed Network Address Translation (NAT) device. This setup resulted in operational complexity in requiring you to manage the external NAT device. You also had no control over NAT traffic, and because all traffic was mapped to one or more IP addresses on the NAT device, the Kubernetes context was lost. You could not instrument monitoring and visibility on the NAT traffic based on which Kubernetes namespaces, deployments, services, and pods that the traffic originated from. It was also not possible to apply firewall rules to NAT traffic because pods are often ephemeral and their IP addresses change.

### SNAT with Cisco ACI CNI

Beginning with Cisco Application Policy Infrastructure Controller (APIC) Release 4.2(1), you can configure and manage Source IP Network Address Translation (SNAT) for Kubernetes from within the Cisco ACI fabric with the Cisco ACI Container Network Interface (CNI) Plug-in. With the SNAT feature, you do not need to install, configure, and maintain a NAT device outside the Cisco ACI fabric. Moving address translation into the fabric also means that you can base policy on SNAT IP addresses as well as on individual pods or a group of pods scoped by cluster, namespace, service, or deployment. This enables you to have granular control and visibility over SNAT traffic by applying a SNAT policy.

# SNAT for Kubernetes Guidelines and Limitations

The following guidelines and limitations apply when configuring SNAT for Kubernetes:

- SNAT is only supported for traffic egressing the Cisco Application Centric Infrastructure (ACI) fabric through an L3Out. EPG-to-EPG traffic is not supported.

# Prerequisites for Configuring SNAT

Perform the following tasks before you can configure Source IP Network Address Translation (SNAT) for Kubernetes.

- Use Linux Kernel 4.6 or later; the SNAT feature requires the Open vSwitch NAT support available in the release.

  See the article "Releases" under that latest FAQ on the Open vSwitch website. Using an unsupported kernel results in errors in the OpFlex agent or Open vSwitch logs related to conntrack, nat, or zones.

- Install Ubuntu 18.04 or later or Red Hat Enterprise Linux Server Release 7.6 or later.
- Make sure that Kubernetes is correctly installed and integrated with the Cisco Application Centric Infrastructure (ACI) fabric.

# Workflow for Configuring SNAT

This section provides a high-level description of the tasks you perform to configure Source Network Address Translation (SNAT) for Kubernetes and apply SNAT policy to Kubernetes resources.

1. (Optional) Configure the source port range for the SNAT IP address.

   Configure the source port range if you do not want to use the default values. You may want to specify a greater range and size if you expect that a single SNAT policy will have more than 3000 pod-initiated connections for each node. See the section Specify Source Port Ranges for the SNAT IP Address, on page 4 in this guide for details.

2. Install the Cisco Application Centric Infrastructure (ACI) Container Network Interface (CNI) Plug-in.

   Installing the plug-in defines the custom resource definitions (CRDs) and implements the controllers that manage them. See the following documents on Cisco.com: sections in section "Provisioning Cisco ACI to Work with Kubernetes" in the guide *Cisco ACI and Kubernetes Integration* on Cisco.com for instructions.

   - **Installation:**
     - *Cisco ACI and Kubernetes Integration*
     - *Cisco ACI and OpenShift Integration*
     - *Cisco ACI and Docker EE Integration*
   - **Upgrade:** *Upgrading the Cisco ACI CNI Plug-in*

3. Create the SNAT policy.

   You define policy by creating a custom resource called `snatpolicy`. See the section Configure the SNAT Policy Resource, on page 6 in this guide.

4. Apply policy to Kubernetes resources.

   You can apply policy to a pod, deployment, and namespace, by setting a label. You also can apply policy to a service or cluster. See the individual sections in this guide.

5. By default, SNAT subnets/IP are not advertised externally. You need to configure external routers with static routes. However, you can configure ACI to advertise the SNAT subnets/IP through a dynamic routing protocol. For details, see the Service Subnet Advertisement section in the *Cisco ACI and Kubernetes Integration* document.

# Deploying SNAT

## Specify Source Port Ranges for the SNAT IP Address

This step is optional. Applying Network Address Translation (NAT) to the traffic involves translating the IP address and the source port. The range of source ports and the number to be used for each node has a default configuration. Instead of accepting these default values, you can set your own port-range configuration values by modifying the `acc-provision` file before you deploy the Cisco Application Centric Infrastructure (ACI) Container Network Interface (CNI) Plug-in.

The following are the default values:

- **Range**: Ports in the range of 5000 to 65000
- **Per-node port range size**: 3000

  With 3000 ports for each node, a single Source Network Address Translation (SNAT) IP address can support up to 20 nodes in a cluster.

  If your SNAT policy maps to a very large number of pods, the SNAT IP address source port range of 5000 to 65000 may become exhausted globally. Or the 3000 ports may get exhausted on a specific node. If that occurs, and you have allocated more than SNAT IP address in the SNAT policy, the new IP address is automatically allocated.

**Note** Be sure to determine the number of SNAT IP addresses correctly. If you expect that a single SNAT policy can have more than 3000 pod-initiated connections for each node, increase the range if you have fewer than 20 nodes. You also can, alternatively or in addition, allocate enough SNAT IP addresses for the SNAT policy.

### Before you begin

Fulfill all the requirements in the section Prerequisites for Configuring SNAT, on page 3 in this guide.

### Procedure

Set the following in the `acc-provision` input file in the `kube_config` section:

**Example:**

```
kube_config:
  …
  snat_operator:
    …
    port_range:
      start: 5000
      end: 65000
      ports_per_node: 3000
```

### What to do next

Deploy the Cisco ACI CNI Plug-in; See the section Install the Cisco ACI CNI Plug-in and Deploy the SNAT Operator, on page 5.

# Install the Cisco ACI CNI Plug-in and Deploy the SNAT Operator

You install the plug-in using the **acc-provision** command, which also programs Cisco Application Policy Infrastructure Controller (APIC) and generates the Kubernetes deployment specification.

The plug-in also contains and deploys the Source Network Address Translation (SNAT) operator. The SNAT operator which defines four custom resource definitions (CRD) and implements the controllers that manage each of the CRDs.

### Before you begin

- Fulfill the requirements in the section Prerequisites for Configuring SNAT, on page 3 in this guide.
- If you want to specify port ranges for the SNAT IP address rather than accept the defaults, follow the instructions in the Specify Source Port Ranges for the SNAT IP Address, on page 4 section of this guide.

### Procedure

Follow the procedure "Provisioning Cisco ACI to Work with Kubernetes" in the guide *Cisco ACI and Kubernetes Integration*

### What to do next

Define SNAT policy; see the section "Configuring and Applying SNAT Policy" in this guide.

# Configuring and Applying SNAT Policy

## Planning SNAT Policy

You apply Source Network Address Translation (SNAT) to Kubernetes resources using the `snatpolicies`. The SNAT policy captures your input for the following:

- The pool of SNAT IP addresses
- Selector criteria that specifies the label value key pair to use to select the Kubernetes resources that you want to associate with the SNAT IP address
- A namespace name to limit the scope of label matching
- destIp CIDRs, which limit the traffic to the CIDRs

You can apply policy with different combinations of SNAT IP addresses, namespace, or label selectors to achieve different outcomes.

The following table shows the different ways that you can apply SNAT to Kubernetes resources, and the results of each combination.

**Note** If multiple SNAT policies are applied to a pod, the order of specificity in choosing the SNAT IP address is as follows, from the most specific to least specific:

- Pod
- Service
- Deployment
- Namespace
- Cluster

| SNAT IP | Namespace | Label | Result |
|---|---|---|---|
| Yes | Yes | Yes | Applies the specified SNAT IP address to any pods in the chosen namespace on which the label is applied, or to pods belonging to deployments in the chosen namespace on which label is applied.<br>**Note** If you want to apply SNAT to all pods in the namespace, you must apply the specified label to the corresponding namespace. |
| Yes | Yes | No | Applies the specified SNAT IP address to every pod in the namespace. |
| Yes | No | Yes | Applies the specified SNAT IP address to any pods on which label is applied, or to pods belonging to deployments or namespace on which label is applied. |
| Yes | No | No | Applies specified the SNAT IP address to every pod in the cluster. |
| No | Yes | Yes | You must apply SNAT to every pod that is an endpoint of a service on which the label is applied in a chosen namespace. You must apply SNAT with the external IP address of the corresponding service. |
| No | Yes | No | You must apply SNAT to every pod that is an endpoint of a service in the chosen namespace. You must apply SNAT with the external IP address of the corresponding service. |
| No | No | Yes | You must apply SNAT to ever pod that is the endpoint of any service in the cluster. You must apply SNAT with the external IP address of the corresponding service. |
| No | No | No | Invalid input |

## Configure the SNAT Policy Resource

You define the Source Network Address Translation (SNAT) traffic from a pod by creating a custom resource called `snatpolicy`. You can use the definition for different Kubernetes elements, such as pods and deployments.

### Before you begin

Fulfill all the prerequisites in the section Prerequisites for Configuring SNAT, on page 3 in this guide.

### Procedure

Define the `snatpolicy` resource.

```
apiVersion: aci.snat/v1
kind: SnatPolicy
metadata:
  name: <aName>
spec:
  selector:
    namespace: testns
    labels:
      <snat-label-key>: <snat-label-value>
  snatIp:
  - <ip or subnet>
  - ...
  destIp:
  - <ip or subnet>
  - ...
```

**Example:**

```
apiVersion: aci.snat/v1
kind: SnatPolicy
metadata:
  name: my-snat-name
spec:
  selector:
    namespace: testns
    labels:
      my-snat-label: backend-apps
  snatIp:
  - 10.20.30.40
  destIp:
  - 100.100.100.100/24
```

The preceding example sets the SNAT IP address (`snatIP`) to **`10.20.30.40`**, the namespace to **`testns`**, and the label selector criteria to:

```
my-snat-label: backend-apps
```

To apply this policy to a resource, add the label **`my-snat-label=backend-apps`** to the resource specification.

`destIp` limits the traffic to specified destinations. In this preceding example we are restricting the traffic to 100.100.100.0/24 network.

### What to do next

You can apply the label in the preceding example to specifications for the pod, deployment, and other elements. See the following sections for examples.

## Configure SNAT for a Pod

In the pod configuration, you can set a label to apply Source Network Address Translation to it. You set the label when you defined the `snatpolicy`. In the example, the label is `my-snat-label: backend-apps`.

### Before you begin

- Fulfill all the prerequisites in the section Prerequisites for Configuring SNAT, on page 3 in this guide.
- Define the Source Network Address Translation (SNAT) traffic from a pod by creating a custom resource.

  See the section Configure the SNAT Policy Resource, on page 6 in this guide.

### Procedure

Set the label to the pod configuration.

**Example:**

```
apiVersion: v1
kind: Pod
metadata:
  name: busybox
  labels:
    my-snat-label: backend-apps
  namespace: default
spec:
  containers:
  - image: busybox
    command:
      - sleep
      - "3600"
    imagePullPolicy: IfNotPresent
    name: busybox
  restartPolicy: Always
```

## Set SNAT for a Deployment

In the deployment configuration, you can set a label to apply Source Network Address Translation (SNAT) policy to it. You set the label when you defined the `snatpolicy`. In the example, the label is `my-snat-label: backend-apps`.

### Before you begin

- Fulfill all the prerequisites in the section Prerequisites for Configuring SNAT, on page 3 in this guide.
- Define the (SNAT) traffic from a pod by creating a custom resource.

  See the section Configure the SNAT Policy Resource, on page 6 in this guide.

### Procedure

Set the label to the deployment configuration.

**Example:**

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  labels:
    app: redis
    role: master
    tier: backend
    my-snat-label: backend-apps
  name: redis-master
  namespace: default
  progressDeadlineSeconds: 600
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app: redis
      role: master
      tier: backend
  strategy:
    rollingUpdate:
```

```
      maxSurge: 1
      maxUnavailable: 1
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: redis
        role: master
        tier: backend
    spec:
      containers:
      - image: gcr.io/google_containers/redis:e2e
        imagePullPolicy: IfNotPresent
        name: master
        ports:
        - containerPort: 6379
          protocol: TCP
        resources:
          requests:
            cpu: 100m
            memory: 100Mi
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
```

## Configure SNAT for a Namespace

In the namespace configuration, you can set a label to apply Source Network Address Translation (SNAT) to it. You set the label when you defined the `snatpolicy`. In the example, the label is `my-snat-label: backend-apps`.

### Before you begin

- Fulfill all the prerequisites in the section Prerequisites for Configuring SNAT, on page 3 in this guide.
- Define the SNAT traffic from a pod by creating a custom resource.

  See the section Configure the SNAT Policy Resource, on page 6 in this guide.

### Procedure

Set the label to the namespace configuration.

**Example:**

```
apiVersion: v1
kind: Namespace
metadata:
  labels:
    my-snat-label: backend-apps
  name: foo
spec:
  finalizers:
  - kubernetes
```

## Configure SNAT for a Service

You can request Source Network Address Translation (SNAT) for pods that are endpoints for a Kubernetes service, specifying that the external IP address of the service be used as the SNAT IP. You do so by creating a `snatpolicy` without specifying the SNAT IP address.

### Before you begin

- Fulfill all the prerequisites in the section Prerequisites for Configuring SNAT, on page 3 in this guide.
- Define the SNAT traffic from a pod by creating a custom resource.

  See the section Configure the SNAT Policy Resource, on page 6 in this guide.

### Procedure

**Step 1** Request SNAT for pods that are endpoints for a Kubernetes service, specifying use of the service IP address:

**Example:**

```
apiVersion: aci.snat/v1
kind: SnatPolicy
metadata:
  name: <my-snat-name>
spec:
  Selector:
    namespace: testns
    labels:
      my-snat-label: backend-apps
```

**Step 2** Add the label to the service specification.

**Example:**

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app: guestbook
    my-snat-label: backend-apps
    tier: frontend
  name: frontend
  namespace: default
spec:
  externalTrafficPolicy: Cluster
  ports:
  - nodePort: 30601
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: guestbook
    tier: frontend
  sessionAffinity: None
  type: LoadBalancer
```

## Configure SNAT for a Cluster

You can request Source Network Address Translation (SNAT) for all pods in a cluster by creating a `snatpolicy` without specifying a selector.

**Note** Make sure that the default port allocation is large enough for the expected connection load.

### Before you begin

- Fulfill all the prerequisites in the section Prerequisites for Configuring SNAT, on page 3 in this guide.
- Define the SNAT traffic from a pod by creating a custom resource.

  See the section Configure the SNAT Policy Resource, on page 6 in this guide.

### Procedure

Request SNAT for all pods in a cluster.

**Example:**

```
apiVersion: aci.snat/v1
kind: SnatPolicy
spec:
  snatIp:
  - 10.20.30.40
  destIp:
  - 100.100.100.0/24 <if we need to limit the traffic otherwise don’t specify the destIp section>
```

## List SNAT IP Address Allocated to All the Nodes

### Procedure

To validate or find the specific Source Network Address Translation (SNAT) IP addresses that are allocated to nodes for each SNAT policy, use the following kubectl command:

```
kubectl get snatglobalinfo --all-namespaces -o json
{
    "apiVersion": "v1",
    "items": [
        {
            "apiVersion": "aci.snat/v1",
            "kind": "SnatGlobalInfo",
            "metadata": {
                "creationTimestamp": "2020-05-14T03:05:45Z",
                "generation": 5,
                "name": "snatglobalinfo",
                "namespace": "aci-containers-system",
                "resourceVersion": "16565955",
                "selfLink":
"/apis/aci.snat/v1/namespaces/aci-containers-system/snatglobalinfos/snatglobalinfo",
                "uid": "9082ecfa-3612-447c-adb5-3ff3c0ab26cd"
            },
            "spec": {
                "globalInfos": {
                    "k8s24-node-1.local.lan": [
```

```
                        {
                            "macAddress": "00:50:56:97:a2:39",
                            "portRanges": [
                                {
                                    "end": 16999,
                                    "start": 14000
                                }
                            ],
                            "protocols": null,
                            "snatIp": "10.10.0.8",
                            "snatIpUid": "00000000-0000-0000-0000-ffff0a0a0008",
                            "snatPolicyName": "it-ns-ns-snatpolicy"
                        }
                    ],
                    "k8s24-node-2.local.lan": [
                        {
                            "macAddress": "00:50:56:97:40:68",
                            "portRanges": [
                                {
                                    "end": 13999,
                                    "start": 11000
                                }
                            ],
                            "snatIp": "10.10.0.8",
                            "snatIpUid": "00000000-0000-0000-0000-ffff0a0a0008",
                            "snatPolicyName": "it-ns-ns-snatpolicy"
                        }
                    ],

    "kind": "List",
    "metadata": {
        "resourceVersion": "",
        "selfLink": ""
    }
}
```

## Configure the Routing Domain CRD

This custom resource definition (CRD) allows you to specify one or more subnets for which Source Network Address Translation (SNAT) should not be performed (when traffic originates from a pod in the cluster). You must provide the subnet in CIDR form.

### Procedure

configure the routing domain CRD:

```
kubectl describe rdconfig -n aci-containers-system
Name:         routingdomain-config
Namespace:    aci-containers-system
Labels:       <none>
Annotations:  <none>
API Version:  aci.snat/v1
Kind:         RdConfig
Metadata:
  Creation Timestamp:  2020-05-14T03:05:46Z
  Generation:          33
  Resource Version:    16572514
  Self Link:          /apis/aci.snat/v1/namespaces/aci-containers-system/rdconfigs/routingdomain-config

  UID:                 52b121fa-9be3-4aed-bc97-8223cc2f8f04
Spec:
  Discoveredsubnets:
    10.2.0.1/16
```

```
    192.168.24.1/24
    10.2.0.1/16
    192.168.24.1/24
    10.5.0.1/24
  Usersubnets:  <nil>
Status:
```

**Americas Headquarters**
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

**Asia Pacific Headquarters**
CiscoSystems(USA)Pte.Ltd.
Singapore

**Europe Headquarters**
CiscoSystemsInternationalBV
Amsterdam,TheNetherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.