NSX Container Plugin 2.4.1 Release Notes

VMware NSX Container Plugin 2.4.1 | 9 May, 2019

Check regularly for additions and updates to this document.

What's in the Release Notes

The release notes cover the following topics:

What's New

NSX Container Plugin (NCP) 2.4.1 has the following new features:

- Use of a single distributed firewall section for healthcheck
  Use a single distributed firewall section per cluster to include all the firewall rules needed for pods with liveness probe and readiness probe. The limit is a maximum of 1000 pods with liveness probe or readiness probe in a cluster because there can be at most 1000 rules in a distributed firewall section.
- Make NSX Node Agent handle the unexpected termination of the privsep daemon
  NSX Node Agent has been enhanced to handle and recover from an unexpected privsep daemon termination.
- Define a maximum limit for Kubernetes service autoscaling
  With a new NCP configMap option, max_allowed_virtual_servers, users can define the maximum virtual servers that are allowed to be created within the cluster.
- Ability to assign a specific IP for Kubernetes Ingress
  Users can assign an IP address to Ingresses by using the option http_and_https_ingress_ip in NCP configMap.
- Ability to set X-Forwarded-For for Kubernetes ingress
- Ability to set Kubernetes Ingress Persistence Timeout
  An NCP configMap option, l7_persistence_timeout, was added to control the timeout on the persistence profile for the layer 7 virtual servers backing Kubernetes Ingresses.
- Support for Kubernetes service of type NodePort
  NoderPort allows a Kubernetes service to be accessed from outside the cluster. kube-proxy automatically configures the VM host to relay the traffic to the Pod. Proper iptables rule should be configured on the VM host to allow the forwarding to happen (for example, iptables -I FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT). If target Pods are isolated by Kubernetes network policy, the administrator should configure the network policy to allow traffic

from host IP CIDR to access service in the Pod, then NCP automatically adds the respective firewall rules to allow the traffic to pass through.

Compatibility Requirements

Product	Version
NCP / NSX-T Tile for PAS	2.4.1
NSX-T	2.3.1, 2.4.0.1, 2.4.1
Kubernetes	1.13, 1.14
OpenShift	3.11
Kubernetes Host VM OS	Ubuntu 16.04, CentOS 7.5, CentOS 7.6
OpenShift Host VM OS	RHEL 7.6
OpenShift BMC	RHEL 7.6
PAS (PCF)	Ops Manager 2.6 + PAS 2.6 Ops Manager 2.5 + PAS 2.5 Ops Manager 2.4 + PAS 2.4

Known Issues

- **Issue 2118515: In a large-scale setup, NCP takes a long time to create firewalls on NSX-T**
  In a large-scale setup (for example, 250 Kubernetes nodes, 5000 pods, 2500 network policies), it can take NCP a few minutes to create the firewall sections and rules in NSX-T.

  Workaround: None. After the firewall sections and rules are created, performance should be back to normal.

- **Issue 2125755: A StatefullSet could lose network connectivity when performing canary updates and phased rolling updates**
  If a StatefulSet was created before NCP was upgraded to the current release, the StatefullSet could lose network connectivity when performing canary updates and phased rolling updates.

  Workaround: Create the StatefulSet after NCP is upgraded to the current release.

- **Issue 2131494: NGINX Kubernetes Ingress still works after changing the Ingress class from nginx to nsx**
  When you create an NGINX Kubernetes Ingress, NGINX create traffic forwarding rules. If you change the Ingress class to any other value, NGINX does not delete the rules and continues to apply them, even if you delete the Kubernetes Ingress after changing the class. This is a limitation of NGINX.

Workaround: To delete the rules created by NGINX, delete the Kubernetes Ingress when the class value is nginx. Than re-create the Kubernetes Ingress.

- **For a Kubernetes service of type ClusterIP, Client-IP based session affinity is not supported**
  NCP does not support Client-IP based session affinity for a Kubernetes service of type ClusterIP.

  Workaround: None

- **For a Kubernetes service of type ClusterIP, the hairpin-mode flag is not supported**
  NCP does not support the hairpin-mode flag for a Kubernetes service of type ClusterIP.

  Workaround: None

- **Issue 2193901: Multiple PodSelectors or multiple NsSelectors for a single Kubernetes network policy rule is not supported**
  Applying multiple selectors allows only incoming traffic from specific pods.

  Workaround: Use matchLabels with matchExpressions in a single PodSelector or NsSelector instead.

- **Issue 2194646: Updating network policies when NCP is down is not supported**
  If you update a network policy when NCP is down, the destination IPset for the network policy will be incorrect when NCP comes back up.

  Workaround: Recreate the network policy when NCP is up.

- **Issue 2192489: After disabling 'BOSH DNS server' in PAS director config, the Bosh DNS server (169.254.0.2) still appears in the container's resolve.conf file.**
  In a PAS environment running PAS 2.2, after you disable 'BOSH DNS server' in PAS director config, the Bosh DNS server (169.254.0.2) still appears in the container's resove.conf file. This causes a ping command with a fully qualified domain name to take a long time. This issue does not exist with PAS 2.1.

  Workaround: None. This is a PAS issue.

- **Issue 2199504: The display name of NSX-T resources created by NCP is limited to 80 characters**
  When NCP creats an NSX-T resouce for a resource in the container environment, it generates the display name of the NSX-T resource by combining the cluster name, namespace or project name, and the name of the resource in the container environment. If the display name is longer than 80 characters, it is truncated to 80 characters.

  Workaround: None

- **Issue 2199778: With NSX-T 2.2, Ingress, Service and Secrets with names longer than 65 characters are not supported**
  With NSX-T 2.2, when use_native_loadbalancer is set to True, the names of Ingresses, Secrets and Services referenced by the Ingress, and Services of type LoadBalancer, must be 65 characters or less. Otherwise, the Ingress or Service will not work properly.

Workaround: When configuring an Ingress, Secret, or Service, specify a name that is 65 characters or less.

- **Issue 2065750: Installing the NSX-T CNI package fails with a file conflict**
  In a RHEL environment with kubernetes installed, if you install the NSX-T CNI Package using `yum localinstall` or `rpm -i`, you get an error indicating a conflict with a file from the kubernetes-cni package.

  Workaround: Install the NSX-T CNI package with the command `rpm -i --replacefiles nsx-cni-2.3.0.xxxxxxxx-1.x86_64.rpm`.

- **Issue 2224218: After a service or app is deleted, it takes 2 minutes to release the SNAT IP back to the IP pool**
  If you delete a service or app and recreate it within 2 minutes, it will get a new SNAT IP from the IP pool.

  Workaround: After deleting a service or app, wait 2 minutes before recreating it if you want to reuse the same IP.

- **Issue 2330811: When creating Kubernetes services of type LoadBalancer while NCP is down, the services might not get created when NCP is restarted**
  When NSX-T resources are exhausted for Kubernetes services of type LoadBalancer, you can create new services after deleting some of the existing services. However, if you delete and create the services while NCP is down, NCP will fail to create the new services.

  Workaround: When NSX-T resources are exhausted for Kubernetes services of type LoadBalancer, do not perform both the delete and the create operations while NCP is down.

- **Issue 2317608: Multiple CNI plugins not supported**
  Kubernetes expects a CNI configuration file of type .conflist containing a list of plugin configurations. The kubelet will call the plugins defined in this conflist file one by one in the order defined. Currently, the nsx-cf-cni bosh release only supports a single CNI plugin configuration. Any additional CNI plugin will overwrite the existing CNI configuration file 10-nsx.conf in the specified cni_config_dir.

  Workaround: None. This issue is fixed in NCP 2.5.

- **Issue 2389094: When NCP deletes a load balancer server, the corresponding tier-1 router is not deleted**
  With automatic scaling enabled, if you create multiple services of type LoadBalancer, NCP creates the required number of load balancer virtual servers. If you then decrease the number of services, and as a result NCP deletes a load balancer virtual server, the corresponding tier-1 router is not deleted.

  Workaround: None.