DNS Operator in OpenShift Container Platform

The DNS Operator deploys and manages Core DNS to provide a name resolution service to pods, enabling DNS-based Kubernetes Service discovery in OpenShift Container Platform.

DNS Operator

The DNS Operator implements the dns API from the operator.openshift.io API group. The Operator deploys Core DNS using a daemon set, creates a service for the daemon set, and configures the kubelet to instruct pods to use the Core DNS service IP address for name resolution.

**Procedure**

The DNS Operator is deployed during installation with a Deployment object.

1. Use the oc get command to view the deployment status:

$ oc get -n openshift-dns-operator deployment/dns-operator

1. Use the oc get command to view the state of the DNS Operator:

$ oc get clusteroperator/dns

AVAILABLE, PROGRESSING and DEGRADED provide information about the status of the operator. AVAILABLE is True when at least 1 pod from the Core DNS daemon set reports an Available status condition.

Changing the DNS Operator managementState

DNS manages the Core DNS component to provide a name resolution service for pods and services in the cluster. The managementState of the DNS Operator is set to Managed by default, which means that the DNS Operator is actively managing its resources. You can change it to Unmanaged, which means the DNS Operator is not managing its resources.

The following are use cases for changing the DNS Operator managementState:

* You are a developer and want to test a configuration change to see if it fixes an issue in Core DNS. You can stop the DNS Operator from overwriting the fix by setting the managementState to Unmanaged.
* You are a cluster administrator and have reported an issue with Core DNS, but need to apply a workaround until the issue is fixed. You can set the managementState field of the DNS Operator to Unmanaged to apply the workaround.

**Procedure**

* Change managementState DNS Operator:

oc patch dns.operator.openshift.io default --type merge --patch '{"spec":{"managementState":"Unmanaged"}}'

Controlling DNS pod placement

The DNS Operator has two daemon sets: one for Core DNS and one for managing the /etc/hosts file. The daemon set for /etc/hosts must run on every node host to add an entry for the cluster image registry to support pulling images. Security policies can prohibit communication between pairs of nodes, which prevents the daemon set for Core DNS from running on every node.

As a cluster administrator, you can use a custom node selector to configure the daemon set for CoreDNS to run or not run-on certain nodes.

**Prerequisites**

* You installed the oc CLI.
* You are logged in to the cluster with a user with cluster-admin privileges.

**Procedure**

* To prevent communication between certain nodes, configure the spec.nodePlacement.nodeSelector API field:
  1. Modify the DNS Operator object named default:

$ oc edit dns.operator/default

* 1. Specify a node selector that includes only control plane nodes in the spec.nodePlacement.nodeSelector API field:
  2. spec:
  3. nodePlacement:
  4. nodeSelector:

node-role.kubernetes.io/worker: ""

* To allow the daemon set for CoreDNS to run on nodes, configure a taint and toleration:
  1. Modify the DNS Operator object named default:

$ oc edit dns.operator/default

* 1. Specify a taint key and a toleration for the taint:
  2. spec:
  3. nodePlacement:
  4. tolerations:
  5. - effect: NoExecute
  6. key: "dns-only"
  7. operators: Equal
  8. value: abc

tolerationSeconds: 3600 **1**

[**1**](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.9/html/networking/dns-operator#CO3-1)

If the taint is dns-only, it can be tolerated indefinitely. You can omit tolerationSeconds.

View the default DNS

Every new OpenShift Container Platform installation has a dns.operator named default.

**Procedure**

1. Use the oc describe command to view the default dns:

$ oc describe dns.operator/default

**Example output**

Name: default

Namespace:

Labels: <none>

Annotations: <none>

API Version: operator.openshift.io/v1

Kind: DNS

...

Status:

Cluster Domain: cluster.local **1**

Cluster IP: 172.30.0.10 **2**

...

[**1**](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.9/html/networking/dns-operator#CO4-1)

The Cluster Domain field is the base DNS domain used to construct fully qualified pod and service domain names.

[**2**](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.9/html/networking/dns-operator#CO4-2)

The Cluster IP is the address pods query for name resolution. The IP is defined as the 10th address in the service CIDR range.

1. To find the service CIDR of your cluster, use the oc get command:

$ oc get networks.config/cluster -o jsonpath='{$.status.serviceNetwork}'

Using DNS forwarding

You can use DNS forwarding to override the forwarding configuration identified in /etc/resolv.conf on a per-zone basis by specifying which name server should be used for a given zone. If the forwarded zone is the Ingress domain managed by OpenShift Container Platform, then the upstream name server must be authorized for the domain.

**Procedure**

1. Modify the DNS Operator object named default:

$ oc edit dns.operator/default

This allows the Operator to create and update the ConfigMap named dns-default with additional server configuration blocks based on Server. If none of the servers has a zone that matches the query, then name resolution falls back to the name servers that are specified in /etc/resolv.conf.

**Sample DNS**

apiVersion: operator.openshift.io/v1

kind: DNS

metadata:

name: default

spec:

servers:

- name: foo-server **1**

zones: **2**

- example.com

forwardPlugin:

upstreams: **3**

- 1.1.1.1

- 2.2.2.2:5353

- name: bar-server

zones:

- bar.com

- example.com

forwardPlugin:

upstreams:

- 3.3.3.3

- 4.4.4.4:5454

[**1**](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.9/html/networking/dns-operator#CO5-1)

name must comply with the rfc6335 service name syntax.

[**2**](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.9/html/networking/dns-operator#CO5-2)

zones must conform to the definition of a subdomain in rfc1123. The cluster domain, cluster.local, is an invalid subdomain for zones.

[**3**](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.9/html/networking/dns-operator#CO5-3)

A maximum of 15 upstreams is allowed per forwardPlugin.

**Note**

If servers is undefined or invalid, the ConfigMap only contains the default server.

1. View the ConfigMap:

$ oc get configmap/dns-default -n openshift-dns -o yaml

**Sample DNS ConfigMap based on previous sample DNS**

apiVersion: v1

data:

Corefile: |

example.com:5353 {

forward . 1.1.1.1 2.2.2.2:5353

}

bar.com:5353 example.com:5353 {

forward . 3.3.3.3 4.4.4.4:5454 **1**

}

.:5353 {

errors

health

kubernetes cluster.local in-addr.arpa ip6.arpa {

pods insecure

upstream

fallthrough in-addr.arpa ip6.arpa

}

prometheus :9153

forward . /etc/resolv.conf {

policy sequential

}

cache 30

reload

}

kind: ConfigMap

metadata:

labels:

dns.operator.openshift.io/owning-dns: default

name: dns-default

namespace: openshift-dns

[**1**](https://access.redhat.com/documentation/en-us/openshift_container_platform/4.9/html/networking/dns-operator#CO6-1)

Changes to the forwardPlugin triggers a rolling update of the CoreDNS daemon set.

**ADDITIONAL RESOURCES**

* For more information on DNS forwarding, see the [CoreDNS forward documentation](https://coredns.io/plugins/forward/).

DNS Operator status

You can inspect the status and view the details of the DNS Operator using the oc describe command.

**Procedure**

View the status of the DNS Operator:

$ oc describe clusteroperators/dns

DNS Operator logs

You can view DNS Operator logs by using the oc logs command.

**Procedure**

View the logs of the DNS Operator:

$ oc logs -n openshift-dns-operator deployment/dns-operator -c dns-operator

# NTP Deployment

OpenShift Enterprise requires NTP to synchronize the system and hardware clocks. This synchronization is necessary for communication between the broker and node hosts; if the clocks are not synchronized correctly, messages are dropped by MCollective. It is also helpful to have accurate time stamps on files and in log file entries.

On each host, use the ntpdate command to set the system clock, replacing the NTP servers to suit your environment:

# ntpdate clock.redhat.com

You must also configure the /etc/ntp.conf file to keep the clock synchronized during operation.

If the error message "the NTP socket is in use, exiting" is displayed after running the ntpdate command, it means that the ntpd daemon is already running. However, the clock may not be synchronized due to a substantial time difference. In this case, run the following commands to stop the ntpd service, set the clock, and start the service again:

# service ntpd stop

# ntpdate clock.redhat.com

# service ntpd start

If you are installing OpenShift Enterprise on physical hardware, use the hwclock command to synchronize the hardware clock to the system clock. Skip this step if you are installing on a virtual machine, such as an Amazon EC2 instance. For a physical hardware installation, run the following command:

# hwclock --systohc