



Ingress Operator degraded with CanaryChecksRepetitiveFailures

👍 SOLUTION VERIFIED - Updated December 23 2022 at 3:19 PM - English ▾

Environment

- Red Hat OpenShift Container Platform 4.x

Issue

- Ingress Operator degraded due to CanaryChecksRepetitiveFailures:

```
The "default" ingress controller reports Degraded=True: DegradedConditions: One or more other status conditions indicate a degraded state: CanaryChecksSucceeding=False (CanaryChecksRepetitiveFailures: Canary route checks for the default ingress controller are failing)
```

Resolution

- Removing unnecessary NetworkPolicy from namespace openshift-ingress-operator helped to restore connectivity.
- Ingress cluster operator became Ready since connectivity is established to the canary route.

Root Cause

- There was a NetworkPolicy in the openshift-ingress-operator namespace which was blocking egress traffic from the ingress-operator-XXXX pod to another namespace.

```

apiVersion: networking.k8s.io/v1
items:
- apiVersion: networking.k8s.io/v1
  kind: NetworkPolicy
  metadata:
    name: open
    namespace: openshift-ingress-operator
  spec:
    egress:
    - to:
      - namespaceSelector: {}
        podSelector: {}
    ingress:
    - from:
      - namespaceSelector: {}
        podSelector: {}
    podSelector: {}
    policyTypes:
    - Ingress
    - Egress
kind: NetworkPolicyList

```

- From OpenShift 4.10 OpenShift-sdn started supporting egress NetworkPolicy but in the previous version, it does not so there was no impact of this NetworkPolicy.
- Issue was seen after the upgrade from 4.9 to 4.10.

Diagnostic Steps

- Ingress cluster Operator degraded due to failed canary route check.

The "default" ingress controller reports Degraded=True: DegradedConditions: One or more other status conditions indicate a degraded state: CanaryChecksSucceeding=False (CanaryChecksRepetitiveFailures: Canary route checks for the default ingress controller are failing)

- The connectivity to the canary pod from ingress-operator-XXXX pod running in openshift-ingress-operator was failing.

```

$ oc exec -n openshift-ingress-operator ingress-operator-XXXX -- curl
http://canary-openshift-ingress-canary.apps.example.com --resolve canary-
openshift-ingress-canary.apps.example.com:80:10.1.x.y >
ingress_operator_canary_bypass_router.txt
Defaulted container "ingress-operator" out of: ingress-operator, kube-rbac-proxy
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %         %         Dload  Upload  Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0  --:--:--  0:02:09  --:--:--
0curl: (7) Failed to connect to canary-openshift-ingress-canary.apps.example.com
port 80: Connection timed out
command terminated with exit code 7

```

This solution is part of Red Hat's fast-track publication program, providing a huge library of solutions that Red Hat engineers have created while supporting our customers. To give you the knowledge you need the instant it becomes available, these articles may be presented in a raw and unedited form.

People who viewed this solution also viewed

[Upgrade process fails in Openshift Container Platform 4 because the Ingress Operator failed: CanaryChecksRepetitiveFailures](#)

Solution - 14. Apr. 2023

[`Ingress Operator failed: CanaryChecksRepetitiveFailures` during upgrade when ingress sharding is used](#)

Solution - 8. März 2023

[mTLS breaks ingress canary check in OCP 4](#)

Solution - 21. Dez. 2021

Comments