



Part 15: Kubernetes Real-Time Troubleshooting

Introduction

Welcome to the world of Kubernetes troubleshooting, where every challenge is an opportunity to sharpen your skills and emerge victorious. Join us as we embark on a journey through common real-time scenarios, unraveling mysteries, and uncovering solutions along the way.

PART 15- KUBERNETES REAL-TIME TROUBLESHOOTING

 **kubernetes**

- CronJob Not Executing Issue
- Node Port Service not accessible
- Service Account Permission Issue
- Network Policy Blocking Traffic
- Ingress Controller Not Routing traffic

Scenario 71: CronJob Not Executing

kubernetes/kubernetes

#93578 Job not running on scheduled time - Kubernetes CronJob

 5 comments



caslurjf opened on July 30, 2020





Symptoms: Scheduled CronJobs are not running at the expected times.

Diagnosis: Check the status of the CronJob (`kubectl get cronjob <cronjob_name>`) and describe it for detailed information (`kubectl describe cronjob <cronjob_name>`).

Solution:

1. Verify the schedule format and ensure it is correctly defined in cron syntax.
2. Check the Kubernetes controller manager logs for errors related to CronJobs.
3. Ensure the time zone of the Kubernetes cluster aligns with the CronJob schedule.

Scenario 72: Node Port Service Not Accessible

kubernetes-sigs/kind

#808 NodePort service is not reachable outside cluster using...



10 comments



gagara opened on August 23, 2019



Symptoms: Services exposed via NodePort are not accessible from outside the cluster.

Diagnosis: Describe the NodePort service (`kubectl describe svc <service_name>`) and verify node firewall and network configurations.

Solution:

1. Ensure that firewall rules on the nodes allow traffic to the specified NodePort range.
2. Verify that the nodes are reachable from the client's network.
3. Consider using LoadBalancer services if NodePort accessibility is problematic.

Scenario 73: Service Account Permission Issues

Symptoms: Pods fail to perform actions due to insufficient permissions associated with their service accounts.

Diagnosis: Describe the affected pods (`kubectl describe pod <pod_name>`) and review the associated service accounts and role bindings.



kubernetes/kubernetes

#48208 Insufficient permissions for RBAC role system:kube-...



6 comments



antoin eco opened on June 28, 2017



Solution:

1. Update RoleBindings or ClusterRoleBindings to grant necessary permissions to the service accounts.
2. Ensure that the service accounts are correctly specified in the pod specifications.
3. Use the principle of least privilege to grant only required permissions.

Scenario 74: Network Policy Blocking Traffic

```
student-node ~ → kubectl describe networkpolicies.networking.k8s.io cyan-np-cka28-trb -n cyan-ns-cka28-trb
Name:          cyan-np-cka28-trb
Namespace:     cyan-ns-cka28-trb
Created on:   2023-04-23 08:38:19 +0000 UTC
Labels:        <none>
Annotations:   <none>
Spec:
  PodSelector:    app=cyan-app-cka28-trb
  Allowing ingress traffic:
    To Port: 80/TCP
    From:
      NamespaceSelector: test=work
      PodSelector: app=cyan-white-cka28-trb1
  Not affecting egress traffic
  Policy Types: Ingress
```

Symptoms: Pods cannot communicate due to restrictive Network Policies.

Diagnosis: Describe the network policies (`kubectl describe networkpolicy <network_policy_name>`) and check pod labels and policy selectors.



Solution:

1. Review and adjust Network Policies to allow required traffic.
2. Ensure that pod labels and selectors match the intended traffic rules.
3. Test connectivity with and without Network Policies to isolate issues.

Scenario 75: Ingress Controller Not Routing Traffic

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: manipmcv
  namespace: ingress-nginx
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - host: application.eastus.cloudapp.azure.com
    http:
      paths:
      - path: /app1
        pathType: Prefix
      backend:
        service:
          name: manipmcv-service
          port:
            name: http
```

Symptoms: Ingress resources are created, but traffic is not being routed to the appropriate backend services.

Diagnosis: Check the Ingress controller logs and describe the Ingress resource (`kubectl describe ingress <ingress_name>`).

Solution:

1. Ensure the Ingress controller is properly deployed and running (`kubectl get pods -n <ingress_namespace>`).
2. Verify that the Ingress resource is correctly configured with proper rules and paths.
3. Check for any conflicting Ingress resources and resolve conflicts.



In the up-coming parts, we will discuss on more troubleshooting steps for the different Kubernetes based scenarios. So, stay tuned for the and follow @Prasad Suman Mohan for more such posts.

