Traefik API Gateway and Nginx Ingress Troubleshooting Runbook

Traefik and Nginx Ingress are popular solutions for managing API gateways and ingress controllers in Kubernetes. This runbook will help you troubleshoot common issues with Traefik and Nginx Ingress in an AWS environment using kubectl commands and other relevant tools.

Step 1: Verify kubectl Configuration Ensure that kubectl is properly configured to interact with the target Kubernetes cluster.

bash

Copy code

```bash
kubectl version
kubectl get nodes
```

Step 2: Check Ingress Controllers Confirm that the Traefik and Nginx Ingress controllers are deployed and running.

bash

Copy code

```bash
# For Traefik
kubectl get pods -n traefik-system

# For Nginx Ingress
kubectl get pods -n nginx-ingress
```

Step 3: Review Ingress Resources Check the Ingress resources in the namespace where your services are running.

bash

Copy code

```bash
kubectl get ingress -n YOUR_NAMESPACE
kubectl describe ingress YOUR_INGRESS_NAME -n YOUR_NAMESPACE
```

Step 4: Verify DNS Resolution Ensure that the Ingress hostname/domain resolves to the correct IP address.

bash

Copy code

```
# Use nslookup or dig to verify DNS resolution
nslookup YOUR_INGRESS_HOSTNAME
```

Step 5: Check Service Endpoints Inspect the endpoints associated with your services.

bash

Copy code

```
kubectl describe endpoints YOUR_SERVICE_NAME -n YOUR_NAMESPACE
```

Step 6: Review Ingress Annotations Inspect any annotations applied to the Ingress resources that might affect their behavior.

bash

Copy code

```
kubectl describe ingress YOUR_INGRESS_NAME -n YOUR_NAMESPACE | grep -i
"annotations"
```

Step 7: Check Path-Based Routing (Traefik) If you are using path-based routing with Traefik, verify the routing configuration.

bash

Copy code

```
kubectl get ingressroutes -n traefik-system
kubectl describe ingressroute YOUR_INGRESS_ROUTE_NAME -n traefik-system
```

Step 8: Inspect Traefik Dashboard (Traefik) If you are using Traefik, check its dashboard for any errors or information.

bash

Copy code

```
# Make sure you have Traefik dashboard enabled
kubectl port-forward -n traefik-system YOUR_TRAEFIK_POD_NAME 8080:8080
```

Open your web browser and navigate to `http://localhost:8080/dashboard/`.

Step 9: Review Nginx Configuration (Nginx Ingress) For Nginx Ingress, check the configuration of the Nginx server.

bash

Copy code

```
kubectl exec -n nginx-ingress YOUR_NGINX_POD_NAME -- cat /etc/nginx/nginx.conf
```

Step 10: Check AWS Load Balancer (If Applicable) If you are using an AWS Elastic Load Balancer (ELB) with your Ingress, check its status and configuration in the AWS Management Console or via the AWS CLI.

bash

Copy code

```
aws elbv2 describe-load-balancers --names YOUR_ELB_NAME
```

Step 11: Verify Security Group Rules Ensure that the Ingress controllers' security group rules allow the necessary traffic.

bash

Copy code

```
# For Traefik
aws ec2 describe-security-groups --group-ids YOUR_TRAEFIK_SECURITY_GROUP_ID

# For Nginx Ingress
aws ec2 describe-security-groups --group-ids YOUR_NGINX_INGRESS_SECURITY_GROUP_ID
```

Step 12: Check Pod and Service Logs Inspect the logs of the pods and services associated with the Ingress controllers for any errors or issues.

bash

Copy code

```
# For Traefik
kubectl logs YOUR_TRAEFIK_POD_NAME -n traefik-system
```

```
# For Nginx Ingress
kubectl logs YOUR_NGINX_POD_NAME -n nginx-ingress
```

Step 13: Check for AWS Support (If Required) If the issue persists or is beyond your expertise, consider reaching out to AWS Support for further assistance.