

AWS EKS Troubleshooting Runbook

Amazon Elastic Kubernetes Service (EKS) can encounter various issues due to misconfigurations, resource constraints, or other factors. This runbook will help you troubleshoot common EKS issues

Step 1: Identify the Issue

- Gather information about the symptoms, error messages, and any recent changes made to the EKS cluster or applications running on it.

Step 2: Check EKS Cluster Status Use the following AWS CLI command to describe the EKS cluster and check its status:

bash

[Copy code](#)

```
aws eks describe-cluster --name YOUR_CLUSTER_NAME --region YOUR_REGION
```

Step 3: Verify Nodes in the Cluster Use `eksctl` to check the status and health of nodes in the cluster:

bash

[Copy code](#)

```
eksctl get nodegroup --cluster YOUR_CLUSTER_NAME
```

Step 4: Review Cluster Events Check for any relevant events that may indicate issues in the cluster:

AWS Management Console:

- Navigate to the EKS cluster in the AWS Management Console.
- Under the "Monitoring" tab, review the "Events" section.

AWS CLI Command:

bash

Copy code

```
aws eks describe-cluster --name YOUR_CLUSTER_NAME --region YOUR_REGION --query  
"cluster.events"
```

Step 5: Check Kubectl Configuration Ensure that your kubectl is configured correctly to interact with the EKS cluster:

bash

Copy code

```
kubectl config get-contexts  
kubectl config current-context
```

If needed, use the `aws eks update-kubeconfig` command to update your kubeconfig.

Step 6: Check EKS Cluster and Node Logs Review the EKS control plane logs and the logs from the worker nodes for any errors:

AWS Management Console:

- In the EKS dashboard, select the cluster, go to the "Logging" tab, and view the control plane logs.

AWS CLI Command:

bash

Copy code

```
aws eks describe-cluster --name YOUR_CLUSTER_NAME --region YOUR_REGION --query  
"cluster.logging.clusterLogging"
```

Step 7: Verify IAM Roles and Permissions Ensure that IAM roles and permissions are correctly configured for the EKS cluster, worker nodes, and any applications running in the cluster.

Step 8: Check for Insufficient Resources Verify that the cluster has sufficient resources for its workload:

bash

[Copy code](#)

```
kubectl get nodes
```

Step 9: Review Security Group and Network Configurations Check the security group rules associated with the EKS nodes and ensure that networking is configured correctly.

[bash](#)

[Copy code](#)

```
aws ec2 describe-security-groups --group-ids YOUR_SECURITY_GROUP_ID
```

Step 10: Check for Application-Specific Issues Review logs and application-specific configurations to identify any issues related to the applications running in the EKS cluster.

Step 11: Restart Unresponsive Pods If specific pods are unresponsive, consider restarting them:

[bash](#)

[Copy code](#)

```
kubectl get pods  
kubectl delete pod YOUR_POD_NAME
```

Step 12: Check for AWS Support (If Required) If the issue persists or is beyond your expertise, consider reaching out to AWS Support for further assistance.