DevOps Shack

100 Kubernetes Errors

& Troubleshooting in Detail

1. Error: "Unable to connect to the server: dial tcp :443: connect: connection refused"

- Description: This error suggests that the `kubectl` command is unable to connect to the Kubernetes API server.
- Troubleshooting:
  - Check if the Kubernetes API server is running: `kubectl get pods --all-namespaces`
  - Verify the API server's endpoint: `kubectl cluster-info`
  - Check if there are any network issues between the client and the API server.
  - Ensure the kubeconfig file is correctly configured: `kubectl config view`

2. Error: "Error from server (NotFound): error when retrieving current configuration of..."

- Description: Indicates that the resource being queried does not exist.
- Troubleshooting:
  - Ensure the correct resource name is specified.
  - Verify the namespace if the resource is namespaced: `kubectl get <resource> -n <namespace>`
  - Check for typos in the resource name or kind.

3. Error: "Error: UPGRADE FAILED: failed to replace object: cannot patch..."

- Description: Occurs during a Helm chart upgrade when a resource cannot be updated.
- Troubleshooting:
  - Check if the resource being updated is immutable.
  - Ensure the Helm chart version is compatible with the Kubernetes version.
  - Verify permissions to modify resources.

4. Error: "error: the server doesn't have a resource type..."

- Description: Indicates that the requested resource type is not supported by the Kubernetes API server.
- Troubleshooting:
  - Check the API server version for compatibility.
  - Verify the resource type is spelled correctly.
  - Check if the API server supports custom resource definitions (CRDs) if using custom resources.

5. Error: "Failed to pull image..."

- Description: Indicates a failure to pull a container image from the specified registry.
- Troubleshooting:
  - Check if the image name and tag are correct.
  - Verify network connectivity to the image registry.
  - Ensure proper authentication to the image registry: `docker login <registry>`

6. Error: "Pod in Terminating state"

- Description: Occurs when a pod is stuck in the terminating state and not being removed.
- Troubleshooting:
  - Identify the reason for the termination using `kubectl describe pod <pod_name>`.
  - Check if any finalizers are preventing pod deletion: `kubectl get pod <pod_name> -o=jsonpath='{.metadata.finalizers}'`.

    - Manually force deletion of the pod: `kubectl delete pod <pod_name> --grace-period=0 --force`.

7. Error: "Insufficient memory"

- Description: Indicates that a container cannot allocate enough memory.
- Troubleshooting:
    - Check resource requests and limits in the pod specification.
    - Verify the node's memory usage: `kubectl top nodes`.
    - Scale the cluster or resize nodes to increase available memory.

8. Error: "Forbidden: pods is forbidden: User..."

- Description: Indicates that the user does not have permission to access pods.
- Troubleshooting:
    - Check RBAC rules to ensure the user has necessary permissions: `kubectl auth can-i list pods`.
    - Verify the correct kubeconfig context is being used.
    - Review cluster-level RBAC policies.

9. Error: "Failed to attach volume..."

- Description: Occurs when a volume cannot be attached to a pod.
- Troubleshooting:
    - Check if the volume plugin is properly configured on the node.
    - Verify the volume name and type in the pod specification.
    - Check if the volume is already in use by another pod.

10. Error: "No resources found in default namespace"

- Description: Indicates that no resources are present in the default namespace.
- Troubleshooting:
    - Specify a different namespace using `-n <namespace>`.
    - Check if resources exist in a different namespace.
    - Ensure the correct kubeconfig context is set.

11. Error: "Back-off restarting failed container"

- Description: Indicates that a container within a pod is repeatedly failing to start.
- Troubleshooting:
  - Check container logs for the specific error: `kubectl logs <pod_name> -c <container_name>`.
  - Verify resource constraints and requests are appropriate.
  - Investigate application code for issues.

12. Error: "Volume is already exclusively attached to one node and can't be attached to another"

- Description: Occurs when attempting to attach a volume that is already attached to another node.
- Troubleshooting:
  - Ensure the volume is detached from the previous node.
  - Check for orphaned resources in the cluster.
  - Verify the volume attachment status using `kubectl describe volumeattachment <volume_attachment_name>`.

13. Error: "Service Unavailable"

- Description: Indicates that the Kubernetes API server is temporarily unavailable.
- Troubleshooting:
  - Check API server logs for errors.
  - Verify network connectivity to the API server.
  - Monitor system resources on the Kubernetes control plane nodes.

14. Error: "Invalid value: must be no more than"

- Description: Indicates that a resource's value exceeds a specified limit.
- Troubleshooting:
  - Check resource quotas and limits in the namespace.
  - Review resource requests and limits in the pod specification.
  - Increase resource quotas if necessary.

15. Error: "Unable to mount volumes for pod"

- Description: Occurs when Kubernetes cannot mount volumes specified in the pod definition.
- Troubleshooting:
  - Verify volume specifications in the pod definition.
  - Check if the volume plugin is installed and configured on the node.
  - Ensure the volume exists and is accessible.

16. Error: "Error: image operating system..."

- Description: Indicates an incompatible operating system between the container image and the node OS.
- Troubleshooting:
  - Use images compatible with the node's operating system.
  - Verify the image's base OS using `docker inspect <image_name>`.

17. Error: "The connection to the server localhost:8080 was refused..."

- Description: Indicates the Kubernetes control plane is not running or not accessible.
- Troubleshooting:
  - Check if the Kubernetes control plane components are running: `kubectl get pods -n kube-system`.
  - Verify kubeconfig settings and the cluster context.
  - Restart the Kubernetes control plane components if necessary.

18. Error: "timed out waiting for the condition..."

- Description: Occurs when a resource fails to reach the desired state within the specified timeout.
- Troubleshooting:
  - Check the resource status and events: `kubectl describe <resource> <resource_name>`.
  - Verify the health of underlying components.
  - Adjust the timeout or investigate the cause of delays.

19. Error: "Error from server (BadRequest): a container name must be specified for pod..."

- Description: Indicates that a container name was not specified in the pod definition.
- Troubleshooting:
  - Ensure each container in the pod definition has a unique name.
  - Verify the pod specification for syntax errors.

20. Error: "unknown field 'replicas' in..."

- Description: Indicates an incorrect field in the resource definition, often caused by a version mismatch.
- Troubleshooting:
  - Verify the Kubernetes API version being used.
  - Check the resource definition against the correct API version.
  - Update the resource definition if necessary.

21. Error: "cannot create serviceaccount..."

- Description: Occurs when attempting to create a service account without sufficient permissions.
- Troubleshooting:
  - Ensure the user has RBAC permissions to create service accounts.
  - Check if there are any admission controllers blocking service account creation.
  - Verify the namespace for any namespace-scoped restrictions.

22. Error: "failed to find a usable init container"

- Description: Indicates that an init container specified in the pod definition cannot be scheduled.
- Troubleshooting:
  - Check init container readiness and liveness probes.
  - Verify the image name and tag for the init container.
  - Ensure the init container's resource requests and limits are appropriate.

23. Error: "the server could not find the requested resource"

- Description: Indicates that the requested resource does not exist.
- Troubleshooting:
    - Check the spelling of the resource name.
    - Verify if the resource belongs to the correct API group.
    - Ensure the correct namespace is specified.

24. Error: "invalid configuration: unable to load..."

- Description: Occurs when the kubeconfig file is incorrectly configured.
- Troubleshooting:
    - Verify the kubeconfig file path and permissions.
    - Check for syntax errors or missing configurations in the kubeconfig file.
    - Use the `KUBECONFIG` environment variable to specify the kubeconfig file.

25. Error: "Error from server (InternalError): an error on the server..."

- Description: Indicates an internal error on the Kubernetes API server.
- Troubleshooting:
    - Check Kubernetes API server logs for details.
    - Verify the health of the Kubernetes control plane components.
    - Restart the Kubernetes control plane components if necessary.

26. Error: "failed to start container..."

- Description: Occurs when a container fails to start within a pod.
- Troubleshooting:
    - Check container logs for errors: `kubectl logs <pod_name> -c <container_name>`.
    - Verify resource requests and limits for the container.
    - Investigate application code for issues.

27. Error: "unable to recognize..."

- Description: Indicates that the Kubernetes API server does not recognize the specified resource.
- Troubleshooting:
    - Check the API server logs for errors.

  - Verify the API server version for compatibility with the resource.
  - Ensure the correct API group is specified.

28. Error: "invalid namespace"

- Description: Occurs when specifying an invalid namespace.
- Troubleshooting:
  - Verify the namespace spelling.
  - Check if the namespace exists: `kubectl get namespace`.
  - Ensure RBAC permissions allow access to the specified namespace.

29. Error: "service account cannot act as a user..."

- Description: Occurs when a service account attempts to perform actions restricted to regular users.
- Troubleshooting:
  - Check RBAC policies to ensure service accounts have appropriate permissions.
  - Verify if impersonation is required for the service account.
  - Review Kubernetes RBAC documentation for best practices.

30. Error: "pod has unbound immediate PersistentVolumeClaims"

- Description: Indicates that a pod's PersistentVolumeClaims (PVCs) are not bound to any PersistentVolumes (PVs).
- Troubleshooting:
  - Check if the appropriate storage classes and PVs are available.
  - Verify PVC definitions and storage class requirements.
  - Troubleshoot PV provisioning issues if necessary.

31. Error: "pod has unbound PersistentVolumeClaims"

- Description: Similar to the previous error, but indicates PVCs that are not bound to PVs in a pod.
- Troubleshooting:
  - Follow the same troubleshooting steps as for "pod has unbound immediate PersistentVolumeClaims".

32. Error: "No nodes are available that match all of the predicates..."

- Description: Occurs when no nodes satisfy the scheduling constraints specified in the pod definition.
- Troubleshooting:
    - Check node labels and selectors in the pod definition.
    - Verify node resources and taints.
    - Ensure nodes are not cordoned or unschedulable.

33. Error: "the server was unable to return a response in the time..."

- Description: Indicates a timeout waiting for a response from the Kubernetes API server.
- Troubleshooting:
    - Check network connectivity to the API server.
    - Verify the API server's health and performance.
    - Increase timeout values if necessary.

34. Error: "requested access to the resource is denied"

- Description: Indicates that the user or service account does not have permission to access the specified resource.
- Troubleshooting:
    - Check RBAC policies to ensure the user or service account has appropriate permissions.
    - Verify the kubeconfig context and credentials being used.
    - Review cluster-level RBAC policies.

35. Error: "no matches for kind..."

- Description: Occurs when specifying an invalid or unsupported resource kind.
- Troubleshooting:
    - Verify the spelling of the resource kind.
    - Check if the resource kind belongs to the correct API group.
    - Ensure the correct API version is specified.

36. Error: "service is not found"

- Description: Indicates that the specified service does not exist.
- Troubleshooting:
    - Check the spelling of the service name.
    - Verify if the service belongs to the correct namespace.
    - Ensure the service has been created successfully.

37. Error: "invalid resource requests..."

- Description: Occurs when the resource requests specified in the pod definition are

invalid.

- Troubleshooting:
    - Verify resource requests syntax in the pod specification.
    - Check for typos or incorrect units in resource requests.
    - Ensure resource requests do not exceed node capacity.

38. Error: "no resources found in..."

- Description: Indicates that no resources are found matching the specified criteria.
- Troubleshooting:
    - Verify the spelling of the resource name.
    - Check if the resource exists in a different namespace.
    - Ensure the correct kubeconfig context is set.

39. Error: "unable to load config, invalid configuration: no configuration has been provided"

- Description: Occurs when the kubeconfig file is missing or incorrectly specified.
- Troubleshooting:
    - Specify the correct path to the kubeconfig file using the `--kubeconfig` flag.
    - Ensure the kubeconfig file exists and has appropriate permissions.
    - Set the `KUBECONFIG` environment variable to the correct file path.

40. Error: "unknown flag..."

- Description: Indicates that an unknown flag was provided in the command.
- Troubleshooting:
  - Check the command syntax for typos.
  - Verify if the flag is supported by the `kubectl` command.
  - Use `kubectl --help` to list available flags and options.

41. Error: "failed to provision volume with StorageClass..."

- Description: Occurs when a dynamic volume provisioner fails to provision a volume.
- Troubleshooting:
  - Check if the storage class is configured correctly.
  - Verify if the underlying storage provider is functioning properly.
  - Review dynamic provisioning configuration and permissions.

42. Error: "failed to fetch node info..."

- Description: Indicates that `kubectl` is unable to retrieve information about nodes in the cluster.
- Troubleshooting:
  - Check if the API server is reachable from the client machine.
  - Verify network connectivity to the API server.
  - Ensure the correct kubeconfig context is set.

43. Error: "operation cannot be fulfilled on..."

- Description: Occurs when attempting to perform an operation that is not supported by the resource.
- Troubleshooting:
  - Check the Kubernetes API version for compatibility with the operation.
  - Verify if the resource is in a valid state for the operation.
  - Review Kubernetes API documentation for supported operations.

44. Error: "unable to remove finalizers"

- Description: Indicates that finalizers preventing resource deletion cannot be removed.
- Troubleshooting:

    - Check if there are any controllers or processes blocking finalizer removal.
    - Manually remove finalizers from the resource using `kubectl edit`.

45. Error: "unable to connect to server: x509: certificate signed by unknown authority"

- Description: Indicates an issue with the Kubernetes API server's SSL certificate.
- Troubleshooting:
    - Verify the certificate authority (CA) used to sign the API server's certificate.
    - Ensure the client machine trusts the CA by adding it to the trust store.
    - Check if the kubeconfig file is configured with the correct server certificate authority.

46. Error: "operation not supported for token requests"

- Description: Occurs when attempting to perform an unsupported operation with a service account token.
- Troubleshooting:
    - Check RBAC policies to ensure the service account has appropriate permissions.
    - Verify if the operation is supported for service account tokens.
    - Review Kubernetes RBAC documentation for service account permissions.

47. Error: "certificate has expired or is not yet valid"

- Description: Indicates that the SSL certificate used by the Kubernetes API server is expired or not yet valid.
- Troubleshooting:
    - Check the expiration date of the API server's SSL certificate.
    - Renew or replace the SSL certificate if it has expired.
    - Ensure the client machine's system time is synchronized.

48. Error: "invalid character '<' looking for beginning of value"

- Description: Occurs when parsing invalid JSON or YAML in a Kubernetes resource definition.
- Troubleshooting:

  - Check for syntax errors in the resource definition file.
  - Ensure the file is valid JSON or YAML format.
  - Use a YAML linter or validator to identify and fix syntax errors.

49. Error: "operation cannot be fulfilled on persistentvolumeclaims..."

- Description: Occurs when attempting to perform an operation not supported on PersistentVolumeClaims (PVCs).
- Troubleshooting:
  - Verify if the PVC is in a valid state for the operation.
  - Check if there are any controllers or processes blocking the operation.
  - Review Kubernetes API documentation for supported operations on PVCs.

50. Error: "failed to pull image..."

- Description: Indicates a failure to pull a container image from the specified registry.
- Troubleshooting:
  - Check if the image name and tag are correct.
  - Verify network connectivity to the image registry.
  - Ensure proper authentication to the image registry: `docker login <registry>`.

51. Error: "error: the server doesn't have a resource type..."

- Description: Indicates that the requested resource type is not supported by the Kubernetes API server.
- Troubleshooting:
  - Check the API server version for compatibility.
  - Verify the resource type is spelled correctly.
  - Check if the API server supports custom resource definitions (CRDs) if using custom resources.

52. Error: "Failed to attach volume..."

- Description: Occurs when a volume cannot be attached to a pod.
- Troubleshooting:
  - Check if the volume plugin is properly configured on the node.

- Verify the volume name and type in the pod specification.
- Check if the volume is already in use by another pod.

53. Error: "No resources found in default namespace"

- Description: Indicates that no resources are present in the default namespace.
- Troubleshooting:
  - Specify a different namespace using `-n <namespace>`.
  - Check if resources exist in a different namespace.
  - Ensure the correct kubeconfig context is set.

54. Error: "Back-off restarting failed container"

- Description: Indicates that a container within a pod is repeatedly failing to start.
- Troubleshooting:
  - Check container logs for the specific error: `kubectl logs <pod_name> -c <container_name>`.
  - Verify resource constraints and requests are appropriate.
  - Investigate application code for issues.

55. Error: "Volume is already exclusively attached to one node and can't be attached to another"

- Description: Occurs when attempting to attach a volume that is already attached to another node.
- Troubleshooting:
  - Ensure the volume is detached from the previous node.
  - Check for orphaned resources in the cluster.
  - Verify the volume attachment status using `kubectl describe volumeattachment <volume_attachment_name>`.

56. Error: "Service Unavailable"

- Description: Indicates that the Kubernetes API server is temporarily unavailable.
- Troubleshooting:
  - Check API server logs for errors.
  - Verify network connectivity to the API server.
  - Monitor system resources on the Kubernetes control plane nodes.

57. Error: "Invalid value: must be no more than"

- Description: Indicates that a resource's value exceeds a specified limit.
- Troubleshooting:
  - Check resource quotas and limits in the namespace.
  - Review resource requests and limits in the

pod specification.

- Increase resource quotas if necessary.

58. Error: "Unable to mount volumes for pod"

- Description: Occurs when Kubernetes cannot mount volumes specified in the pod definition.
- Troubleshooting:
  - Verify volume specifications in the pod definition.
  - Check if the volume plugin is installed and configured on the node.
  - Ensure the volume exists and is accessible.

59. Error: "Unknown field 'replicas' in..."

- Description: Indicates an incorrect field in the resource definition, often caused by a version mismatch.
- Troubleshooting:
  - Verify the Kubernetes API version being used.
  - Check the resource definition against the correct API version.
  - Update the resource definition if necessary.

60. Error: "cannot create serviceaccount..."

- Description: Occurs when attempting to create a service account without sufficient permissions.
- Troubleshooting:
  - Ensure the user has RBAC permissions to create service accounts.
  - Check if there are any admission controllers blocking service account creation.
  - Verify the namespace for any namespace-scoped restrictions.

61. Error: "failed to find a usable init container"

- Description: Indicates that an init container specified in the pod definition cannot be scheduled.
- Troubleshooting:
  - Check init container readiness and liveness probes.
  - Verify the image name and tag for the init container.
  - Ensure the init container's resource requests and limits are appropriate.

62. Error: "the server could not find the requested resource"

- Description: Indicates that the requested resource does not exist.
- Troubleshooting:
  - Check the spelling of the resource name.
  - Verify if the resource belongs to the correct API group.
  - Ensure the correct namespace is specified.

63. Error: "invalid configuration: unable to load..."

- Description: Occurs when the kubeconfig file is incorrectly configured.
- Troubleshooting:
  - Verify the kubeconfig file path and permissions.
  - Check for syntax errors or missing configurations in the kubeconfig file.
  - Use the `KUBECONFIG` environment variable to specify the kubeconfig file.

64. Error: "failed to start container..."

- Description: Occurs when a container fails to start within a pod.
- Troubleshooting:
  - Check container logs for errors: `kubectl logs <pod_name> -c <container_name>`.
  - Verify resource requests and limits for the container.
  - Investigate application code for issues.

65. Error: "unable to recognize..."

- Description: Indicates that the Kubernetes API server does not recognize the specified resource.
- Troubleshooting:
  - Check the API server logs for errors.

  - Verify the API server version for compatibility with the resource.
  - Ensure the correct API group is specified.

66. Error: "invalid namespace"

- Description: Occurs when specifying an invalid namespace.
- Troubleshooting:
  - Verify the namespace spelling.
  - Check if the namespace exists: `kubectl get namespace`.
  - Ensure RBAC permissions allow access to the specified namespace.

67. Error: "service account cannot act as a user..."

- Description: Occurs when a service account attempts to perform actions restricted to regular users.
- Troubleshooting:
  - Check RBAC policies to ensure service accounts have appropriate permissions.
  - Verify if impersonation is required for the service account.
  - Review Kubernetes RBAC documentation for best practices.

68. Error: "pod has unbound immediate PersistentVolumeClaims"

- Description: Indicates that a pod's PersistentVolumeClaims (PVCs) are not bound to any PersistentVolumes (PVs).
- Troubleshooting:
  - Check if the appropriate storage classes and PVs are available.
  - Verify PVC definitions and storage class requirements.
  - Troubleshoot PV provisioning issues if necessary.

69. Error: "pod has unbound PersistentVolumeClaims"

- Description: Similar to the previous error, but indicates PVCs that are not bound to PVs in a pod.
- Troubleshooting:
  - Follow the same troubleshooting steps as for "pod has unbound immediate PersistentVolumeClaims".

70. Error: "No nodes are available that match all of the predicates..."

- Description: Occurs when no nodes satisfy the scheduling constraints specified in the pod definition.
- Troubleshooting:
    - Check node labels and selectors in the pod definition.
    - Verify node resources and taints.
    - Ensure nodes are not cordoned or unschedulable.

71. Error: "the server was unable to return a response in the time..."

- Description: Indicates a timeout waiting for a response from the Kubernetes API server.
- Troubleshooting:
    - Check network connectivity to the API server.
    - Verify the API server's health and performance.
    - Increase timeout values if necessary.

72. Error: "requested access to the resource is denied"

- Description: Indicates that the user or service account does not have permission to access the specified resource.
- Troubleshooting:
    - Check RBAC policies to ensure the user or service account has appropriate permissions.
    - Verify the kubeconfig context and credentials being used.
    - Review cluster-level RBAC policies.

73. Error: "no matches for kind..."

- Description: Occurs when specifying an invalid or unsupported resource kind.
- Troubleshooting:
    - Verify the spelling of the resource kind.
    - Check if the resource kind belongs to the correct API group.
    - Ensure the correct API version is specified.

74. Error: "service is not found"

- Description: Indicates that the specified service does not exist.
- Troubleshooting:
  - Check the spelling of the service name.
  - Verify if the service belongs to the correct namespace.
  - Ensure the service has been created successfully.

75. Error: "invalid resource requests..."

- Description: Occurs when the resource requests specified in the pod definition are invalid.
- Troubleshooting:
  - Verify resource requests syntax in the pod specification.
  - Check for typos or incorrect units in resource requests.
  - Ensure resource requests do not exceed node capacity.

76. Error: "no resources found in..."

- Description: Indicates that no resources are found matching the specified criteria.
- Troubleshooting:
  - Verify the spelling of the resource name.
  - Check if the resource exists in a different namespace.
  - Ensure the correct kubeconfig context is set.

77. Error: "unable to load config, invalid configuration: no configuration has been provided"

- Description: Occurs when the kubeconfig file is missing or incorrectly specified.
- Troubleshooting:
  - Specify the correct path to the kubeconfig file using the `--kubeconfig` flag.
  - Ensure the kubeconfig file exists and has appropriate permissions.
  - Set the `KUBECONFIG` environment variable to the correct file path.

78. Error: "unknown flag..."

- Description: Indicates that an unknown flag was provided in the command.
- Troubleshooting:
  - Check the command syntax for typos.
  - Verify if the flag is supported by the `kubectl` command.
  - Use `kubectl --help` to list available flags and options.

79. Error: "failed to provision volume with StorageClass..."

- Description: Occurs when a dynamic volume provisioner fails to provision a volume.
- Troubleshooting:
  - Check if the storage class is configured correctly.
  - Verify if the underlying storage provider is functioning properly.
  - Review dynamic provisioning configuration and permissions.

80. Error: "failed to fetch node info..."

- Description: Indicates that `kubectl` is unable to retrieve information about nodes in the cluster.
- Troubleshooting:
  - Check if the API server is reachable from the client machine.
  - Verify network connectivity to the API server.
  - Ensure the correct kubeconfig context is set.

81. Error: "operation cannot be fulfilled on..."

- Description: Occurs when attempting to perform an operation that is not supported by the resource.
- Troubleshooting:
  - Check the Kubernetes API version for compatibility with the operation.
  - Verify if the resource is in a valid state for the operation.
  - Review Kubernetes API documentation for supported operations.

82. Error: "unable to remove finalizers"

- Description: Indicates that finalizers preventing resource deletion cannot be removed.

- Troubleshooting:
  - Check if there are any controllers or processes blocking finalizer removal.
  - Manually remove finalizers from the resource using `kubectl edit`.

83. Error: "unable to connect to server: x509: certificate signed by unknown authority"

- Description: Indicates an issue with the Kubernetes API server's SSL certificate.
- Troubleshooting:
  - Verify the certificate authority (CA) used to sign the API server's certificate.
  - Ensure the client machine trusts the CA by adding it to the trust store.
  - Check if the kubeconfig file is configured with the correct server certificate authority.

84. Error: "operation not supported for token requests"

- Description: Occurs when attempting to perform an unsupported operation with a service account token.
- Troubleshooting:
  - Check RBAC policies to ensure the service account has appropriate permissions.
  - Verify if the operation is supported for service account tokens.
  - Review Kubernetes RBAC documentation for service account permissions.

85. Error: "certificate has expired or is not yet valid"

- Description: Indicates that the SSL certificate used by the Kubernetes API server is expired or not yet valid.
- Troubleshooting:
  - Check the expiration date of the API server's SSL certificate.
  - Renew or replace the SSL certificate if it has expired.
  - Ensure the client machine's system time is synchronized.

86. Error: "invalid character '<' looking for beginning of value"

- **Description**: Occurs when parsing invalid JSON or YAML in a Kubernetes resource definition.
- **Troubleshooting**:
  - Check for syntax errors in the resource definition file.
  - Ensure the file is valid JSON or YAML format.
  - Use a YAML linter or validator to identify and fix syntax errors.

87. Error: "operation cannot be fulfilled on..."

- **Description**: Occurs when attempting to perform an operation that is not supported by the resource.
- **Troubleshooting**:
  - Check the Kubernetes API version for compatibility with the operation.
  - Verify if the resource is in a valid state for the operation.
  - Review Kubernetes API documentation for supported operations.

88. Error: "unable to remove finalizers"

- **Description**: Indicates that finalizers preventing resource deletion cannot be removed.
- **Troubleshooting**:
  - Check if there are any controllers or processes blocking finalizer removal.
  - Manually remove finalizers from the resource using `kubectl edit`.

89. Error: "unable to connect to server: x509: certificate signed by unknown authority"

- **Description**: Indicates an issue with the Kubernetes API server's SSL certificate.
- **Troubleshooting**:
  - Verify the certificate authority (CA) used to sign the API server's certificate.
  - Ensure the client machine trusts the CA by adding it to the trust store.
  - Check if the kubeconfig file is configured with the correct server certificate authority.

90. Error: "operation not supported for token requests"

- **Description**: Occurs when attempting to perform an unsupported operation with a service account token.
- **Troubleshooting**:
  - Check RBAC policies to ensure the service account has appropriate permissions.
  - Verify if the operation is supported for service account tokens.
  - Review Kubernetes RBAC documentation for service account permissions.

91. Error: "certificate has expired or is not yet valid"

- **Description**: Indicates that the SSL certificate used by the Kubernetes API server is expired or not yet valid.
- **Troubleshooting**:
  - Check the expiration date of the API server's SSL certificate.
  - Renew or replace the SSL certificate if it has expired.
  - Ensure the client machine's system time is synchronized.

92. Error: "error: the server doesn't have a resource type..."

- **Description**: Indicates that the requested resource type is not supported by the Kubernetes API server.
- **Troubleshooting**:
  - Check the API server version for compatibility.
  - Verify the resource type is spelled correctly.
  - Check if the API server supports custom resource definitions (CRDs) if using custom resources.

93. Error: "Failed to attach volume..."

- **Description**: Occurs when a volume cannot be attached to a pod.
- **Troubleshooting**:
  - Check if the volume plugin is properly configured on the node.
  - Verify the volume name and type in the pod specification.
  - Check if the volume is already in use by another pod.

94. Error: "No resources found in default namespace"

- **Description**: Indicates that no resources are present in the default namespace.
- **Troubleshooting**:
  - Specify a different namespace using `-n <namespace>`.
  - Check if resources exist in a different namespace.
  - Ensure the correct kubeconfig context is set.

95. Error: "Back-off restarting failed container"

- **Description**: Indicates that a container within a pod is repeatedly failing to start.
- **Troubleshooting**:
  - Check container logs for the specific error: `kubectl logs <pod_name> -c <container_name>`.
  - Verify resource constraints and requests are appropriate.
  - Investigate application code for issues.

96. Error: "Volume is already exclusively attached to one node and can't be attached to another"

- **Description**: Occurs when attempting to attach a volume that is already attached to another node.
- **Troubleshooting**:
  - Ensure the volume is detached from the previous node.
  - Check for orphaned resources in the cluster.
  - Verify the volume attachment status using `kubectl describe volumeattachment <volume_attachment_name>`.

97. Error: "Service Unavailable"

- **Description**: Indicates that the Kubernetes API server is temporarily unavailable.
- **Troubleshooting**:
  - Check API server logs for errors.
  - Verify network connectivity to the API server.
  - Monitor system resources on the Kubernetes control plane nodes.

98. Error: "Invalid value: must be no more than"

- **Description**: Indicates that a resource's value exceeds a specified limit.
- **Troubleshooting**:
  - Check resource quotas and limits in the namespace.
  - Review resource requests and limits in the pod specification.
  - Increase resource quotas if necessary.

99. Error: "Unable to mount volumes for pod"

- **Description**: Occurs when Kubernetes cannot mount volumes specified in the pod definition.
- **Troubleshooting**:
  - Verify volume specifications in the pod definition.
  - Check if the volume plugin is installed and configured on the node.
  - Ensure the volume exists and is accessible.

100. Error: "Unknown field 'replicas' in..."

- **Description**: Indicates an incorrect field in the resource definition, often caused by a version mismatch.
- **Troubleshooting**:
  - Verify the Kubernetes API version being used.
  - Check the resource definition against the correct API version.
  - Update the resource definition if necessary.