Epic Games - Cloud Developer Platform

Debugging Your Helm Chart

Downloaded from Epic Games Confluence
Date: 2025-07-12 04:09:20
Original URL: https://confluence-epicgames.atlassian.net/wiki/spaces/CDE/pages/81068379

Document Level Classification

300

Introduction

When you deploy your Helm charts to substrate, it is possible you will run into issues with deployment due to a number of factors. For example, is the syntax correct, is the configuration correct, is there an issue at the Kubernetes cluster level? Any one of these issues could result in a deployment failure. However, helm offers linting tools as well as other commands that will aid in debugging and/or troubleshooting your deployment. This document serves as a reference for debugging and troubleshooting your Helm chart deployments.

Linting

If you want to examine your chart for possible issues, you can use the command `helm lint` . This command takes a path to a chart and runs a series of tests to verify the chart is well-formed. If the linter encounters things that will cause the chart to fail installation, it will emit **[ERROR]** messages. If it encounters issues that break with convention or recommendation, it will emit **[WARNING]** messages. Additionally you can pass flags to the command. For example if you need help you can pass the -h flag. For a complete list of flags, reference the helm lint command documentation.

```
# Replace PATH with the path to your chart
helm lint PATH [flags]

# Example
helm lint .\deploy\chart\
==> Linting .\deploy\chart\
[INFO] Chart.yaml: icon is recommended
[WARNING] templates/: directory not found

1 chart(s) linted, 0 chart(s) failed
```

Testing Templates Locally

If you want to test your templates by rendering them locally the `helm template` command can be used along with the `--debug` flag. You will also wat to pass in your values file using the `-f` followed by the path and name of the values file. For example you may be debugging the dev environment values, in which case you would pass the `values.dev.yaml` file.

```
# Replace PATH with the path to your chart
helm template PATH -f values.dev.yaml --debug
```

Configuration Checks

If you want the server to render your templates then return the resulting manifest file, you can use the `helm install` command with the `--dry-run` and `--debug` flags.

```
# Replace PATH with the path to your chart
helm install PATH --dry-run --debug
```

Troubleshooting Errors

There are many great resources already available on the internet for the most common errors when deploying, debugging, and troubleshooting your Helm charts and supporting Kubernetes manifests. Below is a list of these resources.

- Helm Troubleshooting - Troubleshooting documentation from official Helm Docs.
- Helm Template Debugging - Helm template debugging documentation from official Helm Docs.
- Common Errors Found in Kubernetes Manifests - Documentation of common errors and solutions when working with Kubernetes manifest files.
- Airflow Helm chart: FAQs – Errors & Troubleshooting Note: While this links to a specific Helm chart, the errors are commonly encountered with other helm charts. You would just need to be aware that you would not be using airflow and instead referencing the name of the Helm chart you are attempting to install.

Page Information:
Page ID: 81068379
Space: Cloud Developer Platform
Downloaded: 2025-07-12 04:09:20