Converting a Tag Role from OldProd to a Helm Chart for Substrate

Document Level Classification

300

Introduction

When migrating from OldProd to Substrate, the mechanism for deploying your services will be different. Instead of using Tag Roles, you use Helm to deploy a chart. A Helm chart is a package that contains all the necessary resources to deploy an application to a Kubernetes cluster. This includes YAML configuration files for deployments, services, secrets, and config maps that define the desired state of your application.

Converting a Tag Role to a Helm Chart Manually

When converting from a Tag Role to a Helm Chart, it is strongly recommended that epic-app be utilized as a dependency Helm chart. Below you will find examples of Tag Role sections that will be carried over to a values file for the Helm chart.

- For a more in depth walkthrough of creating your first helm chart, reference Deploying your first application to Substrate and go to the section titled Create a Helm Chart.
- For a more in depth explanation of epic app and what it offers, reference the Using epic-app with Substrate documentation.

Placeholders

Placeholder	Description	Example
`<NAME-OF-SERVICE>`	Name of your service	datastorage-service
`<SERVICE-DESCRIPTION>`	Description of your service	Datastorage Service
`<MAINTAINER-NAME>`	Name of the service maintainer	john.doe
`<MAINTAINER-EMAIL>`	Email address of the service maintainer	team-name@epicgames.com
`<UNIQUE-ALPHANUMERIC-IDENTIFIER>`	Unique alphanumeric identifier of the container image tag	1.0.0-111A

Service Details

The `Chart.yaml` file is where you will set up the details of the service, including the name of the `service` , `description` , `maintainers` , `type` , `version` , `appVersion` , and dependencies. We will be using epic-app as a dependency here.

The item that will carry over from your Tag Role will be the `servicename`. This will be placed in the `name:` field of the Chart.yaml file.

Chart.yaml

```
apiVersion: v2
name: <NAME-OF-SERVICE>
description: <SERVICE-DESCRIPTION>
maintainers:
  - name: <MAINTAINER-NAME>
  - email: <MAINTAINER-EMAIL>
# A chart can be either an 'application' or a 'library' chart.
#
# Application charts are a collection of templates that can be packaged
# to be deployed.
#
# Library charts provide useful utilities or functions for the chart de
# a dependency of application charts to inject those utilities and func
# pipeline. Library charts do not define any templates and therefore ca
type: application
# This is the chart version. This version number should be incremented
# to the chart and its templates, including the app version.
# Versions are expected to follow Semantic Versioning (https://semver.o
version: 1.0.0
# This is the version number of the application being deployed. This ve
# incremented each time you make changes to the application. Versions a
# follow Semantic Versioning. They should reflect the version the appli
appVersion: 1.0.0
```

```
  - name: epic-app
    repository: https://artifacts.ol.epicgames.net/artifactory/substr-h
    # epic-app versions and associated release notes is available here:
    # https://github.ol.epicgames.net/charts/epic-app/releases
    version: ">= 2.4.6, < 2.6.0"
```

Environment Variables

Tag Roles will have an `environment` section that references all environment variables passed to a container.

Tag Role Environment Variables

```
pod:
        containers:
                name-of-service: # This section will be the same name a
                        image_name:  # This will be the path to the doc
                        image_tag:   # This will be the image tag
                        command:
                                start
                        hot_config_enabled: false
                        environment:
                                EPIC_ENV: "{{env}}" # This is one examp
```

The environment variables will be carried over into the `environment` subsection of the `containers` under `epic-app`.

values.yaml

```
epic-app:
  #
  # Reference:
  # * https://github.ol.epicgames.net/charts/epic-app/blob/main/values.
  #
```

```
    containers: # This is a map of containers that are in the Pod

    # Application container
    example-container:
      image:
        name: "artifacts.ol.epicgames.net/<DOCKER_REPO>/example-contain
        tag: "1.0.0-<UNIQUE-ALPHANUMERIC-IDENTIFIER>"
      ports:
        - 80
      environment:
        EPIC_ENV: gamedev
```

Secrets

In a Tag Role, secrets can be identified in the `environment` section as well. However, the values would be looked up in OldProd vault and can be identified by looking for values that contain a vault lookup.

```
pod:
        containers:
                name-of-service: # This section will be the same name a
                        image_name:  # This will be the path to the doc
                        image_tag:   # This will be the image tag
                        command:
                                start
                        hot_config_enabled: false
                        environment:
                                EPIC_ENV: "{{env}}" # This is one examp
                                PWD: "{{lookup('vault', 'secret/data/ar
                                # Above is an example of a secret in a
```

This secrets would then be carried over to the `externalSecrets` section under `epic-app`. The example below shows the use of External Secrets Operator (ESO) to inject secrets from Substrate Vault. The secret PWD will carry over from the Tag Role as pwd in the values.yaml file.

Starting from August 2024, it is mandatory that all new Kubernetes clusters use External Secrets Operator (ESO). We strongly recommend using ESO, but the Vault Injector sidecar will still remain available with limited support, bug fixes will be provided if needed to maintain core functionality.

Note: Before the secret can be access, it would first have to be created in Substrate Vault as the Tag Role uses OldProd Vault. For information on utilizing secrets in Substrate reference Using External Secrets Operator (ESO) in epic-app to inject secrets.

```
epic-app:
  #
  # Reference:
  # * https://github.ol.epicgames.net/charts/epic-app/blob/main/values.
  #
  containers: # This is a map of containers that are in the Pod
    # Application container
    example-container:
      image:
        name: "artifacts.ol.epicgames.net/<DOCKER_REPO>/example-contain
        tag: "1.0.0-<UNIQUE-ALPHANUMERIC-IDENTIFIER>"
      ports:
        - 80
      environment:
        EPIC_ENV: gamedev
      externalSecrets:
        DATABASE_PASSWORD:
          key: <cluster-name>/<namespace-name>/<secret>
```

```
        property: dbpassword

# Enable externalSecrets
externalSecrets:
  enabled: true
  refreshInterval: 12h
  deletionPolicy: Retain
```
