Scopearctl User Guide: Simplifying ScopeAR Configuration Management

Introduction

- *Purpose of the Document*
- *Overview of Scopearctl*

Scopearctl at a Glance

- *Key Functionality*
- *Benefits*

Installation Guide

- *Prerequisites*
- *Installation via Script*
  - *Run the Script*
  - *Detailed Script Steps*
  - *Example Install Scenarios*

Using Scopearctl

- *Configuration Files*
- *Deployment Process*
  - *Running k8s apply command*
  - *Detailed Script Steps*
  - *Example Usage Scenarios*
- *Handling Multiple Environments*

Advanced Features and Customizations

- *Custom Scripts and Extensions*
- *Security and Compliance*

Troubleshooting and Support

- *Common Issues and Resolutions*
- *Contact*

Introduction

Purpose of the Document

This document provides a comprehensive guide on how to install, use, and understand the purpose and benefits of "scopearctl," a tool designed to streamline configuration management across Kubernetes assets in Scopear environments.

Overview of Scopearctl

Scopearctl is a command-line tool that simplifies the management of configurations across multiple deployments within Scopear Kubernetes (K8s) environments.

Scopearctl at a Glance

Key Functionality

- **Single Configuration File Management:** Centralizes the configuration process into a single file, replacing the need for multiple, disparate configuration files.
- **Programmatic Injection:** Automates the process of injecting configurations into various deployment settings.
- **Logical Manifest Apply:** Deploys the manifest to a K8s cluster in a logical order for maximum dependency success.

Benefits

- **Simplicity and Efficiency:** Reduces complexity and enhances efficiency in managing configurations.
- **Consistency:** Ensures uniformity across different deployments and configurations.
- **Error Reduction:** Minimizes the potential for errors that can arise from manual configuration management.

Installation Guide

Prerequisites

- Operating System: Linux or macOS
- Required Tools: `curl`, `shasum`, `tar`
- Sufficient permissions to install software (may require `sudo` access)

Installation via Script

1. *Run the Script*

The script accepts several optional parameters to customize the installation process:

- `--os=<os>`: Specify the operating system (default: detected automatically).
- `--version=<version>`: Specify the version to install (default: latest).
- `--channel=<channel>`: Specify the release channel (default: stable).
- `--docsdir=<docsdir>`: Specify the directory for documentation (default: current directory).
- `--tmpdir=<tmpdir>`: Specify the temporary directory for downloads (default: `/tmp/scopearctl`).
- `--installdir=<installdir>`: Specify the installation directory (default: `/usr/local/bin`).
- `--nodocs`: Skip downloading and setting up documentation.
- `--nodownload`: Skip downloading the binary and only perform installation steps.

Example Command:

```
    curl -sSL https://release.scopear.com/scopearctl/install | sh -s --
--version=1.0.0 --installdir=/opt/scopearctl
```

1. *Detailed Script Steps*
    - **Validation:** Checks if the operating system is either Linux or macOS. If not, it exits with an error.
    - **Download:** Constructs the download URL based on the specified version and channel, then downloads the archive to the temporary directory.
    - **Checksum Verification:** If checksum verification is enabled, the script downloads the checksum file and verifies the integrity of the downloaded archive.
    - **Extraction:** The downloaded archive is extracted to the temporary directory.
    - **Documentation:** If documentation setup is enabled, the script downloads the documentation files to the specified directory.
    - **Installation:** The binary is moved to the specified installation directory, and execution permissions are set.
    - **Cleanup:** Temporary files and directories are removed.

Example Install Scenarios

- **Install the Latest Version:**

```
  curl -sSL https://release.scopear.com/scopearctl/install | sh
```

- **Install a Specific Version:**

```
   curl -sSL https://release.scopear.com/scopearctl/install | sh -s --
--version=3.37.2
```

- **Install Without Documentation:**

```
   curl -sSL https://release.scopear.com/scopearctl/install | sh -s --
```

```
--nodocs
```

- **Install to Custom Directories:**

```
  curl -sSL https://release.scopear.com/scopearctl/install | sh -s -- \
  --installdir=/opt/scopearctl \
  --docsdir=/opt/scopearctl/docs \
  --tmpdir=/opt/scopearctl/tmp
```

Using Scopearctl

Configuration Files

- **Structure and Format:** The unified configuration file enables the end user to set unique configurations to almost all available settings ScopeAR has. Using higher-order stanzas, configs are nested under their respective application.
- **Best Practices:** It is recommended to have a separate file per environment and store those files in a version control system such as Git.

Deployment Process

1. **Set Kubeconfig Context:** Ensure that your kubeconfig context is set to the proper cluster.
2. *Running k8s apply command*

The script accepts several optional parameters to customize the installation process:

```
  scopearctl k8s apply -h

     â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ     â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ     â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ
â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ        â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ        â–ˆâ–ˆâ–ˆâ–ˆâ–ˆ
â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ
  â–ˆâ–ˆ       â–ˆâ–ˆ        â–ˆâ–ˆ      â–ˆâ–ˆ â–ˆâ–ˆ     â–ˆâ–ˆ â–ˆâ–ˆ
â–ˆâ–ˆ   â–ˆâ–ˆ â–ˆâ–ˆ   â–ˆâ–ˆ
  â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ â–ˆâ–ˆ          â–ˆâ–ˆ      â–ˆâ–ˆ â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ
â–ˆâ–ˆâ–ˆâ–ˆâ–ˆ   â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ
      â–ˆâ–ˆ â–ˆâ–ˆ       â–ˆâ–ˆ     â–ˆâ–ˆ â–ˆâ–ˆ       â–ˆâ–ˆ        â–ˆâ–ˆ
â–ˆâ–ˆ â–ˆâ–ˆ   â–ˆâ–ˆ
  â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ  â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ  â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ  â–ˆâ–ˆ
â–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆâ–ˆ â–ˆâ–ˆ   â–ˆâ–ˆ â–ˆâ–ˆ   â–ˆâ–ˆ

                              â”Œâ”€               Build               info
â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â
”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”€â”•
  |                                                    |
   | â”Œâ”€ Version â”€â”• â”Œâ”€ Plugin Revision â”€â”• â”Œâ”€ Channel
â”€â”• |
  | | 3.37.2    | | da6326c           | | stable    | |
```

```
                        |            â""â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"~
â""â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"~
â""â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"~ |
  |                                                  |

â""â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â
"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"€â"
€â"€â"~

   Generates & deploys ScopeAR components manifests. Configured by your
environment manifest.

  Usage:
    scopearctl k8s apply [flags]

  Examples:
  scopearctl k8s apply -f manifest.yaml

  Flags:
        --dry               Generate manifests only.
        --exclude strings   List of components to exclude from apply
    -h, --help              help for apply
    -f, --manifest string   ScopeAR CR file.
        --only strings      List of components to apply; e.g. --only scopear
    -o, --out string         Write the manifests to the given path (default
"__gen__")
        --skip-delays       Disabled manifest apply delays

  Global Flags:
        --debug                 Set log level to debug
              --kubeconfig  string          Path  to  kubeconfig  (default
"/Users/sgraham/.kube/config")
    -v, --verbose count           Increase logging verbosity (use multiple
times for more output)
```

Example Command:

```
   scopearctl k8s apply -f /path/to/config.yaml
```

1. *Detailed Script Steps*
   - **Defaulting:** The tool sets default values for any configuration item that is left unset in your environment manifest.
   - **Validation:** The tool checks if the values set in your ScopeAR envirnoment manifest. If error exist, it will return them all and exit the process.
   - **YAML Manifest Generation:** Scopearctl programmatically injects the single configuration file into multiple K8s assets and outputs them to a local directory, default `./__gen__`.
   - **Apply Generated Manifests to Cluster:** If the `--dry` flag is not set and the user has the appropriate cluster context authorization, the tool will apply the generated manifest similar to the way `kubectl apply -f` would.

Example Usage Scenarios

- **Apply the configs set in your ScopeAR manifest:**

```
scopearctl k8s apply \
  -f /path/to/scopear-manifest.yaml
```

- **Only generate yaml manifests:**

```
scopearctl k8s apply \
  -f /path/to/scopear-manifest.yaml \
  --dry
```

- **Set a custom kubeconfig file:**

```
scopearctl k8s apply \
  -f /path/to/scopear-manifest.yaml \
  --kubeconfig /path/to/custom-config
```

Handling Multiple Environments

- **Environment Specific Configurations:** It is best practice to create a configuration file per environment and use a logical file naming scheme to easily differentiate between configs.
- **Version Control Integration:** Covers integration with version control systems for tracking changes.

Advanced Features and Customizations

Custom Scripts and Extensions

- **Integration with Other Tools and Scripts:** Scopearctl can be easily installed into any CI/CD environment either in custom scripting or by

running raw Linux commands to download and install. These installations can be controlled by downloading specific versions and installing into permissioned paths. You can read about the installation above.

Security and Compliance

- **Compliance Checks:** Scopearctl is scanned continuously for vulnerable dependencies and patched regularly.
- **Security Best Practices:** Security vulnerabilities rated "High" or above are patched before releasing any new versions, ensuring you only get the safest builds possible.

Troubleshooting and Support

Common Issues and Resolutions

- **Installation:** - Ensure you have `curl`, `shasum`, and `tar` installed. Verify you have sufficient permissions to install to the specified directory (use `sudo` if necessary). Check for network issues if the download fails.
- **Misconfigurations:** These can be common. Simply edit the config file and re-apply the changes. Kubernetes will handle most of the changes by default; however, if it's just a `ConfigMap` change, the pod that uses the config might need to be restarted in the environment.
- **Apply issues:** Ensure you have the correct expected version of `scopearctl` installed, you can find it by running `scopearctl help`. Verify you have sufficient permissions to apply manifests to the K8s cluster in current context. Check for network issues if the connection fails.

Contact

For any issues or further assistance, please contact the `scopearctl` support team at `devops@scopear.com`.