

Software Requirement Specification for ksctl

By Kubesimplify

Last Modified: 28 Mar 2023



<https://github.com/kubesimplify/kubesimpctl>

Release Cycle: 2 months

Maintainers

- Dipankar Das (dipankardas0115@gmail.com)
- Avinesh Tripathi (avineshtripathi1@gmail.com)

Reviewers

- Saiyam Pathak

Contents

Contents	2
Changelog	4
Purpose	5
Abstract	5
Project Scope	5
Symbols	5
Functional Requirements	6
1. Local Kubernetes Cluster	6
1.1. Local Kubernetes cluster (Docker containers)	6
1.1.1. Create Cluster	6
1.1.2. Delete Cluster	6
1.1.3. View Clusters	7
1.1.4. Context Switch	7
2. Cloud Provider Cluster	7
2.1. Amazon EKS	7
2.1.1. Create Cluster	7
2.1.2. Delete Cluster	7
2.1.3. View Clusters/Profiles	7
2.1.4. Context Switch	7
2.1.5. Context Switch	7
2.2. Azure AKS	8
2.2.1. Create Cluster	8
2.2.2. Delete Cluster	8
2.2.3. Init Auth	8
2.2.3. View Clusters/Profiles	9
2.2.4. Context Switch	9
2.2.5. Context Switch	9
2.3. Civo K3s	9
2.3.1. Create Cluster	9
2.3.2. Delete Cluster	10
2.3.3. Init Auth	10
2.3.4. View Clusters/Profiles	10
2.3.5. Context Switch	10
Non-Functional Requirements	11
1. Constraints	11
2. Safety	11
3. Security	11
Related links	12
23 Nov 2022 ksctl meeting	12

Changelog

Version	Date	Remark	Status
0.1.0	14-11-22	Can create Local and managed Civo k8s cluster	
0.1.1	17-2-23	Can create ha civo clusters	
0.2.1	7-3-23	Able to create managed and ha clusters on azure	
1.0.1-rc1	19-2-23	Better Logging and more stable	
1.0.1-rc2			

Project Backlog	Current Backlog	Completed
https://github.com/kubesimplify/ksctl/issues/17	Check the issues	Local Cluster Managed CIVO HA CIVO based on K3s Managed Azure HA Azure based on K3s

Purpose

Kubernetes Distribution that can create clusters and HA clusters in local as well as on cloud platform

Abstract

Using Different cloud providers APIs(AWS, GCP, Azure, etc) to create and manage K8s clusters in a single CLI call!

For the Local cluster, it uses API calls to Docker for creating both HA or multi-node clusters

Project Scope

Users who use Kubernetes and want a simpler CLI tool that can handle multiple clusters through a single CLI interface

will simplify the cluster creation and deletion rather than using all sorts of tools to create and delete clusters on cloud providers.

Symbols

-  -> UnPlanned
-  -> Started (Planned)
-  -> EarlyStage(Latest, API)
-  -> Development (CLI, Implementation)
-  -> Testing
-  -> Added

Functional Requirements

1. Local Kubernetes Cluster

1.1. Local Kubernetes cluster (Docker containers)

Using Containers to create nodes and connect them together

1.1.1. Create Cluster

STATUS

1.1.1.1. Create Profile

Description: Create Kubernetes Cluster with the given name, name is unique among different clusters and used for each cluster for its naming

Input: user provides cluster name with valid parameters Ex. no. of nodes

Output: Created the profile with the given name

1.1.1.2. Create Nodes

Description: Creating node for the cluster as per the requests

Output: Status of node creation is displayed

1.1.1.3. Configure Control Nodes & Worker Nodes

Description: Configure the nodes install the required components

System: Scripts to configure the nodes and connect to the control node

Output: Status of Kubernetes on all the nodes

1.1.1.4. Get Kubeconfig file

Description: Copy the Kubeconfig file from control plane to the host computer

Output: Saved the file to some default location

1.1.2. Delete Cluster

STATUS

1.1.2.1. Delete Profile

Description: User gives the profile to delete

1.1.2.2. Delete Nodes

Description: Cleanup the nodes/containers associated with the profile

1.1.2.3. Delete Context

Description: Remove the profile from the list

Output: Deleted the Profile/Cluster successfully

1.1.3. View Clusters

STATUS

1.1.3.1. Get the clusters

Description: View all the cluster with are available to be used

Output: List of all the profiles

1.1.4. Context Switch

STATUS

1.1.4.1. Switch Current Profile

Description: user can switch from the current profile to another one. For example from (default) to prod-cluster

Output: Current context switched

2. Cloud Provider Cluster

2.1. Amazon EKS

2.1.1. Create Cluster

STATUS

2.1.2. Delete Cluster

STATUS

2.1.3. View Clusters/Profiles

STATUS

2.1.4. Context Switch

STATUS

2.1.5. Context Switch

STATUS

2.2. Azure AKS

2.2.1. Create Cluster

STATUS

2.2.1.1. Get required and additional info

Description: collect the name, region, and more

Input: Enter cluster name and region

2.2.1.2. Create config and create cluster

Description: initialize the config for the cluster to be made and create a cluster by using a suitable API call

System: initialize and store the cluster's unique ID for later use

2.2.1.3. kubeconfig

Description: fetch the kubeconfig and other required config and switch this context

System&Output: Status is shown and context is changed

2.2.2. Delete Cluster

STATUS

2.2.1.1. Get required and additional info

Description: collect the name, region, and more

Input: Enter cluster name and region

2.2.1.2. delete config and delete cluster

Description: initialize the config for the cluster to be made and delete the cluster by using a suitable API call

System&Output: cluster is deleted

2.2.1.3. kubeconfig

Description: remove the kubeconfig and other related configs switch back to previous context

System&Output: Status is shown and context is changed

2.2.3. Init Auth

STATUS

2.3.3.1. Get API Key

Description: users save their API token for later use

System: updated the credential file

2.3.3.2. Verify for validity

Description: ping with Cloud provider api to get whether API token is valid or not

2.2.3. View Clusters/Profiles

STATUS

2.2.3.1. View all cluster option

Description: will output all the clusters created by this tool

Output: JSON output

2.2.4. Context Switch

STATUS

2.4.4.1. View all cluster option

Description: will output all the clusters created by this tool

Output: JSON output

2.2.5. Context Switch

STATUS

2.3.5.1. Accept input from user

Description: parameters by the users will be validated and correct config will be shown

Output: output of command or set Environment

2.3. Civo K3s

2.3.1. Create Cluster

STATUS

2.3.1.1. Get required and additional info

Description: collect the name, region, and more

Input: Enter cluster name and region

2.3.1.2. Create config and create cluster

Description: initialize the config for the cluster to be made and create a cluster by using a suitable API call

System: initialize and store the cluster's unique ID for later use

2.3.1.3. kubeconfig

Description: fetch the kubeconfig and other required config and switch this context

System&Output: Status is shown and context is changed

2.3.2. Delete Cluster

STATUS

2.3.1.1. Get required and additional info

Description: collect the name, region, and more

Input: Enter cluster name and region

2.3.1.2. delete config and delete cluster

Description: initialize the config for the cluster to be made and delete the cluster by using a suitable API call

System&Output: cluster is deleted

2.3.1.3. kubeconfig

Description: remove the kubeconfig and other related configs switch back to previous context

System&Output: Status is shown and context is changed

2.3.3. Init Auth

STATUS

2.3.3.1. Get API Key

Description: users save their API token for later use

System: updated the credential file

2.3.3.2. Verify for validity

Description: ping with Cloud provider api to get whether API token is valid or not

2.3.4. View Clusters/Profiles

STATUS

2.3.4.1. View all cluster option

Description: will output all the clusters created by this tool

Output: JSON output

2.3.5. Context Switch

STATUS

2.3.5.1. Accept input from user

Description: parameters by the users will be validated and correct config will be shown

Output: output of command or set Environment

Non-Functional Requirements

1. Constraints

Local System

Hardware: RAM > 6Gi and CPU > 3 cores

Software: Linux, Windows, and macOS with Docker

Cloud providers

Hardware: N/A

Software: Depends on providers (PaaS)

2. Safety

- Local Environment
 - Deleting one cluster does not delete other cluster's containers
 - Preserve the data till the cluster is deleted
- Cloud provider
 - As PaaS offering infrastructure is provider's responsibility, but the application and operational safety is the responsible of admins

3. Security

- LocalEnvironment
 - The containers should not be accessible by other containers running
- Cloud provider
 - Application security and other operational security like RBAC, etc.

@@END

Related links

<https://kubernetes.io/docs/reference/ports-and-protocols/>

<https://github.com/kubesimplify/ksctl/blob/main/docs/proposals.md>

23 Nov 2022 ksctl meeting

Agenda items

Kubeadm usage

- ☒ ~~Explore which cloud providers are giving us SDK to run custom bash script (-dipankardas0115@gmail.com-)~~
- ☒ ~~Evaluate how to get the logs after sshing into the terminal(-avineshtrpathi1@gmail.com-)~~
- ☒ Restrict firewalls
- ☒ Try to create Load Balancer and attach each control plane to it (-Dipankar Das, contact.anurag7@gmail.com, avineshtrpathi1@gmail.com-)
- ☒ Check the status of each control plane
- ☒ Added worker nodes