

Deploying SMILE with Kubernetes and Helm Chart

SMILE can be deployed using Kubernetes clusters.

For more information about kubernetes, please visit [\[here\]\(https://kubernetes.io/\)](https://kubernetes.io/)

Prerequisite

To deploy SMILE, kubernetes clusters be ready.

- Deployment can be done by using helm chart
 - there are two versions one with the ArgoCD and the other is without using ArgoCD
 - [ArgoCD](#) streamlines Kubernetes application deployment through declarative GitOps practices, enabling version-controlled configuration, automated updates, and enhanced collaboration among teams.
 - ArgoCD version of helm chart is in [here](#)
 - The users who are not familiar or want to control directly can use the helm chart without using ArgoCD in [here](#)
 - there is helm folder in the repository
- In this documentation, everything will be explained based on the assumption that there is kubernetes cluster already set up.
- Kubernetes cluster should already have traefik installed.
- There should be kubectl command and helm installed in local machine to deploy SMILE.

Secrets

There are some protected information that should not be revealed in the public repository.

These information should be kept in the kubernetes secret file.

Currently, SMILE needs several passwords or secret information to be protected.

If there is any necessity to create one, create a secret.yaml file under smile/templates folder in helm chart, then add necessary information in the file.

Deployment Steps

Create a name space SMILE

SMILE will be deployed in the namespace of the SMILE so the other packages, such as BAE, can also be deployed in the same cluster without hindering each other at all.

Create a name space first by using the following command

```
kubectl create namespace smile
```

Create a secret for obtaining the containers (optional)

If you need to create a secret to download the docker images from the repositories that needs the credential, create a secret using the following command

```
kubectl -n smile create secret generic regcred --from-file=.dockerconfigjson=regcred.json --type=kubernetes.io/dockerconfigjson
```

This command will create a secret named regcred from the file regcred.json, and regcred.json should contain the credential information about accessing the repository for the containers that are used to SMILE

Modifying Values

- All the necessary information, such as container name, tags, and other controlling parameters are recorded in values.yaml
- You can directly modify the values in there, or you can create a supplementary files that contains the only information that you frequently modify and updates, and name it something like values-smile.yaml.
- The information in the values-smile.yaml will override the information in values.yaml when you deploy, if you choose to use it.

Deployment

Deploy helm chart by following command

```
helm upgrade --namespace smile smile . --values values-smile.yaml
```

Check out the result

After deploying the chart, check the URL to see if smile works correctly from the URL of https://smile.{server_main_url}

Examples

- values-smile.yaml

```
# To deploy very first time (probably done already)
# kubectl create namespace smile
# kubectl -n smile create secret generic regcred --from-file=.dockerconfigjson=regcred.json --
type=kubernetes.io/dockerconfigjson
#
# To deploy new version
# helm upgrade --install --namespace smile --create-namespace smile . --values values-smile-dev.yaml
# need to set
# - jupyterhub.proxy.secretToken
# - postgresql.postgresqlPassword

hostname: &hostname smile.smm.ncsa.illinois.edu
storage: &storage csi-cinder-sc-retain

image:
  pullPolicy: Always

ingress:
  traefik: true
  hosts:
    - host: *hostname

cors:
  origin:
    - http://localhost:3000/
    - http://localhost:5000/

# -----
# DEPLOYMENT
# -----
smile_server:
  replicas: 1
  image:
    tag: 0.3.0
  persistence:
    size: 10Gi
    storageClass: *storage

smile_graphql:
  replicas: 1
  image:
    tag: 0.2.8

algorithm_autophrase:
  replicas: 1
  image:
    tag: 0.1.4

algorithm_classification_predict:
  replicas: 1
  image:
    tag: 0.1.2

algorithm_classification_split:
  replicas: 1
  image:
    tag: 0.1.2

algorithm_classification_train:
  replicas: 1
  image:
    tag: 0.1.2

algorithm_histogram:
  replicas: 1
  image:
    tag: 0.1.1
```

```

algorithm_network_analysis:
  replicas: 1
  image:
    tag: 0.1.3

algorithm_preprocessing:
  replicas: 1
  image:
    tag: 0.1.3

algorithm_sentiment_analysis:
  replicas: 1
  image:
    tag: 0.1.5

algorithm_screen_name_prompt:
  replicas: 1
  image:
    tag: 0.1.1

algorithm_topic_modeling:
  replicas: 1
  image:
    tag: 0.1.3

algorithm_name_entity_recognition:
  replicas: 1
  image:
    tag: 0.1.2

image_crawler:
  replicas: 1
  image:
    tag: 0.1.2

collect_reddit_comment:
  replicas: 1
  image:
    tag: 0.1.2

# -----
# DEPENDENCIES
# -----

# -----
# rabbitmq
# -----
rabbitmq:
  persistence:
    size: 1Gi
    storageClass: *storage
  ingress:
    hostname: rabbitmq.smm.ncsa.illinois.edu

# -----
# minio
# -----
minio:
  persistence:
    size: 10Gi
    storageClass: *storage
  ingress:
    hostname: minio.smm.ncsa.illinois.edu
  apiIngress:
    hostname: minio-api.smm.ncsa.illinois.edu

```

- secret.yaml

```

---
apiVersion: v1
kind: Secret
metadata:
  name: {{ include "smile.fullname" . }}-server
  labels:
    {{- include "smile.labels" . | nindent 4 }}
stringData:
  HOST_IP: "smile-minio"
  MINIO_URL: "URL for the minio"
  REDIS_URL: "redis://smile-redis-master:6379"
  RABBITMQ_URL: "amqp://smile-rabbitmq:5672"
  SMILE_GRAPHQL_URL: "http://smile-graphql:5050/graphql"
  RABBITMQ_HOST: "smile-rabbitmq"
  REDDIT_CALLBACK_URL: "Call back url for REDDIT"
  CILOGON_CLIENT_ID: "CILOGON client ID"
  CILOGON_CLIENT_SECRET: "CILOGON client secret"
  REDDIT_CLIENT_ID: "REDDIT client ID"
  REDDIT_CLIENT_SECRET: "REDDIT client secret"
  BOX_CLIENT_ID: "BOX client ID"
  BOX_CLIENT_SECRET: "BOX client secret"
  DROPBOX_CLIENT_ID: "DROPBOX client ID"
  DROPBOX_CLIENT_SECRET: "DROPBOX client secret"
  GOOGLE_CLIENT_ID: "Google client ID"
  GOOGLE_CLIENT_SECRET: K8m-"Google client secret"
  EMAIL_HOST=smtp.ncsa.illinois.edu
  EMAIL_FROM_ADDRESS=devnull+smm@ncsa.illinois.edu
  EMAIL_PORT=25
  EMAIL_PASSWORD=

---
apiVersion: v1
kind: Secret
metadata:
  name: {{ include "smile.fullname" . }}-minio-secret
  labels:
    {{- include "smile.labels" . | nindent 4 }}
stringData:
  root-user: "minio root user"
  root-password: "minio root user password"

---
apiVersion: v1
kind: Secret
metadata:
  name: {{ include "smile.fullname" . }}-graphql
  labels:
    {{- include "smile.labels" . | nindent 4 }}
stringData:
  TWITTER_CONSUMER_KEY: "Twitter consumer key"
  TWITTER_CONSUMER_SECRET: Twitter consumer secret

```