

**KAFKA\_service\_deploy\_방법**

KAFKA

Rev 1.0, 2023/03/14



<문서의 이력>

| 버 전 | 날 짜 | 작 성 | 변경사항 |
| --- | --- | --- | --- |
| 1.0 | 2023-03-14 | 한태현 | 최초 작성 |
| 1.1 | 2023-03-15 | 한태현 | 오타 수정 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[**1. 필요 환경 3**](#_khb5ben0ktf5)

[**2. 사전 준비 4**](#_lt4wzk3r8hy2)

[**3. 설치 방법 6**](#_3tgmlop7anq5)

[**4. 서비스 접근 9**](#_dypxh6vhbtee)

KAFKA\_service\_deploy\_방법

# 필요 환경

* 1. kubernetes (ser : v1.17.4 / cli : v1.24.0)
  2. kubectl 을 이용한 kubernetes 제어 환경

| $ curl -LO "https://dl.k8s.io/release/$(curl -L -s <https://dl.k8s.io/release/stable.txt>)/bin/linux/amd64/kubectl"  $ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl |
| --- |

확인 방법

| $ kubectl version  WARNING: This version information is deprecated and will be replaced with the output from kubectl version --short. Use --output=yaml|json to get the full version.  Client Version: version.Info{Major:"1", Minor:"24", GitVersion:"v1.24.0",  – 중략 –  supported minor version skew of +/-1 |
| --- |

* 1. Helm v3 이상

설치 방법

| $ curl -fsSL -o get\_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3  $ chmod 700 get\_helm.sh  $ ./get\_helm.sh |
| --- |

확인 방법

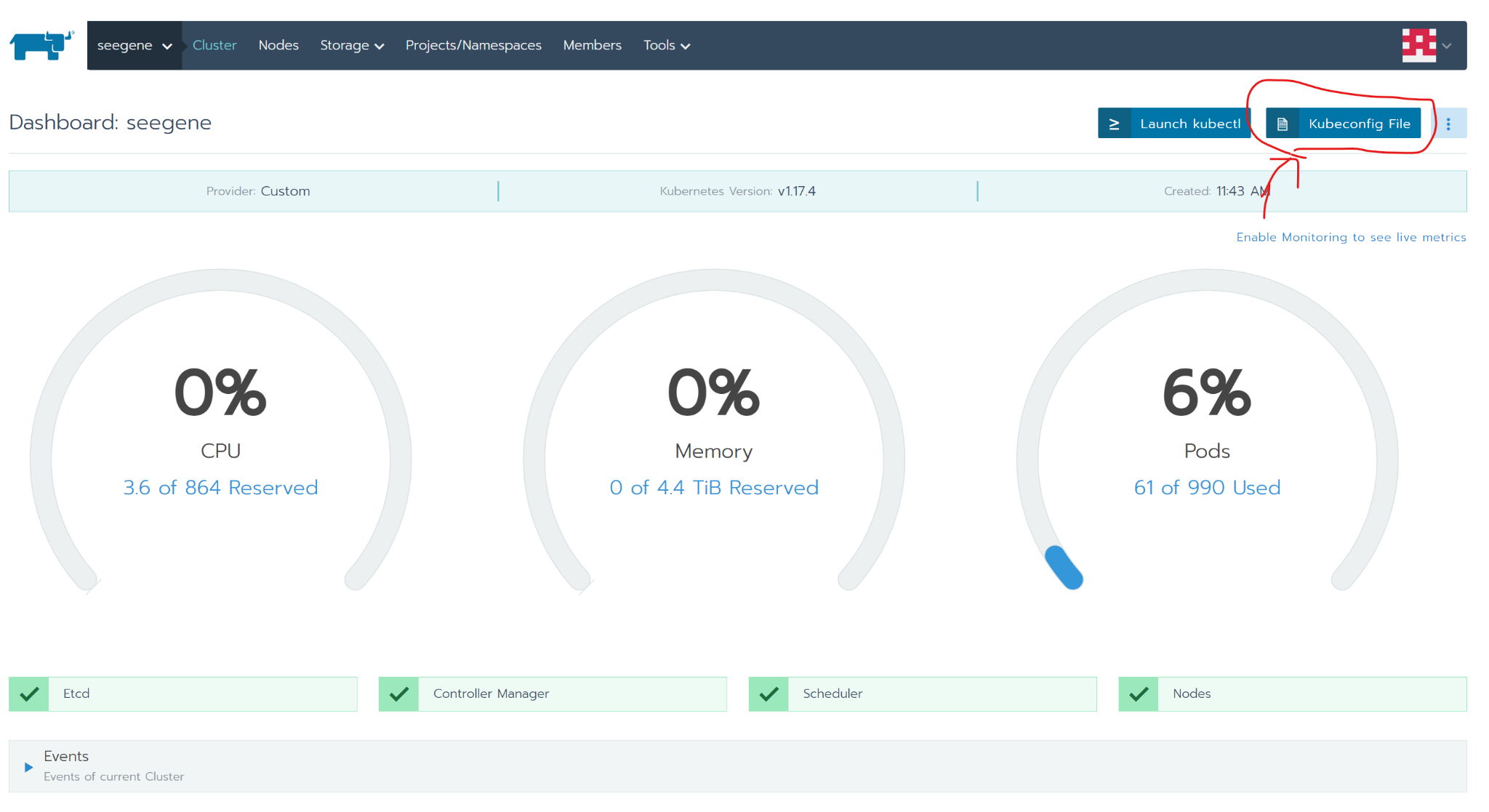
| $ helm version  version.BuildInfo{Version:"v3.8.2", GitCommit:"6e3701edea09e5d55a8ca2aae03a68917630e91b", GitTreeState:"clean", GoVersion:"go1.18.1"} |
| --- |

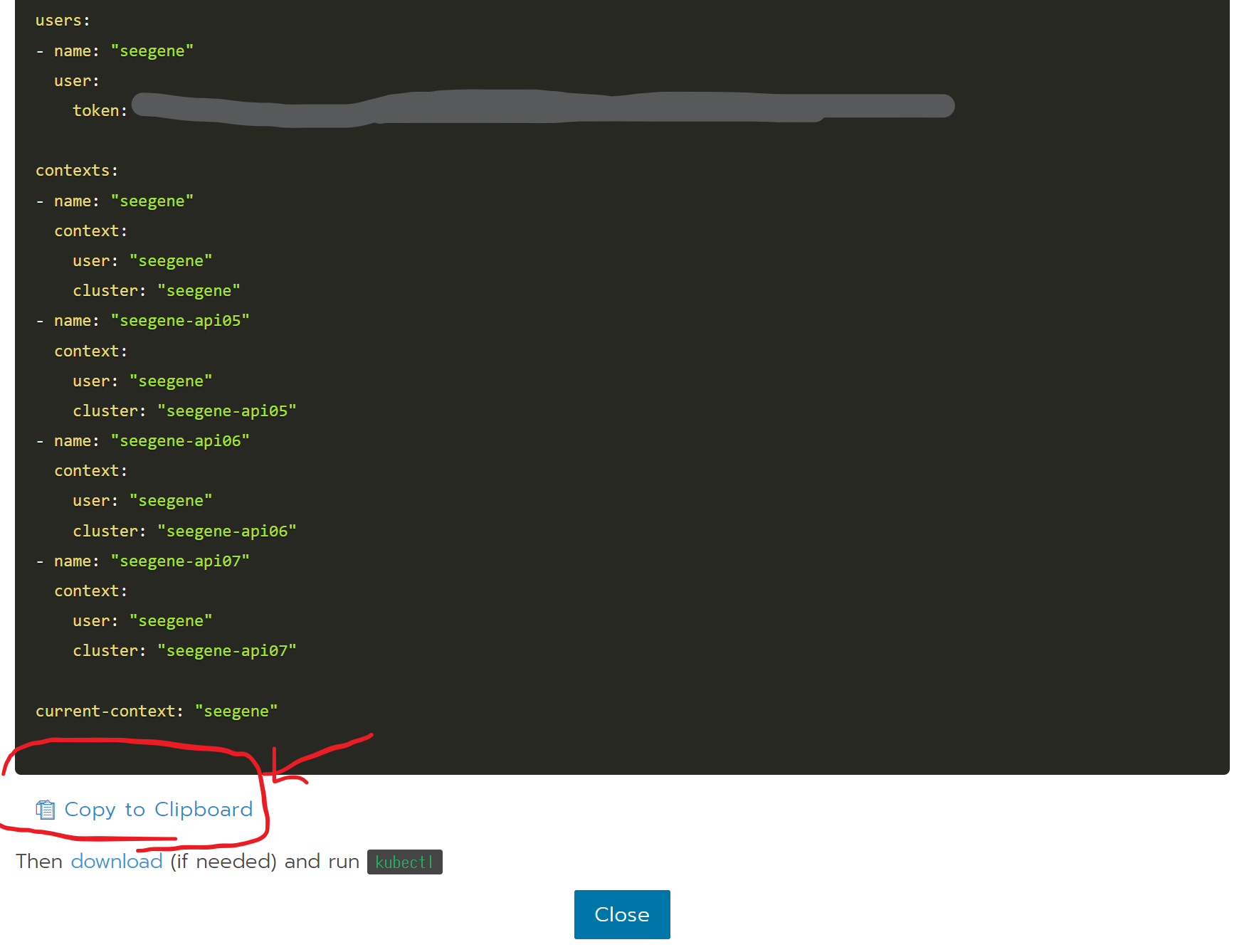
# 

# 

# 사전 준비

* 1. 개발머신의 kubectl 설정
     1. Cluster 의 config file을 복사 한다.





* + 1. 개발 머신의 .kube/config 내용을 수정한다.

| $ vi ~/.kube/config  apiVersion: v1  kind: Config  clusters:  - name: "main"  cluster:  server: "https://rancher.nemopai.com/k8s/clusters/c-qf5nw"  users:  - name: "main"  user:  –중략– |
| --- |

* + 1. 개발 머신에서 클러스터 제어가 가능한지 확인

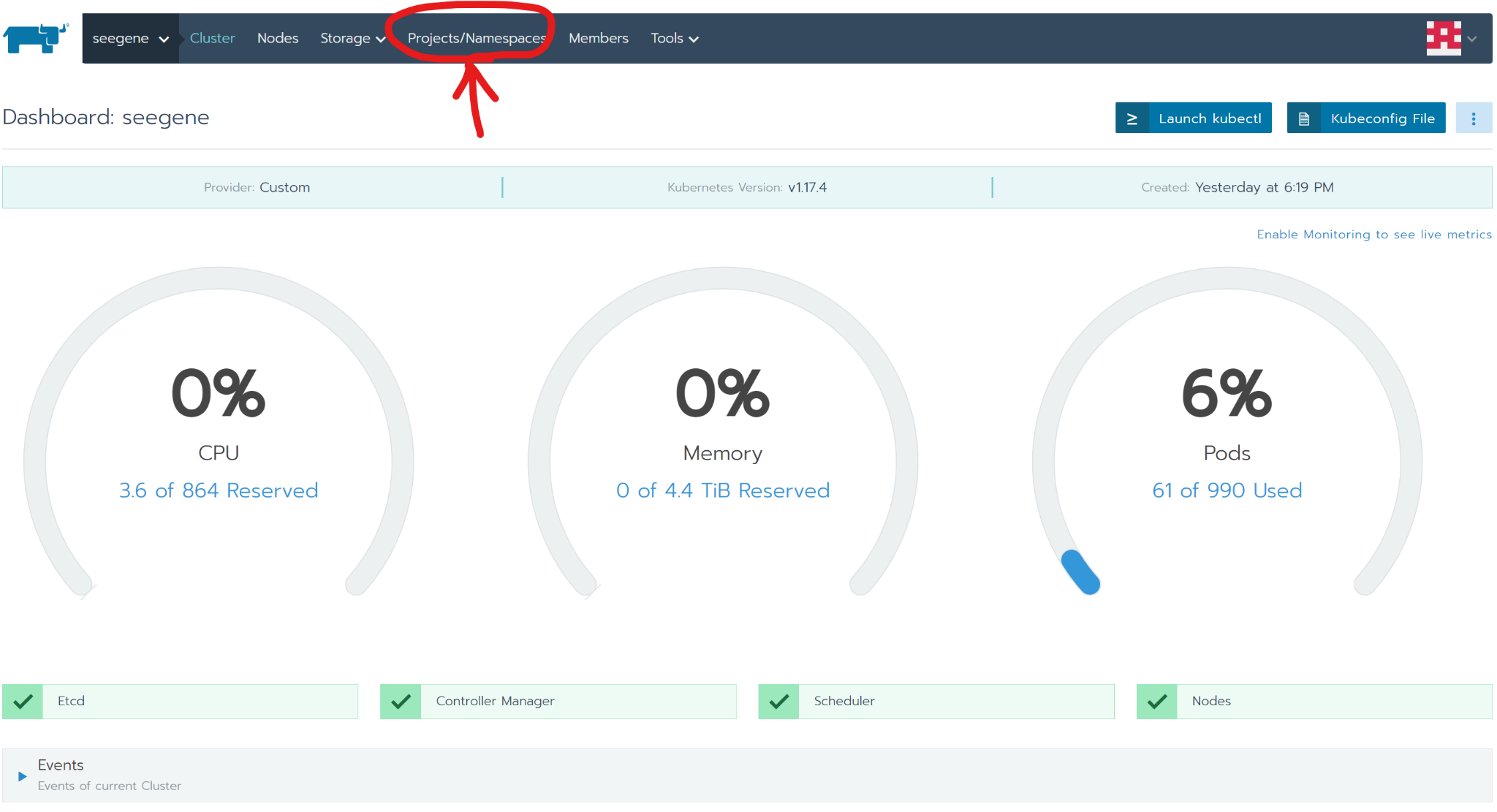
| $ kubectl get all  NAME READY STATUS RESTARTS AGE  pod/glusterfs-dgnb2 1/1 Running 0 162m  pod/glusterfs-jhzsd 1/1 Running 0 162m  pod/glusterfs-l8nst 1/1 Running 0 162m  pod/glusterfs-pqnqm 1/1 Running 0 162m  pod/glusterfs-sn9gr 1/1 Running 0 162m  pod/glusterfs-ssvnc 1/1 Running 0 162m  pod/heketi-6f5df899d5-mrwxm 1/1 Running 0 160m  NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  service/glusterfs-dynamic-2c7b6557-33c9-455e-a4ae-7f2d55941b0a ClusterIP 10.43.123.223 <none> 1/TCP 33m  service/glusterfs-dynamic-434ac56e-a8e5-4468-8962-62452ba4a2bc ClusterIP 10.43.167.173 <none> 1/TCP 13m  service/glusterfs-dynamic-a5d27412-c0d2-4a50-9a0e-9ee452cb3fe1 ClusterIP 10.43.137.79 <none> 1/TCP 49m  service/glusterfs-dynamic-ec036e6f-4630-4275-895a-8c2eaab5b4ff ClusterIP 10.43.238.21 <none> 1/TCP 46m  service/heketi ClusterIP 10.43.15.144 <none> 8080/TCP 160m  service/heketi-storage-endpoints ClusterIP 10.43.142.123 <none> 1/TCP 161m  service/kubernetes ClusterIP 10.43.0.1 <none> 443/TCP 3h10m  NAME DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR AGE  daemonset.apps/glusterfs 6 6 6 6 6 storagenode=glusterfs 162m  NAME READY UP-TO-DATE AVAILABLE AGE  deployment.apps/heketi 1/1 1 1 160m  NAME DESIRED CURRENT READY AGE  replicaset.apps/heketi-6f5df899d5 1 1 1 160m |
| --- |

* 1. 프로젝트 git repository download

| $ git clone git@github.com:seegenelab/insilico-devops.git |
| --- |

* 1. kubernetes 에 kafka helm chart 추가

| $ helm repo add bitnami https://charts.bitnami.com/bitnami |
| --- |

* 1. kubernetes 상에 kafka namespace 추가 (Rancher)

# 

# 

# 설치 방법

* 1. kafka 의 deploy 에 앞서 service 의 설정을 한다.

| $ cd k8s/seegene-kafka  $ cd chart/kafka  $ vi values.yaml |
| --- |

* + 1. 주요 설정

replicaCount

broker 의 갯수 설정 파라미터 (rancher node 의 갯수에 영향을 받음)

externalAccess

Kafka endpoint 에 접근할 export endpoint 설정

kubernetes cluster 밖에서 kafka 에 접근 할 때 필요한 설정

| replicaCount: 3  –중략–  externalAccess:  enable: true  –중략–  nodePorts:   * 31090 # broker 별로 해당 포트로 접근 가능해진다. * 31091 * 31092 |
| --- |

* 1. kafka 서비스를 deploy 한다.

| $ cd ../../cli/  $ ./001-kafka-helm-deploy.sh |
| --- |

* 1. kafka 서비스 삭제

| $ cd cli/  $ ./002-kafka-helm-delete.sh |
| --- |

* 1. (선택사항) kafka ui 서비스를 Deploy 하기위해 namespace 에 인증서를 추가한다.

| $ cd cert/  $ ./001-install-cert.sh |
| --- |

* 1. kafka ui 서비스 deploy 를 위해 서비스의 value.yaml 파일을 수정한다.

| $ cd chart/kafka-ui/  $ vi values.yaml  –중략–  ingress:  enabled: true  host: “kafka.seegene.com” # DNS 설정해 놓은 웹 서비스 접속 주소  tls:  enabled: true  secretName: “kafka-seegene-com-tls” # ssl 인증서 주소  –중략– |
| --- |

* 1. kafka ui 서비스를 deploy 한다.

| $ cd ../../cli/  $ ./003-kafka-ui-helm-deploy.sh |
| --- |

# 서비스 접근

* 1. kafka endpoint
     1. Kubernetes 내부 서비스 에서 접속할때

endpoint : kafka.kafka:9092 ([svcname].[namespace]:[svcport])

예시 ) kubernetes 의 pod 로 동작중인 ubuntu 에서 접근할 경우

| falinux@kafka-ubuntu-6b8c758cc5-9h6dp:~$ kafkacat -b kafka.kafka:9092 -L  Metadata for all topics (from broker -1: kafka.kafka:9092/bootstrap):  3 brokers:  broker 0 at kafka-0.kafka-headless.kafka.svc.cluster.local:9092 (controller)  broker 2 at kafka-2.kafka-headless.kafka.svc.cluster.local:9092  broker 1 at kafka-1.kafka-headless.kafka.svc.cluster.local:9092  5 topics:  topic "mas-ssh-result" with 1 partitions:  partition 0, leader 2, replicas: 2, isrs: 2  topic "topic-no1" with 1 partitions:  partition 0, leader 0, replicas: 0, isrs: 0  topic "\_\_consumer\_offsets" with 50 partitions:  partition 0, leader 2, replicas: 2, isrs: 2  partition 1, leader 1, replicas: 1, isrs: 1  partition 2, leader 2, replicas: 2, isrs: 2  –중략– |
| --- |

* + 1. Cluster 내부 노드에서 접속 할때

endpoint : 10.10.100.35:31090 ([rancher nodename]:[externalport])

예시 ) Cluster 내부망에 포함된 노드에서 접근할때 (ex/ pandora node local )

| ubuntu@pandora07:~$ kafkacat -b 10.10.100.35:31090 -L  Metadata for all topics (from broker -1: 10.10.100.35:31090/bootstrap):  3 brokers:  broker 0 at 10.10.100.32:31090 (controller)  broker 2 at 10.10.100.36:31092  broker 1 at 10.10.100.33:31091  5 topics:  topic "mas-ssh-result" with 1 partitions:  partition 0, leader 2, replicas: 2, isrs: 2  topic "topic-no1" with 1 partitions:  partition 0, leader 0, replicas: 0, isrs: 0  –중략– |
| --- |

* 1. kafka ui
     1. 웹 브라우저에 kafka.seegene.com

|  |
| --- |

* + 1. 기능
       - broker 상태 확인
       - topic 현황 확인
       - topic 메세지 확인
       - consumer 목록 및 상태 확인