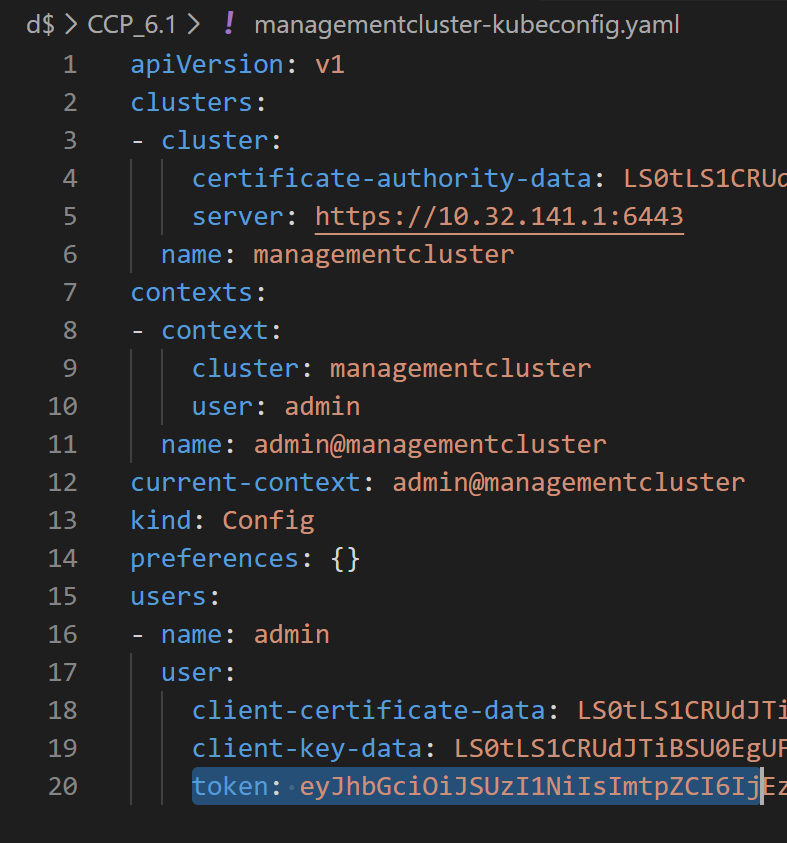
**Anleitung für die Cluster-Erstellung unter CCP 6.1:**

* Im CCP „New Cluster“ anklicken und den Cluster-Erstellungsprozess durchlaufen
* Wenn das Cluster erstellt wurde, müssen die Add-Ons installiert werden. Hierfür auf das erstellte Cluster klicken, dann auf den Reiter Add-Ons und dann auf den Button „Install Add-Ons“. Hier nun die gewünschten Add-Ons auswählen (eines zurzeit).
* Um an die IP-Adresse des Loadbalancers vom Cluster zu kommen, muss man sich per ssh mit dem Master-Knoten verbinden und dort folgenden Befehl eingeben:
  + kubectl get svc nginx-ingress-controller -n ccp
* Die “External IP-Address” kann man dann für die einzelnen Services (Add-Ons) nutzen, beispielsweise: „https://“External IP“/dashboard/“ um das Kubernetes Dashboard aufzurufen
* Damit man sich an dem Dashboard anmelden kann, muss zunächst der token ausgelesen werden. Hierfür muss man auf dem Master Knoten folgende Befehle ausführen:
  + kubectl get secrets -n kube-system 🡪 Suche nach dem secret „default-token-XXXXX“ und ersetze den folgenden Befehl damit
  + kubectl describe secret default-token-XXXXX -n kube-system
* Das Ergebnis dieses Befehls kann man entweder direct als Token zum Anmelden verwenden, oder man kopiert diesen und fügt ihn ans Ende der Kubeconfig-File ein, s. Screenshot. Dann kann man die kubecon-file zum Anmelden nutzen



**Manuelle Nacharbeiten in allen Clustern (Fluentd)**

Nachdem man das Add-On „Logging“ (ccp-efk) installiert hat, muss die ConfigMap im Cluster angepasst werden:

kubectl edit cm -n ccp fluentd-es-config-v0.1.1

 <source>

      type tail

      path /var/log/containers/\*.log

      pos\_file /var/log/es-containers.log.pos

      time\_format %Y-%m-%dT%H:%M:%S.%NZ

      tag kubernetes.\*

      exclude\_path "#{ENV['FLUENT\_CONTAINER\_TAIL\_EXCLUDE\_PATH'] || use\_default}"   <---- Add this line

      read\_from\_head true

      format multi\_format

      <pattern>

        format json

        time\_key time

        time\_format %Y-%m-%dT%H:%M:%S.%NZ

      </pattern>

      <pattern>

        format /^(?<time>.+) (?<stream>stdout|stderr) [^ ]\* (?<log>.\*)$/

        time\_format %Y-%m-%dT%H:%M:%S.%N%:z

      </pattern>

    </source>

Danach muss noch das Deployment im Cluster angepasst warden:

kubectl edit ds -n ccp fluentd-es-v2.0.2

spec:

      containers:

      - env:

        - name: FLUENTD\_ARGS

          value: --no-supervisor -q

        - name: FLUENT\_CONTAINER\_TAIL\_EXCLUDE\_PATH   <---- Add this line

          value: /var/log/containers/fluentd\*        <---- Add this line

        image: registry.ci.ciscolabs.com/cpsg\_base-apps/fluentd-elasticsearch/releases/fluentd-elasticsearch:v2.4.0-cisco2

        imagePullPolicy: IfNotPresent

        name: fluentd-es

        resources:

          limits:

            memory: 500Mi

          requests:

            cpu: 100m

            memory: 200Mi

**Manuelle Nacharbeit im Managementcluster**

Bei der Erstellung des Managementclusters ist noch zu beachten, dass das certmananger Root Zertifikat auf alle Nodes des Managementclusters im Nachgang aufgebracht werden muss, da sonst ein Deployment aus dem Harbor heraus fehlschlägt, da die docker engine offenbar das certmananger – Zertifikat nicht kennt

ToDo:

Einloggen auf die jeweilige Node mit Putty

Befehle:

Cd ..

Cd ..

Cd usr/local/share/ca-certificates

Sudo Touch root.crt

Sudo vi root.crt

Im Editor mit I das Root-Zertifikat reinkopieren

Beispiel:

-----BEGIN CERTIFICATE-----

MIIEOTCCAqGgAwIBAgIRAOCc3Gq/v2KG6VRFQeTs9mQwDQYJKoZIhvcNAQELBQAw

NzEaMAwGA1UEChMFY2lzY28wCgYDVQQKEwNjY3AxGTAXBgNVBAMTEGNjcC1jZXJ0

LW1hbmFnZXIwHhcNMjAwNTA4MDkxMjU0WhcNMzUwNTA1MDkxMjU0WjA3MRowDAYD

VQQKEwVjaXNjbzAKBgNVBAoTA2NjcDEZMBcGA1UEAxMQY2NwLWNlcnQtbWFuYWdl

cjCCAaIwDQYJKoZIhvcNAQEBBQADggGPADCCAYoCggGBAKpPsGgr+ZnJ7xJ/GiC/

sENtLSRxpaiLFct8lq+LY+tkOROXRfEuUM6/hh1CMR7xdpmtxnaGA4SSPLFpD9c5

5/iSFgL3KNk+R3oLrSVXwv9VjjmYxZrWfzAPsbBPEJZzfVHXjaWOhTf4yYkxuWng

w0LuhwR7RjsoTkyWMATxhcU23+bN+wYceFppKsBlBfdxi0+cYQV7rcXp6KkO68fM

0p9IOP4Hb3K6YegLDjvnzUTO1XvombjIusk/MyGKVUmMREna29eU3A2HWuuOp898

FIU3YFrrQMYNS2PSs/k9P4xJXiYc+62+e1+3eJtfxIPEZ4Y4wpUZIwp3HiMe26FG

gtIwZro+qCeyaAR4keT0EVIhDfBdRdDenKh3cNgB7q0ZWwn4BFPgoTYkfzaE/g1E

Z1pE00gqal0DxD1+5fPksId52TvsTCA3L7rsG0DIUFmKW90jiEym9cBWrbQ2AphE

Wr/K1OSH8kMzBTK69/F4tBMeHGLAF9MjeDPzUu6PoN1ihQIDAQABo0AwPjAOBgNV

HQ8BAf8EBAMCAqQwDwYDVR0TAQH/BAUwAwEB/zAbBgNVHREEFDASghBjY3AtY2Vy

dC1tYW5hZ2VyMA0GCSqGSIb3DQEBCwUAA4IBgQA63yYXbWHSsZ7NIlQjbogEJuKI

DlnlGYD6mLJ5qh/Zifp+nhUfiWsUAa7rEVmlOXpCtMOn5lqS912yNACDtiF/MV9b

T1JlhIECjq5zwLYv4H4LfoMLSrzKAGJKzPtYQnTxuELt3mqnIeqOlXZlMxCFUv+e

+KedB5129L9LGB3b7DaLZTqaHb+TQxl5M5ZPtm6QQ6kg6BJE2OXuoefQ0RpTXdgk

XAMbHw6KTXmMz8/jL+s8Di9eN5z+EWDT8AICqST53L+/WrRD/9TeCx8duYF03E9z

6dtbMFcE0vzxo0WGvU019WY4x2X31kyedBFSGI4DOYeuZTlv0OsWzeW4UPzmnahw

bygTIUVBaCSJ1cKWVRls4bVDXy7/CxYfT1OW42O93ujynarYfMNuKe5FUX5fcehB

m9vRq9pN9BmLcZDJqIAR4TVbUpGl+RsxF7jHu55HNiwSu57b1eCYmzD8rdxq1cfe

aZg1RbHbqlIQlflDbAURODESSxvESyd8nwtYRU4=

-----END CERTIFICATE-----

Über :wq den vi-editor verlassen

Sudo update-ca-certificates ausführen

Sudo service docker restart ausführen

Überprüfen des Harbor login über beispielsweise: sudo docker login 10.32.141.9

