

Opis problému:

Ťažko dostupná a neprehľadná dokumentácia podov v Kubernetes API

Architektúra:

- Kubernetes API ktorá zbiera informácie z Kubernetes clustru
- Golang service, ktoré cez client-go komunikujú s Kubernetes API
 - generovanie dokumentácie v asciidoc a v JSON formáte
 - pravidelné ukladanie dokumentácie do MongoDB

Stav - použitie:

- uvedieme si jeden z príkladov z výsledných endpointov ktorý vracia informácie v podobe JSON
- informácie z endpoints sú vložené do ascii tabuľky pre lepší výstup

```
"podInformations": [  
  {  
    "podName": "cluster-processor-service-68d76b4d89-vm9xt",  
    "namespace": "project-atlas-system",  
    "hostIp": "192.168.65.3",  
    "podIP": "10.1.0.18",  
    "startTime": "2021-01-22 11:02:33 +0000 UTC",  
    "volumeName": "cluster-processor-service-token-xbz65",  
    "volumeMount": "/var/run/secrets/kubernetes.io/serviceaccount",  
    "cpuPodName": "cluster-processor-service",  
    "cpuUsage": "462849n",  
    "memoryUsage": "10996Ki",  
    "configMapName": "projct-atlas-configmap",  
    "nodeName": "",  
    "nodeMemory": "",  
    "mdbPort": 3000,  
    "mExpressPort": 0,  
    "cpsPort": 0  
  },  
]
```

Vlastnosti:

- jednoduchý a prehľadný spôsob získavania informácií o clusteru
- získanie runtime informácií (cpu, memory usage) a následne systematické uloženie týchto informácií do databázy
- generovanie prehľadnej dokumentácie v ascii docu.

Tento proces je jednoducho aplikovateľný na rôzne druhy aplikácií/servisov bežiacich v Kubernetes

Vyhodnotenie riešenia:

Funkcionalita bude slúžiť na generovanie dokumentácie v iných projektoch.