

# Scavenger Jobs PoC

Je dôležité, aby fungovali všetky štyri komponenty, ktoré boli v návrhu a spolu tvorili nejak funkčný systém. Počítame s tým, že niektoré funkcionality (napr. To s tou prioritou) budú implementované "hlúpo", teda že nebudú robiť presne to, čo by sme chceli, ale ide o ten myšlienkový postup.

Dôležité je, aby sme od začiatku mysleli na to, že sa chcem vyhnúť plytvaniu zdrojov samotnými SJ ak by boli ukončené tesne pred tým, než u nich nastane checkpoint. Rovnako, chceme predísť zahlteniu clustru, čo sa môže stať buď tým, že by nový SJ už prekročil nejaký práh alebo že sa aktuálne celkovo zvýšila spotreba a teda je prekročený práh.

Implementovala by som nasledujúce. Myslím si, že to bude stačiť na dlpompk v pohode ešte aj s doporodným textom keďže okrem operatora spomínam aj validating webhook. Niektoré časti sú založené dokonca na článkoch (počítanie prahu).

## Komponenta 1: Controller spolu s CRD

- Musí existovať definícia SJ ako yaml (tá openAPI schéma myslím)
- Controller musí byť schopný vytvoriť Scavenger Job (job s controllerom na SJ, tak ako to už máš)
  - o Zatiaľ bez akýchkoľvek prídavkov, modifikovaných front a iného. Proste sa vytvorí SJ, ak neprekračuje práh zaplnenia svojím vytvorením
    - Rozhodnutie, či SJ prekračuje prah alebo nie by som vyňala von z controlleru, neni jeho starosť to počítať, iba reagovať na udalosti, ktoré sa udejú so Scavenger Jobom a nejak ho "rozhodia"
    - Tu mi napadá myšlienka, popísaná nižšie za ružovým nadpisom
- Controller musí byť schopný zmazať Scavenger Job ak je signalizované, že sa má zmazať
  - o Zatiaľ bez akýchkoľvek prídavkov V controlleri
  - o Controller by sa mal postarať o vhodné Zmazanie – cleanup, finalization, error handling
  - o SJ sa má zmazať zatiaľ vtedy
    - keď sa prekročí celkové zaplnenie clustru
      - To rieši komponenta 2
    - Keď je jeho miesto nutné pre vyššie prioritný iný workload
      - To by mal riešiť ten postFilter (jak preemption) na základe toho preemptible labelu (napr. cerit.io/sj-preemptible)
- Controller musí byť schopný monitorovať running SJ
  - o Meniť týmto label sj-preemptible podľa vzdialenosti od posledného checkpointu
  - o To je v kompetencii controllera pretože táto zmena sa počíta rovno z polícií, ktoré sú v specu (abo statuse) Scavenger Jobu a súvisia iba s nimi, nezávisle na ničom externom
  - o Popísané v [https://ucnmuni-my.sharepoint.com/:w:/g/personal/456543\\_muni\\_cz/EWd-3Gtn5m9MI9GxrN02TcIBMbezxsuLdD9Zv35PerAQ?wdOrigin=TEAMS-MAGLEV.p2p\\_ns.rwc&wdExp=TEAMS-](https://ucnmuni-my.sharepoint.com/:w:/g/personal/456543_muni_cz/EWd-3Gtn5m9MI9GxrN02TcIBMbezxsuLdD9Zv35PerAQ?wdOrigin=TEAMS-MAGLEV.p2p_ns.rwc&wdExp=TEAMS-)

[TREATMENT&wdhostclicktime=1707819744366&web=1](#) komponenta 3 prideľovanie priorít pre bežiaci SJ, ale vlastne je to menenie preemptible-nonpreemptible

- Toto sa môže diať ako reconcile na čas
- Controller musí byť schopný reagovať na zmeny SJ
  - Napr zmena statusu jobu, podu, propagácia nejakých správ ap.
- Z toho vychádza subscribe minimálne na create, update, delete

Komponenta 2: Regulátor práhu

- Bavíme sa o úrovni zaplnenia clustru
  - Eventuelne sa asi bude riešiť aj zaplnenie jednotlivých nodov v rámci clustru
- Extra komponenta, ktorá počíta zaplnenie
- Popísaná v [https://ucnmuni-my.sharepoint.com/:w:/g/personal/456543\\_muni\\_cz/EWd-3Gtn5m9MI9GxrN02TcIBMbezxsuLdD9Zvdv35PerAQ?wdOrigin=TEAMS-MAGLEV.p2p\\_ns.rwc&wdExp=TEAMS-TREATMENT&wdhostclicktime=1707819744366&web=1](https://ucnmuni-my.sharepoint.com/:w:/g/personal/456543_muni_cz/EWd-3Gtn5m9MI9GxrN02TcIBMbezxsuLdD9Zvdv35PerAQ?wdOrigin=TEAMS-MAGLEV.p2p_ns.rwc&wdExp=TEAMS-TREATMENT&wdhostclicktime=1707819744366&web=1) komponenta 2

Komponenta 3: Priradovanie priorít bežiacim SJ

- Keďže už nepoužívame priority classes na určenie preemptcie, tak toto stratilo význam ako samostatná komponenta a je to rovnaké ako text
- Všetky SJ majú nižšiu prioritu než iné workloady a teda nejakú priorityclass treba vyrobiť

Komponenta 4: Plugin do plánovaču, ktorý preemptne i non-preemptible SJ

- To je úplne voliteľné, nezahrňovala by som do tejto práce

Prekračuje novovytvorený SJ práh?

Validating webhook.

Pripadalo by mi rozumné vlastne pred tým, než sa vytvorí nový SJ ho validovať cez validating webhook ktorý request buď označí ako Accept alebo Rejected. Accepted ak nepresahuje tu stanovenú hranicu a rejected ak presahuje. A je to vyriešené bez toho, aby operator čosi počítal.