MetaCentrum cloud use-cases & best-practices

Cíl

- vložit do dokumentace několik způsobů použití našeho cloudu které odpovídají best practices tak jak je sami vnímáme
- odkazovat tyto use-cases ve zpracovávaných RT ticketech jako referenční použití
- alespoň jeden z use-cases by měl být dosti detailní a ten by měl jít spustit kýmkoliv

Best practices které chceme ukázat

- i pro velké projekty nepotřebujeme více než jednu veřejnou IP adresu (použití sshuttle)
- jak ukládat velká a malá data (S3 minio, ???)
- jak počítat vědeckou úlohu na jednom či více strojích
    - na jednom například s využitím MPI
    - ~~na více strojích s využitím frameworku na paralelní výpočty (například 0mq nebo fiber)~~ -> PBS
- jak udržovat alokované stroje v cloudu (ansible, ~~puppet~~)
- jak synchronizovat výpočty mezi pracovní stanicí a cloudem (kontejnerový image, openstack-glance image)
- jak stahovat data z výpočtů z cloudu na pracovní stanici (minio client, PBS alternativa)
- jak balancovat požadavky na více strojů v cloudu (OpenStack Octavia)
    - ano, fakulta socialnich studii
- jak oddělit sítě jednotlivých VMek
- DNS pro web-server + monitoring icinga + zalohovani dat
- uloziste (ceph)
    - obecne HDD(SATA) i mensi mnozstvi rychle SSD
    - pro velke iops existuji specialni hosty kde je lokalni SSD disk (bez garance)

Návrhy use-cases

A. Iterativní výpočet π na jednom stroji (MPI)

- sběhnout paralelně (více instancí na jednom stroji) tento příklad https://github.com/thomasWeise/distributedComputingExamples/tree/master/mpi#112-gather-scatter-estimate-pi
- ansible playbook pro update VM stroje a instalací dockeru
- container image s kódem
- data + logy uloženy do S3 minio serveru (každý experiment do separatního bucketu)
- minio klient vytáhne data + logy na pracovní stanici kde je uživatel analyzuje

B. Distribuovaný výpočet frekvence slov z korpusu anglického jazyka

- Data z korpusu https://www.corpusdata.org/iweb_samples.asp
- sběhnout základní notoricky známé počítání frekvence použitých slov v textech anglického korpusu
- architektura 2 VMka (A (master+worker)), B (worker))
- Použít python fiber nebo 0MQ na delegaci počítání frekvence slov, každý worker uloží svůj výsledek do S3,
- minio klient vytáhne data + logy na pracovní stanici kde je uživatel analyzuje
- agregator pak agreguje dohromady na pracovni stanici

PBS>

C. Webová aplikace s vysokou dostupností

- využít load-balancer na několik VMek
- je to use-case který chceme zvýraznit???
- jako use case software lze použít docker-registry nebo hastebin server které lze velice jednoduše škálovat
- pridat authentizace pres Perun | ala app https://projects.cloud.muni.cz

Kompozice use-cases do současné dokumentace

- Přidat Jak správně používat MetaCloud do https://cloud.gitlab-pages.ics.muni.cz/documentation/faq/
- Na stránku Introduction https://cloud.gitlab-pages.ics.muni.cz/documentation/ přidat sumář že cloud používá OpenStack a hlavní komponentami s nimiž uživatel přijde do sstyku jsou:
    - virtualizované stroje
    - floating IP adresy
    - síťování
    - load balancing
- Use-cases přidat mezi Introduction a Get Access tedy téměř na začátek

Ostatní

- přenesení na novou float IP
- migrace legacy aplikace do cloudu (kontejnerizace, automatizace v cloudu)
-