Deploying GitLab on Kubernetes at Stash

Chris Del Pino
Senior DevOps Engineer

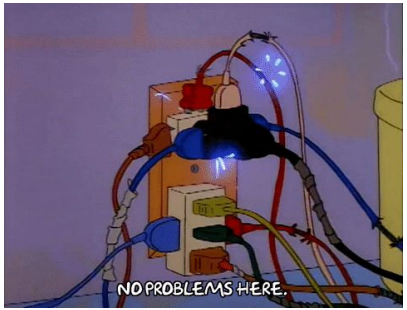
About me

- Senior DevOps engineer
- DevSecOps Team
- Joined Stash in May
- Focus on Kubernetes among other things

DevSecOps Team

How did we get here:

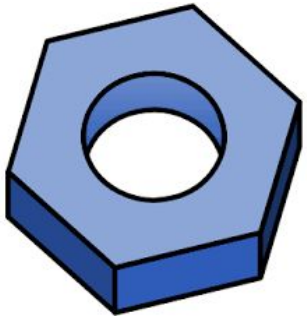
- Currently using CircleCI
  - Not the tool that we expected
    - Easy to setup
    - But difficult to administer
  - Outages
    - Too frequent
    - Affecting our builds
  - Job configuration was an issue
    - Config was confusing to use



The search begins.....

- Not all CI/CD tools are made alike
  - Start searching for one that can help us achieve what we need
- So many tools to choose from.....

Our options...



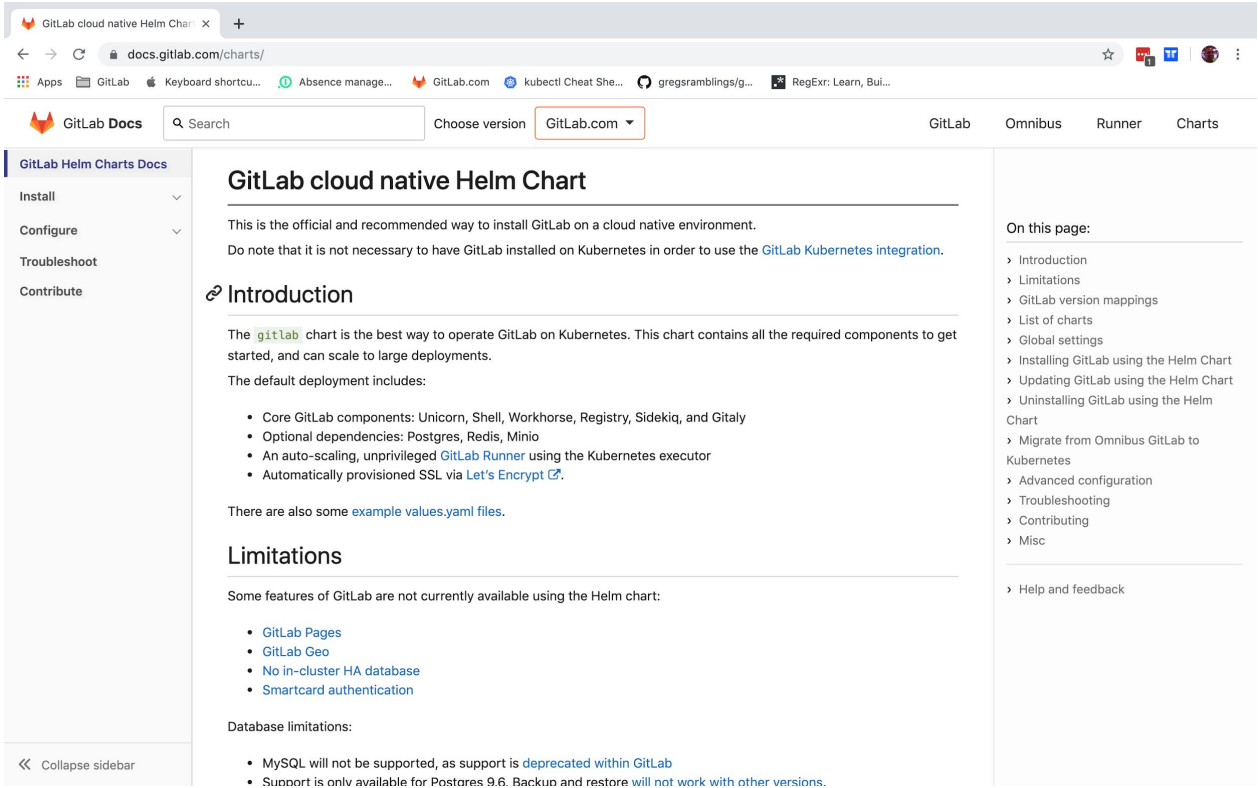
GitLab, GitLab, GitLab.....

- Has the tools and features we need
  - CI/CD pipeline
  - Repositories
  - Registry
  - Issue tracking
  - Wiki
  - Integrations
- Cloud offering as well as on-prem version available for use

But really, why?

- It has what we are looking for:
  - No need to tie different external integrations together
  - Can run on Kubernetes
  - PCI compliant:
    - Our compliance team sleeps better at night

Let’s install



Helm Charts - Components

YAML!!!

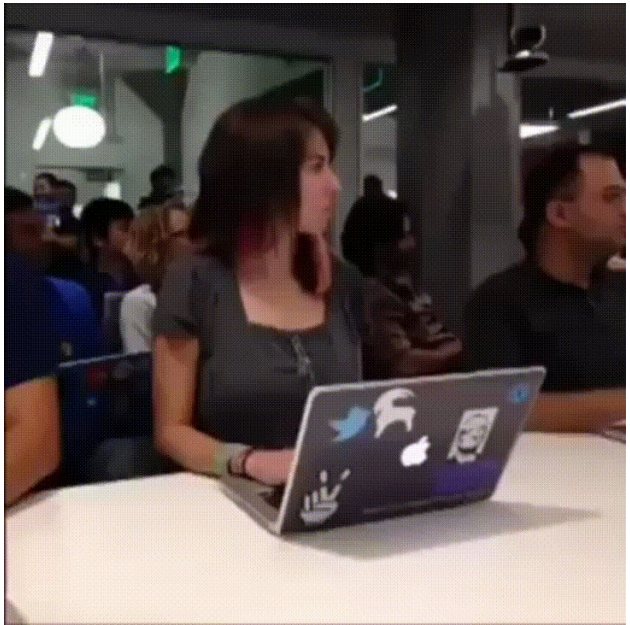
gitlab-deploy.yaml

```yaml
certmanager:
  install: false
gitlab:
  task-runner:
    backups:
      cron:
        enabled: true
        extraArgs: --skip db --skip registry
        schedule: "0 * * * *"
      objectStorage:
        config:
          key: config
          secret: storage-config
gitlab-runner:
  install: false
global:
  ingress:
    configureCertmanager: false
    tls:
      secretName: stshnet-cert
  appConfig:
    artifacts:
      bucket: gitlab-artifacts-storage-stash
      connection:
        key: connection
        secret: gitlab-object-s3-storage
      enabled: true
      proxy_download: true
    backups:
      bucket: gitlab-backup-storage-stash
      tmpBucket: gitlab-tmp-storage-stash
    externalDiffs:
      bucket: gitlab-externaldiffs-storage-stash
      connection:
        key: connection
        secret: gitlab-object-s3-storage
      enabled: null
      proxy_download: true
      when: null
    lfs:
      bucket: gitlab-lfs-storage-stash
      connection:
        key: connection
        secret: gitlab-object-s3-storage
      enabled: true
      proxy_download: true
    omniauth:
```

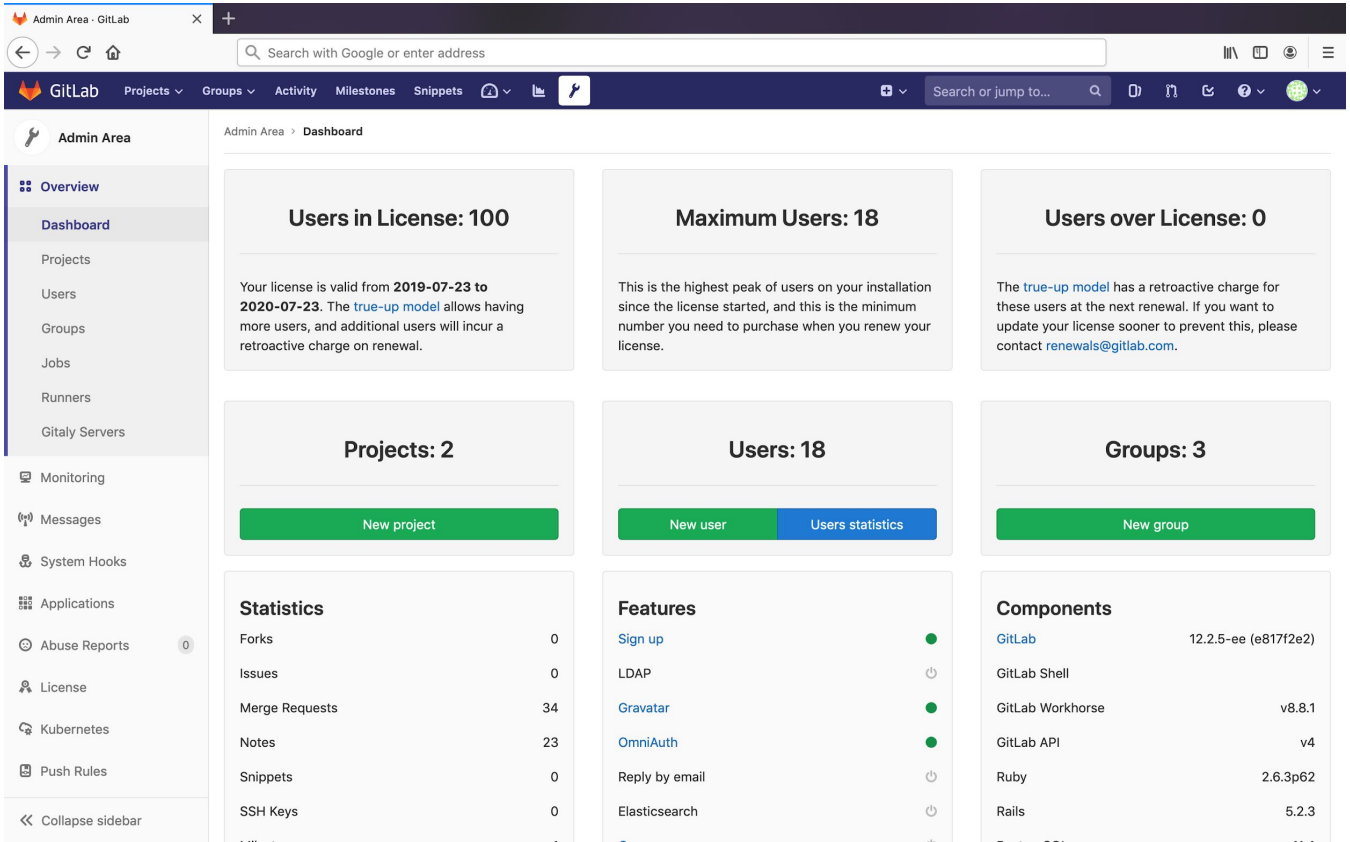
~/gitlab-bootstrap/deployments/gitlab-deploy.yaml 1:1 LF UTF-8 YAML master Fetch GitHub Git (5) 15:32

Deploy....

- We have our config set, let's deploy:
  - `helm upgrade --install gitlab gitlab/gitlab --version=2.1.6 -f gitlab-deploy.yaml`



We have a UI


Admin Area · GitLab
Search with Google or enter address
GitLab
Projects
Groups
Activity
Milestones
Snippets
Search or jump to...
Admin Area
Overview
Dashboard
Projects
Users
Groups
Jobs
Runners
Gitaly Servers
Monitoring
Messages
System Hooks
Applications
Abuse Reports
License
Kubernetes
Push Rules
Collapse sidebar
Admin Area › Dashboard
Users in License: 100
Your license is valid from 2019-07-23 to 2020-07-23. The true-up model allows having more users, and additional users will incur a retroactive charge on renewal.
Maximum Users: 18
This is the highest peak of users on your installation since the license started, and this is the minimum number you need to purchase when you renew your license.
Users over License: 0
The true-up model has a retroactive charge for these users at the next renewal. If you want to update your license sooner to prevent this, please contact renewals@gitlab.com.
Projects: 2
New project
Users: 18
New user
Users statistics
Groups: 3
New group
Statistics
Forks 0
Issues 0
Merge Requests 34
Notes 23
Snippets 0
SSH Keys 0
Features
Sign up
LDAP
Gravatar
OmniAuth
Reply by email
Elasticsearch
Components
GitLab 12.2.5-ee (e817f2e2)
GitLab Shell
GitLab Workhorse v8.8.1
GitLab API v4
Ruby 2.6.3p62
Rails 5.2.3

But we need a runner

- All we did was just deploy the CI
- Now we need a runner
  - Runs jobs
  - Sends results back to CI

More YAML!!!!!

gitlab-runners.yaml

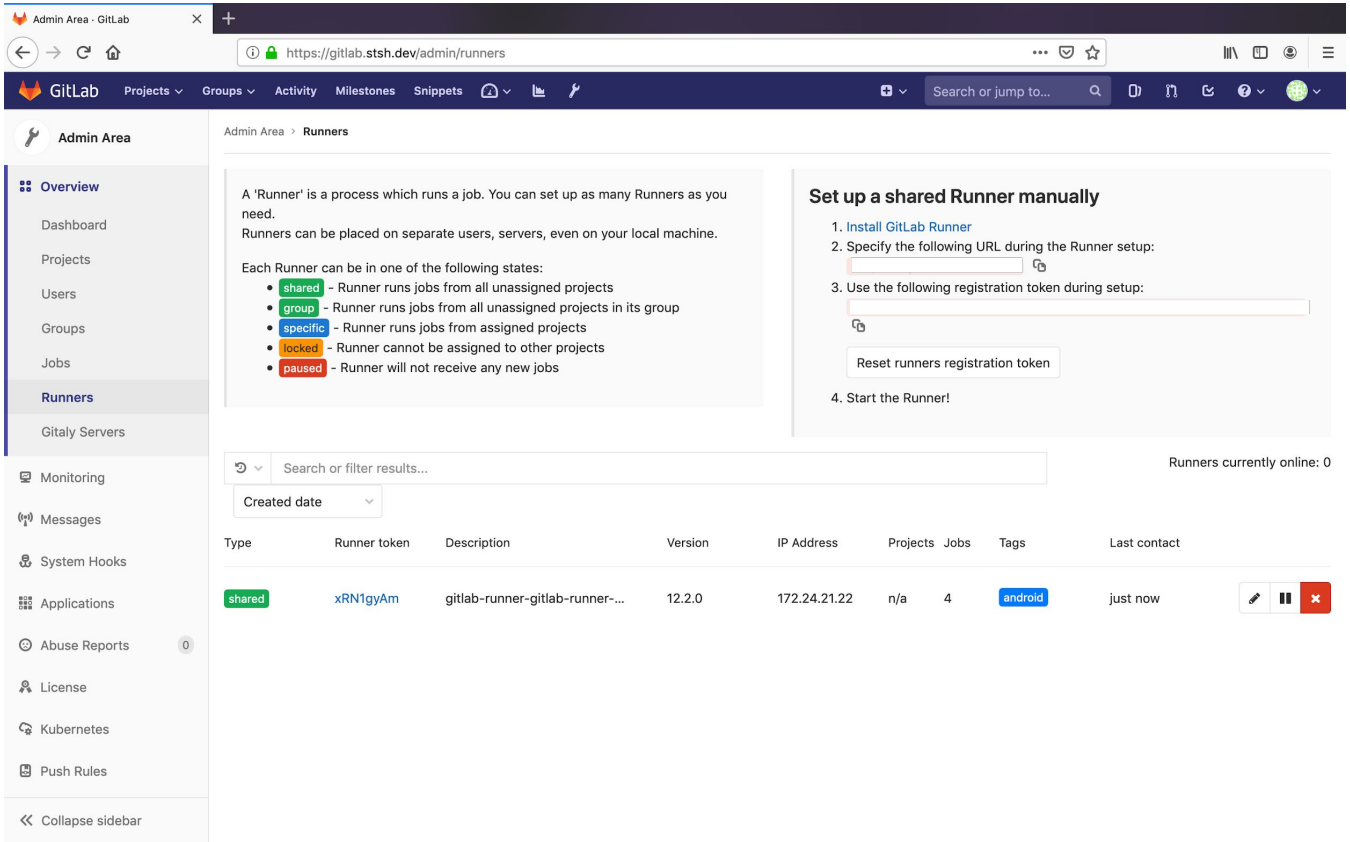
```yaml
## The GitLab Server URL (with protocol) that want to register the runner against
## ref: https://docs.gitlab.com/runner/commands/README.html#gitlab-runner-register
##
gitlabUrl:

## The registration token for adding new Runners to the GitLab server. This must
## be retrieved from your GitLab instance.
## ref: https://docs.gitlab.com/ee/ci/runners/
##
runnerRegistrationToken: ""

## Set the certsSecretName in order to pass custom certificates for GitLab Runner to use
## Provide resource name for a Kubernetes Secret Object in the same namespace,
## this is used to populate the /etc/gitlab-runner/certs directory
## ref: https://docs.gitlab.com/runner/configuration/tls-self-signed.html#supported-options-for-self-signed-certificates
##
#certsSecretName:

## Configure the maximum number of concurrent jobs
## ref: https://docs.gitlab.com/runner/configuration/advanced-configuration.html#the-global-section
##
concurrent: 10

## Defines in seconds how often to check GitLab for a new builds
## ref: https://docs.gitlab.com/runner/configuration/advanced-configuration.html#the-global-section
##
checkInterval: 30

## For RBAC support:
rbac:
  create: false

  ## Run the gitlab-bastion container with the ability to deploy/manage containers of jobs
  ## cluster-wide or only within namespace
  clusterWideAccess: false

  serviceAccountName: gitlab-runner-service-account

  ## If RBAC is disabled in this Helm chart, use the following Kubernetes Service Account name.
  ##

## Configuration for the Pods that the runner launches for each new job
##
runners:
  ## Default container image to use for builds when none is specified
  ##
  #image: docker:stable-dind
```

~/gitlab-bootstrap/deployments/gitlab-runners.yaml* 4:12 LF UTF-8 YAML master Fetch GitHub Git (5) 15:22

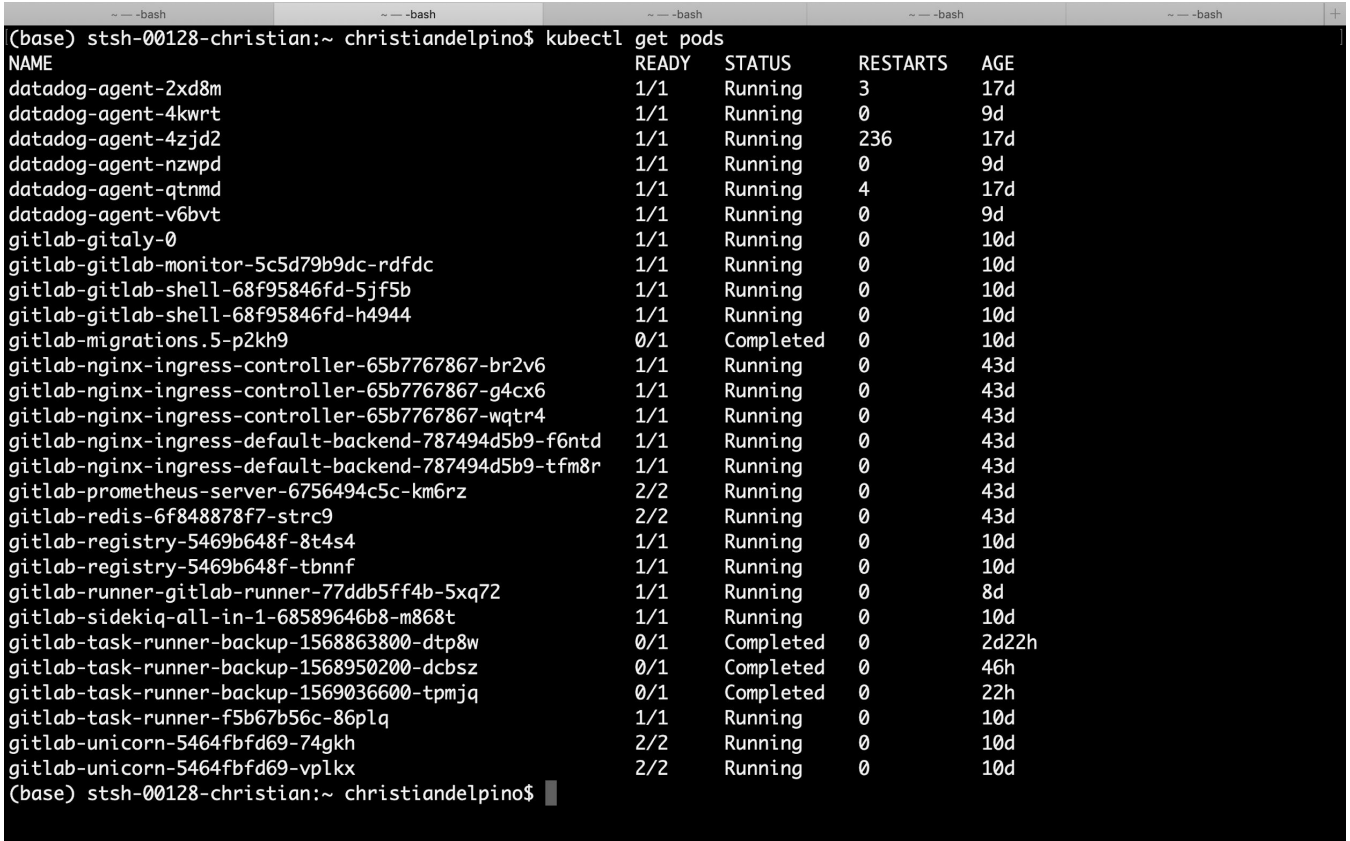
And deploy……

- Runner config is ready:
  - `helm install --name gitlab-runner -f gitlab-runners.yaml gitlab/gitlab-runne`

We have a runner!!!



CLI view


~ — -bash
~ — -bash
~ — -bash
~ — -bash
~ — -bash
(base) stsh-00128-christian:~ christiandelpino$ kubectl get pods
NAME READY STATUS RESTARTS AGE
datadog-agent-2xd8m 1/1 Running 3 17d
datadog-agent-4kwrt 1/1 Running 0 9d
datadog-agent-4zjd2 1/1 Running 236 17d
datadog-agent-nzwpd 1/1 Running 0 9d
datadog-agent-qtnmd 1/1 Running 4 17d
datadog-agent-v6bvt 1/1 Running 0 9d
gitlab-gitaly-0 1/1 Running 0 10d
gitlab-gitlab-monitor-5c5d79b9dc-rdfdc 1/1 Running 0 10d
gitlab-gitlab-shell-68f95846fd-5jf5b 1/1 Running 0 10d
gitlab-gitlab-shell-68f95846fd-h4944 1/1 Running 0 10d
gitlab-migrations.5-p2kh9 0/1 Completed 0 10d
gitlab-nginx-ingress-controller-65b7767867-br2v6 1/1 Running 0 43d
gitlab-nginx-ingress-controller-65b7767867-g4cx6 1/1 Running 0 43d
gitlab-nginx-ingress-controller-65b7767867-wqtr4 1/1 Running 0 43d
gitlab-nginx-ingress-default-backend-787494d5b9-f6ntd 1/1 Running 0 43d
gitlab-nginx-ingress-default-backend-787494d5b9-tfm8r 1/1 Running 0 43d
gitlab-prometheus-server-6756494c5c-km6rz 2/2 Running 0 43d
gitlab-redis-6f848878f7-strc9 2/2 Running 0 43d
gitlab-registry-5469b648f-8t4s4 1/1 Running 0 10d
gitlab-registry-5469b648f-tbnnf 1/1 Running 0 10d
gitlab-runner-gitlab-runner-77ddb5ff4b-5xq72 1/1 Running 0 8d
gitlab-sidekiq-all-in-1-68589646b8-m868t 1/1 Running 0 10d
gitlab-task-runner-backup-1568863800-dtp8w 0/1 Completed 0 2d22h
gitlab-task-runner-backup-1568950200-dcbsz 0/1 Completed 0 46h
gitlab-task-runner-backup-1569036600-tpmjq 0/1 Completed 0 22h
gitlab-task-runner-f5b67b56c-86plq 1/1 Running 0 10d
gitlab-unicorn-5464fbfd69-74gkh 2/2 Running 0 10d
gitlab-unicorn-5464fbfd69-vplkx 2/2 Running 0 10d
(base) stsh-00128-christian:~ christiandelpino$

All done?

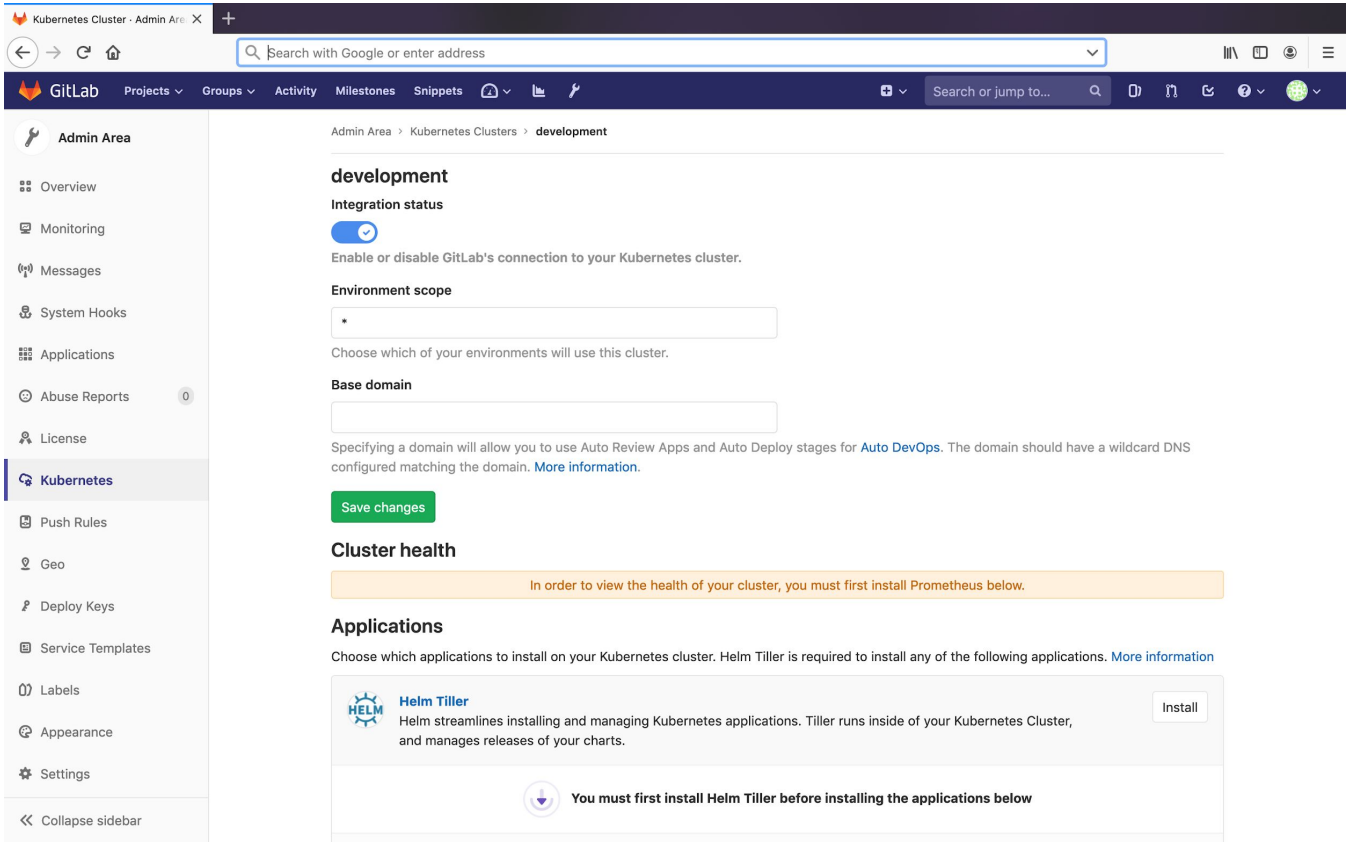
Is On-Prem the real solution?

- A good amount of administration
  - GitLab backups
    - Registry backups
    - Repo backups
    - Etc...
  - RDS
  - Lots of k8s secrets
  - Object store config
  - Etc.....
- Fairly new to k8s
- Do we really want to own all of this?

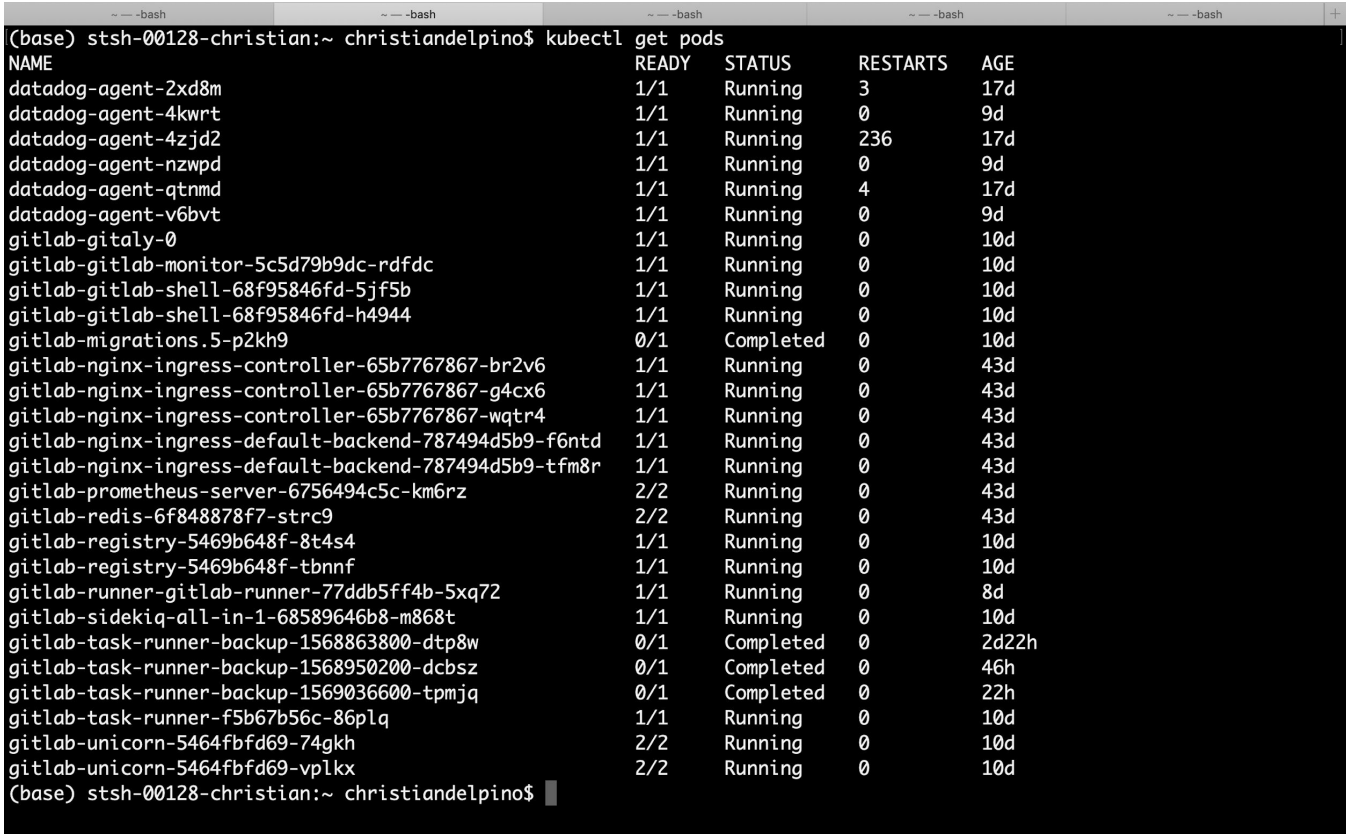
And then there was a hybrid solution

- Using GitLab.com as a managed solution
  - GitLab.com will be the brains of our CI/CD system
  - Kubernetes integration connects to our cluster
    - Allowing us to deploy our workers

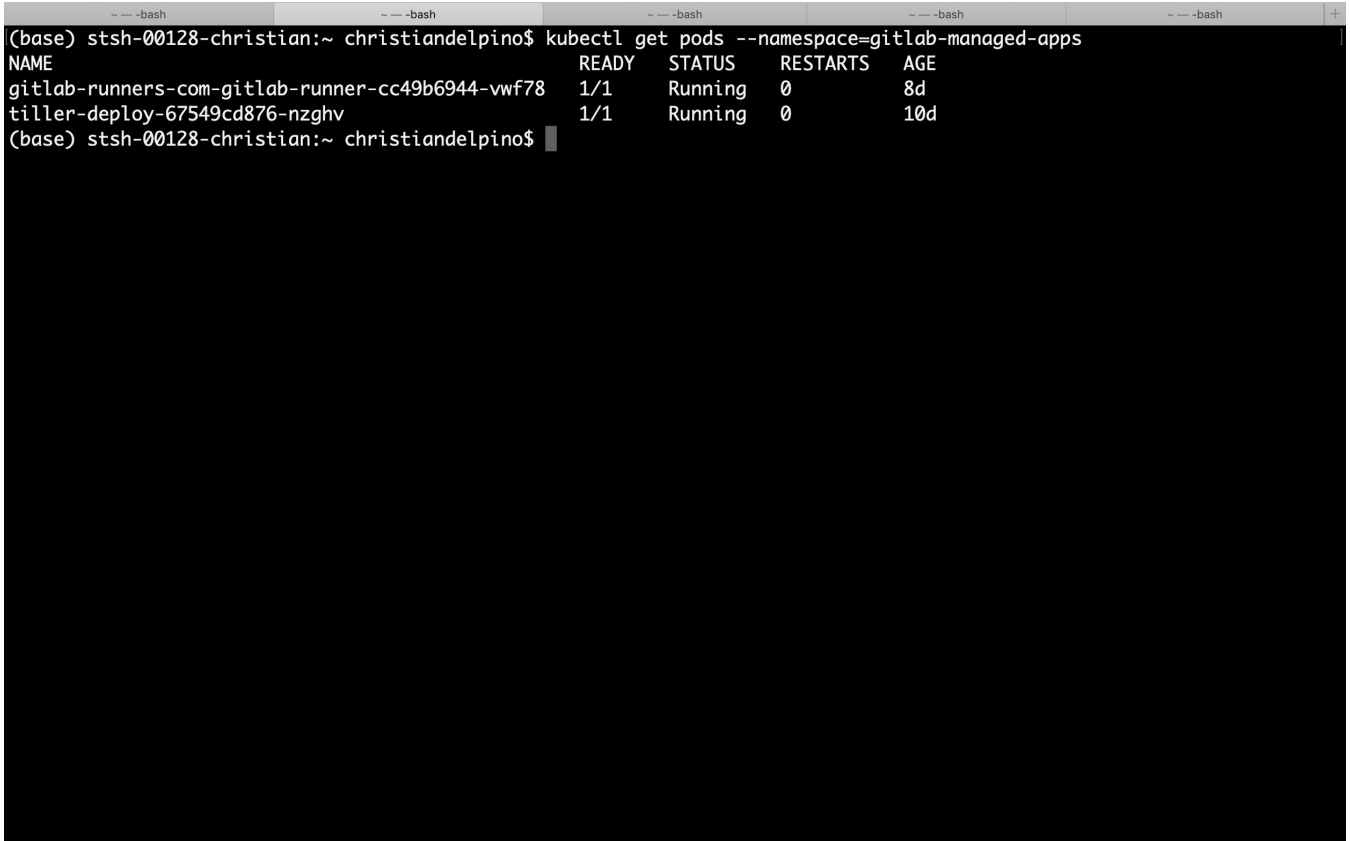
Integration



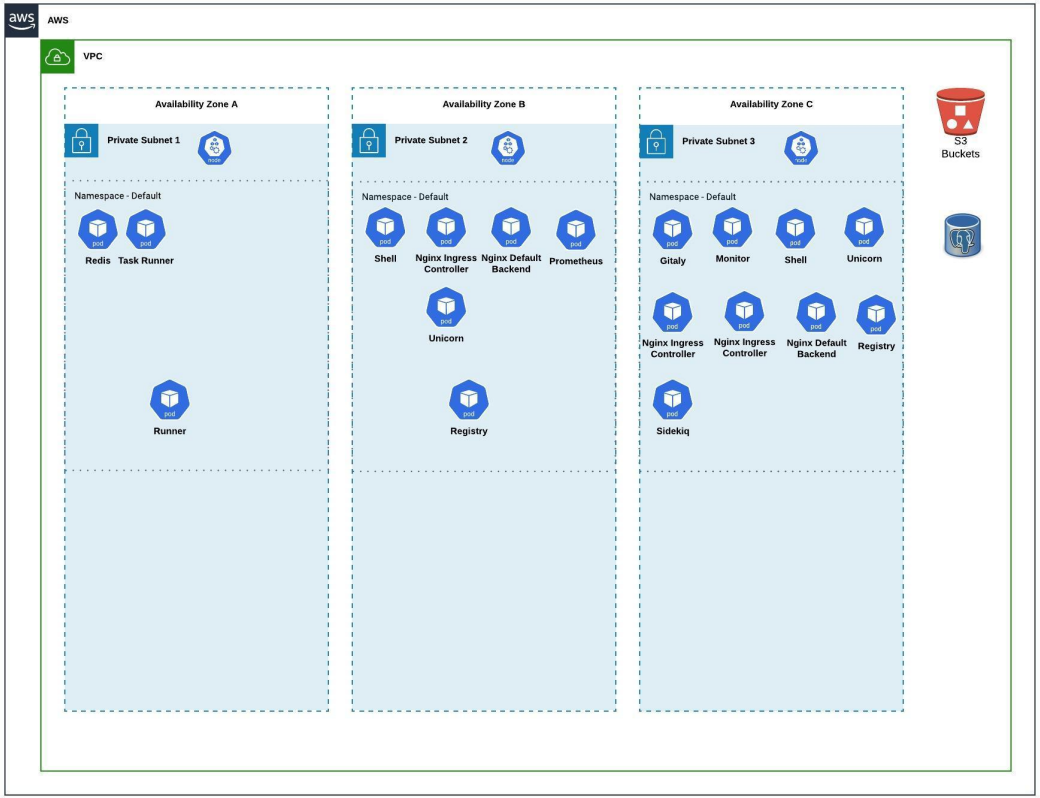
CLI view - On-prem



CLI view - Hybrid



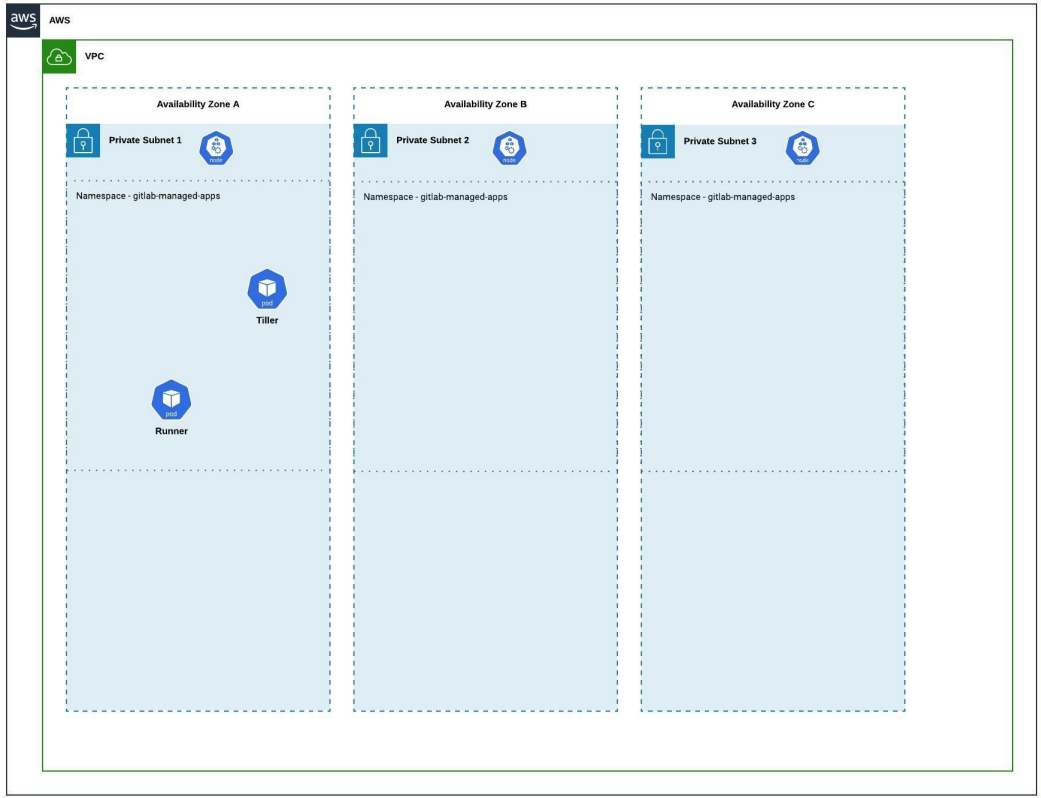
Our GitLab On-Prem Environment



DATADOG

HUMIO

Our GitLab Hybrid Environment



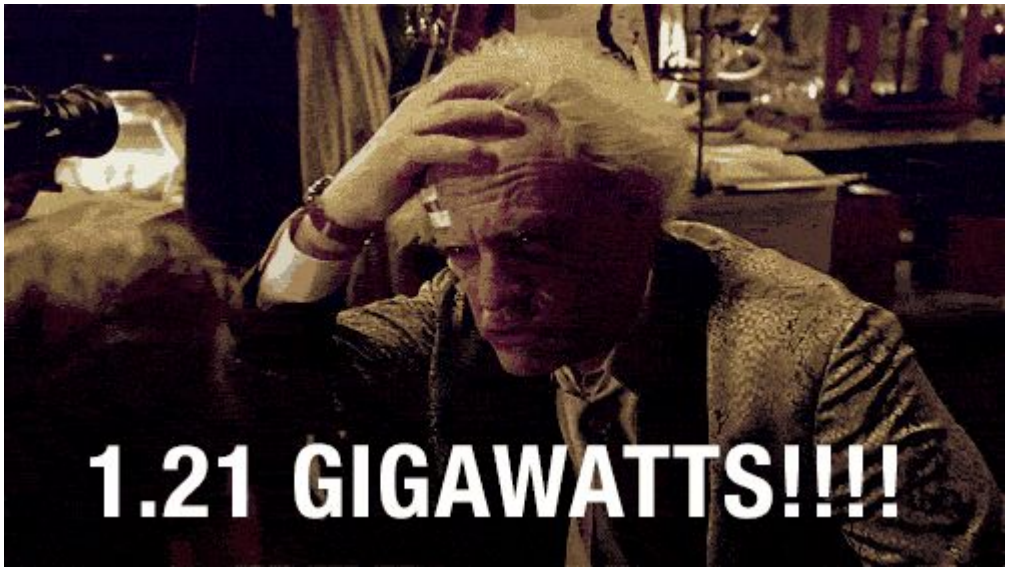
Now are we done?

Analysis still ongoing

- Comparing CCI vs GitLab
  - Testing builds
  - Performance
  - Configuration Automation
  - Disaster Recovery

What’s ahead

- CI Migration
- Deployment tools for k8s
- Further integrations
- Proof of concept on other products



Thank you!

Chris Del Pino

Stash

stashinvest.com

cdelpino@stashinvest.com

@delpic

GitLab Helm Documentation:

https://docs.gitlab.com/charts/