

Deploying GitLab on Kubernetes at Stash

Chris Del Pino
Senior DevOps Engineer

About me

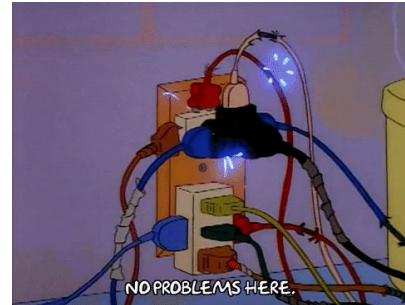
- Senior DevOps engineer
- DevSecOps Team
- Joined Stash in May
- Focus on Kubernetes among other things

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



S |

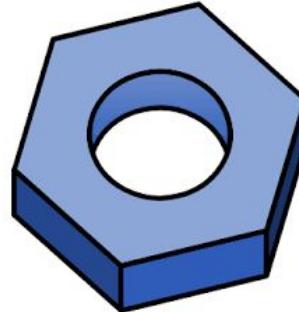
The search begins.....

- 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.....



S |

Our options...



S |

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

The screenshot shows a web browser displaying the official GitLab Helm Chart documentation at docs.gitlab.com/charts/. The page title is "GitLab cloud native Helm Chart". The left sidebar contains navigation links for "Install", "Configure", "Troubleshoot", and "Contribute". The main content area starts with an "Introduction" section, which describes the chart as the best way to operate GitLab on Kubernetes. It lists the default deployment components: Unicorn, Shell, Workhorse, Registry, Sidekiq, and Gitaly, along with optional dependencies like Postgres, Redis, Minio, auto-scaling, and Let's Encrypt SSL provisioning. Below this is a "Limitations" section, which notes that some features like GitLab Pages and Geo are not available. A "Database limitations" section points out MySQL support issues. The right sidebar includes a "On this page:" table of contents with links to various sections like Introduction, Limitations, and Help and feedback.

GitLab cloud native Helm Chart

This is the official and recommended way to install GitLab on a cloud native environment.

Do note that it is not necessary to have GitLab installed on Kubernetes in order to use the [GitLab Kubernetes integration](#).

Introduction

The `gitlab` chart is the best way to operate GitLab on Kubernetes. This chart contains all the required components to get started, and can scale to large deployments.

The default deployment includes:

- Core GitLab components: Unicorn, Shell, Workhorse, Registry, Sidekiq, and Gitaly
- Optional dependencies: Postgres, Redis, Minio
- An auto-scaling, unprivileged [GitLab Runner](#) using the Kubernetes executor
- Automatically provisioned SSL via [Let's Encrypt](#).

There are also some [example values.yaml](#) files.

Limitations

Some features of GitLab are not currently available using the Helm chart:

- [GitLab Pages](#)
- [GitLab Geo](#)
- [No in-cluster HA database](#)
- [Smartcard authentication](#)

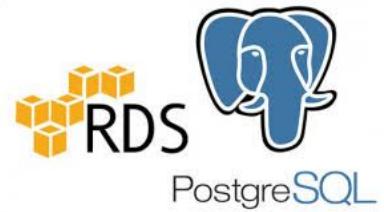
Database limitations:

- MySQL will not be supported, as support is [deprecated within GitLab](#)
- Support is only available for Postgres 9.6. [Backup and restore](#) will not work with other versions.

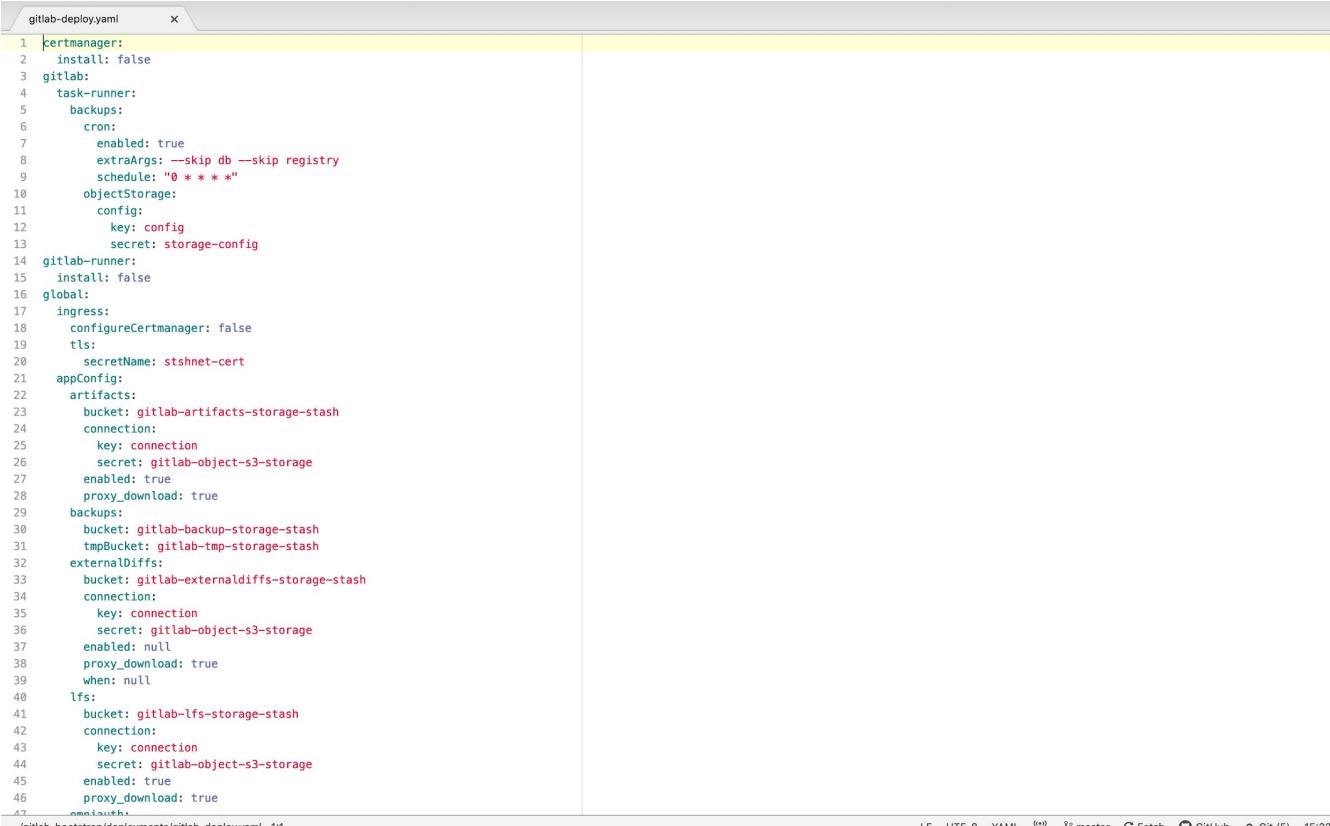
[Help and feedback](#)

S |

Helm Charts - Components



YAML!!!



```
gitlab-deploy.yaml
1 certmanager:
2   install: false
3 gitlab:
4   task-runner:
5     backups:
6       cron:
7         enabled: true
8         extraArgs: --skip db --skip registry
9         schedule: "0 * * * *"
10    objectStorage:
11      config:
12        key: config
13        secret: storage-config
14 gitlab-runner:
15   install: false
16 global:
17   ingress:
18     configureCertmanager: false
19     tls:
20       secretName: stshnet-cert
21 appConfig:
22   artifacts:
23     bucket: gitlab-artifacts-storage-stash
24     connection:
25       key: connection
26       secret: gitlab-object-s3-storage
27     enabled: true
28     proxy_download: true
29   backups:
30     bucket: gitlab-backup-storage-stash
31     tmpBucket: gitlab-tmp-storage-stash
32   externalDiffs:
33     bucket: gitlab-externaldiffs-storage-stash
34     connection:
35       key: connection
36       secret: gitlab-object-s3-storage
37     enabled: null
38     proxy_download: true
39     when: null
40   lfs:
41     bucket: gitlab-lfs-storage-stash
42     connection:
43       key: connection
44       secret: gitlab-object-s3-storage
45     enabled: true
46     proxy_download: true
47   omnibus:
```

~/.gitlab-bootstrap/deployments/gitlab-deploy.yaml 1:1

LF UTF-8 YAML master ⌂ Fetch ⌂ GitHub ⌂ Git (5) 15:32

S |

Deploy....

- We have our config set, let's deploy:

- `helm upgrade --install gitlab gitlab/gitlab --version=2.1.6 -f gitlab-deploy.yaml`



S

We have a UI

The screenshot shows the GitLab Admin Area dashboard. The left sidebar contains navigation links for Admin Area, Overview, Dashboard, Projects, Users, Groups, Jobs, Runners, GitLab Servers, Monitoring, Messages, System Hooks, Applications, Abuse Reports (with 0 notifications), License, Kubernetes, Push Rules, and a Collapse sidebar button.

The main dashboard features several cards:

- 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**: Includes a "New project" button.
- Users: 18**: Includes "New user" and "Users statistics" buttons.
- Groups: 3**: Includes a "New group" button.
- Statistics**: Shows Forks (0), Issues (0), Merge Requests (34), Notes (23), Snippets (0), and SSH Keys (0).
- Features**: Shows Sign up (green dot), LDAP (grey dot), Gravatar (green dot), OmniAuth (green dot), Reply by email (grey dot), and Elasticsearch (grey dot).
- Components**: Shows GitLab (12.2.5-ee (e817f2e2)), GitLab Shell, GitLab Workhorse (v8.8.1), GitLab API (v4), Ruby (2.6.3p62), and Rails (5.2.3).

S |

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
1 ## The GitLab Server URL (with protocol) that want to register the runner against
2 ## ref: https://docs.gitlab.com/runner/commands/README.html#gitlab-runner-register
3 ##
4 gitlabUrl:
5
6 ## The registration token for adding new Runners to the GitLab server. This must
7 ## be retrieved from your GitLab instance.
8 ## ref: https://docs.gitlab.com/ee/ci/runners/
9 ##
10 runnerRegistrationToken: ""
11
12 ## Set the certsSecretName in order to pass custom certificates for GitLab Runner to use
13 ## Provide resource name for a Kubernetes Secret Object in the same namespace,
14 ## this is used to populate the /etc/gitlab-runner/certs directory
15 ## ref: https://docs.gitlab.com/runner/configuration/tls-self-signed.html#supported-options-for-self-signed-certificates
16 ##
17 #certsSecretName:
18
19 ## Configure the maximum number of concurrent jobs
20 ## ref: https://docs.gitlab.com/runner/configuration/advanced-configuration.html#the-global-section
21 ##
22 concurrent: 10
23
24 ## Defines how often to check GitLab for a new builds
25 ## ref: https://docs.gitlab.com/runner/configuration/advanced-configuration.html#the-global-section
26 ##
27 checkInterval: 30
28
29 ## For RBAC support:
30 rbac:
31   create: false
32
33 ## Run the gitlab-bastion container with the ability to deploy/manage containers of jobs
34 ## cluster-wide or only within namespace
35 clusterWideAccess: false
36
37 serviceAccountName: gitlab-runner-service-account
38
39 ## If RBAC is disabled in this Helm chart, use the following Kubernetes Service Account name.
40 ##
41
42 ## Configuration for the Pods that the runner launches for each new job
43 ##
44 runners:
45   ## Default container image to use for builds when none is specified
46   ##
47   image: docker:stable-dind
```

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

S |

And deploy.....

- Runner config is ready:

```
- helm install --name gitlab-runner -f gitlab-runners.yaml gitlab/gitlab-runne
```



S

We have a runner!!!

The screenshot shows the GitLab Admin Area interface. The left sidebar has a 'Runners' section selected under 'Admin Area'. The main content area displays information about runners, a manual setup guide, and a table of currently online runners.

Set up a shared Runner manually

1. Install GitLab Runner
2. Specify the following URL during the Runner setup:
3. Use the following registration token during setup:

Reset runners registration token
4. Start the Runner!

Type	Runner token	Description	Version	IP Address	Projects	Jobs	Tags	Last contact
shared	xRN1gyAm	gitlab-runner-gitlab-runner...	12.2.0	172.24.21.22	n/a	4	android	just now

CLI view

```
~ — -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-tbnrf 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-tpmqj 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$
```

S |

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

The screenshot shows the GitLab Admin Area interface, specifically the 'Kubernetes Cluster - Admin Area' section. The left sidebar has a 'Kubernetes' section selected. The main content area shows the 'development' environment configuration for a Kubernetes cluster.

development

Integration status

Enable or disable GitLab's connection to your Kubernetes cluster.

Environment scope

Choose which of your environments will use this cluster.

Base domain

Specifying a domain will allow you to use Auto Review Apps and Auto Deploy stages for [Auto DevOps](#). The domain should have a wildcard DNS configured matching the domain. [More information](#).

Save changes

Cluster health

In order to view the health of your cluster, you must first install Prometheus below.

Applications

Choose which applications to install on your Kubernetes cluster. Helm Tiller is required to install any of the following applications. [More information](#)

Helm Tiller

Helm streamlines installing and managing Kubernetes applications. Tiller runs inside of your Kubernetes Cluster, and manages releases of your charts.

Install

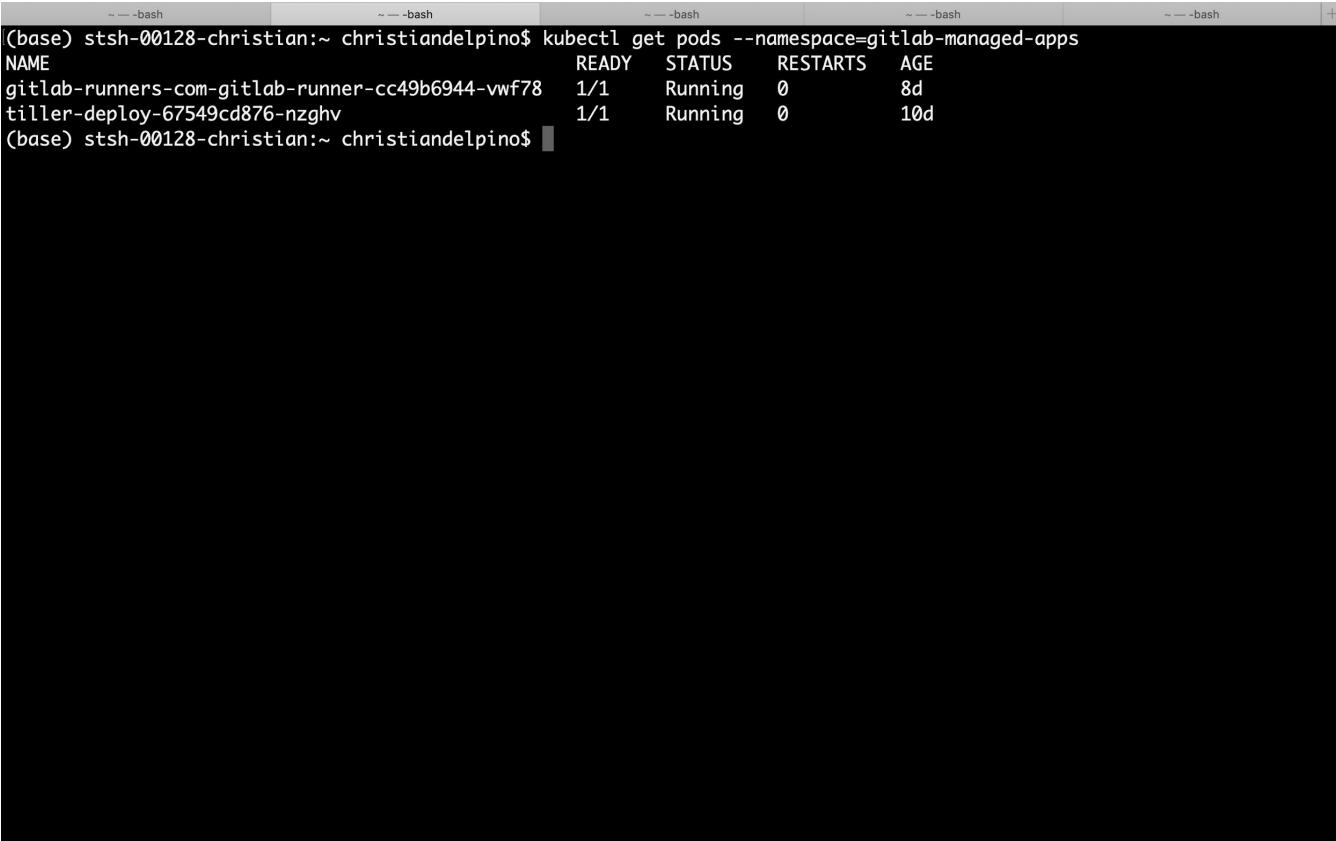
You must first install Helm Tiller before installing the applications below

CLI view - On-prem

```
~ --- -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-tbnff     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$
```

S |

CLI view - Hybrid

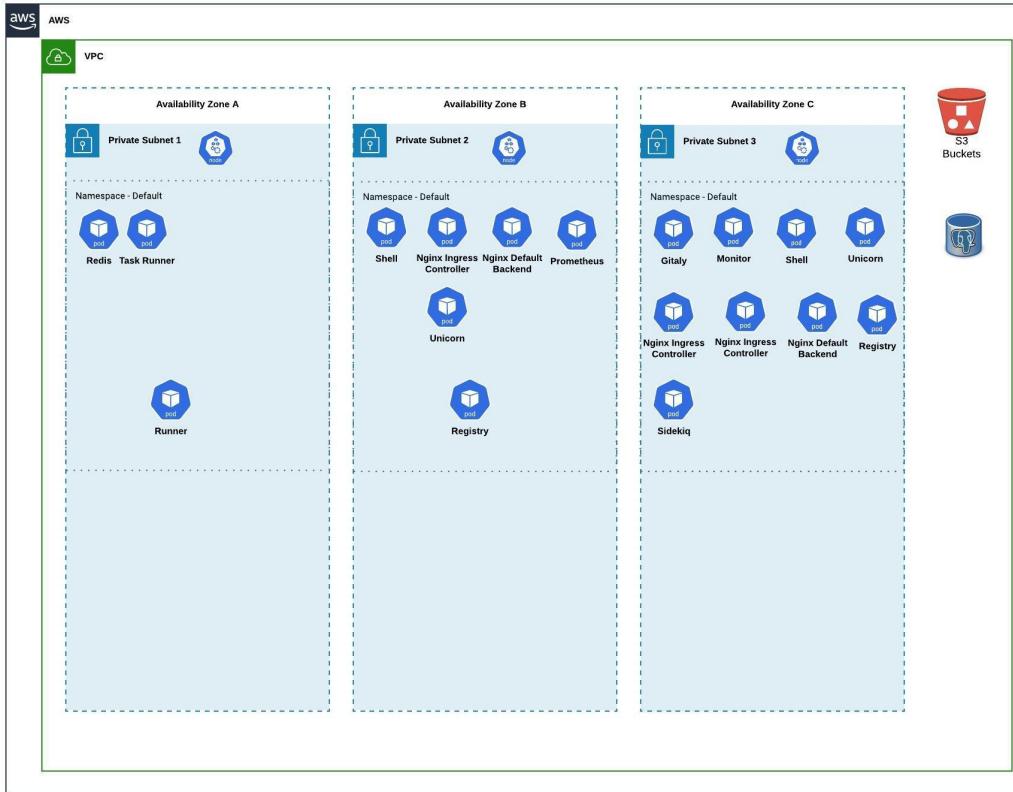


The screenshot shows a terminal window with five tabs, each labeled with a bash prompt (~ — -bash). The active tab displays the command `kubectl get pods --namespace=gitlab-managed-apps`. The output lists two pods:

NAME	READY	STATUS	RESTARTS	AGE
gitlab-runners-com-gitlab-runner-cc49b6944-vwf78	1/1	Running	0	8d
tiller-deploy-67549cd876-nzghv	1/1	Running	0	10d

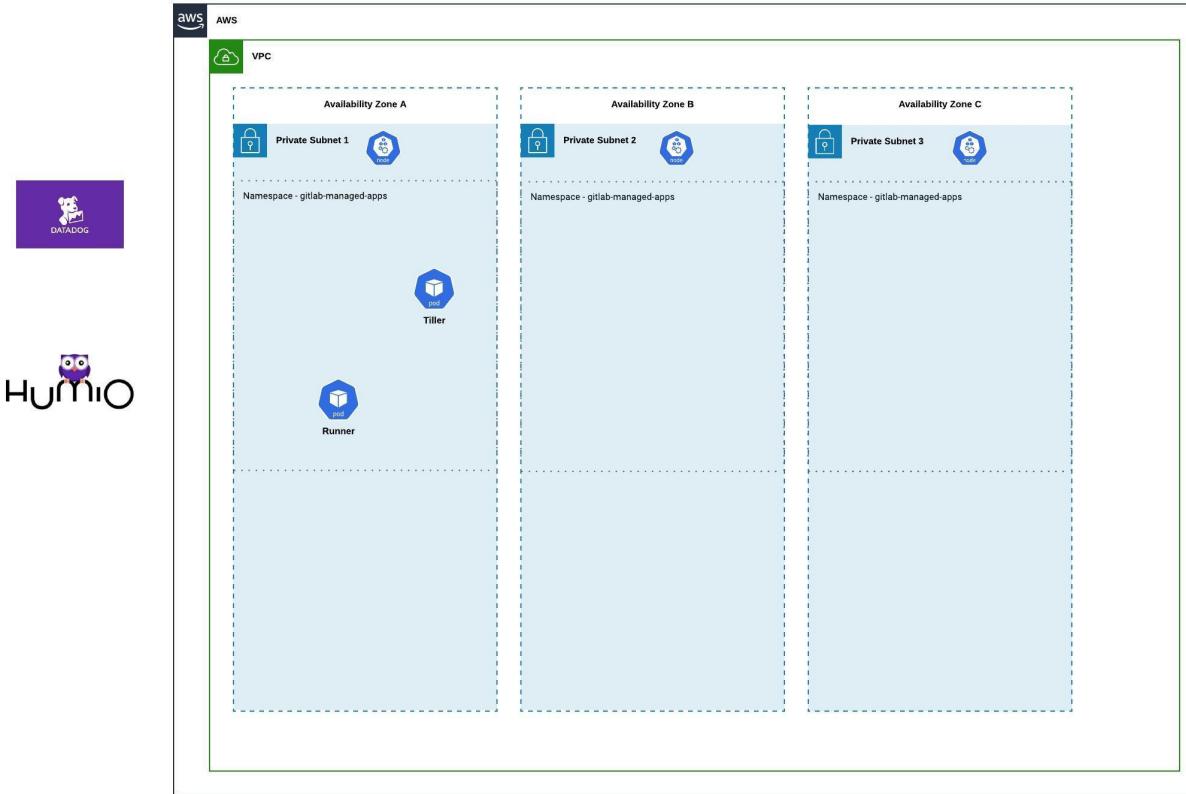
Below the table, the command `(base) stsh-00128-christian:~ christiandelpino$` is shown.

Our GitLab On-Prem Environment



S

Our GitLab Hybrid Environment



S |

Now are we done?



S |

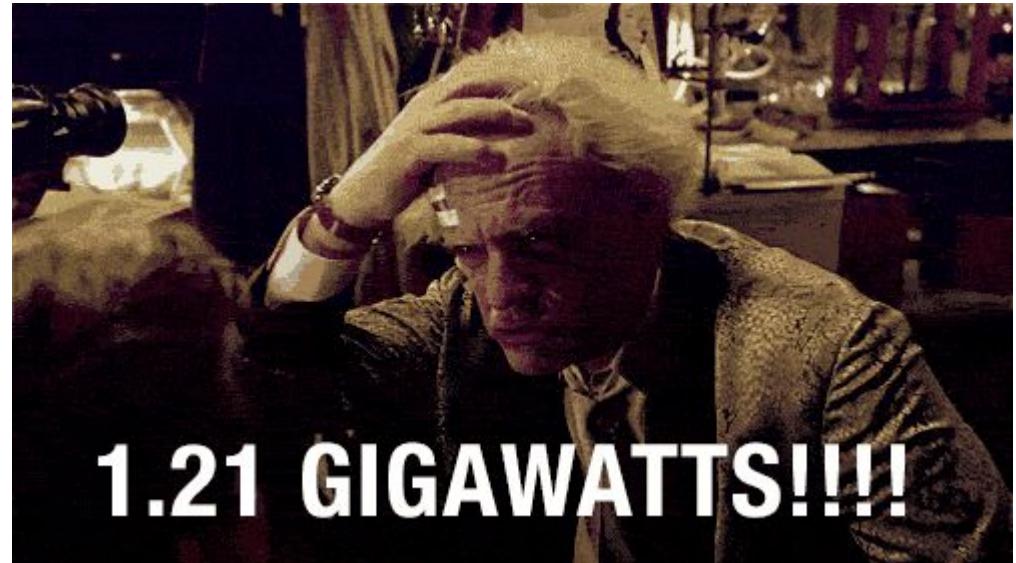
Analysis still ongoing

- 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



S |

Thank you!

Chris Del Pino

Stash

stashinvest.com

cdelpino@stashinvest.com

@delpic

GitLab Helm Documentation:

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

