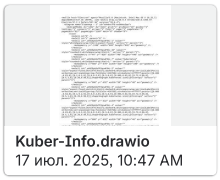
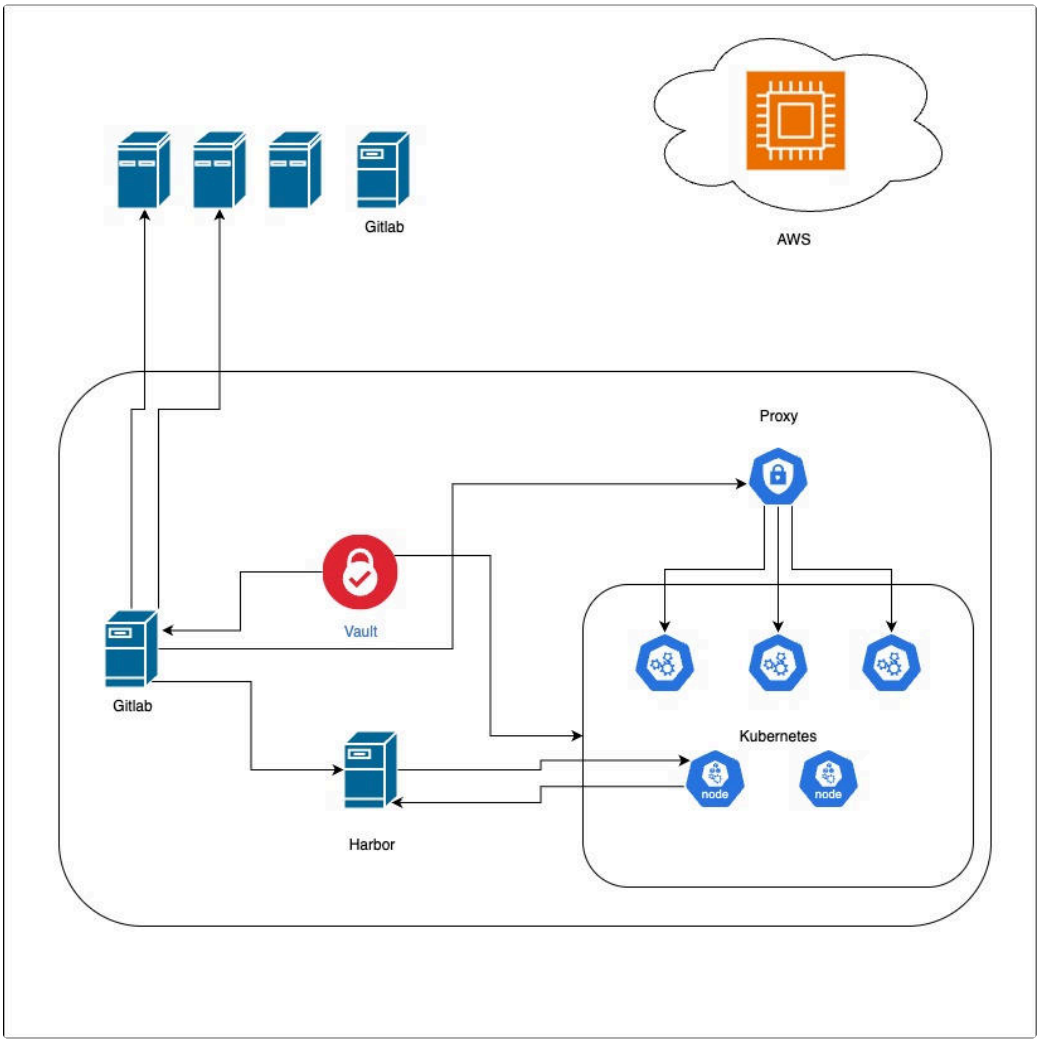
DevOps инфраструктура



Kuber-Info.drawio
17 июл. 2025, 10:47 AM



Необходимые ресурсы для кластера Kubernetes

Имя	количество	CPU	RAM	SDD
KubeControl	3	2	4	40
KubeWorker	2	4	16	256
KubeProxy	1	2	2	20

> примечание!
> в качестве прокси можно использовать инстанс в AWS, главное что бы у него был доступ к сети и машинам на которых развернут KubeControl

Kube Worker необязательно должен находиться в той же сети что и KubeControl, но они должны иметь доступ к друг другу.

Name	Count	CPU	RAM	SDD
Gitlab	1	2	4	512
Nexus3/Harbor	1	4	8	256
HashicorpVault	1	2	4	60

Gitlab предлагается вынести на отдельный сервер, он должен иметь доступ к KubeProxy (прим. в данном контексте имеется ввиду HAProxy сервер, а не компонент kubernetes) и к Vault, так же желательно обеспечить доступ от gitlab-server к Harbor.

Gitlab-Runner предполагается использовать внутри KubeWorker как сервис.

> !Примечание!
> Если KubeWorker будет построен на ARM архитектуре, то для деплоя на стенды с x86 архитектурой понадобиться отдельный инстенс KubeWorker на x86 и отдельный Gitlab-Runner.

Рекомендации по улучшению DevOps процессов и инфраструктуры

1. **Установить и настроить Vault в HA-Mode (не обязательно в HA)**
    a. провести тест на резервное копирование и восстановление
2. **Перенести секреты из Gitlab в Vault**
3. **Настроить кластер Kubernetes**

a. развернуть кластер через kubespray
b. настроить доступ к кластеру через сертификат
c. настроить права и группы в RBAC

4. **Перенести в Kubernetes Gitlab-Runner**
   a. подготовить helm чарт с раннером и развернуть его
5. **Описать инфраструктуру AWS в Terraform (Возможно CloudFormation подойдет больше)**
   a. определить наиболее подходящий инструмент (terraform+terragrunt VS CloudFormation)
   b. в стучае с Teaaraform настроить dynamoDB (необязательно),
   c. настроить S3 для хранения файла с состоянием Terraform.
   d. описать сети, сервисы и другие инстансы в terraform
6. **Рассмотреть возможность использования Nexus для репозитория (возможно нет необходимости)**
   a. Nexus нужен только как Docker-registry, рекомендуется использовать Harbor.
7. **Нужно вынести Gitlab на On-premise, но надо учесть возможность не только резервного копирования, но и обновления, сделать эти процессы максимально простыми, сам процесс установки, настройки и резервного копирования должен быть описан в Ansible роли.**
   a. описать роль в Ansible для развертования на хосте Gitlab-Server
   b. Протестировать обновление Gitlab-server
   c. Протестировать Backup/Restore сервера Gitlab
8. **Перенести существующие таски в роли, переработать структуру, добавить Values, Group Vars, Templates.**
   a. Привести в порядок текущую структуру в Ansible,
      i. Таски переложить в роли
      ii. переписать таски с Shell на модули
      iii. по возможности протестировать, найти и устранить возможные ошибки.

Сделать работу с Ansible более гибкой.

9. **Часть инфраструктуры на ARM архитектуре, инфраструктура в облаке на x86, могут возникать конфликты, нужно учесть этот момент и внести необходимые изменения в темплейты.**
10. **Настроить BackUP On-Premise инфраструктуры.**

а. Выбрать инструмент для резеврного копирования развернутых сервисов.