Финальный проект.
Группа DOS12-onl

«Jenkins CI\CD в Yandex Cloud в ‘полностью автоматическом’ режиме»

Можно ли автоматизировать все?

...I show you how deep the rabbit hole goes.

Что я хотел:

- Инфраструктуру для разарботки (dev и prod);
- Рабочие pipelines без вмешательства пользователя(меня);
- Тестирование почти всего, что можно протестировать;
- Хранение артефактов разработки/сборки;
- Мониторинг;
- Сообщения от инфраструктуры (ошибки, алерты и т.д.);
- Шифрование «секретов»;
- Создать, по возможности, универсальные модули (скрипты) которые можно использовать в других проектах;
- Попробовать в проекте все, что мы проходили на курсе(по возможности).

Что я не хотел:

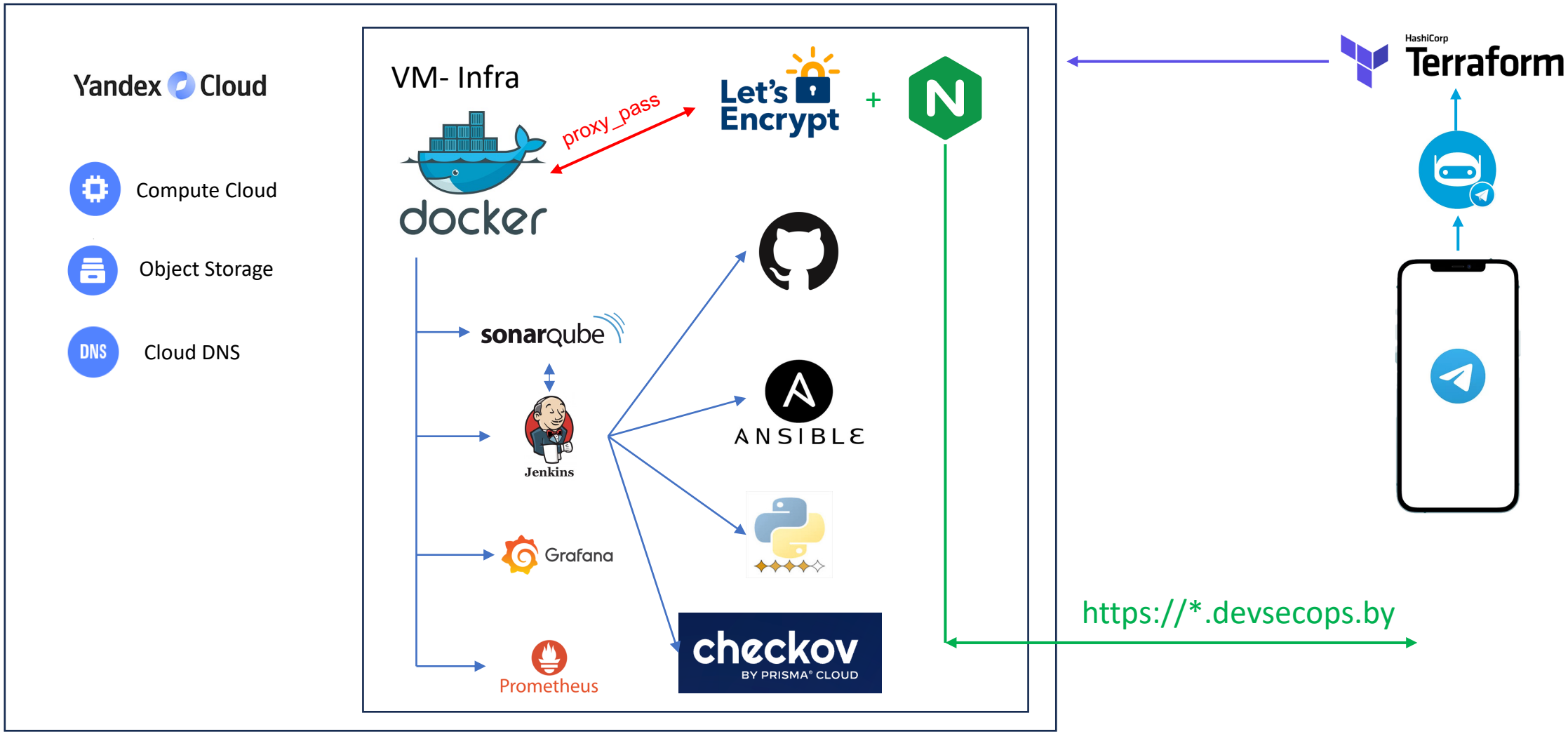
- «ковыряться» в коде и инфраструктуре, после развертывания, чтобы что-то донастроить.

P.S. почти получилось

Архитектура (используемые программы/технологии/яп и т.д.):

- Docker (Docker-compose);
- Jenkins(IAC);
- SonarQube;
- Python (bot);
- Groovy (pipeline);
- Bash;
- Nginx + SSL (certbot);
- Prometheus;
- Loki;
- Grafana;
- caDvisor;
- Ansible;
- Pylint;
- Apache Benchmark;
- Telegram (notification);
- Checkov;
- Trivy;
- GPG;

Архитектура (как это выглядит):



Универсальные скрипты: telegram.sh, gpg_secret.sh

```
./telegram.sh 'hello word' #отправка сообщения
./telegram.sh /path/to/file #отправка файла
```

```
#Создание ключей
./gpg_secret.sh --create (Real name = ID ключа)
./gpg_secret.sh --export
#после команды вводим ID ключа, появятся в каталоге со скриптом 2 #файла: ID-private.key и ID-public.key
#Шифрование:
./gpg_secret.sh --enc --k name_public.key /path/to/file
#Дешифрование:
./gpg_secret.sh --dec --k name_private.key /path/to/file
#Рекурсивная дешифровка
./gpg_secret.sh --rcs --k name_private.key /path/to/folder
```

Что можно добавить/улучшить:

- gpg_secret.sh – переписать на go, добавить отдельную опции кодирования строк по маске;
- Jenkins – добавить плэйбук k8s, использовать для сборки образов kaniko (что сделает деплой унивнерсальным);
- Добавить sec тестирование кода IaC;
- Post-deploy сканирование с помощью ZAP;
- Переделать remote-exec в «один» скрипт пригодный и для локальной установки;
- Выводы сканеров переделать в json и парсить с помощью дашборда DefectDojo;
- Output_deploy = completed_process.stdout.strip() – сделать парсинг полезной информации во время развертывания и вывести в телеграм бот;
- Добавить в механизм пайплайна инструменты(команды и описание) для сборки проектов на java, go и т.д;
- Добавить отдельные пайплайны для dev и prod для отката на передыдущую версию. ‘TAG’ – 1;
- Терраформ модуль для DNS;
- Сделать хранение Jenkins пайплайнов в проекте в jenkinsfile;
- По другом реализовать генерацию конфигов для NGINX;
- Заменить Ansible на jenkins агенты.