

CI/CD в Kubernetes

План

- Helm
- Ci/CD в GitLab
- Стратегии деплоя в Kubernetes

The background is a light gray space-themed pattern. It features numerous small, faint stars scattered across the canvas. Several stylized planets with rings, resembling Saturn, are positioned at various points: one in the top right, one in the bottom left, and one near the bottom center. There are also several thin, diagonal lines representing comets or meteor streaks. The overall aesthetic is clean and modern, typical of a tech or open-source project branding.

Helm

Проблемы?

Проблемы?

- Устали писать манифесты вручную (они большие и их много)

Проблемы?

- Устали писать манифесты вручную (они большие и их много)
- Версии манифестов не привязаны к версиям приложений

Проблемы?

- Устали писать манифесты вручную (они большие и их много)
- Версии манифестов не привязаны к версиям приложений
- Нет шаблонов манифестов (не понимают переменных окружений и т.д.)

Проблемы?

- Устали писать манифесты вручную (они большие и их много)
- Версии манифестов не привязаны к версиям приложений
- Нет шаблонов манифестов (не понимают переменных окружений и т.д.)
- Что если надо сделать Rollback нескольких манифестов

Проблемы?

- Устали писать манифесты вручную (они большие и их много)
- Версии манифестов не привязаны к версиям приложений
- Нет шаблонов манифестов (не понимают переменных окружений и т.д.)
- Что если надо сделать Rollback нескольких манифестов
- Как поделиться пакетом манифестов?

Helm



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)
Сейчас это также проект Kubernetes.



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)

Сейчас это также проект Kubernetes.

Что умеет Helm?



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)

Сейчас это также проект Kubernetes.

Что умеет Helm?



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)

Сейчас это также проект Kubernetes.

Что умеет Helm?

- упаковывать несколько ресурсов k8s в один пакет - **Chart**



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)

Сейчас это также проект Kubernetes.

Что умеет Helm?

- упаковывать несколько ресурсов k8s в один пакет - **Chart**
- шаблонизировать установку



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)

Сейчас это также проект Kubernetes.

Что умеет Helm?

- упаковывать несколько ресурсов k8s в один пакет - **Chart**
- шаблонизировать установку
- устанавливать **Chart'ы** - делать **Release**



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)

Сейчас это также проект Kubernetes.

Что умеет Helm?

- упаковывать несколько ресурсов k8s в один пакет - **Chart**
- шаблонизировать установку
- устанавливать **Chart'ы** - делать **Release**
- делать **Upgrade** (обновления) и **Rollback** (откаты) выкатываемых приложений



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)

Сейчас это также проект Kubernetes.

Что умеет Helm?

- упаковывать несколько ресурсов k8s в один пакет - **Chart**
- шаблонизировать установку
- устанавливать **Chart'ы** - делать **Release**
- делать **Upgrade** (обновления) и **Rollback** (откаты) выкатываемых приложений
- управлять зависимостями между пакетами



Helm - пакетный менеджер для Kubernetes, разработанный ребятами из Deis (<https://deis.com/>)

Сейчас это также проект Kubernetes.

Что умеет Helm?

- упаковывать несколько ресурсов k8s в один пакет - **Chart**
- шаблонизировать установку
- устанавливать **Chart'ы** - делать **Release**
- делать **Upgrade** (обновления) и **Rollback** (откаты) выкатываемых приложений
- управлять зависимостями между пакетами
- хранить пакеты в удаленных репозиториях

Charts

Chart - пакет в Helm.

По сути, chart - коллекция файлов и папок.

Структура Chart'а приложения **ui**

```
ui/  
  Chart.yaml  
  README.md  
  requirements.yaml  
  values.yaml  
  charts/  
  templates/  
  templates/NOTES.txt
```

Chart.yaml

Обязательный файл, содержащий информацию о Chart'е:

- ИМЯ
- описание
- версию чарта
- версию приложения
- различная метайнформация (разработчик, ссылки и т.д)

Chart.yaml

```
name: ui
version: 0.1.1
description: OTUS reddit application
maintainers:
  - name: Dmitry Mischenko
    email: my@mail.com
appVersion: 0.7.0
...
```

Версии приложения и версия Chart'a могут различаться

Templates

Шаблоны манифестов Chart'a, содержатся в директории **templates/**

ui/

Chart.yaml

templates/

ui-deployment.yml

ui-ingress.yml

ui-service.yml

Templates

ui-service.yml

```
---
apiVersion: v1
kind: Service
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
spec:
  type: NodePort
  ports:
  - port: 9292
    protocol: TCP
    targetPort: 9292
  selector:
    app: reddit
    component: ui
```


Templates

ui-service.yml

apiVersion: v1

kind: Service

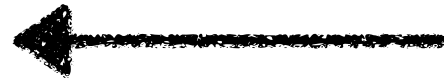
metadata:

name: ui

labels:

app: reddit

component: ui



spec:

type: NodePort

ports:

- port: 9292

protocol: TCP

targetPort: 9292

selector:

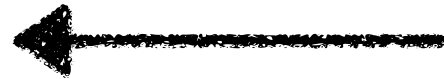
app: reddit

component: ui

Templates

ui-service.yml

```
---
apiVersion: v1
kind: Service
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
spec:
  type: NodePort
  ports:
  - port: 9292
    protocol: TCP
    targetPort: 9292
  selector:
    app: reddit
    component: ui
```



```
---
apiVersion: v1
kind: Service
metadata:
  name: {{ .Release.Name }}-ui
  labels:
    app: reddit
    component: ui
spec:
  type: {{ .Values.service.type }}
  ports:
  - port: {{ .Values.service.port }}
    protocol: TCP
    targetPort: {{ .Values.pod.port }}
  selector:
    app: {{ .Values.metadata.app }}
    component: {{ .Values.metadata.component }}
```

+ values.yml

Templates

ui-service.yml

apiVersion: v1

kind: Service

metadata:

name: {{ .Release.Name }}-ui

labels:

app: reddit

component: ui

spec:

type: {{ .Values.service.type }}

ports:

- port: {{ .Values.service.port }}

protocol: TCP

targetPort: 9292

selector:

app: {{ .Values.metadata.app }}

component: {{ .Values.metadata.component }}

Предопределенные переменные
Helm

Кастомные переменные,
передаваемые через
values.yml

Values

values.yaml

```
---
service:
  type: NodePort
  port: 9292

metadata:
  app: reddit
  component: ui
```

- Указывает, что должно быть подставлено в template
- У каждого chart'а есть свой default-ный файл **values.yaml**
- Значения могут быть перезаписаны для конкретного Release

Go Templating

В основе Helm лежит шаблонизатор Go с 50+ встроенными функциями.

Go Templating

В основе Helm лежит шаблонизатор Go с 50+ встроенными функциями.

```
{{- if .Values.server.persistentVolume.enabled }}  
    persistentVolumeClaim:
```

```
...  
{{- else }}
```

Условия

Go Templating

В основе Helm лежит шаблонизатор Go с 50+ встроенными функциями.

```
{{- if .Values.server.persistentVolume.enabled }}  
    persistentVolumeClaim:
```

Условия

```
...  
{{- else }}
```

```
{{- range $key, $value := .Values.server.annotations }}  
    {{ $key }}: {{ $value }}  
{{- end }}
```

Циклы

Go Templating

В основе Helm лежит шаблонизатор Go с 50+ встроенными функциями.

```
{{- if .Values.server.persistentVolume.enabled }}  
    persistentVolumeClaim:
```

Условия

```
...  
{{- else }}
```

```
{{- range $key, $value := .Values.server.annotations }}  
    {{ $key }}: {{ $value }}  
{{- end }}
```

Циклы

```
value: {{required ".Values.who required!" .Values.who }}
```

Функции

Go Templating

В основе Helm лежит шаблонизатор Go с 50+ встроенными функциями.

```
{{- if .Values.server.persistentVolume.enabled }}  
    persistentVolumeClaim:
```

Условия

```
...  
{{- else }}
```

```
{{- range $key, $value := .Values.server.annotations }}  
    {{ $key }}: {{ $value }}  
{{- end }}
```

Циклы

```
value: {{required ".Values.who required!" .Values.who }}
```

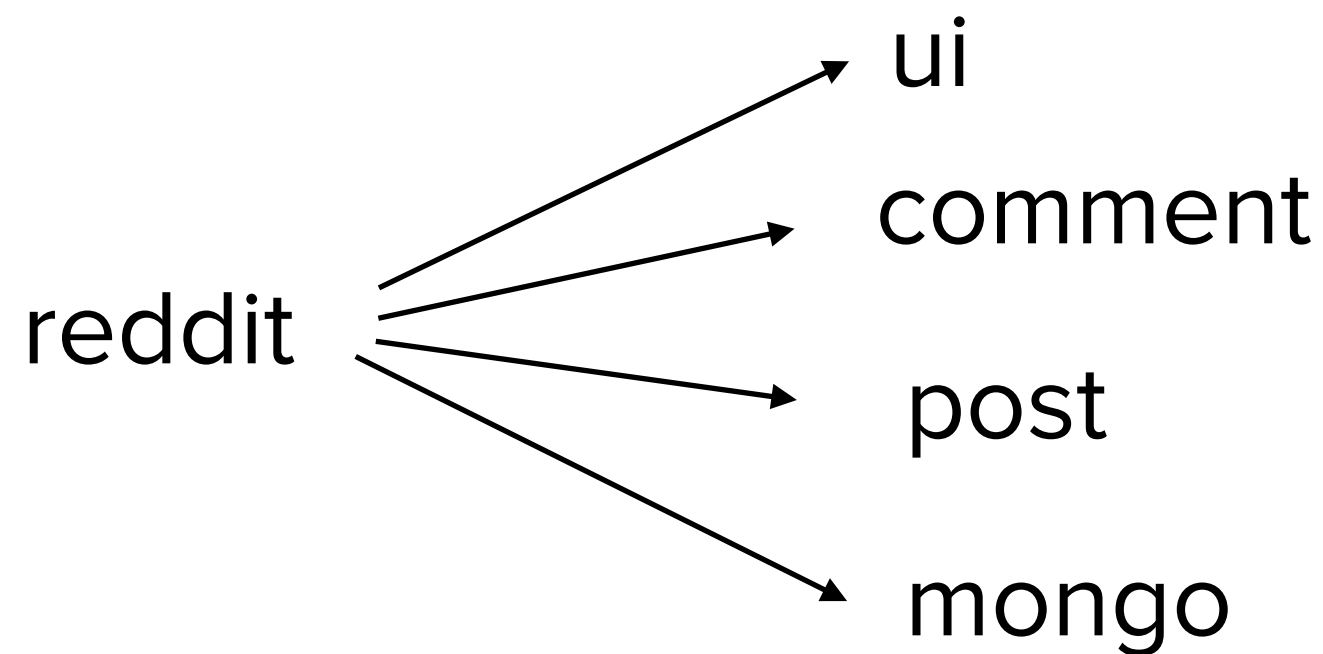
Функции

```
food: {{ .food | upper | quote }}
```

Пайплайны

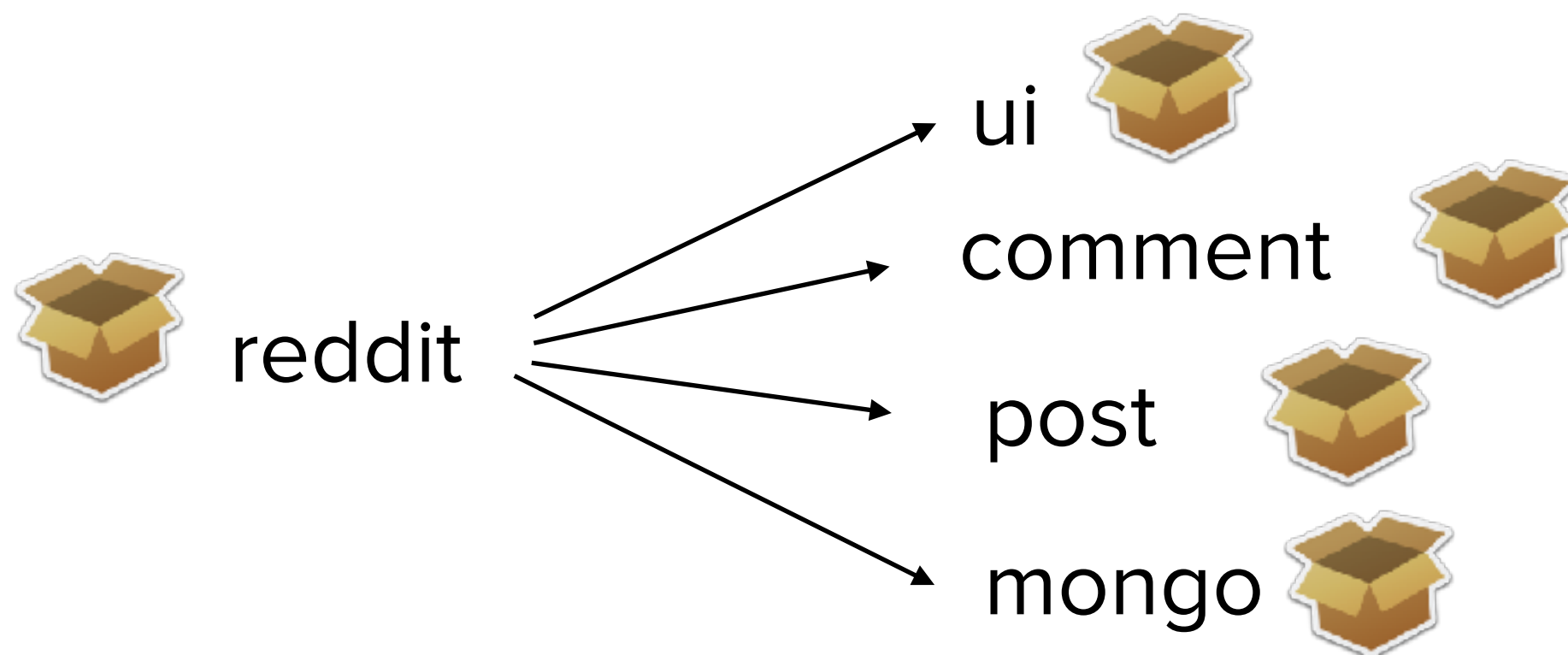
Управление зависимостями

Структура приложения reddit



Управление зависимостями

Структура приложения reddit



Управление зависимостями

Зависимости описываются в **requirements.yml**



```
reddit/  
  requirements.yml  
  charts/  
    ui/  
      Chart.yaml  
      templates/
```

Управление зависимостями






requirements.yml

dependencies:

- name: ui
version: 1.0.0
repository: "file://../ui"
- name: post
version: 2.0.1
repository: "file://../post"
- name: comment
version: 1.0.2
repository: "file://../comment"
- name: mongo
version: 3.2
repository: "http://example.com/charts"

Управление зависимостями

А храним в **charts/**

 reddit/
requirements.yml
charts/
 ui/
Chart.yaml
...
 post/
Chart.yaml
...
 comment/
Chart.yaml
...
 mongo-3.2.tgz

зависимость в виде Chart'a

зависимость в виде архива Chart'a

Repository

Где хранить Chart'ы ?

1) Рядом с кодом приложений

2) Chart-репозиторий

- простой HTTP-сервер
 - файл индекса (**index.yml**) со список данных о чартах в репозитории
 - архивы чартов в **.tgz**-формате
 - **.prov**-файлы для проверки целостности

Helm-client

helm - консольное приложение

- Используем при локальной работе с Chart'ами
- Управление репозиториями
- Взаимодействие с сервером Tiller
 - Отправка информации об установке Chart'ов (Releas'ax)
 - Опрос информации о релизах
 - Запрос на удаление или обновление релиза

Проще говоря Helm-клиент отвечает за работу с **Chart'ами**



Tiller- это серверная часть HELM, расположенная в кластере (это тоже POD)

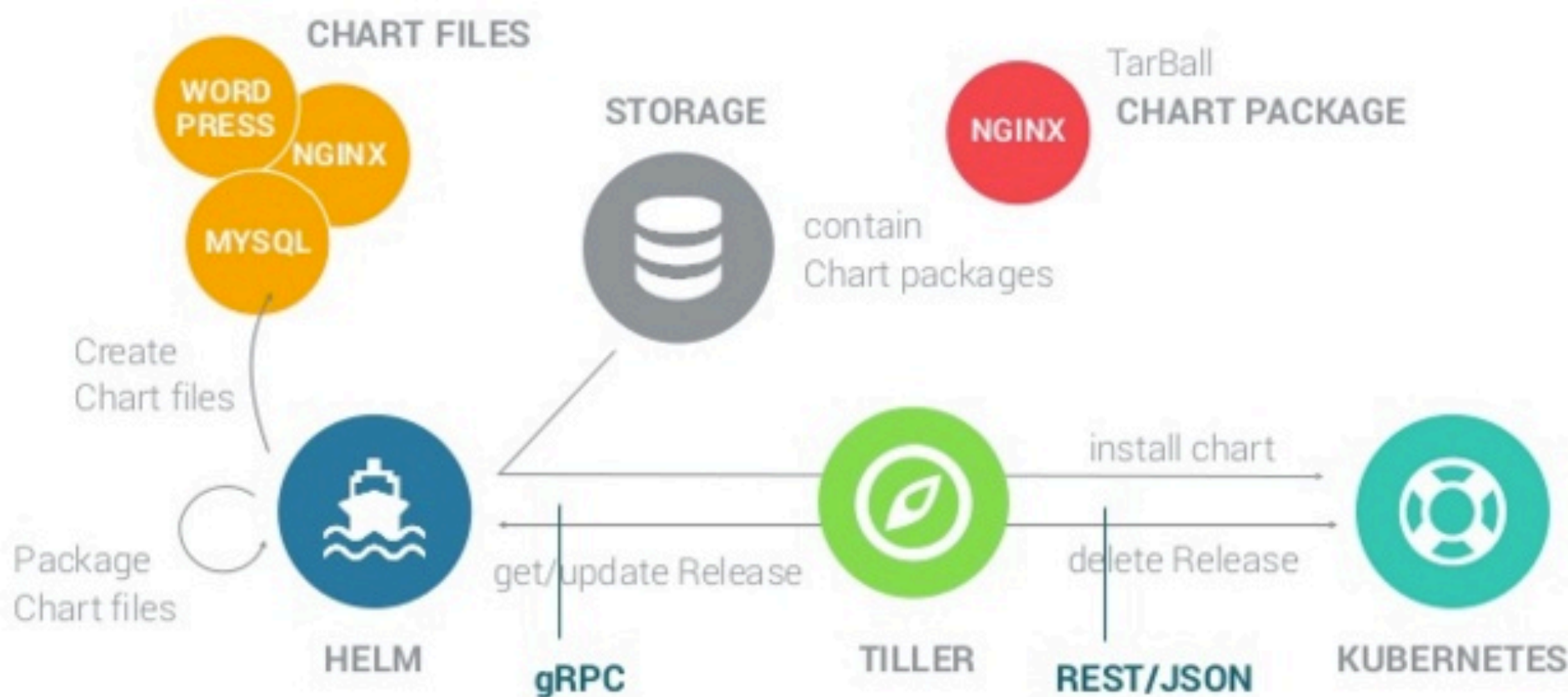
Сервер общается с Kubernetes API и отвечает за:

- Ожидание входящих запросов от Helm-клиента
- Сборку конфигурации чарта в Release
- Установку Chart'ов в Kubernetes и отслеживание соответствующего Release'a
- Обновление и удаление Release'ов

Проще говоря **Tiller**-сервер отвечает за работу с **Release'ами**

Архитектура Helm

Helm Architecture



```
$ helm init
```

Helm install

```
$ helm install reddit --values dev.yml --name reddit --namespace dev
```

```
NAME: reddit
LAST DEPLOYED: Thu Nov 30 15:31:01 2017
NAMESPACE: dev
STATUS: DEPLOYED
```

RESOURCES:

```
==> v1/Service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
comment	ClusterIP	10.11.243.233	<none>	9292/TCP	2s
comment-db	ClusterIP	10.11.241.40	<none>	27017/TCP	2s
post-db	ClusterIP	10.11.251.233	<none>	27017/TCP	2s
post	ClusterIP	10.11.242.113	<none>	5000/TCP	2s
ui	NodePort	10.11.245.194	<none>	9292:32395/TCP	2s

```
==> v1beta1/Deployment
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
comment	1	1	1	0	2s
mongo	1	1	1	0	2s
ui	3	3	3	0	2s

```
==> v1beta2/Deployment
```

```
post 1 1 1 0 2s
```

```
==> v1beta1/Ingress
```

NAME	HOSTS	ADDRESS	PORTS	AGE
ui	*	80	2s	

Release

Release - установленный Chart

```
$ helm ls
```

NAME	REVISION	UPDATED	STATUS	CHART	NAMESPACE
gitlab	14	Thu Nov 30 10:43:07 2017	DEPLOYED	gitlab-omnibus-0.1.36	default
reddit	2	Wed Nov 29 18:17:57 2017	DEPLOYED	reddit-0.1.1	default



имя
релиза



ревизия
(версия релиза)



Версия Chart'a

Helm rollback

```
$ helm rollback reddit 1
```

Номер ревизии

```
$ helm ls
```

NAME	REVISION	UPDATED	STATUS	CHART	NAMESPACE
gitlab	14	Thu Nov 30 10:43:07 2017	DEPLOYED	gitlab-omnibus-0.1.36	default
reddit	3	Wed Nov 29 18:17:57 2017	DEPLOYED	reddit-0.1.0	default

Hooks

Hooks - определенные действия, выполняемые в различные моменты жизненного цикла поставки. Hook, как правило, запускает **Job**.

Виды hook'ов

- pre/post-install
- pre/post-delete
- pre/post-upgrade
- pre/post-rollback

Тесты

- Когда хотим проверить, что запущенный Chart работает.
- Тест - описание Pod'а для проверки
- Лежат в **templates/tests/**
- Тест возвращает либо success (exit code 0) , либо failure

Недостатки HELM

- Проект еще довольно новый
- Kubernetes ничего не знает о **Chart**'ах
- Нет встроенной поддержки окружений
- Нет информативных логов о проблемах
- Слабый Lint'ер (пропускает много ошибок)
- При ошибках установки изменения придется удалять вручную

Почему HELM, а не ...?

Почему HELM, а не ...?

- Единый стандарт упаковки

Почему HELM, а не ...?

- Единый стандарт упаковки
- Дорабатывается сообществом Kubernetes

Почему HELM, а не ...?

- Единый стандарт упаковки
- Дорабатывается сообществом Kubernetes
- Пакеты просто устанавливать и управлять ими

Почему HELM, а не ...?

- Единый стандарт упаковки
- Дорабатывается сообществом Kubernetes
- Пакеты просто устанавливать и управлять ими
- Поддержание консистентного состояния

Почему HELM, а не ...?

- Единый стандарт упаковки
- Дорабатывается сообществом Kubernetes
- Пакеты просто устанавливать и управлять ими
- Поддержание консистентного состояния
- Работающие механизмы обновлений и откатов

Почему HELM, а не ...?

- Единый стандарт упаковки
- Дорабатывается сообществом Kubernetes
- Пакеты просто устанавливать и управлять ими
- Поддержание консистентного состояния
- Работающие механизмы обновлений и откатов
- Предложены механизмы тестирования

GitLab + Kubernetes

CI/CD Инструментарий



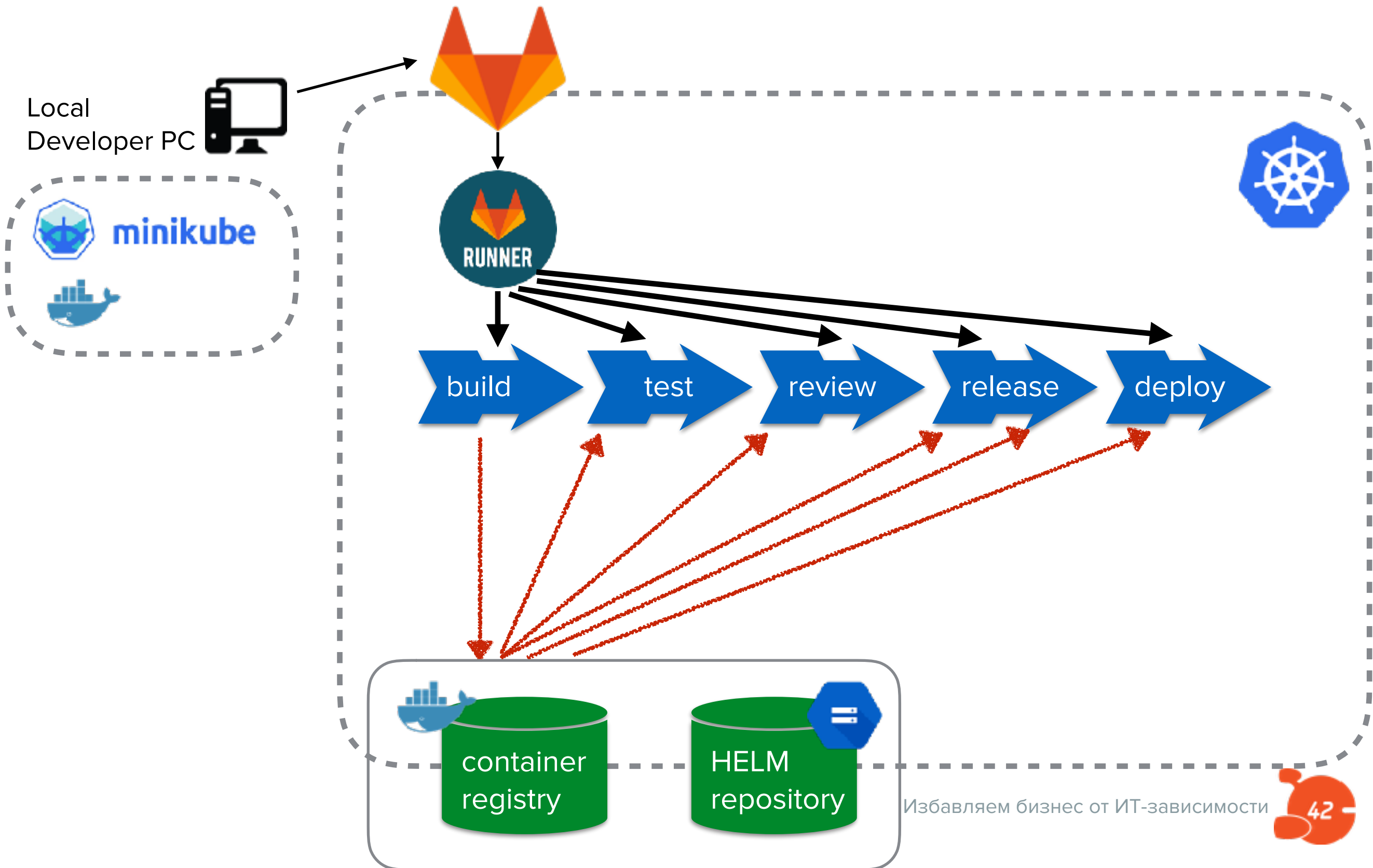
Development

- Можно создать несколько виртуальных кластеров в рамках одного физического кластера.
- **Namespace** - один виртуальный кластер

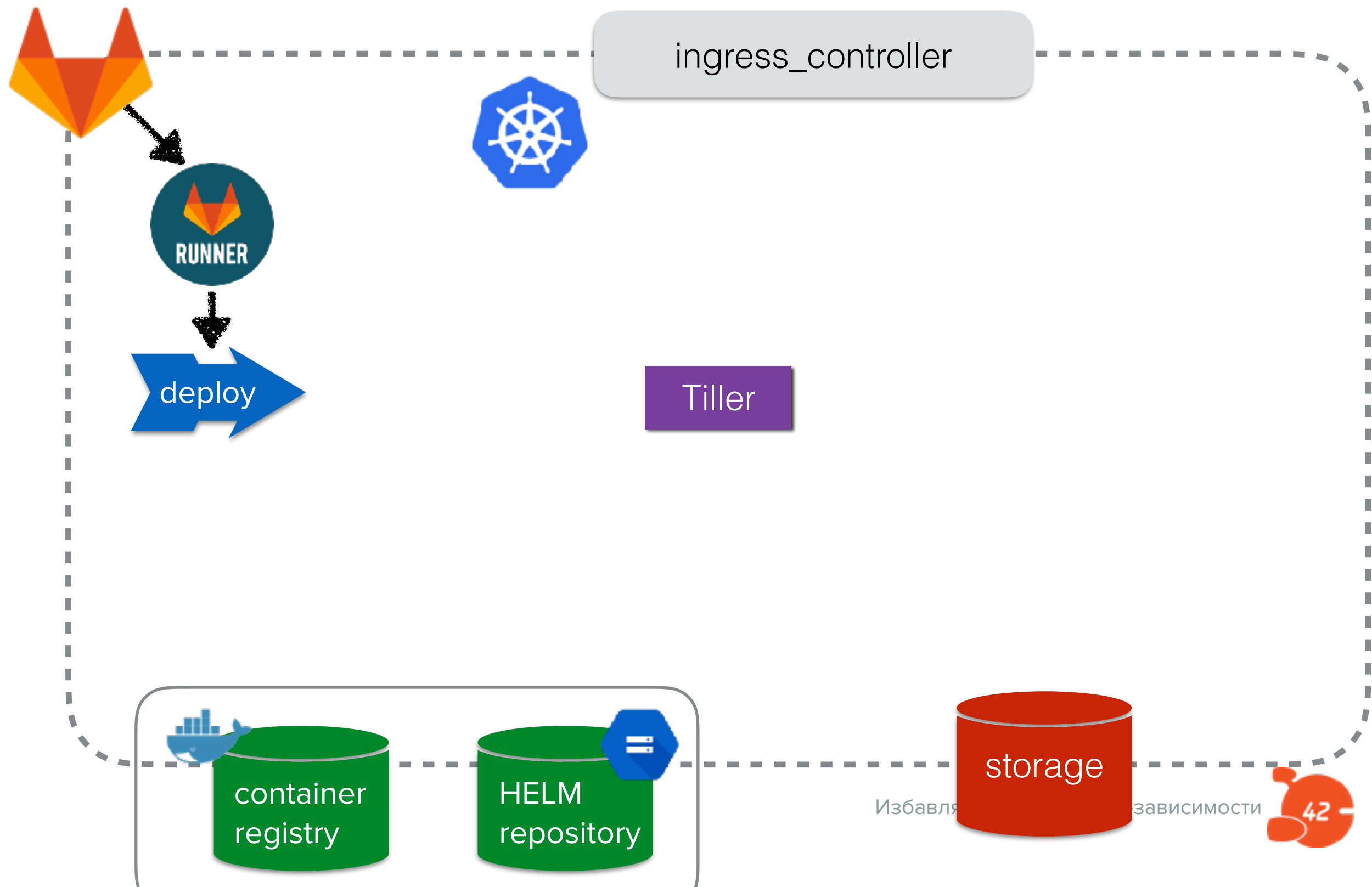
Environments

- Можно создать несколько виртуальных кластеров в рамках одного физического кластера.
- **Namespace** - один виртуальный кластер

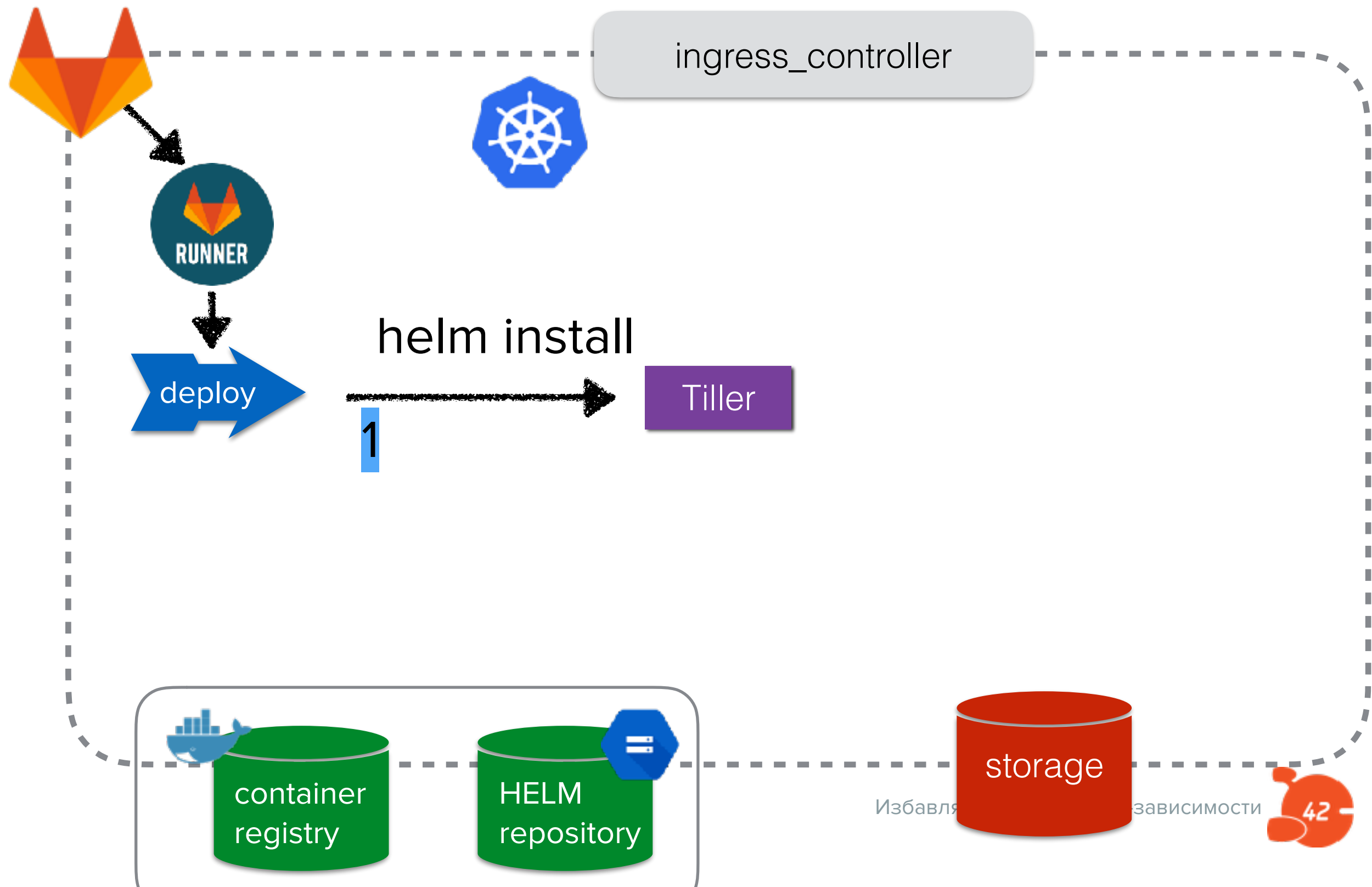
Pipeline



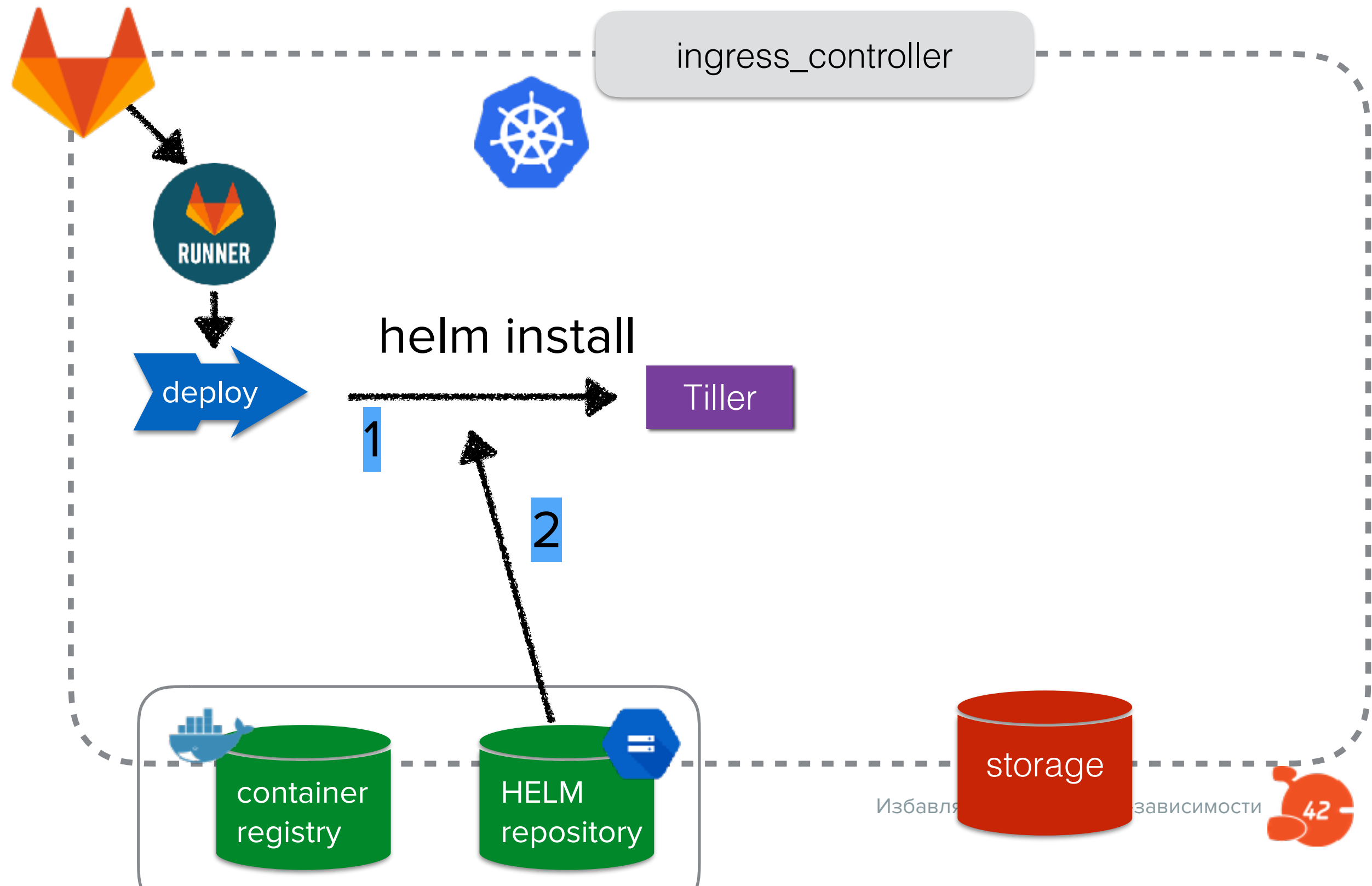
Pipeline



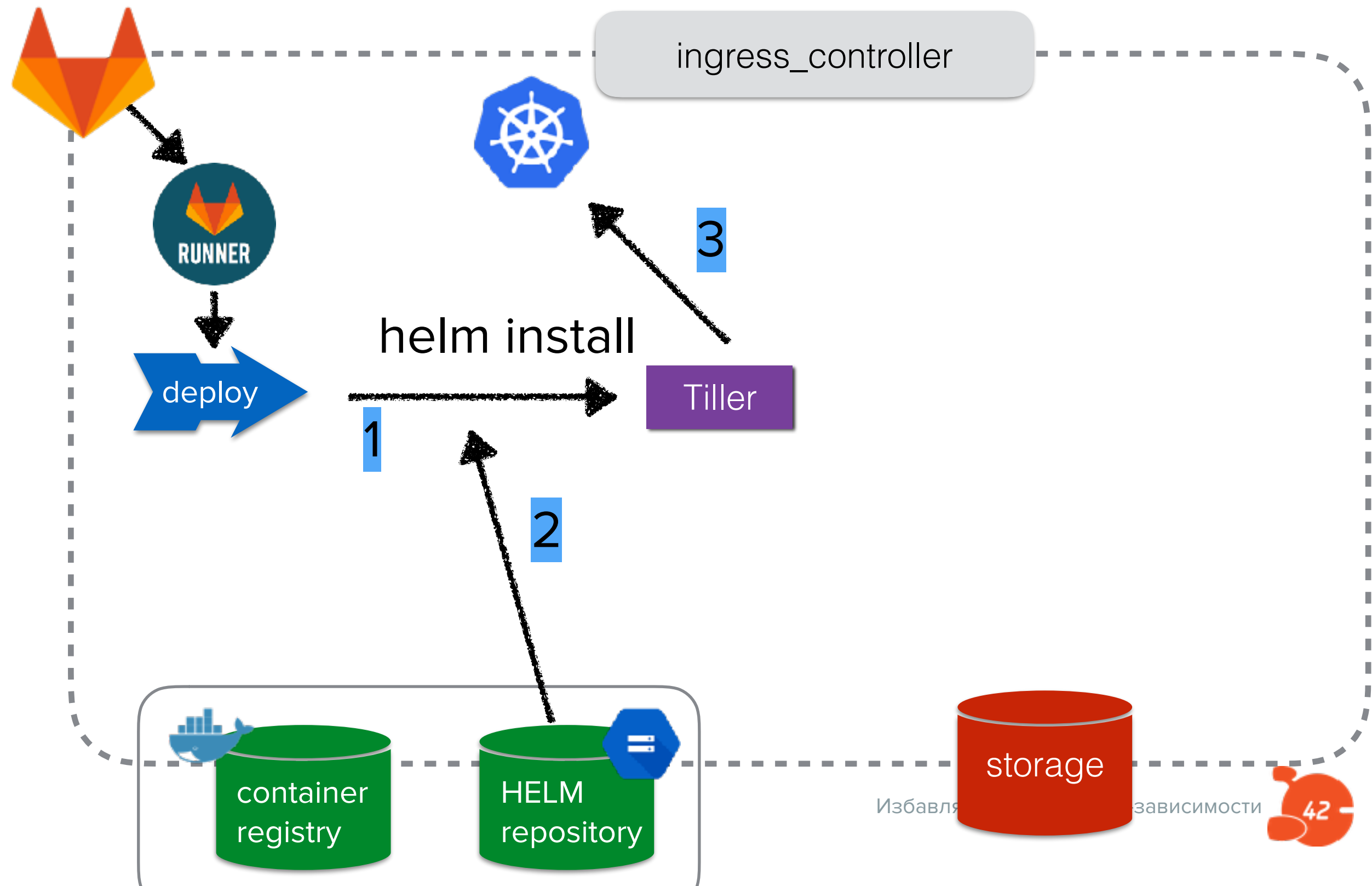
Pipeline



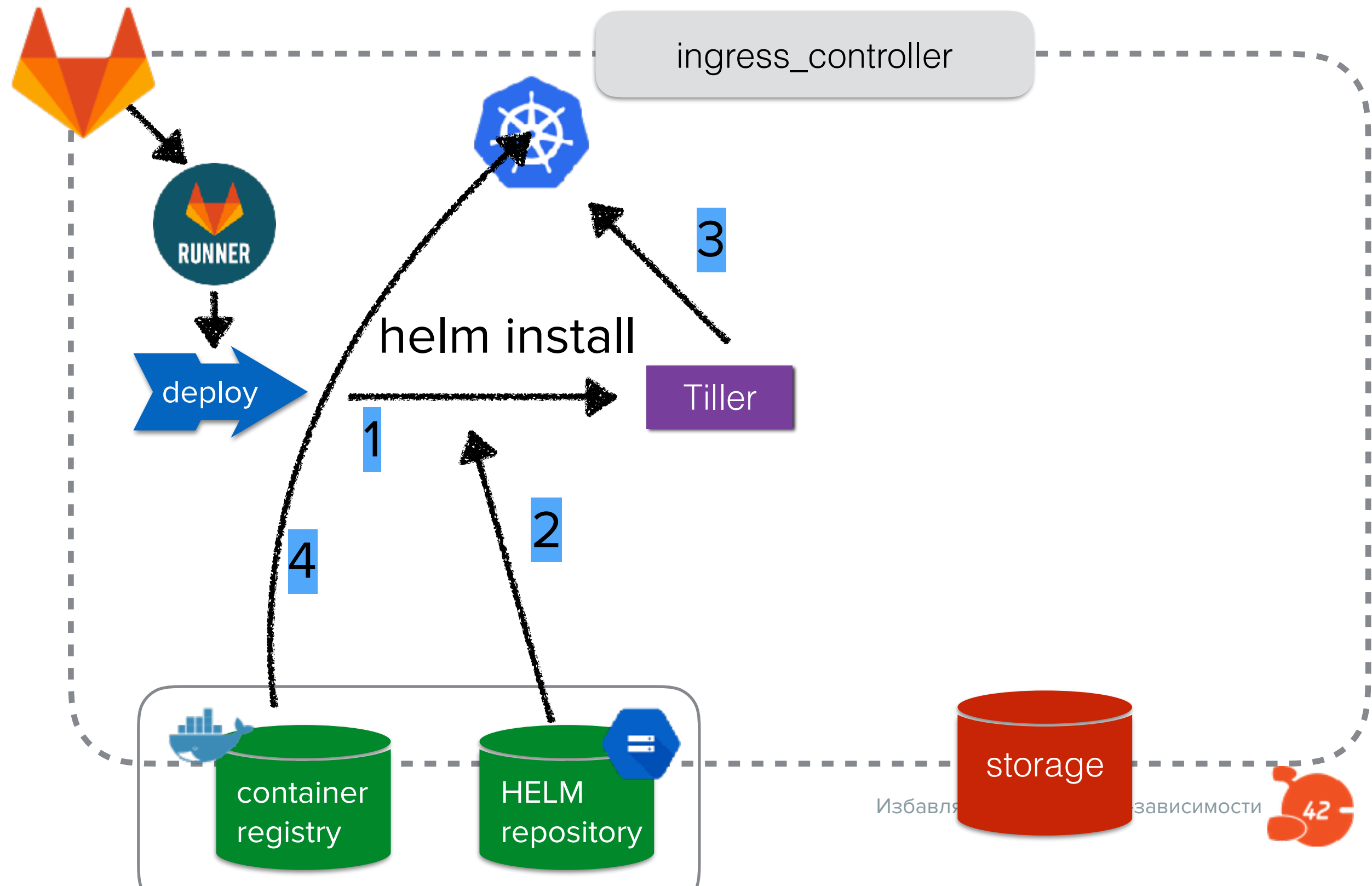
Pipeline



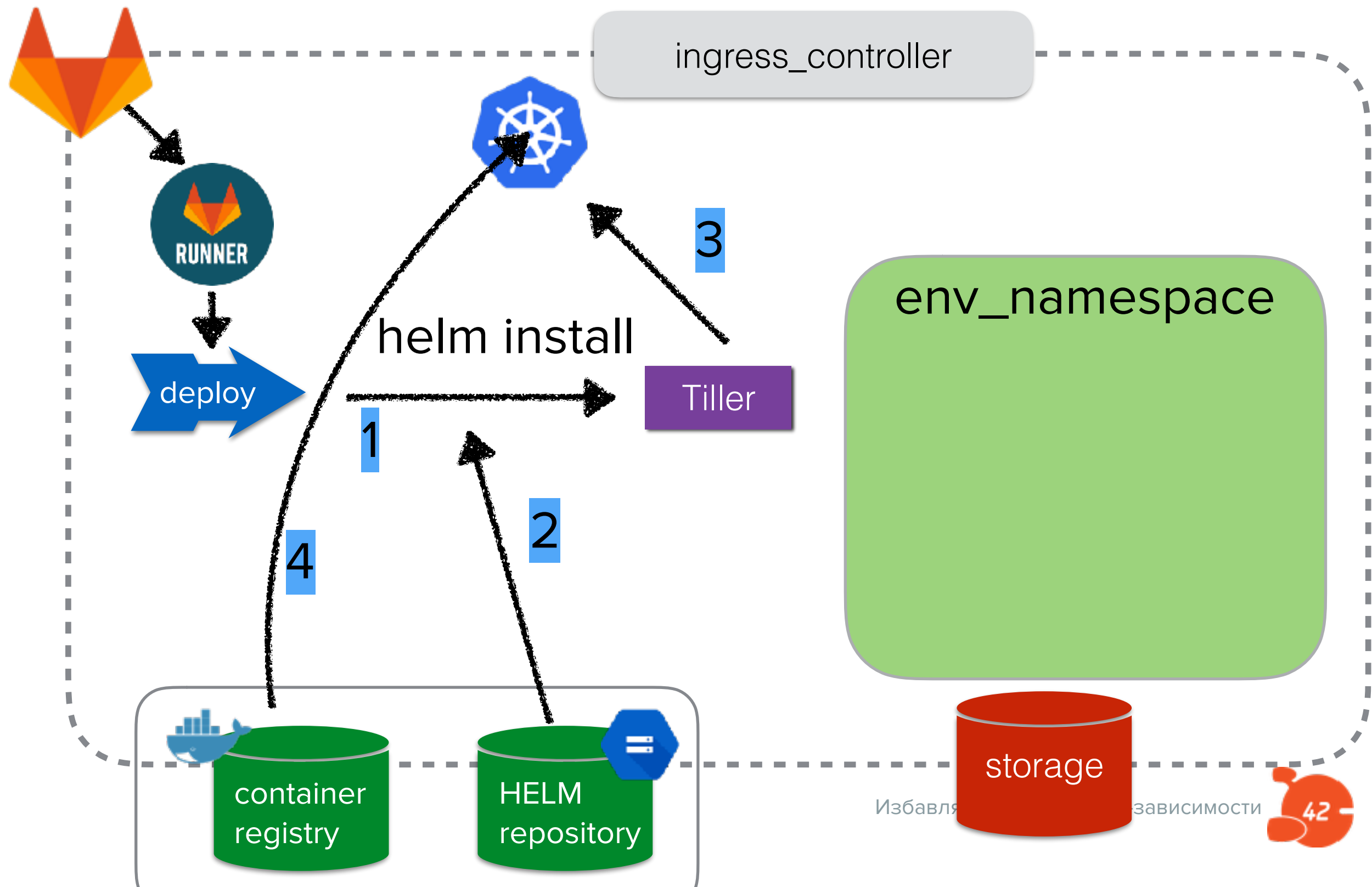
Pipeline



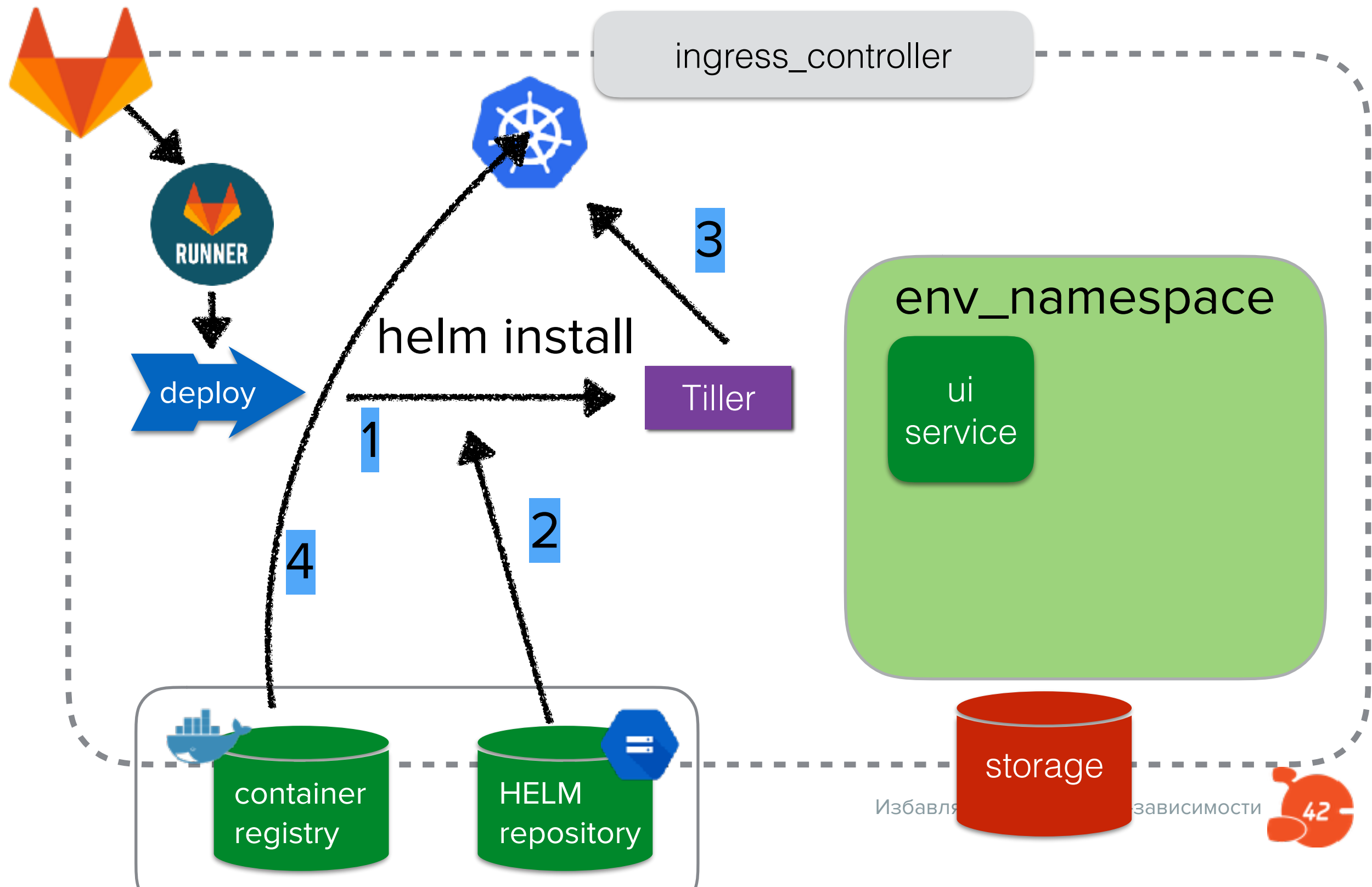
Pipeline



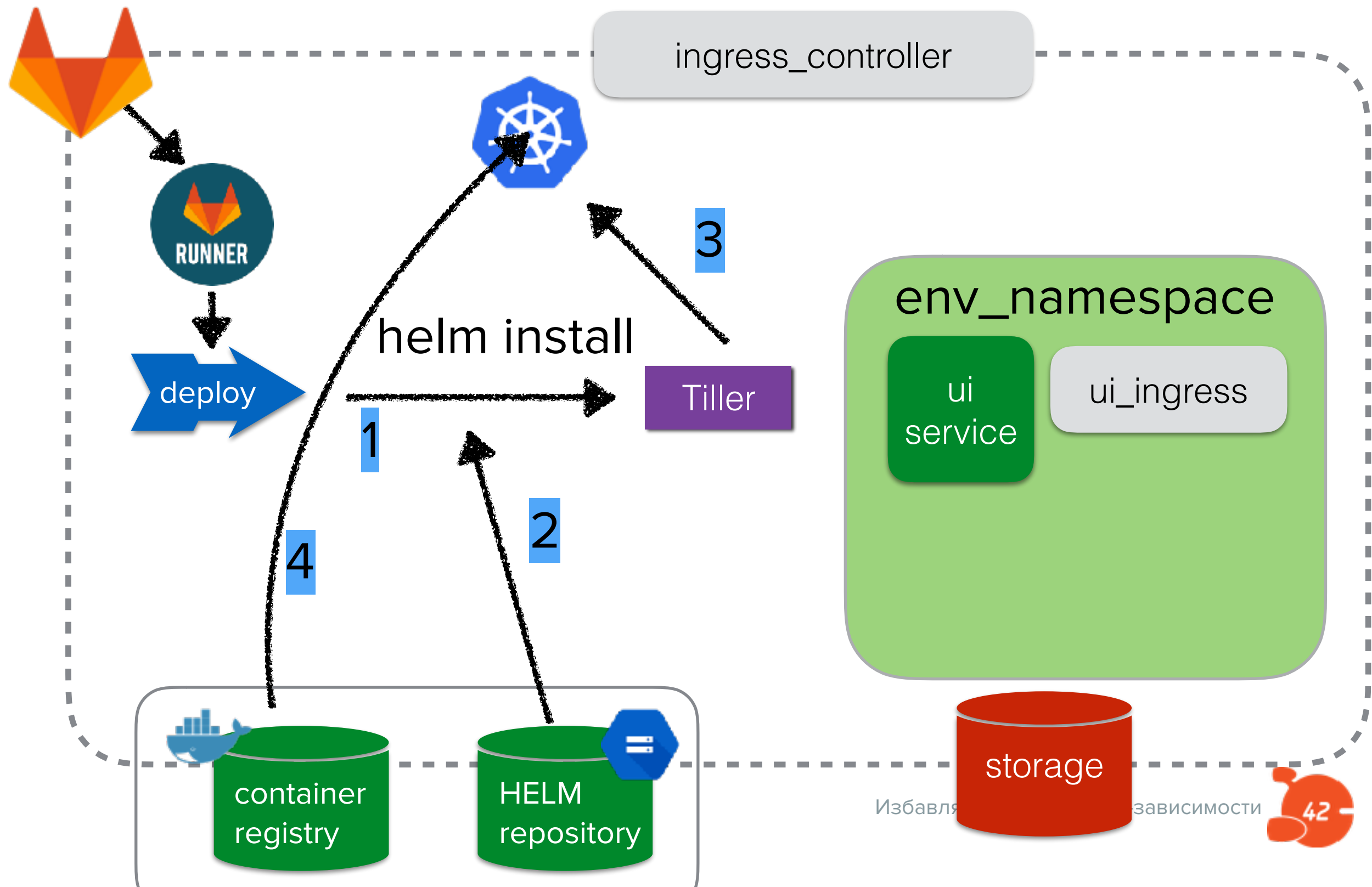
Pipeline



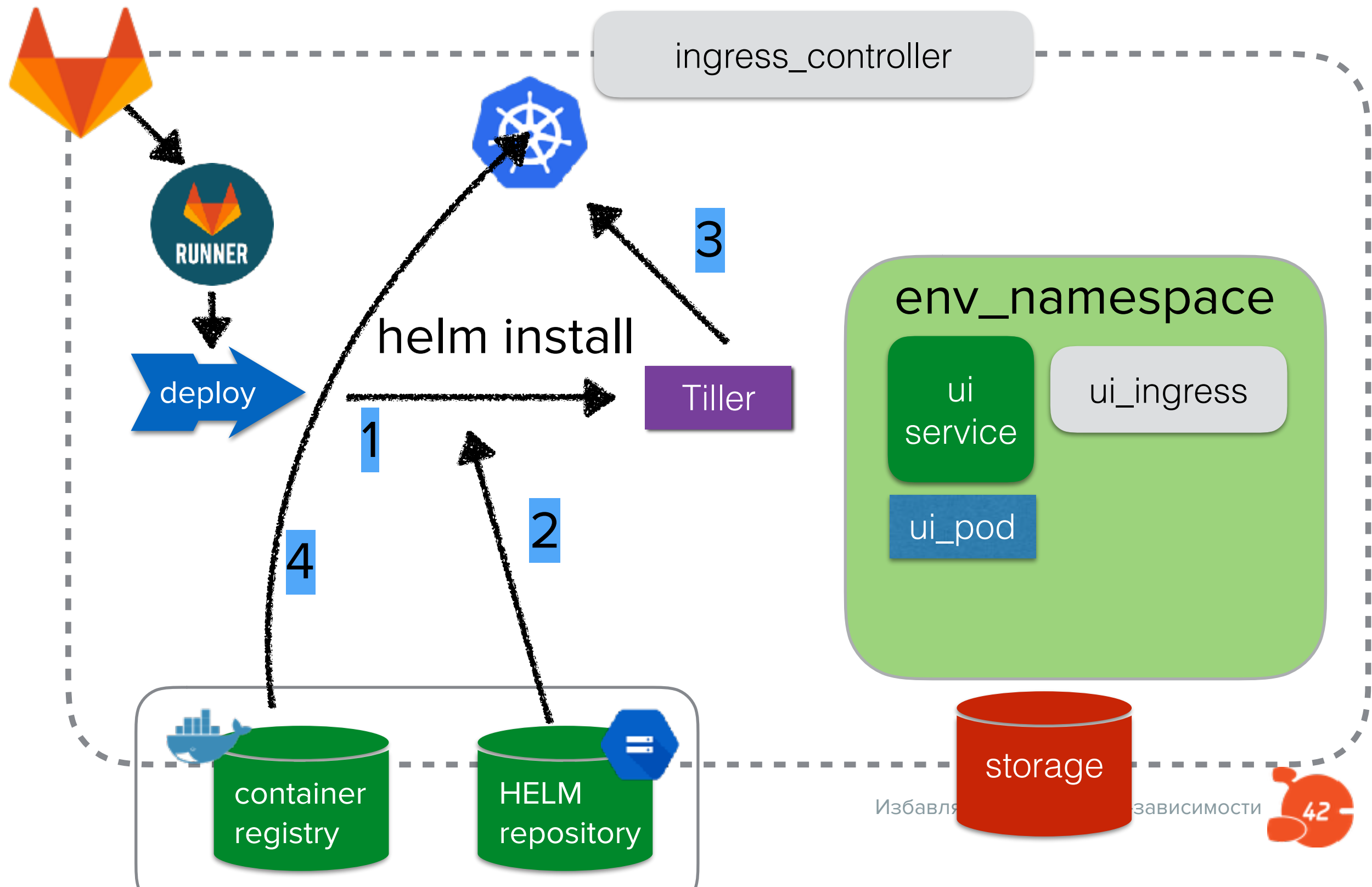
Pipeline



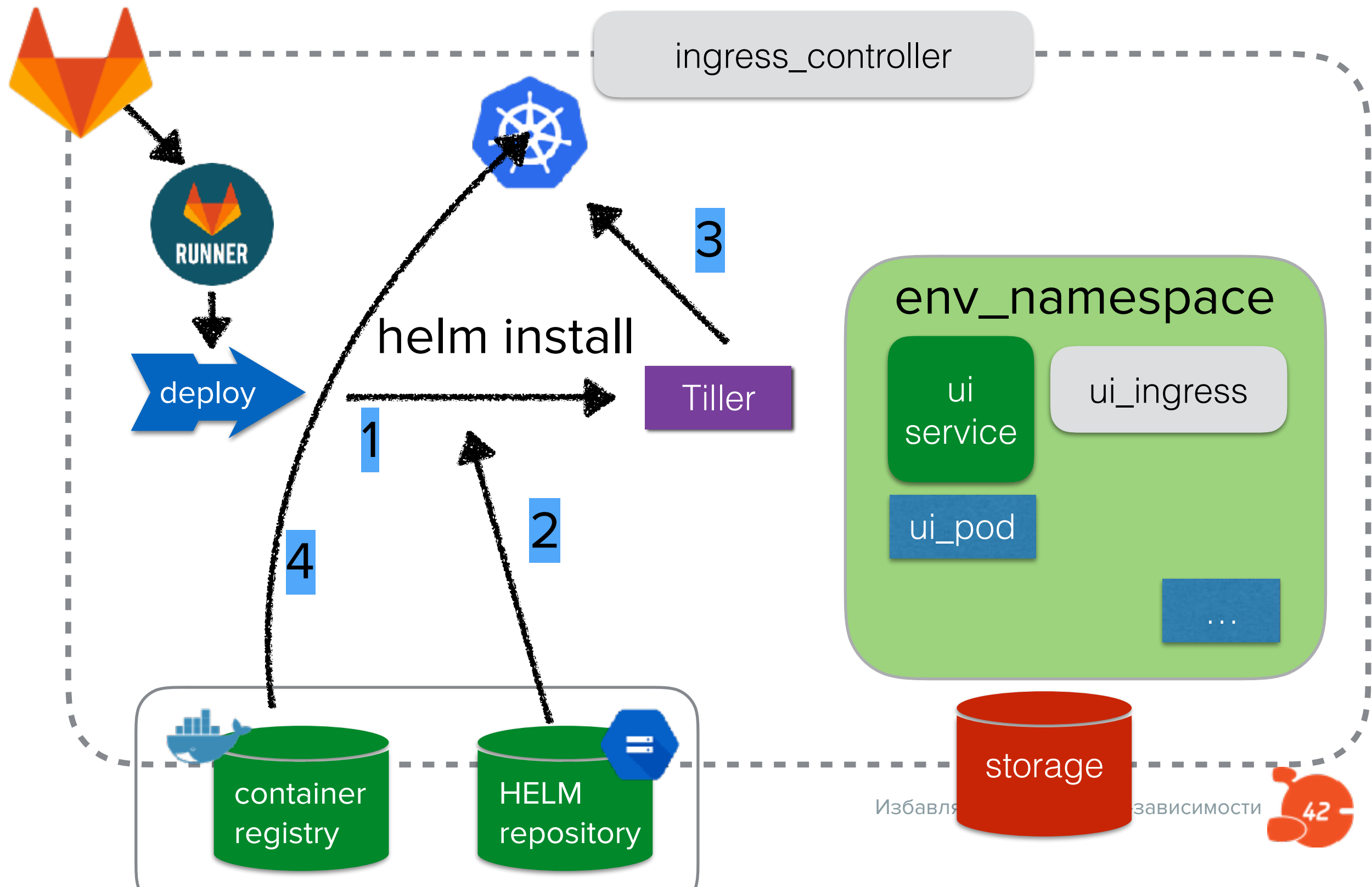
Pipeline



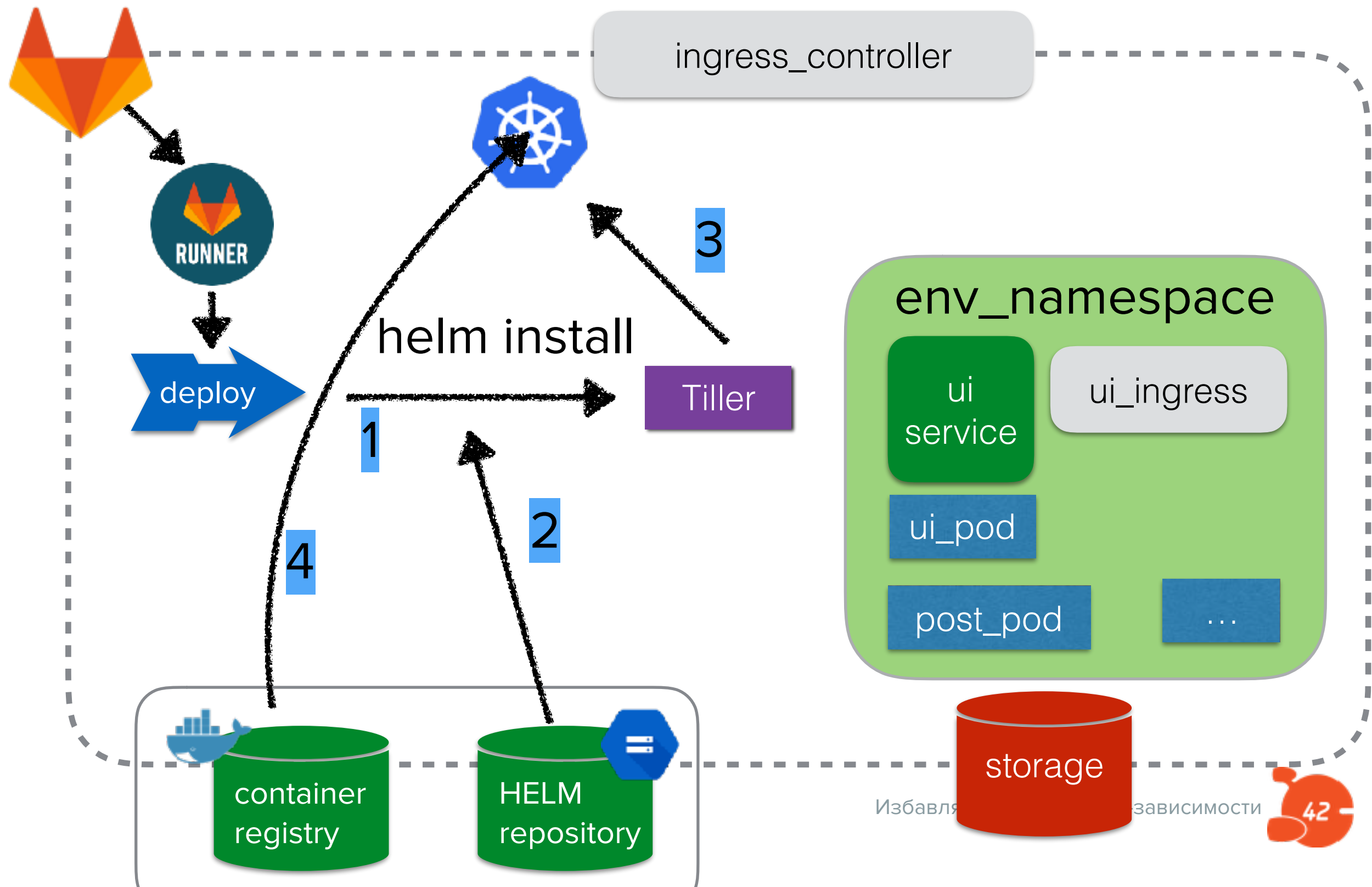
Pipeline



Pipeline



Pipeline



Gitlab + Kubernetes

Kubernetes ●

Kubernetes / Openshift integration

To enable terminal access to Kubernetes environments, label you

Active ☒

API URL

https://10.11.240.1:443

CA Certificate

```
-----BEGIN CERTIFICATE-----  
MIIDCzCCAfOgAwIBAgIQ6pWGovtcMVxy4R  
MS0wKwYDVQQDEyQ0MWNjYzg5Yi05ODJm  
HhcNMTcxMTI2MTUwNzAyWhcNMjIxMTI2  
Yzg5Yi05ODJmLTRmYmMtOTNiYS1hNzNIOV
```

Project
namespace
(optional/unique)

ui-1

Token

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ

Save changes

Environments

homework > example > Pipelines > Environments

Available 6

Stopped 0

New environment

Environment	Deployment	Job	Commit	Updated	
integration	#10 by 	integration #64	 f2d10d92 New login page design	about an hour ago	 Re-deploy
production	#9 by 	production #67	 f2d10d92 New login page design	about an hour ago	 Re-deploy
release	#22 by 	release #143	 99a5b8c9 #5 update ruby version to 2.4.2	7 minutes ago	 Re-deploy
review	#21 by 	dev #141	 99a5b8c9 #5 update ruby version to 2.4.2	7 minutes ago	 Re-deploy
stage	#23 by 	stage #144	 99a5b8c9 #5 update ruby version to 2.4.2	7 minutes ago	 Re-deploy
test	#3 by 	test #42	 12a889cb New site design	about 2 hours ago	 Re-deploy

Pipelines



Deploy Board

Только в Enterprise Edition =(

Project: Repository Registry Issues 0 Merge Requests 1 Pipelines Wiki Settings					
Pipelines Jobs Environments Charts					
Available 3		Stopped 1		New environment:	
Environment	Last deployment	Job	Commit	Updated	
production	#3 by	production #12342187	5da5c440 Merge branch 'auto-deploy' ...	17 minutes ago	Re-deploy
23% Complete		Instances 			
review	#4 by	review #12343014	38b81923 Update server.rb	7 minutes ago	Re-deploy
100% Complete		Instance 			
staging	#2 by	staging #12342186	5da5c440 Merge branch 'auto-deploy' ...	17 minutes ago	Re-deploy
100% Complete		Instances 			

Gitlab AutoDevops

1. Auto Build
2. Auto Test
3. Auto Code Quality
4. Auto Review Apps
5. Auto Deploy
6. Auto Monitoring

Gitlab AutoDevops

- Это набор шаблонов для настройки CI/CD стандартных веб-приложений
- Есть возможность не писать свои (генерирует собственные)
 - Dockerfile
 - gitlab-ci.yml
 - HELM Charts

Деплой в Kubernetes

Стратегии Деплоя

- **recreate**: выключить все старые версии и запустить новые
- **rolling release**: выкатить новую версию, **постепенно** выключая старую
- **blue/green**: запустить новую версию параллельно старой
- **canary**: выпустить новую версию на ограниченное число пользователей

Recreate

recreate: ВЫКЛЮЧИТЬ все старые версии и запустить новые.

Всю работу за нас делает Deployment

```
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
spec:
  replicas: 3
  strategy:
    type: Recreate
```

← Меняем стратегию

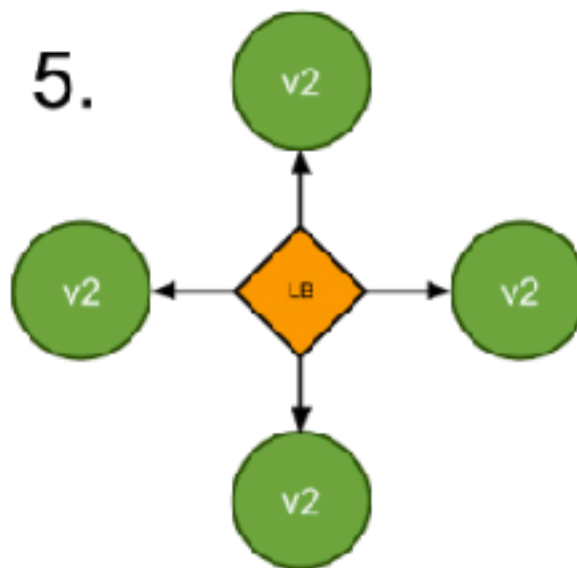
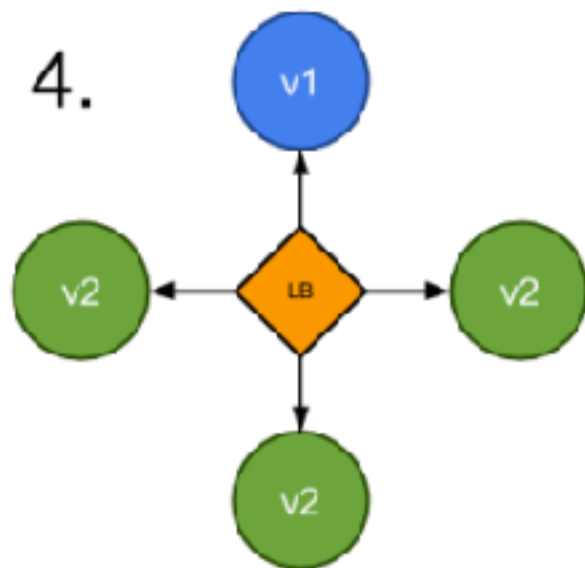
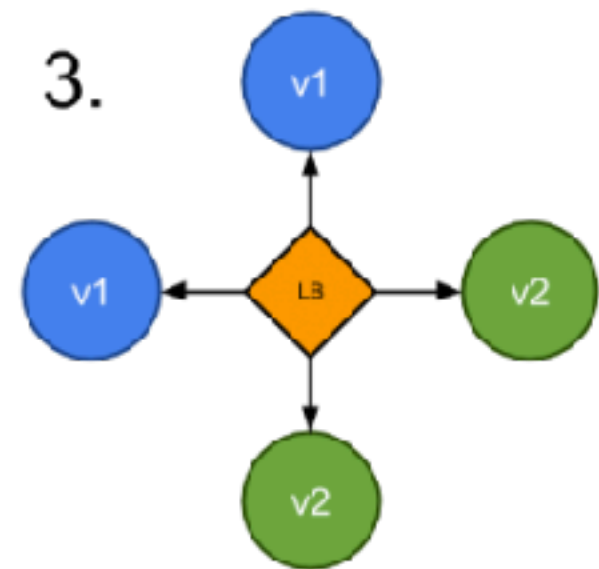
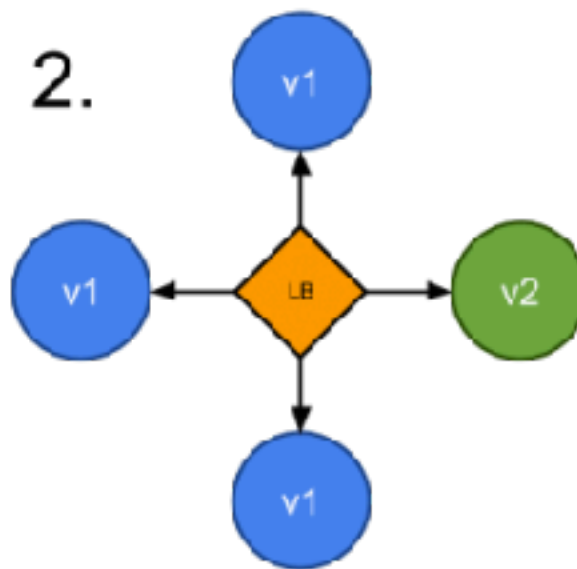
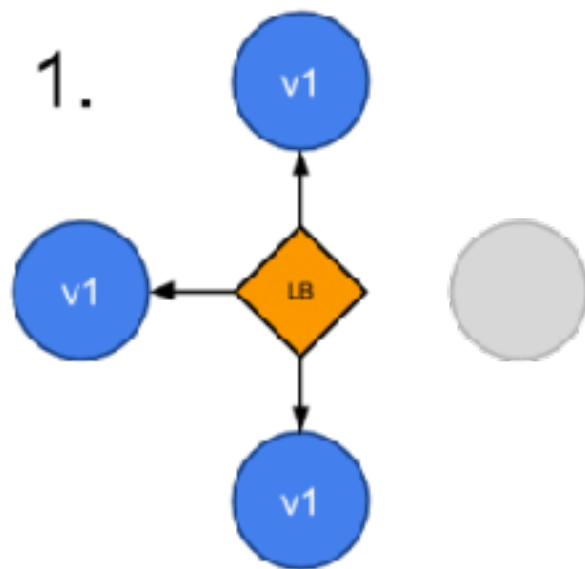
Rolling Release

rolling release (default): ВЫКАТИТЬ НОВУЮ ВЕРСИЮ, ПОСТЕПЕННО
ВЫКЛЮЧАЯ СТАРУЮ

```
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
spec:
  replicas: 3
  strategy:
    type: RollingUpdate
```

← Меняем стратегию

Rolling Release



Blue/Green

Запустить новую версию параллельно старой

Управляется конфигурацией балансировщика!

Blue/Green

Запустить новую версию параллельно старой

Управляется конфигурацией балансировщика!

```
---
apiVersion: v1
kind: Service
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
    version: 1.0
spec:
...
```

Blue/Green

Запустить новую версию параллельно старой

Управляется конфигурацией балансировщика!

```
---  
apiVersion: v1  
kind: Service  
metadata:  
  name: ui  
  labels:  
    app: reddit  
    component: ui  
    version: 1.0  
spec:  
...
```



Blue/Green

Запустить новую версию параллельно старой

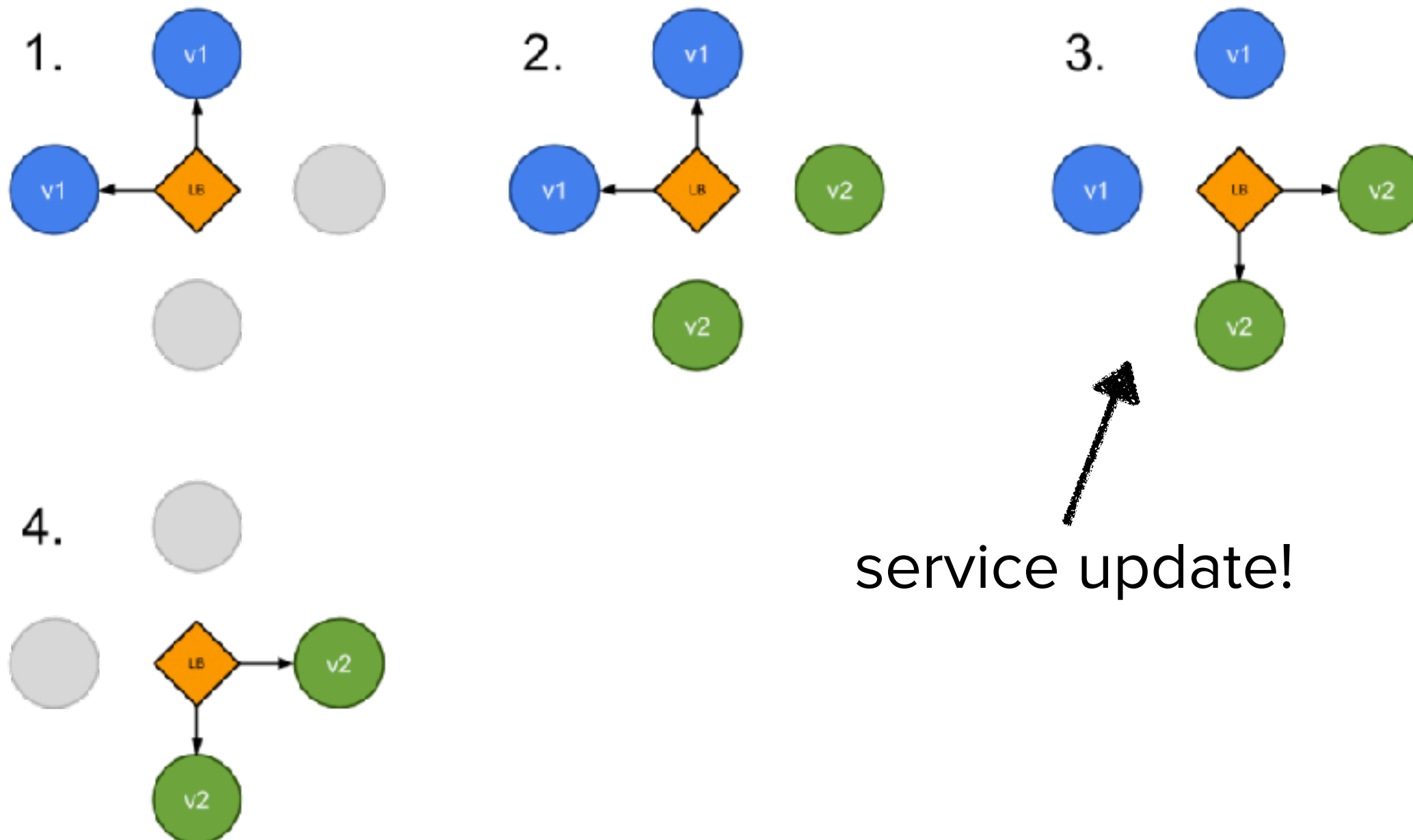
Управляется конфигурацией балансировщика!

```
---  
apiVersion: v1  
kind: Service  
metadata:  
  name: ui  
  labels:  
    app: reddit  
    component: ui  
    version: 1.0  
spec:  
...
```



```
---  
apiVersion: v1  
kind: Service  
metadata:  
  name: ui  
  labels:  
    app: reddit  
    component: ui  
    version: 2.0  
spec:  
...
```


Blue/Green



service update!

Canary

Выпустить новую версию на ограниченное число пользователей

```
---
apiVersion: v1
kind: Service
metadata:
  name: ui
  labels:
    app: reddit
    component: ui
spec:
  type: NodePort
...
selector:
  app: reddit
  component: ui
```

Service перенаправляет
трафик на все
найденные POD-ы

Canary

Выпустить новую версию на ограниченное число пользователей

```
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: ui-ver-1
  labels:
    app: reddit
    component: ui
spec:
  replicas: 3
  template:
    metadata:
      name: ui-pod
    labels:
      app: reddit
      component: ui
```

Canary

Выпустить новую версию на ограниченное число пользователей

apiVersion: apps/v1beta1

kind: Deployment

metadata:

name: ui-ver-1

labels:

app: reddit

component: ui

spec:

replicas: 3

template:

metadata:

name: ui-pod

labels:

app: reddit

component: ui



Canary

Выпустить новую версию на ограниченное число пользователей

```
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: ui-ver-1
  labels:
    app: reddit
    component: ui
spec:
  replicas: 3
  template:
    metadata:
      name: ui-pod
    labels:
      app: reddit
      component: ui
```



```
---
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: ui-ver-2
  labels:
    app: reddit
    component: ui
spec:
  replicas: 1
  template:
    metadata:
      name: ui-pod
    labels:
      app: reddit
      component: ui
```

Canary

Выпустить новую версию на ограниченное число пользователей

