

Ярославский государственный университет им. П.Г. Демидова  
Факультет информатики и вычислительной техники

Отчет по курсу: «Операционные системы семейства UNIX и  
их администрирование»  
Настройка и интеграция Gitea, Jenkins

Работу выполнила  
Студентка ИВТ-32Б0  
Целикова Анна

Установка Docker, Docker Compose.

Переходим на официальный сайт и следуем инструкциям.

<https://docs.docker.com/engine/install/debian/>

<https://docs.docker.com/compose/install/>

Устанавливаем Docker.

```
user@debian:~/unix$ curl -fsSL https://download.docker.com/linux/[80/867]
pg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
user@debian:~/unix$ echo \
> "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring
.gpg] https://download.docker.com/linux/debian \
> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.
list > /dev/null
user@debian:~/unix$ sudo apt-get update
Сущ:1 http://deb.debian.org/debian buster InRelease
Сущ:2 http://security.debian.org/debian-security buster/updates InRelease
Сущ:3 http://deb.debian.org/debian buster-updates InRelease
Пол:4 https://download.docker.com/linux/debian buster InRelease [54,0 kB]
Пол:5 https://download.docker.com/linux/debian buster/stable amd64 Packag
es [19,4 kB]
Получено 73,3 kB за 1с (72,5 kB/s)
Чтение списков пакетов... Готово
user@debian:~/unix$ sudo apt-get install docker-ce docker-ce-cli contain
erd.io
user@debian:~/unix$ sudo docker --version
Docker version 20.10.7, build f0df350
```

Устанавливаем Docker Compose.

```
user@debian:~$ sudo curl -L "https://github.com/docker/compose/releases/
ownload/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin
docker-compose
[sudo] пароль для user:
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
current                                  Dload  Upload  Total  Spent  Left
peed
  0     0     0     0     0     0      0      0  --:--:-- --:--:-- --:--:--
  0     0     0     0     0     0      0      0  --:--:-- --:--:-- --:--:--
100  633 100  633     0     0  1461      0  --:--:-- --:--:-- --:--:--
1458
  0 12.1M    0 30298     0     0 29216      0  0:07:15 0:00:01 0:07:14
 14 12.1M   14 1745k     0     0  857k      0  0:00:14 0:00:02 0:00:12
 33 12.1M   33 4177k     0     0 1358k      0  0:00:09 0:00:03 0:00:06
 48 12.1M   48 6065k     0     0 1502k      0  0:00:08 0:00:04 0:00:04
 63 12.1M   63 7841k     0     0 1556k      0  0:00:07 0:00:05 0:00:02
 80 12.1M   80  9.8M     0     0 1665k      0  0:00:07 0:00:06 0:00:01
100 12.1M 100 12.1M     0     0 1775k      0  0:00:07 0:00:07 --:--:--
2152k
user@debian:~$ sudo chmod +x /usr/local/bin/docker-compose
user@debian:~$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-
ompose
user@debian:~$ sudo docker-compose --version
docker-compose version 1.29.2, build 5becea4c
```

## Docker Compose.

Наша сборка состоит из трех сервисов: Gitea, MySQL (в ней Gitea хранит данные о репозиториях, пользователей и пр.), Jenkins. Gitea зависит от MySQL, поэтому нужно сначала поставить MySQL. Jenkins от них не зависит. В файле указаны переменные окружения, порты и volumes.

```
1  version: "3"
2
3  networks:
4    net:
5      external: false
6
7  services:
8    server:
9      build: ./gitea
10     container_name: gitea
11     environment:
12       - USER_UID=1000
13       - USER_GID=1000
14       - GITEA__database__DB_TYPE=mysql
15       - GITEA__database__HOST=db:3306
16       - GITEA__database__NAME=gitea
17       - GITEA__database__USER=gitea
18       - GITEA__database__PASSWD=gitea
19     restart: always
20     networks:
21       - net
22     volumes:
23       - ./gitea:/data
24       - /etc/timezone:/etc/timezone:ro
25       - /etc/localtime:/etc/localtime:ro
26     ports:
27       - "3000:3000"
28       - "222:22"
29     depends_on:
30       - db
31
```

Мной использовались официальные образы по ссылкам:

1. <https://hub.docker.com/r/jenkins/jenkins>
2. [https://hub.docker.com/\\_/mysql](https://hub.docker.com/_/mysql)
3. <https://hub.docker.com/r/gitea/gitea>

## Конфигурация файлов

Docker-compose.yml

version: "3"

networks:

net:

external: false

services:

server:

build: ./gitea

container\_name: gitea

environment:

- USER\_UID=1000
- USER\_GID=1000
- GITEA\_\_database\_\_DB\_TYPE=mysql
- GITEA\_\_database\_\_HOST=db:3306
- GITEA\_\_database\_\_NAME=gitea
- GITEA\_\_database\_\_USER=gitea
- GITEA\_\_database\_\_PASSWD=gitea

restart: always

networks:

- net

volumes:

- ./gitea:/data
- /etc/timezone:/etc/timezone:ro
- /etc/localtime:/etc/localtime:ro

ports:

- "3000:3000"
- "222:22"

depends\_on:

- db

db:

build: ./mysql

container\_name: db

restart: always

environment:

- MYSQL\_ROOT\_PASSWORD=gitea
- MYSQL\_USER=gitea
- MYSQL\_PASSWORD=gitea
- MYSQL\_DATABASE=gitea

networks:

- net

volumes:

- ./mysql:/var/lib/mysql

jenkins:

build: ./jenkins

container\_name: jenkins

networks:

- net

user: root

ports:

- 8080:8080

volumes:

- ./jenkins:/var/jenkins\_home

Dockerfile (Jenkins)

FROM jenkins/jenkins

Dockerfile (MySQL)

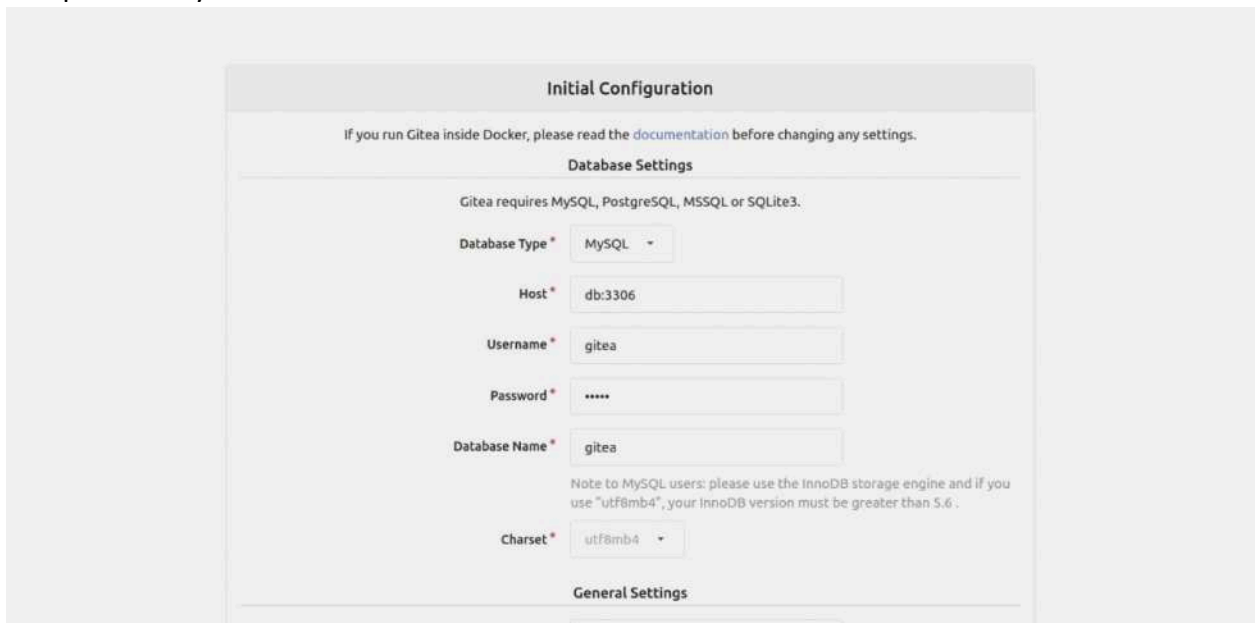
FROM mysql:8

Dockerfile (Gitea)

FROM gitea/gitea:1.14.2

Пример настройки проекта.

Заходим в Gitea по адресу: `http:*ip адрес виртуальной машины*:3000`. Оставляем настройки по умолчанию.



The screenshot shows the 'Initial Configuration' page of Gitea. It includes a warning about Docker usage, a 'Database Settings' section with fields for Database Type (MySQL), Host (db:3306), Username (gitea), Password (masked), and Database Name (gitea), and a 'General Settings' section. A note for MySQL users is also present.

**Initial Configuration**

If you run Gitea inside Docker, please read the [documentation](#) before changing any settings.

**Database Settings**

Gitea requires MySQL, PostgreSQL, MSSQL or SQLite3.

Database Type \*

Host \*

Username \*

Password \*

Database Name \*

Note to MySQL users: please use the InnoDB storage engine and if you use "utf8mb4", your InnoDB version must be greater than 5.6 .

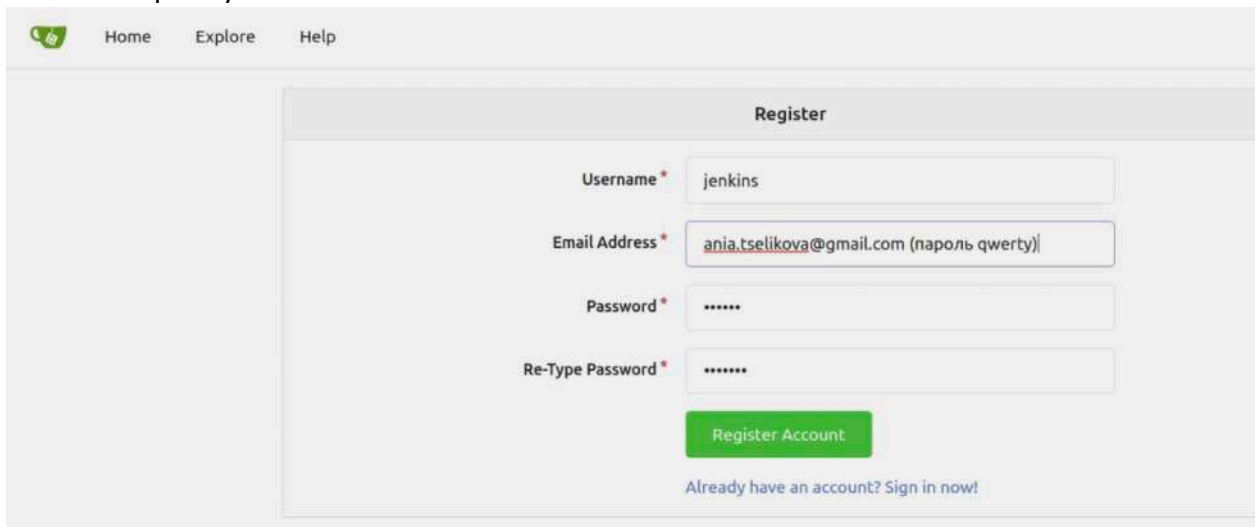
Charset \*

**General Settings**

Создаем нового пользователя.

Username: Jenkins

Password: qwerty



The screenshot shows the 'Register' page of Gitea. It includes a navigation bar with Home, Explore, and Help links. The registration form has fields for Username (jenkins), Email Address (ania.tselikova@gmail.com (наполь qwerty)), Password (masked), and Re-Type Password (masked). A green 'Register Account' button is at the bottom, followed by a link 'Already have an account? Sign in now!'.

**Register**

Username \*

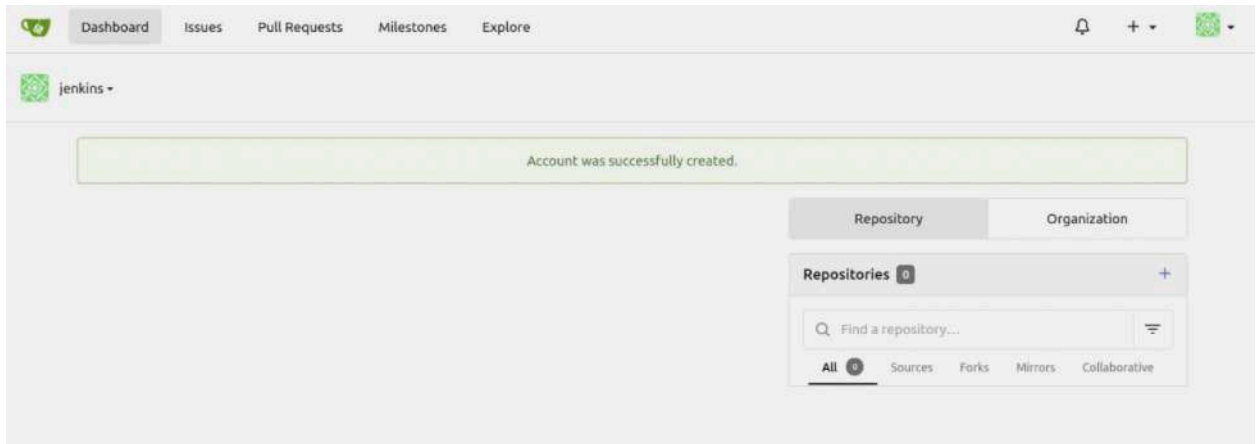
Email Address \*

Password \*

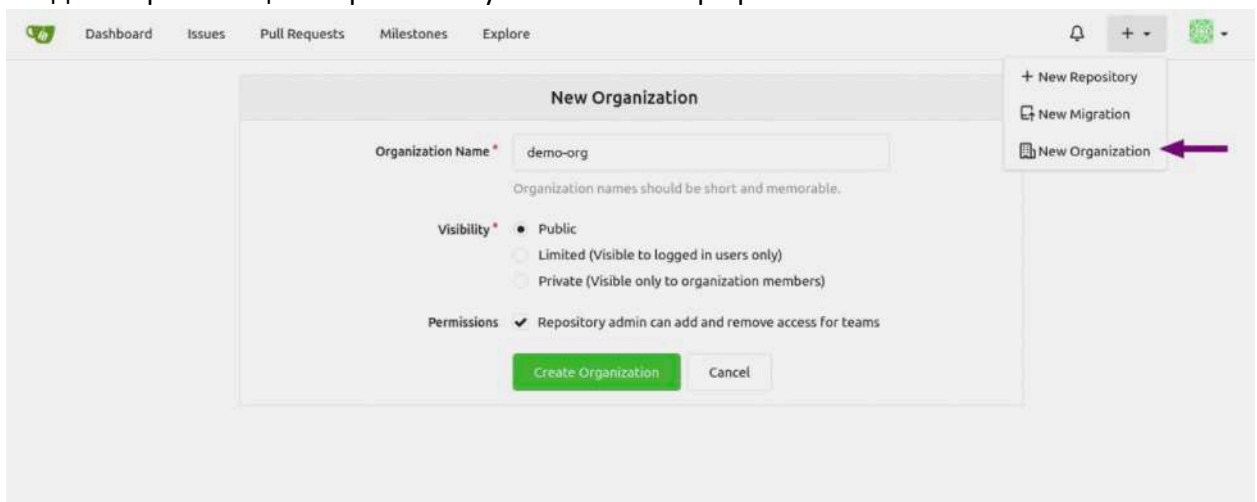
Re-Type Password \*

[Register Account](#)

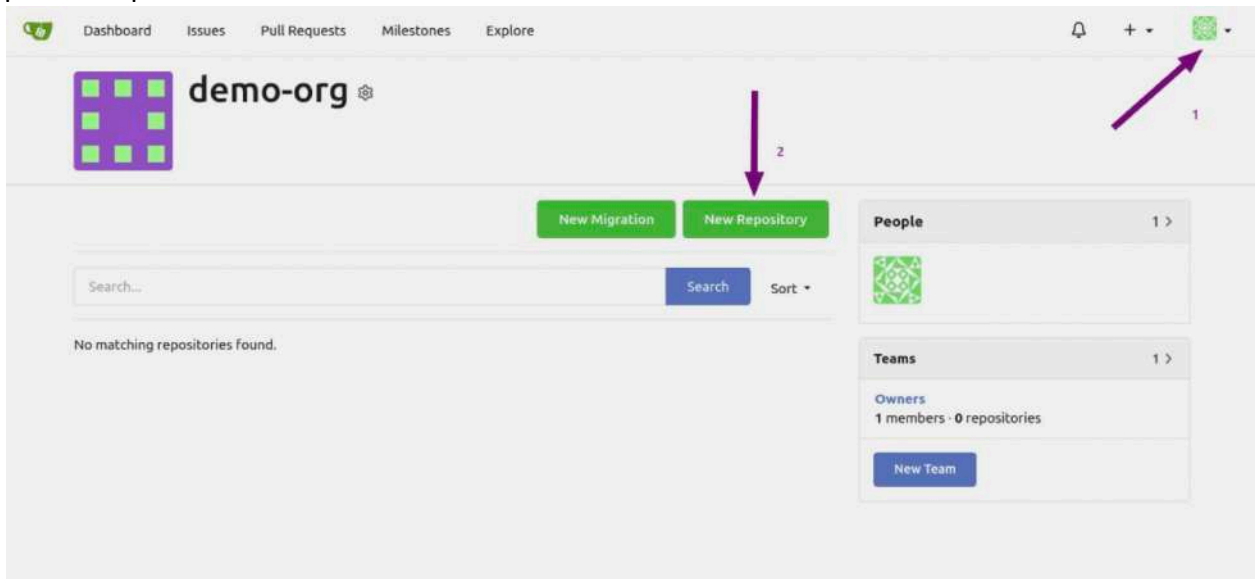
[Already have an account? Sign in now!](#)



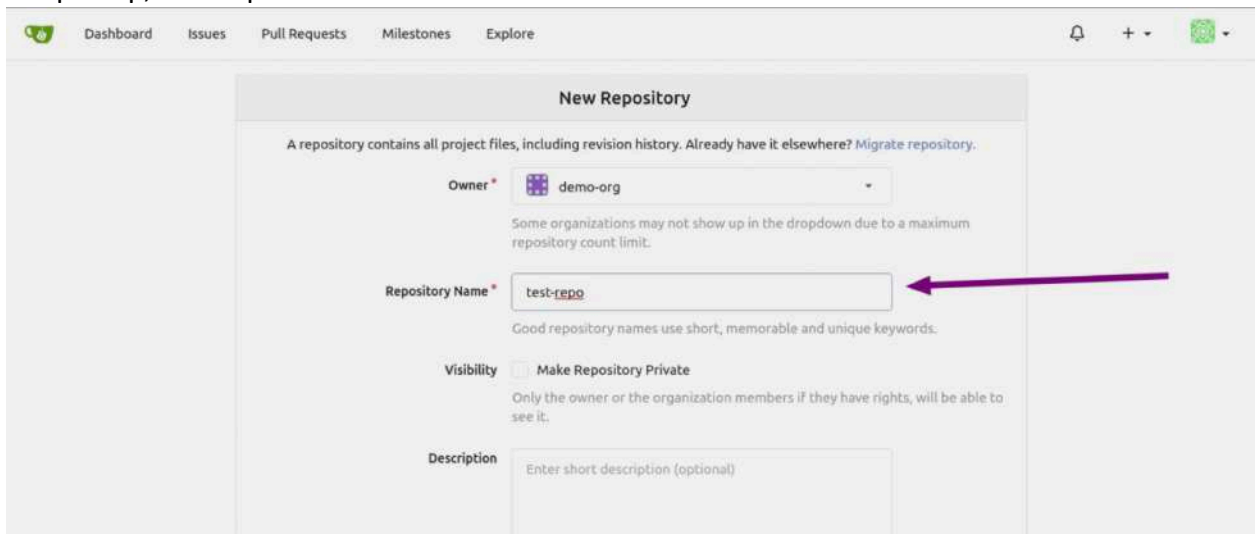
Создаем организацию через кнопку «+» слева от профиля.



Переходим на главную страницу организации для того, чтобы создать новый репозиторий.



Например, test-repo.



The screenshot shows the GitHub 'New Repository' page. The 'Repository Name' field is highlighted with a red arrow and contains the text 'test-repo'. The 'Owner' dropdown is set to 'demo-org'. The 'Visibility' section has the 'Make Repository Private' checkbox unchecked. The 'Description' field is empty.

Клонируем пустой репозиторий. Заметим, что он клонируется по адресу localhost:3000. Gitea как раз «слушает» на порту 3000.

```
user@debian:~/unix$ git clone http://localhost:3000/demo-org/test-repo.git
Клонирование в «test-repo»...
warning: Похоже, что вы клонировали пустой репозиторий.
user@debian:~/unix$ cd test-repo/
user@debian:~/unix/test-repo$ echo "README" > README.md
user@debian:~/unix/test-repo$ git add .
user@debian:~/unix/test-repo$ git config user.name "Anna Tselikova"
user@debian:~/unix/test-repo$ git config user.email "ania.tselikova@gmail.com"
user@debian:~/unix/test-repo$ git commit -m "first commit"
[master (корневой коммит) 79e3214] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
user@debian:~/unix/test-repo$ git push
Username for 'http://localhost:3000': jenkins
Password for 'http://jenkins@localhost:3000':
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 223 bytes | 223.00 KiB/s, готово.
Всего 3 (изменения 0), повторно использовано 0 (изменения 0)
remote: . Processing 1 references
remote: Processed 1 references in total
To http://localhost:3000/demo-org/test-repo.git
 * [new branch]      master -> master
user@debian:~/unix/test-repo$
```



Теперь настроим Jenkins. Перейдем по ссылке: [http://\\*ip](http://*ip) адрес виртуальной машины\*:8080. Для того, чтобы начать им пользоваться нам нужно получить секретный ключ. Получить его можно из логов командой: `sudo docker logs *Jenkins id*`. Уникальный идентификатор можно получить командой: `sudo docker ps`.

```
Please use the following password to proceed to installation:

1515825f3b9f4e44b60f5f90fecb4b95

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*****
*****
*****

2021-06-23 19:32:27.974+0000 [id=29] INFO jenkins.InitReactorRunner
$1#onAttained: Completed initialization
2021-06-23 19:32:28.290+0000 [id=22] INFO hudson.WebAppMain$3#run:
Jenkins is fully up and running
```

Getting Started

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

Continue

Далее необходимо установить рекомендуемые плагины.

Getting Started

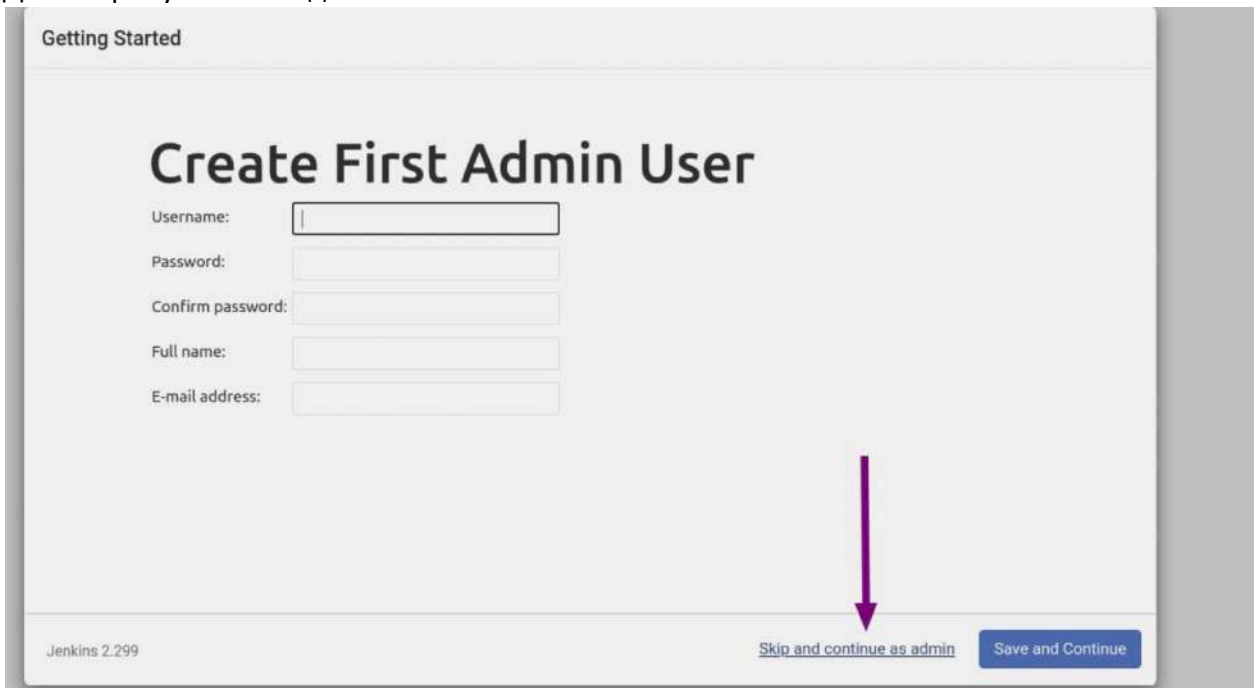
## Getting Started

✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding	** SSH server
Timestampers	Workspace Cleanup	Ant	Gradle	Folders
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication	
LDAP	Email Extension	Mailer		

\*\* - required dependency

Jenkins 2.289.1

Далее пропускаем создание admin пользователя.



Getting Started

## Create First Admin User

Username:

Password:

Confirm password:

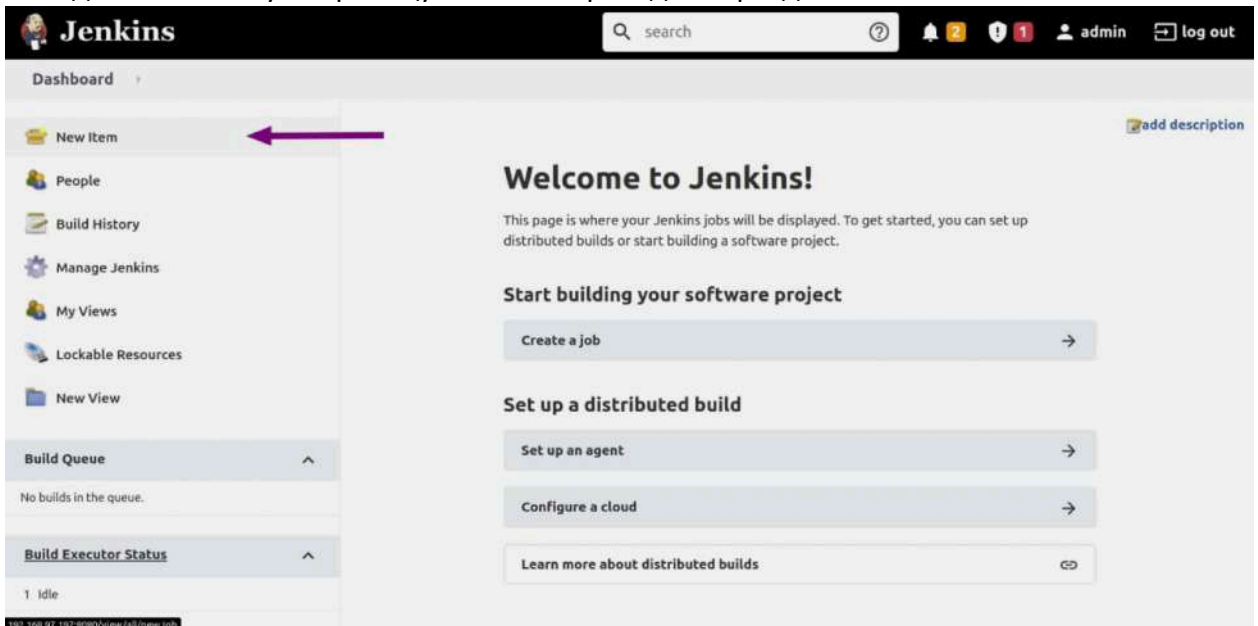
Full name:

E-mail address:

Jenkins 2.299

[Skip and continue as admin](#) [Save and Continue](#)

Попадаем на главную страницу Jenkins. Переходим в раздел “New Item”.



Jenkins

search

Dashboard

New Item

People

Build History

Manage Jenkins

My Views

Lockable Resources

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

add description

## Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

### Start building your software project

Create a job →

### Set up a distributed build

Set up an agent →

Configure a cloud →


Learn more about distributed builds ↗

Создаем Freestyle project и настраиваем.


shboard > All >

## Enter an item name


» Required field

**Freestyle project**

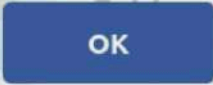
This is the central feature of Jenkins. Jenkins will build your project used for something other than software build.

**Pipeline**

Orchestrates long-running activities that can span multiple build and workflows) and/or organizing complex activities that do not easily

**Multi-configuration project**

Suitable for projects that need a large number of different configurations specific builds, etc.



В Source Code Management добавляем ссылку на репозиторий. Репозиторий будет «лежать» по ссылке `http:gitea:3000/*название организации*/*название репозитория*`.  
gitea – кодовое имя контейнера  
300 – порт

## Source Code Management

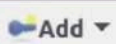
☐ None  
☒ Git

### Repositories

Repository URL

`http://gitea:3000/demo-org/test-repo`

Credentials

- none - 

Там же добавляем новые Credentials. Указываем логин и пароль от Gitea. В моем случае: «jenkins», «qwerty».

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Domain

Global credentials (unrestricted)

Kind

Username with password

Scope

Global (Jenkins, nodes, items, all child items, etc)

Username

jenkins

☐ Treat username as secret

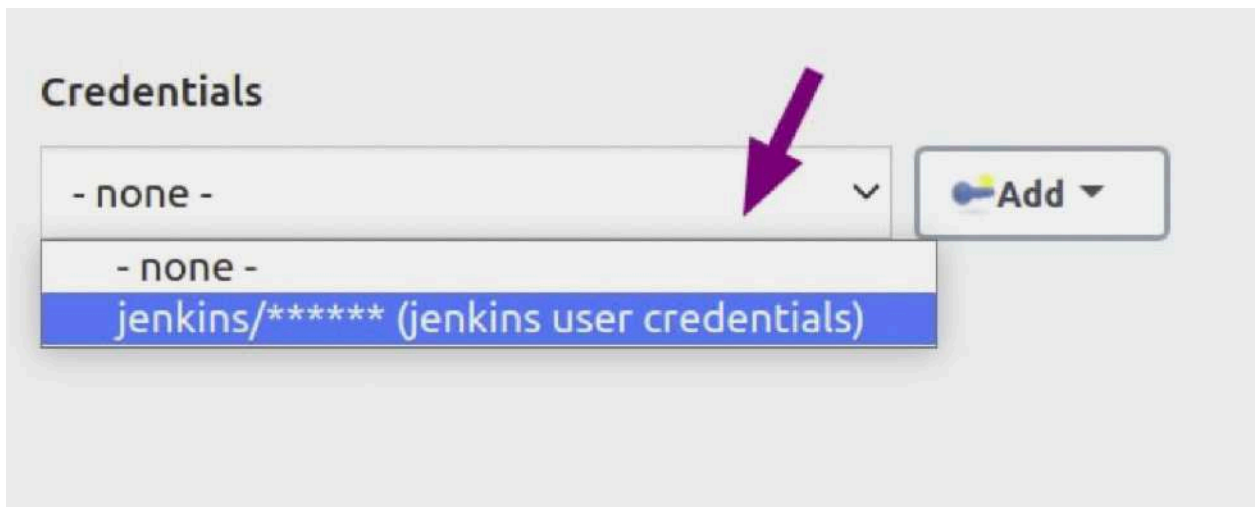
Password

\*\*\*\*\*

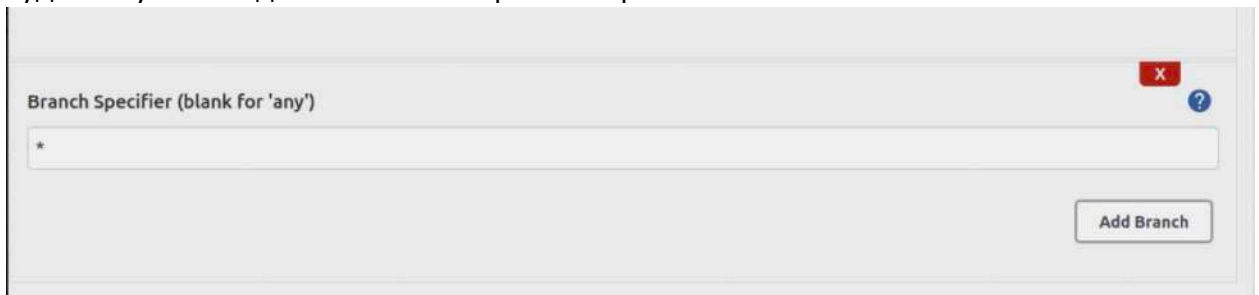
ID

Branch Specifier (blank for 'any')

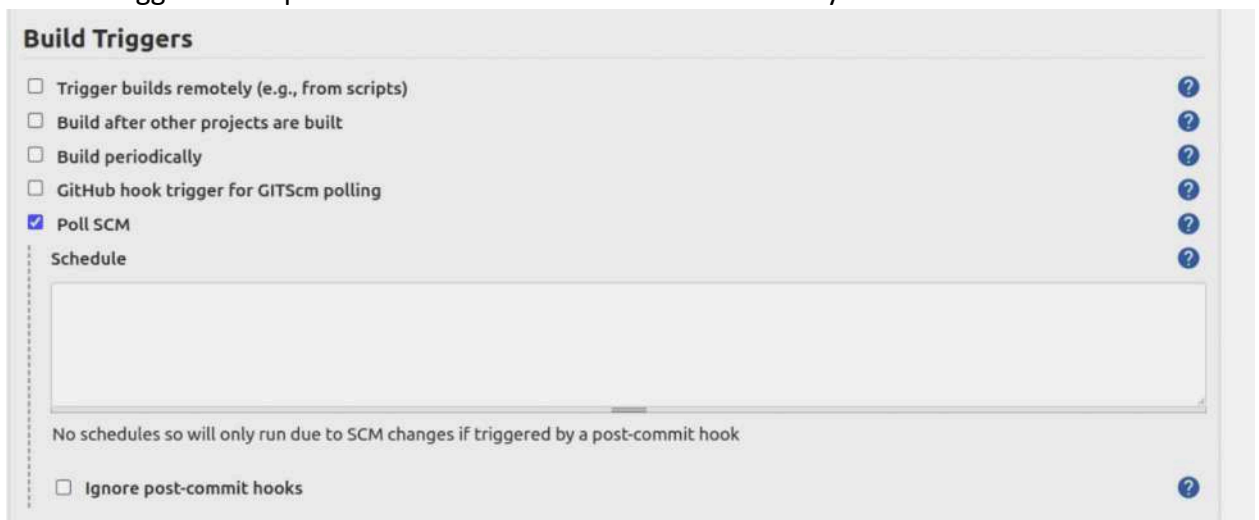
В дропдауне появится новый элемент. Нужно его выбрать.



Настраиваем для каких веток будет запускаться job в Jenkins. “\*” — обозначает, что будет запускаться для любых веток репозитория.



В Build Triggers выбираем Poll SCM и оставляем поле ниже пустым.



**ПРОВЕРЬ** В Build можно прописать команды для исполнения. Результат выполнения можно будет посмотреть в

**Build**

Execute shell X ?

Command

```
echo "I just did something inside Jenkins build action"
```

See [the list of available environment variables](#)

Advanced...

Add build step ▾

Сохраняем.

**Build**

Add build step ▾

**Post-build Actions**

Add post-build action ▾

Save Apply

Перейдем в Manage Plugins.

Dashboard

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Lockable Resources
- New View

**Build Queue** ^

No builds in the queue.

**Build Executor Status** ^

1 Idle

2 Idle

## Manage Jenkins




Building on the controller node can be a security issue. You should set up distributed builds. See [the documentation](#). Set up agent Set up cloud Dismiss

Jenkins URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as BUILD\_URL. Dismiss

Please provide an accurate value in [Jenkins configuration](#).

Java 11 is the recommended version to run Jenkins on, please consider upgrading. More info Dismiss

## System Configuration

-  **Configure System**  
Configure global settings and paths.
-  **Global Tool Configuration**  
Configure tools, their locations and automatic installers.
-  **Manage Plugins**  
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.  
There are updates available

Установим плагин Gitea и Gogs.

The image shows two screenshots of the Jenkins Plugin Manager interface. The top screenshot displays the 'Installed' tab for a search of 'gogs'. A purple arrow points to the 'Gogs plugin' entry, which is checked and has a description: 'This plugin adds Gogs integration to Jenkins.' The bottom screenshot shows the 'Available' tab for a search of 'gitea'. It lists two plugins: 'Gitee' (unchecked) and 'Gitea' (checked). The 'Gitea' plugin is categorized under 'git' and 'Source Code Management', with a description: 'SCM API implementation for Gitea'. At the bottom of the 'Gitea' entry, there are two buttons: 'Install without restart' (highlighted by a purple arrow) and 'Download now and install after restart'. A third button, 'Update information', is partially visible on the right.

**Top Screenshot: Jenkins Plugin Manager - Installed Tab**

Search: gogs

Updates Available **Installed** Advanced

Enabled	Name
<input checked="" type="checkbox"/>	<b>bouncycastle API Plugin</b> This plugin provides an stable API to Bouncy Castle related tasks.
<input checked="" type="checkbox"/>	<b>Command Agent Launcher</b> Allows agents to be launched using a specified command.
<input checked="" type="checkbox"/>	<b>Git plugin</b> This plugin integrates <a href="#">Git</a> with Jenkins.
<input checked="" type="checkbox"/>	<b>Gogs plugin</b> This plugin adds Gogs integration to Jenkins.
<input checked="" type="checkbox"/>	<b>Oracle Java SE Development Kit Installer Plugin</b> Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.

**Bottom Screenshot: Jenkins Plugin Manager - Available Tab**

Dashboard > Plugin Manager

Back to Dashboard Manage Jenkins Update Center

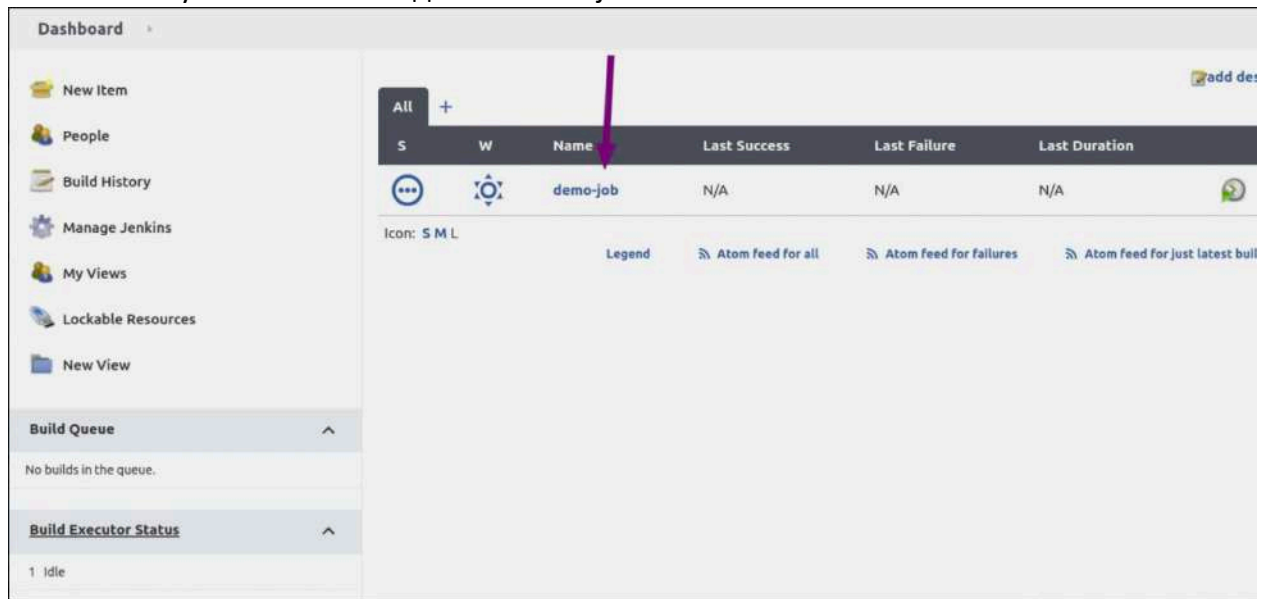
Search: gitea

Updates **Available** Installed Advanced

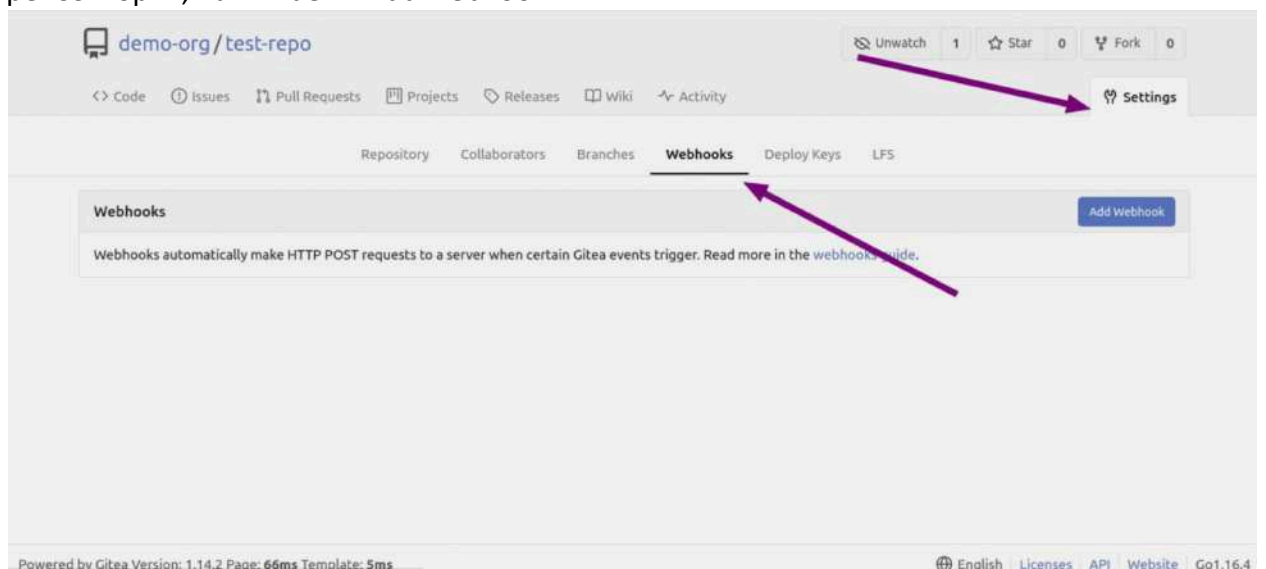
Install	Name
<input type="checkbox"/>	<b>Gitee</b> This plugin integrates <a href="#">Gitee</a> to Jenkins by faking a Gitee CI Server. This plugin trigger builds in Jenkins when code is committed or pull requests are opened. It also send build status back to Gitee.
<input checked="" type="checkbox"/>	<b>Gitea</b> <span>git</span> <span>Source Code Management</span> SCM API implementation for Gitea

Install without restart Download now and install after restart Update information

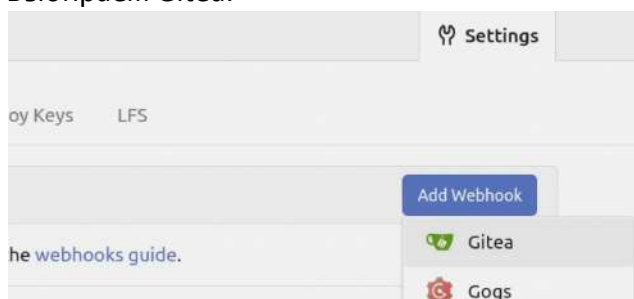
В Dashboard установилась созданная нами job.



Возвращаемся в Gitea для настройки webhooks. Переходим в настройки нашего репозитория, нажимаем «Add webhook».



Выбираем Gitea.





В Update Webhook пишем ссылку на Jenkins и его порт, указываем название job (ранее созданной в Jenkins).

### Update Webhook

Gitea will send POST requests with a specified content type to the target URL. Read more in the [webhooks guide](#).

**Target URL \***

**HTTP Method**

POST

**POST Content Type**

application/json

**Secret**

**Trigger On:**

☒ Push Events

☐ All Events

☐ Custom Events...

**Branch filter**

Для проверки запустим коммит.

```
user@debian:~/unix/test-repo$ echo "second file" > second.txt
user@debian:~/unix/test-repo$ git add second.txt
user@debian:~/unix/test-repo$ git commit -m "second"
[master 25583cc] second
 1 file changed, 1 insertion(+)
 create mode 100644 second.txt
user@debian:~/unix/test-repo$ git push
Username for 'http://localhost:3000': jenkins
Password for 'http://jenkins@localhost:3000':
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 289 bytes | 289.00 KiB/s, готово.
Всего 3 (изменения 0), повторно использовано 0 (изменения 0)
remote: . Processing 1 references
remote: Processed 1 references in total
To http://localhost:3000/demo-org/test-repo.git
   79e3214..25583cc master -> master
user@debian:~/unix/test-repo$
```

Идем в Dashboard Jenkins. Да, Jenkins увидел наш последний коммит.



All	+				
S	W	Name ↓	Last Success	Last Failure	Last D
		demo-job	3.7 sec - #2	N/A	0.3 sec

Icon: S M L

Legend

Atom feed for all

Atom feed for failures

## Руководство по использованию инструмента для создания резервных копий и Руководство по восстановлению состояния системы из резервной копии

В папке backup расположены два скрипта на языке bash. Для создания резервной копии и восстановления состояние по копии. Также приложен markdown-файл с описанием файлов.

### Backup.sh

```
#!/bin/bash
```

```
tar -cvpzf backup.tar.gz \  
  --exclude=./backup.tar.gz \  
  --exclude=./jenkins/plugins \  
  --one-file-system \  
  ../gitea ../jenkins ../mysql
```

Использование: `backup.sh`

\* Должен выполняться в собственном каталоге, так как использует относительные пути к файлам, которые он должен резервировать.

\* Должен выполняться с привилегиями root, так как некоторые файлы, например, ключи ssh, требуют прав root для чтения и архивирования.

### Restore.sh

```
#!/bin/bash
```

```
if [ $# -eq 0 ]; then  
  echo "You must supply a backup to restore from."  
  exit 0  
fi
```

```
tar xvf "$1" --directory ..
```

Использование: `restore.sh path/to/backup/file`

\* Должен выполняться в собственном каталоге, так как использует относительные пути к местам, где будут восстанавливаться данные.