

SYSTEMZ PLATFORM  
Инструкция по  
разворачиванию системы

---

## Оглавление

1	Аннотация.....	3
2	Установка системы .....	4
2.1	Требования к серверу.....	4
2.2	Разворачивание системы.....	4

# 1 АННОТАЦИЯ

---

Данная инструкция по разворачиванию системы **SYSTEMZ Platform** предназначена для системных администраторов.

Документ описывает требования к серверу установки и последовательность шагов по разворачиванию системы на сервере.

## 2 УСТАНОВКА СИСТЕМЫ

---

### 2.1 ТРЕБОВАНИЯ К СЕРВЕРУ

---

Установка системы SYSTEMZ Platform возможна на сервер со следующими характеристиками:

- Технические характеристики: 32 Gb Ram, 6 Core CPU, 100 Gb HDD;
- Операционная система: Windows, Linux (*далее в инструкции рассмотрен способ установки на операционной системе Centos 7*);
- Предустановленное ПО: jre8, nano.

### 2.2 РАЗВОРАЧИВАНИЕ СИСТЕМЫ

---

Для установки системы на сервер выполните следующие шаги:

#### 1. Установка дополнительного ПО

Из предоставленного дистрибутива платформы необходимо скопировать в каталог ***/opt/app*** следующие ***\*.jar*** файлы:

- admin-service.jar
- auth-service.jar
- bpm-service.jar
- config-server.jar
- discovery.jar
- entity-service.jar
- file-service.jar
- gateway.jar
- hystrix-service.jar
- modules-service.jar
- notification-service.jar
- swagger-service.jar
- user-service.jar
- zforms-service.jar
- zipkin.jar

#### 2. Установка PostgreSQL

В платформе используется PostgreSQL Pro. Для установки необходимо выполнить следующие действия:

1. `rpm -ivh http://repo.postgrespro.ru/pgpro-11/keys/postgrespro-std-11.centos.pro.yum-1.0-1.noarch.rpm`
2. `yum install postgrespro-std-11`
3. `rm -rf /var/lib/pgpro/std-11/data/postmaster.opts`
4. `echo "/opt/pgpro/std-11/bin/pg_ctl -D /var/lib/pgpro/std-11/data -l logfile start " >> /var/lib/pgpro/std-11/data/postmaster.opts`
5. `su - postgres`
6. `export PGDATA=/var/lib/pgpro/std-11/data`
7. `pg_ctl restart`

### 3. Установка MongoDB

Для установки MongoDB необходимо выполнить действия:

- a. Создать файл:

```
/etc/yum.repos.d/mongodb-org.repo
```

- b. Добавить в файл:

```
[mongodb-org-3.4]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.4/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.4.asc
```

- c. Выполнить установку:

```
yum install mongodb-org
```

### 4. Регистрация сервисов в операционной системе

Для регистрации сервиса в операционной системе необходимо:

- a. Создать файл запуска и указать ссылку на него в системном разделе ОС. Файлы запуска должны располагаться в каталоге ***/opt/app/run\_scripts/***.

Для создания пустых файлов выполните команды:

```
echo >> admin-service
echo >> auth-service
```

```

echo >> bpm-service
echo >> config-server
echo >> discovery
echo >> entity-service
echo >> file-service
echo >> gateway
echo >> hystrix-service
echo >> modules-service
echo >> notification-service
echo >> swagger-service
echo >> user-service
echo >> zforms-service
echo >> zipkin-service

```

### Пример содержимого файла для discovery:

```

#!/bin/sh

dir="/opt/app/"
cmd="java -Xms256m -Xmx768m -jar /opt/app/discovery.jar"

name=`basename $0`
pid_file="/opt/app/run/$name.pid"
stdout_log="/var/log/discovery/$name.log"
stderr_log="/var/log/discovery/$name.err"

get_pid() {
    cat "$pid_file"
}

is_running() {
    [ -f "$pid_file" ] && ps `get_pid` > /dev/null 2>&1
}

case "$1" in
    start)
        if is_running; then
            echo "Already started"
        else
            echo "Starting $name"
            cd "$dir"
            $cmd >> "$stdout_log" 2>> "$stderr_log" &

            echo $! > "$pid_file"
            if ! is_running; then
                echo "Unable to start, see $stdout_log and
                $stderr_log"
                exit 1
            fi
        fi
        ;;
    stop)
        if is_running; then
            echo -n "Stopping $name.."
            kill `get_pid`
            for i in {1..10}
            do

```

```

        if ! is_running; then
            break
        fi

        echo -n "."
        sleep 1
    done
    echo

    if is_running; then
        echo "Not stopped; may still be shutting down or
shutdown may have failed"
        exit 1
    else
        echo "Stopped"
        if [ -f "$pid_file" ]; then
            rm "$pid_file"
        fi
    fi
else
    echo "Not running"
fi
;;
restart)
$0 stop
if is_running; then
    echo "Unable to stop, will not attempt to start"
    exit 1
fi
$0 start
;;
status)
if is_running; then
    echo "Running"
else
    echo "Stopped"
    exit 1
fi
;;
*)
echo "Usage: $0 {start|stop|restart|status}"
exit 1
;;
esac

exit 0

```

- b. Сделать созданный файл запускаемым. Для этого выполните команду:

```
chmod +x /opt/app/run_scripts/discovery
```

- c. Выполнить регистрацию сервиса в ОС. Для этого создайте файлы в каталоге **/etc/systemd/system**

```

echo >> admin-service.service
echo >> auth-service.service
echo >> bpm-service.service
echo >> config-server.service

```

```
echo >> discovery.service
echo >> entity-service.service
echo >> file-service.service
echo >> gateway.service
echo >> hystrix-service.service
echo >> modules-service.service
echo >> notification-service.service
echo >> swagger-service.service
echo >> user-service
echo >> zforms-service
echo >> zipkin-service
```

Пример содержимого файла для discovery:

```
Description = discovery
After network.target = discovery.service
[Service]
Type = forking
Restart=always
RestartSec=1
SuccessExitStatus=143
ExecStart = /opt/app/run_scripts/discovery start
ExecStop = /opt/app/run_scripts/discovery stop
ExecReload = /opt/app/run_scripts/discovery restart
ExecStatus = /opt/app/run_scripts/discovery status
```

d. В завершении регистрации выполнить команды:

```
systemctl enable discovery
systemctl enable admin-service
systemctl enable auth-service
systemctl enable bpm-service
systemctl enable config-server
systemctl enable discovery
systemctl enable entity-service
systemctl enable file-service
systemctl enable gateway
systemctl enable hystrix-service
systemctl enable modules-service
systemctl enable notification-service
systemctl enable swagger-service
systemctl enable user-service
systemctl enable zforms-service
systemctl enable zipkin-service
```

## 5. Запуск сервисов

Сервисы необходимо запускать в строгом порядке с контролем успешного запуска каждого. Контроль запуска рассматривается в пункте 6. «Регистрация сервисов в *discovery*» настоящей инструкции.

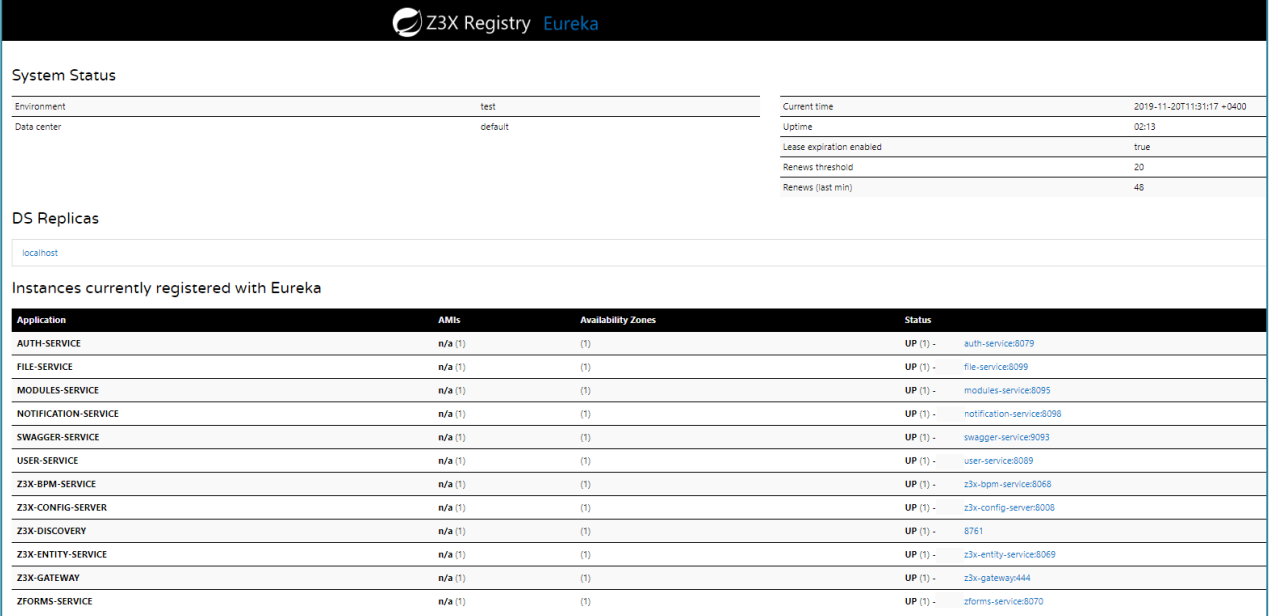


Для запуска выполните следующие команды:

```
service discovery-service start
service config-server start
service auth-service start
service modules-service start
service user-service start
service admin-service start
service entity-service start
service zforms-service start
service bpm-service start
service file-service start
service hystrix-service start
service notification-service start
service gateway start
service swagger start
```

## 6. Регистрация сервисов в discovery

Для проверки успешного запуска сервиса следует использовать веб-интерфейс Discovery (<http://localhost:8761/discovery>).



The screenshot shows the Z3X Registry Eureka web interface. It includes a 'System Status' section with various system parameters and a table of 'Instances currently registered with Eureka'.

Application	AMIs	Availability Zones	Status
AUTH-SERVICE	n/a (1)	(1)	UP (1) - auth-service:8079
FILE-SERVICE	n/a (1)	(1)	UP (1) - file-service:8099
MODULES-SERVICE	n/a (1)	(1)	UP (1) - modules-service:8095
NOTIFICATION-SERVICE	n/a (1)	(1)	UP (1) - notification-service:8098
SWAGGER-SERVICE	n/a (1)	(1)	UP (1) - swagger-service:9093
USER-SERVICE	n/a (1)	(1)	UP (1) - user-service:8089
Z3X-BPM-SERVICE	n/a (1)	(1)	UP (1) - z3x-bpm-service:8068
Z3X-CONFIG-SERVER	n/a (1)	(1)	UP (1) - z3x-config-server:8008
Z3X-DISCOVERY	n/a (1)	(1)	UP (1) - 8761
Z3X-ENTITY-SERVICE	n/a (1)	(1)	UP (1) - z3x-entity-service:8069
Z3X-GATEWAY	n/a (1)	(1)	UP (1) - z3x-gateway:444
ZFORMS-SERVICE	n/a (1)	(1)	UP (1) - zforms-service:8070

Рисунок 1 – Веб-интерфейс Discovery

После успешного запуска в таблице появится запись, а в колонке Application можно видеть запускаемый сервис (Рисунок 1).

## 7. Логирование

Файлы логирования платформы располагаются в каталоге **/var/log/**.

Для каждого сервиса необходимо создать одноименный каталог, в котором будут располагаться одноименные файлы с расширениями **\*.log** и **\*.err** для сбора общей

информации работы сервиса и ошибок соответственно. Соответствующие файлы будут созданы самим сервисом в момент запуска.

Для создания каталогов выполните команды:

```
mkdir /var/log/discovery
mkdir /var/log/admin-service
mkdir /var/log/auth-service
mkdir /var/log/bpm-service
mkdir /var/log/config-service
mkdir /var/log/entity-service
mkdir /var/log/file-service
mkdir /var/log/hystrix-service
mkdir /var/log/modules-service
mkdir /var/log/notification-service
mkdir /var/log/swagger-service
mkdir /var/log/user-service
mkdir /var/log/zforms-service
mkdir /var/log/zipkin-service
```