

1. Установка ПО Саунч на сервер	2
1.1 Установка ПО для работы сервера (Открытый контур)	2
1.1.1 Установка ПО с Носителя	2
1.1.2 Установка ПО если сервер имеет доступ в интернет	5
1.2 Установка ПО для работы сервера (Закрытый контур)	12
1.2.1 Установка ПО с Носителя	12

1. Установка ПО Саунч на сервер

1.1 Установка ПО для работы сервера (Открытый контур)

1.1.1 Установка ПО с Носителя

Для установки По Необходимо выполнить поочередно команды, либо запустить скрипт `pre_install_local.sh` командой `bash ./pre_install_local.sh` с правами суперпользователя
Команды для установки

Для начала необходимо скопировать папку `support` в папку `/home/support`
`sudo apt update`

Установка `nginx`

```
sudo dpkg -i /home/support/nginx.deb
sudo systemctl enable nginx
sudo service nginx status
sudo systemctl is-enabled nginx
ufw allow 'Nginx Full'
ufw allow ssh
ufw enable
```

Установка `nodejs`

```
sudo dpkg -i /home/support/nodejs.deb
```

Установка менеджера процессов `pm2`

```
sudo bash /home/support/pm2.sh
```

Перенос ПО

Копируем с носителя, то необходимо перенести папку `/etc/nginx` и папку `/var/www` с заменой файлов, так же если присутствует папка `/etc/ssl` её так же необходимо перенести

Далее выполнить команды

```
ln -s /etc/nginx/sites-available/service /etc/nginx/sites-enabled/service
service nginx reload
```

Установка `mongodb`

```
sudo dpkg -i /home/support/mongodb.deb
sudo systemctl enable --now mongod
service mongod status
```

Если папки `/etc/ssl` то необходимо создать `ssl` для подключения к базе

Генерация `ssl`

```
openssl genrsa -out /tmp/mongoCA.key 4096
openssl req -x509 -new -days 10000 -key /tmp/mongoCA.key -subj
'/C=RU/ST=Moscow/L=Moscow/O=${Название организации
кириллицей}/OU=Personal/emailAddress=${email адрес}CN=${ip адрес машины на
которой располагается база данных}' -out /tmp/mongoCA.crt
cat /tmp/mongoCA.key /tmp/mongoCA.crt > /etc/ssl/mongoCA.pem
rm /tmp/mongoCA.key /tmp/mongoCA.crt
```

```
openssl genrsa -out /tmp/server.key 4096
openssl req -new -key /tmp/server.key -subj '/C=RU/ST=Moscow/L=Moscow/O=${Название
организации кириллицей}/OU=Personal/emailAddress=${email адрес}/CN=${ip
адрес машины на которой располагается база данных}' -out /tmp/server.csr
openssl x509 -req -in /tmp/server.csr -CA /etc/ssl/mongoCA.pem -CAcreateserial -out
/tmp/server.crt -days 10000
cat /tmp/server.key /tmp/server.crt > /etc/ssl/server.pem
rm /tmp/server.key /tmp/server.crt /tmp/server.csr
openssl genrsa -out /tmp/client.key 4096
openssl req -new -key /tmp/client.key -subj '/C=RU/ST=Moscow/L=Moscow/O=${Название
организации кириллицей}/OU=Personal/emailAddress=${email адрес}/CN=${ip
адрес машины на которой располагается база данных}' -out /tmp/client.csr
openssl x509 -req -in /tmp/client.csr -CA /etc/ssl/mongoCA.pem -CAcreateserial -out
/tmp/client.crt -days 10000
cat /tmp/client.key /tmp/client.crt > /etc/ssl/client.pem
rm /tmp/client.key /tmp/client.crt /tmp/client.csr
```

Создание пользователя в БД

```
mongosh
use admin
db.createUser({ user: "saunch" , pwd: "33r3CM75KVTB5Uy38CdE2wJ1zNt", roles: ["read",
"readWrite", "dbAdmin", "userAdmin", "clusterAdmin", "readAnyDatabase",
"readWriteAnyDatabase", "userAdminAnyDatabase", "dbAdminAnyDatabase"]})
```

Обновление конфигурации mongoDB

```
nano /etc/mongod.conf
Необходимо заменить
net:
  port: 27017
  bindIp: 127.0.0.1,${ip адрес машины}
  tls:
    mode: requireTLS
    certificateKeyFile: /etc/ssl/server.pem
    CAFile: /etc/ssl/mongoCA.pem
    clusterFile: /etc/ssl/mongoCA.pem
    allowInvalidCertificates: true
    allowInvalidHostnames: true
security:
  authorization: enabled
  clusterAuthMode: x509
  clusterIpSourceAllowlist:
    - 127.0.0.1
```

Далее закрываем файл с сохранением

```
service mongod restart
service mongod status
```

При необходимости открываем доступ к базе для внешних подключений

```
ufw allow from ${ip адрес с которого будет подключение} to any port 27017
```

Устанавливаем доп ПО

```
sudo dpkg -i /home/support/zip.deb
```

Далее необходимо настроить БД из дампа выданного разработчиками ACLCO Саунч
Для этого необходимо скопировать файл дампа по пути /home/mongodump/saunch/
И запустить команду

```
mongorestore --ssl --sslAllowInvalidCertificates --sslAllowInvalidHostnames --sslPEMKeyFile  
/etc/ssl/server.pem --sslCAFile /etc/ssl/mongoCA.pem --authenticationDatabase admin -u  
saunch -p 33r3CM75KVTB5Uy38CdE2wJ1zNt --drop --verbose --host="127.0.0.1"  
--port=27017 -d saunch /home/mongodump/saunch/
```

запускаем внутренний сервер для доступа в админ часть

```
cd /var/www/saunch_internal_server  
pm2 start eco.config.js
```

Далее необходимо создать сертификат доступа для запуска серверов

Для этого необходимо перейти в браузере по ссылке [http://\\${ip}](http://${ip}) адрес машины на
которой располагается внутренняя админ часть}/

После входа необходимо перейти в раздел **Конфигурации Саунч**

Выбрать конфигурацию и в открывшемся окне

Создать сертификат открытый сервер

Создать сертификат внутренний сервер

Подписать сертификат открытого сервера закрытым сервером

Далее можно закрыть web приложение и перейти обратно в терминал

Далее необходимо перенести созданные сертификаты

```
cp /var/www/saunch_internal_server/ssl/ssl_servers/openServer/server_cert.pem  
/var/www/saunch_open_server/ssl/
```

```
cp /var/www/saunch_internal_server/ssl/ssl_servers/openServer/server_key.pem  
/var/www/saunch_open_server/ssl/
```

```
cp
```

```
/var/www/saunch_internal_server/ssl/ssl_servers/openServerSignInternalServer/open_sign_i  
nternal_cert.pem /var/www/saunch_open_server/ssl/
```

```
cp
```

```
/var/www/saunch_internal_server/ssl/ssl_servers/openServerSignInternalServer/open_sign_i  
nternal_key.pem /var/www/saunch_open_server/ssl/
```

Установка сервера логирования

```
sudo apt-get update -y
```

```
sudo dpkg -i /home/support/erlang.deb
```

```
sudo dpkg -i /home/support/rabbitmq-server.deb
```

```
sudo systemctl stop rabbitmq-server
```

```
sudo systemctl start rabbitmq-server
```

```
sudo systemctl status rabbitmq-server
```

для внешнего доступа необходимо добавить ip адрес

```
ufw allow from ${ip адрес с которого будет запрос} to any port 15672
```

```
rabbitmq-plugins list
```

```
rabbitmqctl list_vhosts
rabbitmq-plugins enable rabbitmq_management
```

```
rabbitmqctl add_user saunch VlcN6F1SXv3ivKq39L7GnH
rabbitmqctl set_user_tags saunch administrator
rabbitmqctl delete_user guest
rabbitmqctl list_users
rabbitmqctl add_vhost saunch-logs
rabbitmqctl add_vhost saunch-analytics
rabbitmqctl list_vhosts
rabbitmqctl delete_vhost /
rabbitmqctl list_vhosts
sudo rabbitmqctl set_permissions -p saunch-logs saunch ".*" ".*" ".*"
sudo rabbitmqctl set_permissions -p saunch-analytics saunch ".*" ".*" ".*"
sudo rabbitmqctl list_permissions -p saunch-logs
sudo rabbitmqctl enable_feature_flag all
```

```
далее запускаем все сервера
cd /var/www/saunch_open_server
pm2 start eco.config.js
cd /var/www/saunch_internal_server
pm2 start eco.config.js
cd /var/www/saunch_encryption_server
pm2 start eco.config.js
cd /var/www/saunch_logging_server
pm2 start eco.config.js
```

```
для сохранения настроек при перезагрузке выполним ЭТИМ команды
pm2 save
pm2 startup
```

1.1.2 Установка ПО если сервер имеет доступ в интернет

Для установки По Необходимо выполнить поочередно команды, либо запустить скрипт pre_install.sh командой `bash ./pre_install.sh` с правами суперпользователя

Команды для установки

Установка nginx

```
sudo apt update
sudo apt install nginx
sudo systemctl enable nginx
sudo service nginx status
sudo systemctl is-enabled nginx
ufw allow 'Nginx Full'
ufw allow ssh
ufw enable
```

Установка nodejs

```
wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
```

```
export NVM_DIR="$([ -z "${XDG_CONFIG_HOME-}" ] && printf %s "${HOME}/.nvm" || printf %s "${XDG_CONFIG_HOME}/nvm")"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
nvm ls
nvm ls-remote
nvm install --lts
```

Установка менеджера процессов pm2

```
npm install pm2@latest -g
после установки необходимо добавить конфигурацию в .profile
nano /root/.profile
В конец файла добавляем
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm
bash_completion
Перегружаем файл
source /root/.profile
```

Для оперативного получения обновлений есть возможность подключиться к основному репозиторию для этого необходимо сгенерировать ssh ключ

```
ssh-keygen -t ed25519 -C "saunch-server-{название организации кириллицей}"
```

```
tail /root/.ssh/id_ed25519.pub
```

команда tail выведет ключ в терминал, его необходимо передать разработчикам АЦЦО Саунч

Если по сервера предоставлено на носителе, то необходимо перенести папку /etc/nginx и папку /var/www с заменой файлов

Установка mongodb

```
curl -fsSL https://pgp.mongodb.com/server-6.0.asc | \sudo gpg -o
/etc/apt/trusted.gpg.d/mongodb-server-6.0.gpg \--dearmor
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
jammy/mongodb-org/6.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list
sudo apt update
apt install mongodb-org
sudo systemctl enable --now mongod
```

```
service mongod status
```

Генерация ssl

```
openssl genrsa -out /tmp/mongoCA.key 4096
openssl req -x509 -new -days 10000 -key /tmp/mongoCA.key -subj
'/C=RU/ST=Moscow/L=Moscow/O={Название организации кириллицей}/OU=Personal/emailAddress={email адрес}CN={ip адрес машины на которой располагается база данных}' -out /tmp/mongoCA.crt
cat /tmp/mongoCA.key /tmp/mongoCA.crt > /etc/ssl/mongoCA.pem
rm /tmp/mongoCA.key /tmp/mongoCA.crt
openssl genrsa -out /tmp/server.key 4096
```

```
openssl req -new -key /tmp/server.key -subj '/C=RU/ST=Moscow/L=Moscow/O=${Название
организации кириллицей}/OU=Personal/emailAddress=${email адрес}/CN=${ip
адрес машины на которой располагается база данных}' -out /tmp/server.csr
openssl x509 -req -in /tmp/server.csr -CA /etc/ssl/mongoCA.pem -CAcreateserial -out
/tmp/server.crt -days 10000
cat /tmp/server.key /tmp/server.crt > /etc/ssl/server.pem
rm /tmp/server.key /tmp/server.crt /tmp/server.csr
openssl genrsa -out /tmp/client.key 4096
openssl req -new -key /tmp/client.key -subj '/C=RU/ST=Moscow/L=Moscow/O=${Название
организации кириллицей}/OU=Personal/emailAddress=${email адрес}/CN=${ip
адрес машины на которой располагается база данных}' -out /tmp/client.csr
openssl x509 -req -in /tmp/client.csr -CA /etc/ssl/mongoCA.pem -CAcreateserial -out
/tmp/client.crt -days 10000
cat /tmp/client.key /tmp/client.crt > /etc/ssl/client.pem
rm /tmp/client.key /tmp/client.crt /tmp/client.csr
```

Создание пользователя в БД

```
mongosh
use admin
db.createUser({ user: "saunch" , pwd: "33r3CM75KVTB5Uy38CdE2wJ1zNt", roles: ["read",
"readWrite", "dbAdmin", "userAdmin", "clusterAdmin", "readAnyDatabase",
"readWriteAnyDatabase", "userAdminAnyDatabase", "dbAdminAnyDatabase"]})
```

Обновление конфигурации mongoDB

```
nano /etc/mongod.conf
Необходимо заменить
net:
  port: 27017
  bindIp: 127.0.0.1,${ip адрес машины}
  tls:
    mode: requireTLS
    certificateKeyFile: /etc/ssl/server.pem
    CAFile: /etc/ssl/mongoCA.pem
    clusterFile: /etc/ssl/mongoCA.pem
    allowInvalidCertificates: true
    allowInvalidHostnames: true
security:
  authorization: enabled
  clusterAuthMode: x509
  clusterIpSourceAllowlist:
    - 127.0.0.1
```

Далее закрываем файл с сохранением

```
service mongod restart
service mongod status
```

При необходимости открываем доступ к базе для внешних подключений
ufw allow from \${ip адрес с которого будет подключение} to any port 27017

Если ПО предоставлено не на носителе необходимо скачать актуальную версию с репозитория

```
cd /var/www/
```

```
git clone git@git.dev-serv.ru:saunch/saunch_open_server.git saunch_open_server
git clone git@git.dev-serv.ru:saunch/saunch_internal_server.git saunch_internal_server
git clone git@git.dev-serv.ru:saunch/saunch_internal_front.git saunch_internal_front
git clone git@git.dev-serv.ru:saunch/saunch_encryption_server.git
saunch_encryption_server
git clone git@git.dev-serv.ru:saunch/saunch_logging_server.git saunch_logging_server
```

далее необходимо установить зависимости

```
cd /var/www/saunch_open_server
npm i
cd /var/www/saunch_internal_server
npm i
cd /var/www/saunch_encryption_server
npm i
cd /var/www/saunch_logging_server
npm i
```

Если конфигурация nginx не предоставлена на носителе необходимо добавить

```
cd /etc/nginx/sites-available/
nano service
Добавляем в файл
include /etc/nginx/sites-available/*on;
Закрываем с сохранением
Добавляем симлинк
ln -s /etc/nginx/sites-available/service /etc/nginx/sites-enabled/service
```

```
nano saunch_internal_front-on
Добавляем в файл
```

```
upstream tile {
    server server оффлайн карт;
    keepalive 8;
}

server {
    listen 80;
    server_name {ip адрес машины};

    location ~ /\.well-known {
        root /var/www/html;
        allow all;
    }
    location / {
        try_files $uri $uri/ /index.html;
```



```

    root /var/www/saunch_internal_front/dist;
    index index.html;
}
location /tiles/ {
    root /var/www/tiles/;
    try_files $uri $uri/ @applicationMap;

    expires 1h;
    add_header Cache-Control "public";
    add_header Access-Control-Allow-Origin *;
    add_header Access-Control-Allow-Headers "*" always;
    add_header Access-Control-Allow-Methods "*" always;
    add_header Access-Control-Allow-Origin "*" always;

}
location @applicationMap {
    resolver      8.8.4.4 8.8.8.8 valid=300s;
    resolver_timeout 10s;
    proxy_set_header Host "${сервер оффлайн карт}";
    rewrite_log on;
    rewrite ^/tiles/(\d+)/(\d+)/(\d+.*)*$ /tile/$1/$2/$3 break;
    proxy_pass ${сервер оффлайн карт};
}
}

```

service nginx reload

Устанавливаем дополнительные пакеты для работы

apt install zip

Далее необходимо настроить БД из дампа выданного разработчиками АСЦО Саунч
 Для этого необходимо скопировать файл дампа по пути /home/mongodump/saunch/
 И запустить команду

```

mongorestore --ssl --sslAllowInvalidCertificates --sslAllowInvalidHostnames --sslPEMKeyFile
/etc/ssl/server.pem --sslCAFile /etc/ssl/mongoCA.pem --authenticationDatabase admin -u
saunch -p 33r3CM75KVTB5Uy38CdE2wJ1zNt --drop --verbose --host="127.0.0.1"
--port=27017 -d saunch /home/mongodump/saunch/

```

запускаем внутренний сервер для доступа в админ часть

```
cd /var/www/saunch_internal_server
```

```
pm2 start eco.config.js
```

Далее необходимо создать сертификат доступа для запуска серверов

Для этого необходимо перейти в браузере по ссылке [http://\\$ip](http://$ip) адрес машины на которой располагается внутренняя админ часть}/

После входа необходимо перейти в раздел **Конфигурации Саунч**

Выбрать конфигурацию и в открывшемся окне

Создать сертификат открытый сервер

Создать сертификат внутренний сервер

Подписать сертификат открытого сервера закрытым сервером
Далее можно закрыть web приложение и перейти обратно в терминал

Далее необходимо перенести созданные сертификаты

```
cp /var/www/saunch_internal_server/ssl/ssl_servers/openServer/server_cert.pem  
/var/www/saunch_open_server/ssl/
```

```
cp /var/www/saunch_internal_server/ssl/ssl_servers/openServer/server_key.pem  
/var/www/saunch_open_server/ssl/
```

```
cp
```

```
/var/www/saunch_internal_server/ssl/ssl_servers/openServerSignInternalServer/open_sign_i  
nternal_cert.pem /var/www/saunch_open_server/ssl/
```

```
cp
```

```
/var/www/saunch_internal_server/ssl/ssl_servers/openServerSignInternalServer/open_sign_i  
nternal_key.pem /var/www/saunch_open_server/ssl/
```

Установка сервера логирования

Все команды надо выполнить пошагово для установка RabbitMq или его обновления
установки erlang или его обновления и добавления пользователей хостов и очередей

```
sudo apt-get update -y
```

```
sudo apt-get install curl gnupg -y
```

```
sudo apt-get install apt-transport-https
```

```
sudo apt-get install curl gnupg apt-transport-https -y
```

```
curl -1sLf
```

```
"https://keys.openpgp.org/vks/v1/by-fingerprint/0A9AF2115F4687BD29803A206B73A36E60  
26DFCA" | sudo gpg --dearmor | sudo tee /usr/share/keyrings/com.rabbitmq.team.gpg >  
/dev/null
```

```
curl -1sLf
```

```
https://github.com/rabbitmq/signing-keys/releases/download/3.0/cloudsmith.rabbitmq-erlang.  
E495BB49CC4BBE5B.key | sudo gpg --dearmor | sudo tee  
/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg > /dev/null
```

```
curl -1sLf
```

```
https://github.com/rabbitmq/signing-keys/releases/download/3.0/cloudsmith.rabbitmq-server.  
9F4587F226208342.key | sudo gpg --dearmor | sudo tee  
/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg > /dev/null
```

```
sudo tee /etc/apt/sources.list.d/rabbitmq.list <<EOF
```

```
## Provides modern Erlang/OTP releases from a Cloudsmith mirror
```

```
##
```

```
deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg]
```

```
https://ppa1.rabbitmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu jammy main
```

```
deb-src [signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg]
```

```
https://ppa1.rabbitmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu jammy main
```

```
# another mirror for redundancy
```

```
deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg]
```

```
https://ppa2.rabbitmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu jammy main
```

```
deb-src [signed-by=/usr/share/keyrings/rabbitmq.E495BB49CC4BBE5B.gpg]
https://ppa2.rabbitmq.com/rabbitmq/rabbitmq-erlang/deb/ubuntu jammy main
```

```
## Provides RabbitMQ from a Cloudsmith mirror
```

```
##
```

```
deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg]
https://ppa1.rabbitmq.com/rabbitmq/rabbitmq-server/deb/ubuntu jammy main
deb-src [signed-by=/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg]
https://ppa1.rabbitmq.com/rabbitmq/rabbitmq-server/deb/ubuntu jammy main
```

```
# another mirror for redundancy
```

```
deb [arch=amd64 signed-by=/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg]
https://ppa2.rabbitmq.com/rabbitmq/rabbitmq-server/deb/ubuntu jammy main
deb-src [signed-by=/usr/share/keyrings/rabbitmq.9F4587F226208342.gpg]
https://ppa2.rabbitmq.com/rabbitmq/rabbitmq-server/deb/ubuntu jammy main
EOF
```

```
sudo apt-get update -y
```

```
sudo apt-get install -y erlang-base \
    erlang-asn1 erlang-crypto erlang-eldap erlang-ftp erlang-inets \
    erlang-mnesia erlang-os-mon erlang-parsetools erlang-public-key \
    erlang-runtime-tools erlang-snmp erlang-ssl \
    erlang-syntax-tools erlang-tftp erlang-tools erlang-xmerl
```

```
sudo apt-get install rabbitmq-server -y --fix-missing
```

```
sudo systemctl stop rabbitmq-server
sudo systemctl start rabbitmq-server
sudo systemctl status rabbitmq-server
```

для внешнего доступа необходимо добавить ip адрес
ufw allow from **\${ip адрес с которого будет запрос}** to any port 15672

```
rabbitmq-plugins list
rabbitmqctl list_vhosts
rabbitmq-plugins enable rabbitmq_management
```

```
rabbitmqctl add_user saunch VlcN6F1SXv3ivKq39L7GnH
rabbitmqctl set_user_tags saunch administrator
rabbitmqctl delete_user guest
rabbitmqctl list_users
rabbitmqctl add_vhost saunch-logs
rabbitmqctl add_vhost saunch-analytics
rabbitmqctl list_vhosts
rabbitmqctl delete_vhost /
rabbitmqctl list_vhosts
```

```
sudo rabbitmqctl set_permissions -p saunch-logs saunch ".*" ".*" ".*"  
sudo rabbitmqctl set_permissions -p saunch-analytics saunch ".*" ".*" ".*"  
sudo rabbitmqctl list_permissions -p saunch-logs  
sudo rabbitmqctl enable_feature_flag all
```

далее запускаем все сервера

```
cd /var/www/saunch_open_server  
pm2 start eco.config.js  
cd /var/www/saunch_internal_server  
pm2 start eco.config.js  
cd /var/www/saunch_encryption_server  
pm2 start eco.config.js  
cd /var/www/saunch_logging_server  
pm2 start eco.config.js
```

для сохранения настроек при перезагрузке выполним эти команды

```
pm2 save  
pm2 startup
```

1.2 Установка ПО для работы сервера (Закрытый контур)

1.2.1 Установка ПО с Носителя

Для установки ПО необходимо выполнить поочередно команды, либо запустить скрипт `pre_install_local.sh` командой `bash ./pre_install_local.sh` с правами суперпользователя

Команды для установки

Для начала необходимо скопировать папку `support` в папку `/home/support`

```
sudo apt update
```

Установка nginx

```
sudo dpkg -i /home/support/nginx.deb
```

```
sudo systemctl enable nginx
```

```
sudo service nginx status
```

```
sudo systemctl is-enabled nginx
```

```
ufw allow 'Nginx Full'
```

```
ufw allow ssh
```

```
ufw enable
```

Установка nodejs

```
sudo dpkg -i /home/support/nodejs.deb
```

Установка менеджера процессов pm2

```
sudo bash /home/support/pm2.sh
```

Перенос ПО

Копируем с носителя, то необходимо перенести папку `/etc/nginx` и папку `/var/www` с заменой файлов, так же папку `/etc/ssl`

Далее выполнить команды

```
In -s /etc/nginx/sites-available/service /etc/nginx/sites-enabled/service
service nginx reload
```

Установка mongodb

```
sudo dpkg -i /home/support/mongodb.deb
sudo systemctl enable --now mongod
service mongod status
```

Далее необходимо перенести конфигурацию mongod.conf /etc/mongod.conf
Командой cp

Далее необходимо настроить БД из дампа выданного разработчиками АСЦО Саунч
Для этого необходимо скопировать файл дампа по пути /home/mongodump/saunch/
И запустить команду
mongorestore --ssl --sslAllowInvalidCertificates --sslAllowInvalidHostnames --sslPEMKeyFile
/etc/ssl/server.pem --sslCAFile /etc/ssl/mongoCA.pem --authenticationDatabase admin -u
saunch -p 33r3CM75KVTB5Uy38CdE2wJ1zNt --drop --verbose --host="127.0.0.1"
--port=27017 -d saunch /home/mongodump/saunch/

запускаем внутренний сервер для доступа в админ часть
cd /var/www/saunch_internal_server
pm2 start eco.config.js

Далее необходимо создать сертификат доступа для запуска серверов
Для этого необходимо перейти в браузере по ссылке [http://\\$ip](http://$ip) адрес машины на
которой располагается внутренняя админ часть}/

После входа необходимо перейти в раздел **Конфигурации Саунч**

Выбрать конфигурацию и в открывшемся окне

Создать сертификат открытый сервер

Создать сертификат внутренний сервер

Подписать сертификат открытого сервера закрытым сервером

Далее можно закрыть web приложение и перейти обратно в терминал

Далее необходимо перенести созданные сертификаты

```
cp /var/www/saunch_internal_server/ssl/ssl_servers/openServer/server_cert.pem
```

```
/var/www/saunch_open_server/ssl/
```

```
cp /var/www/saunch_internal_server/ssl/ssl_servers/openServer/server_key.pem
```

```
/var/www/saunch_open_server/ssl/
```

```
cp
```

```
/var/www/saunch_internal_server/ssl/ssl_servers/openServerSignInternalServer/open_sign_i
```

```
nternal_cert.pem /var/www/saunch_open_server/ssl/
```

```
cp
```

```
/var/www/saunch_internal_server/ssl/ssl_servers/openServerSignInternalServer/open_sign_i
```

```
nternal_key.pem /var/www/saunch_open_server/ssl/
```

Установка сервера логирования

```
sudo apt-get update -y
```

```
sudo dpkg -i /home/support/erlang.deb
sudo dpkg -i /home/support/rabbitmq-server.deb
sudo systemctl stop rabbitmq-server
sudo systemctl start rabbitmq-server
sudo systemctl status rabbitmq-server
```

для внешнего доступа необходимо добавить ip адрес
ufw allow from **{ip адрес с которого будет запрос}** to any port 15672

```
rabbitmq-plugins list
rabbitmqctl list_vhosts
rabbitmq-plugins enable rabbitmq_management
```

```
rabbitmqctl add_user saunch VlcN6F1SXv3ivKq39L7GnH
rabbitmqctl set_user_tags saunch administrator
rabbitmqctl delete_user guest
rabbitmqctl list_users
rabbitmqctl add_vhost saunch-logs
rabbitmqctl add_vhost saunch-analytics
rabbitmqctl list_vhosts
rabbitmqctl delete_vhost /
rabbitmqctl list_vhosts
sudo rabbitmqctl set_permissions -p saunch-logs saunch ".*" ".*" ".*"
sudo rabbitmqctl set_permissions -p saunch-analytics saunch ".*" ".*" ".*"
sudo rabbitmqctl list_permissions -p saunch-logs
sudo rabbitmqctl enable_feature_flag all
```

далее запускаем все сервера

```
cd /var/www/saunch_open_server
pm2 start eco.config.js
cd /var/www/saunch_internal_server
pm2 start eco.config.js
cd /var/www/saunch_encryption_server
pm2 start eco.config.js
cd /var/www/saunch_logging_server
pm2 start eco.config.js
```

для сохранения настроек при перезагрузке выполним эти команды

```
pm2 save
pm2 startup
```