v1.27

1.27

Eltex SC - . OS Ubuntu 20.

, .

*:

- — 1;
- — i5 3,0 ;
- — 8 ;
- — 1000 ;
- (/) — 2000 IOPS.

.

Eltex SC

Eltex SC 1.27 Ubuntu 20.04. Eltex SC (1.26) : Eltex SC.

1.19.4 deb- _.

1.25 eltex-notification-gw(ngw) **notification.properties.j2.** **vars/default.yml**

Eltex SC , , !

Ansible . Ansible .

Ansible — , Python . , Eltex SC.

:

1. Ansible Ubuntu 20.04.

_:

Ansible

```
apt update
apt install --install-recommends linux-generic-hwe-20.04-edge
apt install software-properties-common
add-apt-repository --yes --update ppa:ansible/ansible
apt install ansible
```

Ansible .

2. (**v2.9**):

Ansible

```
ansible --version
```

3. Ansible .

:

```
ansible-galaxy collection install community.general
ansible-galaxy collection install community.crypto
ansible-galaxy collection install community.docker
```

4. .

.
tar.gz, **/etc** **root**.

:

```
tar -C /etc -xvf ansible-iot-1.27.tar.gz
```

.

(/) Ansible **/etc/ansible-iot-1.27**

5. **/etc/ansible-iot-1.27/inventory**

, **nano**. **root** **ansible_sudo_pass**:

root **rootpasswd**

.

:

inventory

```
[iot]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
[elk]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
[monitoring]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
```

6. Eltex SC.

/etc/ansible-iot-1.27/vars/default.yml

, **nano**. **IP-** **server_name:**

vars/default.yml

```
---
#   .
iot:
  #  (IP-) ,       IoT.
  #   'localhost',     .
  #  !!!  'serverName'     (IP-),     .
  #   'localhost',      'localhost'!
  serverName: "my.test.server"
  #    ,     .
  installDir: /storage/iot

#     (Elasticsearch + Logstash + Kibana).
elk:
  #      appender,    logstash.
  #    ,  ELK     ;
  #   .
  enable: false
  #  (IP-) ,     ELK.
  #     'iot.serverName',       (   ).
  #         [iot]  [monitoring]  .
  serverName: "{{ iot.serverName }}"
  #     .
  installDir: /storage/elk

#     (Prometheus + Grafana).
monitoring:
  #  (IP-) ,       (Prometheus + Grafana).
  #     'iot.serverName',       (   ).
  #         [iot]  [elk]  .
  serverName: "{{ iot.serverName }}"
  #     .
  installDir: /storage/monitoring
```

vars/default.yml

```
#  MongoDB.
mongodb:
  #  MongoDB.   ,   ,    `4`.
  version: 5
  external:
    #    true,    MongoDB.
    # !!! MongoDB   ,       'addr'  'port'.
    enable: false
    #   MongoDB.
    addr: "{{ iot.serverName }}"
    #   MongoDB.
    port: 27017

#  WEB.
web:
  #  (IP-) ,     WEB.
  #     'iot.serverName',       (   ).
  serverName: "{{ iot.serverName }}"
  #  HTTP,       WEB.
  httpPort: 80
  #  HTTPS,       WEB.
  httpsPort: 443
  #      HTTP   HTTPS
  redirectHttpToHttps: false
  certbot:
    #   certbot    Let's Encrypt.
    enable: false
    # Email  .         Let's Encrypt.
    email: test@email.com

#    email (NGW).
mail:
  smtp:
    submitter: test@email.com
    password: "password"
    senderPrefix: " Eltex-SC"
    auth: "true"
    host: email.com
    port: 587

#   IoT core.
core:
  #    IoT Core.
  logLevel: INFO

  #     zway-.
  ctlGate:
    port: 8070
    tcpPort: 8069
    sslPort: 8072
```

vars/default.yml

```
#  API .
api:
  port: 8071
  sslPort: 8073

#   : easy, medium, hard
captchaLevel: "easy"

server:
  #    HTTPS     WEB ('true'  ,     ,
  #   'web.httpsPort').    'false',   HTTP  ,   'web.httpPort'.
  useHttpsForUi: true
  #    HTTPS        (, ).
  useHttpsForApi: false
  #    HTTPS        .
  useHttpsForCameraLinks: true
  #    'web.serverName'  'iot.serverName'  'web.httpPort'/'web.httpsPort'
  #  'core.api.port'/'core.api.sslPort'     API.
  useUiProxyForApi: false

#     .
selfRegistration:
  allow: true
  allowDemo: true
  allowSocialNetworks: false

#
video:
  #  Flussonic.
  flussonic:
    url: ""
    apiKey: ""
    operatorId: ""
    adminLogin: ""
  #   eltex
  eltex_server:
    url: ""
    #   true,   ,   flussonic   flussonic.url
    #  eltex.   1.28          .
    hls_player: false

#    ( ).
clientRegistrations:
  google:
    clientId: "GoogleClientIdChangeMe"
    clientSecret: "GoogleClientSecretChangeMe"
```

vars/default.yml

```
#         ().    .
yandexSkill:
  #   Basic Authentication.
  clientId: "YandexClientIdChangeMe"
  password: "PasswordChangeMe"
  # Id ,      .
  skillId: ""
  # OAuth-,      .
  oauthToken: ""

#         .    .
sberSkill:
  #   Basic Authentication.
  clientId: "SberClientIdChangeMe"
  password: "PasswordChangeMe"
  # Bearer-,      .
  bearerToken: ""

#     Mail.ru    .    /.
marusyaSkill:
  #   Basic Authentication.
  clientId: "MarusyaClientIdChangeMe"
  password: "PasswordChangeMe"
  # App ID,     VK  .
  appId: "MarusyaAppIdChangeMe"
  # OAuth-,      .
  oauthToken: ""
```

, , e-mail. :

mail:

smtp:

submitter — e-mail;

password — e-mail;

auth — smtp ( );

senderPrefix — ;

host — smtp-;

port — smtp- .

vars/service_parameters.yml :

/vars/service_parameters.yml

```
---
#  .
release: 1.27

#   docker registry,  docker-  .
registry: hub.eltex-co.ru

#     (   restart_*.yml).
#   ,       .
services:

#       .     ""
#     .        IoT  ansible,
#             .
withDistroPreparingStep: true

# ,      (    ).
containerNameSuffix: ""

# ,      docker (     docker).
networkNameSuffix: ""

#     .   :
# warnThreshold > criticalThreshold > 0,      .
diskUsage:
  #    ( %),          WARN
  # (    WARN  ERROR).
  #  criticalThreshold.
  warnThreshold: 20
  #    ( %),          ERROR
  # (     ERROR).    .
  criticalThreshold: 10
```

/vars/service_parameters.yml

```
#   IoT ( docker-compose),   .
# 'enable' -      docker-compose.yml.
# 'port.map' -      .
# 'port.export' -          .
# 'db.name' -  ,   ( ).
iotServices:
  db:
    port:
      map: 27017
      export: false
  broker:
    enable: true
    external:
      port:
        map: 8883
    internal:
      port:
        map: 8083
        export: false
    db:
      name: iot-broker
  olapservice:
    enable: true
    port:
      map: 8023
      export: false
    db:
      name: iotcore
      port:
        map: 8123
        export: false
  ngw:
    port:
      map: 8040
      export: false
    db:
      name: notification-gw
      user: javauser
      password: javapassword
      port:
        map: 3306
        export: false
  captcha:
    enable: true
    port:
      map: 8088
      export: false
    caseSensitive: true
    allowedSizes:
      - "312x45"
      - "270x40"
    instance: "captcha:8088"
    proportion: 100
```

/vars/service_parameters.yml

```
zscaptcha:
  enable: false
  port:
    map: 8089
    export: false
  caseSensitive: true
  instance: "zs-captcha:8089"
  proportion: 0
core:
  #    ,    core.
  enable: true
web:
  #    WEB,    web.
  enable: true
zwayproxy:
  enable: false
  port:
    map: 8070
    export: true
  sslPort:
    map: 8072
    export: true
  rabbit:
    port:
      map: 5672
      export: false
    queue:
      #         .
      platform: platform
    exchange:
      #  -.
      proxy: zway-proxy-topic-exchange
    #
    platformConsumers:
      #        maxCount.
      count: 1
      #     .
      maxCount: 1
    #     zwayProxy service
    proxyConsumers:
      #    .        maxCount.
      count: 1
      #     .
      maxCount: 1
```

/vars/service_parameters.yml

```
#   ELK ( docker-compose),   .
# 'port.map' -      .
# 'port.export' -          .
elkServices:
  elasticsearch:
    rest:
      port:
        map: 9200
    nodes:
      port:
        map: 9300
  logstash:
    port:
      map: 5001
    api:
      port:
        map: 9600
  kibana:
    port:
      map: 5601

#    ( docker-compose),   .
# 'port.map' -      .
# 'port.export' -          .
monitoringServices:
  prometheus:
    port:
      map: 9090
  grafana:
    port:
      map: 3000
  nginxExporter:
    enable: true
    port:
      map: 9113

coreInternal:
  #        .
  testdata:
    enable: false
  #    swagger ( API ).
  swagger:
    enable: false
  #       MQTT Broker.
  mqttbroker:
    enable: true
  #       OlapService.
  olapservice:
    enable: true
```

/vars/service_parameters.yml

```
#    MongoDB,  .
core:
  db:
    name: iot-core
fs:
  db:
    name: iot-fs
licenses:
  db:
    name: iot-licenses
events:
  db:
    name: iot-events
mjollnir:
  # URL  Mjollnir.
  url: "http://smart.eltex-co.ru:8078/api/v1"
```

7. :

```
cd /etc/ansible-iot-1.27
ansible-playbook install_iot.yml
```


Eltex SC c MongoDB, mongodb.org APT (, **/etc/apt/sources.list.d/mongodb-org-4.4.list**).

8. :

```
docker ps
```

docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
25c08d17a4ae	hub.eltex-co.ru/iot-clickhouse-backup:1.27	/entrypoint.sh /ent…	2 minutes ago	Up 2 minutes		iot-olapservice-db-backup
36c21b863cc9	hub.eltex-co.ru/iot-double-web:1.27	/docker-entrypoint.…	2 minutes ago	Up 2 minutes	0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp	iot-double-web
01fe2697e5ff	hub.eltex-co.ru/iot-core:1.27	java -Dspring.profi…	2 minutes ago	Up 2 minutes	.0.0.0:8069-8073->8069-8073/tcp, :::8069-8073->8069-8073/tcp	iot-core
07d1f93831bd	hub.eltex-co.ru/iot-mqttbroker-mongo:1.27	java -cp @/app /jib-…	2 minutes ago	Up 2 minutes	0.0.0.0:8883->8883/tcp, :::8883->8883/tcp	iot-mqtt-broker
d1c736dc27d0	hub.eltex-co.ru/eltex-ngw:1.27-602	/usr/sbin /ngw_start…	2 minutes ago	Up 2 minutes		iot-ngw-sc
228d41c96cba	hub.eltex-co.ru/iot-olapservice:1.27	java -cp @/app /jib-…	2 minutes ago	Up 2 minutes		iot-olapservice
e8e2899f2c8d	hub.eltex-co.ru/iot-captcha:1.27	java -jar LibreCapt…	2 minutes ago	Up 2 minutes	8888/tcp	iot-captcha
57c02941cc4f	hub.eltex-co.ru/iot-mongo5:1.27	/entrypoint.sh	2 minutes ago	Up 2 minutes	0.0.0.0:27017->27017/tcp, :::27017->27017/tcp	iot-mongo
7c3d8d5c4137	hub.eltex-co.ru/iot-mysql:1.27	docker-entrypoint.s…	2 minutes ago	Up 2 minutes	3306/tcp, 33060/tcp	iot-iot-mysql
9009dd4cd675	hub.eltex-co.ru/iot-olapservice:1.27	java -cp @/app /jib-…	2 minutes ago	Up 2 minutes	0.0.0.0:8023->8023/tcp, :::8023->8023/tcp	iot-olapservice

6c61b34c3a41	hub.eltex-co.ru/iot-clickhouse-server:1.27	/entrypoint.sh	2 minutes ago	Up 2 minutes	9000/tcp, 0.0.0.0:8123->8123/tcp, :::8123->8123/tcp, 9009/tcp	iot-olapservice-db

: http://[Eltex SC]

server_name /etc/ansible-iot-1.27/vars/default.yml

API .

9. .

.

, .

Eltex SC MongoDB

MongoDB, .. Eltex SC / mongodb:

mongodb.org . deb- (,).

1. MongoDB (**4.4**):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install -y software-properties-common gnupg build-essential net-tools dkms
wget https://www.mongodb.org/static/pgp/server-4.4.asc
sudo apt-key add server-4.4.asc
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-4.4.list
sudo apt-get update
sudo apt-get install -y mongodb-org
sudo service mongod start
sudo service mongod status
```

2. **/etc/mongod.conf net** :

```
port: 27017
bindIp: 0.0.0.0
```

mongod:

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

3. Ansible :

```
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
sudo ansible-galaxy collection install community.general
sudo ansible-galaxy collection install community.crypto
sudo ansible-galaxy collection install community.docker
```

4. :

```
tar -C /etc -xvf ansible-iot-1.27.tar.gz
```

. (/) Ansible **/etc/ansible-iot-1.27.**

5. **/etc/ansible-iot-1.27/inventory.**

, **nano**. **root** **ansible_sudo_pass**:

> **root** **rootpasswd**
>
> .

inventory

```
[iot]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
[elk]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
[monitoring]
localhost   ansible_connection=local    ansible_sudo_pass=rootpasswd
```

6. , . **/etc/ansible-iot-1.27/vars/default.yml** **/etc/ansible-iot-1.27/vars/service_parameters.yml.**

default.yml , **nano**. IP- **server_name.**

enable true, '**addr**' '**port**'.

vars/default.yml

```
mongodb:
  #  MongoDB.    ,   ,   `4`
  version: 5
  external:
    #    true,    MongoDB
    # !!! MongoDB   ,       'addr'  'port'
    enable: false
    #   MongoDB
    addr: "{{ iot.serverName }}"
    #   MongoDB
    port: 27017
```

8. :

```
cd /etc/ansible-iot-1.27
sudo ansible-playbook install_iot.yml
```

: http://[Eltex SC]

server_name /etc/ansible-iot-1.27/vars/default.yml

9. .

.

, .

:

```
ansible-playbook restart_iot.yml --extra-vars '{"services":["web", "core", "broker", "olapservice"]}'
```

, ("web", "core", "broker", "olapservice"). "services" --extra-vars, .
:

```
ansible-playbook stop_iot.yml --extra-vars '{"services":["web", "core", "broker", "olapservice"]}'
```

:

```
ansible-playbook update_iot.yml
```

:

```
ansible-playbook install_iot.yml
```

/etc/ansible-iot-1.27/templates : **/etc/ansible-iot-1.27/templates/iot/default-for-docker.yml.j2** -: **/etc/ansible-iot-1.27/templates/iot/web/base_config**

: **/storage/iot/core/var/log/eltex-sc/server.log**

. : server-YYYY-MM-DD.NN.log, YYYY-MM-DD — , NN — .

install_iot.yml:

ansible-playbook install_iot.yml

/etc/ansible-iot-1.27/vars/default.yml . , **/etc/ansible-iot-1.27/vars/default.yml,** , .

default-for-docker.yml.j2:

```
#    : , , ,
testData:
  environment: {{ 'true' if coreInternal.testdata.enable else 'false' }}

fileStorage:
  path: "/var/lib/eltex-sc/files"

controllerGateTCP:
  port: {{ core.ctlGate.tcpPort }}

controllerGate:
  port: {{ core.ctlGate.port }}

controllerGateSecurity:
  port: {{ core.ctlGate.sslPort }}
  key: "/etc/ssl/private/eltex-sc-ctl-gate.key"
  crt: "/etc/ssl/certs/eltex-sc-ctl-gate.crt"

api:
  port: {{ core.api.port }}
  sslPort: {{ core.api.sslPort }}

ui:
  serverName: "{{ web.serverName }}"
{% if web.httpPort != 80 or web.httpsPort != 443 %}
  #  UI      ,      UI
  #  .    port/sslPort      URL  UI (WEB)
{% if web.httpPort != 80 %}
  port: {{ web.httpPort }}
{% endif %}
{% if web.httpsPort != 443 %}
  sslPort: {{ web.httpsPort }}
{% endif %}
{% endif %}
```

```
mqttBroker:
  enabled: {{ 'true' if coreInternal.mqttbroker.enable else 'false' }}
  #  MQTT ,
  host: "broker"
  #  MQTT ,       MQTT
  port: 8883
  #       REST API,
  apiPort: 8083
  #  URL    MQTT ,  MQTT
  remoteAccessURL: "{{ iot.serverName }}:{{ iotServices.broker.external.port.map }}"
  #         "Offline" mqtt
  offlineTimeoutSec: 300

olapservice:
  enabled: {{ 'true' if coreInternal.olapservice.enable else 'false' }}
  host: "olapservice"
  port: 8023

zwayproxy:
  enabled: {{ 'true' if iotServices.zwayproxy.enable else 'false' }}
  manager:
    replicaCheckPeriodSeconds: 5
    replicaDeadPeriodSeconds: 15
    connectAliveWaitPeriodSeconds: 15
  rabbit:
    queue:
      #         .
      platform: {{ iotServices.zwayproxy.rabbit.queue.platform }}
    exchange:
      #  -.
      proxy: {{ iotServices.zwayproxy.rabbit.exchange.proxy }}
    consumers:
      #    .        maxCount.
      count: {{ iotServices.zwayproxy.rabbit.platformConsumers.count }}
      #     .
      maxCount: {{ iotServices.zwayproxy.rabbit.platformConsumers.maxCount }}

captcha:
  #    (Libre CAPTCHA, Zero Storage Captcha)
  instances: {{ captcha_instances | join(', ') }}
  proportions: {{ captcha_proportions | join(', ') }}
  level: "{{ core.captchaLevel }}"
```

```
#      UI(WEB)  API
server:
  #   ,
  name: "{{ iot.serverName }}"
  #   https://     UI
  useHttpsForUi: {{ 'true' if core.server.useHttpsForUi else 'false' }}
  #   https://     API
  useHttpsForApi: {{ 'true' if core.server.useHttpsForApi else 'false' }}
  #   https://
  useHttpsForCameraLinks:  {{ 'true' if core.server.useHttpsForCameraLinks else 'false' }}
  #  ui.serverName  server.name  ui.port/ui.sslPort  api.port/api.sslPort
  #     API
  useUiProxyForApi: {{ 'true' if core.server.useUiProxyForApi else 'false' }}
  oauth2:
    #   access-  .     5
    #   ,  authorization server    60   ,
    # . ..    : accessTokenTimeToLive + 60
    accessTokenTimeToLive: 3600
  #  jetty
  jetty:
    #
    connection-idle-timeout: 120000ms
    max-http-form-post-size: 200000B
    #
    threads:
      acceptors: -1
      selectors: -1
      #
      idle-timeout: 120000ms
      #
      min: 32
      #
      max: 256
      #
      max-queue-capacity: 32768

selfRegistration:
  allow: {{ 'true' if core.selfRegistration.allow else 'false' }}
  allowDemo: {{ 'true' if core.selfRegistration.allowDemo else 'false' }}
  allowSocialNetworks:  {{ 'true' if core.selfRegistration.allowSocialNetworks else 'false' }}

electricMeterScheduler:
  cron: "0 0/30 * * * ?"

electricMeterArchiveScheduler:
  cron: "0 0 17 * * ?"

batch:
  maxPoolSize: 2
  corePoolSize: 1
```

```
services:
  ngw:
    host: "ngw-sc"
    port: 8040
  alarmService:
    enabled: false
  diskUsage:
    warnThreshold: {{ diskUsage.warnThreshold }}
    criticalThreshold: {{ diskUsage.criticalThreshold }}
  loginInfo:
    #     loginInfo  ,
    ttl: 180
    #     ( )
    #     loginInfo (    )
    activityTimeLimit: 180
  user:
    #     ( )
    #
    allowedInactivePeriod: 180
  push:
    firebase:
      enabled: true
    apns:
      enabled: true

languageInterface:
  language: "ru"

# Hazelcast instance configuration -> Move it to separate microservice -> Use hz-client here
hazelcast:
  instanceName: "iot-core-hz-instance{{ containerNameSuffix }}"
  clusterName: "iot-core{{ containerNameSuffix }}"
  network:
    port: "5705"
#    member: "127.0.0.1:5701"

mjollnir:
  sync-period: "0 0 23 1/1 * ?"
  login: "platform"
  password: "platform"
  url: "{{ coreInternal.mjollnir.url }}"

#    iot-core
iot-core:
  host: "{{ mongodb_addr }}"
  port: {{ mongodb_port }}
  user: ""
  password: ""
  database: "{{ coreInternal.core.db.name }}"
```

```
#     iot-fs
file-storage:
  host: "{{ mongodb_addr }}"
  port: {{ mongodb_port }}
  user: ""
  password: ""
  database: "{{ coreInternal.fs.db.name }}"

#
license-storage:
  host: "{{ mongodb_addr }}"
  port: {{ mongodb_port }}
  user: ""
  password: ""
  database: "{{ coreInternal.licenses.db.name }}"

#     eventlog
eventlog:
  host: "{{ mongodb_addr }}"
  port: {{ mongodb_port }}
  user: ""
  password: ""
  database: "{{ coreInternal.events.db.name }}"

monitoring-system:
  countAttempt: 3
  delayAttempt: 0

spring:
  security:
    oauth2:
      client:
        registration:
          google:
            client-id: "{{ core.clientRegistrations.google.clientId }}"
            client-secret: "{{ core.clientRegistrations.google.clientSecret }}"
  kafka:
    enabled: false
    producer:
      bootstrap-servers: "localhost:9092"
      key-serializer: org.apache.kafka.common.serialization.StringSerializer
      value-serializer: org.springframework.kafka.support.serializer.JsonSerializer
  rabbitmq:
    host: zwayproxy-rabbitmq
    port: 5672
```

```
#   ,      android. ApiKey  .
video:
  flussonic:
    url: "{{ core.video.flussonic.url }}"
    apiKey: "{{ core.video.flussonic.apiKey }}"
    operator_id: "{{ core.video.flussonic.operatorId }}"
    admin_login: "{{ core.video.flussonic.adminLogin }}"
  ivideon:
    # ip  ivideon   ,
    ip_for_catch_event: ""
  eltex_server:
    url: "{{ core.video.eltex_server.url }}"
    hls_player: "{{ core.video.eltex_server.hls_player }}"

#    SSL
ssl:
  key: "/etc/ssl/private/eltex-sc-api.key"
  crt: "/etc/ssl/certs/eltex-sc-api.crt"

proxy:
  enabled: false
  host: ""
  port: 8050

billing:
  cron: "0 0 0 * * ?"
  ftp:
    host: "127.0.0.1"
    port: 21
    login: "user"
    password: "password"
    workdir: "test"

notifications:
  duplicate_delay_sec: 0
  antispam_time_sec: 0
  delay_push_time_millisec: 250

guard:
  deviceRequestDelay: 15000

#   swagger.    key -     swagger.
springdoc:
  swagger-ui:
    enabled: {{ 'true' if coreInternal.swagger.enable else 'false' }}
    key: ""

controller:
  delay_answer_from_control_millisec: 12000
  number_of_ping_threads: 1
  zway:
    logStringsLimit: 50
```

```
oauth2:
  clientParameters:
    web-client:
      type: WEB
      password: password
      clientAuthenticationMethod: client_secret_basic
      authorizationGrantTypes:
        - authorization_code
        - password
        - refresh_token
{% if web.httpsPort != 443 %}
      redirectUri: "https://{{ web.serverName }}:{{ web.httpsPort }}/ng/login"
{% else %}
      redirectUri: "https://{{ web.serverName }}/ng/login"
{% endif %}
      allowAdminLogin: true
      allowRevokeAll: true
    android-client:
      type: ANDROID
      password: password
      clientAuthenticationMethod: client_secret_basic
      authorizationGrantTypes:
        - authorization_code
        - password
        - refresh_token
      redirectUri: "https://eltex.iot.app/authorization"
      allowRevokeAll: true
    ios-client:
      type: IOS
      password: password
      clientAuthenticationMethod: client_secret_basic
      authorizationGrantTypes:
        - authorization_code
        - password
        - refresh_token
      scopes:
        - read
        - write
        - trust
      redirectUri: "https://eltex.iot.app/authorization"
      allowRevokeAll: true
    {{ core.yandexSkill.clientId }}:
      type: YANDEX
      password: "{{ core.yandexSkill.password }}"
      clientAuthenticationMethod: client_secret_post
      authorizationGrantTypes:
        - authorization_code
        - refresh_token
      redirectUri: "https://social.yandex.net/broker/redirect"
      additionalAccount: YandexAccount
    {{ core.sberSkill.clientId }}:
      type: SBER
      password: "{{ core.sberSkill.password }}"
      clientAuthenticationMethod: client_secret_post
      authorizationGrantTypes:
        - authorization_code
        - refresh_token
```

```
      redirectUri: "https://social.yandex.net/broker/redirect"
      additionalAccount: YandexAccount
    {{ core.sberSkill.clientId }}:
      type: SBER
      password: "{{ core.sberSkill.password }}"
      clientAuthenticationMethod: client_secret_post
      authorizationGrantTypes:
        - authorization_code
        - refresh_token
      redirectUri: "https://gateway.iot.sberdevices.ru/gateway/v1/binder/backward"
      additionalAccount: SberAccount
    {{ core.marusyaSkill.clientId }}:
      type: MARUSYA
      password: "{{ core.marusyaSkill.password }}"
      clientAuthenticationMethod: client_secret_post
      authorizationGrantTypes:
        - authorization_code
        - refresh_token
      redirectUri: "https://vc.go.mail.ru/smarthouse/{{ core.marusyaSkill.appId }}/callback"
      additionalAccount: MarusyaAccount

#         ()
# https://yandex.ru/dev/dialogs/smart-home/doc/concepts/about.html
yandex-skill:
  #       oauth2.clientParameters   client-id
  client-id: "{{ core.yandexSkill.clientId }}"
  #   API
  callback-uri: "https://dialogs.yandex.net/api/v1/skills"
  # Id ,       (   )
  skill-id: "{{ core.yandexSkill.skillId }}"
  # OAuth-,       (    )
  oauth-token: "{{ core.yandexSkill.oauthToken }}"

#
# https://developers.sber.ru/docs/ru/smarthome/overview
sber-skill:
  #       oauth2.clientParameters   client-id
  client-id: "{{ core.sberSkill.clientId }}"
  #   API
  callback-uri: "https://partners.iot.sberdevices.ru"
  # Bearer-,       (    )
  bearer-token: "{{ core.sberSkill.bearerToken }}"

#     Mail.ru
# https://help.mail.ru/marusia/smart_home
marusya-skill:
  #       oauth2.clientParameters   client-id
  client-id: "{{ core.marusyaSkill.clientId }}"
  #   API
  callback-uri: "https://vc.go.mail.ru/smarthouse/events/yandex"
```

```
  # Id ,       (   )
  app-id: "{{ core.marusyaSkill.appId }}"
  # OAuth-,       (    )
  oauth-token: "{{ core.marusyaSkill.oauthToken }}"

scriptengine:
  #        executionNumberLimit.
  #       scriptTimeLimit     scriptTimeLimit (),
  #   .
  frequentScript:
    executionNumberLimit: 60
    scriptTimeLimit: 60000
  #          executionNumberLimit.
  #       scriptTimeLimit     scriptTimeLimit (),
  #     .
  frequentScriptsInHouse:
    executionNumberLimit: 120
    scriptTimeLimit: 120000

logging:
  config: "{{ coreInternal.logbackConfig }}"
  logback:
    dir: "/var/log/eltex-sc"
{% if elk.enable %}
  logstash:
    host: "{{ 'logstash' if iot.serverName == elk.serverName else elk.serverName }}"
    port: {{ 5000 if iot.serverName == elk.serverName else elkServices.logstash.port.map }}
{% endif %}
  level:
    root: {{ core.logLevel }}
    org.springframework: WARN
    org.springframework.cache: WARN
    org.springframework.data: WARN
    org.springframework.web: WARN
    _org.springframework.web: WARN
    org.springframework.security: WARN
    org.springframework.security.oauth2: WARN
    org.springdoc: WARN
    org.mongodb: WARN
    org.eclipse.jetty: WARN
    org.apache.http: WARN
    org.hibernate: WARN
    org.jboss: WARN
    org.quartz.core.QuartzSchedulerThread: WARN
    io.swagger: WARN
    io.github.resilience4j: WARN
    io.netty: WARN
    io.mongock: WARN
    io.micrometer: WARN
    com.hazelcast: WARN
    com.hivemq: WARN
```

```
entitiesLimit:
  support:
    #      30
    maxNumberOfOpenTickets: 30
    #
    maxNumberOfMessagesPerDay: 1000
    #  ,      .     .
    numberOfDaysForTicketExpired: 365

fileLimits:
  support:
    #        ()    UI
    maxNumberOfUiLogFilesLinkToTicket: 2
    maxSizeOfUiLogFilesLinkToTicket: 10
    #        ()
    maxNumberOfAttachedFilesLinkToTicket: 15
    maxSizeOfAttachedFilesLinkToTicket: 15
    #        ()
    maxNumberOfAttachedFilesLinkToMessage: 1
    maxSizeOfAttachedFilesLinkToMessage: 10
    #    ()   .        .
    maxSizeForFilesAttachedToChat: 20
    #            .
    #     .
    numberOfDaysForExpiredAttachmentsInChat: 7
```

, :

- **8069** — Ethernet TCP-;
- **8070** — WS- ;
- **8071** — HTTP- API-;
- **8072** — WSS- ;
- **8073** — HTTPS- API-;
- **8883** — MQTT-;
- **8088** — CAPTCHA.

Eltex SC . - , Eltex Home.

. : **< Eltex SC>**.

- .

```
htop
```


MEM% CPU% mongo docker-.

```
df -h
```


.

Z-Wave Wi-Fi

```
netstat -na | grep 8070
netstat -na | grep 8883
```

, :

- :
  - **ESTABLISHED**, **LISTEN** — ;
  - **LAST_ACK** — , IP;
  - **TIME_WAIT**, **CLOSE_WAIT** — , ;
- 2- — ;
- 3- — , ;
- 5- — IP- .

, . , - , -.

- **/storage/iot/core/var/log/eltex-sc/server***. . (15 2022) , :

```
grep <  > server-2022-01-15* > <   >
```

, :

- **-i** — ;
- **-n** — ;
- **-h** — ;
- **-A** — ;
- **-B** — .

:

```
grep -i -n -A 5 -B 2 error server-2022-01-15* > errors.log
```

:

- *ERROR;*
- *ID IP* ;
- *PONG* — , , .

-

- — (IP —).

API . URL - :

```
<  Eltex SC>:< API>/api/v1/version
```

API — API HTTP (**8071/8073**). , API JSON.

:

```
{
  "version" : "1.27-3477",
  "api" : "1.0",
  "currentTime" : "2022-07-25T09:24:12.544842Z[Etc/UTC]"
}
```

http https

⚠ PKCS#8.

```
# head -1 /tmp/eltex-sc-api.key
-----BEGIN PRIVATE KEY-----

# head -1 /storage/iot/ssl/private/eltex-sc-api.key
-----BEGIN RSA PRIVATE KEY-----
```

- BEGIN PRIVATE KEY — PKCS#8;
- BEGIN RSA PRIVATE KEY — PKCS#1.

PKCS#1, PKCS#8.

PKCS#1 PKCS#8, letsencrypt:

```
openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in __ -out ___.
```

CAPTCHA

CAPTCHA . , - .

CAPTCHA **/etc/ansible-iot-1.27/templates/default-for-docker.yml.j2**

/etc/ansible-iot-1.27/vars/service_parametrs.yml

caseSensitive .

/etc/ansible-iot-1.27/vars/default.yml

easy	. .
medium	. .
hard	. .