QUICK START. v1.26

1.26

Eltex SC (Eltex Smart Cloud) , (, , web- ..). web- :

- (, ,);
- ;
- ;
- .

( ):

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

Eltex SC

Eltex SC 1.26 Ubuntu 20.04. Eltex SC (1.25) : Eltex SC.

1.19.4 deb- _.

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-1.26.tar.gz
```

.

(/) Ansible **/etc/ansible-iot-1.26**.

5. **/etc/ansible-iot-1.26/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-1.26/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
    # E-mail  .         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
```

vars/default.yml

```
#   IoT core
core:
  #    IoT Core
  logLevel: INFO

  #      Z-Way
  ctlGate:
    port: 8070
    tcpPort: 8069
    sslPort: 8072

  #  API
  api:
    port: 8071
    sslPort: 8073

  #   CAPTCHA: 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

  #
  video:
    #  Flussonic.
    flussonic:
      url: ""
      apiKey: ""
      operatorId: ""
      adminLogin: ""
    #   eltex
    eltex_server:
      url: ""
```

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

vars/default.yml

```
#     (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

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

vars/default.yml

```
#  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

  #      Z-Way
  ctlGate:
    port: 8070
    tcpPort: 8069
    sslPort: 8072

  #  API
  api:
    port: 8071
    sslPort: 8073

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

vars/default.yml

```
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

#
video:
  #  Flussonic
  flussonic:
    url: ""
    apiKey: ""
    operatorId: ""
    adminLogin: ""
  #   eltex
  eltex_server:
    url: ""

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

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

vars/default.yml

```
#     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- .

7. :

```
cd /etc/ansible-iot-1.26
ansible-playbook install_iot.yml
```

: **http://[Eltex SC]**.

server_name /etc/ansible-iot-1.26/vars/default.yml.

API .

Eltex SC.

: **/etc/ansible-iot-1.26/templates/iot/default-for-docker.yml.j2** -: **/etc/ansible-iot-1.26/templates/iot/web/base_config**.

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

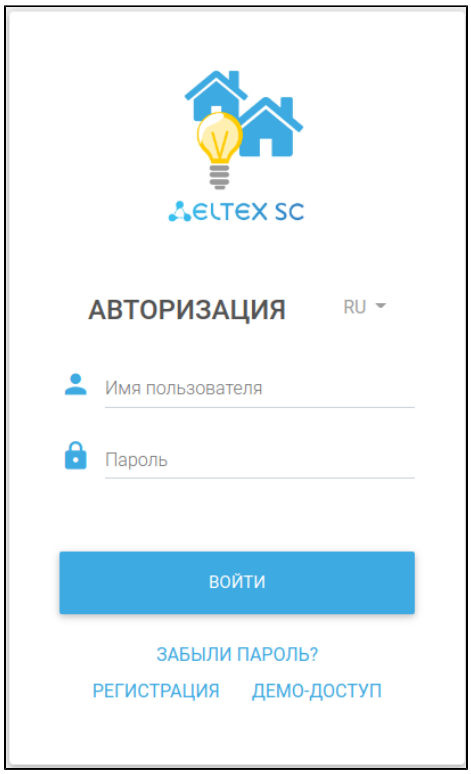
Eltex SC

web- Eltex SC, **/etc/ansible-iot-1.26/vars/default.yml**.

1. : **http://[Eltex SC]**. .
2. . .

«»:

: **admin**
: **Test18plat34Form**



v1.26 « » : Eltex SC.