


# SP5000 интеллектуальная облачная платформа интернета вещей "IoT SP5000 "Элемент"



Инструкция по установке и настройке

Екатеринбург 2018

Все права защищены.

Технические данные и спецификации являются обязательными только в том случае, если они отдельно согласованы в договоре в письменной форме.

Сохраняется право на внесение технических изменений.

## Оглавление

1.	О документе	4
2.	Подготовка к развертыванию платформы IAPE	4
2.1.	Инсталляция ansible-машины	4
2.2.	Создание виртуальных машин для IAPE в OpenStack	5
2.3.	Конфигурация OpenStack для развёртывания IAPE	9
2.4.	Конфигурация ansible для развёртывания IAPE	13
3.	Процедура развертывания платформы IAPE	26
3.1.	Варианты развертывания платформы IAPE	26
	Вариант развертывания 1 (с кластером узлов)	26
	Вариант развертывания 2 (с одиночными узлами)	27
3.2.	Команды для развертывания платформы IAPE	28
3.3.	Проверка работы компонентов платформы IAPE	30
3.3.1.	FreeIPA	30
3.3.2.	Keycloak	31
3.3.3.	Cassandra	32
3.3.4.	Стек ELK	32
3.3.5.	Компоненты ServiceMix	33
3.3.6.	Kafka и Zookeeper в архитектуре с одиночными узлами	34
3.3.7.	Модули мониторинга	35
3.3.8.	Узел Spark	37
3.3.9.	Интерфейс пользования IAPE GUI	38
3.3.10.	HAProxy, Keepalived и HTTPD	39
3.3.11.	Шлюз API GW	40
3.3.12.	Веб-интерфейс репозитория CIMrepo	41
3.3.13.	Узлы в кластере Mariadb-Galera	43
3.3.14.	Интерфейс шлюза SDP	44

# 1. О документе

Документ описывает процесс автоматизированной инсталляции платформы.

## 2. Подготовка к развертыванию платформы IAPE

### 2.1. Инсталляция ansible-машины

Табл. 2.1. Технические требования для ansible-машины

ОС	Ядер ЦП	ОЗУ (ГБ)	Жесткий диск (ГБ)
Centos - 1611	1	2	40

Подготовка OpenStack к инсталляции ansible-машины:

- Создать проект ("tenant") и пользователя ("user") с соответствующими административными правами;
- Создать внутреннюю сеть ("network") с параметром **Общая = Да**;
- Создать маршрутизатор ("router");
- Создать тип инстанса ("flavor") с параметрами, указанными в требованиях для ansible-машины;
- Создать группу безопасности ("security group") и добавить в нее правила для протоколов TCP/UDP/ICMP/112 (все порты должны быть открыты);
- Импортировать ключевую пару ("key pair") с именем "**ansible**", используя публичный ключ ssh для ansible-машины:

**"ssh-rsa**

```
AAAAB3NzaC1yc2EAAAADAQABAAQCAQC73kPKmhUAJUo5wO96pRb40TAFpIHBgK6XAdPF
ALyHPSQ0eqHjBOiybinGuxHOo+hFuFnFUzfsdaVonutFKI+PuyZLLDAC8eVxp1ZDc+uRI6okqyp
SiyIWH0ect5UiXXh1f0qB0Ch0ticEnK9ThIR48zUSIAgQ0DVZ6WaZQG10z2Se21bSDWZfbz19M/kA
GSV6Z18GTUUQwhLHxS/QkOukjpkX0n69CtMWpKiuBNhL1QtmdmFQfOOmd3u6urz49quMaWO
0o9iMxjf5hbA0hKriUV5v+DMiOSi0TjSW8X76FbEQ3M5RZ3VGJI0cTtw48ZZGO+AlqXoryyEhyHR
udTft Generated-by-Nova"
```

Инсталляция ansible-машины в OpenStack:

1. Скопировать ansible-образ для IAPE на контрольную ноду OpenStack (например, при помощи WinSCP).
2. Скачать файл "OpenStack RC-файл версии 3" из графического интерфейса OpenStack. и скопировать его на контрольную ноду OpenStack (например, при помощи WinSCP).
3. Подключиться по ssh к контрольной ноду OpenStack (например, при помощи Putty) с логином/паролем = **root/fai** и выполнить команду "**source <RC-file name>.sh**"
4. Импортировать ansible-образ с помощью команды:
 

```
"openstack image create --disk-format raw --container-format bare --public --file <image-file name>.raw ansible"
```
5. Создать и запустить ansible-машину в OpenStack на основе импортированного образа и ранее заданных параметров. Подключиться по ssh к созданной ansible-машине (например, при помощи Putty) с логином/паролем = **root/iskratel**

6. Выполнить конфигурацию модуля shade для работы ansible-машины с OpenStack путём изменения конфигурационного файла “~/config/openstack/clouds.yml” под значения параметров, созданных в OpenStack ранее.

```
clouds:
#Name of tenant
  mpt-prod:
  auth:
#Username and password for login to tenant
  username: mpt-user
  password: iskratel
#Name of tenant
  project_name: mpt-prod
  project_domain_name: Default
  user_domain_name: Default
#Auth url from openstack keystone
  auth_url: http://iotstack.iskrauraltel.ru:5000/v3
```

## 2.2. Создание виртуальных машин для IAPE в OpenStack

1. Выполнить конфигурацию ansible роли “*create\_cluster\_OSinstance*” путём внесения в файл: “/etc/ansible/role/*create\_cluster\_OSinstance/defaults/main.yml*” (в варианте с кластером узлов) или “/etc/ansible/role/*create\_single\_OSinstance/vars/main.yml*” (в варианте с одиночными узлами) значений, соответствующих созданным в OpenStack параметрам.

```
---
# Create network and router
create_network_component: false
# Create network ( Internal )
SHARED_NETWORK: iape_internal_net
network_subnet : 172.16.37.0/24
network_gateway : 172.16.37.1
network_dns : 8.8.8.8
# Name from router
router_name : iape_router
#Don't change name of external_net if dont change of openstack environment
name_external_net: public_floating_net
#####
attach_key: false
PUBLIC_KEY: ssh-rsa
```

```

AAAAB3NzaC1yc2EAAAADAQABAAQBAQC73kPKmhUAJUo5wO96pRb40TAFpIHBgK6XAdPFA
LyHPSQ0eqHjBOiybinGuxHOo+hFuFnFUzfsdaVonuTFKI+PuyZLLDAC8eVxp1ZDc+uRI6okqypSiy
IWH0ect5UiXXh1f0qB0Ch0ticEnK9ThIR48zU$
#Instance auth key
KEY_NAME: ansible
# Attach new image on openstack
attach_image: true
# Name of image, attach on openstack or to create instance from this images.
OSIMG: centos7-iape
#Image Format who need to upload on openstack end attach (ex. raw,qcow2c)
img_format: qcow2
#Name of security_group
security_group: iape_security_group
#####
# Create volume storage
create_volume: true
openstack_availability_zone: nova
openstack_volume_device: /dev/vdb
type: volumes_ceph

image_prefix: /opt/SI5000/qcow
export_inventory_file: "/tmp/ansible-hosts.cfg"
vm_parameter:
- { name: cassandra01, cpu: 3, ram: 8192, disk: 50, volumes: 100 }
- { name: cassandra02, cpu: 3, ram: 8192, disk: 50, volumes: 100 }
- { name: cassandra03, cpu: 3, ram: 8192, disk: 50, volumes: 100 }
- { name: zookeeper01, cpu: 2, ram: 3072, disk: 10, volumes: 50 }
- { name: zookeeper02, cpu: 2, ram: 3072, disk: 10, volumes: 50 }
- { name: zookeeper03, cpu: 2, ram: 3072, disk: 10, volumes: 50 }
- { name: skz01, cpu: 8, ram: 20480, disk: 100, volumes: 100 }
- { name: skz02, cpu: 8, ram: 20480, disk: 100, volumes: 100 }
- { name: skz03, cpu: 8, ram: 20480, disk: 100, volumes: 100 }
- { name: esb01, cpu: 3, ram: 6144, disk: 50, volumes: 10 }
- { name: esb02, cpu: 3, ram: 6144, disk: 50, volumes: 10 }
- { name: esb03, cpu: 3, ram: 6144, disk: 50, volumes: 10 }
- { name: apigw01, cpu: 2, ram: 8192, disk: 10, volumes: 20 }
- { name: apigw02, cpu: 2, ram: 8192, disk: 10, volumes: 20 }
- { name: puzzle, cpu: 4, ram: 4096, disk: 30, volumes: 10 }

```

```
- { name: cimrepo, cpu: 4, ram: 8192, disk: 100, volumes: 10 }
```

```
vm_parameter1:
```

```
- { name: qminer, cpu: 3, ram: 8192, disk: 20 }
- { name: sdp-controller, cpu: 1, ram: 2048, disk: 8 }
- { name: sdp-gateway, cpu: 1, ram: 2048, disk: 8 }
- { name: keycloak, cpu: 2, ram: 8192, disk: 10 }
- { name: monitoring, cpu: 1, ram: 8192, disk: 20 }
- { name: freeipa-replica, cpu: 1, ram: 8192, disk: 8 }
- { name: dns, cpu: 1, ram: 2048, disk: 8 }
- { name: haproxy-master, cpu: 2, ram: 2048, disk: 8 }
- { name: haproxy-backup, cpu: 2, ram: 2048, disk: 8 }
- { name: iapegui, cpu: 1, ram: 8192, disk: 20 }
- { name: elkstack, cpu: 4, ram: 8192, disk: 50 }
```

или

```
---
# Create network and router
create_network_component: false
# Create network ( Internal )
SHARED_NETWORK: iape_internal_net
network_subnet : 172.16.37.0/24
network_gateway : 172.16.37.1
network_dns : 8.8.8.8
# Name from router
router_name : iape_router
#Don't change name of external_net if dont change of openstack environment
name_external_net: public_floating_net
#####
attach_key: false
PUBLIC_KEY: ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCA73kPKmhUAJUo5wO96pRb40TAFpIHBgK6XAdPFA
LyHPSQ0eqHjBOiybinGuxHOo+hFuFnFUzfsdaVonuTFKI+PuyZLLDAC8eVxp1ZDc+uRI6okqypSiy
IWH0ect5UiXXh1f0qB0Ch0ticEnK9ThIR48zU$
#Instance auth key
KEY_NAME: ansible
# Attach new image on openstack
```

```

attach_image: true
# Name of image, attach on openstack or to create instance from this images.
OSIMG: centos7-iape
#Image Format who need to upload on openstack end attach (ex. raw,qcow2)
img_format: qcow2
#Name of security_group
security_group: iape_security_group
#####
# Create volume storage
create_volume: true
openstack_availability_zone: nova
openstack_volume_device: /dev/vdb
type: volumes_ceph

image_prefix: /opt/SI5000/qcow
export_inventory_file: "/tmp/ansible-hosts.cfg"
vm_parametar:
- { name: cassandra , ram: 8192 , cpu: 3 , disk: 10 , volumes: 20 }
- { name: spark, cpu: 3, ram: 8192, disk: 10, volumes: 10 }
- { name: zookeeper-kafka, cpu: 2, ram: 3072, disk: 10, volumes: 10 }
- { name: esb, cpu: 3, ram: 6144, disk: 10, volumes: 10 }
- { name: apigw, cpu: 2, ram: 8192, disk: 10, volumes: 20 }
- { name: puzzle, cpu: 4, ram: 4096, disk: 30, volumes: 10 }
- { name: qminer, cpu: 3, ram: 8192, disk: 20, volumes: 0 }
- { name: sdp-controller, cpu: 1, ram: 2048, disk: 8, volumes: 0 }
- { name: sdp-gateway, cpu: 1, ram: 2048, disk: 8, volumes: 0 }
- { name: keycloak, cpu: 2, ram: 8192, disk: 10, volumes: 0 }
- { name: monitoring, cpu: 1, ram: 8192, disk: 20, volumes: 0 }
- { name: dns, cpu: 1, ram: 2048, disk: 8, volumes: 0 }
- { name: iapegui, cpu: 1, ram: 8192, disk: 20, volumes: 0 }
- { name: cimrepo, cpu: 4, ram: 8192, disk: 100, volumes: 10 }

```

2. Из директории “**/etc/ansible**” запустить ansible-плейбук для развёртывания виртуальных машин IAPE при помощи команды:
  - в варианте с кластером узлов - “**ansible-playbook createClusterOS.yml**”. После успешного выполнения этого плейбука в OpenStack должны появиться 27 новых виртуальных машин.
  - в варианте с одиночными узлами - “**ansible-playbook createSingleOS.yml**”. После успешного выполнения этого плейбука в OpenStack должны появиться 14 новых виртуальных машин.



3. Выполнить конфигурацию ansible-роли “*iape.freeipa.import*” путём внесения в файл “*/etc/ansible/role/iape.freeipa.import/defaults/main.yml*” значений, которые соответствуют созданным в OpenStack параметрам.

```

---
# Create network and router
#Name from Tenant
cloud_name: mpt-prod
create_network_component: false
# Create internal network (set name and subnet)
SHARED_NETWORK: iape_internal_net
network_subnet : 172.16.37.0/24
network_gateway : 72.16.37.1
network_dns : 8.8.8.8
router_name : iape_router
name_external_net: public_floating_net
#####
#####
#Create FLAVOR
name_flavor: freeipa
vram: 8192
vcpu: 1
vhdd: 20
#####
# Create instance
INSTNAME: freeipa-master
KEY_NAME: ansible
# Attach new image on openstack
attach_image: true

image_prefix: /opt/SI5000/qcow

# Name of image, attach on openstack or which need to attach
OSIMG: freeipa-07122018
#####
### Test #####
#float_ip: 192.168.142.71
#####
#Securiy Group Name
secgr: freeipa_secgr

```

4. Из директории “*/etc/ansible*” запустить ansible-плейбук для развёртывания виртуальной машины FreeIPA при помощи команды “*ansible-playbook freeipa.yml*”. После успешного выполнения этого плейбука в OpenStack должна появиться новая виртуальная машина с именем “*freeipa-master*”.

### 2.3. Конфигурация OpenStack для развёртывания IAPE

**Замечание:** Для варианта развёртывания с одиночными узлами выполнение описанных ниже пунктов 1. - 8. не требуется!

1. Скачать файл “OpenStack RC-файл версии 3” из графического интерфейса OpenStack и скопировать его на контрольную ноду OpenStack (например, при помощи WinSCP).
2. Подключиться по ssh к контрольной ноде OpenStack (например, при помощи Putty) с логином/паролем = **root/fai** и выполнить команду:

```
# source <RC-file name>.sh
```

3. Создать VIP-порт, выполнив команду:

```
# neutron port-create --name vip-port <internal_interface_name>
```

4. Вывести значения **<vm\_port\_id>**, выполнив команды::

```
# neutron port-list | grep <internal_ip_from_haproxy1> | awk '{print $2}'
# neutron port-list | grep <internal_ip_from_haproxy2> | awk '{print $2}'
```

5. Вывести значение **<ip\_vip\_port>**, выполнив команду::

```
# neutron port-list --name vip-port | grep ip_address | awk -F "\"" '{print $8}'
```

6. Присоединить VIP-порт к виртуальным машинам HA Proxy, выполнив команды:

```
# neutron port-update <vm_port_id> --allowed-address-pairs type=dict list=true
ip_address=<ip_vip_port>
# neutron port-update <vm_port_id> --allowed-address-pairs type=dict list=true
ip_address=<ip_vip_port>
```

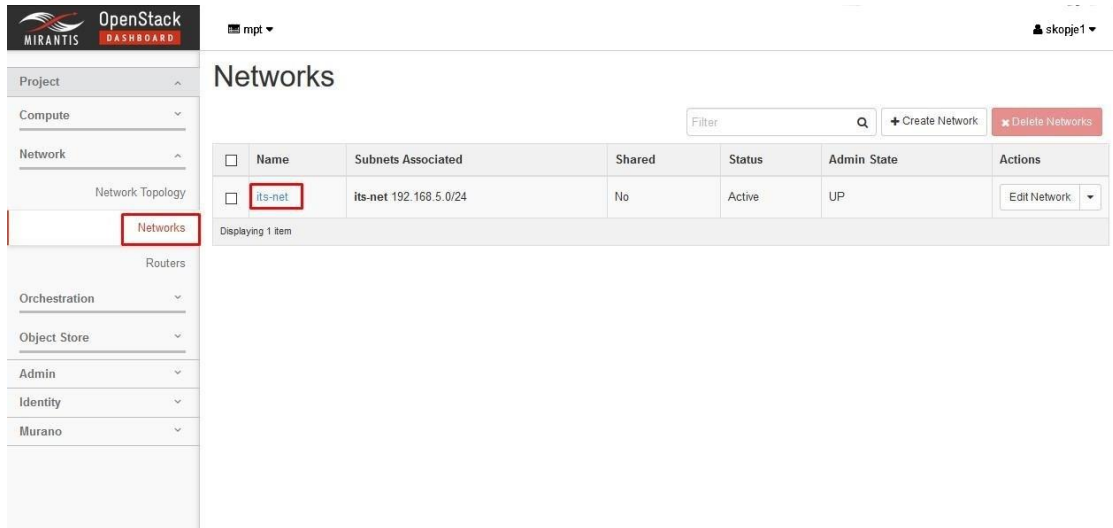
7. Связать IP-адрес с VIP-портом, выполнив команды:

```
# neutron floatingip-create <floating_interface_name>
# neutron floatingip-associate <floating_id> <vip_id>
где:
- <floating_id> значение параметра id, полученное из вывода команды "neutron floating-create ..."
- <vip_id> значение параметра id, полученное из вывода команды "neutron port-create ..." на шаге 3.
```

8. Отключить настройки безопасности для VIP-порта, выполнив команды:

```
# neutron port-update <vip_id> --no-security-groups
# neutron port-update <vip_id> --port-security-enabled=false
где: <vip_id> значение параметра id, полученное из вывода команды "neutron port-create ..."
на шаге 3.
```

9. Прописать IP-адрес сервера DNS в сетевых настройках графического интерфейса Openstack
- В интерфейсе панели управления Openstack в разделе **Networks** выбрать внутреннюю сеть:



The screenshot shows the OpenStack dashboard interface. The left sidebar contains navigation menus for Project, Compute, Network, Routers, Orchestration, Object Store, Admin, Identity, and Murano. The 'Networks' menu item is highlighted with a red box. The main content area displays the 'Networks' section with a table of network configurations. The table has columns for Name, Subnets Associated, Shared, Status, Admin State, and Actions. One network, 'its-net', is listed with the subnet '192.168.5.0/24', a status of 'Active', and an admin state of 'UP'. The 'its-net' name in the table is also highlighted with a red box. Above the table, there are buttons for '+ Create Network' and 'Delete Networks', and a search filter box. Below the table, it says 'Displaying 1 item'.

<input type="checkbox"/>	Name	Subnets Associated	Shared	Status	Admin State	Actions
<input type="checkbox"/>	its-net	its-net 192.168.5.0/24	No	Active	UP	Edit Network

Рис.1. Вкладка Network в графическом интерфейсе Openstack

- Выбрать команду **Edit Subnet** для редактирования настроек подсети:

The screenshot shows the OpenStack dashboard interface. On the left is a navigation sidebar with categories like Project, Compute, Network, and Admin. The main content area is titled 'Network Details: its-net'. It includes a 'Network Overview' section with various attributes like Name, ID, Project ID, Status, Admin State, Shared, External Network, MTU, and Provider Network. Below this is a 'Subnets' table with columns for Name, Network Address, IP Version, Gateway IP, and Actions. A single row is shown for 'its-net' with a red box highlighting the 'Edit Subnet' dropdown in the Actions column. There are also buttons for '+ Create Subnet' and 'Delete Subnets' above the table.

**Рис. 2. Конфигурация подсети Openstack**

- На вкладке **Subnet details**, в поле **DNS name servers** ввести IP-адрес сервера DNS и сохранить настройки:

The screenshot shows the 'Edit Subnet' dialog box. At the top, there are two tabs: 'Subnet' and 'Subnet Details', with the latter being selected and highlighted by a red box and the number '1'. Below the tabs, there is a checkbox for 'Enable DHCP' which is checked. A text input field for 'Allocation Pools' contains '192.168.5.2,192.168.5.254'. Below that is a 'DNS Name Servers' field with a red box around the first IP address '192.168.5.45' and the number '2'. The field also contains '8.8.8.8'. At the bottom, there is a 'Host Routes' field which is empty. At the bottom right, there is a 'Save' button highlighted with a red box and the number '3', and a 'Back' button to its left.

**Рис. 3. Настройка IP-адреса сервера DNS**

## 2.4. Конфигурация ansible для развёртывания IAPE

1. Изменить значения IP-адресов в файле `“/etc/ansible/hosts”` в соответствии с внешними IP-адресами виртуальных машин, созданных в OpenStack. Для варианта с кластером узлов изменения необходимо делать в разделе [pipeline], а для варианта с одиночными узлами - в разделе [case2].

```
##### Ansible Controller
#####

[ansible]
<ansible_float_ip>

#####
#####
##### PIPELINE
#####
#####

[pipeline]
#####
#####
<dns_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<ipa_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<cimrepo_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<puzzle_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<haproxy01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<haproxy02_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<skz01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<skz02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<skz03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<keycloak_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

```
<iapegui_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<sdpc_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<sdpg_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<apigw01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<apigw02_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<monitoring_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<qminer_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<elk_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<cassandra01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<cassandra02_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<cassandra03_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<esb01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<esb02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<esb03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<zoo01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<zoo02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<zoo03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

[dns]

```
<dns_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

[ipacli]

```
<ipa_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

[cimrepo]

```
<cimrepo_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

[puzzle]

```
<puzzle_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

```
[haproxy2node]
<haproxy01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<haproxy02_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
[haproxy-master]
<haproxy01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
[haproxy-backup]
<haproxy02_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[skz-cluster]
<skz01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<skz02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<skz03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[skz01]
<skz01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[skz02]
<skz02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[skz03]
<skz03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[keycloak_case1]
<keycloak_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[iapegui-case1]
<iapegui_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[sdpc-case1]
<sdpc_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[sdpg-case1]
<sdpg_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

```
[api_gw-case1]
<apigw01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<apigw02_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[api_gw01]
<apigw01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
[api_gw02]
<apigw02_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[monitoring-case1]
<monitoring_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[EnergyForecast-case1]
<qminer_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[elk-case1]
<elk_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[cassandra-cluster]
<cassandra01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<cassandra02_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<cassandra03_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[cassandra01]
<cassandra01_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[cassandra02]
<cassandra02_float_ip> ansible_ssh_user=centos
```



```
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[cassandra03]
<cassandra03_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[servicemix-cluster]
<esb01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<esb02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<esb03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[servicemix01]
<esb01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[servicemix02]
<esb02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[servicemix03]
<esb03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[zoo-cluster]
<zoo01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<zoo02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem
<zoo03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[zoo1]
<zoo01_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[zoo2]
<zoo02_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

[zoo3]
<zoo03_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#####
#####
##### CASE 2 #####
```

```
#[case2]
#<dns_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<ipa_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<cassandra_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<spark_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<zo_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<esb_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<apigw_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<puzzle_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<qminer_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<sdpc_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<sdpg_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<keycloak_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<monitoring_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<iapegui_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<cimrepo_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[cimrepo]
#<cimrepo_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

```
#[ipareplica]
#<ipaserv_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
#<ipacli_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[ipaserv]
#<ipaserv_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[ipacli]
#<ipacli_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[kafka]
#<kafka_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[zookeeper]
#<zoo_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[spark]
#<spark_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[keycloak]
#<keycloak_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[cassandra]
#<cassandra_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

```
#[esb]
#<esb_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[elk]
#<elk_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[api_gw01]
#<apigw_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[iapegui]
#<iapegui_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[EnergyForecast]
#<qminer_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[monitoring]
#<monitoring_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[sdpc]
#<sdpc_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[sdpg]
#<sdpg_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#[puzzle]
#<puzzle_float_ip> ansible_ssh_user=centos
ansible_ssh_private_key_file=/etc/ansible/ansible.pem
```

```
#[dns]
#<dns_float_ip> ansible_ssh_user=centos ansible_ssh_private_key_file=/etc/ansible/ansible.pem

#####
```

2. Выполнить конфигурацию ansible-роли “*iape.named.install*” путём внесения изменений в файл “*/etc/ansible/role/iape.named.instal/defaults/main.yml*”. Необходимо указать последний октет IP-адреса внутренней сети и внешней сети для каждой виртуальной машины и для созданного в OpenStack VIP-порта. Например: *ipa: { type: A, last: <последние октет>, mac: }*)

```
---
mix_mirror : http://ansible.fuel.its.mak/repo
mix_download_timeout_seconds: 600
mix_install_parent_dir: /opt/SI5000
node_name : 'node_exporter'
node_tgz : '{{node_name}}.tar.gz'
node_tgz_url : '{{mix_mirror}}/{{node_tgz}}'
hostname: dns.iape.iskrauraltel.ru
# defaults file for dns
#Domain name
domain: iape.iskrauraltel.ru
#Internal reverse domain
rev_domain: 37.16.172.in-addr.arpa
#Floating reverse domain
float_rev_domain: 56.168.192.in-addr.arpa
#Internal subnet for systems
dns_subnet: 172.16.37
#Floating subnet for system
float_dns_subnet: 192.168.56
#Dictionary of hosts.
#Type sets the record type.
#Last sets the last octet of the address.
#MAC is optional. The same dictionary can be used with the DHCP role.

#RECORD FOR INTERNAL NETWORK
records:
#FreelIPA
ipa: { type: A, last: 28, mac: }
#DNS node
dns: { type: A, last: "{{ ansible_default_ipv4.address.split('.')[3] }}" , mac: }
#FreelIPA client
cli: { type: A, last: 32, mac: }
#Keycloak
keycloak: { type: A, last: 29, mac: }
#monitoring + elkstack
prometheus: { type: A, last: 22, mac: }
```

```
#Puzzle node
puzzle: { type: A, last: 24, mac: }
#API Gateway cluster
api-gw: { type: A, last: 19, mac: }
api-gw-sec: { type: A, last: 6, mac: }
#SDP controller
sdp-controller: { type: A, last: 4, mac: }
#SDP Gateway
sdp-gateway: { type: A, last: 27, mac: }
#Haproxy nodes
haproxy1: { type: A, last: 31, mac: }
haproxy2: { type: A, last: 20, mac: }
#VIP port
vip: { type: A, last: 3, mac: }
ipa-ha: { type: A, last: 3, mac: }
#IAPE GUI node
ui: { type: A, last: 35, mac: }
orin: { type: A, last: 35, mac: }
analytics: { type: A, last: 35, mac: }
#Cassandra cluster nodes
cassandra01: { type: A, last: 8, mac: }
cassandra02: { type: A, last: 14, mac: }
cassandra03: { type: A, last: 9, mac: }
#SKZ cluster nodes
spark01: { type: A, last: 11, mac: }
spark02: { type: A, last: 13, mac: }
spark03: { type: A, last: 21, mac: }
kafka01: { type: A, last: 11, mac: }
kafka02: { type: A, last: 13, mac: }
kafka03: { type: A, last: 21, mac: }
# Zookeeper
zookeeper01: { type: A, last: 10, mac: }
zookeeper02: { type: A, last: 7, mac: }
zookeeper03: { type: A, last: 5, mac: }
#Apache Service MIX cluster nodes
esb1: { type: A, last: 17, mac: }
esb2: { type: A, last: 12, mac: }
esb3: { type: A, last: 23, mac: }
#Analytics node
qminer01: { type: A, last: 16, mac: }
#CimRepo node
cimrepo: { type: A, last: 25, mac: }
#Monitoring node
monitoring: { type: A, last: 22, mac: }
#Elkstack
elkstack: { type: A, last: 38, mac: }

#RECORD FOR FLOATING NETWORK
float_records:
```

```
#FreelIPA
  ipa: { type: A, last: 79, mac: }
#DNS node
# dns: { type: A, last: "{{ ansible_default_ipv4.address.split('.')[3] }}" , mac: }
#FreelIPA client
  cli: { type: A, last: 76, mac: }
#Keycloak
  keycloak: { type: A, last: 77, mac: }
#monitoring + elkstack
  prometheus: { type: A, last: 57, mac: }
#Puzzle node
  puzzle: { type: A, last: 71, mac: }
#API Gateway cluster
  api-gw: { type: A, last: 55, mac: }
  api-gw-sec: { type: A, last: 96, mac: }
#Haproxy for Api-GW and IAPE GUI (different from internal host)
  haproxyapi01: { type: A, last: 69, mac: }
#SDP controller
  sdp-controller: { type: A, last: 58, mac: }
#SDP Gateway
  sdp-gateway: { type: A, last: 89, mac: }
#Haproxy nodes (different from internal host)
  haproxy01: { type: A, last: 91, mac: }
  haproxy02: { type: A, last: 63, mac: }
#VIP port
  vip: { type: A, last: 139, mac: }
  ipa-ha: { type: A, last: 139, mac: }
#IAPE GUI node
  ui: { type: A, last: 84, mac: }
  orin: { type: A, last: 84, mac: }
  analytics: { type: A, last: 84, mac: }
#Cassandra cluster nodes
  cassandra01: { type: A, last: 88, mac: }
  cassandra02: { type: A, last: 81, mac: }
  cassandra03: { type: A, last: 60, mac: }
#SKZ cluster nodes
  spark01: { type: A, last: 65, mac: }
  spark02: { type: A, last: 85, mac: }
  spark03: { type: A, last: 59, mac: }
  kafka01: { type: A, last: 65, mac: }
  kafka02: { type: A, last: 85, mac: }
  kafka03: { type: A, last: 59, mac: }
# Zookeeper
  zookeeper01: { type: A, last: 66, mac: }
  zookeeper02: { type: A, last: 83, mac: }
  zookeeper03: { type: A, last: 97, mac: }
#Apache Service MIX cluster nodes
  esb1: { type: A, last: 80, mac: }
  esb2: { type: A, last: 52, mac: }
  esb3: { type: A, last: 82, mac: }
#Analytics node
```

```

qminer01: { type: A, last: 86, mac: }
#CimRepo node
cimrepo01: { type: A, last: 75, mac: }
#Monitoring node
monitoring: { type: A, last: 57, mac: }
#Elkstack
elkstack: { type: A, last: 62, mac: }

#DNS forwarders.
forwarders:
- 1.1.1.1
- 8.8.8.8

```

3. Для варианта с кластером узлов добавить внутренний IP-адрес созданного в OpenStack VIP-порта в конфигурационные файлы

***“/etc/ansible/roles/iape.haproxy.master.install/defaults/main.yml”***,

***“/etc/ansible/roles/iape.haproxy.backup.install/defaults/main.yml”***

для ansible-ролей ***“iape.haproxy.master.install”*** и ***“iape.haproxy.backup.install”***

```

---
# defaults file for ansible-mariadb-galera-haproxy-keepalived
node_mirror : http://ansible.fuel.its.mak/repo
node_download_timeout_seconds: 600
node_install_parent_dir: /opt/SI5000
node_name : 'node_exporter'
node_tgz : '{{node_name}}.tar.gz'
node_tgz_url : '{{node_mirror}}/{{node_tgz}}'
# vars file for ansible-haproxy-keepalived
# ip address from vip port (dont work with hostname only with internal ip)
virtual_ip: 172.16.37.15
#internal ip address from haproxy 01 instance
haproxy01: haproxy1.iape.iskrauraltel.ru
#internal ip address from haproxy 02 instance
haproxy02: haproxy2.iape.iskrauraltel.ru

#ESB node internal ip
esb01: esb1.iape.iskrauraltel.ru
esb02: esb2.iape.iskrauraltel.ru
esb03: esb3.iape.iskrauraltel.ru

```



```
#nginx internal ip's
nginx01: esb1.iape.iskrauraltel.ru
nginx02: esb2.iape.iskrauraltel.ru
nginx03: esb3.iape.iskrauraltel.ru

#MariaDB server's
mariadb01: cassandra01.iape.iskrauraltel.ru
mariadb02: cassandra02.iape.iskrauraltel.ru
mariadb03: cassandra03.iape.iskrauraltel.ru

#Kafka node
kafka01: kafka01.iape.iskrauraltel.ru
kafka02: kafka02.iape.iskrauraltel.ru
kafka03: kafka03.iape.iskrauraltel.ru

#Zookeeper nodes
zoo01: zookeeper01.iape.iskrauraltel.ru
zoo02: zookeeper02.iape.iskrauraltel.ru
zoo03: zookeeper03.iape.iskrauraltel.ru

#Api-GW node
apigw01: api-gw.iape.iskrauraltel.ru
apigw02: api-gw-sec.iape.iskrauraltel.ru
```

4. Из директории **“/etc/ansible”** запустить ansible-плейбук для развёртывания DNS при помощи команды **“*ansible-playbook dns.yml*”**.
5. В графическом интерфейсе OpenStack выделить все инстансы, кроме DNS, и выполнить их горячую перезагрузку **“soft reboot”**.

### 3. Процедура развертывания платформы IAPE

#### 3.1. Варианты развертывания платформы IAPE

##### Вариант развертывания 1 (с кластером узлов)

Архитектура платформы IAPE с кластером узлов представлена на рисунке ниже:

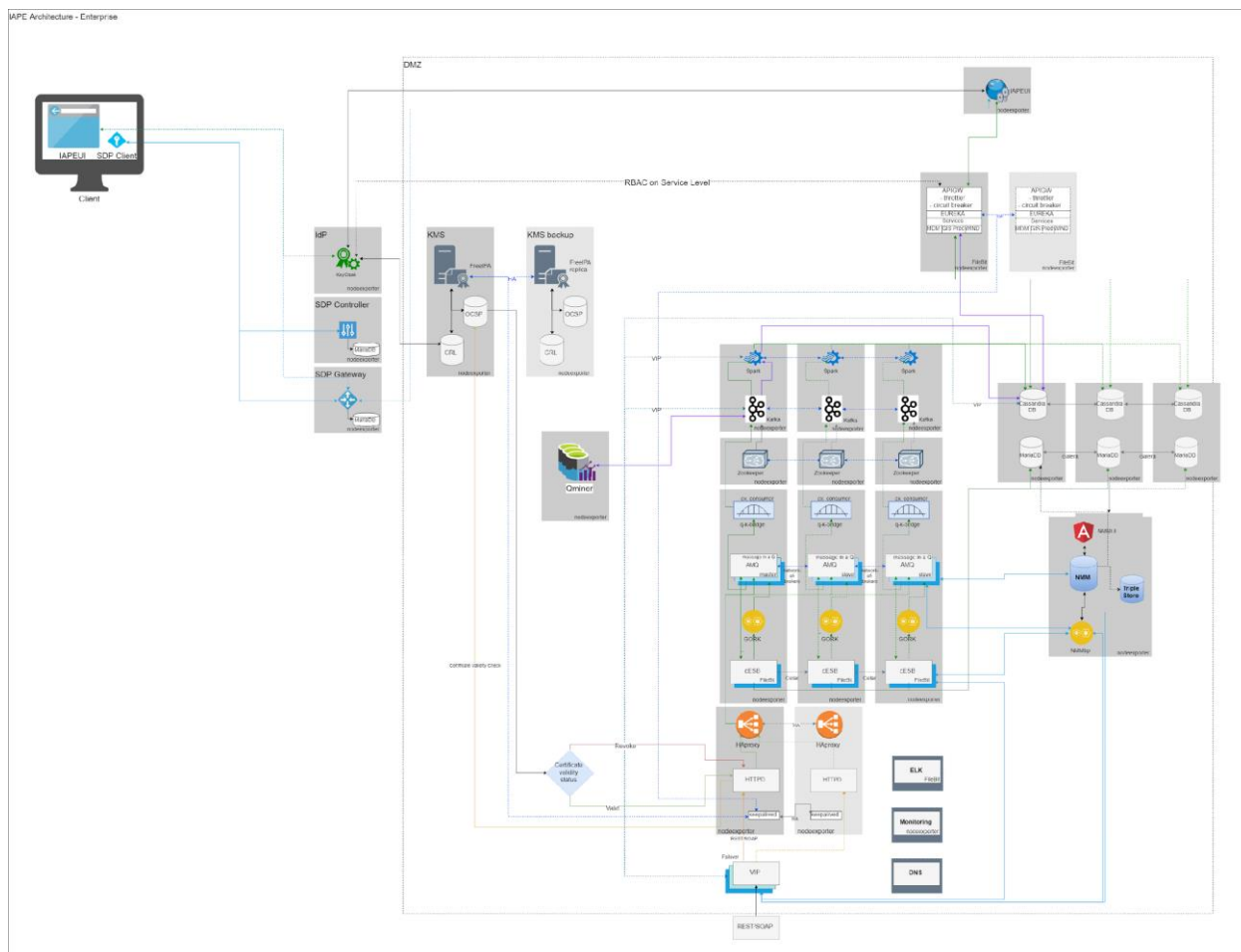


Рис. 2.5. Архитектура развертывания платформы с кластером узлов

## Вариант развертывания 2 (с одиночными узлами)

Архитектура развертывания платформы IAPE с одиночными узлами представлен на рисунке ниже:

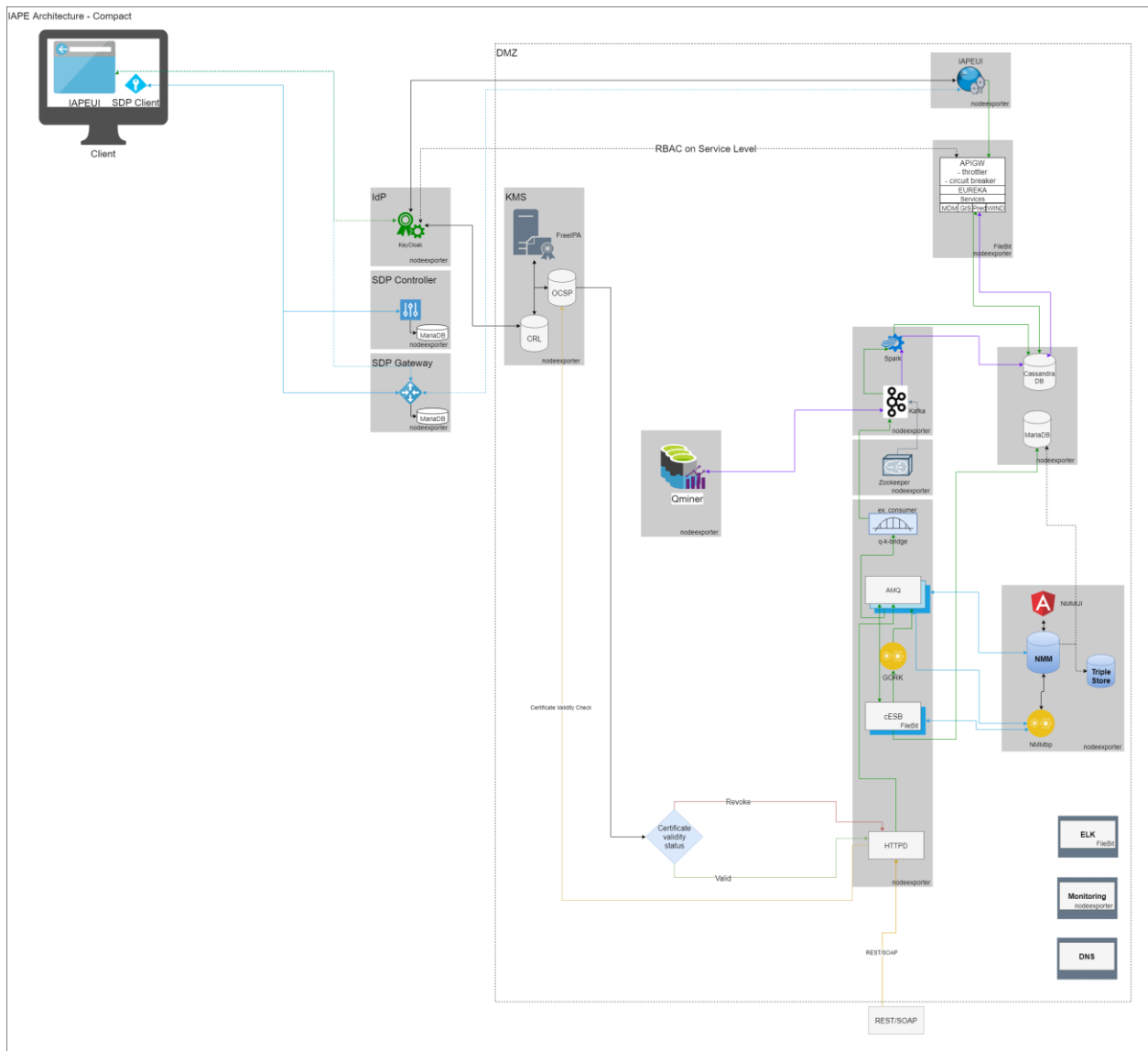


Рис. 2.6. Архитектура развертывания платформы с одиночными узлами

### 3.2. Команды для развертывания платформы IAPE

Команды для развертывания платформы IAPE целиком и ее отдельных компонентов приведены в таблицах ниже. Все они должны выполняться на ansible-машине из директории *“/etc/ansible”*.

**Табл. 1. Команды для развертывания платформы IAPE целиком**

<i>Инсталлируемые компоненты</i>	<i>Команда для развертывания</i>
<i>Все компоненты платформы IAPE для варианта архитектуры с кластером узлов</i>	<code>ansible-playbook pipeline.yml</code>
<i>Все компоненты IAPE для варианта архитектуры с одиночными узлами</i>	<code>ansible-playbook case2.yml</code>

**Табл. 2. Команды для развертывания отдельных компонентов платформы IAPE**

<i>Инсталлируемые компоненты</i>	<i>Команда для развертывания</i>
<i>Шлюз API GW (кластер узлов)</i>	<code>ansible-playbook api_gw.yml</code>
<i>Шлюз API GW (едининый узел)</i>	<code>ansible-playbook iape.apigw01.install.yml</code>
<i>БД Cassandra single node (едининый узел)</i>	<code>ansible-playbook cassandra.yml</code>
<i>БД Cassandra (кластер узлов)</i>	<code>ansible-playbook cassandra-cluster.yml</code>
<i>Репозиторий CIMRepo</i>	<code>ansible-playbook cimrepo.yml</code>
<i>Экземпляр Openstack с необходимыми настройками</i>	<code>ansible-playbook createOSinstance.yml</code>
<i>Eclipse версии MARS</i>	<code>ansible-playbook eclipse-mars.yml</code>
<i>Стек ELK (Elasticsearch, Logstash и Kibana)</i>	<code>ansible-playbook elkstack.yml</code>
<i>Модуль аналитики</i>	<code>ansible-playbook EnergyForecast.yml</code>
<i>ServiceMix (едининый узел)</i>	<code>ansible-playbook esb.yml</code>
<i>FreeIPA (едининый узел)</i>	<code>ansible-playbook freeipa.yml</code>
<i>Kafka+Zookeeper (едининый узел)</i>	<code>ansible-playbook kafka.yml</code>
<i>Kafka (кластер узлов)</i>	<code>ansible-playbook kafkaCluster.yml</code>
<i>Keycloak</i>	<code>ansible-playbook keycloak.yml</code>
<i>Модуль мониторинга</i>	<code>ansible-playbook monitoring.yml</code>
<i>ServiceMix (кластер узлов)</i>	<code>ansible-playbook servicemixCluster.yml</code>
<i>Spark (едининый узел)</i>	<code>ansible-playbook spark.yml</code>
<i>Spark (кластер узлов)</i>	<code>ansible-playbook spark-cluster.yml</code>
<i>Zookeeper (кластер узлов)</i>	<code>ansible-playbook zookeeperCluster.yml</code>
<i>Zeppelin</i>	<code>ansible-playbook zeppelin.yml</code>
<i>Контроллер SDP</i>	<code>ansible-playbook sdp-controller.yml</code>
<i>Шлюз SDP</i>	<code>ansible-playbook sdp-gateway.yml</code>

<i>FreeIPA (обновление версии)</i>	<code>ansible-playbook freeipa_upgrade.yml</code>
<i>Keycloak (обновление версии)</i>	<code>ansible-playbook keycloak_upgrade.yml</code>
<i>HAProxy и Keeralived (кластер узлов)</i>	<code>ansible-playbook haproxy2node.yml</code>
<i>Пользовательский интерфейс IAPE GUI</i>	<code>ansible-playbook iapegui.yml</code>
<i>Docker и QMiner</i>	<code>ansible-playbook iape.docker.qminer.yml</code>
<i>DNS-сервер Bind</i>	<code>ansible-playbook dns.yml</code>
<i>SKZ (Spark, Scala, Kafka)</i>	<code>ansible-playbook iape.skz.cluster.install.yml</code>
<i>Реплика FreeIPA (клиент)</i>	<code>ansible-playbook ipaccli.yml</code>

### 3.3. Проверка работы компонентов платформы IАРЕ

После завершения развертывания пакета ПО для IАРЕ, нужно проверить рабочее состояние всех сервисов согласно представленным ниже инструкциям. Все программное обеспечение будет установлено в директорию `/opt/SI5000/`, за исключением пакетов **FreeIPA**, **HAproxy**, **Keepalived**, сервера **HTTPD**, DNS-сервера **Bind** и **JAVA**, которое устанавливаются в директорию `/etc`.

#### 3.3.1. FreeIPA

Чтобы проверить состояние сервиса IPA на главном сервере **FreeIPA**, используйте команду:  
`service ipa status`

```
[root@ipa ~]# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
ntpd Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

Рис. 2.7. Ответ от сервиса сервера FreeIPA

Веб-интерфейс FreeIPA доступен по адресу в формате:  
<https://сервер.домен/ipa/ui/> (например, <https://ipa.iape.iskrauraltel.ru/ipa/ui/>)



**Username**

**Password**

[Login Using Certificate](#) [Sync OTP Token](#) [Login](#)

- To login with **username and password**, enter them in the corresponding fields, then click Login.
- To login with **Kerberos**, please make sure you have valid tickets (obtainable via kinit) and **configured** the browser correctly, then click Login.
- To login with **certificate**, please make sure you have valid personal certificate.

Рис. 2.8. Главная страница веб-интерфейса сервера FreeIPA

Реквизиты доступа по умолчанию: `admin/Freeipa1234!`

### 3.3.2. Keycloak

Чтобы проверить состояние сервиса на узле Keycloak, используйте команду:

```
service keycloak status
```

```
[centos@l1 ~]$ service keycloak status
Redirecting to /bin/systemctl status keycloak.service
● keycloak.service - Keycloak Service
   Loaded: loaded (/etc/systemd/system/keycloak.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2018-02-12 13:28:16 UTC; 1min 51s ago
   Main PID: 12052 (standalone.sh)
   CGroup: /system.slice/keycloak.service
           └─12052 /bin/sh /etc/keycloak/bin/standalone.sh -b 0.0.0.0
             └─12101 java -D[Standalone] -server -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -Djava.net.preferIPv4Stack=true -Djboss.mo...

Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,531 INFO [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 51) RESTEASY00...plication
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,532 INFO [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 51) RESTEASY00...plication
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,532 INFO [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 51) RESTEASY00...plication
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,532 INFO [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 51) RESTEASY00...plication
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,650 INFO [org.wildfly.extensionundertow] (ServerService Thread Pool -- 51) WFLYUT0021: Registering 'server'
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,703 INFO [org.jboss.as.server] (ServerService Thread Pool -- 45) WFLYSRV0010: Deployed "keycloak-ver.war"
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,777 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0212: Resuming server
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,779 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://0.0.0.0:9990
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,779 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://0.0.0.0:9990
Feb 12 13:28:38 l1-1 standalone.sh[12052]: 13:28:38,779 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: Keycloak 3.4.1.Final (WildFly Core n-demand)
Hint: Some lines were ellipsized, use -l to show in full.
```

Рис. 2.9. Ответ от сервиса Keycloak

Или проверьте состояние через веб-интерфейс, доступный по адресам в формате:

<http://localhost:8080> или

<https://hostname:8443/auth> (например, <https://keycloak.iape.iskrauraltel.ru:8443/auth>)

Реквизиты доступа по умолчанию: *admin/Keycloak1234!*



Примечание. Чтобы получить доступ к веб-интерфейсу, нужно импортировать сертификат FreeIPA в веб-браузер клиента.

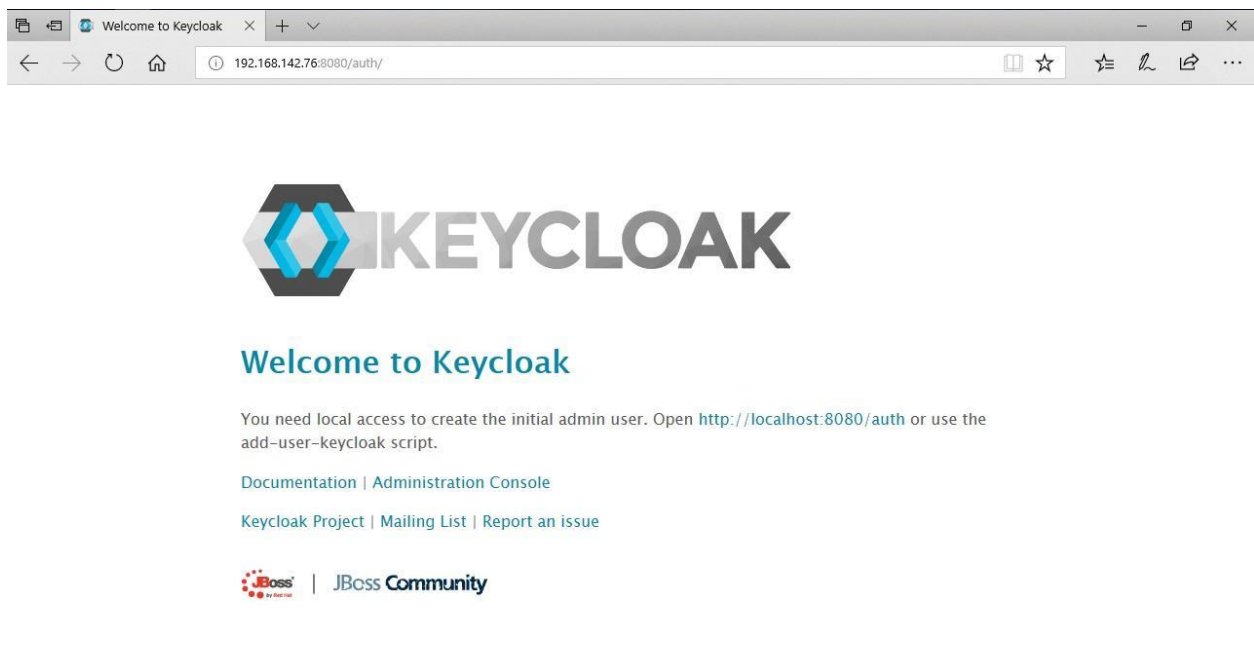


Рис. 2.10. Главная страница Keycloak

### 3.3.3. Cassandra

Чтобы проверить состояние сервиса БД Cassandra, используйте команду:

```
service cassandra.service status
```

```
[centos@l1 system]$ service cassandra.service status
Redirecting to /bin/systemctl status cassandra.service.service
● cassandra.service - Cassandra
   Loaded: loaded (/etc/systemd/system/cassandra.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2018-02-12 13:51:26 UTC; 1min 18s ago
   Main PID: 22432 (java)
   CGroup: /system.slice/cassandra.service
           └─22432 java -ea -javaagent:/etc/cassandra/bin/./lib/jamm-0.3.0.jar -XX:+CMSClassUnloadingEnabled -XX:+UseThreadPriorities -XX:ThreadPriorityP...

Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Loading org.apache.cassandra.config.CFMetaData@flaffc7[cfId=8826e8e9-e16a-3728-8753-3bc1fc71...a.db.mars
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Initializing system traces.events
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Enqueuing flush of local: 594 (0%) on-heap, 0 (0%) off-heap
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Writing Memtable-local@179916508(0.081KiB serialized bytes, 6 ops, 0%/0% of on/off-heap limit)
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Completed flushing /etc/cassandra/bin/./data/data/system/local-7ad54392bccd35a684174e047860...n=102594)
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Enqueuing flush of local: 51462 (0%) on-heap, 0 (0%) off-heap
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Writing Memtable-local@869698778(8.349KiB serialized bytes, 259 ops, 0%/0% of on/off-heap limit)
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Completed flushing /etc/cassandra/bin/./data/data/system/local-7ad54392bccd35a684174e047860...n=114567)
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Compacting [SSTableReader(path='/etc/cassandra/bin/./data/data/system/local-7ad54392bccd35a...ableRead
Feb 12 13:51:32 l-1 cassandra[22389]: INFO 13:51:32 Node localhost/127.0.0.1 state jump to NORMAL
Hint: Some lines were ellipsized, use -l to show in full.
```

Рис. 2.11. Ответ от сервиса cassandra

### 3.3.4. Стек ELK

Чтобы проверить состояние сервисов стека ELK (Elasticsearch, Logstash и Kibana, используйте команды:

```
service elasticsearch.service status
service filebeat.service status
service kibana.service status
service logstash.service status
```

Веб-интерфейс Kibana доступен по адресу в формате:

<http://hostname:5601> (например, <http://elkstack.iape.iskrauraltel.ru:5601>)

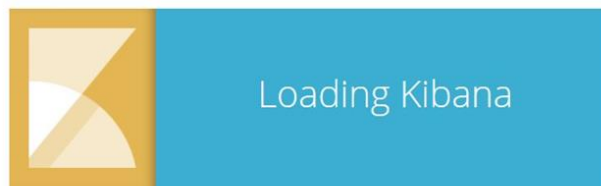
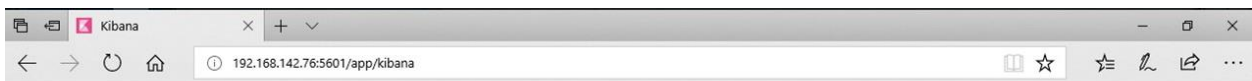


Рис. 2.12. Главная страница веб-интерфейса Kibana

Реквизиты доступа не нужны, вход в веб-интерфейс Kibana будет выполнен автоматически.



### 3.3.5. Компоненты ServiceMix

Чтобы проверить работу сервиса ServiceMix Karaf, используйте команду:

```
systemctl status KARAF-service
```

```
[centos@1-1 ~]$ systemctl status KARAF-service
● KARAF-service.service - SYSV: Karaf
   Loaded: loaded (/etc/rc.d/init.d/KARAF-service; bad; vendor preset: disabled)
   Active: active (exited) since Tue 2018-02-13 07:32:19 UTC; 1min 38s ago
     Docs: man:systemd-sysv-generator(8)

Feb 13 07:32:18 1-1 systemd[1]: Starting SYSV: Karaf...
Feb 13 07:32:19 1-1 KARAF-service[12557]: Starting Karaf...
Feb 13 07:32:19 1-1 systemd[1]: Started SYSV: Karaf.
```

Рис. 2.13. Ответ от сервиса ServiceMix Karaf

Если в работе сервиса обнаружены проблемы, перезапустите шину ESB с помощью команды:

```
service KARAF-service start
/etc/init.d/KARAF-service start
```

или через веб-интерфейс узла ESB, доступный по адресу в формате:

<http://hostname:8181/system/console> (например, <http://esb1.iape.iskrauraltel.ru:8181/system/console>)  
или через VIP-порт (<http://vip.iape.iskrauraltel.ru:8181/system/console/>)

## Apache Karaf Web Console Bundles



Main OSGi Web Console <span style="float: right;">Log out</span>						
Bundle information: 269 bundles in total - all 269 bundles active						
Apply Filter		Filter All		Reload	Instal/Update...	Refresh Packages
Id	Name	Version	Category	Status	Actions	
0	System Bundle ( <i>org.apache.felix.framework</i> )	5.6.2		Active		
34	activeio-core ( <i>org.apache.activemq.activeio-core</i> )	3.1.4		Active		
35	activemq-blueprint ( <i>org.apache.activemq.activemq-blueprint</i> )	5.14.5		Fragment		
36	activemq-camel ( <i>org.apache.activemq.activemq-camel</i> )	5.14.5		Active		
14	activemq-karaf ( <i>activemq-karaf</i> )	5.14.5		Active		
37	activemq-osgi ( <i>org.apache.activemq.activemq-osgi</i> )	5.14.5		Active		
38	activemq-web-console ( <i>org.apache.activemq.activemq-web-console</i> )	5.14.5		Active		
117	Apache Apache HttpClient OSGi bundle ( <i>org.apache.httpcomponents.httpclient</i> )	4.5.1		Active		
118	Apache Apache HttpCore OSGi bundle ( <i>org.apache.httpcomponents.httpcore</i> )	4.4.4		Active		
39	Apache Aries Blueprint API ( <i>org.apache.aries.blueprint.api</i> )	1.0.1		Active		
40	Apache Aries Blueprint CM ( <i>org.apache.aries.blueprint.cm</i> )	1.0.9		Active		
41	Apache Aries Blueprint Core ( <i>org.apache.aries.blueprint.core</i> )	1.7.1		Active		
42	Apache Aries Blueprint Core Compatibility Fragment Bundle ( <i>org.apache.aries.blueprint.core.compatibility</i> )	1.0.0		Fragment		
43	Apache Aries JMX API ( <i>org.apache.aries.jmx.api</i> )	1.1.5		Active		
44	Apache Aries JMX Blueprint API ( <i>org.apache.aries.jmx.blueprint.api</i> )	1.1.5		Active		
45	Apache Aries JMX Blueprint Core ( <i>org.apache.aries.jmx.blueprint.core</i> )	1.1.5		Active		

Рис. 2.14. Главная страница ServiceMix

Реквизиты доступа Karaf по умолчанию: *karaf/karaf*

Веб-интерфейс HAWTIO для ServiceMIX доступен по адресу в формате:

<http://hostname:8181/hawtio> (например, <http://esb1.iape.iskrauraltel.ru:8181/hawtio>)  
или через VIP-порт (<http://vip.iape.iskrauraltel.ru:8181/hawtio>).



Рис. 2.15. Страница входа в HAWTIO

Реквизиты доступа HAWTIO по умолчанию: karaf/karaf

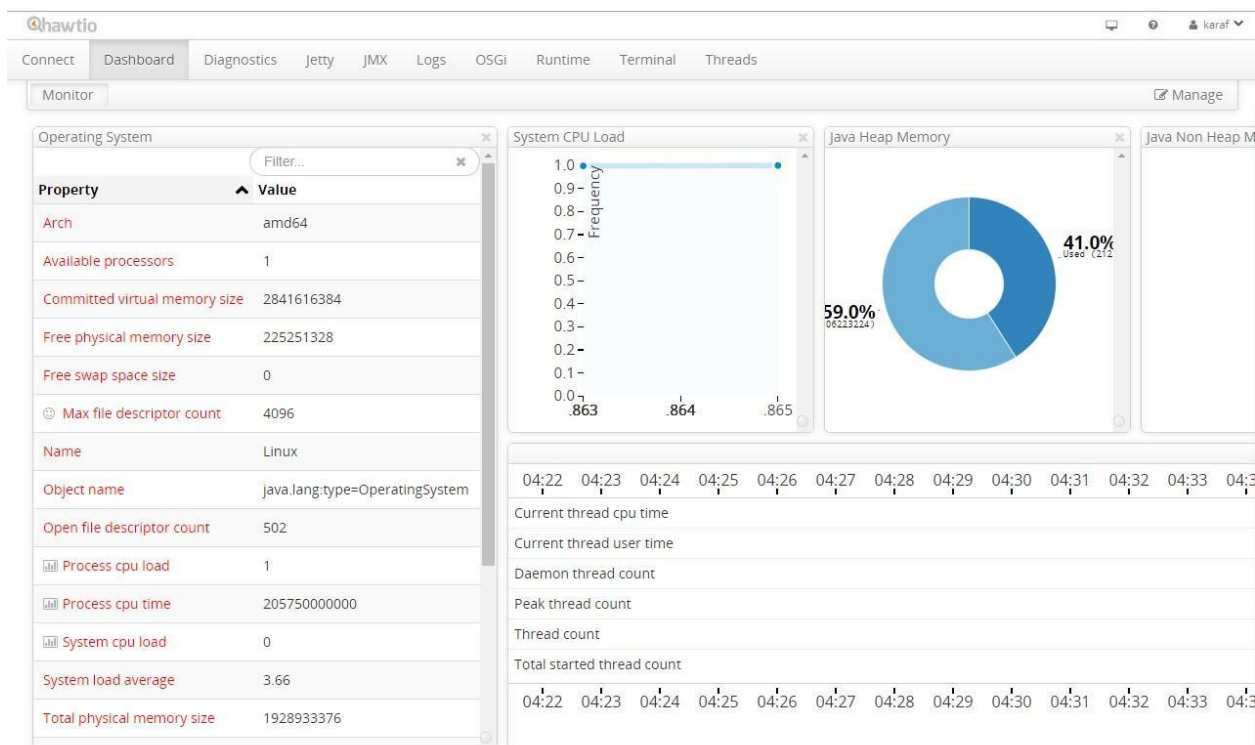


Рис. 2.16. Панель управления HAWTIO

### 3.3.6. Kafka и Zookeeper в архитектуре с одиночными узлами

В случае развертывания варианта решения с одиночными узлами модули Kafka и Zookeeper должны быть установлены на одной и той же узле виртуальной машины. Поэтому плейбук для инсталляции Kafka дополнительно установит на узел еще и Zookeeper, а плейбук для инсталляции Zookeeper дополнительно установит на узел еще и Kafka.

Чтобы проверить состояние сервиса Kafka, используйте команду:

```
# service kafka status
```

```
[root@kafka centos]# service kafka status
Redirecting to /bin/systemctl status kafka.service
● kafka.service - Apache Kafka server (broker)
   Loaded: loaded (/etc/systemd/system/kafka.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-02-13 14:11:52 UTC; 3h 18min ago
     Docs: http://kafka.apache.org/documentation.html
   Main PID: 13277 (java)
   CGroup: /system.slice/kafka.service
           └─13277 /usr/java/jdk1.8.0_121/bin/java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35 -XX:+DisableExplicitGC -Djava.awt.headless=true -Xloggc:/etc/kafka/bin/./logs/kafkaSe...

Feb 13 15:53:54 kafka kafka-server-start.sh[13277]: [2018-02-13 15:53:54,873] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 16:03:54 kafka kafka-server-start.sh[13277]: [2018-02-13 16:03:54,872] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 16:13:54 kafka kafka-server-start.sh[13277]: [2018-02-13 16:13:54,873] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 16:23:54 kafka kafka-server-start.sh[13277]: [2018-02-13 16:23:54,872] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 16:33:54 kafka kafka-server-start.sh[13277]: [2018-02-13 16:33:54,873] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 16:43:54 kafka kafka-server-start.sh[13277]: [2018-02-13 16:43:54,873] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 16:53:54 kafka kafka-server-start.sh[13277]: [2018-02-13 16:53:54,872] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 17:03:54 kafka kafka-server-start.sh[13277]: [2018-02-13 17:03:54,872] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 17:13:54 kafka kafka-server-start.sh[13277]: [2018-02-13 17:13:54,872] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
Feb 13 17:23:54 kafka kafka-server-start.sh[13277]: [2018-02-13 17:23:54,872] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
```

Рис. 2.17. Ответ от сервиса Kafka

Чтобы проверить состояние сервиса Zookeeper, используйте команду:

```
# service zookeeper status
```

```
Redirecting to /bin/systemctl status zookeeper.service
● zookeeper.service - Apache Zookeeper server
   Loaded: loaded (/etc/systemd/system/zookeeper.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-02-13 09:23:33 UTC; 4min 1s ago
     Docs: http://zookeeper.apache.org
   Main PID: 23358 (java)
   CGroup: /system.slice/zookeeper.service
           └─23358 java -Dzookeeper.log.dir=. -Dzookeeper.root.logger=INFO,CONSOLE -cp /etc/zookeeper/bin/./build/classes:/etc/zookeeper/bin/./build/lib...

Feb 13 09:23:32 1-1 systemd[1]: Starting Apache Zookeeper server...
Feb 13 09:23:32 1-1 zkServer.sh[23348]: ZooKeeper JMX enabled by default
Feb 13 09:23:32 1-1 zkServer.sh[23348]: Using config: /etc/zookeeper/bin/./conf/zoo.cfg
Feb 13 09:23:33 1-1 zkServer.sh[23348]: Starting zookeeper ... STARTED
Feb 13 09:23:33 1-1 systemd[1]: Started Apache Zookeeper server.
```

Рис. 2.18. Ответ от сервиса Zookeeper

### 3.3.7. Модули мониторинга

Чтобы получить статус экспортера узлов, используйте команду:

```
#systemctl status node_exporter.service
```

Чтобы получить статус jmx exporter, используйте команду:

```
#systemctl status jmx_exporter
```

Чтобы получить статус сервиса prometheus, используйте команду:

```
#systemctl status prometheus.service
```

Веб-интерфейс Prometheus доступен по адресу в формате:

<http://hostname:9090/graph> (например, <http://monitoring.iape.iskrauratel.ru:9090/graph>)

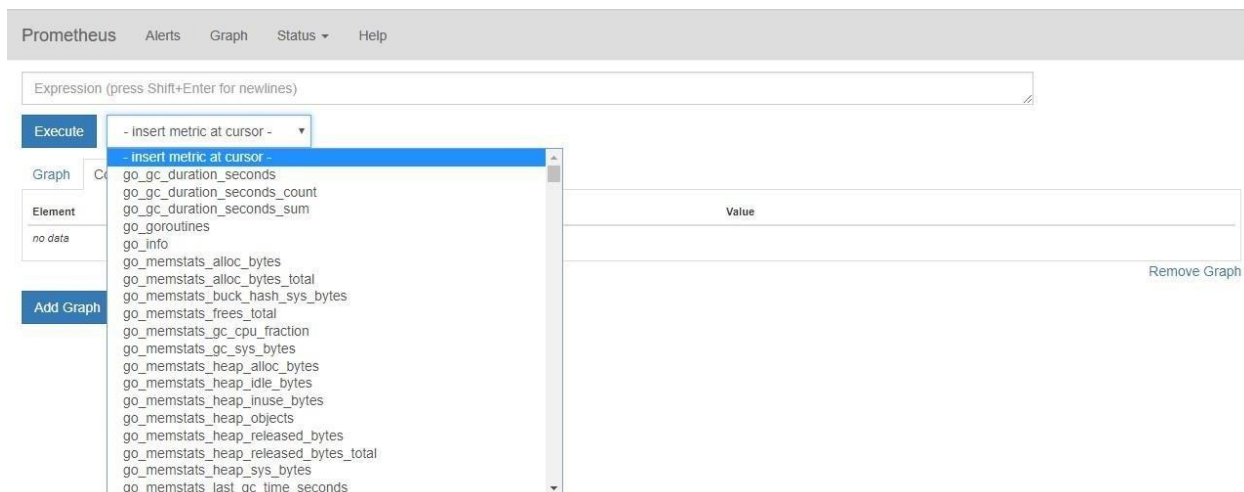


Рис. 2.19. Главная страница Prometheus

Чтобы получить статус Grafana, используйте команду:  
`#systemctl status grafana-server.service`

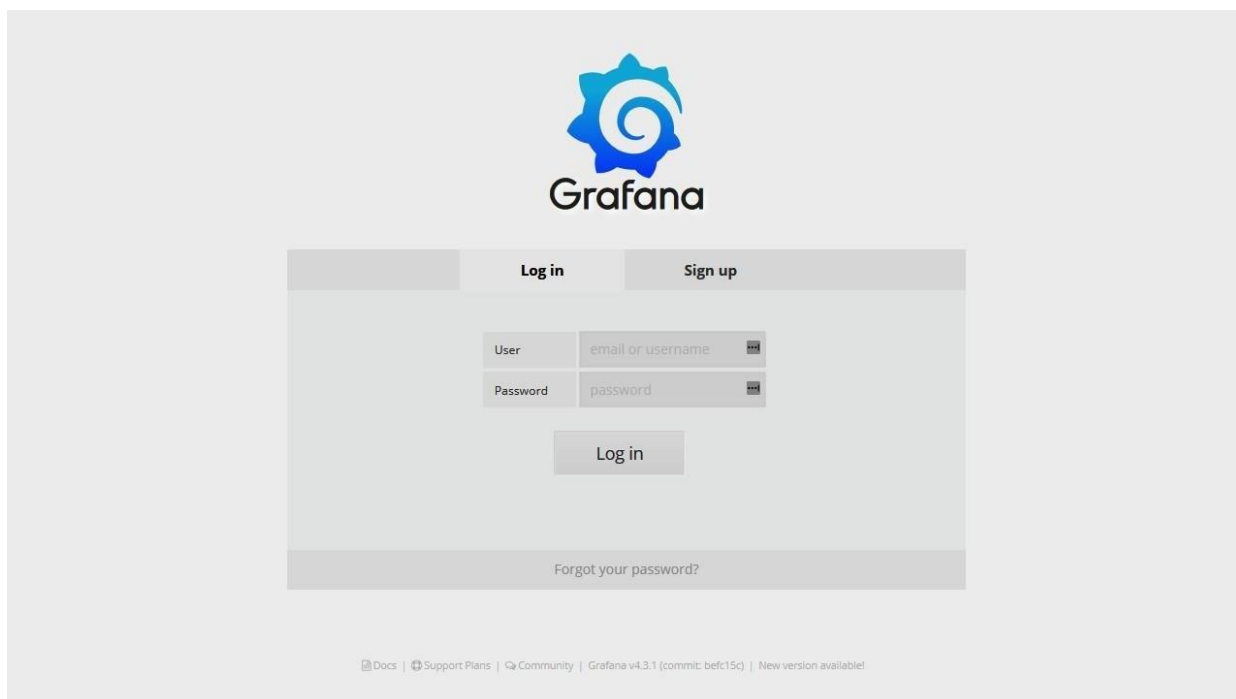


Рис. 2.20. Главная страница Grafana

Веб-интерфейс узла шлюза SDP доступен по адресу в формате:  
<https://sdp-gateway.iape.iskrauraltel.ru/grafana/>

Реквизиты доступа по умолчанию: *admin/admin*



Примечание. Перед запуском Grafana пропишите в таблицах файервола на шлюзе SPD IP-адрес клиента, который используется для запуска Grafana.

### 3.3.8. Узел Spark

Чтобы получить статус узла spark, используйте команду:

```
# systemctl status spark.service
```

```
[centos@1-1 spark-1.6.1]$ service spark status
Redirecting to /bin/systemctl status spark.service
● spark.service - Spark Master node
   Loaded: loaded (/etc/systemd/system/spark.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-02-13 12:11:12 UTC; 2min 58s ago
   Main PID: 2554 (java)
   CGroup: /system.slice/spark.service
           └─2554 /usr/java/jdk1.8.0_121/jre/bin/java -cp /etc/spark/conf/:/etc/spark/assembly/target/scala-2.10/spark-assembly-1.6.1-hadoop2.2.0.jar:/etc...

Feb 13 12:11:10 1-1 systemd[1]: Starting Spark Master node...
Feb 13 12:11:10 1-1 start-master.sh[2536]: starting org.apache.spark.deploy.master.Master, logging to /etc/spark/logs/spark-root-org.apache.spark...1-1-1.out
Feb 13 12:11:12 1-1 systemd[1]: Started Spark Master node.
Hint: Some lines were ellipsized, use -l to show in full.
```

Рис. 2.21. Ответ от сервиса Spark

Веб-интерфейс главного узла доступен по адресу в формате:

<http://hostname:8080> (например: <http://spark01.iape.iskrauraltel.ru:8080>)

**Spark 1.6.1 Spark Master at spark://spark01:7077**

URL: spark://spark01:7077  
 REST URL: spark://spark01:6066 (cluster mode)  
 Alive Workers: 2  
 Cores in use: 2 Total, 0 Used  
 Memory in use: 13.3 GB Total, 0.0 B Used  
 Applications: 0 Running, 0 Completed  
 Drivers: 0 Running, 0 Completed  
 Status: ALIVE

**Workers**

Worker Id	Address	State	Cores	Memory
worker-20180516075533-192.168.5.60-46629	192.168.5.60:46629	ALIVE	1 (0 Used)	6.6 GB (0.0 B Used)
worker-20180516080842-192.168.5.63-37721	192.168.5.63:37721	ALIVE	1 (0 Used)	6.6 GB (0.0 B Used)

**Running Applications**

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
No running applications.							

**Completed Applications**

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
No completed applications.							

Рис. 2.22. Веб-интерфейс главного узла Spark

Веб-интерфейс подчиненного узла spark02 доступен по адресу:

<http://spark02.iape.iskrauraltel.ru:8081/>

Веб-интерфейс подчиненного узла spark03 доступен по адресу:

<http://spark03.iape.iskrauraltel.ru:8081/>

**Рис. 2.23. Веб-интерфейс подчиненного узла Spark**

Чтобы получить статус узла только сервисов SparkMeters на первом узле SKZ, используйте команды:

```
# service gis-streamer.service status
# service historic-streamer.service status
# service prediction-streamer.service status
```

### 3.3.9. Интерфейс пользования IAPE GUI

#### Предварительные условия

Чтобы получить доступ к пользовательскому интерфейсу IAPE, нужно указать параметры веб-доступа к IAPE GUI в конфигурации Keusloak. Рекомендуется указать сетевое имя интерфейса, а не IP-адрес.



Примечание. Чтобы получить доступ к веб-интерфейсу, нужно импортировать сертификат FreeIPA в веб-браузер клиента.

Пользовательский веб-интерфейс доступен по адресу:

<https://sdp-gateway.iape.iskrauraltel.ru/orin>

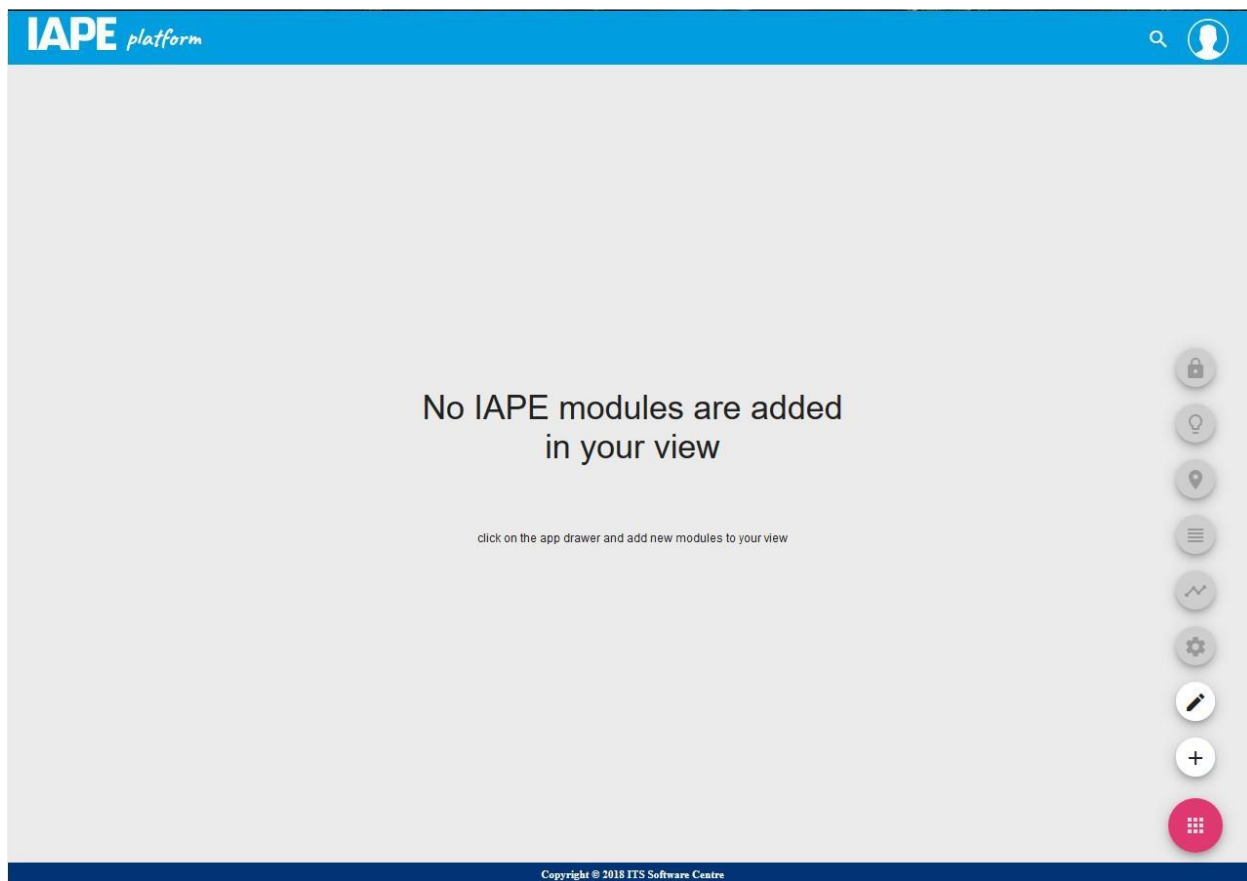


Рис. 2.24. Главная страница IAPE GUI

Реквизиты доступа по умолчанию: *angular-root / Iskratel@1234*

Веб-интерфейс Модуля аналитики доступен по адресу:

<https://sdp-gateway.iape.iskrauraltel.ru/analytics>

Реквизиты доступа по умолчанию: *angular-root / Iskratel@1234*

### 3.3.10. HAProxy, Keepalived и HTTPD

Чтобы проверить работу сервисов HAProxy, Keepalived и HTTPD, используйте команды:

```
# service httpd status  
# service haproxy status  
# service keepalived status
```

```
[root@api-gw centos]# service haproxy status
Redirecting to /bin/systemctl status haproxy.service
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-02-13 13:48:20 UTC; 16s ago
   Main PID: 5994 (haproxy-systemd)
   CGroup: /system.slice/haproxy.service
           └─5994 /usr/sbin/haproxy-systemd-wrapper -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid
             └─5998 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds
               └─6010 /usr/sbin/haproxy -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -Ds

Feb 13 13:48:20 api-gw.iape.iskrauraltel.ru systemd[1]: Started HAProxy Load Balancer.
Feb 13 13:48:20 api-gw.iape.iskrauraltel.ru systemd[1]: Starting HAProxy Load Balancer...
Feb 13 13:48:20 api-gw.iape.iskrauraltel.ru haproxy-systemd-wrapper[5994]: haproxy-systemd-wrapper: executing /usr/sbin/haproxy -f /etc/haproxy/hapr...id -Ds
Hint: Some lines were ellipsized, use -l to show in full.
```

Рис. 2.25. Ответ с данными о состоянии от сервиса Нароуху

```
[root@api-gw centos]# service keepalived status
Redirecting to /bin/systemctl status keepalived.service
● keepalived.service - LVS and VRRP High Availability Monitor
   Loaded: loaded (/usr/lib/systemd/system/keepalived.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-02-13 13:48:14 UTC; lmin 24s ago
   Process: 5916 ExecStart=/usr/sbin/keepalived $KEEPALIVED_OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 5917 (keepalived)
   CGroup: /system.slice/keepalived.service
           └─5917 /usr/sbin/keepalived -D
             └─5918 /usr/sbin/keepalived -D
               └─5919 /usr/sbin/keepalived -D

Feb 13 13:48:14 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Initializing ipvs 2.6
Feb 13 13:48:14 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Netlink reflector reports IP 192.168.5.16 added
Feb 13 13:48:14 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Netlink reflector reports IP fe80::f816:3eff:feee:6a7f added
Feb 13 13:48:14 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Registering Kernel netlink reflector
Feb 13 13:48:14 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Registering Kernel netlink command channel
Feb 13 13:48:14 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Opening file '/etc/keepalived/keepalived.conf'.
Feb 13 13:48:14 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Configuration is using : 6103 Bytes
Feb 13 13:48:14 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Using LinkWatch kernel netlink reflector...
Feb 13 13:48:31 api-gw.iape.iskrauraltel.ru Keepalived_healthcheckers[5918]: Netlink reflector reports IP 192.168.5.16 added
Feb 13 13:48:31 api-gw.iape.iskrauraltel.ru Keepalived_vrrp[5919]: Netlink reflector reports IP 192.168.5.16 added
```

Рис. 2.26. Ответ с данными о состоянии от сервиса Keepalived

### 3.3.11. Шлюз API GW

Чтобы проверить все сервисы шлюза API GW, используйте команды:

```
# service gis-translator.service status
# service mdma-translator.service status
# service aggregation.service status
# service expanded.service status
# service graphql.service status
# service measurements.service status
# service snmp.service status
# service eureka.service status
# service real-time.service status
# service compoiste-backend-ui.service status
# service apigw.service status
# service ifttt.service status
# service gis-rest.service status
# service pred-rest.service status
# service widget-rest.service status
# service subscription-graphql.service status
# service pki-app.service status
```



Примечание. Чтобы получить доступ к веб-интерфейсу, нужно импортировать сертификат FreeIPA в веб-браузер клиента.

Адрес api-gw01: <https://api-gw.iape.iskrauraltel.ru:8787/api>

Адрес api-gw02: <https://api-gw-sec.iape.iskrauraltel.ru:8787/api>

Реквизиты доступа по умолчанию: `angular-root / Iskratel@1234`



Адрес Swagger01: <https://api-gw.iape.iskrauraltel.ru:8787/swagger-ui.html>

Адрес Swagger02: <https://api-gw-sec.iape.iskrauraltel.ru:8787/swagger-ui.html>



Рис. 2.27. Веб-интерфейс Swagger

Адрес Eureka01: <http://api-gw.iape.iskrauraltel.ru:8761/>

Адрес Eureka02: <http://api-gw-sec.iape.iskrauraltel.ru:8761/>

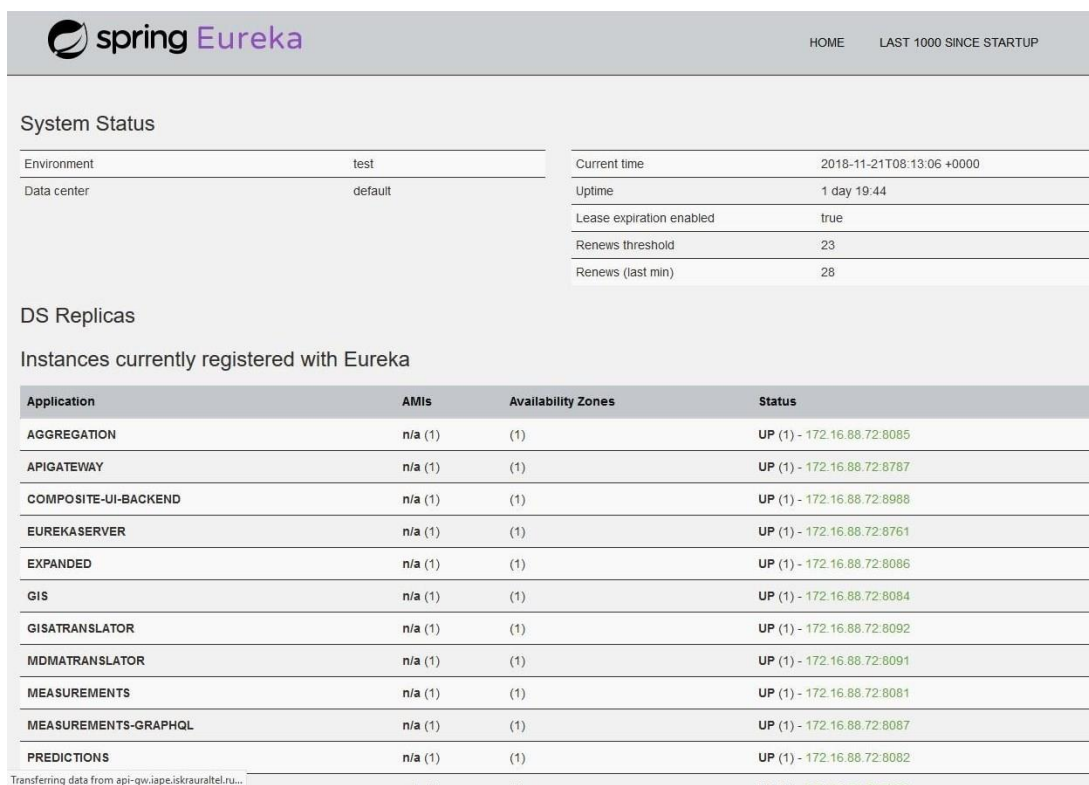


Рис. 2.28. Веб-интерфейс Eureka

### 3.3.12. Веб-интерфейс репозитория CIMгеро

Чтобы проверить работу всех сервисов репозитория CIMгеро, используйте команды:

```
#systemctl status NMM-IS.service
```

```

#systemctl status NMM-IS-SNAPSHOT.service
#systemctl status NMM-LS.service
#systemctl status NMM-LS-SNAPSHOT.service
#systemctl status NMM-ORS.service
#systemctl status NMM-ORS-SNAPSHOT.service
#systemctl status rbac.service
# service nginx status

[root@cimrepo ~]# service nginx status
Redirecting to /bin/systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2018-04-17 14:58:41 UTC; 1 day 16h ago
     Process: 29201 ExecReload=/bin/kill -s HUP $MAINPID (code=exited, status=0/SUCCESS)
     Process: 26543 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
     Process: 26541 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 26539 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
  Main PID: 26546 (nginx)
   CGroup: /system.slice/nginx.service
           └─26546 nginx: master process /usr/sbin/nginx
             └─29205 nginx: worker process

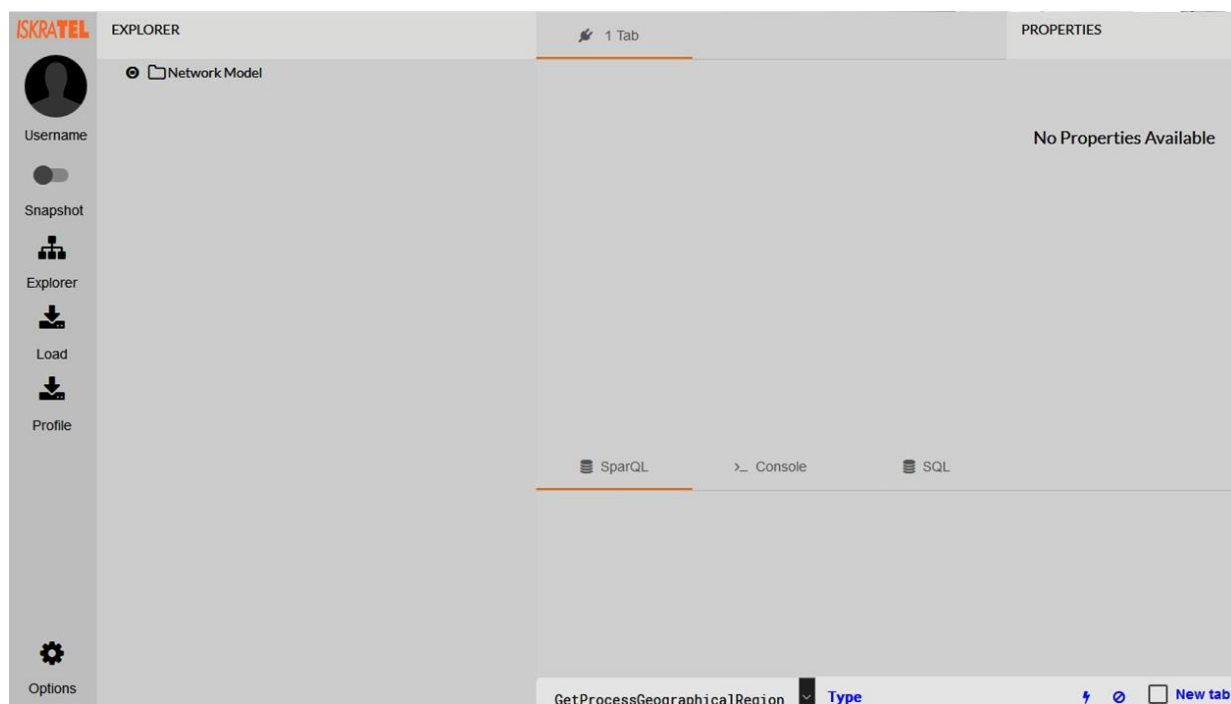
Apr 17 14:58:40 cimrepo systemd[1]: Starting The nginx HTTP and reverse proxy server...
Apr 17 14:58:40 cimrepo nginx[26541]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Apr 17 14:58:40 cimrepo nginx[26541]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Apr 17 14:58:41 cimrepo systemd[1]: Failed to read PID from file /run/nginx.pid: Invalid argument
Apr 17 14:58:41 cimrepo systemd[1]: Started The nginx HTTP and reverse proxy server.
Apr 17 15:11:14 cimrepo systemd[1]: Reloaded The nginx HTTP and reverse proxy server.

```

**Рис. 2.29. Ответ с данными о состоянии от сервиса nginx service**

Веб-интерфейс репозитория CIMrepo доступен по адресу в формате:  
[http://<IP\\_узла\\_CIM>:81](http://<IP_узла_CIM>:81) (например, <http://cimrepo01.iape.iskrauraltel.ru:81>).

Реквизиты доступа по умолчанию: *admin@iskratel.com/iskratel*



**Рис. 2.30. Главная страница веб-интерфейса СИМгеро**



**Предупреждение!** Узел СИМгеро использует два разных доменных имени в среде Openstack: одно для внутреннего интерфейса (simpero), а другое для плавающего интерфейса (simpero01). Чтобы получить доступ к веб-интерфейсу, используйте доменное имя для плавающего интерфейса.

### 3.3.13. Узлы в кластере Mariadb-Galera

Кластер Galera (MariaDB) устанавливается на узлы Cassandra. Нужно проверить состояние сервисов и их принадлежность к кластеру.

Чтобы получить статус кластера Galera, используйте команду:

```
# service mariadb.service status
```

```
[root@cassandra03 ~]# service mariadb status
Redirecting to /bin/systemctl status mariadb.service
● mariadb.service - MariaDB 10.1.33 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/mariadb.service.d
           └─migrated-from-my.cnf-settings.conf
   Active: active (running) since Tue 2018-11-13 11:50:31 UTC; 1 weeks 0 days ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 17875 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 17656 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= [ | VAR="/usr/bin/galera_recovery"; [ $? -eq 0 ] && systemctl set-envir
onment _WSREP_START_POSITION=$VAR || exit 1 (code=exited, status=0/SUCCESS)
   Process: 17653 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
  Main PID: 17758 (mysqld)
    Status: "Taking your SQL requests now..."
   CGroup: /system.slice/mariadb.service
           └─17758 /usr/sbin/mysqld --wsrep_start_position=00000000-0000-0000-0000-000000000000:-1

Nov 13 11:50:31 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 11:50:31 140122227652864 [Note] /usr/sbin/mysqld: ready for connections.
Nov 13 11:50:31 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 11:50:31 140121915516672 [Note] WSREP: Member 0.0 (db1) synced with group.
Nov 13 11:50:31 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 11:50:31 140121915516672 [Note] WSREP: Shifting JOINED -> SYNCED (TO: 0)
Nov 13 11:50:31 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 11:50:31 140122227333888 [Note] WSREP: Synchronized with group, ready fo...ections
Nov 13 11:50:31 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 11:50:31 140122227333888 [Note] WSREP: wsrep_notify_cmd is not defined, ...cation.
Nov 13 11:50:31 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 11:50:31 140122227333888 [Note] WSREP: wsrep_notify_cmd is not defined, ...cation.
Nov 13 12:57:05 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 12:57:05 140122227333888 [ERROR] Slave SQL: Error 'Table 'request_status' alrea...
Nov 13 12:57:05 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 12:57:05 140122227333888 [Warning] WSREP: RBR event 1 Query apply warning: 1, 12
Nov 13 12:57:05 cassandra03.iape.iskrauraltel.ru mysqld[17758]: 2018-11-13 12:57:05 140122227333888 [Warning] WSREP: Ignoring error for TO isolated...489249)
Hint: Some lines were ellipsized, use -l to show in full.
[root@cassandra03 ~]#
```

Рис. 2.31. Ответ с данными о статусе Mariadb

Чтобы получить подробные сведения о кластере, используйте команду:

```
#mysql -uroot -p -e "show status like 'wsrep%'"
```

```
[root@cassandra03 ~]# mysql -uroot -p -e "show status like 'wsrep%'"
Enter password:
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_apply_oooe | 0.000000 |
| wsrep_apply_ool | 0.000000 |
| wsrep_apply_window | 1.000000 |
| wsrep_causal_reads | 0 |
| wsrep_cert_deps_distance | 1.000000 |
| wsrep_cert_index_size | 4 |
| wsrep_cert_interval | 0.000000 |
| wsrep_cluster_conf_id | 3 |
| wsrep_cluster_size | 3 |
| wsrep_cluster_state_uuid | 7056e9ea-e736-11e8-90aa-3f49b3e406fb |
| wsrep_cluster_status | Primary |
| wsrep_commit_oooe | 0.000000 |
| wsrep_commit_ool | 0.000000 |
| wsrep_commit_window | 1.000000 |
| wsrep_connected | ON |
| wsrep_desync_count | 0 |
| wsrep_ews_delayed | |
| wsrep_ews_evict_list | |
| wsrep_ews_repl_latency | 0/0/0/0/0 |
| wsrep_ews_state | OPERATIONAL |
| wsrep_flow_control_paused | 0.000000 |
| wsrep_flow_control_paused_ns | 0 |
| wsrep_flow_control_recv | 0 |
| wsrep_flow_control_sent | 0 |
| wsrep_gcomm_uuid | 5063ala0-e73a-11e8-bf90-f2b393f3ff58 |
| wsrep_incoming_addresses | 172.16.88.59:3306,172.16.88.58:3306,172.16.88.57:3306 |
+-----+-----+
```

Рис. 2.32. Информация о размере кластера

Реквизиты доступа по умолчанию для MariaDB: *root/iskratel*

### 3.3.14. Интерфейс шлюза SDP

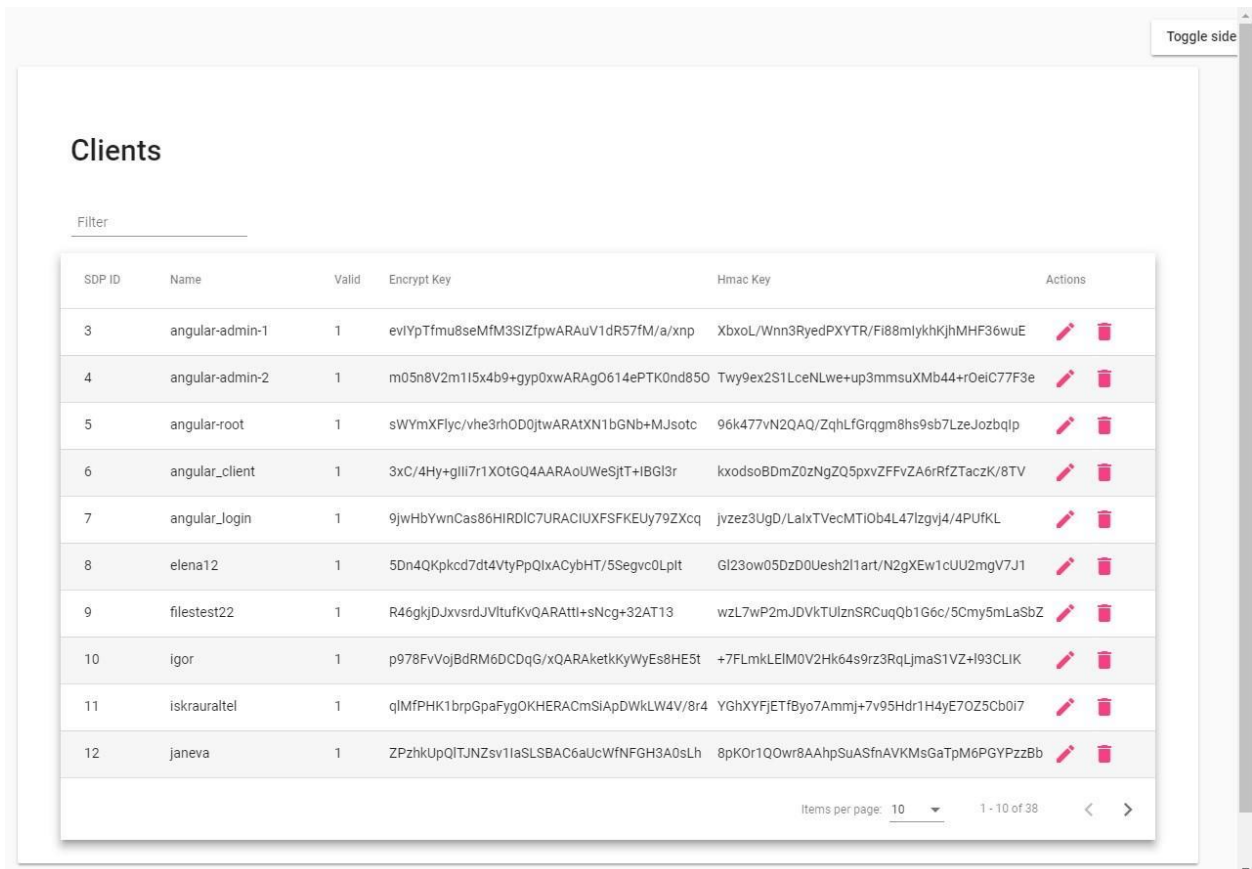
Чтобы проверить состояние сервисов SDP, используйте команду:

```
# service sdp-gateway.service status
# service sdpmonitoring.service status
```



Примечание. Перед запуском веб-интерфейса SDP пропишите в таблицу файервола на шлюзе SPD IP-адрес клиента, который используется для запуска веб-интерфейса.

Веб-интерфейс шлюза SDP доступен по адресу: <https://sdp-gateway.iape.iskrauraltel.ru:8443>



The screenshot displays the 'Clients' page of the SDP Gateway web interface. It features a table with columns for SDP ID, Name, Valid status, Encrypt Key, Hmac Key, and Actions. The table lists 12 clients, including 'angular-admin-1', 'angular-admin-2', 'angular-root', 'angular\_client', 'angular\_login', 'elena12', 'filestest22', 'igor', 'iskrauraltel', and 'janeva'. Each row includes edit and delete icons. A filter input is located above the table, and pagination controls at the bottom indicate 'Items per page: 10' and '1 - 10 of 38'.

SDP ID	Name	Valid	Encrypt Key	Hmac Key	Actions
3	angular-admin-1	1	evYpTfmu8seMfM3SIZfpwARAUv1dR57fM/a/xnp	XbxoL/Wnn3RyedPXyTR/Fi8mlykhKjhMHF36wuE	
4	angular-admin-2	1	m05n8V2m115x4b9+gyp0xwARAgO614ePTK0nd850	Twy9ex2S1LceNLwe+up3mmsuXMB44+rOeiC77F3e	
5	angular-root	1	sWYmXFlyc/vhe3rhOD0JtwARAtXN1bGNb+MJsotc	96k477vN2QAQ/ZqhlFGrqgm8hs9sb7LzeJozbqlp	
6	angular_client	1	3xC/4Hy+glll7r1X0tGQ4AARAoUWeSjtT+IBGI3r	kxodsoBDmZ0zNgZQ5pxvZFFvZA6rRfZTacZK/8TV	
7	angular_login	1	9jwHbYwnCas86HIRD7URACIUxFSFKELy79Zxcq	jvzez3UgD/LalxTVecMTIOb4L47lZgvj4/4PUfKL	
8	elena12	1	5Dn4QKpkcd7dt4VtyPpQixACyBHT/5Segvc0Lplt	Gl23ow05DzD0Uesh2l1art/N2gXEw1cUU2mgV7J1	
9	filestest22	1	R46gkjDjxvsrdJVitufkvQARAtti+sNcg+32AT13	wzL7wP2mJDVktUlznSRCuqQb1G6c/5Cmy5mLaSbZ	
10	igor	1	p978FvVojBdRM6DCDqG/xQARAKetkKyWyEs8HE5t	+7FLmkLEIM0V2Hk64s9rz3RqLjmaS1VZ+I93CLIK	
11	iskrauraltel	1	qIMfPHK1brpGpaFygOKHERACmSIApDWkLW4V/8r4	YGhXYFJETfByo7Ammj+7v95Hdr1H4yE7OZ5Cb0i7	
12	janeva	1	ZPzhkUpQITJNZsv11aSLSBAC6aUcWfNFGH3A0sLh	8pKOr1QOwr8AAhpSuASfnAVKMsGaTpM6PGYPzzBb	

Рис. 2.33. Главная страница веб-интерфейса SDP

**АО «Искра Технологии»**

620066, г. Екатеринбург, ул. Комвузовская, 9а  
тел.: +7 343 210 69 51  
факс: +7 343 341 52 40

эл. почта: [jut@iskratechno.ru](mailto:jut@iskratechno.ru)  
[www.iskratechno.ru](http://www.iskratechno.ru)