Kerberos-

Astra Linux

Astra Linux Directory

Related links

- https://help.ubuntu.com/community/Kerberos
- : http://k5wiki.kerberos.org/wiki/Pkinit_configuration

- Key Distribution Center (KDC) -
- Admin server - kerberos. KDC admin server
- Realm - "",
- Principal - , .

Kerberos - **Key Distribution Center (KDC)**, . Kerberos, , KDC. , realm.

realm . , realm . , example.com EXAMPLE.COM realm.

, realm Kerberos . realm . .

Host Names

Kerberos realm **Fully Qualified Domain Name (FQDN)**.

Kerberos , FQDN reverse-resolvable. IP , rdns false krb5.conf

Active Directory DNS, Active Directory Domain Controller DNS. , FQDN , .

FQDN, forward reverse :

```
$ nslookup server.example.com
$ nslookup <server ip address>
```

Astra Linux ( ), nslookup, dnsutils.

Synaptic Package Manager $ apt-get install dnsutils

IP . FQDN .

FQDN DNS , hosts (/etc) :

```
127.0.0.1 server.aktiv-test.ru localhost server
```

```
<IP-address> server.aktiv-test.ru <IP-address> server
```

IP-address - IP . 10.0.0.1.

DNS nslookup .

ping FQDN:

```
$ ping server.aktiv-test.ru
PING server.aktiv-test.ru (10.0.0.1) 56(84) bytes of data.
64 bytes from server.aktiv-test.ru (10.0.0.1): icmp_seq=1 ttl=128 time=0.176ms
```

ping IP FQDN, . , .

ping .

Kerberos : , . - **Network Time Protocol (NTP)** . , , Astra Linux - NTP-. NTP- (, UbuntuTime Ubuntu).

Active Directory Domain Controllers NTP .

, Kerberos . Kerberos System Administration Manual , .

1. USB- .
2. USB- :

```
$ lsusb
```

USB-:

```
[dmitrieva@localhost ~]$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 004: ID 0a89:0030 Aktiv Rutoken ECP
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

, : **Aktiv Rutoken ECP**

- Ubuntu

: . .

- `<username> = testuser`
- `<realm> = AKTIV-TEST`
- `<server> = aktiv-test.ru`

- `krb5-kdc, krb5-admin-server, krb5-pkinit`
- Kerberos realm: `AKTIV-TEST`, `aktiv-test.ru` (/etc/hosts)
  :
- : `testuser@AKTIV-TEST`

- `krb5-user, libpam-krb5, libpam-ccreds, auth-client-config, krb5-pkinit, opensc, libengine-pkcs11-openssl`
- default realm: `AKTIV-TEST`
- (kdc, admin) IP- (/etc/hosts)

realm

```
$ sudo apt-get install krb5-kdc krb5-admin-server krb5-pkinit
#   :
# realm = AKTIV-TEST
#  = aktiv-test.ru
$ sudo krb5_newrealm
#
```

/etc/krb5.conf

/etc/krb5.conf

```
[domain_realm]
        .aktiv-test.ru = AKTIV-TEST
        aktiv-test.ru = AKTIV-TEST
```

```
$ sudo kadmin.local
# username = testuser
# password = test
kadmin.local:$ addprinc <username>
# ...
kadmin.local:$ quit
```

,

```
$ kinit <username>
...
$ klist
...
$ kdestroy
```

rtengine openssl sdk

```
$ wget https://download.rutoken.ru/Rutoken/SDK/sdk-180919-80c054.zip
$ unzip -q sdk-180919-80c054.zip
$ sudo cp -P sdk/openssl/rtengine/bin/linux_glibc-x86_64/lib/librtengine.so* /usr/lib/x86_64-linux-gnu/engines-
1.1/
```

pkcs11 rtpkcs11ecp.so

kerberos

```
$ sudo apt-get install krb5-user libpam-krb5 libpam-ccreds auth-client-config krb5-pkinit opensc libengine-
pkcs11-openssl
#   :
# realm = AKTIV-TEST
#  = aktiv-test.ru
$ sudo dpkg-reconfigure krb5-config
```

/etc/krb5.conf

/etc/krb5.conf

```
[domain_realm]
        ...
        .aktiv-test.ru = AKTIV-TEST
        aktiv-test.ru = AKTIV-TEST
```

,

```
$ kinit <username>
...
$ klist
...
$ kdestroy
```

```
$ openssl genrsa -out cakey.pem 2048
$ openssl req -key cakey.pem -new -x509 -out cacert.pem
```

pkinit_extensions

pkinit_extensions

```
[ kdc_cert ]
basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, keyAgreement

#Pkinit EKU
extendedKeyUsage = 1.3.6.1.5.2.3.5

subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

# Copy subject details

issuerAltName=issuer:copy

# Add id-pkinit-san (pkinit subjectAlternativeName)
subjectAltName=otherName:1.3.6.1.5.2.2;SEQUENCE:kdc_princ_name

[kdc_princ_name]
realm = EXP:0, GeneralString:${ENV::REALM}
principal_name = EXP:1, SEQUENCE:kdc_principal_seq

[kdc_principal_seq]
name_type = EXP:0, INTEGER:1
name_string = EXP:1, SEQUENCE:kdc_principals

[kdc_principals]
princ1 = GeneralString:krbtgt
princ2 = GeneralString:${ENV::REALM}

[ client_cert ]

# These extensions are added when 'ca' signs a request.

basicConstraints=CA:FALSE

keyUsage = digitalSignature, keyEncipherment, keyAgreement

extendedKeyUsage =  1.3.6.1.5.2.3.4
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer


subjectAltName=otherName:1.3.6.1.5.2.2;SEQUENCE:princ_name


# Copy subject details

issuerAltName=issuer:copy

[princ_name]
realm = EXP:0, GeneralString:${ENV::REALM}
principal_name = EXP:1, SEQUENCE:principal_seq

[principal_seq]
name_type = EXP:0, INTEGER:1
name_string = EXP:1, SEQUENCE:principals

[principals]
princ1 = GeneralString:${ENV::CLIENT}
```

KDC

```
$ openssl genrsa -out kdckey.pem 2048
#
$ openssl req -new -out kdc.req -key kdckey.pem
#
$ REALM=<realm>; export REALM
$ CLIENT=<server>; export CLIENT
#   pkinit_extensions
$ openssl x509 -req -in kdc.req -CAkey cakey.pem -CA cacert.pem -out kdc.pem -extfile pkinit_extensions -
extensions kdc_cert -CAcreateserial
```

kdc.pem, kdckey.pem, cacert.pem /etc/krb5/

```
$ sudo mkdir /etc/krb5
$ sudo cp kdc.pem kdckey.pem cacert.pem /etc/krb5/
```

preauth . realm AKTIV-TEST /etc/krb5kdc/kdc.conf

/etc/krb5kdc/kdc.conf

```
[realms]
    AKTIV-TEST = {
        database_name = /var/lib/krb5kdc/principal
        admin_keytab = FILE:/etc/krb5kdc/kadm5.keytab
        acl_file = /etc/krb5kdc/kadm5.acl
        key_stash_file = /etc/krb5kdc/stash
        max_life = 10h 0m 0s
        max_renewable_life = 7d 0h 0m 0s
        master_key_type = des3-hmac-sha1
        supported_enctypes = aes256-cts:normal arcfour-hmac:normal des3-hmac-sha1:normal des-cbc-crc:normal des:
normal des:v4 des:norealm des:onlyrealm des:afs3
        default_principal_flags = +preauth
        pkinit_anchors = FILE:/etc/krb5/cacert.pem
        pkinit_identity = FILE:/etc/krb5/kdc.pem,/etc/krb5/kdckey.pem
    }
```

preauth

```
$ sudo kadmin.local
$ kadmin.local$: modprinc +requires_preauth <username>
```

rtadmin

. .

```
#    ID!
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so --keypairgen --key-type rsa:2048 -l --id 45
openssl
OpenSSL> engine dynamic -pre SO_PATH:/usr/lib/x86_64-linux-gnu/engines-1.1/pkcs11.so -pre ID:pkcs11 -pre
LIST_ADD:1 -pre LOAD -pre MODULE_PATH:librtpkcs11ecp.so
...
OpenSSL> req -engine pkcs11 -new -key 45 -keyform engine -out client.req -subj "/C=RU/ST=Moscow/L=Moscow/O=Aktiv
/OU=dev/CN=testuser/emailAddress=testuser@mail.com"
```

(client.req). .

```
$ REALM=<realm>; export REALM
$ CLIENT=<username>; export CLIENT
$ openssl x509 -CAkey cakey.pem -CA cacert.pem -req -in client.req -extensions client_cert -extfile
pkinit_extensions -out client.pem
```

5.3.2 KDC

```
 $ /etc/init.d/krb5-admin-server restart
 $ /etc/init.d/krb5-kdc restart
```

. **/etc/krb5/**

```
$ pkcs11-tool --module /usr/lib/librtpkcs11ecp.so -l -y cert -w ./client.pem --id 45
$ sudo cp cacert.pem /etc/krb5/cacert.pem
```

/etc/krb5.conf

/etc/krb5.conf

```
[libdefaults]
        default_realm = <realm>
        pkinit_anchors = FILE:/etc/krb5/cacert.pem
#
#        pkinit_identities = FILE:/etc/krb5/client.pem,/etc/krb5/clientkey.pem
#
        pkinit_identities = PKCS11:/usr/lib/librtpkcs11ecp.so
```

```
$ kinit <username>
```

- , . **/etc/krb5.conf /etc/krb5kdc/kdc.conf.**

```
[logging]
        default = FILE:/var/log/krb5.log
        kdc = FILE:/var/log/krb5kdc.log
        admin_server = FILE:/var/log/kadmin.log
```

KDC .