1. Fresh CentOS 6.4 install:
    - group: basic server
    - locality: english
    - GT+6
    - dns: 77.88.8.88, 77.88.8.2
    - FQDN: ipaserver.example.com
2. Updated + epel repo installed. Additional packages:
    - htop
    - nmap
    - bash-completion
3. installation according to this https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/index.html
4. iptables:
    - iptables -I INPUT -p tcp -m multiport --dports 80,443,389,636,88,464,53,7389 -j ACCEPT
    - iptables -I INPUT -p udp -m multiport --dports 88,464,53,123 -j ACCEPT
5. # cat /etc/hosts
   127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
   ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
   10.39.254.16 ipaserver.example.com ipaserver
6. ipa-packages installed:
    - ipa-server
    - bind
    - bind-dyndb-ldap
7. additional libsss package: libsss_sudo
8. ipa-server instalation:
    - # ipa-server-install --setup-dns
9. enabling mkhomedir:
    - # authconfig --enablemkhomedir --update
10. somehow after this sssd autostart disabled, so I needed to do this:
    - # chkconfig sssd on
    - # service sssd start
11. creating test user:
    - # ipa user-add
      Имя: test
      Фамилия: user
      User login [tuser]:
12. changing login shell:
    - # ipa user-mod tuser --shell=/bin/bash
13. adding group for sudo users:
    - # ipa group-add allsudo --desc=all_commands_allowed_evrywhere
14. adding tuser to admins:
    - # ipa group-add-member admins --users=tuser
15. adding tuser to allsudo:
    - ipa group-add-member allsudo --users=tuser
16. adding sudorule:
    - # ipa sudorule-add allsudo --desc=all_commands_allowed_evrywhere --hostcat=all --cmdcat=all --runasusercat=all --runasgroupcat=all

- # ipa sudorule-add-user allsudo --groups=allsudo

17. configuring sudo on this host according to this http://www.freeipa.org/images/7/77/Freeipa30_SSSD_SUDO_Integration.pdf
18. that is all about ipaserver!
19. $ ipa host-add --desc=postgresql_server postgresql.example.com --locality=Ufa --location=national_library --platform=ovirt --os=EL6 --force --ip-address=10.39.254.17
20. $ ipa service-add postgresql/postgresql.example.com
21. Fresh CentOS 6.4 install:
    - group: basic server
    - locality: english
    - GT+6
    - dns: 10.39.254.17
    - FQDN: postgresql.example.com
22. Updated + epel repo installed. Additional packages:
    - htop
    - nmap
    - bash-completion
    -
23. # ipa-client-install --mkhomedir --enable-dns-updates
24. additional libsss package: libsss_sudo
25. configuring sudo on this host according to this http://www.freeipa.org/images/7/77/Freeipa30_SSSD_SUDO_Integration.pdf
26. adding postgresql repository
    - $ sudo yum install http://yum.postgresql.org/9.3/redhat/rhel-6-x86_64/pgdg-centos93-9.3-1.noarch.rpm
27. installing postgresql
    - $ sudo yum install postgresql93*
28. initialising postgresql
    - $ sudo service postgresql-9.3 initdb
29. now $PGDATA=/var/lib/pgsql/9.3/data
30. so, according to this http://www.postgresql.org/docs/9.3/interactive/ssl-tcp.html we need to make:
    - $PGDATA/server.crt
    - $PGDATA/server.key
31. and that is where I meet the problem!
32. documentation about is https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/certmongerX.html
33. situation looks similar to example https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Identity_Management_Guide/certmongerX.html#certmonger-req
34. ipa-getcert help:
    Usage: ipa-getcert request [options]

    Required arguments:
    * If using an NSS database for storage:
      -d DIR   NSS database for key and cert
      -n NAME   nickname for NSS-based storage (only valid with -d)
      -t NAME   optional token name for NSS-based storage (only valid with -d)

* If using files for storage:
  -k FILE PEM file for private key
  -f FILE PEM file for certificate (only valid with -k)
* If keys are to be encrypted:
  -p FILE file which holds the encryption PIN
  -P PIN PIN value

Optional arguments:
* Certificate handling settings:
  -I NAME nickname to assign to the request
  -g SIZE size of key to be generated if one is not already in place
  -r attempt to renew the certificate when expiration nears (default)
  -R don't attempt to renew the certificate when expiration nears
  -T PROFILE ask the CA to process the request using the named profile or template
* Parameters for the signing request:
  -N NAME set requested subject name (default: CN=<hostname>)
  -U EXTUSAGE set requested extended key usage OID
  -K NAME set requested principal name
  -D DNSNAME set requested DNS name
  -E EMAIL set requested email address
* Bus options:
  -S connect to the certmonger service on the system bus
  -s connect to the certmonger service on the session bus
* Other options:
  -B command to run before saving the certificate
  -C command to run after saving the certificate
  -v report all details of errors

35. so my case is file storage, and I shall do something like this:
    - $ ipa-getcert request \
      -k /var/lib/pgsql/9.3/data/server.key \
      -f /var/lib/pgsql/9.3/data/server.crt \
      -K postgresql/postgresql.example.com \
      -N CN=postgresql.example.com \
      -D postgresql.example.com
36. running like this gives write access denied, so I do it like this^
    - $ sudo -i
    - # kinit tuser
    - # ipa-getcert request \
      -k /var/lib/pgsql/9.3/data/server.key \
      -f /var/lib/pgsql/9.3/data/server.crt \
      -K postgresql/postgresql.example.com \
      -N CN=`postgresql.example.com` \
      -D `postgresql.example.com`
37. but this gives this error: The parent of location "/var/lib/pgsql/9.3/data/server.crt" must be a valid directory.
38. Okay! I'll try to create keys at home directory and then copy to $PGDATA
    - $ sudo ipa-getcert request -f /home/tuser/server.crt -k /home/tuser/server.key -K postgresql/postgresql.example.com -N CN=postgresql.example.com -D

postgresql.example.com
[sudo] password for tuser: ******
New signing request "20131106075356" added.

39. The question is what to do next?!!!
40. I can list requests by command:
    - $ sudo ipa-getcert list
    - Number of certificates and requests being tracked: 2.
    - Request ID '20131101115647':
    - status: MONITORING
    - stuck: no
    - key pair storage: type=NSSDB,location='/etc/pki/nssdb',nickname='IPA Machine Certificate - postgresql.example.com',token='NSS Certificate DB'
    - certificate: type=NSSDB,location='/etc/pki/nssdb',nickname='IPA Machine Certificate - postgresql.example.com',token='NSS Certificate DB'
    - CA: IPA
    - issuer: CN=Certificate Authority,O=EXAMPLE.COM
    - subject: CN=postgresql.example.com,O=EXAMPLE.COM
    - expires: 2015-11-02 11:56:48 UTC
    - eku: id-kp-serverAuth,id-kp-clientAuth
    - pre-save command:
    - post-save command:
    - track: yes
    - auto-renew: yes
    - Request ID '20131106075356':
    - status: NEED_KEY_PAIR
    - stuck: no
    - key pair storage: type=FILE,location='/home/tuser/server.key'
    - certificate: type=FILE,location='/home/tuser/server.crt'
    - CA: IPA
    - issuer:
    - subject:
    - expires: unknown
    - pre-save command:
    - post-save command:
    - track: yes
    - auto-renew: yes
41. But! what to do with it?