Instalação em RedHat (RHEL 7.9)

Material para instalação do RedHat

Instalação e configuração do Elastic em Red Hat (RHEL 7.9)

Preparação

Para executar o procedimento é necessário que o servidor tenha conexão com os repositórios oficiais na internet:

1. https://cdn.redhat.com -> Repositório oficial da Red Hat. Para download de libs, pacotes de apoio a administração e atualização do S.O.
2. https://artifacts.elastic.co -> Repositório oficial do Elastic. Para download da stack e suas bibliotecas.

Licença

Para verificar se o servidor está com a licença ativa e funcional, utilize o comando:

```
$ sudo subscription-manager list --consumed
```

Caso contrário, será necessário fazer o registro com o comando:

```
$ sudo subscription-manager register
```

Instalando a stack Elastic (ELK) 7.4

Importação da chave do repositório Elastic:

```
$ sudo rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

Configuração do repositório:

```
$ sudo vim /etc/yum.repos.d/elasticsearch.repo

[elasticsearch]
name=Elasticsearch repository for 7.x packages
baseurl=https://artifacts.elastic.co/packages/7.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=0
autorefresh=1
type=rpm-md
```

Instalação do componente Elasticsearch

Instalação do **Elasticsearch** versão 7.4.0:

```
$ sudo yum install --enablerepo=elasticsearch elasticsearch-7.4.0-1.x86_64
```

Configuração do serviço no sistema operacional (daemon):

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable elasticsearch.service
$ sudo systemctl start elasticsearch.service
```

Liberação de portas no firewall:

```
$ sudo firewall-cmd --permanent --add-port=9200/tcp
$ sudo firewall-cmd --reload
```

Teste para verificar funcionamento do elasticsearch:

```
$ curl -X GET "localhost:9200/?pretty"

{
  "name" : "elk-ol7",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "jugxM_UASLeLjJwOITvZ5A",
  "version" : {
    "number" : "7.4.0",
    "build_flavor" : "default",
    "build_type" : "rpm",
    "build_hash" : "22e1767283e61a198cb4db791ea66e3f11ab9910",
    "build_date" : "2019-09-27T08:36:48.569419Z",
    "build_snapshot" : false,
    "lucene_version" : "8.2.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Configuração da licença "basic", detalhes da licenças são encontradas no site do Elastic:

```
$ sudo vim /etc/elasticsearch/elasticsearch.yml

xpack.license.self_generated.type: "basic"
transport.host: localhost
transport.tcp.port: 9300
http.port: 9200
network.host: 0.0.0.0
xpack.security.enabled: true

$ sudo systemctl restart elasticsearch.service
```

Configuração de usuários e senhas de autenticação no elasticsearch:

```
sudo /usr/share/elasticsearch/bin/elasticsearch-setup-passwords interactive
sudo systemctl restart elasticsearch.service
```

Teste de autenticação com senha certa:

```
$ curl -u elastic:senhacerta -X GET "localhost:9200/?pretty"

{
  "name" : "elk-ol7",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "jugxM_UASLeLjJwOITvZ5A",
```

```
  "version" : {
    "number" : "7.4.0",
    "build_flavor" : "default",
    "build_type" : "rpm",
    "build_hash" : "22e1767283e61a198cb4db791ea66e3f11ab9910",
    "build_date" : "2019-09-27T08:36:48.569419Z",
    "build_snapshot" : false,
    "lucene_version" : "8.2.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Teste de autenticação com senha errada:

```
$ curl -u elastic:senhaerrada -X GET "localhost:9200/?pretty"

{
  "error" : {
    "root_cause" : [
      {
        "type" : "security_exception",
        "reason" : "failed to authenticate user [elastic]",
        "header" : {
          "WWW-Authenticate" : "Basic realm=\"security\" charset=\"UTF-8\""
        }
      }
    ],
    "type" : "security_exception",
    "reason" : "failed to authenticate user [elastic]",
    "header" : {
      "WWW-Authenticate" : "Basic realm=\"security\" charset=\"UTF-8\""
    }
  },
  "status" : 401
}
```

Instalação do componente Kibana

Instalação do Kibana versão 7.4.0:

```
$ sudo yum install --enablerepo=elasticsearch kibana-7.4.0-1.x86_64
```

Liberação de portas no firewall:

```
$ sudo firewall-cmd --permanent --add-port=5601/tcp
$ sudo firewall-cmd --reload
```

Configuração para conectar no Elasticsearch:

```
$ sudo vim /etc/kibana/kibana.yml

server.host: "192.168.0.1"
elasticsearch.hosts: ["http://localhost:9200"]
elasticsearch.username: "kibana"
elasticsearch.password: "senha"
logging.dest: /var/log/kibana/kibana.log
```

Configuração de diretório para geração de log do Kibana:

```
$ sudo mkdir /var/log/kibana
$ sudo chown -R kibana:kibana /var/log/kibana
```

Configuração do serviço no sistema operacional (daemon):

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable kibana.service
$ sudo systemctl start kibana.service
$ sudo tail -fn100 /var/log/kibana/kibana.log | grep listening
```

Teste de funcionamento do Kibana, no browser:

```
http://<ip>:5601/login
```

Instalação e Configuração do QM 2.40+ em Red Hat (RHEL 7.9)

Preparação

Este tutorial e procedimentos está assumindo que serão 2 máquinas distintas. Uma será a camada de repositório, hospedando o PostgreSQL, e a outra camada, de aplicação, hospedará o web server Apache Tomcat e o MD2 QualityManager.

Para executar o procedimento é necessário:

1. Os 2 servidores terem conexão com os repositórios oficiais na internet:
    1. Camada aplicação:
        1. https://cdn.redhat.com –> Repositório oficial da Red Hat. Para download de libs, pacotes de apoio a administração e atualização do S.O..
        2. https://archive.apache.org –> Repositório oficial da Apache. Para download do web server Tomcat e suas bibliotecas
    2. Camada repositório:
        1. https://cdn.redhat.com –> Repositório oficial da Red Hat. Para download de libs, pacotes de apoio a administração e atualização do S.O.
        2. https://download.postgresql.org –> Repositório oficial do PostgreSQL. Para download do RDBMS PostgreSQL e suas bibliotecas.
2. Com os sistemas operacionais licenciados e sincronizados através do "subscription manager".
3. Fazer os downloads da aplicação e arquivos do repositório oficial MD2:
    1. Aplicação: qualityManager-prj.war
    2. Estrutura de banco de dados: DB_QM.psql.gz
    3. Arquivos da aplicação: qm_static.tar.gz
    4. Ferramenta de apoio: PortalPass.jar

Licença

Para verificar se o servidor está com a licença ativa e funcional através do comando.

```
$ sudo subscription-manager list --consumed
```

Caso contrário, será necessário fazer o registro com o comando

```
$ sudo subscription-manager register
```

Instalando o banco de dados repositório (PostgreSQL 12)

Instalação do repositório RPM:

```
$ sudo yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

Instalação do servidor RDBMS PostgreSQL e o pacote "contrib":

```
$ sudo yum install -y postgresql12-server
$ sudo yum install -y postgresql12-contrib
```

Iniciar o banco de dados:

```
$ sudo /usr/pgsql-12/bin/postgresql-12-setup initdb
```

Configurar o serviço (daemon) para subir automaticamente com o sistema operacional:

```
$ sudo systemctl enable postgresql-12
$ sudo systemctl start postgresql-12
```

Configurar o firewall para aceitar a porta do PostgreSQL. A porta 5432 é a padrão do RDBMS:

```
$ sudo firewall-cmd --permanent --add-port=5432/tcp
$ sudo firewall-cmd --reload
```

Conectar no banco de dados e fazer as criações dos usuários:

```
$ sudo -u postgres psql

create user md2net with encrypted password 'md2net2018';
```

```
alter user md2net with superuser;
create user mdm with encrypted password 'trocarsenha';
```

Permitir conexões remotas e o método de autenticação no PostgreSQL.

```
$ sudo vim /var/lib/pgsql/12/data/pg_hba.conf

# IPv4 local connections:
host    all             md2net             0.0.0.0/0               md5
# IPv6 local connections:
host    all             md2net             ::1/128                 md5
```

Permitir conexão remota:

```
$ sudo vim /var/lib/pgsql/12/data/postgresql.conf

listen_address='*'
```

Reiniciar o serviço:

```
$ sudo systemctl restart postgresql-12.service
```

Configurar o perfil do usuário no SO para conectar no banco:

```
$ cd ~
$ vim .pgpass

127.0.0.1:5432:qualitymanager:md2net:md2net2018

$ chmod 600 .pgpass
```

Criar o banco de dados, repositório da ferramenta MD2 QualityManager, importando o arquivo de estrutura do banco:

```
$ createdb --lc-collate pt_BR.UTF-8 --lc-ctype pt_BR.UTF-8 -E UTF-8 -O postgres -T template0 -
e qualitymanager
$ gunzip -c DB_QM.psql.gz |  psql -h 127.0.0.1 -U md2net qualitymanager
```

Instalando a camada de aplicação web server (Apache Tomcat 9.0.43)

Instalação do Java JDK através do repositório oficial:

```
$ sudo yum install java-1.8.0-openjdk-1:1.8.0.312.b07-1.el7_9.x86_64
```

Criação do grupo e usuário "tomcat" no sistema operacional:

```
$ sudo groupadd --system tomcat
$ sudo useradd -d /opt/tomcat -r -s /bin/false -g tomcat tomcat
```

Download do pacote 'wget' para utilizar no download, descompactação, movimentação e definição de usuário de S.O.:

```
$ sudo yum install wget
$ sudo yum install vim
$ cd /tmp
$ sudo wget https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.43/bin/apache-tomcat-9.0.43.tar.gz
$ sudo tar xvf apache-tomcat-9.0.43.tar.gz -C /opt
$ sudo ln -s /opt/apache-tomcat-9.0.43/ /opt/tomcat
$ sudo chown -R tomcat:tomcat /opt/tomcat
$ sudo chown -R tomcat:tomcat /opt/apache-tomcat-9.0.43
```

Configuração para que seja um serviço de sistema (daemon):

```
$ sudo vim /etc/systemd/system/tomcat.service

[Unit]
Description=Tomcat Server
After=syslog.target network.target
```

```
[Service]
Type=forking
User=tomcat
Group=tomcat

Environment=JAVA_HOME=/usr/lib/jvm/jre
Environment='JAVA_OPTS=-Djava.awt.headless=true'
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment='CATALINA_OPTS=-Xms512M -Xmx1024M'
ExecStart=/opt/tomcat/bin/catalina.sh start
ExecStop=/opt/tomcat/bin/catalina.sh stop

ReadWritePaths=/opt/qm_static/

[Install]
WantedBy=multi-user.target

$ sudo systemctl daemon-reload
$ sudo systemctl start tomcat
$ sudo systemctl enable tomcat
$ systemctl status tomcat
```

Liberação de porta no firewall:

```
$ sudo firewall-cmd --permanent --add-port=8080/tcp
$ sudo firewall-cmd --reload
```

Teste para conferir se aplicação Tomcat está funcional:

```
$ curl -v http://127.0.0.1:8080
```

Descompactação dos arquivos:

```
$ tar -xvzf qm_static.tar.gz
$ sudo mv /tmp/qm_static /opt
$ sudo chown -R tomcat:tomcat /opt/qm_static/
```

Configuração para que a aplicação reconheça a máquina remota que hospeda o repositório (utilizar a ferramenta de apoio "PortalPass.jar" para gerar a senha criptografada)

:

```
$ sudo vim /opt/tomcat/lib/qm.app.properties

qm.app.db.hibernate.connection.url=jdbc:postgresql://ipDoPostgreSQL:5432/qualitymanager
qm.app.db.hibernate.connection.username=md2net
qm.app.db.hibernate.connection.password=aaAAaadasdweasdeA==
qm.app.db.hibernate.default_schema=public
```

Publicação do WAR da aplicação no Tomcat:

```
$ sudo mv /tmp/qualityManager-prj.war /opt/tomcat/webapps/
$ sudo systemctl restart tomcat.service
```

Pronto! Teste via browser.

```
http://ipdoqm:8080/qualityManager-prj/login.xhtml
```