

NSD ARCHITECTURE DAY06

1. [案例1：安装与部署](#)
2. [案例2：Hadoop词频统计](#)
3. [案例3：节点管理](#)
4. [案例4：NFS配置](#)

1 案例1：安装与部署

1.1 问题

本案例要求：

- 对mapred和yarn文件进行配置
- 验证访问Hadoop

1.2 方案

在day05准备好的环境下给master（nn01）主机添加ResourceManager的角色，在node1，node2，node3上面添加NodeManager的角色，如表-1所示：

表-1

主机	角色	软件
192.168.1.21 master	NameNode SecondaryNameNode ResourceManager	HDFS YARN
192.168.1.22 node1	DataNode NodeManager	HDFS YARN
192.168.1.23 node2	DataNode NodeManager	HDFS YARN
192.168.1.24 node3	DataNode NodeManager	HDFS YARN

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装与部署hadoop

1) 配置mapred-site（nn01上面操作）

[Top](#)

```

01. [ root@nn01 ~] # cd /usr/local/hadoop/etc/hadoop/
02. [ root@nn01 hadoop] # mv mapred-site.xml.template mapred-site.xml
03. [ root@nn01 hadoop] # vim mapred-site.xml
04. <configuration>
05. <property>
06.     <name>mapreduce.framework.name</name>
07.     <value>yarn</value>
08. </property>
09. </configuration>

```

2) 配置yarn-site (nn01上面操作)

```

01. [ root@nn01 hadoop] # vim yarn-site.xml
02. <configuration>
03.
04. <!-- Site specific YARN configuration properties -->
05. <property>
06.     <name>yarn.resourcemanager.hostname</name>
07.     <value>nn01</value>
08. </property>
09. <property>
10.     <name>yarn.nodemanager.aux-services</name>
11.     <value>mapreduce_shuffle</value>
12. </property>
13. </configuration>

```

3) 同步配置 (nn01上面操作)

```

01. [ root@nn01 hadoop] # for i in { 22..24 }; do rsync -aSH --delete /usr/local/hadoop/ 192.:
02. [ 1] 712
03. [ 2] 713
04. [ 3] 714

```

4) 验证配置 (nn01上面操作)

[Top](#)

```

01. [ root@nn01 hadoop] # cd /usr/local/hadoop

```

02. [root@nn01 hadoop] # ./sbin/start- dfs.sh
03. Starting namenodes on [nn01]
04. nn01: namenode running as process 23408. Stop it first.
05. node1: datanode running as process 22409. Stop it first.
06. node2: datanode running as process 22367. Stop it first.
07. node3: datanode running as process 22356. Stop it first.
08. Starting secondary namenodes [nn01]
09. nn01: secondary namenode running as process 23591. Stop it first.
10. [root@nn01 hadoop] # ./sbin/start- yarn.sh
11. starting yarn daemons
12. starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-root-resourcemanager
13. node2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager
14. node3: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager
15. node1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager
16. [root@nn01 hadoop] # jps //nn01查看有ResourceManager
17. 23408 NameNode
18. 1043 ResourceManager
19. 1302 Jps
20. 23591 Secondary NameNode
21. [root@nn01 hadoop] # ssh node1 jps //node1查看有NodeManager
22. 25777 Jps
23. 22409 DataNode
24. 25673 NodeManager
25. [root@nn01 hadoop] # ssh node2 jps //node1查看有NodeManager
26. 25729 Jps
27. 25625 NodeManager
28. 22367 DataNode
29. [root@nn01 hadoop] # ssh node3 jps //node1查看有NodeManager
30. 22356 DataNode
31. 25620 NodeManager
32. 25724 Jps

5) web访问hadoop

01. http://192.168.1.21:50070/ //-- namenode web页面 (nn01)
02. http://192.168.1.21:50090/ //-- secondary namenode web页面 (nn01)
03. http://192.168.1.22:50075/ //-- datanode web页面 (node1,node2,node3)
04. http://192.168.1.21:8088/ //-- resourcemanager web页面 (nn01) [Top](#)
05. http://192.168.1.22:8042/ //-- nodemanager web页面 (node1,node2,node3)

2 案例2 : Hadoop词频统计

2.1 问题

本案例要求：

- 在集群文件系统里创建文件夹
- 上传要分析的文件到目录中
- 分析上传文件
- 展示结果

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：词频统计

```

01. [root@nn01 hadoop] # ./bin/hadoop fs - ls /      //查看集群文件系统的根，没有内容
02. [root@nn01 hadoop] # ./bin/hadoop fs - mkdir /aaa
03. //在集群文件系统下创建aaa目录
04. [root@nn01 hadoop] # ./bin/hadoop fs - ls /      //再次查看，有刚创建的aaa目录
05. Found 1 items
06. drwxr-xr-x - root supergroup      0 2018-09-10 09:56 /aaa
07. [root@nn01 hadoop] # ./bin/hadoop fs - touchz /fa //在集群文件系统下创建fa文件
08. [root@nn01 hadoop] # ./bin/hadoop fs - put *.txt /aaa
09. //上传*.txt到集群文件系统下的aaa目录
10. [root@nn01 hadoop] # ./bin/hadoop fs - ls /aaa //查看
11. Found 3 items
12. -rw-r--r--  2 root supergroup      86424 2018-09-10 09:58 /aaa/LICENSE.txt
13. -rw-r--r--  2 root supergroup     14978 2018-09-10 09:58 /aaa/NOTICE.txt
14. -rw-r--r--  2 root supergroup      1366 2018-09-10 09:58 /aaa/README.txt
15. [root@nn01 hadoop] # ./bin/hadoop fs - get /aaa //下载集群文件系统的aaa目录
16. [root@nn01 hadoop] # ./bin/hadoop jar \
17.   share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.6.jar wordcount /aaa /bbb
18. [root@nn01 hadoop] # ./bin/hadoop fs - cat /bbb/* //查看集群里的数据

```

3 案例3 : 节点管理

3.1 问题

本案例要求：

- 增加一个新的节点
- 查看状态
- 删除节点

[Top](#)

3.2 方案：

另外准备两台主机，node4和nfsgw，作为新添加的节点和网关，具体要求如表-2所示：

表-2

主机名	IP	作用
node4	192.168.1.25	新增节点
nfsgw	192.168.1.26	浏览访问 HDFS 文件系统

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：增加节点

1) 增加一个新的节点node4

```

01. [ root@hadoop5 ~] # echo node4 > /etc/hostname //更改主机名为node4
02. [ root@hadoop5 ~] # hostname node4
03. [ root@node4 ~] # yum -y install rsync
04. [ root@node4 ~] # yum -y install java-1.8.0-openjdk-devel
05. [ root@node4 ~] # mkdir /var/hadoop
06. [ root@nn01.ssh] # ssh-copy-id 192.168.1.25
07. [ root@nn01.ssh] # vim /etc/hosts
08. 192.168.1.21 nn01
09. 192.168.1.22 node1
10. 192.168.1.23 node2
11. 192.168.1.24 node3
12. 192.168.1.25 node4
13. [ root@nn01.ssh] # scp /etc/hosts 192.168.1.25:/etc/
14. [ root@nn01 ~] # cd /usr/local/hadoop/
15. [ root@nn01.hadoop] # vim ./etc/hadoop/slaves
16. node1
17. node2
18. node3
19. node4
20. [ root@nn01.hadoop] # for i in { 22..25 }; do rsync -aH --delete /usr/local/hadoop/
21. \ 192.168.1.$i:/usr/local/hadoop/ -e 'ssh' & done //同步配置
22. [ 1] 1841
23. [ 2] 1842
24. [ 3] 1843
25. [ 4] 1844
26. [ root@node4.hadoop] # ./sbin/hadoop-daemon.sh start datanode //启动

```

[Top](#)

2) 查看状态

```
01. [root@node4 hadoop] # jps
02. 24439 Jps
03. 24351 DataNode
```

3) 设置同步带宽

```
01. [root@node4 hadoop] # ./bin/hdfs dfsadmin -setBalancerBandwidth 60000000
02. Balancer bandwidth is set to 60000000
03. [root@node4 hadoop] # ./sbin/start-balancer.sh
```

4) 删除节点

```
01. [root@nn01 hadoop] # vim /usr/local/hadoop/etc/hadoop/slaves
02. //去掉之前添加的node4
03. node1
04. node2
05. node3
06. [root@nn01 hadoop] # vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
07. //在此配置文件里面加入下面四行
08. <property>
09.     <name>dfs.hosts.exclude</name>
10.     <value>/usr/local/hadoop/etc/hadoop/exclude</value>
11. </property>
12.
13. [root@nn01 hadoop] # vim /usr/local/hadoop/etc/hadoop/exclude
14. node4
```

5) 导出数据

```
01. [root@nn01 hadoop] # ./bin/hdfs dfsadmin -refreshNodes
02. Refresh nodes successful
03. [root@nn01 hadoop] # ./bin/hdfs dfsadmin -report //查看node4显示Decommissioned
04. Dead datanodes ( 1 ):
05.
```

[Top](#)

```

06. Name: 192.168.1.25:50010 ( node4)
07. Hostname: node4
08. Decommission Status : Decommissioned
09. Configured Capacity : 17168314368 ( 15.99 GB)
10. DFS Used: 12288 ( 12 KB)
11. Non DFS Used: 1656664064 ( 1.54 GB)
12. DFS Remaining: 15511638016 ( 14.45 GB)
13. DFS Used%: 0.00%
14. DFS Remaining%: 90.35%
15. Configured Cache Capacity : 0 ( 0 B)
16. Cache Used: 0 ( 0 B)
17. Cache Remaining: 0 ( 0 B)
18. Cache Used%: 100.00%
19. Cache Remaining%: 0.00%
20. Xceivers: 1
21. Last contact: Mon Sep 10 10:59:58 CST 2018
22.
23. [ root@node4 hadoop] # ./sbin/hadoop-daemon.sh stop datanode //停止datanode
24. stopping datanode
25. [ root@node4 hadoop] # ./sbin/yarn-daemon.sh start nodemanager
26. //yarn 增加 nodemanager
27. [ root@node4 hadoop] # ./sbin/yarn-daemon.sh stop nodemanager //停止nodemanager
28. stopping nodemanager
29. [ root@node4 hadoop] # ./bin/yarn node - list
30. //yarn 查看节点状态，还是有node4节点，要过一段时间才会消失
31. 18/09/10 11:04:50 INFO client.RMProxy: Connecting to ResourceManager at nn01/192.168.
32. Total Nodes: 4
33.
34.      Node-Id      Node-State  Node-Http-Address  Number-of-Running-Containers
35.      node3:34628   RUNNING    node3:8042         0
36.      node2:36300   RUNNING    node2:8042         0
37.      node4:42459   RUNNING    node4:8042         0
38.      node1:39196   RUNNING    node1:8042         0

```

4 案例4 : NFS配置

4.1 问题

本案例要求：

- 创建代理用户
- 启动一个新系统，禁用Selinux和firewalld
- 配置NFSWG

[Top](#)

- 启动服务
- 挂载NFS并实现开机自启

4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：基础准备

1) 更改主机名，配置/etc/hosts (/etc/hosts在nn01和nfsgw上面配置)

```
01. [ root@localhost ~] # echo nfsgw > /etc/hostname
02. [ root@localhost ~] # hostname nfsgw
03. [ root@nn01 hadoop] # vim /etc/hosts
04. 192.168.1.21 nn01
05. 192.168.1.22 node1
06. 192.168.1.23 node2
07. 192.168.1.24 node3
08. 192.168.1.25 node4
09. 192.168.1.26 nfsgw
```

2) 创建代理用户 (nn01和nfsgw上面操作) , 以nn01为例子

```
01. [ root@nn01 hadoop] # groupadd -g 200 nfs
02. [ root@nn01 hadoop] # useradd -u 200 -g nfs nfs
```

3) 配置core-site.xml

```
01. [ root@nn01 hadoop] # ./sbin/stop-all.sh //停止所有服务
02. This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh
03. Stopping namenodes on [ nn01]
04. nn01: stopping namenode
05. node2: stopping datanode
06. node4: no datanode to stop
07. node3: stopping datanode
08. node1: stopping datanode
09. Stopping secondary namenodes [ nn01]
10. nn01: stopping secondary namenode
11. stopping yarn daemons
12. stopping resourcemanager
13. node2: stopping nodemanager
```

[Top](#)


```
14. node3: stopping nodemanager
15. node4: no nodemanager to stop
16. node1: stopping nodemanager
17. ...
18.
19. [ root@nn01 hadoop] # cd etc/hadoop
20. [ root@nn01 hadoop] # >exclude
21. [ root@nn01 hadoop] # v im core-site.xml
22.     <property>
23.         <name>hadoop.proxyuser.nfs.groups</name>
24.         <value>*</value>
25.     </property>
26.     <property>
27.         <name>hadoop.proxyuser.nfs.hosts</name>
28.         <value>*</value>
29.     </property>
```

4) 同步配置到node1 , node2 , node3

```
01. [ root@nn01 hadoop] # for i in { 22..24 }; do rsync - aSH -- delete /usr/local/hadoop/ 192.:
02. [ 4] 2722
03. [ 5] 2723
04. [ 6] 2724
```

5) 启动集群

```
01. [ root@nn01 hadoop] # /usr/local/hadoop/sbin/start-dfs.sh
```

6) 查看状态

```
01. [ root@nn01 hadoop] # /usr/local/hadoop/bin/hdfs dfsadmin - report
```

步骤二：NFSGW配置

1) 安装java-1.8.0-openjdk-devel和rsync

[Top](#)

```

01. [ root@nfs gw ~] # yum -y install java-1.8.0-openjdk-devel
02. [ root@nfs gw ~] # yum -y install rsync
03. [ root@nn01 hadoop] # rsync -av SH --delete \
04. /usr/local/hadoop/ 192.168.1.26: /usr/local/hadoop/ -e 'ssh'

```

2) 创建数据根目录 /var/hadoop (在NFSGW主机上面操作)

```
01. [ root@nfs gw ~] # mkdir /var/hadoop
```

3) 创建转储目录，并给用户nfs 赋权

```

01. [ root@nfs gw ~] # mkdir /var/nfstmp
02. [ root@nfs gw ~] # chown nfs:nfs /var/nfstmp

```

4) 给/usr/local/hadoop/logs赋权 (在NFSGW主机上面操作)

```

01. [ root@nfs gw ~] # setfacl -m u:nfs:rwx /usr/local/hadoop/logs
02. [ root@nfs gw ~] # vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
03.     <property>
04.         <name>nfs.exports.allowed.hosts</name>
05.         <value>*rw</value>
06.     </property>
07.     <property>
08.         <name>nfs.dump.dir</name>
09.         <value>/var/nfstmp</value>
10.     </property>

```

5) 可以创建和删除即可

```

01. [ root@nfs gw ~] # su - nfs
02. [ nfs@nfs gw ~] $ cd /var/nfstmp/
03. [ nfs@nfs gw nfstmp] $ touch 1
04. [ nfs@nfs gw nfstmp] $ ls
05. 1
06. [ nfs@nfs gw nfstmp] $ rm -rf 1

```

[Top](#)

```

07. [ nfs@nfs gw nf stmp] $ ls
08. [ nfs@nfs gw nf stmp] $ cd /usr/local/hadoop/logs/
09. [ nfs@nfs gw logs] $ touch 1
10. [ nfs@nfs gw logs] $ ls
11. 1 hadoop-root-secondary namenode- nn01.log y arn-root-resource manager- nn01.log
12. hadoop-root-namenode- nn01.log hadoop-root-secondary namenode- nn01.out y arn-ro
13. hadoop-root-namenode- nn01.out hadoop-root-secondary namenode- nn01.out.1
14. hadoop-root-namenode- nn01.out.1 Security Auth-root.audit
15. [ nfs@nfs gw logs] $ rm -rf 1
16. [ nfs@nfs gw logs] $ ls

```

6) 启动服务

```

01. [ root@nfs gw ~] # /usr/local/hadoop/sbin/hadoop-daemon.sh --script ./bin/hdfs start pc
02. starting portmap, logging to /usr/local/hadoop/logs/hadoop-root-portmap-nfs gw.out
03. [ root@nfs gw ~] # jps
04. 23714 Jps
05. 23670 Portmap
06.
07. [ root@nfs gw ~] # su - nfs
08. Last login: Mon Sep 10 12:31:58 CST 2018 on pts/0
09. [ nfs@nfs gw ~] $ cd /usr/local/hadoop/
10. [ nfs@nfs gw hadoop] $ ./sbin/hadoop-daemon.sh --script ./bin/hdfs start nfs3
11. //nfs3只能用代理用户启动
12. starting nfs3, logging to /usr/local/hadoop/logs/hadoop-nfs-nfs3-nfs gw.out
13. [ nfs@nfs gw hadoop] $ jps
14. 1362 Jps
15. 1309 Nfs3
16. [ root@nfs gw hadoop] # jps //root用户执行可以看到portmap和nfs3
17. 1216 Portmap
18. 1309 Nfs3
19. 1374 Jps

```

7) 实现客户端挂载 (客户端可以用node4这台主机)

```

01. [ root@node4 ~] # rm -rf /usr/local/hadoop
02. [ root@node4 ~] # yum -y install nfs-utils
03. [ root@node4 ~] # mount -t nfs -o \

```

[Top](#)

```
04. vers=3,proto=tcp,nolock,noatime,sync,noacl 192.168.1.26: / /mnt/ //挂载
05. [ root@node4 ~] # cd /mnt/
06. [ root@node4 mnt] # ls
07. aaa bbb fa system tmp
08. [ root@node4 mnt] # touch a
09. [ root@node4 mnt] # ls
10. a aaa bbb fa system tmp
11. [ root@node4 mnt] # rm -rf a
12. [ root@node4 mnt] # ls
13. aaa bbb fa system tmp
```

8) 实现开机自动挂载

```
01. [ root@node4 ~] # vim /etc/fstab
02. 192.168.1.26: / /mnt/ nfs vers=3,proto=tcp,nolock,noatime,sync,noacl,_netdev 0 0
03. [ root@node4 ~] # mount -a
04. [ root@node4 ~] # df -h
05. 192.168.1.26: / 64G 6.2G 58G 10% /mnt
06.
07. [ root@node4 ~] # rpcinfo -p 192.168.1.26
08. program vers proto port service
09. 100005 3 udp 4242 mountd
10. 100005 1 tcp 4242 mountd
11. 100000 2 udp 111 portmapper
12. 100000 2 tcp 111 portmapper
13. 100005 3 tcp 4242 mountd
14. 100005 2 tcp 4242 mountd
15. 100003 3 tcp 2049 nfs
16. 100005 2 udp 4242 mountd
17. 100005 1 udp 4242 mountd
```

[Top](#)