Write systemD Service file in Linux(centos and ubuntu)

Rohit · Follow
Published in Dev Genius
5 min read · Jun 26, 2022

Listen Share More

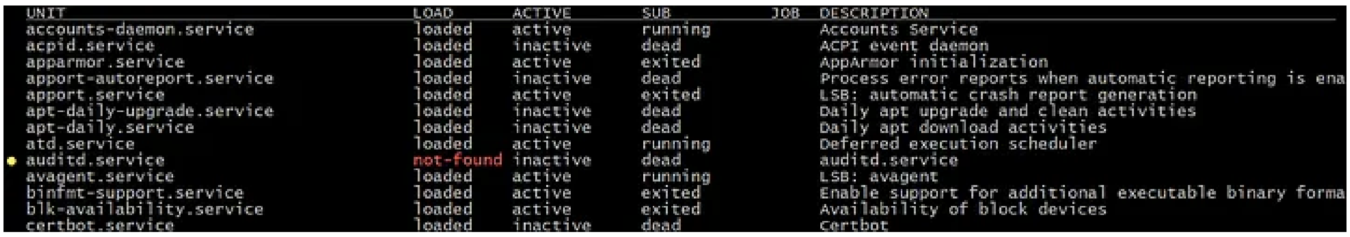
What is services…?

A variety of services run continuously on a Linux background, such as network and system services. **Services** running on Linux are also known as daemons, which refers to a group of processes working on the back-end.

In **Linux**, *Systemctl* is a utility with the responsibility to manage and control the systemd system. The *systemctl* command can be used to list all **services** in Linux.

List of some *systemctl* commands:

```
$ systemctl list-units --type=service --all
```

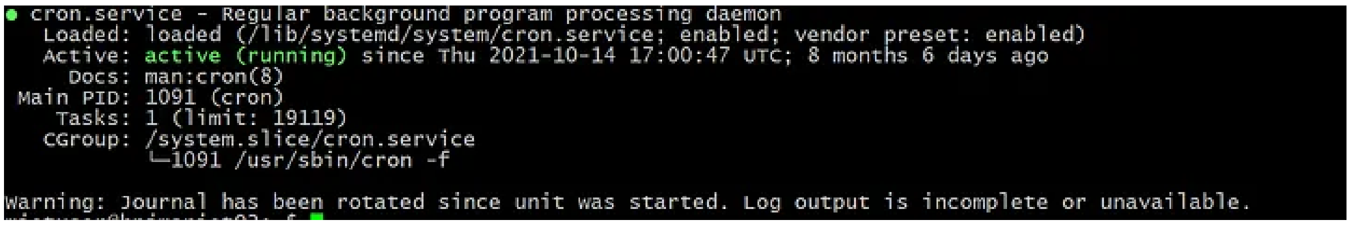


```
  UNIT                          LOAD      ACTIVE     SUB        JOB  DESCRIPTION
  accounts-daemon.service       loaded    active     running         Accounts Service
  acpid.service                 loaded    inactive   dead            ACPI event daemon
  apparmor.service              loaded    active     exited          AppArmor initialization
  apport-autoreport.service     loaded    inactive   dead            Process error reports when automatic reporting is ena
  apport.service                loaded    active     exited          LSB: automatic crash report generation
  apt-daily-upgrade.service     loaded    inactive   dead            Daily apt upgrade and clean activities
  apt-daily.service             loaded    inactive   dead            Daily apt download activities
  atd.service                   loaded    active     running         Deferred execution scheduler
● auditd.service                not-found inactive   dead            auditd.service
  avagent.service               loaded    active     running         LSB: avagent
  binfmt-support.service        loaded    active     exited          Enable support for additional executable binary forma
  blk-availability.service      loaded    active     exited          Availability of block devices
  certbot.service               loaded    inactive   dead            Certbot
```

```
$ systemctl list-units --type=service --state=running
```

Above command shows list of running servcies..

```
$ systemctl status cron.service
```



```
● cron.service - Regular background program processing daemon
   Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-10-14 17:00:47 UTC; 8 months 6 days ago
     Docs: man:cron(8)
 Main PID: 1091 (cron)
    Tasks: 1 (limit: 19119)
   CGroup: /system.slice/cron.service
           └─1091 /usr/sbin/cron -f

Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.
```

For more command details, please folow:

https://www.tecmint.com/list-all-running-services-under-systemd-in-linux/

Where to find Systemd Service files in Linux

Usually there are mainly two places (third place is **/lib/systemd/system/**)

where we find service file:

1. **/etc/systemd/system/** :- Files in `/etc/systemd/system` are manually placed here by the operator or Admin of the system for ad-hoc software installations that are not in the form of a package.
2. **/usr/lib/systemd/system/** :- Only contain systemd unit files which were put there by the package manager (YUM/DNF/RPM/APT/etc).

For more details checkout link:
https://unix.stackexchange.com/questions/206315/whats-the-difference-between-usr-lib-systemd-system-and-etc-systemd-system

How Systemd Service file look like!

```
[Unit]
Description=Foo

[Service]
ExecStart=/usr/sbin/foo-daemon
```

```
[Install]
WantedBy=multi-user.target
```

Create and attached permissions with command:

```
$ sudo touch /etc/systemd/system/foo-daemon.service
$ sudo chmod 664 /etc/systemd/system/foo-daemon.service
```

Basic controls command:

```
$ sudo systemctl start foo-daemon
$ sudo systemctl stop foo-daemon
$ sudo systemctl restart foo-daemon
$ systemctl status foo-daemon
```

Systemd Service File Options

Systemd service files typically consist of three sections.

The common configuration items are configured in the generic `[Unit]` and `[Install]` sections.

The service specific configuration options are configured in the `[Service]` section.

We can find more details with commands:

```
$ man systemd.unit

[UNIT] SECTION OPTIONS
       ....

Description=
           A human readable name for the unit. This is used by
systemd (and other UIs) as the label for the unit, so this string
           should identify....

[INSTALL] SECTION OPTIONS
       ......

Alias=
           A space-separated list of additional names this unit
shall be installed under. The names listed here must have the same
           suffix (i.e. type)...
```

For service section:

```
$ man systemd.service

OPTIONS
       Service files must include a "[Service]" section, which
carries information about the service and the process it supervises.
      ...

Type=
           Configures the process start-up type for this service
unit. One of simple, exec, forking, oneshot, dbus, notify or idle:

•   If set to si......
```

Create our Own Service file

For demonstrate, will create service file "**node_server.service**" and use already created "**mongodb.service**", which automatically created after installing **mongodb** database software in linux machine.

mongodb.service

```
Description=MongoDB Database Server
Documentation=https://docs.mongodb.org/manual
After=network-online.target
Wants=network-online.target

[Service]
User=mongodb
Group=mongodb
EnvironmentFile=-/etc/default/mongod
ExecStart=/usr/bin/mongod --config /etc/mongod.conf
PIDFile=/var/run/mongodb/mongod.pid
# file size
LimitFSIZE=infinity
# cpu time
LimitCPU=infinity
# virtual memory size
LimitAS=infinity
# open files
LimitNOFILE=64000
# processes/threads
LimitNPROC=64000
# locked memory
LimitMEMLOCK=infinity
```

```
# total threads (user+kernel)
TasksMax=infinity
TasksAccounting=false

# Recommended limits for mongod as specified in
# https://docs.mongodb.com/manual/reference/ulimit/#recommended-ulimit-settings

[Install]
WantedBy=multi-user.target
```

Enable the service:

```
$ sudo systemctl enable mongodb.service
```

For installing **mongodb**, please refer:
https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/

node_server.service

```
$ sudo touch /etc/systemd/system/node_server.service
$ sudo chmod 664 /etc/systemd/system/node_server.service
```

It's used to run **node** server at **localhost** with specified **port** e.g at **http://localhost:4001/** and connect **mongodb** database and show **Database** information.

Main part in this service file are **After** and **Wants** field. It clearly mention that it's depend on **mongodb.service.** Every time we restart machine **node_server.service** always run only after **mongodb.service.**

[Service] tag

Added **Type**=*simple* (based on requirement we can also use **notify** or **forking**)

Also we can specify **user** and **group** by which process will run.

Added **Environment**="*PORT=4001*", also called in **ExecStart.** For more info: https://www.freedesktop.org/software/systemd/man/systemd.exec.html#Environment

Added **ExecStart**=*/usr/bin/node /opt/node_server/app.js ${PORT}*. In which /usr/bin/node represent nodejs binary command and it will run **app.js** script with argument **${PORT}**

Also added **StandardOutput**=file:/opt/node_server/**output_log**.log and **StandardError**=file:/opt/node_server/**error_log**.log. For more info: https://www.freedesktop.org/software/systemd/man/systemd.exec.html#StandardOutput=

Can also config the ulimit for this process, checkout: https://www.freedesktop.org/software/systemd/man/systemd.exec.html#Process%20Properties

Added **Restart**=always, it will restart service when process break unexpectedly.

```
Description=NodeJs Server
Documentation=https://docs.mongodb.org/manual
After=mongodb.service
Wants=mongodb.serivce

[Service]
Type=simple

#User="we can also specify any user by which process will run"
#Group="we can also specify any group"

Environment="PORT=4001"

ExecStart=/usr/bin/node /opt/node_server/app.js ${PORT}

StandardOutput=file:/opt/node_server/output_log.log
StandardError=file:/opt/node_server/error_log.log

Restart=always

[Install]
WantedBy=multi-user.target
```

Start the service:

```
$ sudo systemctl start node_server.service
$ sudo systemctl status node_server.service

node_server.service
 Loaded: loaded (/etc/systemd/system/node_server.service; disabled;
vendor preset: enabled)
 Active: active (running) since Sun 2022-06-26 11:08:51 UTC; 12s ago
```

```
   Main PID: 1500 (node)
   Tasks: 11 (limit: 2274)
   Memory: 18.1M
   CGroup: /system.slice/node_server.service
   └─1500 /usr/bin/node /opt/node_server/app.js 4001
```

Enable the service:

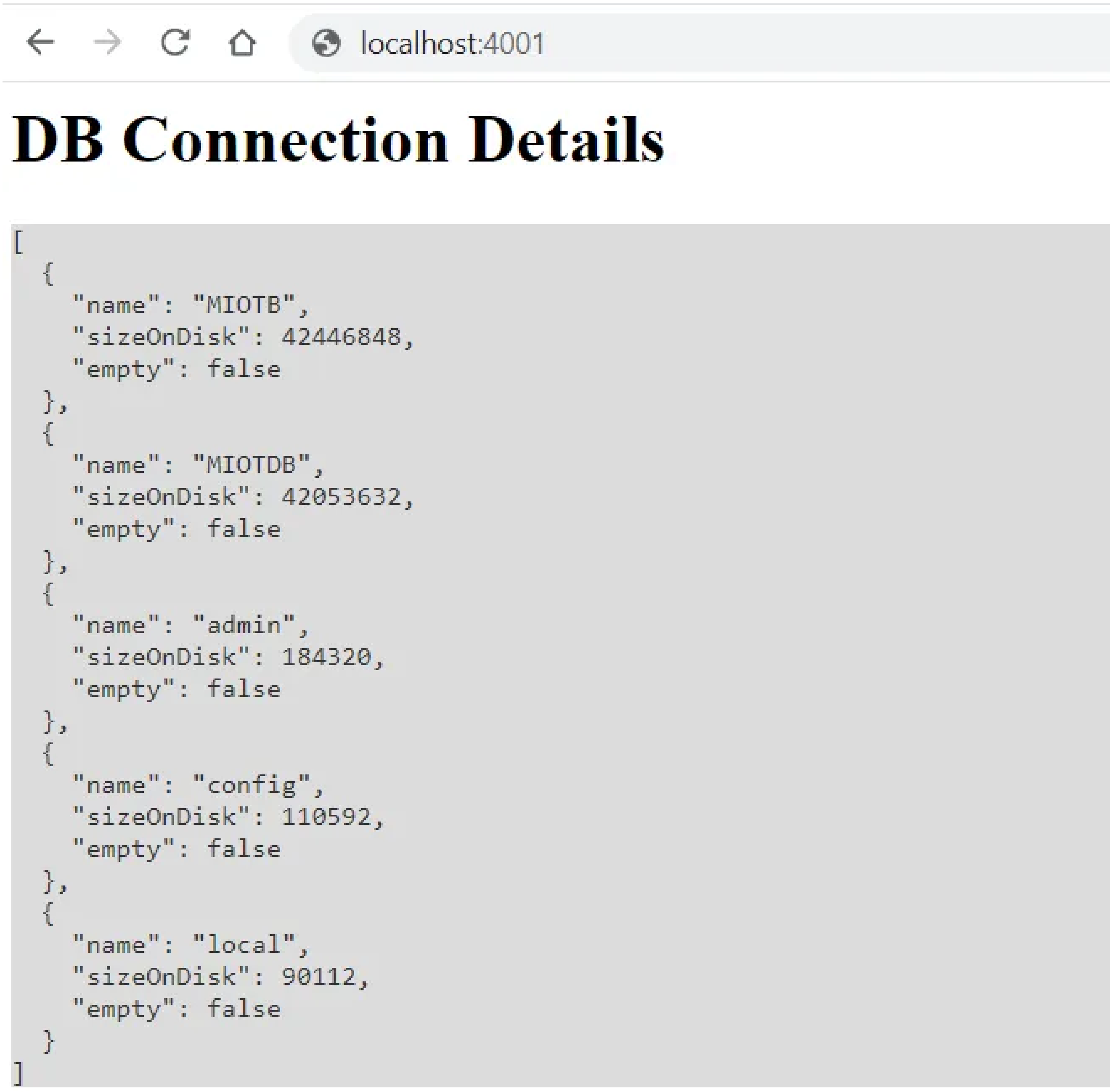
```
$ sudo systemctl enable node_server.service
```

Reboot the machine:

```
$ sudo shutdown -r now
```

After reboot you will see both **mongodb.service** and **node_server.service** are running

At http://localhost:4001/



DB Connection Details

```
[
  {
    "name": "MIOTB",
    "sizeOnDisk": 42446848,
    "empty": false
  },
  {
    "name": "MIOTDB",
    "sizeOnDisk": 42053632,
    "empty": false
  },
  {
    "name": "admin",
    "sizeOnDisk": 184320,
    "empty": false
  },
  {
    "name": "config",
    "sizeOnDisk": 110592,
    "empty": false
  },
  {
    "name": "local",
    "sizeOnDisk": 90112,
    "empty": false
  }
]
```

app.js Script

Path: /opt/node_server/**app.js**

Run command:

1. npm install mongodb -save
2. node app.js

```
$ node app.js 4001
Server is running on http://localhost:4001
```

Code snippet for **app.js**:

```
const http = require("http");
const {MongoClient} = require('mongodb');
const SERVER_PORT = process.argv[2] || 4001;

const DB_USER="admin";
const DB_PASSWORD="admin@123"
const DB_HOST_NAME="127.0.0.1"
const DB_PORT="27017"
const DATABASE="admin"

const uri =
`mongodb://${DB_USER}:${encodeURIComponent(DB_PASSWORD)}@${DB_HOST_NAME}:${DB_PORT}/${DATABASE}`;

const client = new MongoClient(uri);

async function startDbConnection(){
 try {
   //Connect to the MongoDB cluster
  await client.connect();

} catch (e) {
   console.error(e);
}

//start db connection
startDbConnection();

const requestListener = function (req, res) {
   res.setHeader("Content-Type", "text/html");
   res.writeHead(200);
   sendDbDetails(req,res);
};

async function sendDbDetails(req,res){
   let dbDetail = await getDbDetails();
   res.end(`<html><body>
            <h1>DB Connection Details</h1>
            <pre style="background-color:#DCDCDC;
            color:#333;">
                ${JSON.stringify(dbDetail,undefined,2)}
             </pre>
             </body></html>`);
}


async function getDbDetails(){
    databasesList = await client.db().admin().listDatabases();
    return databasesList.databases;
}

const server = http.createServer(requestListener);
```

```
server.listen(SERVER_PORT, () => {
  console.log(`Server is running on
http://localhost:${SERVER_PORT}`);

});


process.on('SIGTERM', () => {
  properlyCloseConnection()
});

process.on('SIGINT', () => {
 properlyCloseConnection()
});

process.on("uncaughtException",(e)=>{
  console.log(e);
  properlyCloseConnection()
});

process.on('exit', function () {
  console.log("process is exited..")
});

function properlyCloseConnection(){
    client.close(()=>{
        console.log('DB client closed');
    });

    server.close(() => {
        console.log('Process terminated');
     });
}
```

Hope article will help in writing service file in Linux. All code tested in Linux OS(Centos and Ubuntu)