

- [syslog-ng](#)
 -
 -
 -
 - [rsyslog](#)
 -
 - [logrotate](#)
 -
 - [rsyslog/syslog-ng](#)
 -
 -
 - [astra-create-debug-logs](#)



:

- Astra Linux Special Edition .10015-01 .10015-10 (1.7)
- Astra Linux Special Edition .10015-37 (7.7)
- Astra Linux Special Edition .10152-02 (4.7)
- Astra Linux Special Edition .10015-01 (1.6)
- Astra Linux Special Edition .10015-16 . 1 . 2
- Astra Linux Special Edition .10265-01 (8.1)
- Astra Linux Common Edition 2.12



Astra Linux Special Edition x.7 syslog-ng. Astra Linux rsyslog. syslog-ng:

- Astra Linux x.7;
- rsyslog. rsyslog .

rsyslog . rsyslog Astra Linux Special Edition (). rsyslog syslog-ng.

- **syslog-ng** - syslog-ng Astra Linux Special Edition x.7, ;
- **rsyslog** - rsyslog Astra Linux Common Edition Astra Linux Special Edition x.7, Astra Linux Special Edition x.7;
- **syslog (sysklogd)** - syslog Astra Linux ;
- **logrotate** - () .

:

- TCP/UDP/... IP- IP-;
- ;
- ;
- ;

syslog-ng

- : /etc/syslog-ng/syslog-ng.conf;
- — .conf /etc/syslog-ng/conf.d.

syslog-ng . , , . (-). . :

- — , ;
- (parsers) — ;
- (rewriting rules) — .



1. `sudo` :

```
sudo syslog-ng -sv
```

2. `(, )` (include), `,` `.` `log {}`.

`. :`

```
<> <> { <>(<> ..., ...) ...; ... };
```

:

- `- :`
 - `source -;`
 - `destination -;`
 - `log -;`
 - `filter -;`
 - `parser -;`
 - `rewrite rule -;`
 - `template -;`
 - `option - .`
- `- , . , , "s_" , "d_" .. @define allow-config-dups 1;`
- `- , . " " .`
- `- ;`
- `" " .`

: `s_internal:`

```
source s_internal { internal(); };
```

: :

```
destination d_net { tcp("127.0.0.1" port(1000) log_fifo_size(1000)); };
```

`tcp c - IP-, IP- .`

`, "";`

```
log { source(s_internal); destination(d_net); };
```

`<>(<>)` .

: `s_demo_stream - unix-stream(), - , - max_connection group:`

```
source s_demo_stream1 { unix-stream("<path-to-socket>" max-connections(10) group(log)); };
```

- `;`
- `(inline), , ;`
- `, "#" ;`

:

```
log {
    source(s1); source(s2); ...
    optional_element(filter1|parser1|rewrite1);
    optional_element(filter2|parser2|rewrite2);
    ...
    destination(d1); destination(d2); ...
    flags(flag1[, flag2...]);
};
```

, :

```
source s_localhost {
    network(ip(127.0.0.1) port(1999));
};
destination d_tcp {
    network("10.1.2.3" port(1999) localport(999));
};
log {
    source(s_localhost);
    destination(d_tcp);
};
```

inline- :

```
log {
    source {
        network(ip(127.0.0.1) port(1999));
    };
    destination {
        network("10.1.2.3" port(1999) localport(999));
    };
};
```

syslog-ng DNS, . :

```
options { option1(params); option2(params); ... };
```

:

DNS:

```
options { use-dns(no); };
```

, :

1.

file()	.	
internal()	, syslog-ng.	
network()	BSD-syslog IPv4 IPv6. TCP, UDP TLS..	
pipe()	.	
program()	.	
sun-stream() sun-streams()	STREAMS (Solaris).	
syslog()	IETF-syslog.	
system()	.	

systemd-journal()	, systemd.	
systemd-syslog()	, systemd.	
unix-dgram()	SOCK_DGRAM.	
unix-stream()	SOCK_STREAM.	

2.

elasticsearch elasticsearch2	Elasticsearch. elasticsearch2 Elasticsearch 2 .	
file()	.	
hdfs()	Hadoop Distributed File System (HDFS).	
kafka()	Apache Kafka.	
loggly()	Loggly (https://www.loggly.com/).	
logmatic()	Logmatic.io (https://logmatic.io/).	
mongodb()	MongoDB.	
network()	BSD IPv4 IPv6. TCP, UDP TLS.	
pipe()	.	
program()	.	
sql()	SQL. .	
syslog()	IETF-syslog protocol. TCP, UDP TLS.	
unix-dgram()	BSD SOCK_DGRAM.	
unix-stream()	Linux SOCK_STREAM.	
usertty()	.	

3. .

facility()	(facility).	
filter()	.	
host()	~.	
inlist()	.	
level() priority()	.	
match()	, .	
message()	, .	
netmask()	IP- ~.	
program()	.	
source()	.	
tags()	.	

syslog-ng , **fly-admin-events**

:

```
source remote_events_src {
    network(port(2222) transport(tcp) flags(syslog-protocol));
};
```

:

```
destination astra_remote_events_dst {
    file("/parsec/log/astra/events"
        template("${MESSAGE}\n")
        group("astra-admin")
        hook-commands(
            setup("/usr/bin/astra-protect-event-log")
        )
        overwrite-if-older(2678400)
        persist-name("remote"));
};
```

```
destination astra_unprotect_remote_events_dst {
    file("/parsec/log/astra/remote_events" template("${MESSAGE}\n"));
}
```

:

```
log {
    source(remote_events_src);
    destination(astra_remote_events_dst);
};
```

rsyslog

- /etc/rsyslog.conf;
- .conf /etc/rsyslog.d;

. man rsyslog.conf

logrotate

- /etc/logrotate.conf;
- .conf /etc/logrotate.d;

(/etc/logrotate.conf):

- **weekly** - , ;
- **rotate** <> - , , mail. , , - 4;
- **create** <> <> <> - (postrotate) (, ,). (chmod(2)), , , . , . ncreate;

. man logrotate.

rsyslog/syslog-ng

```
sudo apt install syslog-ng
```

 syslog-ng :

```
sudo syslog-ng -s
```

```
sudo syslog-ng -s
```

```
rsyslog syslog-ng . :
```

rsyslog	syslog-ng
/etc/rsyslog.d/21-cloudinit.conf <pre># Log cloudinit generated log messages to file :syslogtag, isequal, "[CLOUDINIT]" /var/log/cloud- init.log # comment out the following line to allow CLOUDINIT messages through. # Doing so means you'll also get CLOUDINIT messages in /var/log/syslog & stop</pre>	/etc/syslog-ng/conf.d/21-cloudinit.conf <pre>log { source(s_src); filter { message("^.*CLOUDINIT.*\$"); }; destination { file("/var/log/cloud-init. log"); }; flags(final); };</pre>
/etc/rsyslog.d/49-haproxy.conf <pre># Create an additional socket in haproxy's chroot in order to allow logging via # /dev/log to chroot'ed HAProxy processes \$AddUnixListenSocket /var/lib/haproxy/dev/log # Send HAProxy messages to a dedicated logfile :programname, startswith, "haproxy" { /var/log/haproxy.log stop }</pre>	/etc/rsyslog-ng/conf.d/49-haproxy.conf <pre>log { source { unix-dgram("/var/lib/haproxy/dev /log"); }; filter { program("haproxy"); }; destination { file("/var/log/haproxy.log"); }; flags(final); };</pre>
/etc/rsyslog.d/20-ufw.conf	/etc/rsyslog-ng/conf.d/20-ufw.conf

<pre># Log kernel generated UFW log messages to file :msg,contains,"[UFW " /var/log/ufw.log # Uncomment the following to stop logging anything that matches the last rule. # Doing this will stop logging kernel generated UFW log messages to the file # normally containing kern.* messages (eg, /var/log /kern.log) #& stop</pre>	<pre>log { source(s_src); filter { message("^.*UFW.*\$"); }; destination { file("/var/log/ufw.log"); }; # flags(final); };</pre>
/etc/rsyslog.d/jetty9.conf	/etc/rsyslog-ng/conf.d/jetty9.conf
<pre># Send Jetty messages to jetty.out when using systemd :programname, startswith, "jetty9" { /var/log/jetty9/jetty-console.log stop }</pre>	<pre>log { source(s_src); filter { program("jetty9"); }; destination { file("/var/log/jetty9/jetty- console.log"); }; flags(final); };</pre>
/etc/rsyslog.d/postfix.conf	/etc/rsyslog-ng/conf.d/postfix.conf
<pre># Create an additional socket in postfix's chroot in order not to break # mail logging when rsyslog is restarted. If the directory is missing, # rsyslog will silently skip creating the socket. \$AddUnixListenSocket /var/spool/postfix/dev/log</pre>	<pre>log { source { unix-stream("/var/spool/postfix/dev /log" keep-alive(yes)); }; filter { program("postfix"); }; destination(d_syslog); };</pre>
/etc/rsyslog.d/tomcat9.conf	/etc/rsyslog-ng/conf.d/tomcat9.conf
<pre># Send Tomcat messages to catalina.out when using systemd \$template TomcatFormat, "[%timegenerated:::date-year%- %timegenerated:::date-month%- %timegenerated:::date-day% %timegenerated:::date-hour%: %timegenerated:::date-minute%: %timegenerated:::date-second%] [%syslogseverity-text%]%msg%\n" :programname, startswith, "tomcat9" { /var/log/tomcat9/catalina.out;TomcatFormat stop }</pre>	<pre>log { source(s_src); filter { program("tomcat9"); }; destination { file("/var/log/tomcat9/catalina.out", template("[%YEAR-\$MONTH-\$DAY \$HOUR:\$MIN:\$SEC] [%PRIORITY]\$MSG\n")); }; flags(final); };</pre>

astra-create-debug-logs

Astra Linux astra-create-debug-logs, .
:

sudo astra-create-debug-logs

/tmp/.

:

WEB- Apache2	/var/log/apache2/
Cups	/var/log/cups/
IMAP/Pop3 Dovecot	/etc/dovecot/conf.d/10-logging.conf log_path. , : /var/log/dovecot.log
Exim4	/var/log/exim4/
Syslog	/var/log/syslog*
Samba	/var/log/samba/
X server	/var/log/Xorg.*.log
PostgreSQL	/var/lib/postgres/<>/<>/pg_log
	/var/log/kern.log
ALD	/var/log/ald/
parsec	/var/log/parsec/
afick	/var/log/afick/
FTP Vsftpd	/var/log/vsftpd.log.*