LinuC

オープンソースカンファレンス2023 Online/Spring

Linuxシステムログ入門

ALJ Education Plus 株式会社
（LPI-Japanアカデミック認定校）
代表取締役　山本 篤美

LPI-JAPAN

自己紹介

#LinuC学習中

ALJ Education Plus 株式会社
（LPI-Japanアカデミック認定校）
代表取締役　山本 篤美

- **開発/登壇実績**
  - 新入社員/中途社員向け 研修
  - IT資格取得系 研修
  - IT系の専門学校　開発/サーバ/ネットワーク系の授業

LinuCレベル１／レベル２ Version10.0とは

LinuCとは

#LinuC学習中

クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定資格

✓現場で「今」求められている新しい技術要素に対応

- オンプレミス／仮想化を問わず様々な環境下でのサーバー構築
- 他社とのコラボレーションの前提となるオープンソースへの理解
- システムの多様化に対応できるアーキテクチャへの知見

✓全面的に見直した、今、身につけておくべき技術範囲を網羅

今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み

✓Linuxの範疇だけにとどまらない領域までカバー

セキュリティや監視など、ITエンジニアであれば必須の領域もカバー

新しくなった「LinuC」の紹介

Version10.0と従来の出題範囲の比較

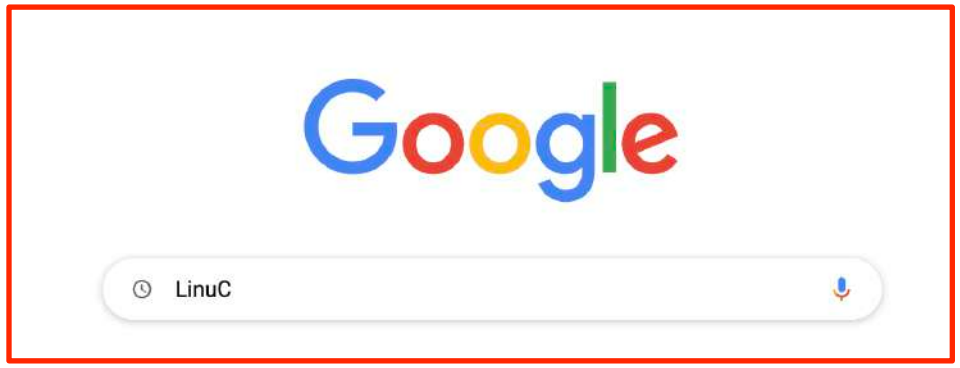
#LinuC学習中

	テーマ	Version 10.0	従来
LinuC-1	仮想化技術	・仮想マシン／コンテナの概念 ・クラウドセキュリティの基礎	←（Version10.0で新設）
	オープンソースの文化	・オープンソースの定義や特徴 ・コミュニティやエコシステムへの貢献	←（Version10.0で新設）
	その他	→（Version10.0で削除）	アクセシビリティ、ディスククォータ、 プリンタの管理、SQLデータ管理、他
LinuC-2	仮想化技術	・仮想マシンの実行と管理（KVM） ・コンテナの仕組みとDockerの導入	←（Version10.0で新設）
	システムアーキテクチャ	・クラウドサービス上のシステム構成 ・高可用システム、スケーラビリティ、他	←（Version10.0で新設）
	その他	・統合監視ツール（zabbix） ・自動化ツール（Ansible）	←（Version10.0で出題範囲に含む）
		→（Version10.0で削除）	RAID、記憶装置へのアクセス方法、 FTPサーバーの保護、他

新しくなった「LinuC」の紹介

- 検索：LinuC　→　https://linuc.org/

#LinuC学習中



新しくなった「LinuC」の紹介

#LinuC学習中

今回のテーマ

Linuxシステムログ入門

主題	1.09 重要なシステムサービス
副題	1.09.2 システムのログ

セミナー内容

主題	1.09 重要なシステムサービス
副題	1.09.2 システムのログ
重要度	5
概要	・rsyslogデーモンを設定できる これには、ログ出力を中央のサーバに送信するようログデーモンを設定できる。 または中央のログサーバーとしてログ出力を受け入れることも含まれる。 ・systemdのジャーナルサブシステムを使用できる ・ログのローテーション、圧縮、削除を自動化できる

セミナー内容

目次

1.ログ管理の概要

ログ管理の全体像

2.ロギング機能

syslogの設定方法

rsyslogの設定方法

systemd-jounalの設定方法

3.ログローテーション

loglotare.confの設定方法

セミナー内容

目次

1.ログ管理の概要

ログ管理の全体像

2.ロギング機能

syslogの設定方法

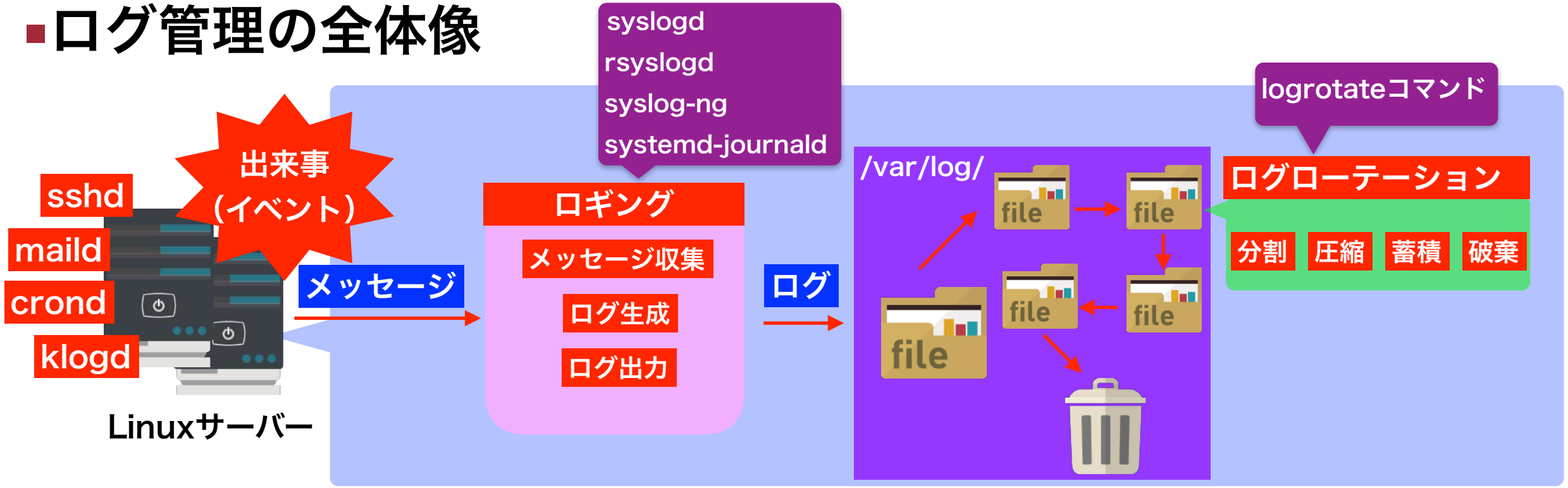
rsyslogの設定方法

systemd jounalの設定方法

3.ログローテーション

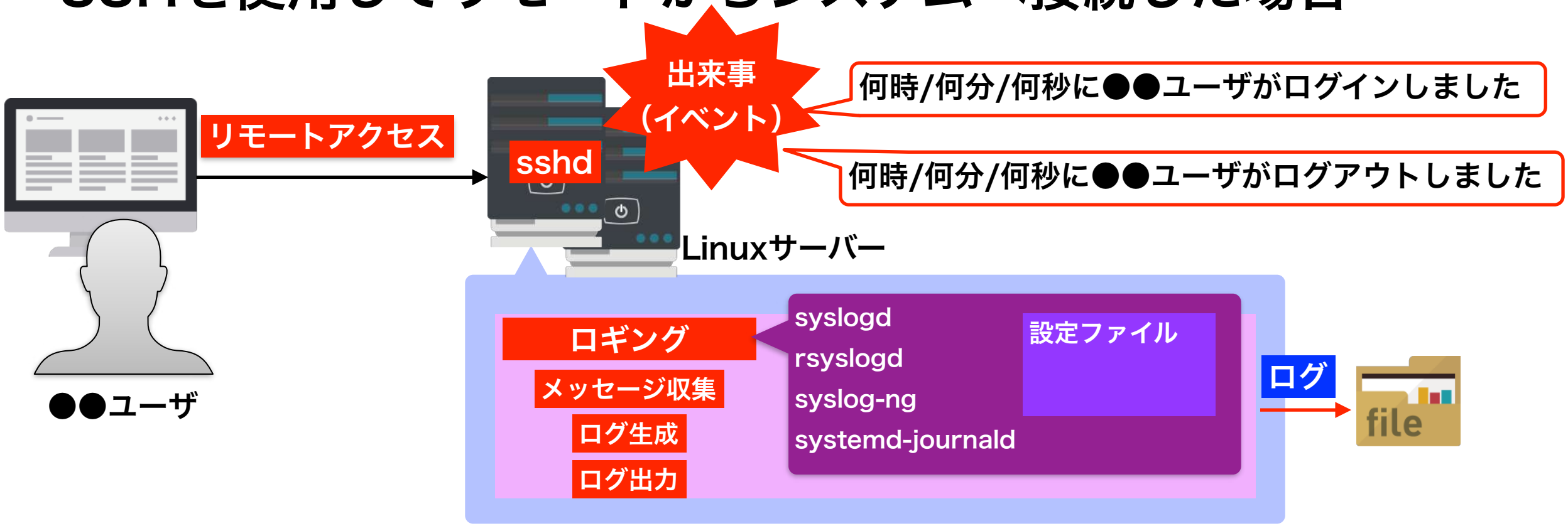
loglotare.confの設定方法

ログ管理の全体像



- ログとは、システムの出来事（イベント）を一定の形式で時系列に記録・蓄積したデータのこと
  → トラブルシューティングを行うときに、もっとも有益な情報源の1つ
- イベント通知メッセージを収集し、ログを生成/出力する行為のことをロギングという
- 一定の期間でログファイルを分割/圧縮/蓄積/破棄する行為のことをログローテーションという

▪SSHを使用してリモートからシステムへ接続した場合



/var/log/secure ファイル

日付 時刻 ホスト名 sshd[PID]:Accepted publickey for ユーザ名 from アクセス元 暗号化方式：暗号化
日付 時刻 ホスト名 sshd[PID]:Disconnected from アクセス元

■tailコマンド（オプションf）

リアルタイムでファイルを監視する

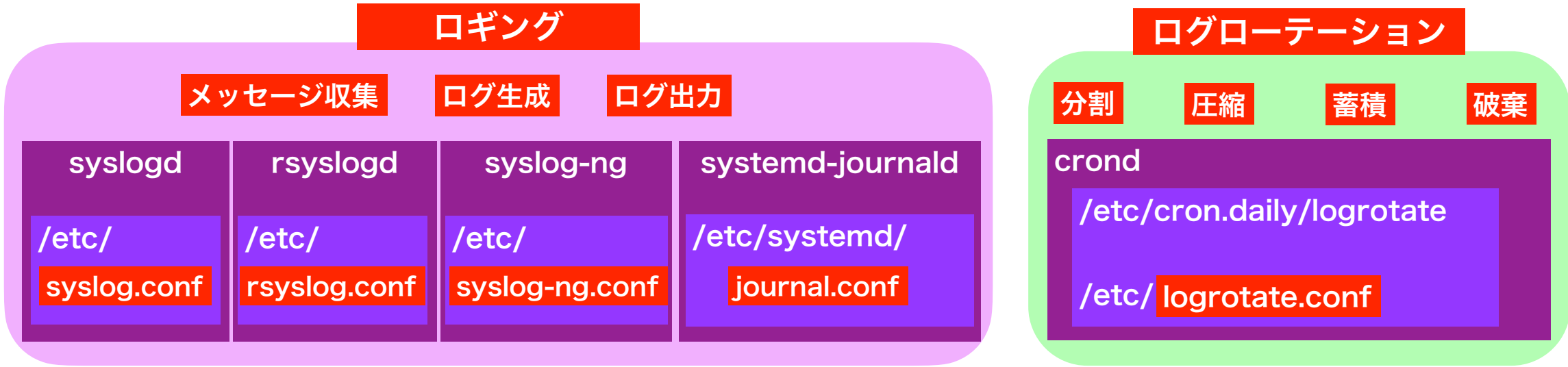
tail -f [ファイル名]

```
# tail  -f   /var/log/secure
Feb 26 15:27:49 centos7_SV1 sshd[1187]: pam_unix(sshd:session): session opened for user vagrant
by (uid=0)
Feb 26 15:31:42 centos7_SV1 sshd[1190]: Received disconnect from 10.0.2.2 port 57870:11:
disconnected by user
Feb 26 15:31:42 centos7_SV1 sshd[1190]: Disconnected from 10.0.2.2 port 57870
Feb 26 15:31:42 centos7_SV1 sshd[1187]: pam_unix(sshd:session): session closed for user vagrant
Feb 26 15:37:26 centos7_SV1 sshd[1221]: Accepted publickey for vagrant from 10.0.2.2 port 57934
```

- 実機確認

  /var/log/secureファイルをtail -fコマンドでファイルを監視したまま、sshログイン、ログアウトを実施し、ログ出力状況を確認する

ログ管理の全体像



- ロギング機能を持ったLinuxで使われているソフトウェア(プロトコル)としてsyslog、rsyslog、syslog-ng、systemd-journalがある
- ログローテーションはcronで動いている（logrotateコマンドが定期的に実行）
- LinuC試験では主に、rsyslog、systemd-journal、ログローテーションが出題される

セミナー内容

- 目次

1.ログ管理の概要

ログ管理の全体像

2.ロギング機能

syslogの設定方法

rsyslogの設定方法

systemd-jounalの設定方法

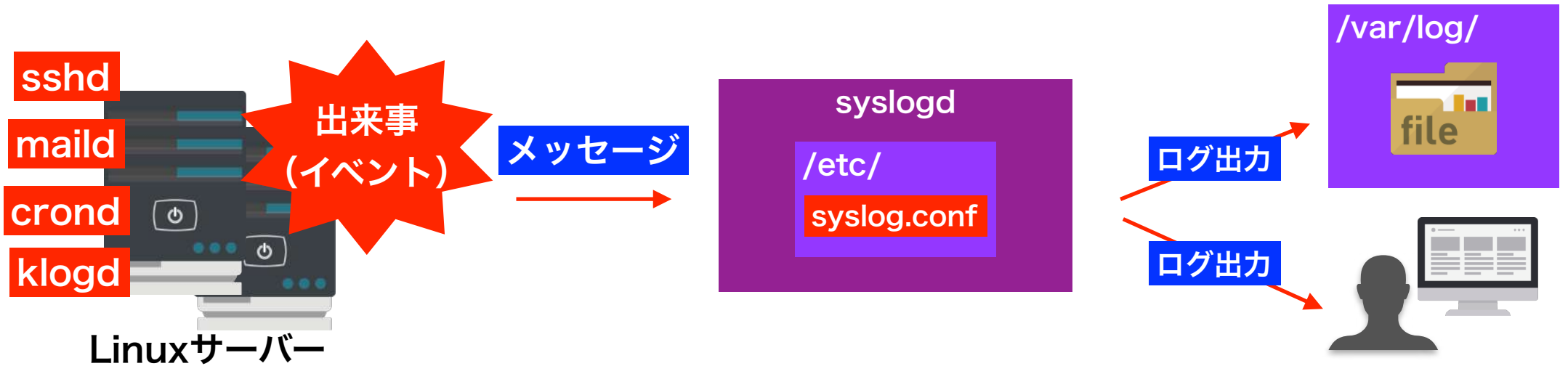
3.ログローテーション

loglotare.confの設定方法

2.ロギング機能（syslog）

syslog(System Logging Protocol)の特徴

最も古くから使用されているロギングプロトコル。CentOS5でデフォルトとして実装されている。



- 1980年代にEric Allman氏が開発した最も古くから使用されているロギングプロトコル
- 元々sendmailの一部として開発され、2001年にRFC3164で標準化された
- その後RFC3164が廃止され、現在はRFC5424で標準化されている
- syslogは、syslogdによって動作しており、/etc/syslog.configの設定によって、収集対象のメッセージの種類とログ出力先が決まる

2.ロギング機能（syslog）

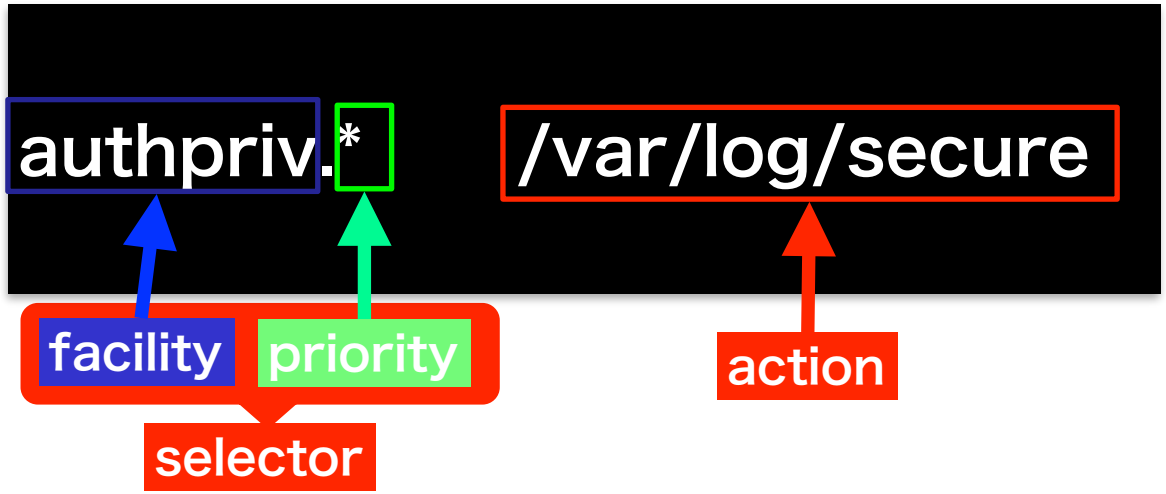
/etc/syslog.conf

syslogdの設定ファイル。selector(facilityとpriority)とactionで構成されている。

書式

selector
メッセージの種類
facility priority

action
ログの出力先



- syslog.confはselectorとactionで構成されており、空白で区切られている
- selectorは、収集対象のメッセージの種類を決める設定項目
- selectorはさらに、facility(収集先)とpriority(重要度)で構成されている
- actionはログの出力先を決める設定項目

2.ロギング機能（syslog）

facilityの種類

メッセージの収集先を表す

コード	内容	syslog.conf指定
0	カーネルメッセージ	kern
1	ユーザーレベルメッセージ	user
2	メールサービス	mail
3	システムデーモン	daemon
4	認証系サービス	auth
5	syslogdデーモン	syslog
6	プリンタシステム	lpr
7	ニュースシステム	news
8	UUCP	uucp
9	クロックデーモン	cron
10	認証メッセージ	authpriv
11	FTPデーモン	ftp
16～23	独自の設定	local0～7

書式

selector　　action

メッセージの種類　　ログの出力先

facility　priority

```
*.info;mail.none;authpriv.none;cron.none    /var/log/messages
authpriv.*              /var/log/secure
mail.*                  -/var/log/maillog
cron.*                  /var/log/cron
*.emerg                 *
uucp,news.crit          /var/log/spooler
local7.*                /var/log/boot.log
cron.=err               /var/log/maillog_err
```

- 収集元のイベントメッセージを0～23までのコード番号で分類している
- facilityを複数指定する場合は、カンマ(,)を指定する

priorityの種類

メッセージの重要度を表す

優先度	コード	priority	内容
高	0	emerg	緊急
	1	alert	警報
	2	crit	致命的
	3	err	エラー
	4	warning	警告
	5	notice	通知
低	6	info	情報
	7	debug	デバック情報
	-	none	出力しない

書式

selector　　action

メッセージの種類　　ログの出力先

facility　priority

```
authpriv.*                  /var/log/secure
mail.*;mail.!crit           -/var/log/maillog
cron.=err                   /var/log/maillog_err
mail.*;mail.!=crit          -/var/log/maillog
```

- priorityは通常、指定したpriority以上の優先度が高いログが出力される
- 指定したpriority以下を出力したい場合は、！を組み合わせて指定する
- 特定のpriorityのみを出力したい場合は = を指定する
- 特定のpriority以外を出力したい場合は、!= を組み合わせて指定する

2.ロギング機能（syslog）

actionの種類

出力先の設定

action	内容
/ファイルの絶対パス	ファイルへ出力
-/ファイルの絶対パス	フラッシュ動作を抑制する
/dev/console /dev/ttyN	コンソールへ出力
| 名前付きパイプ	メッセージを指定した名前付きパイプへ出力する
@ホスト名	リモートホストへ転送
ユーザ名	ユーザへ通知
*	ユーザ全てに通知

書式

selector　　action

メッセージの種類　　ログの出力先

```
authpriv.*          /var/log/secure
mail.*              -/var/log/maillog
cron.*              @192.168.
*.emerg             *
uucp,news.crit      /var/log/spooler
local7.*            /var/log/boot.log
cron.=err           /var/log/maillog_err
```

- ログをファイルに出力したい場合は、ファイルのパスを指定する
- ログをほかのプログラムに渡したい場合は、「|」を指定する
- ログをリモートホストに転送したい場合は「@ホスト名」で指定する
- ログをユーザーのコンソールに表示したい場合は、ユーザー名を指定する

2.ロギング機能（syslog）

- /etc/syslog.confの設定例(デフォルト)

```
# cat /etc/syslog.conf | grep -v '^#' | grep -v '^\s*$'
*.info;mail.none;authpriv.none;cron.none    /var/log/messages
authpriv.*                                  /var/log/secure
mail.*                                      -/var/log/maillog
cron.*                                      /var/log/cron
*.emerg                                     *
uucp,news.crit                              /var/log/spooler
local7.*                                    /var/log/boot.log
```

2.ロギング機能（syslog）

■loggerコマンド

logger　[オプション]　[メッセージ]

オプション	説明
-i(--id)	loggerプロセスのプロセスID（PID）も併せて記録する
-s(--stderr)	記録した内容を標準エラー出力にも出力する
-f ファイル名	指定したファイルの内容をシステムログに追加する
-p(priority) facility.priority	facility.priorityを指定して記録する
-t(--tag) タグ	ログの各行に指定したタグを出力する

```
# logger -i -t maillog -p mail.info 'メール送信しました。'
# tail -n 1 /var/log/maillog
Jan  7 19:46:37 localhost maillog[4186]: メール送信しました。
```

2.ロギング機能（syslog）

■実機確認

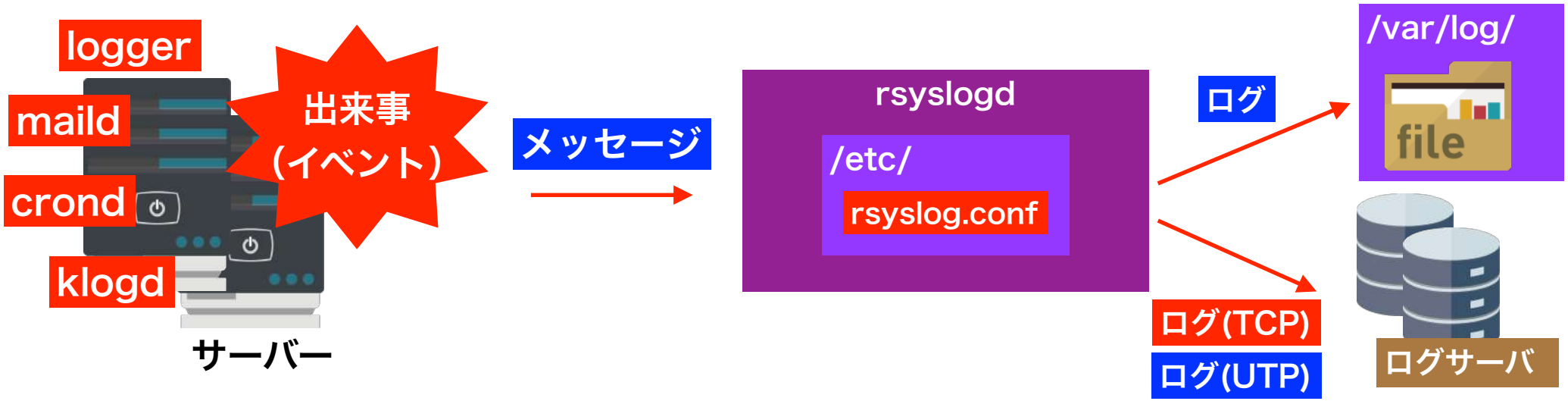
以下のloggerコマンドを実行し、syslog.confファイルの設定どおりにログが出力されているか確認する

#logger -i -t maillog -p mail.info 'メール送信しました。'

```
# logger -i -t maillog -p mail.info 'メール送信しました。'
# tail -n 1 /var/log/maillog
Jan  7 19:46:37 localhost maillog[4186]: メール送信しました。
```

rsyslog(The rocket-fast Syslog)の特徴

syslogの拡張版。rsyslogはreliable-syslogの略。CentOS6からデフォルトで実装されている。



- Reiner Gerhards氏が主開発者として、syslogを元に2004年から開発が始まった
- TCP経由でのログ転送、マルチスレッド対応、各種DBへの書き込み対応
- rsyslogdで動いており、rsyslog.configファイルでログの出力先を設定している
- syslog.confと互換性がある

2.ロギング機能（rsyslog）

/etc/rsyslog.conf

```
#### MODULES ####
$ModLoad imuxsock
$ModLoad imjournal

#### GLOBAL DIRECTIVES ####
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
$IncludeConfig /etc/rsyslog.d/*.conf

#### RULES ####
*.info;mail.none;authpriv.none;cron.none        /var/log/messages
authpriv.*                                      /var/log/secure
mail.*                                          -/var/log/maillog
```

読み込む対象

共通的な設定

ログの処理方法

- rsyslog.conf の設定は主に3つに分かれている
- 「MODULES」はLoad する Module （読み込む対象）を指定する項目
- 「GLOBAL DIRECTIVES」は共通的な設定を指定する項目
- 「RULES」は「Input Module」 から読み込んだログの処理方法を決める項目

2.ロギング機能（rsyslog）

/etc/rsyslog.conf（MODULES）の設定例

```
#### MODULES ####
$ModLoad imuxsock
$ModLoad imjournal

$ModLoad imudp
$UDPServerRun 514
$AllowedSender UDP, 127.0.0.1, *.example.jp, 192.168.0.0/24

$ModLoad imtcp
$InputTCPServerRun 514
$AllowedSender UDP, 127.0.0.1, *.example.jp, 192.168.0.0/24
```

- ローカルロギングのサポート
- systemd journalのサポート
- UDPの受信を許可する場合の設定
- UDPのポートの設定
- 転送元を制限する場合の設定
- TCPの受信を許可する場合の設定
- TCPのポートの設定
- 転送元を制限する場合の設定

- モジュールを追加する場合は「$ModLoad」で指定する
- ローカルのイベントメッセージを取込みたい場合は、「imuxsock」と指定する
- journalからログを読み込みたい場合は、「imjournal」と指定する
- リモートから転送されたログをUDPで受け付ける場合は、「imudp」と指定する
- リモートから転送されたログをTCPで受け付ける場合は、「imtcp」と指定する

2.ロギング機能（rsyslog）

/etc/rsyslog.conf（GLOBAL DIRECTIVES）の設定例

```
#### GLOBAL DIRECTIVES ####
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

$IncludeConfig /etc/rsyslog.d/*.conf

#$ActionFileEnableSync on
```

ログのタイムスタンプの書式

/etc/rsyslog.d/*.conf ファイルの設定も読み込む

フラッシュ動作が抑制（コメントアウトまたはoffを設定する）

- ログタイムスタンプの書式設定は、$ActionFileDefaultTemplateで指定する
- 他の設定ファイルを読み込みたい場合は、$IncludeConfigで指定する
- ログ出力時のバッファのフラッシュ動作設定は、$ActionFileEnableSyncで設定する

/etc/rsyslog.conf（RULES）の設定例

```
#### RULES ####
*.info;mail.none;authpriv.none;cron.none        /var/log/messages
authpriv.*                                      /var/log/secure
cron.*                                          /var/log/cron

*.*                                             @リモートホスト名
*.*                                             @@リモートホスト名

$template mytemplate,"日時：%timegenerated% 内容：%msg%\n"
mail.*                                          /var/log/custom.log;mytemplate
```

/var/log/messagesへログ出力する
/var/log/messagesへログ出力する
/var/log/messagesへログ出力する
リモートホストへUDPで伝送する
リモートホストへTCPで伝送する
ログ出力の形式をmytemplateという名前で定義
mytemplateを適用

- syslog.confと同じ設定方法に加えて、リモートホストへの転送(TCP)が可能
- $templateで、出力先や出力メッセージの形式をカスタマイズすることが可能
(%マクロ名%で設定可能)

■実機確認（rsyslog（リモートホストへの転送(TCP)））

```
# logger -i -t maillog -p mail.alert 'mail test123'
```

出来事
（イベント）

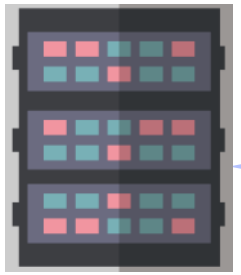
サーバA
(192.168.33.2)

メッセージ

rsyslogd
/etc/
rsyslog.conf

```
#### RULES ####
mail.alert              @@192.168.33.11
```

転送



サーバB
(192.168.33.11)

rsyslogd
/etc/
rsyslog.conf

```
#### MODULES ####
$ModLoad imtcp
$InputTCPServerRun 514

#### RULES ####
mail.*                  -/var/log/maillog
```

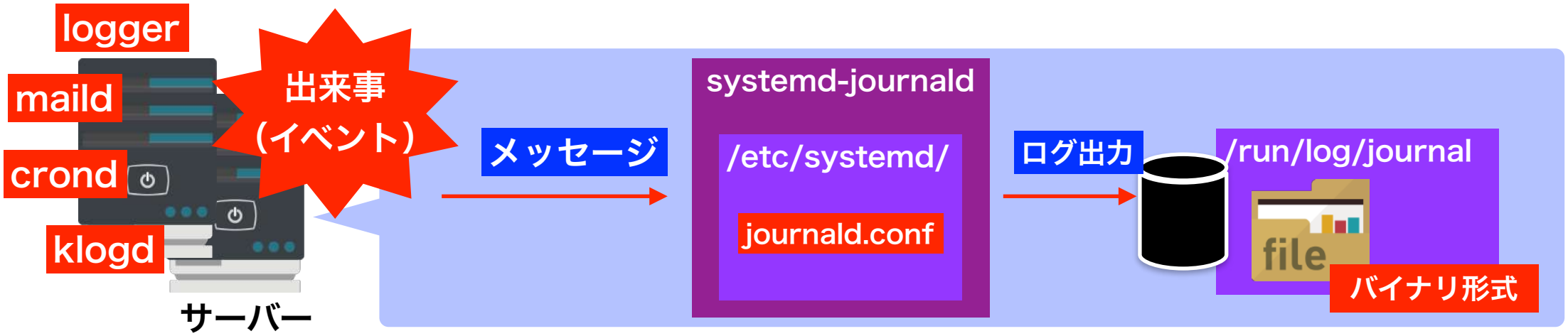
ログ

/var/log/

2.ロギング機能（systemd-journald）

systemd-journald（systemd-journald.service）

systemdのジャーナルサブシステム。



- systemdを採用しているシステムではsystemd-journaldでログを一元管理している
- systemd-journaldは、systemdから起動したプロセスの標準出力やsyslogへのログメッセージをバイナリ形式で記録している
- systemd-journaldの設定ファイルは/etc/systemd/journald.confである
- systemd-journaldで収集したログはjournalctlコマンドで参照することができる

2.ロギング機能（systemd-journald）

/etc/systemd/journald.conf

```
# cat /etc/systemd/journald.conf
        ・
        ・
        ・
[Journal]
#Storage=auto
```

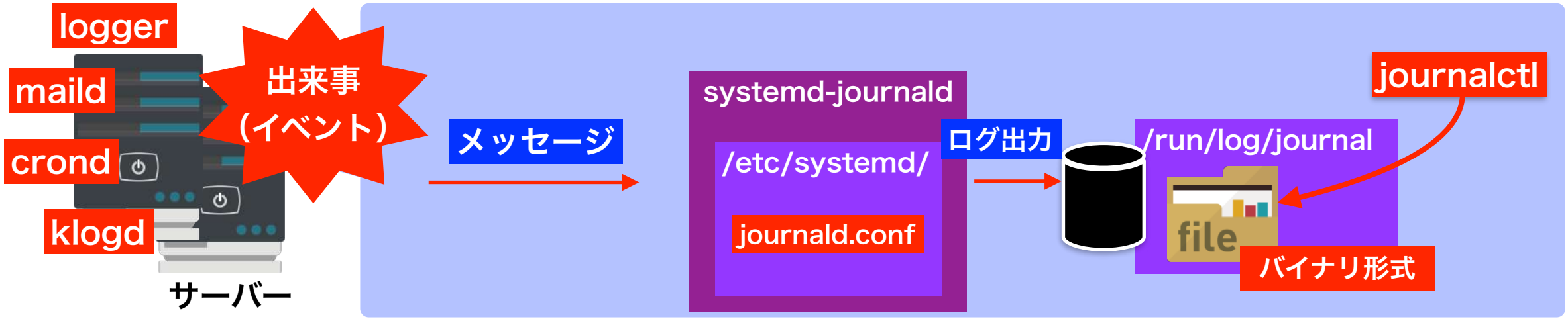
Storageの設定値	意味
volatile	/run/log/journal に記録する
persistent	/var/log/journalに記録する
auto	/var/log/journalに記録する。もしなければ、/run/log/journalに記録する
none	ログを記録せず破棄する

- CentOS7のデフォルトは全て設定項目がコメントアウトされている
- デフォルトのログの出力先は、メモリー上（/run/log/journal/配下）になっており、再起動するとログが消えてしまう
- 永続的にログを残したい場合は、Storage=persistentで設定する（もしくは/var/log/journalディレクトリを作成しておく）

2.ロギング機能（systemd-journald）

journalctlコマンド

systemd journalに格納されたログを参照するコマンド



- systemd-journaldで収集したログはjournalctlコマンドで参照することができる
- ログがカテゴライズされて保管されている、期間を指定して参照できる点が特徴
- オプション無しで実行すると、記録されているログを最初からすべて表示する
  （デフォルトはページャを使って、ページ分割して表示）

2.ロギング機能（systemd-journald）

journalctlコマンド

systemd journalに格納されたログを参照するコマンド

journalctl　[オプション]　[検索文字列]

オプション	説明
-a (--all)	画面表示できない文字列を表示する
-b (--boot)	特定のシステム起動時のログを表示する
-f (--follow)	新規に追加されたログをリアルタイムに表示する
-k (--dmesg)	カーネルからのメッセージを表示する
-l (--full)	画面で表示可能な全てのログを表示する
-n (--lines)	直近のログから指定行数分を表示する（デフォルトは10行）
-o (--output)	ログの出力形式を設定する 通常より詳細（verbose）、JSON形式(json)が指定可能
-r (--reverse)	最古のログからの表示（デフォルト）の逆にし、最新のログから表示
--sence	指定した日付時刻以降のログを表示
-u (--unit)	特定のログからのユニットを表示
-x	追加の説明を表示（メッセージカタログがある場合） ※エマージェンシーモードで起動時に”journalctl -xb”でログを確認できるように表示される

検索文字列	説明
_PID=[プロセス番号]	プロセス番号の指定
_UID=[ユーザ番号]	ユーザIDを指定
_SYSTEMD_UNIT=[ユニット名]	Unit名を指定(オプションの-uと同じ)

2.ロギング機能（systemd-journald）

journalctlコマンド実行例（–fオプション）

tail–fのようにログ監視する

```
# journalctl -f
```

journalctlコマンド実行例（–uオプション）

特定サービスのメッセージだけを表示

```
# journalctl -u sshd
```

2.ロギング機能（systemd-journald）

■systemd-cat

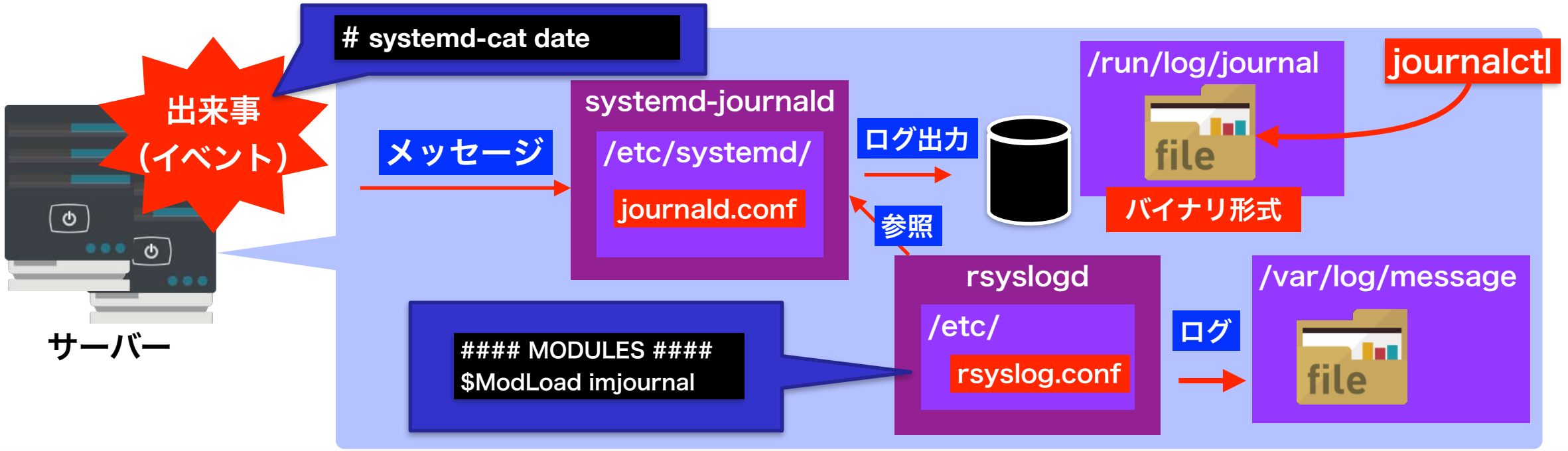
Systemdを採用したシステム上でメッセージをログに記録するコマンド
（指定したコマンドの出力結果をログに記録することができるコマンド）

systemd-cat　[オプション]　[コマンド]

```
# systemd-cat date
# journalctl
Jan 07 19:01:01 CentOS7_SV anacron[2480]:
Jan 07 19:01:01 CentOS7_SV anacron[2480]:
Jan 07 19:04:47 CentOS7_SV systemd[1]: Starting Cleanup of Temporary Directories…
Jan 07 19:04:47 CentOS7_SV systemd[1]: Started Cleanup of Temporary Directories.
Jan 07 19:15:42 CentOS7_SV date[2487]: Thu Jan  7 19:15:42 UTC 2021
# tail -n 1 /var/log/messages
Jan  7 19:15:42 CentOS7_SV journal: Thu Jan  7 19:15:42 UTC 2021
```

2.ロギング機能（systemd-journald）

■実機確認（systemd-catコマンド）




```
# systemd-cat date
# journalctl
Jan 07 19:15:42 CentOS7_SV date[2487]: Thu Jan  7 19:15:42 UTC 2021
# tail -n 1 /var/log/messages
Jan  7 19:15:42 CentOS7_SV journal: Thu Jan  7 19:15:42 UTC 2021
```

- 目次

1.ログ管理の概要

ログ管理の全体像

2.ロギング機能

syslogの設定方法

rsyslogの設定方法

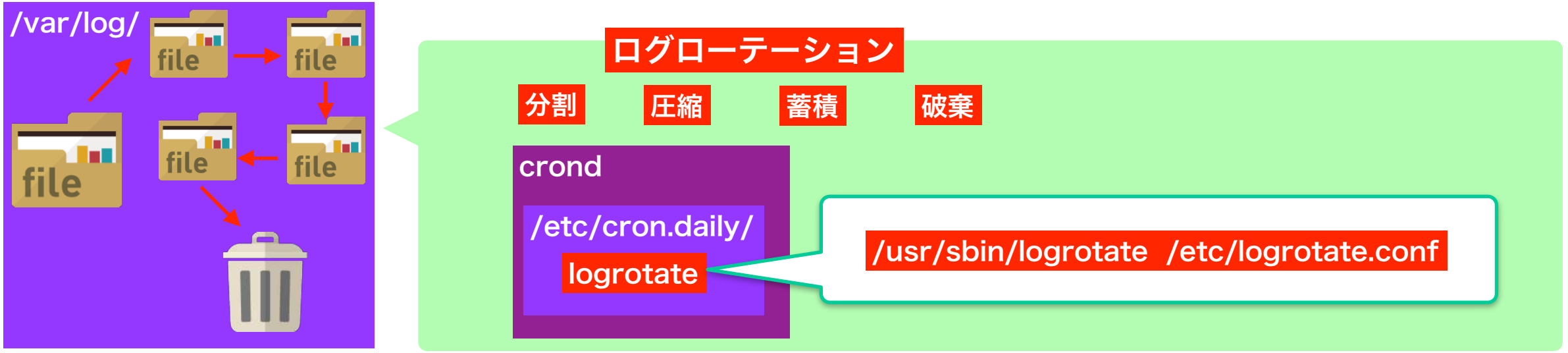
systemd jounalの設定方法

3.ログローテーション

loglotare.confの設定方法

ログローテーションとは

一定の期間でログを分割/圧縮/蓄積/破棄する行為のこと



- ログローテーションはloglotateコマンドによってログを分割/圧縮/蓄積/破棄できる
- loglotateコマンドはcronに登録されており一定の間隔で実行されている
- デフォルトでloglotateコマンドの設定ファイルに/etc/logrotate.confが指定されている

logrotateコマンド

ログローテーションを実行するコマンド

logrotate　[オプション]　設定ファイル

オプション	説明
-d(-debag)	デバッグモードで実行する
-f(-force)	強制的にログローテーションを実行する
-m(-mail)	ログをメール送信する際に使用する

```
# logrotate -f /etc/logrotare.conf
# ls -l /var/log/messages*
-rw-------. 1 root root 940539565 Jan  7 18:50 /var/log/messages
-rw-------. 1 root root    438499 Dec 14 07:41 /var/log/messages-20201214
-rw-------. 1 root root    299964 Dec 22 06:43 /var/log/messages-20201222
-rw-------. 1 root root    233022 Dec 29 03:39 /var/log/messages-20201229
-rw-------. 1 root root    103364 Jan  3 08:25 /var/log/messages-20210103
```

logrotate.confの設定例

```
weekly
su root syslog
rotate 4
create
include /etc/logrotate.d
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}
/var/log/btmp {
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}
```

logrotate.confの主要な設定項目

項目	説明
daily / weekly / monthly/ yearly	ログの切り替えタイミング（日次/週次/月次/年次）
rotate	ローテーションする世代数を設定する 周期がweeklyで4世代の場合、4週間分のログを保存する
compress	ローテーション後のファイルをgz形式で圧縮する
create	ローテーション後に空ファイルを作成する 以下の形式で、作成するファイルの権限を決められる create [権限] [オーナ] [グループ]
dateext	ローテーション後のファイル名に日付を付与する
include ディレクトリ名	指定したディレクトリ配下の設定ファイルを読み込む
mail メールアドレス	ログ切り替えによって削除されるログの内容を指定されたメールアドレスへメールする
postscript～endscript	ログ切り替え後に実行するスクリプト
presctipt～endsctipt	ログ切り替え前に実行するスクリプト
missingok	ログファイルが存在しない場合も、エラーとしない

実機確認（logrotateコマンド）

logrotate -f /etc/logrotare.conf　を実行し、設定ファイルどおりログローテートが実行できるか確認する

```
# logrotate -f /etc/logrotare.conf
# ls -l /var/log/messages*
-rw-------. 1 root root 940539565 Jan  7 18:50 /var/log/messages
-rw-------. 1 root root    438499 Dec 14 07:41 /var/log/messages-20201214
-rw-------. 1 root root    299964 Dec 22 06:43 /var/log/messages-20201222
-rw-------. 1 root root    233022 Dec 29 03:39 /var/log/messages-20201229
-rw-------. 1 root root    103364 Jan  3 08:25 /var/log/messages-20210103
```

ご清聴ありがとうございました