LinuC

LinuC レベル 1 Version10.0
技術解説無料セミナー

2024/2/17 開催

主題1.09：重要なシステムサービス

INTERNOUS

インターノウス株式会社

（LPI-Japanアカデミック認定校）

竹本 季史

LPI-JAPAN

会社紹介、自己紹介

■会社紹介：インターノウス株式会社

- ●人材紹介サービス、人材派遣/SESサービス、IT未経験者の教育及び就職支援サービス、法人研修サービス
- ●未経験からインフラエンジニアやプログラマーになりたい方へ、無料で研修と就職支援サービスを行っています。
  https://programmercollege.jp/session/

■自己紹介：竹本 季史(たけもと ときふみ)

- ●IT業界で約10年間勤務後、インターノウス株式会社エンジニアカレッジ講師。
- ●これまで約1000人を未経験者からエンジニアに養成。Linuxサーバー(メール、OpenSSH、シェルスクリプト、DB、監視、演習)を担当。
- ●LinuCレベル1バージョン10.0の差分教材で「仮想マシン・コンテナの概念と利用」を執筆。

Linux技術者認定「LinuC（リナック)」とは

■LinuCとは

クラウド／DX時代のITエンジニアに求められるシステム構築から運用管理に必要なスキルを証明する技術者認定です。

✓クラウド活用に役立つスキルの習得

- オンプレミス／仮想化・コンテナ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

✓習得できるスキルが実践的

問題作成にはトップエンジニアも参加するコミュニティ内の意見を取り込むことで、本当に必要な内容を網羅的に盛り込んでいます。

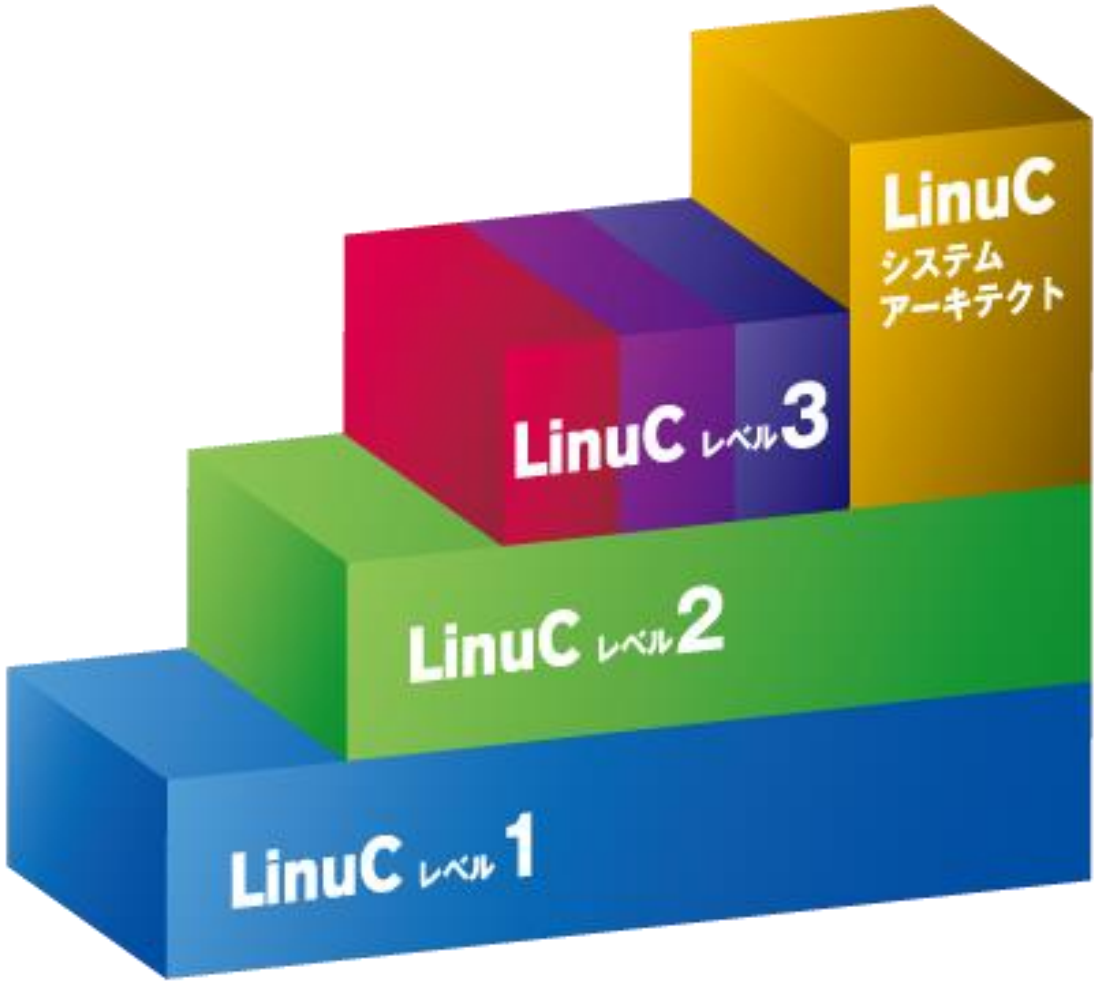
✓上流工程を担うアーキテクトの領域までカバー

システムの運用管理からアーキテクチャ設計までの４つのレベルをひとつずつ習得していくことで、活躍できるエンジニアとして必要なスキルを網羅的に身につけていくことができます

LinuC
Linux技術者認定

LinuCの４つのレベル

LinuCは、サーバーの運用管理からアーキテクト設計まで、システム開発・運用に必要な知識とスキルを体系立てて習得することができます。



LinuC システムアーキテクト
ITプロジェクトを成功に導く上級エンジニア
SA01試験 | SA02試験

LinuC レベル3
高度な技術力を備えた特定分野のスペシャリスト
304試験（仮想化&高可用性）
300試験（混在環境） | 303試験（セキュリティ）

LinuC レベル2
仮想マシン・コンテナを含むLinuxシステム、ネットワークの設計・構築
201試験 | 202試験

LinuC レベル1
物理／仮想Linuxサーバーの操作・運用
101試験 | 102試験

本セミナーについて

- **解説するポイント**
  - 主題1.09 重要なシステムサービスの「1.09.1システム時刻の管理」、「1.09.2システムのログ」、「1.09.3メール転送エージェント(MTA)の基本」について総合的に解説をします。
  - ネットワーク内の相互に関連するシステムの時刻を同期させることは大変重要です。なぜなら、システムの時刻がログが記録された時刻、メールが送信された時刻となるからです。
  - ログはシステムによっては膨大な量になるため、ファシリティ、プライオリティ、出力先をコントロールすることで、管理者が把握しやすいように適切にログを絞り込む工夫が必要です。
  - また、重要なログについてはメール転送エージェント(MTA)を使って複数のシステム管理者に通知することができます。
  - 本セミナーでは、詳細な設定項目よりもそれぞれの副題の関連性を理解することを目指します。

本セミナーについて

- **解説する主題・内容**
  - 主題1.09：重要なシステムサービス
    - 1.09.1 システム時刻の管理
    - 1.09.2 システムのログ
    - 1.09.3 メール転送エージェント(MTA)の基本

- **参加者の想定スキルレベル**
  - Linux初級者
  - システム運用初級者

- **セミナーのゴール**
  - 時刻同期、ログ管理、メール通知の重要性と関連性を理解することができる。

本セミナーについて

- 本セミナー講師の実行環境
  - OS：Windows 11 x86-64
  - 仮想マシン：VMWare Workstation 17 Player
  - LinuxOS：Almalinux9.3 2台

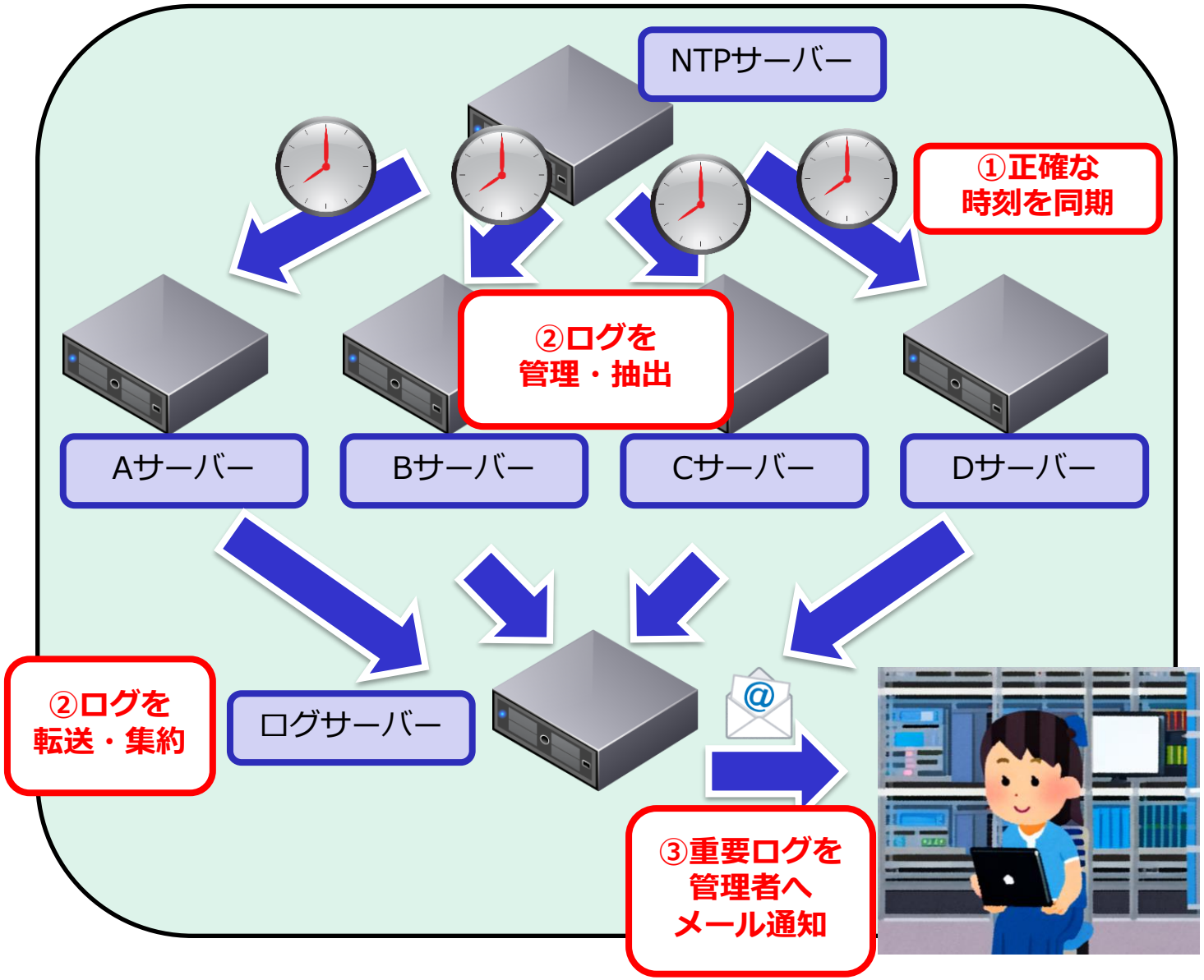
なぜ重要なシステムサービスなのか？

- ひとつのITシステムは多数のサーバーやネットワーク機器が連携します。今回のテーマであるNTPもログもメールも右図のように連携しています。

- **システムの時刻を正確な状態に保つ**
  - システムのログ管理、メール通知、スケジュール実行などに正確な時刻が必要です。

- **適切なログの管理**
  - 適切なログの管理により、システムの状態を常に管理できる。

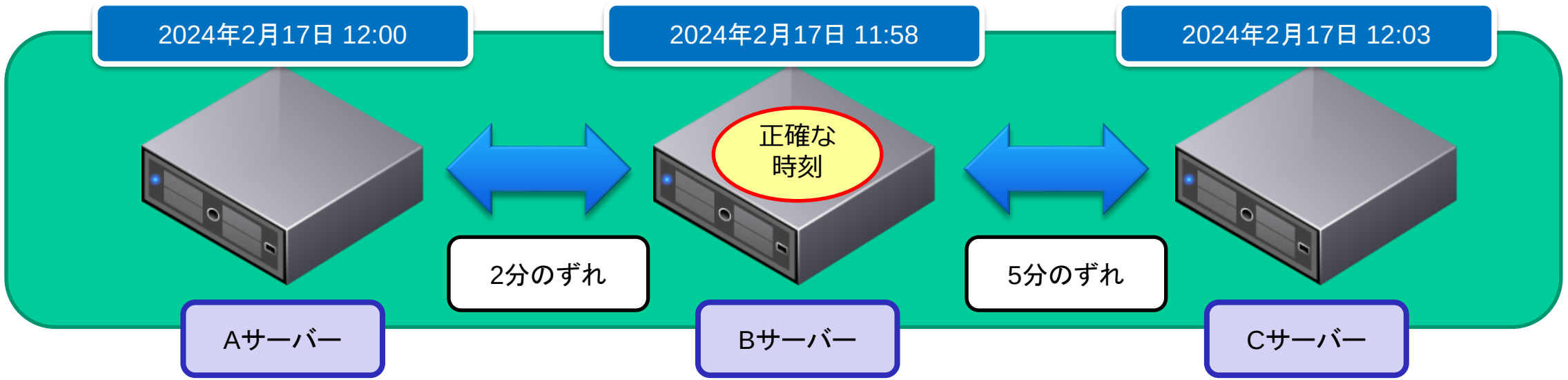
- **システム管理者へメール通知**
  - システム管理者が重要な通知を知るためにメールが役立ちます。



1.09.1 システム時刻の管理

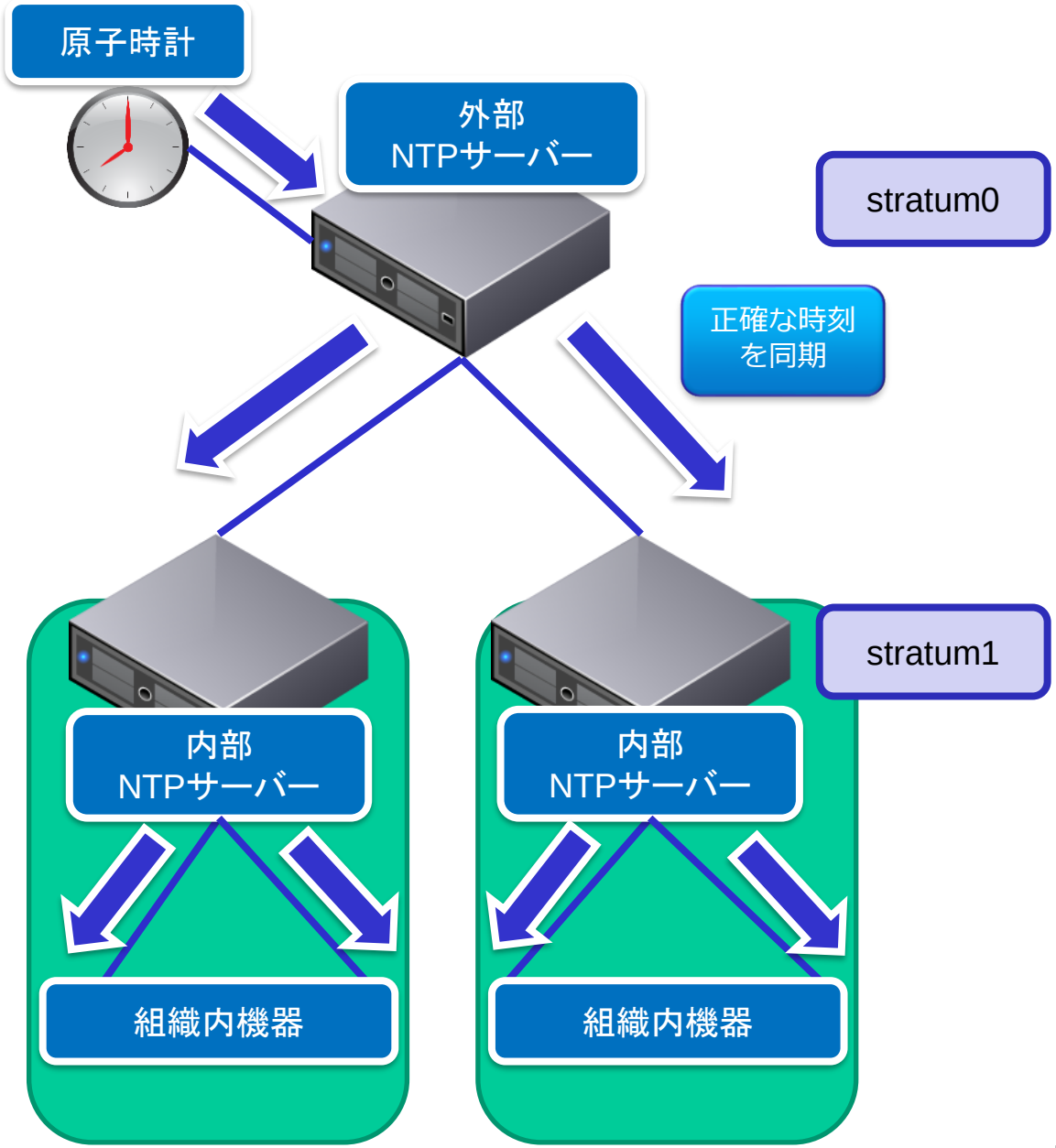
システム時刻を正確に合わせる重要性

- コンピュータのシステム時刻は正確であることが重要です。
  特に連携するサーバーとの時刻がずれていると次のような問題が生じます。
  - タスクをスケジュール実行したときに、誤った時刻に実行されて他ホストの実行処理に問題が出る
  - 認証サーバーと連携できずに認証に失敗する
  - 障害が発生したときに、ログに記録された時刻が他ホストとずれていて、障害復旧の手間が増大する



システムの時刻を正確に保つには

- システムの時刻は補正しないままだと、ずれが大きくなっていきます。正確な時刻情報を持つサーバーと同期することで、システムの時刻を補正します。
- NTP (Network Time Protocol)はネットワーク経由でシステムの機器の時刻を同期するプロトコルです。
- NTPサーバーはstratum(階層)構造です。
  原子時計やGPSなどの正確な時刻ソースを持つNTPサーバーをstratum0、これを参照するサーバーをstratum1と呼びます。stratum2,stratum3・・と参照するサーバーが続きます。
- 各機器がそれぞれstratum0を参照すると負荷が増大するため、組織内にNTPサーバーを立てて、それを組織内の機器が参照します。



システム時刻の管理　設定

- CentOS7以降ではchronyが標準のNTPサーバー/クライアントとなっています。
- chronyは起動時に強制的に設定ファイルに記載のあるNTPサーバーから時刻情報を取得してシステムクロックを合わせます。その後もNTPサーバーと同期をとり続けます。
- 設定ファイルは、/etc/chrony.confです。下表は主な設定項目です。

設定項目	説明
pool	NTPサーバプールを指定します。プールとは複数台のNTPサーバーが登録されたホスト名のことを指します。例えば、2.centos.pool.ntp.orgを指定すると、複数のIPアドレスが返されます。
server	プールではなく、特定のNTPサーバーを指定します。
driftfile	システムクロックとNTPサーバーの時間の差分を記録するファイルのパスを指定します。
makestep	システムクロックをNTPサーバーと一気に合わせる条件を指定します。
rtcsync	ハードウェアクロックとの同期を有効にします。
allow	NTPサーバーにする場合、ローカルで許可するNTPクライアントのIPアドレス範囲を指定します。
logdir	ログファイルの保存先ディレクトリを指定します。

chronyによる時刻同期のデモ

- 意図的に時刻をずらした後にchronyで正確な時刻を同期します

1. chronyc activity（NTPサーバーと同期している状況を確認）
2. chronyc sources（NTPサーバーの同期元ソースを確認）
3. systemctl status chronyd（chronydの起動を確認）
4. systemctl stop chronyd（chronydを停止）
5. systemctl status chronyd（chronydの停止を確認）
6. date（現在のシステムクロックを確認）
7. timedatectl set-time "2025-01-01 00:00:00"（システムクロックを2025年1月1日0時0分0秒に変更）
8. date（システムクロックが変更されていることを確認）
9. systemctl start chronyd（chronydを起動）
10. systemctl status chronyd（chronydを起動を確認）
11. date（システムクロックが現在の正確な時刻に修正されていることを確認）

chronyによる時刻同期のデモ

- 内部NTPサーバーを構築し、NTPクライアントから内部NTPサーバーと時刻同期します

NTPサーバー側

1. vi /etc/chrony.conf （allowでローカルのサブネットからの接続を許可)
2. systemctl restart chronyd (設定を反映させる)
3. firewall-cmd --add-service ntp --permanent (firewallのntpを永続的に許可)
4. firewall-cmd --list-services --permanent (ntpの許可を確認)
5. firewall-cmd --reload (設定を反映させる)

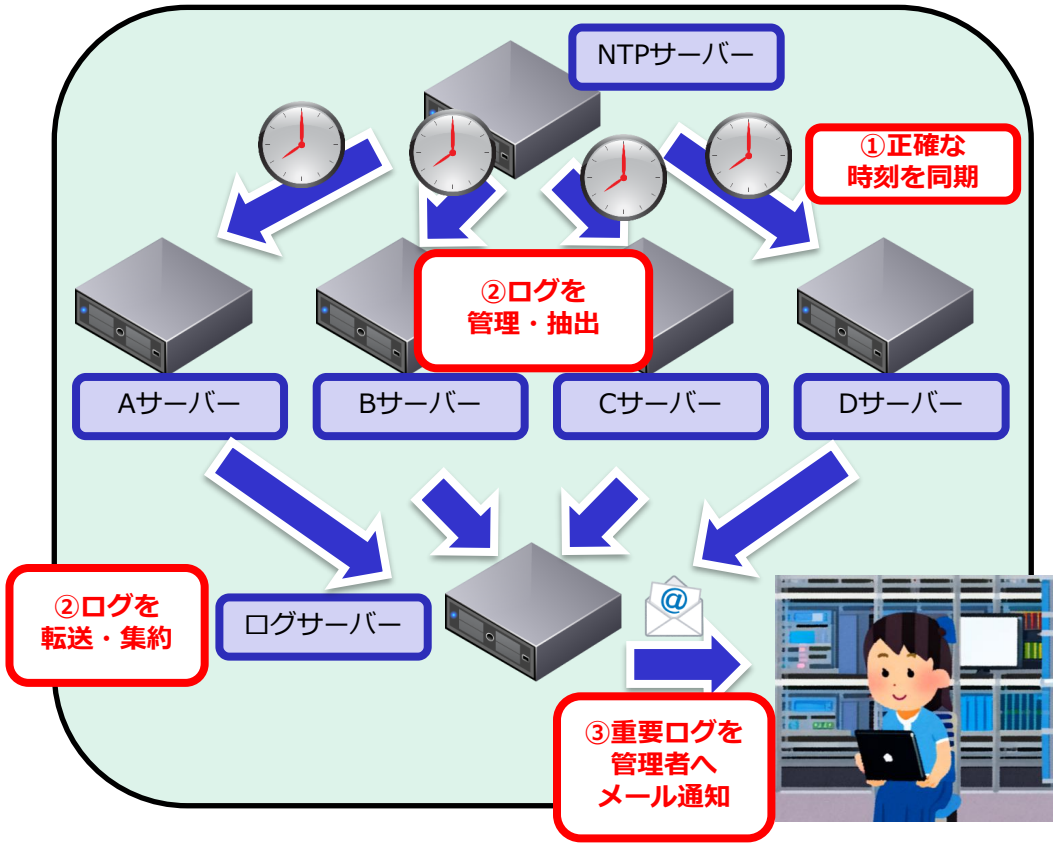
NTPクライアント側

1. vi /etc/chrony.conf （poolをコメントして、serverで内部NTPサーバーを指定)
2. systemctl restart chronyd (設定を反映させる)
3. chronyc sources (時刻同期状況の確認)

1.09.2 システムのログ

システムのログの重要性

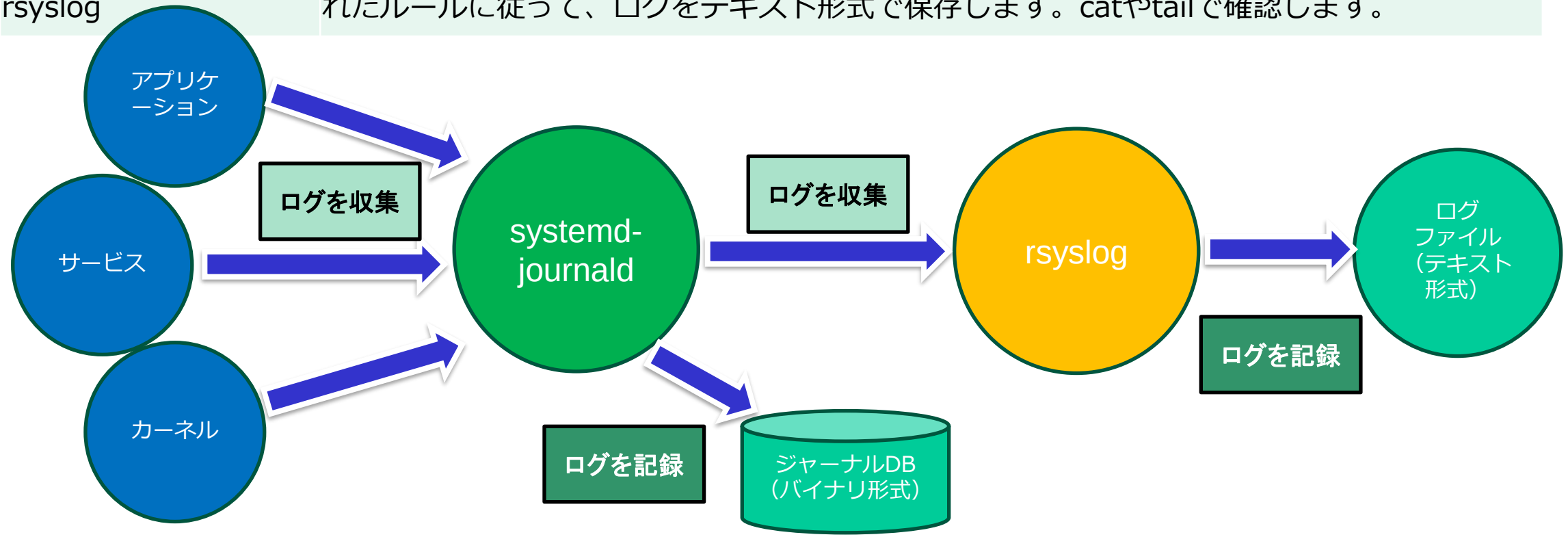
- ログは、カーネルや様々なサービス、プロセスなどのシステムの動作状況を記録したものです。
- 収集したログは、システム管理者が障害を診断したり、パフォーマンスを最適化、セキュリティインシデントの調査、障害の予防に役立てたりするために使用されます。
- ログを全て収集すると莫大な量になるので、システム管理者はシステムに合わせて適切にログを絞り込みます。
- 各ホストでログを収集するほかに、ログを特定のログサーバーに送信して同一ネットワークのログを集中管理することができます。



システムのログ　ログ収集と保管の流れ

- systemd採用のLinuxでは2つのサービスが連携してログの収集と保管を行います

ログ収集サービス	説明
systemd-journald	systemdの一部であり、システムやサービスのログを一手に集めます。これらのログはバイナリ形式で保存され、/run/log/journal(RedHat系)または/var/log/journal(Debian系)に保存されます。journalctlで確認します。
rsyslog	従来からのログ収集・格納サービスです。systemd-journald からログを受け取り、設定されたルールに従って、ログをテキスト形式で保存します。catやtailで確認します。



システムのログ　設定ファイル rsyslog.conf

- 設定ファイル/etc/rsyslog.conf
  の設定カテゴリは下表のとおりです。

設定カテゴリ	説明
GLOBAL DIRECTIVES	全体的な設定をします。rsyslogが使用する作業ディレクトリやinclude(別ファイルから設定を読み込む)するファイルなどを記述します。
MODULES	ロードするモジュールを指定します。 デフォルトでsystemd-journaldのログを取り込むimjournalがロードされて有効となっています。従来のsyslog収集方法のimuxsockはロードされますが無効となっています。 他のホストからのsyslogをUDP受信するimudp、TCP受信するimtcpがありますがデフォルトでは無効です。
RULES	ログの出力ルールを設定します。セレクタ（出力対象ログの設定）とアクション（出力先設定）で構成されます。詳細は次ページへ。

システムのログ　設定ファイル RULES書式

- /etc/rsyslogd.confのRULESでは、収集するログのルールを次の書式で設定します。

ファシリティ.プライオリティ　出力先

ファシリティ	内容
auth,authpriv	ログインなどの認証による出力
cron	cronによる出力
deamon	デーモンによる出力
kern	カーネルによる出力
lpr	印刷システムによる出力
mail	メールサービスによる出力
user	ユーザアプリケーションによる出力
local0～7	ローカルシステムによる出力

優先度	プライオリティ	内容
高	emerg	緊急
↑	alert	早急に対応が必要
↑	crit	深刻な事象
↑	err	一般的なエラー
↑	warning	一般的な警告
↑	notice	一般的な通知
↑	info	一般的な情報
低	debug	デバック情報
-	none	ログを出力しない

出力先	出力先の例	内容
ログファイルの絶対パス	/var/log/message	ログファイルに出力
コンソールのパス	/dev/tty1	コンソールに出力
@[ホスト名 or IP]:514	@192.168.1.1:514	ホストにUDPで出力
@@[ホスト名 or IP]:514	@@192.168.1.1:514	ホストにTCPで出力
ユーザー名	usera	ユーザーの端末に出力
*	*	ログイン中の全てのユーザーの端末に出力

システムのログ　設定ファイル RULES設定例

- RULESの設定例を次の表に示します
  - ファシリティやプライオリティに*(アスタリスク)を使うと、全てのファシリティやプライオリティを選択したことになります。
  - 同じ出力先の場合、複数のファシリティをセミコロン ;で複数指定できます。
  - 通常プライオリティはそれ以上のログを出力しますが、.=でそのプライオリティのログのみを出します。

RULES設定例	説明
*.info;mail.none;authpriv.none;cron.none /var/log/messages	全てのファシリティでプライオリティがinfo以上のログを/var/log/messagesに出力し、ファシリティがmail, authpriv, cronのログは出力しない
cron.* /var/log/cron	cronファシリティの全てのプライオリティを/var/log/cronに出力する
*. warning @@192.168.223.154:514	全てのファシリティでプライオリティがwarning以上のログを192.168.223.154のTCP514番ポートに送る
user.=info /var/log/messages	userファシリティのinfoのプライオリティ**のみ**を/var/log/messeagesに出力する

システムのログ　ログの確認とログへの書き込み

- ログの確認コマンド

コマンド	実行例	説明
journalctl	journalctl -u chronyd	systemd-journald収集ログを確認します 実行例は、-u でchronydサービスのjournalのみを表示します
cat, tail, headなど	tail –f /var/log/messages	rsyslog収集ログを確認します 実行例は、tail –f でログファイルを監視して、最新ログが追記される様子を見ることができます。

- ログへの書き込みコマンド

コマンド	実行例	説明
systemd-cat	systemd-cat df	コマンドの実行結果をjournalに書き込みます 実行例は、dfコマンドの結果をjournalに書き込みます
logger	logger –p user.info "Syslog Test"	ログファイルに直接メッセージを書き込みます 実行例は、user.infoのセレクタでrsyslog.confのRULESの設定に従ってログ出力します。ログ出力のテストにも使われます。

システムのログ　ログ転送のデモ

- 他ホストのログをrsyslogサーバーにTCP転送します

rsyslogサーバー（ログ受信側）

1. vi /etc/rsyslog.conf（module imtcpを有効化）
2. systemctl restart rsyslog（設定を反映させる）
3. firewall-cmd --add-port=514/tcp --permanent（firewallでsyslog受信のTCP514番を永続的に許可する）
4. firewall-cmd --list-ports --permanent (syslog受信のTCP514番の許可を確認)
5. firewall-cmd --reload (設定を反映させる)
6. tail -f /var/log/messages (ログ送信側のloggerのメッセージを確認する。ログの時刻が送信側と同じであることも確認。)

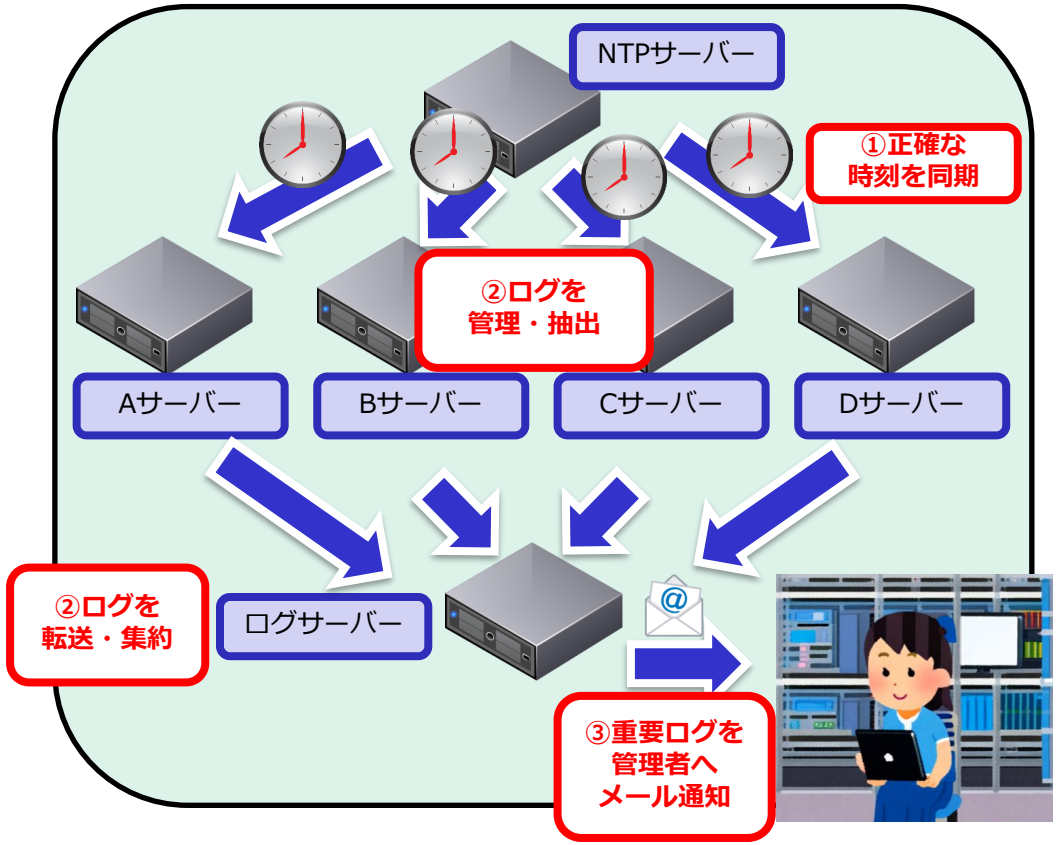
NTPサーバー（ログ送信側）

1. vi /etc/rsyslog.conf （RULESでuser.infoのログをrsyslogサーバーにTCPで送信)
2. systemctl restart rsyslog (設定を反映させる)
3. logger -p user.info "Syslog Test" (テストメッセージをログに書き込む)
4. tail /var/log/messages (テストメッセージの確認)

1.09.3 メール転送エージェント(MTA)の基本

メール転送エージェント(MTA)の重要性

- メールはシステム管理者がシステムに関する重要な情報を通知するために必要な仕組みです。
- 管理者権限を持っていなくとも、担当の業務に関するメールを受信するために、メールを転送することができます。
- メールが頻繁に来ても対応できないので、緊急度、重要度が高いものだけに絞り込みます。



メール転送エージェント(MTA)の基本　メールサーバーを構成するプログラム

- メールサーバーを構成するプログラムには次のものがあります

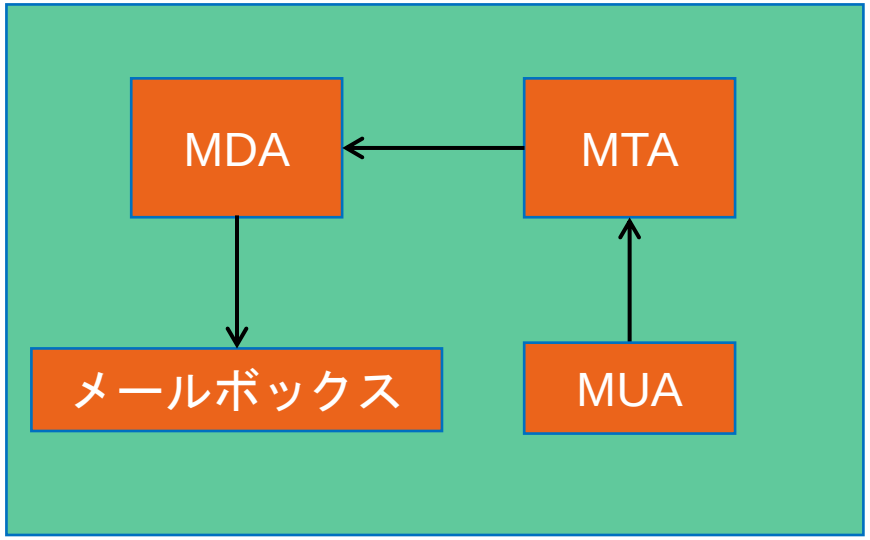
略称	名称	内容
MTA	Mail Transfer Agent	メール配送を担うプログラムです。 SMTP（Simple Mail Transfer Protocol）を使って、他のメールサーバーのMTAにメールを転送したり、受信したメールをMDAに配送したりします。
MDA	Mail Delivery Agent	MTAが受信したメールをローカルのメールボックスに配信するプログラムです。
MUA	Mail User Agent	ユーザーがメールを送受信するプログラムです。 WindowsではOutlookが有名です。 LinuxのCUIではmailコマンド、GUIではThunderbird、Evolutionがあります。

- 主要なMTAには次のものがあります

MTA	内容
sendmail	UNIXで古くから使われてきたMTAで様々な状況に対応できる柔軟性がありますが、設定方法が複雑でセキュリティ対応が難しく、近年使われなくなる傾向にあります。
exim	sendmailとの互換性があり、単一のプログラムで動作します。近年Linuxで多く使われています。
postfix	sendmailとの互換性があり、複数のプログラムで動作します。動作が早く、セキュリティが強化されています。近年Linuxで多く使われています。

メール転送エージェント(MTA)の基本　MTAの役割　ローカル配信

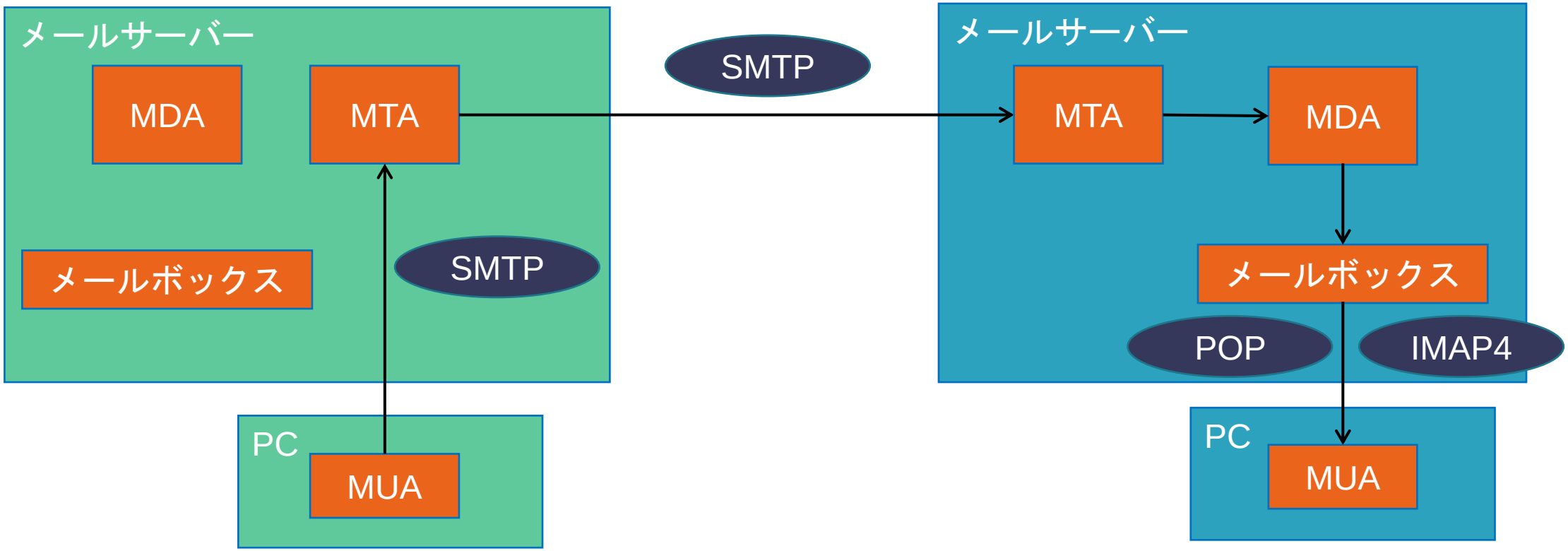
- LinuxサーバーのMUAからLinuxユーザーのメールボックスへローカル配信する場合



26

メール転送エージェント(MTA)の基本　MTAの役割　メールリレー

- PCのMUAからメールサーバーをリレーして宛先メールアドレスへ送る場合



27

メール転送エージェント(MTA)の基本　postfixインストール

- Almalinux９はデフォルトでpostfixとmailコマンドが入っていませんのでインストールします

1. dnf install –y postfix s-nail (postfixとs-nailパッケージのインストール)
2. systemctl start postfix (postfixの起動)
3. systemctl status postfix (postfixの状態確認)

メール転送エージェント(MTA)の基本　メール送信と転送設定

- mailコマンドでメールを送信できます。
  - mail –s 題名 ユーザー名orメールアドレス
  - <本文>
  - Ctrl＋Dで送信

```
[root@logserver ~]# mail -s "test" root
To: root
Subject: test

This is test mail.
^D
-------
(Preliminary) Envelope contains:
To: root
Subject: test
Send this message [yes/no, empty: recompose]? yes
```

- 受信メールを宛先ユーザー以外に自動転送することができます。
  設定は2通りの方法があります。

設定ファイル	記述例	説明
/etc/aliases	root:usera,userb@example.com	転送元:転送先ユーザー名やメールアドレスを記述します。 newaliasesで設定を反映させる必要があります。
~/.forward	usera,userb@example.com	宛先ユーザーのホームディレクトリ配下に作成して、転送先のユーザー名やメールアドレスを記述します。

メール転送エージェント(MTA)の基本　ログ監視とメール送信デモ

- 特定の文字列を検索して一致したらrootにメール送信します。
- また、/etc/aliasesでroot宛のメールをuseraに転送します。

1. grep "chrony" /var/log/messages | mail -s "Alert: chrony found" root
   (chronyの文字を/var/log/messagesから検索した結果をrootにメールする)
2. mail (メールを確認)
3. vi /etc/aliases (rootに届いたメールをuseraに転送の設定)
4. newaliases (設定を反映させる)
5. grep "chrony" /var/log/messages | mail -s "Alert: chrony found" root
   (再度chronyの検索結果をrootにメールする)
6. mail (メールを確認)
7. su – usera (useraに切り替え)
8. mail (メールを確認)

まとめ

以上、見てきたように

- ①時刻同期
- ②ログ
- ③メール通知

は連携していることが分かります。

正確な時刻のログを管理し、
重要なログを管理者へ通知することで、システムの健全性を保っています。