LinuC

LinuC レベル 1 Version10.0
技術解説無料セミナー

2021/5/30 開催

【現場業務の解説】
検証環境におけるサーバー構築

本日の講師

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://engineercollege.jp/lp/

■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800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。

LinuCレベル１／レベル２ Version10.0とは

■LinuCとは

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

✓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・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
- 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

クラウドを活用できるITエンジニアに必須の技術がまとまっている

AWSなどのパブリッククラウドを活用するための技術

間が欠けている状態

オンプレミスのサーバーサイドLinux技術

【今まで／その他】

AWSなどのパブリッククラウドを活用するための技術

仮想マシン／コンテナ技術、クラウドセキュリティ、アーキテクチャ、ほか

オンプレミスのサーバーサイドLinux技術

LinuC Version10.0

■本セミナーについて

- 初級インフラエンジニアの仕事のひとつとして挙げられるのが「検証環境におけるサーバー構築」です。Linuxのインストール、ネットワーク周りの設定、リモートログインの設定、システム時刻の同期設定という、どのサーバーにも始めに共通して設定する項目を解説します。
- プロジェクトによって用語や実施順序は変わる可能性がありますので、必ず現場で確認してください。
- セミナーの内容はCentOS7を前提とした内容となっています。

■本セミナーのねらい

- 今回、インフラエンジニアが業務ベースで見たときに、仕事でどのようにLinuCで学ぶ範囲が役に立つのかを主眼に置きました。
- コマンドの書式の説明には重点を置いていません。
- 主題、副題を横断的にまたいでいますし、一部、LinuCレベル1の範囲を超えてレベル2も含まれます。ご了承ください。
- このように学ぶことで、Linuxを「主題ごと」という点ではなく、線で捉えることができます。さらには、立体的にLinuxを扱うことができ、学ぶことが楽しくなってくるでしょう。

本セミナーについて

■本セミナーに該当するLinuCレベル1の試験範囲

- 副題1.01.1：Linuxのインストール、起動、接続、切断と停止
- 主題1.07：ネットワークの基礎
- 副題1.09.1：システム時刻の管理
- 副題1.10.3：暗号化によるデータの保護

■受講者の想定スキルレベル

- LinuCレベル1の取得を目指している方

■本セミナーのゴール

- 実際の業務におけるサーバー構築での大切なポイントが理解できる
- サーバー構築に共通する設定項目に関してLinuC出題範囲のどこを学習すれば良いかがわかる

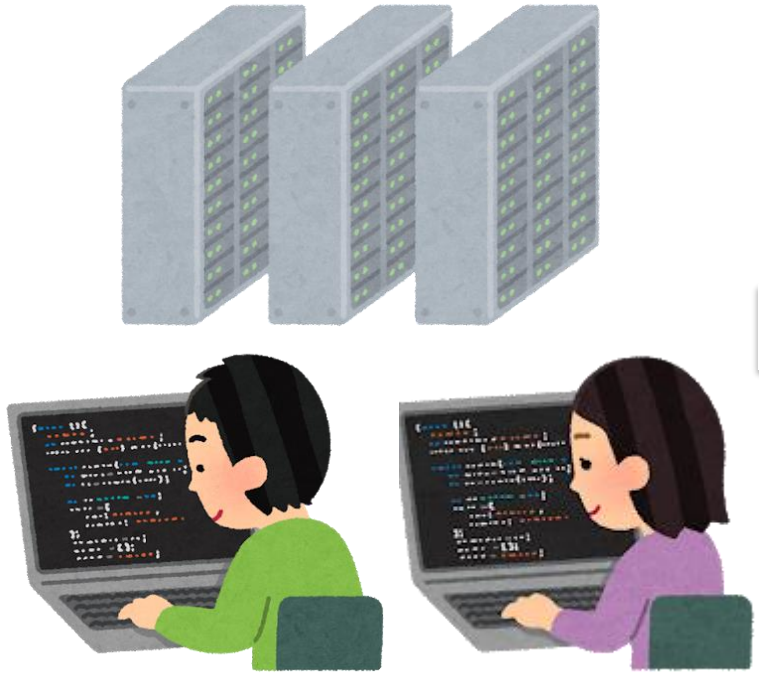
目次

1. ITシステムの構築フェーズ（段階）
2. 本番環境と検証環境
3. 今回解説するLinuxサーバーの設定項目
   - ① Linuxのインストール
   - ② ネットワーク設定
   - ③ システムクロックの同期
   - ④ SSHでのリモートログインと公開鍵認証

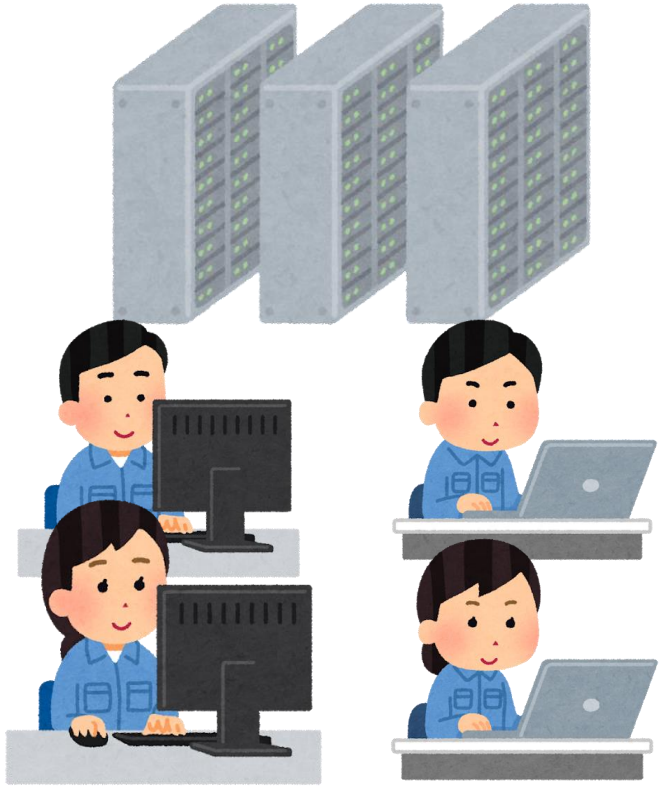
1. ITシステムの構築フェーズ（段階）

●ITシステムにおける構築は、設計の後、運用の前のフェーズとなります。

設計



構築



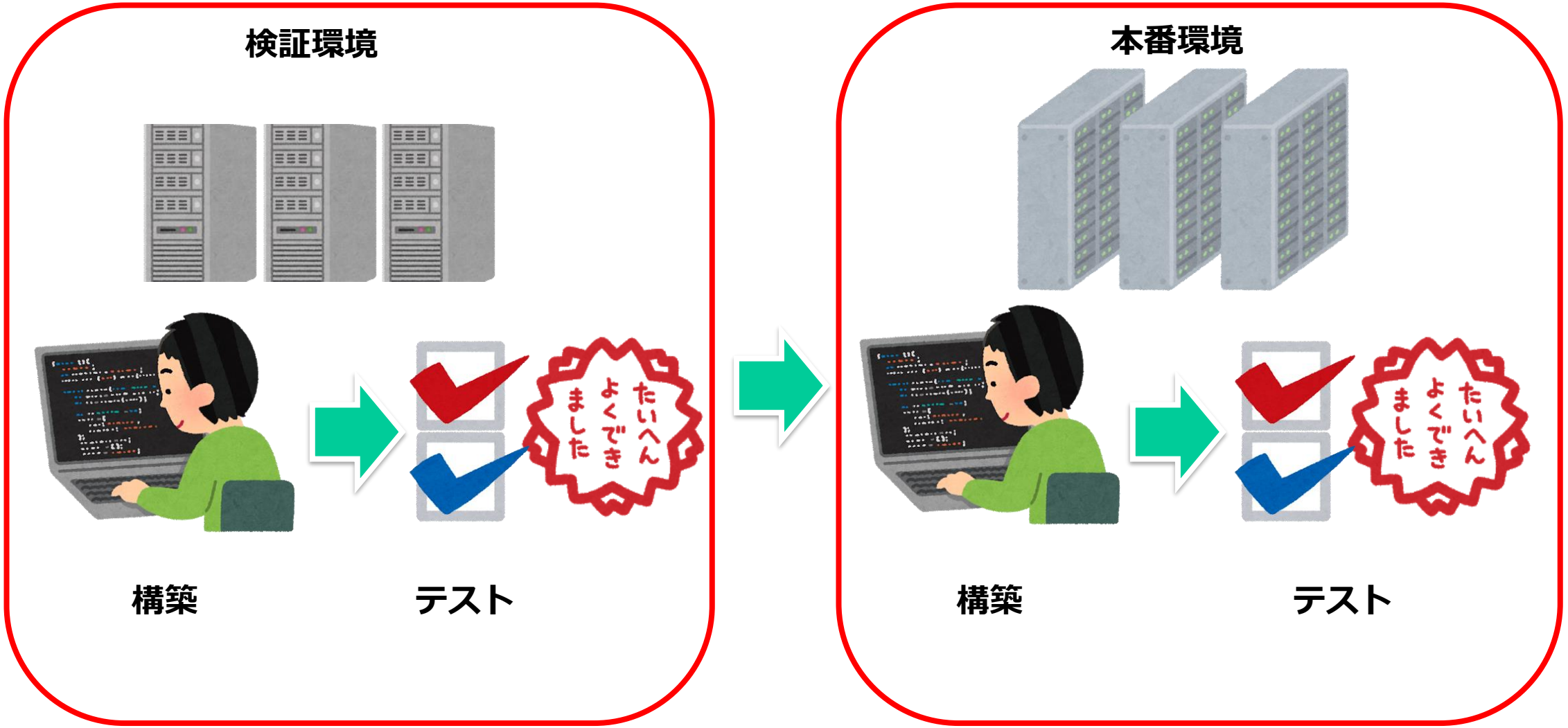
運用

2. 本番環境と検証環境

- 顧客が使用する**本番環境**を構築する前に**検証環境**を構築します。
- 検証環境とは
  - 自社内に本番環境に近づけた環境を構築して、本番環境の構築手順書の作成や構築したシステムのテストなどを実施します。
  - 検証環境は本番環境と同じにするのが望ましいですが、コスト削減のためにネットワークの冗長化など一部機能を削減されることがあります。
  - 検証環境は他の業務に影響を与えないために、別ビルまたは業務エリアとは別に作られます。
  - 検証環境を構築することは新人エンジニアの仕事のひとつになります。

2. 本番環境と検証環境

●検証環境で構築とテストを行ってから本番環境で構築とテストをします。



3. 今回解説するLinuxサーバーの設定項目

●検証環境におけるLinuxサーバーのセットアップを想定して、
Linuxのインストールおよび OS周りの下記項目を解説します。

アプリケーション	
ミドルウェア	DNS、Web、 SSHなどのサーバーソフトウェアの設定 （LinuCレベル2相当）
OS	ネットワーク設定、時刻同期、リモートログイン設定 などOS周りの設定 （LinuCレベル1相当）

今回の解説範囲

3. 今回解説するLinuxサーバーの設定項目

【現場業務の解説】
検証環境におけるサーバー構築

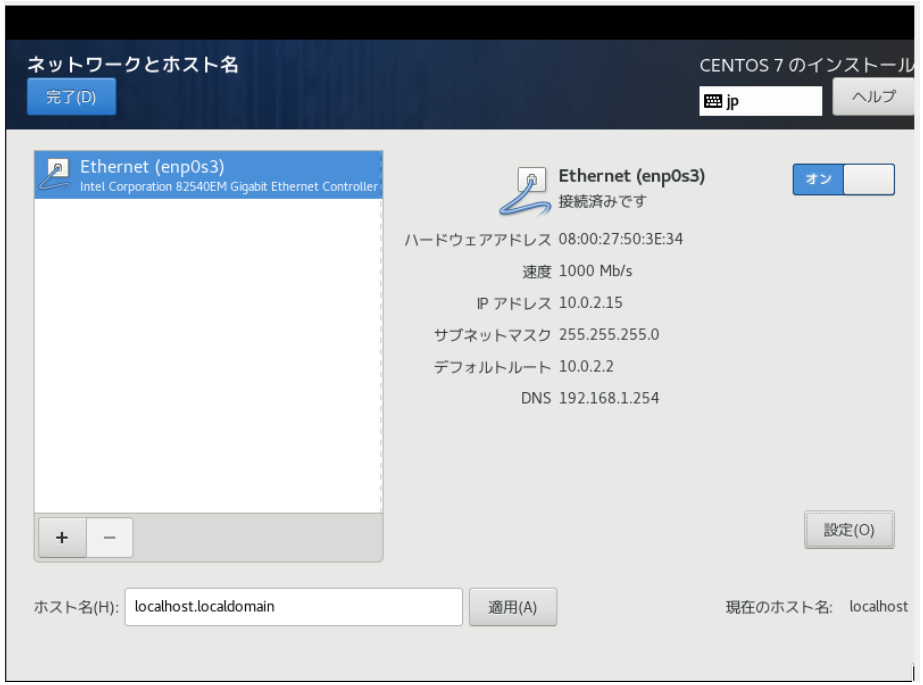
① Linuxのインストール

副題1.01.1：Linuxのインストール、起動、接続、切断と停止

Linuxのインストールについて

- Linuxのインストールには様々な方法がありますが、DVDメディアからインストールする方法が一般的です。

- CentOS7の場合、インストール時に設定できる項目は下記のとおりです。
  - タイムゾーン、時刻同期、言語設定、キーボードレイアウト
  - ストレージのパーティション設定
  - ネットワーク設定（ホスト名、IPアドレス、ゲートウェイ、DNSサーバー指定、ルーティング設定）
  - インストールするパッケージグループ
  - rootパスワード設定
  - 一般ユーザーの作成とパスワード設定



ホスト名、ネットワーク設定画面

Linuxのインストールについて

- パッケージ管理工数の削減、セキュリティを考慮して【インストールするパッケージグループ】は「最小限のパッケージ」を選択します。
- Linuxインストール後、必要に応じてyumやrpmでパッケージをインストールします。
- 本番環境が仮想マシンの場合には、検証環境も仮想マシンを使用します。その場合、あらかじめ作成済みのLinuxイメージをインポートすることがあります。
- 本番環境がクラウドサービスの場合には、検証環境もクラウド事業者の用意したOSイメージを使うのでインストール作業は不要です。

3. 今回解説するLinuxサーバーの設定項目

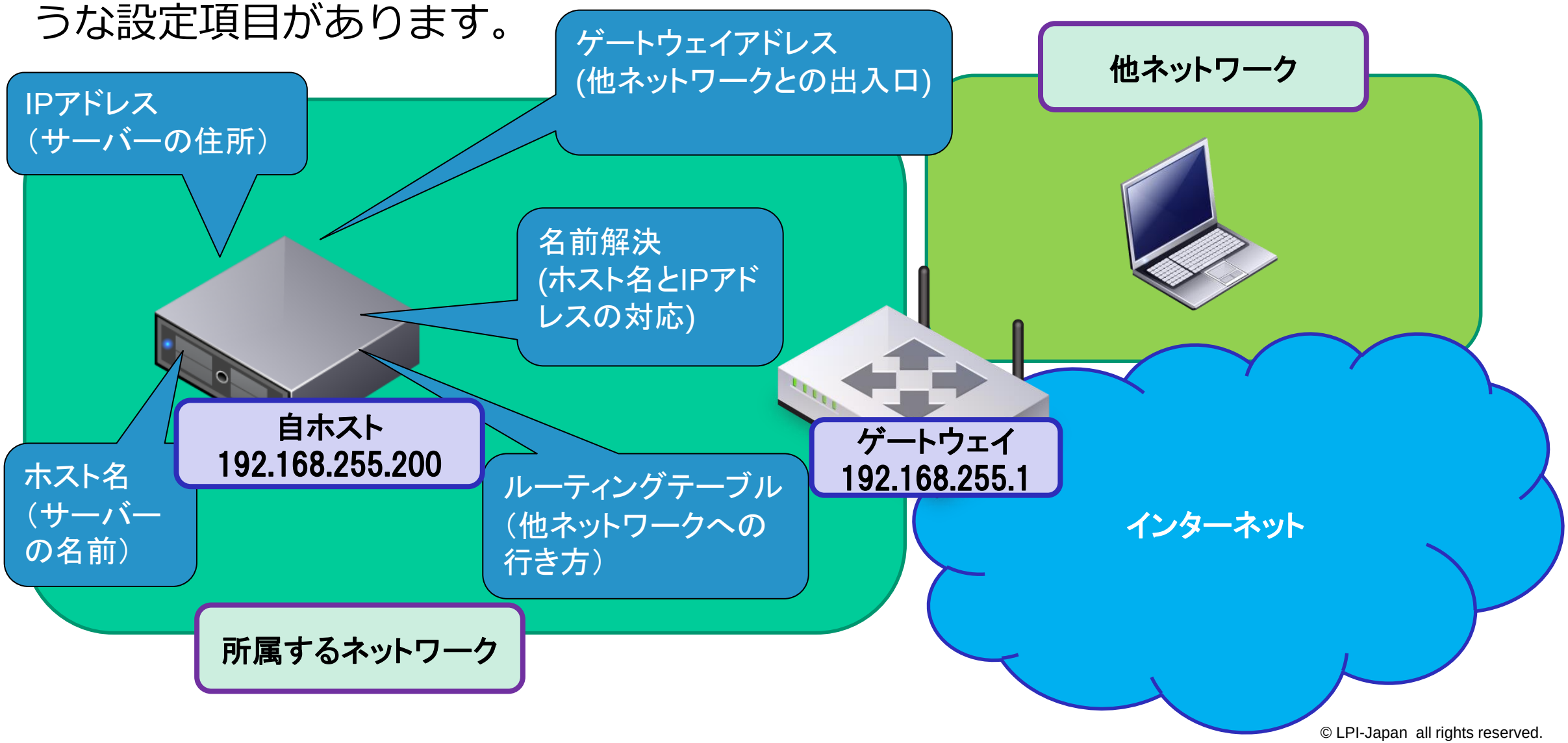
【現場業務の解説】
検証環境におけるサーバー構築

② ネットワーク設定

主題1.07：ネットワークの基礎

ネットワーク設定の全体像

●ネットワーク設定は、他ホストと通信するサーバーには必須の設定です。次のような設定項目があります。



ネットワーク設定項目の役割とコマンド/ファイル

	役割の説明	コマンド/ファイル
IPアドレス	サーバーの住所。所属するネットワークのアドレスを割り当てる。 クライアントPCはDHCPでIPを自動的に割り当てる(動的IP)が、サーバーはIPを固定して使用する(静的IP)ため手動で割り当てる。	ip address nmcli connection ping
ホスト名	IPアドレスでは人が覚えにくいのでサーバーにweb01などの名前をつける。 名前の付け方プロジェクト内の命名規則に従う。	hostnamectl set-hostname nmcli general hostname /etc/hostname
名前解決	ホスト名をIPに変換するために必要。 ローカルとDNSに問い合わせる2種類ある。	/etc/hosts (ローカル) /etc/resolv.conf (DNSの問い合わせ先) host, dig
ゲートウェイアドレス	他ネットワークとの出入口。設定しないと自ネットワーク内のホストとしか通信できない。 デフォルトゲートウェイは宛先ネットワークへの経路がないときに送るゲートウェイのアドレス。	ip route nmcli connection
ルーティングテーブル	他ネットワークへの行き方を示した経路一覧表。ゲートウェイを経由してどのネットワークに行くことができるかを記載している。	ip route nmcli connection

ネットワークの設定と確認の実例

- 必要なパッケージのインストール

1. host（hostコマンドが使えないことを確認）
2. yum install -y bind-utils（host,digコマンドを使えるようにbind-utilsのパッケージをインストール）
3. yum list installed bind-utils または rpm -q bind-utils（bind-utilsパッケージがインストールされたことを確認）
4. host（書式のヘルプが出力されるのでhostコマンドが使用できることを確認）

- ホスト名の設定と確認

1. hostnamectl set-hostname web01.engineer.jp または nmcli general hostname web01.engineer.jp （ホスト名web01.engineer.jpを設定）
2. hostname または cat /etc/hostname （ホスト名が設定できたことを確認）

- ホストの名前解決の設定と確認

1. echo "192.168.255.200 `hostname`" >> /etc/hosts（自ホストのIPアドレスとホスト名の対応の記述を/etc/hostsに追記）
2. echo "192.168.255.6 host01.engineer.jp" >> /etc/hosts（他ホストのIPアドレスとホスト名の対応の記述を/etc/hostsに追記）
3. cat /etc/hosts（正しく追記されたことを確認）

- IPアドレスの確認、設定、設定の確認

1. ip address show dev enp0s3 【省略形】ip a s enp0s3（デバイスenp0s3のIPアドレスを表示）
2. nmcli connection modify enp0s3 ipv4.method manual ipv4.addresses 192.168.255.200/24 ipv4.gateway 192.168.255.1 ipv4.dns 192.168.255.1（enp0s3のIPアドレスを192.168.255.200/24、enp0s3のゲートウェイとdnsを192.168.255.1に設定）
3. nmcli connection up enp0s3 （設定を反映させるためにコネクションをup、IPが変更されたのでTeraterm接続中であれば切断される）
4. ip a s enp0s3 （IPアドレスの変更を確認）
5. ip route （デフォルトゲートウェイが設定されていることを確認）
6. cat /etc/resolv.conf（DNSのIPアドレスを確認）

ネットワークの設定と確認の実例

- 名前解決と疎通（宛先と通信できること）の確認

1. ping web01.engineer.jp （自ホスト宛にホスト名でpingを打って/etc/hostsでの名前解決とping応答を確認）
2. ping host01.engineer.jp （他ホスト宛にホスト名でpingを打って/etc/hostsでの名前解決とping応答を確認）
3. ping 192.168.255.1 （デフォルトゲートウェイ宛にpingを打って応答を確認）
4. host linuc.org（/etc/resolv.confに記載のあるDNSに問い合わせて、linuc.orgの名前解決ができることを確認）
5. ping linuc.org （インターネット上のlinuc.org宛にpingを打ってDNSによる名前解決と応答を確認）

- ルーティングの設定、設定の確認

1. nmcli connection modify enp0s3 +ipv4.routes “192.168.2.0/24 192.168.255.1” （ルーティングの追加）
2. nmcli connection up enp0s3 （設定を反映させるためにコネクションをup）
3. ip route （ルーティングテーブルの確認）
4. ping 192.168.2.1 （追加先ルーティングへの通信の確認）

※補足
IPアドレスを元のDHCP設定に戻す場合は、下記コマンドを実行。設定した静的IPアドレス、ゲートウェイ、DNSを空文字(””)で削除。
nmcli connection modify enp0s3 ipv4.method auto ipv4.addresses ”” ipv4.gateway ”” ipv4.dns ””

【番外】検証環境での構築手順書作成

●検証環境では、本番環境での構築のために下記のような構築手順書を作成します。

	作業名：Zabbixサーバー構築手順書	作業日：	2021年5月30日		
		作業者：	作業者の名前		
		確認者：	確認者の名前		

項番	作業内容	想定結果	チェック	時刻
1	phpのインストール　yumの準備（リポジトリの追加）			
2	yum -y install https://rpms.remirepo.net/enterprise/remi-release-7.rpm	インストール: remi-release.noarch 0:7.7-1.el7.remi と表示されること。	□	
3	phpのインストール			
4	yum -y install --enablerepo=remi --enabler=remi-php73 php php-cli php-pdo php-common php-pgsql php-bcmath php-xml php-gd php-mbstring php-ldap	インストール: php-bcmath.x86_64 0:7.3.13-1.el7.remi php-ldap.x86_64 0:7.3.13-1.el7.remi php-mbstring.x86_64 0:7.3.13-1.el7.remi php-pgsql.x86_64 0:7.3.13-1.el7.remi と表示されること。	□	
5	phpのインストール確認			
6	rpm -qa | grep php | sort	php-7.3.14-1.el7.remi.x86_64 php-bcmath-7.3.14-1.el7.remi.x86_64 php-cli-7.3.14-1.el7.remi.x86_64 php-common-7.3.14-1.el7.remi.x86_64 php-gd-7.3.14-1.el7.remi.x86_64 php-json-7.3.14-1.el7.remi.x86_64 php-ldap-7.3.14-1.el7.remi.x86_64 php-mbstring-7.3.14-1.el7.remi.x86_64 php-pdo-7.3.14-1.el7.remi.x86_64 php-pgsql-7.3.14-1.el7.remi.x86_64 php-xml-7.3.14-1.el7.remi.x86_64 と表示されること。	□	
7	yumリポジトリの登録と確認			
8	yum -y install https://repo.zabbix.com/zabbix/4.4/rhel/7/x86_64/zabbix-release-4.4-1.el7.noarch.rpm	インストール: zabbix-release.noarch 0:4.4-1.el7 と表示されること。	□	

【番外】検証環境での構築手順書作成

- 手順書には設定コマンド実行後、設定の正常性を確認する手順も記載します。
- 例：abc.txtのパーミッションを所有者だけが読み、書き、実行できるようにしたいとき、設定コマンドと確認コマンドは何でしょうか？

- 設定コマンド

  chmod 700 abc.txt

- 確認コマンド

  ls –l abc.txt

- 手順書の作成は、設定した内容をどのように確認できるだろうか？という思考の訓練となります。

3. 今回解説するLinuxサーバーの設定項目

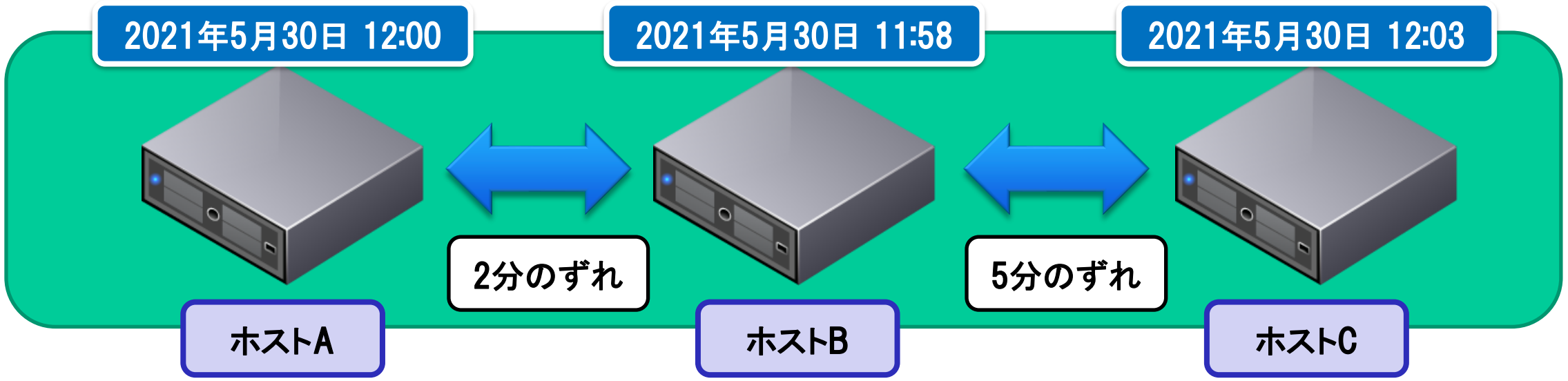
【現場業務の解説】
検証環境におけるサーバー構築

③ システムクロックの同期

副題1.09.1：システム時刻の管理

正確なシステムクロックの重要性

- OSの時刻であるシステムクロックは正確であることが重要です。時刻が正しくないと下記のような問題が生じる可能性があります。
  - スクリプトをcronでスケジューリング実行する際、誤った時刻に実行される。
  - 認証や監視など他サーバーとの連携に問題が出る。
  - システムに障害が発生したときに、ネットワーク内の他ホストとログファイルに記録された時刻に差があり、トラブルの解決が困難になる。

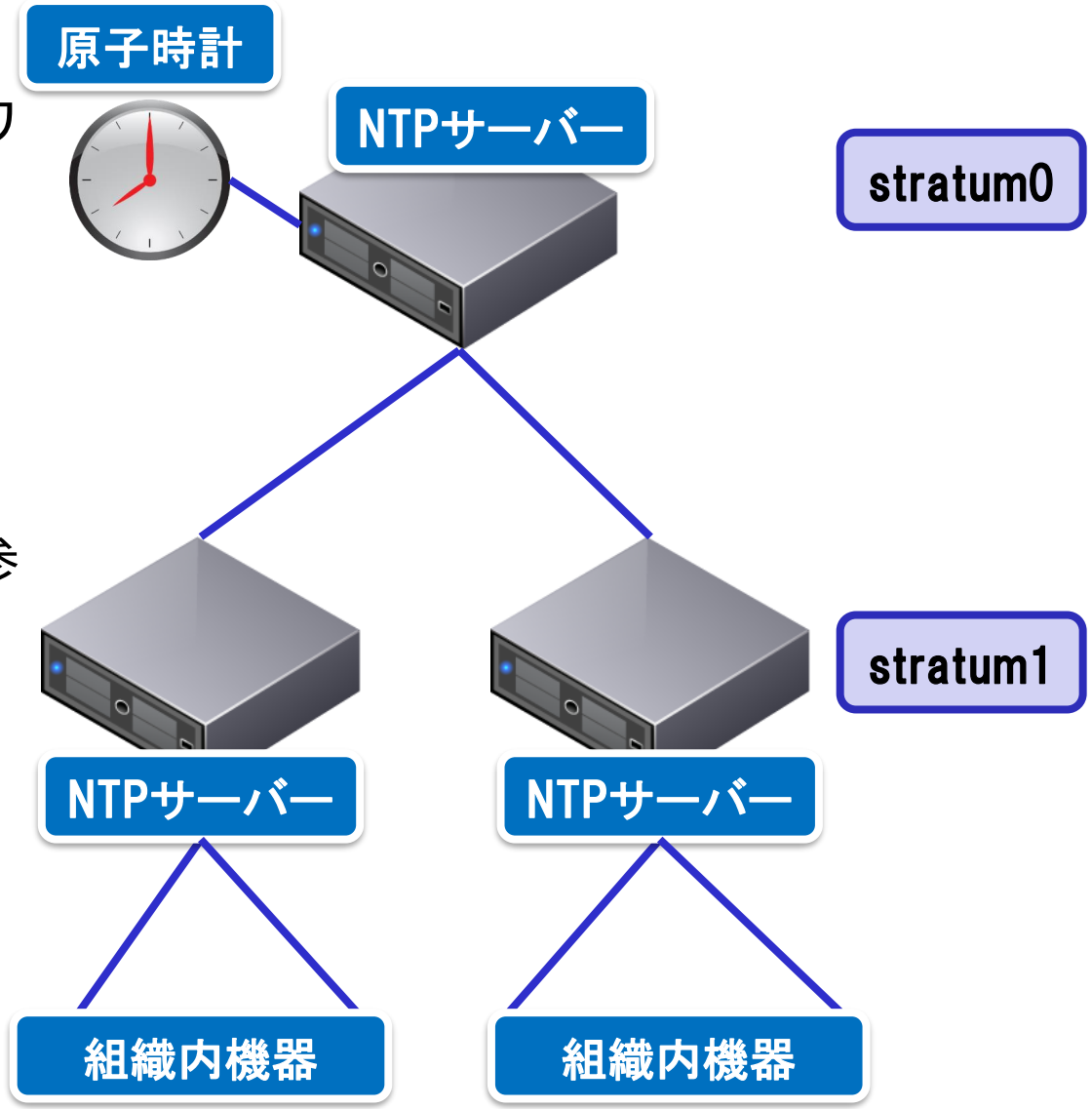


コンピュータにある2つの時刻

- コンピュータにある2つ時刻
  - ハードウェアクロック（ハードウェア時刻）はPCやサーバーのマザーボードに搭載された時計の時刻です。
  - システムクロック(システム時刻)は、LinuxなどOSが管理する時刻です。
  - OSは起動時にハードウェアクロックから時刻情報を取得してメモリ上のシステムクロックに反映します。電源を落とすとメモリからシステムクロックが消去されます。
  - ハードウェアクロックとシステムクロックの精度は高くないため、定期的にシステムクロックを補正して、ハードウェアクロックに反映させる必要があります。

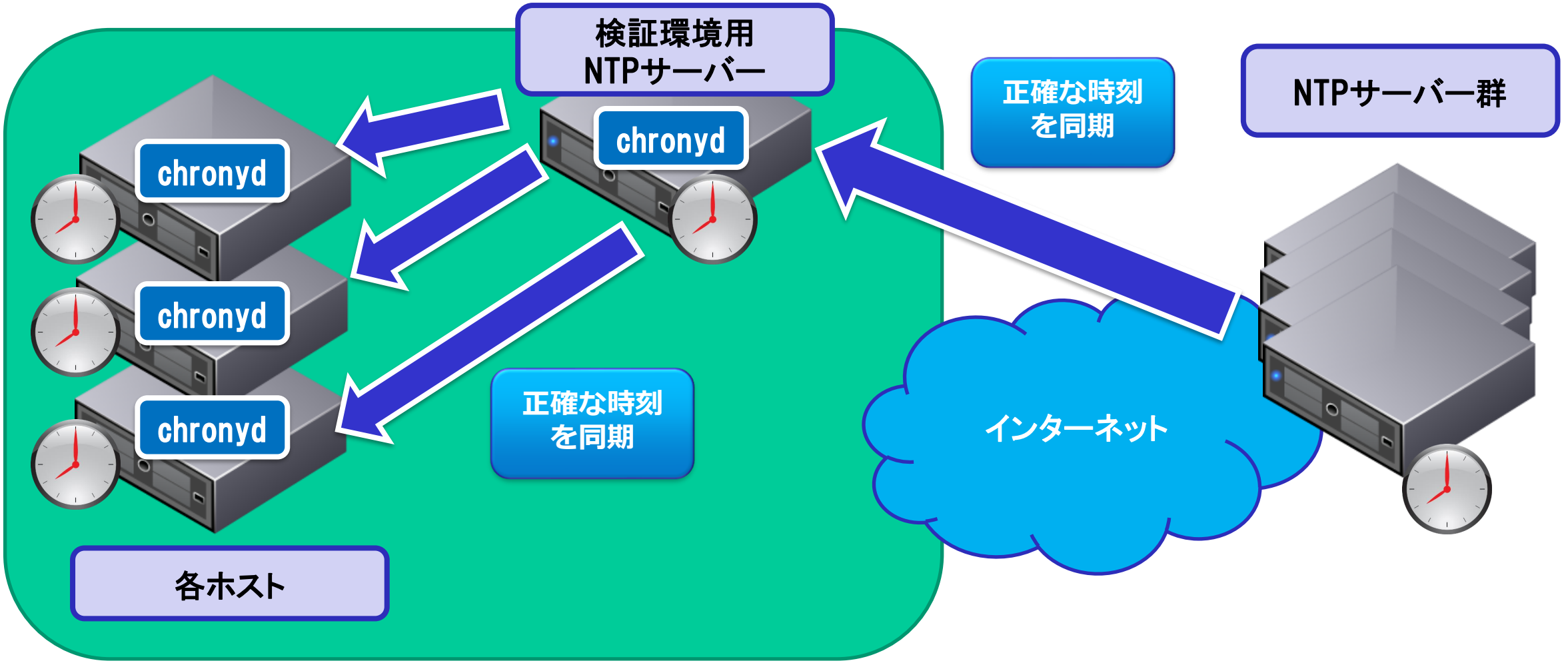
NTPを用いた時刻同期

- NTPを用いた時刻同期
  - NTP (Network Time Protocol)は、ネットワーク経由でコンピュータや各種機器の時刻同期に用いられるプロトコルです。
  - 原子時計やGPSなど正確な時刻ソースに接続されたNTPサーバーと同期します。
  - NTPサーバーは階層構造となっており、時刻ソースのNTPサーバーをstratum0、これを参照するNTPサーバーをstratum1と呼びます。
  - 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する運用が行われます。
  - NTPを使えばシステムクロックの補正作業は不要になります。



検証環境のシステムクロックをNTPで同期

●検証環境内にNTPサーバーを用意して各ホストのシステムクロックと同期します。



chronyによる時刻同期の実例

- CentOS7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。

1. systemctl status chronyd（chronydの起動を確認）
2. systemctl stop chronyd（chronydを停止）
3. systemctl status chronyd （chronydの停止を確認）
4. date（現在のシステムクロックを確認）
5. date 05291130（システムクロックを5月29日11時30分に設定）
6. systemctl start chronyd （chronydを起動）
7. systemctl status chronyd （chronydを起動を確認）
8. 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）
9. chronyc activity（NTPサーバーと同期している状況を確認）
10. chronyc sources（NTPサーバーの同期元ソースを確認）

【現場業務の解説】
検証環境におけるサーバー構築

④ SSHでのリモートログインと公開鍵認証

副題1.10.3：暗号化によるデータの保護

SSHでのリモートログインと公開鍵認証

- 検証環境が遠隔にある場合、インターネットを経由してサーバーにリモートログインできると便利です。
- SSHにより接続先までの**通信経路の暗号化**とユーザー認証に**公開鍵認証**を用いることで安全に作業が可能です。

SSHのユーザー認証方式

- SSHサーバーへのログインを許可するユーザー認証の方法として、パスワード認証と公開鍵認証があります。

	パスワード認証	公開鍵認証
認証方法	ユーザーのパスワードを入力することで認証する	公開鍵と対となる秘密鍵がペアであることを確認することで認証する
設定方法	passwdコマンド	ログインするユーザーの~/.ssh/authorized_keysに公開鍵を設置
利用シーン	LANなど閉じられたネットワークのサーバーにログインする場合	インターネットに接続されたサーバーにログインする場合

公開鍵認証の準備

- 公開鍵認証は公開鍵と秘密鍵がペアであることを証明することでユーザー認証をします。下記①～③は公開鍵を設置する流れです。

公開鍵認証方式でのユーザー認証の流れ

●公開鍵認証は下記のような流れでユーザー認証を行います。

SSHクライアント

①クライアントの秘密鍵でデータを暗号化する

②暗号化したデータをサーバーに送付

SSHサーバー

③暗号化したデータが設置した公開鍵で復号できることを確認
→OKならログインを許可

33

公開鍵認証の実例

- SSHサーバー（web01.engineer.jp）でログイン先ユーザーの作成

1. systemctl status sshd（SSHサーバーの起動を確認）
2. useradd usera（ログイン先のユーザーを作成）
3. passwd usera（パスワードを設定）

公開鍵認証の実例

●SSHクライアント(host01.engineer.jp)の設定

● ユーザー作成とパスワード認証の確認

1. useradd usera（ログイン元のユーザーを作成）
2. passwd usera（パスワードを設定）
3. su - usera（ログインするユーザーに切替）
4. ssh usera@web01.engineer.jp（パスワード認証でSSHサーバーにログインできることを確認）
5. exit（SSHサーバーからログアウト）

● 公開鍵認証の準備

【①クライアントで公開鍵と秘密鍵のペアを作成】

1. ssh-keygen -t rsa（公開鍵認証のためのキーペアを作成）
2. ls -l .ssh（秘密鍵のid_rsaと公開鍵のid_rsa.pubが存在することを確認）

【②公開鍵をサーバーに送付】

3. scp ~/.ssh/id_rsa.pub usera@web01.engineer.jp:~ （公開鍵のid_rsa.pubをweb01.engineer.jpのuseraのホームディレクトリにscpで転送）

公開鍵認証の実例

●SSHサーバー（web01.engineer.jp）でログイン先ユーザーに公開鍵を設置

```
1. su - usera（ログイン先のユーザーに切り替える）
2. ls -l（SSHクライアントの公開鍵id_rsa.pubが転送されていることを確認）
3. mkdir .ssh（公開鍵を置く.sshディレクトリを作成）
4. chmod 700 .ssh（.sshのパーミッションを所有者だけが読み書き実行できる700に設定）
5. ls -ld .ssh（パーミッションを700に設定できたことを確認）
6. cat id_rsa.pub > .ssh/authorized_keys（id_rsa.pubの中身を.ssh配下のauthorized_keysに出力）
7. chmod 600 .ssh/authorized_keys（authorized_keysのパーミッションを所有者だけが読み書きできる600に設定）
8. ls -l .ssh/authorized_keys（パーミッションを600に設定できたことを確認）
```

●SSHクライアントからSSHサーバーに公開鍵認証でリモートログイン

```
1. ssh usera@web01.engineer.jp（公開鍵認証でログインできることを確認）
```

まとめ

- 以上で、ネットワーク設定、システムクロックの同期設定、SSHでのリモートログインと公開鍵認証の設定が完了しました。
- 実際の検証環境の構築では、さらにサーバーソフトウェアのパッケージインストールやサーバーの設定、動作確認テストを行っていきます。

- 全体像を把握して、コマンドやファイルの目的、役割、使う理由を知っていくと知識が繋がっていきます。
- 結果的に、LinuCの取得と業務の知識を深めることにつながっていきます。