LinuCレベル2 202試験概要解説
サーバー構築のできるLinuxエキスパートへの道

2024/09/29（Sun）

LPI-Japanプラチナスポンサー 株式会社ゼウス・エンタープライズ
鯨井 貴博（LinuCエヴァンジェリスト）

Who are you?

鯨井貴博
LPI-Japan プラチナスポンサー　株式会社ゼウス・エンタープライズ
LinuCエヴァンジェリスト

Linuxとの出会い

大学時代 Unixの存在を知り、日経Linuxを読み始める。
2000年にVine Linux 2.0で一度挫折を経験。
その悔しさを忘れきれず、2007年 他業種からIT業界に転職しLinuxに再チャレンジ。

SE・商用製品サポート・インストラクター・プロジェクト管理などを経験し、現在に至る。
自分自身が学習で苦労した経験から、初心者を含む受講者に分りやすい講義を行うように心がけている。

また、興味の向くIT技術・オープンソースソフトウェアなどについて、
Opensourcetehブログ（https://www.opensourcetech.tokyo/）で執筆中。
実際に自分でやってみる/使ってみる・開発者本人から話を聞いてみることを大切にしています。

Linus Torvaldsさん(Linux開発者)


Igor Sysoevさん(nginx開発者)


Alexei Vladishevさん(Zabbix開発者)

私が進んできた道

2007

サーバ

LinuC

ネットワーク

CISCO CERTIFIED CCNA

全般

IPA

Web

HTML5 Professional Certification

クラウド基盤

Apache CloudStack 技術者認定試験 by LPI-JAPAN

人工知能

JDLA Deep Learning for GENERAL

Kubernetes

CERTIFIED kubernetes ADMINISTRATOR

2024

https://www.cisco.com/c/ja_jp/training-events/training-certifications/certifications/associate/ccna.html

https://linuc.org/

https://www.ipa.go.jp/shiken/index.html

https://www.jdla.org/certificate/general/

https://www.accel-exam.jp/

https://html5exam.jp/

https://training.linuxfoundation.org/ja/certification/certified-kubernetes-administrator-cka/

Open Source Summit Japan 2023

Open Source Summit Japan 2023: ボランティアリーダーの体験記
https://www.opensourcetech.tokyo/entry/20231224/1703429785

Open Source Summit Japan 2024

https://events.linuxfoundation.org/open-source-summit-japan/

ZEUS enterprise

LinuC について

LinuC

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

本日のゴール

- LinuCレベル2(202)の全体像を掴む
- LinuCレベル2(202)の学習方法を理解する
- LinuCレベル2(202)の受験計画を立てる

LinuC Level1試験

ZEUS enterprise

101試験

1.01：Linuxのインストールと仮想マシン・コンテナの利用
- 1.01.1Linuxのインストール、起動、接続、切断と停止
- 1.01.2仮想マシン・コンテナの概念と利用
- 1.01.3ブートプロセスとsystemd
- 1.01.4プロセスの生成、監視、終了
- 1.01.5デスクトップ環境の利用

1.02：ファイル・ディレクトリの操作と管理
- 1.02.1ファイルの所有者とパーミッション
- 1.02.2基本的なファイル管理の実行
- 1.02.3ハードリンクとシンボリックリンク
- 1.02.4ファイルの配置と検索

1.03：GNUとUnixのコマンド
- 1.03.1コマンドラインの操作
- 1.03.2フィルタを使ったテキストストリームの処理
- 1.03.3ストリーム、パイプ、リダイレクトの使用
- 1.03.4正規表現を使用したテキストファイルの検索
- 1.03.5エディタを使った基本的なファイル編集の実行

1.04：リポジトリとパッケージ管理
- 1.04.1apt コマンドによるパッケージ管理
- 1.04.2Debianパッケージ管理
- 1.04.3yumコマンドによるパッケージ管理
- 1.04.4RPMパッケージ管理

1.05：ハードウェア、ディスク、パーティション、ファイルシステム
- 1.05.1ハードウェアの基礎知識と設定
- 1.05.2ハードディスクのレイアウトとパーティション
- 1.05.3ファイルシステムの作成と管理、マウント

102試験

1.06：シェルおよびスクリプト
- 1.06.1シェル環境のカスタマイズ
- 1.06.2シェルスクリプト

1.07：ネットワークの基礎
- 1.07.1インターネットプロトコルの基礎
- 1.07.2基本的なネットワーク構成
- 1.07.3基本的なネットワークの問題解決
- 1.07.4クライアント側のDNS設定

1.08：システム管理
- 1.08.1アカウント管理
- 1.08.2ジョブスケジューリング
- 1.08.3ローカライゼーションと国際化

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メール配送エージェント(MTA)の基本

1.10：セキュリティ
- 1.10.1セキュリティ管理業務の実施
- 1.10.2ホストのセキュリティ設定
- 1.10.3暗号化によるデータの保護
- 1.10.4クラウドセキュリティの基礎

1.11：オープンソースの文化
- 1.11.1オープンソースの概念とライセンス
- 1.11.2オープンソースのコミュニティとエコシステム

https://linuc.org/linuc1/range/101.html
https://linuc.org/linuc1/range/102.html

Linuxの基礎知識や操作方法の取得

LinuC

LinuC Level2試験

201試験

2.01：システムの起動とLinuxカーネル
- 2.01.1 ブートプロセスとGRUB
- 2.01.2 システム起動のカスタマイズ
- 2.01.3 Linux カーネルの構成要素
- 2.01.4 Linuxカーネルのコンパイル
- 2.01.5 カーネル実行時における管理とトラブルシューティング

2.02：ファイルシステムとストレージ管理
- 2.02.1 ファイルシステムの設定とマウント
- 2.02.2 ファイルシステムの管理
- 2.02.3 論理ボリュームマネージャの設定と管理

2.03：ネットワーク構成
- 2.03.1 基本的なネットワーク構成
- 2.03.2 高度なネットワーク構成
- 2.03.3 ネットワークの問題解決

2.04：システムの保守と運用管理
- 2.04.1 makeによるソースコードからのビルドとインストール
- 2.04.2 バックアップとリストア
- 2.04.3 ユーザへの通知
- 2.04.4 リソース使用状況の把握
- 2.04.5 死活監視、リソース監視、運用監視ツール
- 2.04.6 システム構成ツール

2.05：仮想化サーバー
- 2.05.1 仮想マシンの仕組みとKVM
- 2.05.2 仮想マシンの作成と管理

2.06：コンテナ
- 2.06.1 コンテナの仕組み
- 2.06.2 Dockerコンテナとコンテナイメージの管理

202試験

2.07：ネットワーククライアントの管理
- 2.07.1 DHCPサーバーの設定と管理
- 2.07.2 PAM認証
- 2.07.3 LDAPクライアントの利用方法
- 2.07.4 OpenLDAPサーバーの設定

2.08：ドメインネームサーバー
- 2.08.1 BINDの設定と管理
- 2.08.2 ゾーン情報の管理
- 2.08.3 セキュアなDNSサーバーの実現

2.09：HTTPサーバーとプロキシサーバー
- 2.09.1 Apache HTTPサーバーの設定と管理
- 2.09.2 OpenSSLとHTTPSの設定
- 2.09.3 nginxの設定と管理
- 2.09.4 Squidの設定と管理

2.10：電子メールサービス
- 2.10.1 Postfixの設定と管理
- 2.10.2 Dovecotの設定と管理

2.11：ファイル共有サービス
- 2.11.1 Sambaの設定と管理
- 2.11.2 NFSサーバーの設定と管理

2.12：システムのセキュリティ
- 2.12.1 iptables や firewalld によるパケットフィルタリング
- 2.12.2 OpenSSH サーバーの設定と管理
- 2.12.3 OpenVPNの設定と管理
- 2.12.4 セキュリティ業務

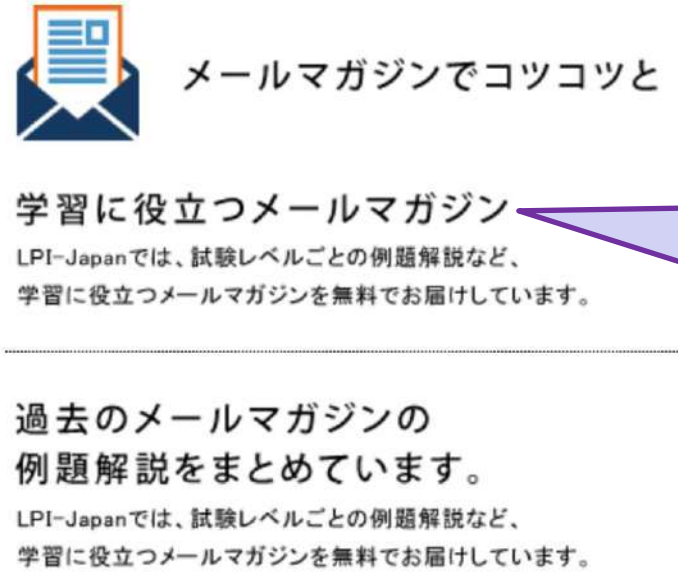
2.13：システムアーキテクチャ
- 2.13.1 高可用システムの実現方式
- 2.13.2 キャパシティプランニングとスケーラビリティの確保
- 2.13.3 クラウドサービス上のシステム構成
- 2.13.4 典型的なシステムアーキテクチャ

サーバ構築が出来る！

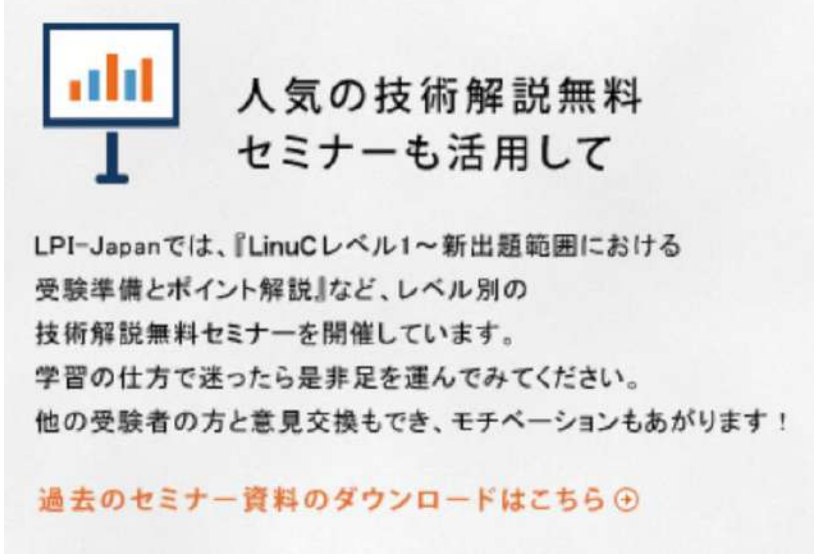
https://linuc.org/linuc2/range/201.html
https://linuc.org/linuc2/range/202.html

LinuC 学習のコツ

①出題範囲の内容について調べてみる
　公式ドキュメント・技術書など
②実際に操作してみる
　これが大事！
③学習の補助教材などを利用する
・メールマガジン
・標準教科書
・過去のセミナー資料
　詳細は、https://lpi.or.jp/learning/



LPI-Japan LinuC通信「レベル2・レベル3を受けてみよう！」でサンプル問題作ってるので、よかったら登録してください！



LinuC 学習のコツ

④過去セミナーの動画

https://www.youtube.com/user/LPIJapan

LinuC 学習のコツ

学習の具体的な進め方(2~3か月程度)

学習開始(0%)

試験範囲の確認(LinuC HP)

●月頃、受験しようという目標を立てる

↓

LinuC認定教材の購入・1週目読込 https://lpi.or.jp/linuc1/book.shtml https://lpi.or.jp/linuc2/book.shtml

↓

LinuC認定教材・Webサイトを参考に、実機操作(サーバ構築やコマンド操作)を試す

※操作やトラブルシュートで力が身に付く！

1ヶ月

↓

LinuC認定教材 2週目読込

●月●日頃、受験しようと決める

↓

問題集やメルマガサンプル問題で理解力確認 https://linuc.org/study/samples/

※理解不足箇所の洗い出し

↓

LinuC認定教材 3週目

※弱点の補強

2ヶ月

↓

仕上げ(80%)

受験申込

受験日の変更も可能なので安心

↓

問題を8~9割以上、正解となるまで繰り返し解く

苦手な部分を重点的に復習

↓

2.5ヶ月

受験

完全に理解した！(100%)

・受験まで継続して学習すること
・繰り返し学習し、理解度/問題正解率を高めた状態で受験すること

202試験の全体像

ZEUS enterprise

2.07：ネットワーククライアントの管理
- 2.07.1 DHCPサーバーの設定と管理
- 2.07.2 PAM認証
- 2.07.3 LDAPクライアントの利用方法
- 2.07.4 OpenLDAPサーバーの設定

2.08：ドメインネームサーバー
- 2.08.1 BINDの設定と管理
- 2.08.2 ゾーン情報の管理
- 2.08.3 セキュアなDNSサーバーの実現

2.09：HTTPサーバーとプロキシサーバー
- 2.09.1 Apache HTTPサーバーの設定と管理
- 2.09.2 OpenSSLとHTTPSの設定
- 2.09.3 nginxの設定と管理
- 2.09.4 Squidの設定と管理

2.10：電子メールサービス
- 2.10.1 Postfixの設定と管理
- 2.10.2 Dovecotの設定と管理

2.11：ファイル共有サービス
- 2.11.1 Sambaの設定と管理
- 2.11.2 NFSサーバーの設定と管理

2.12：システムのセキュリティ
- 2.12.1 iptables や firewalld によるパケットフィルタリング
- 2.12.2 OpenSSH サーバーの設定と管理
- 2.12.3 OpenVPNの設定と管理
- 2.12.4 セキュリティ業務

2.13：システムアーキテクチャ
- 2.13.1 高可用システムの実現方式
- 2.13.2 キャパシティプランニングとスケーラビリティの確保
- 2.13.3 クラウドサービス上のシステム構成
- 2.13.4 典型的なシステムアーキテクチャ

https://linuc.org/linuc2/range/202.html

- ✓DNSサーバ(BIND)
- ✓Webサーバ(Apache・nginx)
- ✓Proxyサーバ(squid)
- ✓メールサーバ(Postfix・Dovecot)
- ✓ファイルサーバ(Samba・NFS)
- ✓セキュリティ(LinuxのFW機能・SSH・VPN)

サーバ構築の基礎♪

LC LinuC

資格を取得する、しない！？(取得するメリット)

ZEUS enterprise

資格(知識)取得 × スキル(構築)保有、両方がある方がいい！

立場の違いはあれど、技術の前では平等！

学習はいつから始めても遅いということはない！
　※ITの分野は幅広く、かつそれぞれが深い

組織にいい流れをもたらす！

コミニティに入るきっかけとなる！

　→好循環な成長のループ♪

LC LinuC

2.07：ネットワーククライアントの管理

ZEUS enterprise

2.07.1 DHCPサーバーの設定と管理
2.07.2 PAM認証
2.07.3 LDAPクライアントの利用方法
2.07.4 OpenLDAPサーバーの設定

LinuC

2.07：ネットワーククライアントの管理

2.07.1 DHCPサーバーの設定と管理

重要度	2
概要	DHCPサーバーを設定できる。これには、デフォルトおよびクライアントごとのオプションの設定と、静的ホストおよびBOOTPホストの追加も含まれる。また、DHCPリレーエージェントの設定とDHCPサーバーの保守も含まれる。
詳細	• DHCPの設定ファイル、用語、ユーティリティ ◦ arp, dhcpd, dhcpd.conf, dhcpd.leases ◦ syslog や systemd のジャーナル内の DHCP のログメッセージ • サブネットと動的割り当て範囲の設定 • DHCPv6 と IPv6 のルータ広告について知っている。 ◦ radvd, radvd.conf

2.07.2 PAM認証

重要度	3
概要	さまざまな方法で認証をサポートするようにPAMを設定できる。これには基本的な SSSD(System Security Services Daemon) の機能を含む。
詳細	• PAMの設定ファイル、用語、ユーティリティ ◦ /etc/pam.d/, pam.conf, nsswitch.conf, sssd.conf ◦ pam_unix, pam_cracklib, pam_limits, pam_listfile, pam_sss

2.07.3 LDAPクライアントの利用方法

重要度	2
概要	LDAPサーバーの照会と更新ができる。また、アイテムの追加およびインポートと、ユーザの追加および管理も含まれる。
詳細	• データ管理のLDAPユーティリティ ◦ ldapadd, ldapdelete, ldapmodify • LDAPディレクトリを照会する。 ◦ ldapsearch • ユーザのパスワードを変更する。 ◦ ldappasswd

2.07.4 OpenLDAPサーバーの設定

重要度	2
概要	LDIF形式および重要なアクセス制御に関する知識も含め、基本的なOpenLDAPサーバーを設定する。
詳細	• OpenLDAP ◦ slapadd, slapcat, slapindex, slapd, /var/lib/ldap/ • ディレクトリベースの設定 ◦ slapd-config • アクセス管理 ◦ slapd.access • 識別名 (DN) • LDIF • ディレクトリ • エントリの操作 • スキーマ ◦ オブジェクト ID、属性、クラス • ホワイトページ

2.07：ネットワーククライアントの管理

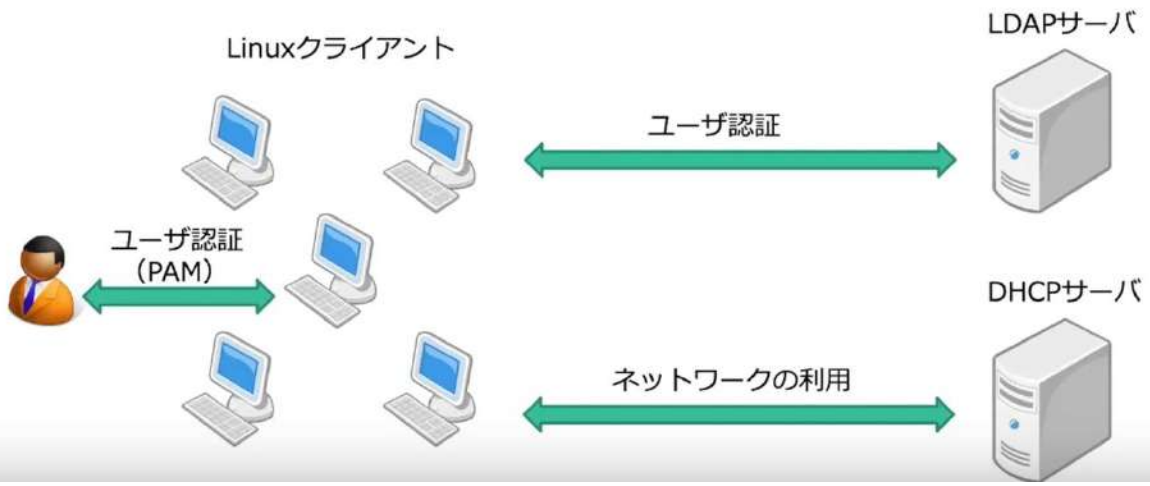
https://www.youtube.com/watch?v=uoyVBva6K1c&feature=youtu.be

2.07：ネットワーククライアントの管理

Linuxによるネットワーククライアント管理とは

■LinuxによるLAN内のクライアント管理

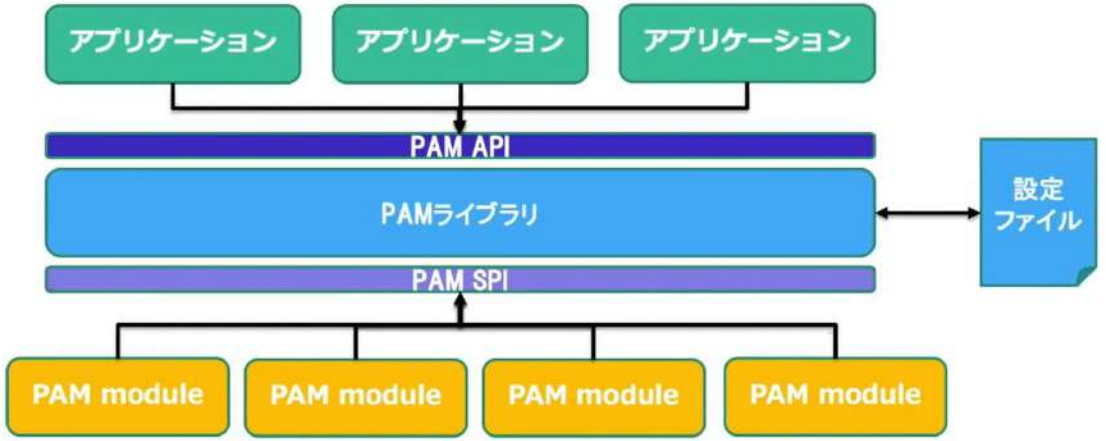
主題2.07におけるPAM、LDAP、DHCPの各テーマの関係性



PAM認証

■PAM認証とは

Linuxのシステムアプリケーションに対して認証、セキュリテイサービスを提供するフレームワーク



PAMサービスモジュール

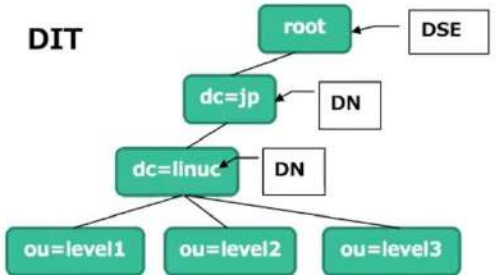
LDAP

■LDAPとは

ディレクトリサービスとは、分散したネットワーク上の各種リソース（ユーザ、サーバ、アプリケーション、プリンタなど）を論理的な名前で管理しやすく系統立ててエンドユーザや管理者に提供する、情報データベースシステム。
→ LDAPはディレクトリサービスのプロトコル。

・LDAPの用語

エントリ：データオブジェクト
DIT(Directory Information Tree)：エントリを階層管理するための管理構造
DSE(Directory Service Entry)：ルートのエントリ
DN(Distinguished Name)：エントリの識別子
RDN(Relative Distinguished Name)：相対識別子



DHCP

■DHCPサーバ（静的割り当て設定）

DHCPサーバとは動的にクライアントに対してネットワーク設定（IP）を割り当てをする。
単に割り当てるだけではなく、ある特定の範囲でクライアントの制限を行うことが可能。

```
subnet 192.168.0.0 netmask 255.255.255.0 {
        option routers                  192.168.0.1;
        option subnet-mask              255.255.255.0;
        ・・・（省略）
        range 192.168.0.10 192.168.0.100;

        host linuc1 {
                hardware ethernet 00:90:96:0f:dc:3a;
                fixed-address 192.168.0.3;
        }
}
```

静的割り当て設定

登録されたMACアドレスのみ割り当てを行うようにすれば、管理外のPCのネットワーク接続を制限することも可能。

2.08：ドメインネームサーバー

2.08.1 BINDの設定と管理
2.08.2 ゾーン情報の管理
2.08.3 セキュアなDNSサーバーの実現

2.08：ドメインネームサーバー

2.08.1 BINDの設定と管理

重要度	3
概要	権威サーバー、再帰サーバー、キャッシュ専用DNSサーバーとして機能するようにBINDを設定できる。これには、稼働中のサーバーを管理すること、ログの設定も含まれる。
詳細	• BIND の設定ファイル、用語、ユーティリティ ◦ named.conf, host, dig, nslookup • BIND の設定ファイルで、BINDゾーンファイルの位置を定義する。 ◦ named.conf • 変更した設定ファイルおよびゾーンファイルの再読込 ◦ rndc, named-checkconf • 代替ネームサーバーとしての dnsmasq, Unbound, NSD, PowerDNS について知っている。

2.08.2 ゾーン情報の管理

重要度	2
概要	正引き、逆引きのゾーンファイルおよびルートヒントファイルを作成できる。これには、レコードに適切な値を設定すること、ホストをゾーンに追加すること、ゾーンをDNSに追加することも含まれる。また、他のDNSサーバーにゾーンの委任を行うことも含まれる。
詳細	• BINDゾーンファイルのレイアウト、内容、ファイル配置 ◦ ゾーンファイルの書式, リソースレコードの書式 • 逆引きゾーンを含む、ゾーンファイルに新しいホストを追加する際の確認方法 ◦ named-compilezone, named-checkzone

2.08.3 セキュアなDNSサーバーの実現

重要度	2
概要	DNSサーバーをroot以外のユーザとしてchroot 環境で実行するよう設定できる。これには、DNSサーバー間で安全なデータ交換を行うことも含まれる。
詳細	• chroot 環境で稼働するようBINDを設定する。 • forwarders文を使用してBINDの構成を分割する。 ◦ named.conf • DNSSEC および基本的なツールについて知っている。 ◦ dnssec-keygen, dnssec-signzone, TSIG(Transaction Signature) • DANE および関連レコードについて知っている。

2.08：ドメインネームサーバー

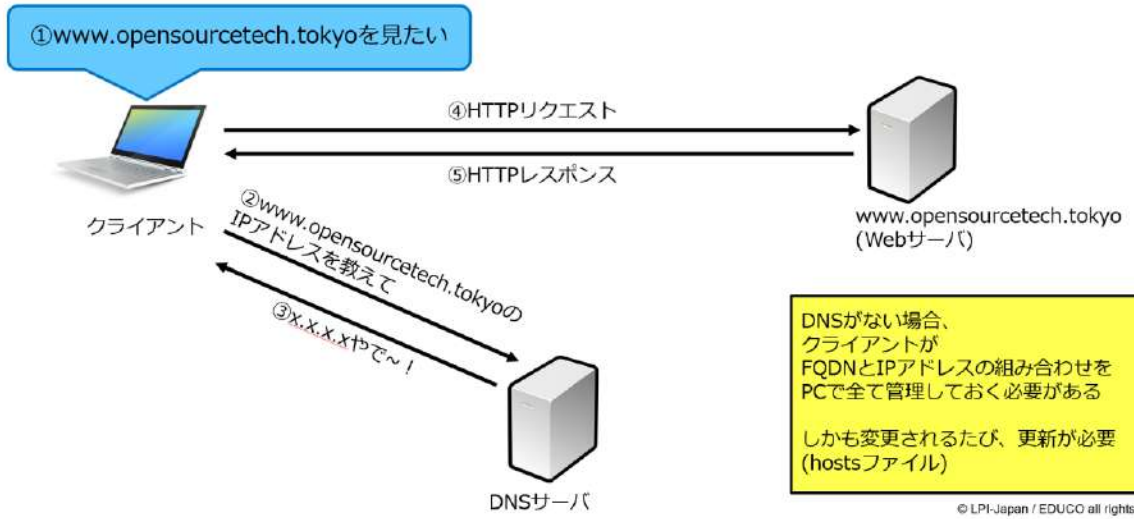
https://www.youtube.com/watch?v=2_CDmJXpKjw

2.08：ドメインネームサーバー

DNSの役割

Domain Name System

www.opensourcetech.tokyo (FQDN、ホスト名) ⇔ x.x.x.x（IPアドレス）の変換を担う



DNSがない場合、
クライアントが
FQDNとIPアドレスの組み合わせを
PCで全て管理しておく必要がある

しかも変更されるたび、更新が必要
(hostsファイル)

DNSの役割

名前解決の種類

正引き

www.opensourcetech.tokyo (FQDN、ホスト名)　x.x.x.x（IPアドレス）の変換を担う

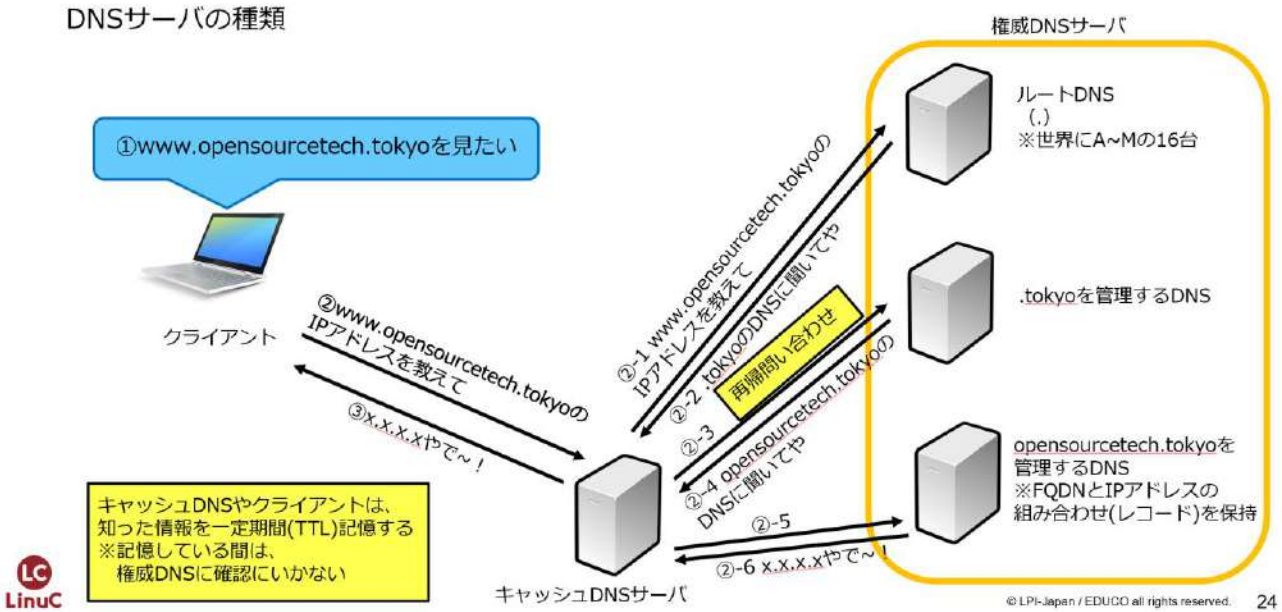
逆引き

※FQDN：*Fully Qualified Domain Name*(完全修飾ドメイン名)

2.08：ドメインネームサーバー

DNSの役割

DNSサーバの種類



DNSの役割

ルートDNSサーバ

図3 各ルートサーバの運用組織と所在地

https://www.nic.ad.jp/ja/newsletter/No45/0800.html

```
ubuntu@linucserver:~$ cat /usr/share/dns/root.hints
.                        3600000      NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA
2001:503:ba3e::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     199.9.14.201
B.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:200::b
;
; FORMERLY C.PSI.NET

(省略)

.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
M.ROOT-SERVERS.NET.      3600000      AAAA  2001:dc3::35
```

2.08：ドメインネームサーバー

BINDの設定

BINDの動作を制御する “named.conf”

レコード情報を保持する “ゾーンファイル” で構成される

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
include "/etc/bind/named.conf.my-zones";

options {
        directory "/var/cache/bind";

        dnssec-validation auto;

        listen-on-v6 { any; };
};

zone "opensourcetech.test" {
        type master;
        file "/etc/bind/test.zone";
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/test.rev";
};
```

test.zoneの内容

```
$ORIGIN opensourcetech.test.
$TTL    604800
@       IN      SOA     dns.opensourcetech.test.
root.opensourcetech.test. (
                        2022051501      ; Serial
                        604800          ; Refresh
                         86400          ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
        IN      NS      dns.opensourcetech.test.
        IN      MX 10   mail.opensourcetech.test.
dns     IN      A       192.168.1.247
www     IN      A       192.168.1.247
mail    IN      A       192.168.1.247
ftp     IN      A       192.168.1.247
smb     IN      A       192.168.1.247
```


BINDの設定

named.conf

```
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";          ・・・・・外部設定ファイルの参照先

options {
        directory "/var/cache/bind";    ・・・・bindの作業ディレクトリ指定
        recursion yes;                  ・・・・再帰問い合わせの実施
        listen-on-v6 { any; };      ・・・・IPv6通信用のIPアドレス指定
};

zone "." {                              ・・・・ルートDNSサーバの情報
        type hint;
        file "/usr/share/dns/root.hints";
};

zone "opensourcetech.test" {    ・・・・ドメイン opensourcetech.test については、権威(master)であり、test.zone にレコード記載あり
        type master;
        file "/etc/bind/test.zone";
};

zone "1.168.192.in-addr.arpa" {    ・・・・ドメイン 192.168.1.0/24については、権威(master)であり、test.rev にレコード記載あり
        type master;
        file "/etc/bind/test.rev";
};
```

設定行は、”;(セミコロン)” を最後に付ける
ディレクティブは、”{“ で始まり、”}“ で閉じる
192.168.1.0/24の逆引きは、” 1.168.192.in-addr.arpa” と書く
など慣れないと記載ミスをしやすい

BINDの設定

ゾーンファイル(正引き)

root@opensourcetech.testを表す

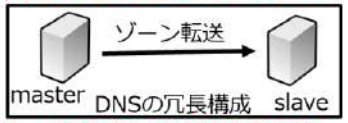
"opensourcetech.test"を意味する

```
$ORIGIN opensourcetech.test.
$TTL    604800      ・・・・問い合わせ時クライアントが取得したレコード情報の有効(キャッシュ)期間 Time To Live
@       IN      SOA     dns.opensourcetech.test. root.opensourcetech.test. (
                        2022051501      ; Serial
                        604800          ; Refresh
                        86400           ; Retry
                        2419200         ; Expire
                        604800 )        ; Negative Cache TTL
;
        IN      NS      dns.opensourcetech.test.
        IN      MX 10   mail.opensourcetech.test.
dns     IN      A       192.168.1.247
www     IN      A       192.168.1.247
mail    IN      A       192.168.1.247
ftp     IN      A       192.168.1.247
smb     IN      A       192.168.1.247
```

無い場合、
自動的に "opensourcetech.test"を補完する

空白は、"@"が補完される

ゾーンファイルのシリアルナンバー
ゾーン情報更新のチェック間隔
ゾーン転送失敗時の待機時間
ゾーンを無効とみなすまでの時間
ネガティブキャッシュ(問い合わせレコードが存在しない)の生存時間



"www" の後に "opensourcetech.test" が補完される

レコードの種類
- SOA：ゾーンの管理情報
- A：IPv4正引き
- AAAA：IPv6正引き
- NS：ネームサーバ
- MX：メールサーバ ※優先度記載必須
- PTR：逆引き
- TXT：テキスト
- CNAME：別名用

パラメータや記載方法・レコードの種類など覚えることが多く、
慣れないと記載ミスをしやすい

Appendix

chroot
"/var/chroot"などのディレクトリを、
ファイルシステムのトップディレクトリである"/"にみせかける手法
これにより攻撃者にBIND(named)を乗っ取られた場合に、
システム全体に影響が及ぶのを防ぐもの
https://linuc.org/study/knowledge/420/

https://linuc.org/study/knowledge/420/
Ctrl キーを押しながらクリックしてリンク先を表示

Appendix

DANE
DNS-Based Authentication of Named Entitiesの略で、
認証情報をDNSを用いて通信するための仕組み
https://www.nic.ad.jp/ja/basics/terms/dane.html

DNSSEC
DNS問い合わせに対する応答が改ざんなどされていないか検証する仕組み
https://www.nic.ad.jp/ja/newsletter/No43/0800.html

TSIG
Transaction SIGnatureの略で、
DNSのメッセージに対して電子署名を行うことで通信経路上における改ざんを防ぐ仕組み
https://jprs.jp/tech/material/rfc/RFC2845-ja.txt

2.09：HTTPサーバーとプロキシサーバー

ZEUS enterprise

2.09.1 Apache HTTPサーバーの設定と管理
2.09.2 OpenSSLとHTTPSの設定
2.09.3 nginxの設定と管理
2.09.4 Squidの設定と管理

LinuC

2.09：HTTPサーバーとプロキシサーバー

2.09.1 Apache HTTPサーバーの設定と管理

重要度	3
概要	Apache HTTP サーバーのインストールと設定ができる。これには、サーバーの負荷と性能の監視、クライアントからのユーザアクセスの制限、モジュールとしてのスクリプト言語をサポートする設定、およびクライアントユーザの認証設定も含まれる。また、サーバーのオプション設定でリソースの使用を制限することも含まれる。仮想ホストを使用するようApache HTTP サーバーを設定し、ファイルへのアクセスをカスタマイズできる。

詳細	• Apache HTTP サーバー の設定ファイル、用語、ユーティリティ
	  ◦ httpd, apache2
	  ◦ httpd.conf, mod_auth_basic, mod_authz_host
	  ◦ apachectl, apache2ctl
	• Apache HTTP サーバーのログファイルの設定と内容
	  ◦ アクセスログとエラーログ
	• アクセス制限の方法とファイル
	  ◦ .htaccess, AuthUserFile, AuthGroupFile
	• クライアントユーザを認証するファイルとユーティリティ
	  ◦ htpasswd
	• 最大リクエスト数、最小/最大サーバー数およびクライアント数の設定
	• Apache HTTP サーバー における仮想ホストの実装
	• ファイルへのアクセスをカスタマイズするために、Apache HTTP サーバーの設定ファイルで Redirect 文を使用する。

2.09.2 OpenSSLとHTTPSの設定

重要度	3
概要	HTTPSを提供するために Apache HTTPサーバーを設定できる。

詳細	• SSL設定ファイル、ツール
	  ◦ /etc/ssl/, /etc/pki/
	  ◦ Apache HTTPサーバーの設定ファイル
	  ◦ SSLEngine, SSLCertificateKeyFile, SSLCertificateFile
	  ◦ SSLProtocol, SSLCipherSuite
	• サーバーの秘密鍵および商用 CA向けのCSR を生成する。
	  ◦ openssl
	• 自己署名証明書を生成する。
	  ◦ openssl
	• 中間 CA を含む鍵および証明書をインストールする。
	  ◦ SSLCACertificateFile, SSLCACertificatePath
	• SSLの使用に関するセキュリティ問題および安全でないプロトコルと cipher を無効にする。

2.09.3 nginxの設定と管理

重要度	3
概要	リバースプロキシサーバーであるnginxのインストールおよび設定ができる。これには、HTTPサーバーとしてのnginx の設定が含まれる。
詳細	• nginx の設定と管理 ◦ /etc/nginx/, nginx • nginx のSSL設定 ◦ ssl, ssl_certificate, ssl_certificate_key, ssl_ciphers, ssl_protocols • リバースプロキシサーバーとしての設定 ◦ proxy_pass, proxy_http_version, proxy_set_header • nginx でリダイレクトを行う

2.09.4 Squidの設定と管理

重要度	2
概要	プロキシサーバーのインストールと設定ができる。これには、アクセスポリシー、認証、リソースの利用方法も含まれる。
詳細	• Squid 3.xの設定ファイル、用語、ユーティリティ ◦ squid.conf, squidclient • アクセス制限の方法 ◦ http_access • クライアントユーザの認証方法 • Squid設定ファイルにおけるACLのレイアウトと内容 ◦ acl

2.09：HTTPサーバーとプロキシサーバー

https://www.youtube.com/watch?v=Xk_PBLoEqMQ

2.09：HTTPサーバーとプロキシサーバー

しかし、業務でサーバ構築をする場合、
考慮すべき点があります。

①利用するソフトウェアが必要とするハードウェア要件

ハードウェア設定の例

以下の表に、様々なハードウェア設定の例を示します：

名前	プラットフォーム	CPU/メモリ	データベース	監視するホスト数
小規模	Ubuntu Linux	PII 350MHz 256MB	SQLite	20
中規模	Ubuntu Linux 64 bit	AMD Athlon 3200 + 2GB	MySQL InnoDB	500
大規模	Ubuntu Linux 64 bit	Intel Dual Core 6400 4GB	RAID10 MySQL InnoDBまたはPostgreSQL	1000以上
非常に大規模	RedHat Enterprise	Intel Xeon 2xCPU 8GB	Fast RAID10 MySQL InnoDBまたはPostgreSQL	10000以上

https://www.zabbix.com/documentation/2.2/jp/manual/installation/requirements

⑤その他

電源管理(冗長)

空調管理

監視/運用管理・障害対応

拠点冗長(オンプレ/クラウド)

アカウント管理・権限管理

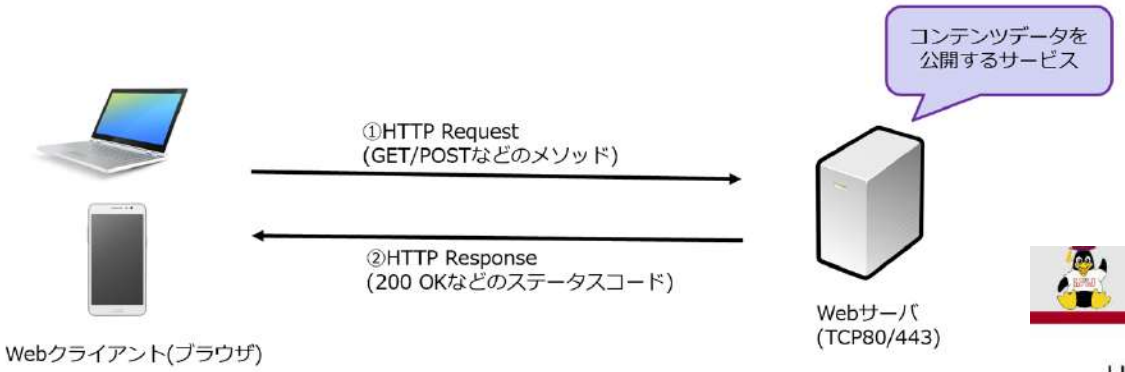
セキュリティ

and more

2.09：HTTPサーバーとプロキシサーバー

技術解説

Web(http/https)の通信



技術解説

HTTPリクエストメソッド

メソッド	意味
GET	データ要求
POST	クライアントからデータ送信
HEAD	HTTPヘッダを要求
PUT	サーバ上へファイルを配置
DELETE	サーバ上のファイルを削除

https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html

2.09：HTTPサーバーとプロキシサーバー

技術解説

HTTPステータスコード

ステータスコード	意味	例
100番台	情報レスポンス	100 Continue など
200番台	成功レスポンス	200 OK など
300番台	リダイレクションメッセージ	301 Moved Permanently 304 Not Modified など
400番台	クライアントエラーレスポンス	401 Unauthorized 403 Forbiden 404 Not Found など
500番台	サーバエラーレスポンス	500 Internal Server Error 502 Bad Gateway 503 Service Unavailable など

https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

構築後のエラー原因を判断するのに重要

技術解説

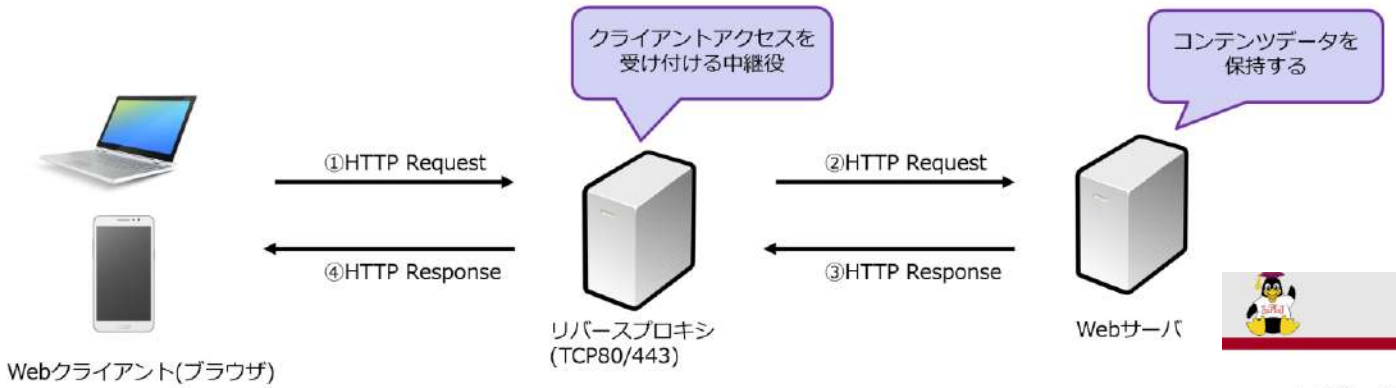
ソフトウェア(nginx/Apache/Squid)のインストールから起動まで

1. インストール(yum・aptなどパッケージ管理・ソースコードなど)
2. 設定ファイル編集
3. コンテンツファイル配置
4. 起動
   ※iptables・firewalld・ufwなどのファイアウォール/SELinuxなどのセキュリティモジュール)は別途実施。

2.09：HTTPサーバーとプロキシサーバー

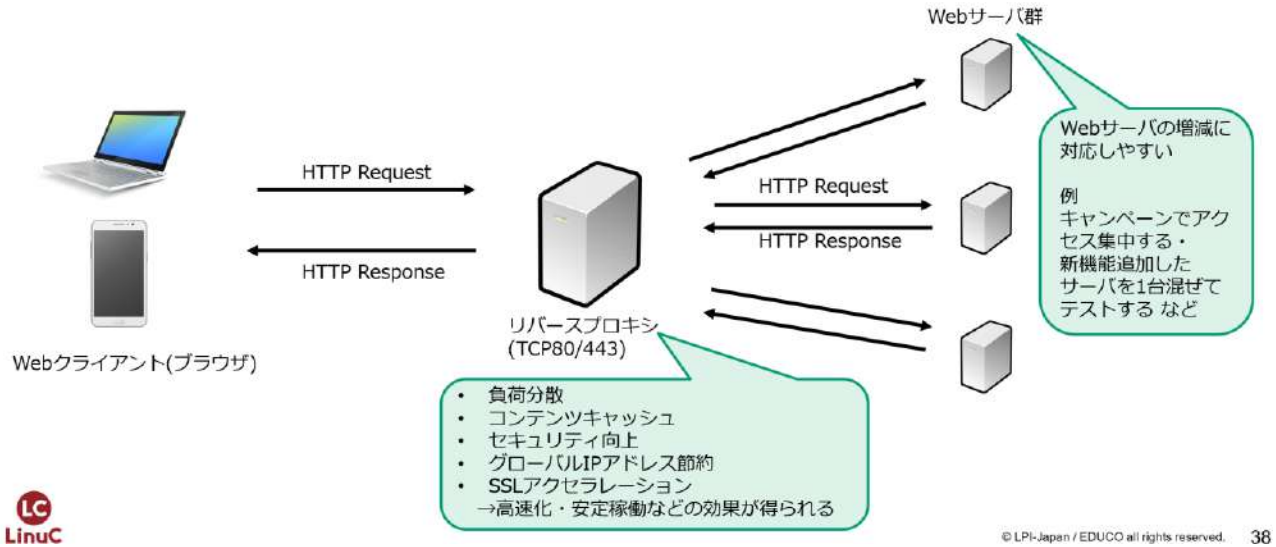
技術解説

リバースプロキシを使ったWeb通信



技術解説

リバースプロキシを使うメリット



2.10：電子メールサービス

ZEUS enterprise

2.10.1 Postfixの設定と管理

2.10.2 Dovecotの設定と管理

LinuC

2.10：電子メールサービス

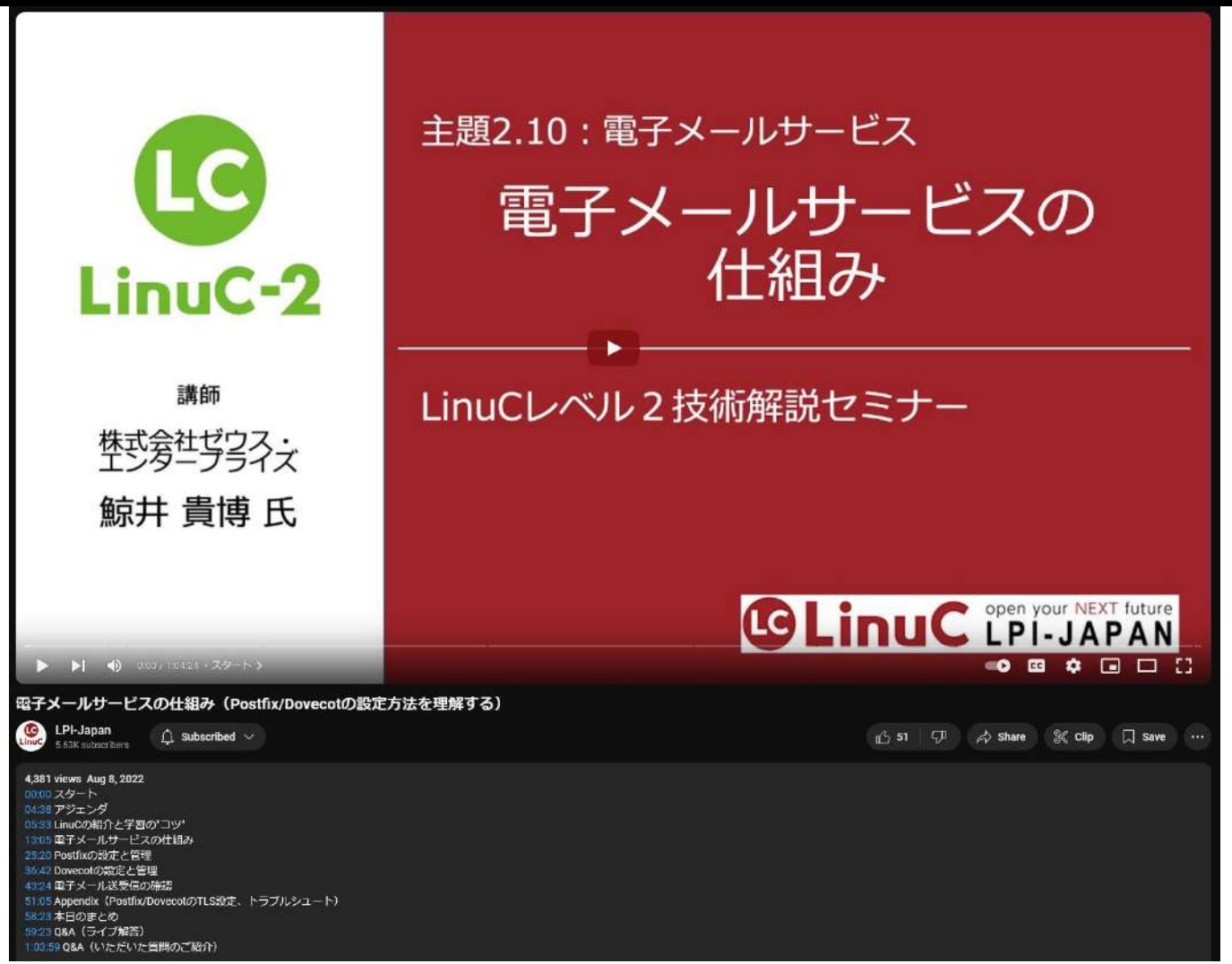
2.10.1 Postfixの設定と管理

重要度	3
概要	電子メールサーバーを管理できる。これには、電子メールのエイリアス、アクセス制限、仮想ドメインの設定も含まれる。また、内部的な電子メールリレーの設定および電子メールサーバーの監視も含まれる。
詳細	• Postfixの設定ファイル、スプール、ログファイル ◦ /etc/postfix/, /etc/aliases, /var/spool/postfix/, /var/log/のメール関連のログ • Postfixの基本的な TLS の設定 • SMTP認証の設定 • SMTPプロトコルに関する基本的な知識 • eximを知っている。

2.10.2 Dovecotの設定と管理

重要度	2
概要	POPおよびIMAPのデーモンのインストールと設定ができる。
詳細	• Dovecot の POP と IMAP の設定と管理 ◦ /etc/dovecot/, dovecot.conf , doveconf, doveadm • Dovecot 向けの基本的な TLS の設定

2.10：電子メールサービス



https://www.youtube.com/watch?v=sJnOHQTsDus

2.10：電子メールサービス

電子メールサービスの仕組み

前提：DNSでドメイン名やメールサーバ情報(MXレコードなど)が管理されていること

xxxx@opensourcetech.tokyo

利用するドメインを取得済みである

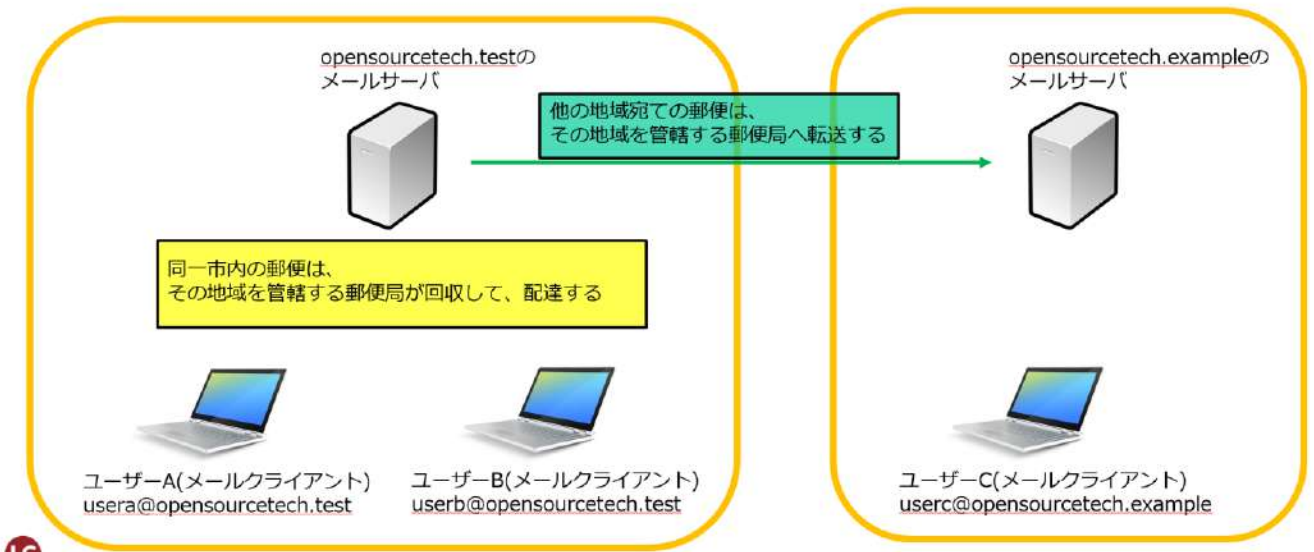
DNSサーバ

DNSサーバでドメインに関するレコード
(MX、Aなど)が管理されていること

なお、DNSサーバについては以下の技術解説セミナーで解説しているのでそちらをご覧ください！
https://linuc.org/study/seminar/3317/

電子メールサービスの仕組み

身近な例で示すと、郵便の仕組みと似ている



2.10：電子メールサービス

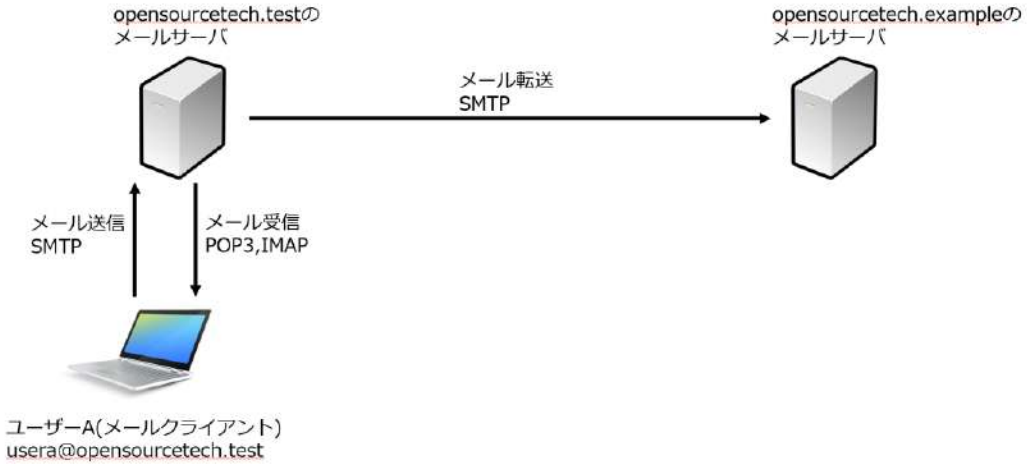
電子メールサービスの仕組み

電子メールサービスで使われるプロトコルとポート番号

プロトコル名	ポート番号	役割
smtp	25	メール送信・転送
smtps	465	メール送信・転送(暗号化)
pop3	110	メール受信
pop3s	995	メール受信(暗号化)
imap	143	メール受信
imaps	993	メール受信(暗号化)
Submission port(SMTP)	587	クライアントからのメール送信(OP25B)

電子メールサービスの仕組み

使用されるプロトコル



 27

Postfixの設定と管理

Postfixの設定(/etc/postfix/main.cf)

```
ubuntu@linucserver:~$ cat /etc/postfix/main.cf
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no
append_dot_mydomain = no
readme_directory = no
compatibility_level = 2
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = mail.opensourcetech.test ・・・・メールサーバの名前
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname ・・・・ドメイン名
mydestination = $myhostname, opensourcetech.test, linucserver, localhost.localdomain, localhost ・・・・管轄するメール宛先
relayhost =
mynetworks = 192.168.1.0/24 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 ・・・・自身の所属するネットワーク情報
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all ・・・・使用するインターフェイス
inet_protocols = all ・・・・使用するプロトコル
home_mailbox = Maildir/ ・・・・メールボックス形式の指定(mbox形式 or Maildir形式)、dovecot側も同じにする必要あり

ubuntu@linucserver:~$ cat /etc/mailname
opensourcetech.test
```


Postfixの設定と管理

mbox形式 vs Maildir形式

main.cf内の “home_mailbox” で設定する

mbox形式
　一つのファイルにユーザのメールデータを格納する

```
ubuntu@linucserver:/var/spool/mail$ ls -l /var/spool/mail
lrwxrwxrwx 1 root root 7 Aug 24  2021 /var/spool/mail -> ../mail

ubuntu@linucserver:/var/spool/mail$ ls -l /var/mail
total 8
-rw------- 1 root   mail    0 Jul 15 12:42 Maildir
-rw------- 1 matt   mail 1656 Jul 30 12:38 matt
-rw------- 1 root   mail    0 Jul 15 12:41 root
-rw------- 1 ubuntu mail 1381 Jul 30 12:38 ubuntu
```

dovecotの設定と管理

dovecotの設定

dovecot.conf + /etc/dovecot/conf.d配下の*.confで構成されている

```
ubuntu@linucserver:~$ ls /etc/dovecot
conf.d  dovecot-dict-auth.conf.ext  dovecot-dict-sql.conf.ext  dovecot-sql.conf.ext  dovecot.conf  private

ubuntu@linucserver:~$ ls /etc/dovecot/conf.d
10-auth.conf       10-tcpwrapper.conf  90-plugin.conf             auth-passwdfile.conf.ext
10-director.conf   15-lda.conf         90-quota.conf              auth-sql.conf.ext
10-logging.conf    15-mailboxes.conf   auth-checkpassword.conf.ext  auth-static.conf.ext
10-mail.conf       20-imap.conf        auth-deny.conf.ext         auth-system.conf.ext
10-master.conf     20-pop3.conf        auth-dict.conf.ext         auth-vpopmail.conf.ext
10-ssl.conf        90-acl.conf         auth-master.conf.ext
```

dovecotの設定と管理

dovecot.confの設定

```
ubuntu@linucserver:~$ cat /etc/dovecot/dovecot.conf
!include_try /usr/share/dovecot/protocols.d/*.protocol
listen = *, ::   ・・・・待ち受けインらーフェイス指定
!include conf.d/*.conf   ・・・・指定ディレクトリ内の外部ファイルも設定として取り込む
!include_try local.conf

ubuntu@linucserver:~$ cat /etc/dovecot/conf.d/10-mail.conf
mail_location = maildir:~/Maildir ・・・・メールボックス形式の指定(mbox形式 or Maildir形式)、postfix側も同じにする必要あり

namespace inbox {
  inbox = yes
}

mail_privileged_group = mail

ubuntu@linucserver:~$ cat /etc/dovecot/conf.d/10-ssl.conf
ssl = no   ・・・・sslの使用
ssl_client_ca_dir = /etc/ssl/certs
ssl_dh = </usr/share/dovecot/dh.pem

ubuntu@linucserver:~$ cat /etc/dovecot/conf.d/10-auth.conf
disable_plaintext_auth = no   ・・・・暗号化されていない認証を許可するかどうか
auth_mechanisms = plain login
!include auth-system.conf.ext
```

2.10：電子メールサービス

電子メール送受信の確認

mailコマンド(メールの作成・送信)

```
ubuntu@linucserver:~$ mail -s mailcommandtest matt@opensourcetech.test   ※メール件名 & Toのアドレス指定
Cc: ubuntu@opensourcetech.test   ・・・・Ccの宛先
Hello!   ・・・・メール本文

ubuntu@linucserver:~$ mail -f ~/Maildir
"/home/ubuntu/Maildir": 2 messages 1 unread
    1 ubuntu                  21/771   テストメールです
>U   2 LinuCserver              14/517   mailcommandtest
? 2
Return-Path: <ubuntu@linucserver>
X-Original-To: ubuntu@opensourcetech.test
Delivered-To: ubuntu@opensourcetech.test
Received: by mail.opensourcetech.test (Postfix, from userid 1000)
        id BC49261123; Sat, 30 Jul 2022 16:13:30 +0000 (UTC)
Subject: mailcommandtest
To: <matt@opensourcetech.test>
Cc: <ubuntu@opensourcetech.test>
X-Mailer: mail (GNU Mailutils 3.7)
Message-Id: <20220730161330.BC49261123@mail.opensourcetech.test>
Date: Sat, 30 Jul 2022 16:13:30 +0000 (UTC)
From: LinuCserver <ubuntu@linucserver>

Hello!
? q
Held 2 messages in /home/ubuntu/Maildir
```

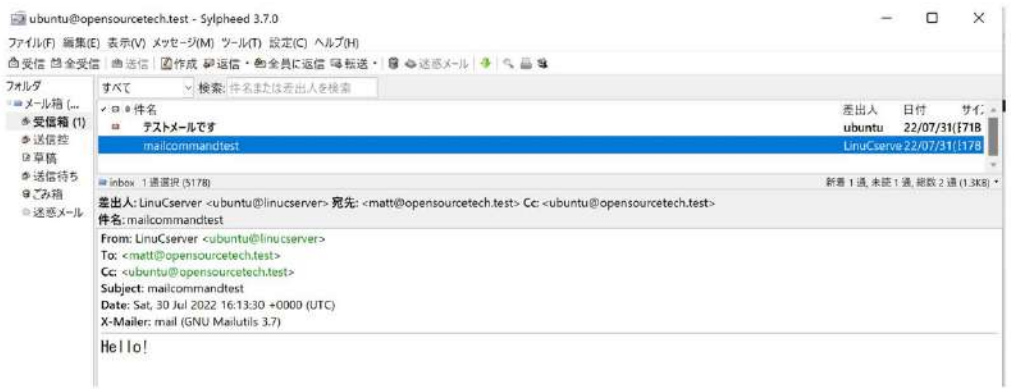
その他、mailコマンドについては以下を参照。
https://www.commandlinux.com/man-page/man1/mail.1.html

電子メール送受信の確認

メーラー(MTU)の使用

- Thunderbird
  https://www.thunderbird.net/ja/
- sylpheed
  https://sylpheed.sraoss.jp/ja/

など



2.11：ファイル共有サービス

ZEUS enterprise

2.11.1 Sambaの設定と管理

2.11.2 NFSサーバーの設定と管理

LinuC

2.11：ファイル共有サービス

2.11.1 Sambaの設定と管理

重要度	4
概要	さまざまなクライアント用にSambaサーバーを設定できる。これには、クライアントがログインするSambaの設定やサーバーが参加するワークグループの設定、共有ディレクトリの定義、インストールにおけるトラブルシューティングも含まれる。

詳細	• Samba の設定ファイルとログファイル
	  ◦ /etc/samba/, /var/log/samba/
	• Samba のユーティリティとデーモン
	  ◦ samba, smbd, nmbd, winbindd
	  ◦ smbcontrol, smbstatus, testparm, smbpasswd, nmblookup, net, smbclient, samba-tool
	• Windowsのユーザ名をLinuxのユーザ名にマッピングする。
	• ACL および AD のセキュリティ
	  ◦ getfacl, setfacl

2.11.2 NFSサーバーの設定と管理

重要度	3
概要	NFSを使用してファイルシステムをエクスポートできる。これには、アクセス制限、クライアントでのNFSファイルシステムのマウント、NFSの保護も含まれる。

詳細	• NFS の設定ファイル
	  ◦ /etc/exports
	• NFSのユーティリティとデーモン
	  ◦ exportfs, showmount, nfsstat, rpcinfo
	  ◦ mountd, portmapper
	• 特定のホストやサブネットへのアクセス制限
	• サーバーとクライアントにおけるマウントオプション
	  ◦ /etc/fstab, /proc/mounts

2.11：ファイル共有サービス

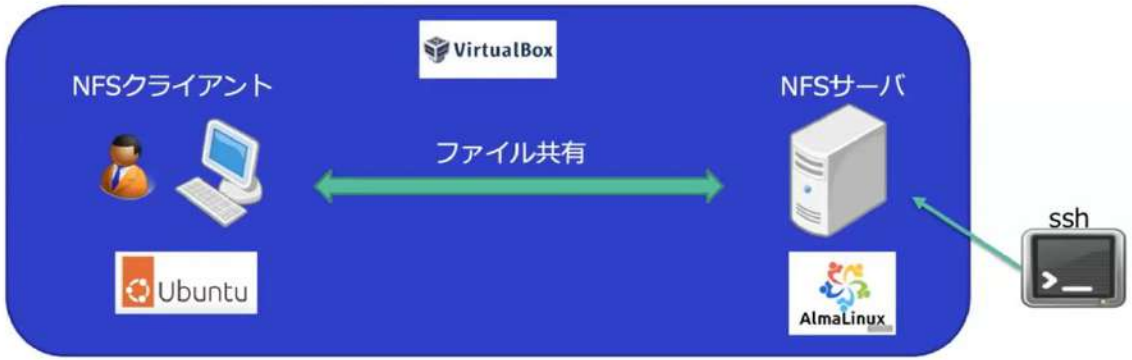
https://www.youtube.com/watch?v=r_eeqrsqY1w

https://www.youtube.com/watch?v=oZftLYDyKLU

今回の実演環境について

■NFSサーバとNFSクライアントを仮想環境で構築

仮想マシンを利用して同一マシン上にサーバ、クライアントのOSをインストールし、
その２つが通信することでNFSの通信環境を構築。



NFSとは

■NFS(Network File System)とは

NFSはUNIX系OS間でファイルシステムの共有を行うためのサービスです。NFSは複数の構成要素によって成り立っており、その複数の要素を最低限理解しておく必要があります。

デーモン名	説明
nfsd	NFSファイルシステム要求を処理するNFSサーバカーネルモジュール
rpcbind (portmapper)	RPCプログラムにポート番号を割り当てる（portmapperの実装）
rpc.nfsd	NFSクライアントのリクエストを処理する
rpc.mountd	NFSクライアントからのマウント要求を処理する
rpc.statd	NFSサーバのステータスモニタ
lockd	NFSクライアントがNFSサーバ側でファイルロックを可能にする

※ NFSはV3とV4により構成要素がことなっており、下位互換性の担保のために複数機能が混在している場合があります。

 8

2.11：ファイル共有サービス

NFSサーバ

■NFSサーバの設定

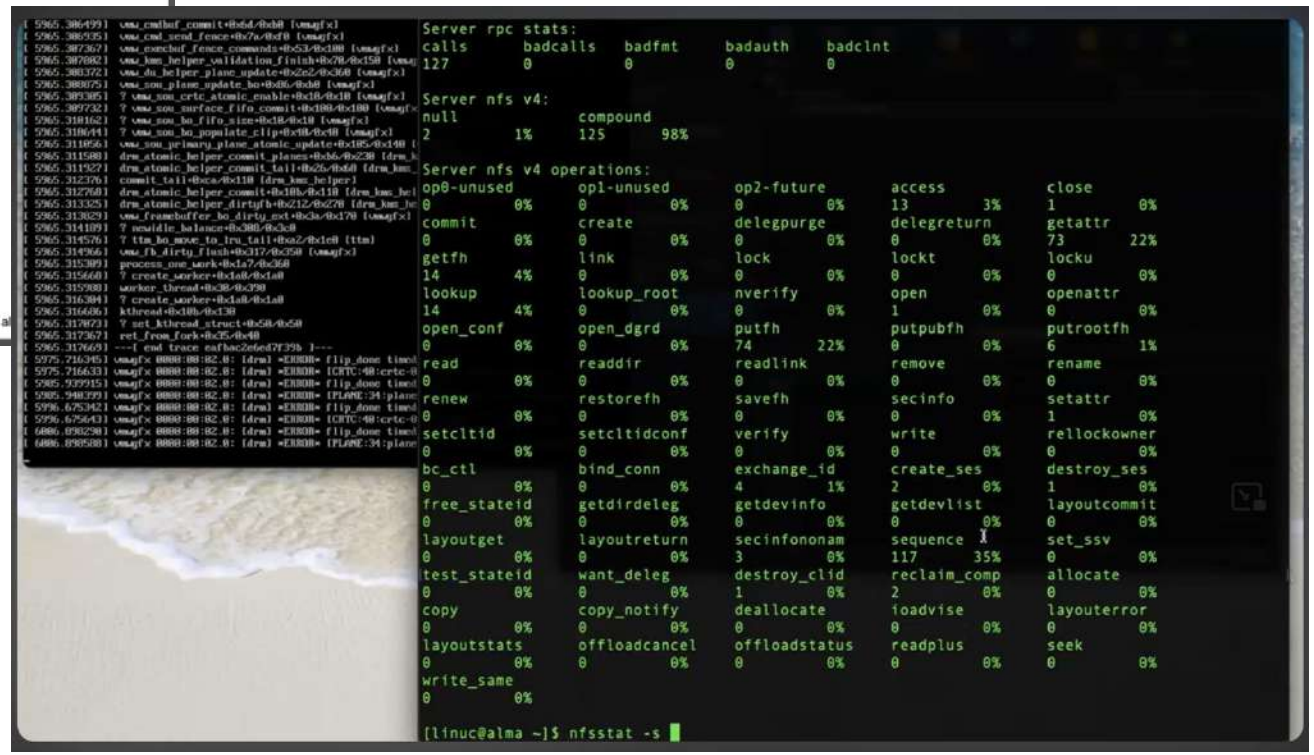
・NFSサーバの設定は/etc/exports ファイルで行う

・書式
[共有したいディレクトリパス] [共有先]([オプション])

・例
/mnt/cdrom 192.168.0.0/255.255.255.0(rw,sysnc)
/home/tmp *.linuc.or.jp(ro) user1.linuc.or.jp(rw)

共有先はIPアドレス、FQDNなどの形で指定も可能ですし、ネットマスクと併せて指定することで一度に複数のクライアントを指定することも可能です。

※共有先とオプション指定の()は続けて記述する必要があります。



2.11：ファイル共有サービス

技術解説

1992年　Andrew Tridgell氏により作成

(https://www.samba.org/~tridge/)

1994年　日本語サポートパッチ追加
1999年　Samba2.0.0リリース　日本Sambaユーザ会設立(http://www.samba.gr.jp/)
2001年　Samba2.2.0リリース Windows NTドメインコントローラ機能追加
2003年　Samba3.0.0リリース Active Directoryクライアント機能追加
.
.　NTFSサポートなど各種機能追加
.
2006年　Active Directoryドメインコントローラ機能の追加がはじまる
2012年　Samba4.0.0リリース　Active Directoryドメインコントローラ機能追加
2021年1月 最新バージョンは Samba4.13.4(https://www.samba.org/samba/history/)

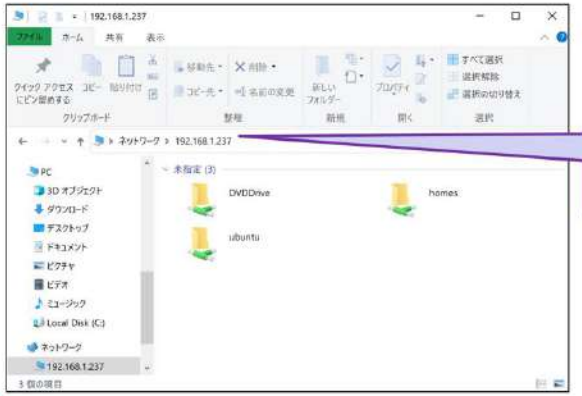
LinuC

技術解説

Sambaとは

“ LinuxなどのUnix系サーバでWindowsサーバ機能を実装させるソフトウェア “

⇒ **周りのWindowsからみると、仲間がいるようにみえる！**



Windows共有へアクセスしているようですが、
その正体はLinux上で動作しているSamba

よくNASなどの製品の内部で使われている！

LinuC

 16

LinuC

2.11：ファイル共有サービス

技術解説

Sambaの機能

- ファイルサーバ機能
  共有フォルダ提供、そのフォルダに関するアクセス制御などWindows共有と同等の機能を提供
- プリントサーバ機能
  Windowsネットワークにおけるネットワークプリンタ機能を提供
- ネットワーク機能
  WindowsネットワークにおけるSambaの表示や名前解決を提供
- ドメインコントローラ機能
  Active DirectoryやNTドメインのドメインコントローラ機能(認証統合)を提供
- ファイル共有クライアント機能
  Windowsクライアントのように振舞い共有へアクセスする機能

技術解説

Sambaを構成するデーモン

- smbd
  ファイル共有やSambaの多くの部分を担当(139/tcp・445/tcp)
- nmbd
  Windowsネットワークにおける名前解決、ブラウジング機能などを担当(137/udp・138/udp)
- winbindd
  NSS(Name Server Switch)機能を担当　※Windowsドメインとのユーザアカウントの統合
- その他(Active Directoryドメイン関連)
  LDAP、Kerberos、DNS、NTPなど

© LPI-Japan / EDUCO all rights reserved.

技術解説

Sambaの設定は、“smb.conf”で行う。

```
ubuntu@ubuntu:~$ cat -n /etc/samba/smb.conf
    22  #======================= Global Settings =======================
    23
    24  [global]
    25
    26  ## Browsing/Identification ###
    27
    28  # Change this to the workgroup/NT-domain name your Samba server will part of
    29     workgroup = WORKGROUP
    30
    31  # server string is the equivalent of the NT Description field
    32     server string = %h server (Samba, Ubuntu)
     .
     .
     .
   175  [homes]
   176  ;   comment = Home Directories
   177  ;   browseable = no
   178     browseable = yes
   179     writeable = yes
   180     readeable = yes
     .
     .
     .
```

セクション

変数

行頭に”#(シャープ)”や”;(セミコロン)”を付けると、コメントとして扱われる

“パラメータ名 = 値”で設定
真偽を設定する場合、
“yes/no”・”true/false”・”1/0”が使える

技術解説

smb.confへのパラメータ設定時の注意

- 別名パラメータ

writable

このパラメーターは右記のパラメーターの別名である：writeable.

write ok

このパラメーターは右記のパラメーターの別名である：writeable.

public

このパラメーターは右記のパラメーターの別名である：guest ok.

guest ok (S)

このパラメーターが yes のサービスに付いては、サービスへの接続の際にパスワードが要求されない。この場合、guest account の権限で操作が行われる。

このパラメーターは restrict anonymous = 2 のメリットを無効にする。

このオプションに関する詳細については、security セクションを参照のこと。

既定値: *guest ok = no*

writable = yes
writeable = yes
write ok = yes　　は同じ意味

public = yes
guest ok = yes　　は同じ意味

- 反意語となるパラメータ

writeable (S)

read only の反意語である。

既定値なし

writeable = yes
read only = no　　は同じ意味

http://www.samba.gr.jp/project/translation/current/htmldocs/manpages/smb.conf.5.html

2.12：システムのセキュリティ

ZEUS enterprise

2.12.1 iptables や firewalld によるパケットフィルタリング
2.12.2 OpenSSH サーバーの設定と管理
2.12.3 OpenVPNの設定と管理
2.12.4 セキュリティ業務

LinuC

2.12：システムのセキュリティ

ZEUS enterprise

2.12.1 iptables や firewalld によるパケットフィルタリング

重要度	3
概要	IPパケットを転送したり、ネットワークアドレス変換（NATやIPマスカレード）を実行するようシステムを設定し、ネットワークを保護することができる。これには、ポートリダイレクトの設定、フィルタルールの管理、攻撃の回避も含まれる。
詳細	• iptables および ip6tables のツール 　◦ iptables, ip6tables • IPパケットの転送 　◦ /proc/sys/net/ipv4/, /proc/sys/net/ipv6/ • ルーティングテーブルを管理するためのツール • ポートリダイレクト • 発信元や宛先のプロトコルやポート、アドレスに基づいて、IP パケットの受入と拒否を行うフィルタおよびルールの表示と保存 　◦ /etc/services • フィルタ設定の保存および再読込 　◦ iptables-save, iptables-restore • firewalld で設定の確認と変更ができる。 　◦ firewalld, firewall-cmd • ufw で設定の確認と変更ができる。 　◦ ufw

2.12.2 OpenSSH サーバーの設定と管理

重要度	4
概要	SSHデーモンの設定と保護ができる。これには、鍵の管理とユーザ用にSSHを設定することも含まれる。
詳細	• OpenSSH サーバーの設定ファイルとデーモン 　◦ sshd, /etc/ssh/sshd_config 　◦ /etc/ssh/ssh_host_*_key および ssh_host_*_key.pub • スーパーユーザおよび一般ユーザのログインを制限する。 　◦ PermitRootLogin, PubKeyAuthentication, AllowUsers, PasswordAuthentication

LC LinuC

2.12.3 OpenVPNの設定と管理

重要度	2
概要	VPN (仮想プライベートネットワーク) の設定および安全なポイントツーポイントまたはサイトツーサイトの接続ができる。
詳細	• OpenVPN の機能概要を理解している。 • OpenVPN の設定ファイルとツール ◦ /etc/openvpn/, openvpn

2.12.4 セキュリティ業務

重要度	3
概要	さまざまな情報源からセキュリティ警告を収集できる。侵入検知システムをインストール、設定、および実行できる。セキュリティパッチやバグ修正を適用できる。
詳細	• サーバーのポートをテストおよびスキャンするユーティリティ ◦ netcat(nc, ncat), nmap, iptables, firewalld • Bugtraq、CERT、CIACやその他のセキュリティ警告を報告する組織と、そのアドレスに関する知識 • IDS（Intrusion Detection System：侵入検知システム）を実装するユーティリティ ◦ fail2ban, snort • OpenVAS や OpenSCAPについて知っている。

2.12：システムのセキュリティ

https://www.youtube.com/watch?v=wVPikaypDas

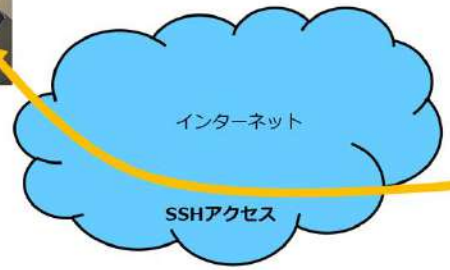
https://www.youtube.com/watch?v=oZftLYDyKLU&t=2184s

SSHとは

- Secure SHellの略
- 何がSecureなのか？ →通信の暗号化
- 以前は、telnet(非暗号化 = 平文でのやりとり)であった

SSHによるクライアントからサーバへのアクセス

- 安全なリモートアクセスを提供
- 秘密鍵/公開鍵のペアによる認証
- サーバへの通信暗号化



OpenSSH環境の用意

■サーバプログラムとして、openssh-serverをインストール

■通信元(クライアント)には、sshコマンドやそれを実装したクライアントソフトウェアを利用する

```
ubuntu@ubuntu: $ dpkg -l | grep ssh
ii  libssh-4:amd64                  0.9.6-2ubuntu0.22.04.3          amd64        tiny C SSH library (OpenSSL flavor)
ii  openssh-client                  1:8.9p1-3ubuntu0.6              amd64        secure shell (SSH) client, for secure access to remote machines
ii  openssh-server                  1:8.9p1-3ubuntu0.6              amd64        secure shell (SSH) server, for secure access from remote machines
ii  openssh-sftp-server             1:8.9p1-3ubuntu0.6              amd64        secure shell (SSH) sftp server module, for SFTP access from remote machi
nes
ii  ssh-import-id                   5.11-0ubuntu1                   all          securely retrieve an SSH p
```

OpenSSH環境の用意

■SSHクライアントソフトウェア

sshコマンド(CLI)
TeraTerm
rlogin
Putty
scpコマンド・winscp ※sshを使ったファイル転送

OpenSSH設定ファイルの概要

■OpenSSHのサーバー設定は、 /etc/ssh/sshd_config ファイルで設定する

```
ubuntu@ubuntu:~$ cat /etc/ssh/sshd_config

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO
```


OpenSSH設定ファイルの概要

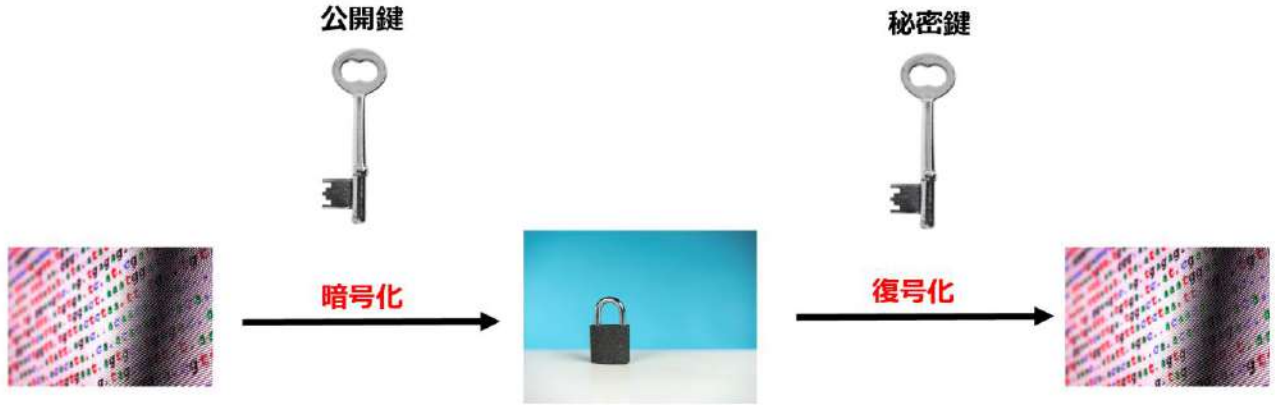
ディレクティブ	設定内容	設定例
Port	OpenSSHが公開するポート番号	22、10022
Protocol	サポートするバージョン	2
ListenAddress	接続を受け付けるアドレス指定	0.0.0.0 ※全てのアドレス
HostKey	秘密鍵の指定	/etc/ssh/ssh_host_ed25519_key
LOGLEVEL	ログの出力レベル指定	INFO・DEBUG・ERROR
PermitRootLogin	rootユーザのログイン可否	yes・no・without-password・forced-commands-only
MaxAuthTries	1接続当たりの認証の試行回数	6
MaxSessions	接続ごとに許可されるセッション数	10
AuthorizedKeysFile	ユーザ認証に使われる公開鍵指定	.ssh/authorized_keys .ssh/authorized_keys2
UsePAM	PAMによる認証の許可	yes・no
Include	外部設定ファイルの参照先	/etc/ssh/sshd_config.d/*.conf
PasswordAuthentication	パスワード認証の許可	yes・no
AllowUsers	許可するユーザを指定	ユーザ名
PubKeyAUthentication	公開鍵認証の許可	yes・no

共通鍵暗号方式とは？

- 共通鍵暗号方式とは、暗号化と復号化に同じ鍵を使用する方式
- 受信者と送信者は、共有の鍵を所有している必要がある
- データの暗号化には、共通鍵アルゴリズムが使用される
- 共通鍵の強度が重要

公開鍵暗号方式とは？

- 暗号化と復号化に異なる鍵を使用する
- 公開鍵は誰でも使用できる一方、秘密鍵は非公開で管理する



2.12：システムのセキュリティ

脆弱性に要注意

緊急度が高い脆弱性が確認された場合には、対応が必要

https://jvndb.jvn.jp/

脆弱性に要注意

公開された脆弱性の内容やその対策をチェック

「OpenSSH」に脆弱性、アップデートがリリース

「OpenSSH」に脆弱性が明らかとなった。特定の条件が重なるとリモートよりコードの実行が可能になるとしており、「OpenSSH」の開発チームは、脆弱性を修正したアップデートを提供している。

過去に修正された「CVE-2016-10009」の修正が不十分だったことに由来する脆弱性「CVE-2023-38408」が明らかとなったもの。Qualysの研究チームが発見、報告した。

SSHエージェント転送を利用している場合に影響があり、「PKCS#11」のサポートによってディストリビューションにおいて共有ライブラリが保存されているディレクトリを読み込むことができることに起因している。

システム内で利用するパッケージに依存するが、複数の特定ライブラリが同ディレクトリ内に存在する場合、リモートより任意のコマンドを実行することが可能としており、同社では、一部パッケージを追加導入したUbuntuの特定バージョンで動作する実証コード（PoC）の作成にも成功したとしている。

同社では、現地時間7月6日にOpenSSHの開発チームへ初期のパッチを提供。その後連携しつつ対応を進めていた。開発チームは、現地時間7月19日に同脆弱性を修正したアップデート「OpenSSH 9.3p2」をリリース。許可リストを指定するなど緩和策についてもアナウンスしている。

（Security NEXT - 2023/07/26 ）

https://www.security-next.com/148143

2.13：システムアーキテクチャ

ZEUS enterprise

2.13.1 高可用システムの実現方式
2.13.2 キャパシティプランニングとスケーラビリティの確保
2.13.3 クラウドサービス上のシステム構成
2.13.4 典型的なシステムアーキテクチャ

LinuC

2.13：システムアーキテクチャ

ZEUS enterprise

2.13.1 高可用システムの実現方式

重要度	2
概要	求められる可用性のレベルを実現するシステム構成を把握している。

詳細	• 可用性に影響のある事象を理解している。
	  ◦ 故障・障害のパターン、メンテナンスによる停止（計画、緊急）など
	  ◦ 物理障害と論理障害
	  ◦ SPoF、回復性(難易度、時間)
	• 可用性の評価方法を知っている。ただし計算式は含まない。
	  ◦ MTBF、MTTR、稼働率、SLA
	  ◦ RPO、RTO
	• 高可用性(HA)を実現するシステム構成を知っている。
	  ◦ 冗長化によるHAの実現
	  ◦ Pacemaker, Corosync
	  ◦ HA構成の種類として クラスタやロードバランシングの概念を知っている。
	• 物理的、地理的な分散による可用性レベルの違いについて知っている。

2.13.2 キャパシティプランニングとスケーラビリティの確保

重要度	2
概要	• 必要なリソース量を事前に予見できるシステムにおいて、近い将来に向けた拡張方法を知っている。 • 将来的に必要なリソース量が容易に予見できないシステムにおいて、現在のリソース使用状況を継続的に把握できる。

詳細	• キャパシティプランを作成するために把握しておくべきシステムリソースの観点と項目
	• リソースを増減させる方法と必要な対応を知っている。
	  ◦ スケールアップ・ダウン
	  ◦ スケールアウト・イン
	• スケールアップの方式を知っている。
	  ◦ 必要リソース量を搭載したマシンの再構成
	• スケールアウトの方式を知っている。
	  ◦ スケールアウトに対応できるアプリケーション構成（ステートレスな構成 - DB、セッションなど）
	  ◦ 構成管理ツールや仮想マシンイメージを使ったノードの増減
	  ◦ アクセスの振り分け - ロードバランサ、DNS ラウンドロビン

LC LinuC

2.13：システムアーキテクチャ

ZEUS enterprise

2.13.3 クラウドサービス上のシステム構成

重要度　2

概要

- クラウドサービス上の IaaS を中心としたシステム構成の特徴を理解している。
- 必要に応じて IaaS リソースの増減が可能であることを理解している。

詳細

- クラウドのストレージの種別を理解している。
  - インスタンス動作中にのみ使用可能なストレージ(エフェメラルストレージ)
  - インスタンス停止/起動をまたいで使用可能なストレージ(永続化ストレージ)
- クラウドのネットワークの種別を理解している。
  - 固定IPアドレス、フローティングIPアドレス
- クラウドのネットワークセキュリティを理解している。
  - テナントネットワーク、ファイアウォール(セキュリティグループ)
- クラウドを支える主要な技術やサービスを理解している。
  - オブジェクトストレージ、メッセージングシステム（キュー）、オートスケーラー

2.13.4 典型的なシステムアーキテクチャ

重要度　3

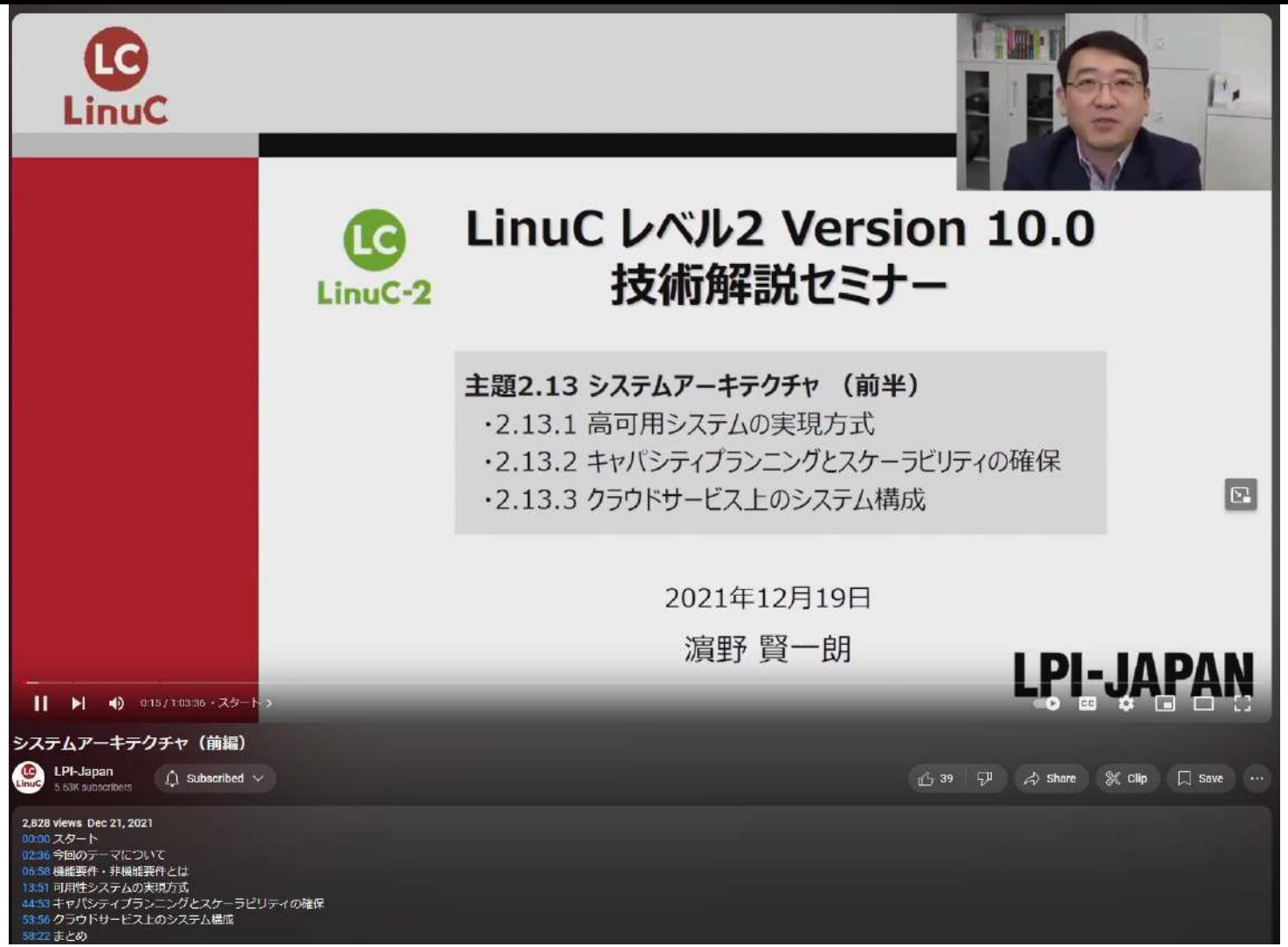
概要　高可用性やスケーラビリティを確保するためのシステム構成のパターンを把握している。

詳細

- 代表的なシステム構成パターンとその特徴を知っている。
  - PHP/Apache HTTP Server+PostgreSQL/MySQLによるLAPP、LAMP構成
  - Webサーバー＋APサーバー＋DBサーバーによるWeb3層モデル
  - ロードバランサ、HA構成、データベースレプリケーションによる冗長性を担保したWeb3層モデル
  - ロードバランサ/DNSラウンドロビン＋WebサーバーのスケールアウトによるスケーラブルなWebシステム
  - プロキシサーバーによるキャッシュやCDNを活用したスケーラブルなWebシステム
  - メッセージングキューを活用した非同期データ処理システム

LinuC

2.13：システムアーキテクチャ



https://www.youtube.com/watch?v=NERo0dYuSSk

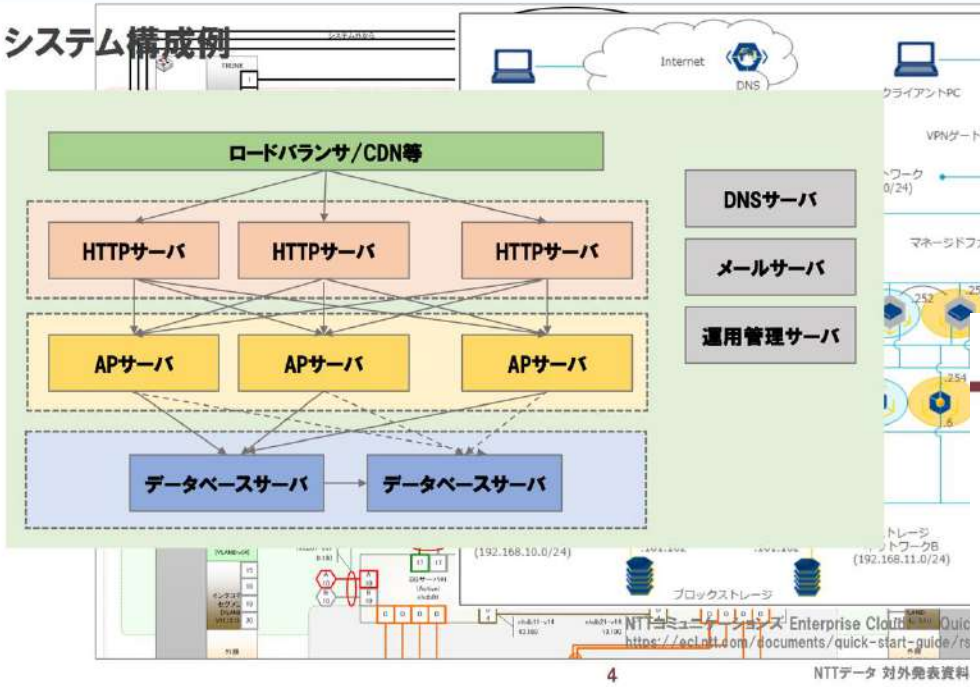
まとめ

- 高可用システムの実現方式
  - 可用性と影響のある事象
  - 指標：MTBF、MTTR、稼働率、RTO、RPO
  - 高可用システムの実現 - HAクラスタ
- キャパシティプランニングとスケーラビリティの確保
  - 性能要件の把握と評価
  - スケールアップとスケールアウト
- クラウドサービス上のシステム構成
  - ストレージサービス、ネットワーク、ネットワークセキュリティ
- 次回は「2.13.4 典型的なシステムアーキテクチャ」
  - Web 3層システムなど、代表的なシステム構成例をご紹介する予定

2.13：システムアーキテクチャ

現実のシステム構成

■ システム構成例



4

NTTデータ 対外発表資料

機能要件・非機能要件

■ ITシステムの要件（要求）は、機能と非機能に2側面がある
- 機能要件:　機能や挙動
- 非機能要件:　機能以外に具備しておくべきもの

■ 非機能要件は、IPA「非機能要求グレード」では大きく6つに分類
- 可用性
- 性能・拡張性
- 運用・保守性
- 移行性
- セキュリティ
- システム環境・エコロジー

- 結果が表示されるまで、どれくらいの時間が許容できるか？（レイテンシ）
- 同時にどの程度のアクセス数に耐えられる必要があるか？（同時接続数、スループット）
- ハードウェア障害が発生した場合にサービス継続ができるか？
：

※ IPA（独立行政法人 情報処理推進機構）非機能要求グレード 2018
https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html

■ システムアーキテクチャは、機能要件・非機能要件の両面を実現できるように検討する必要がある

5

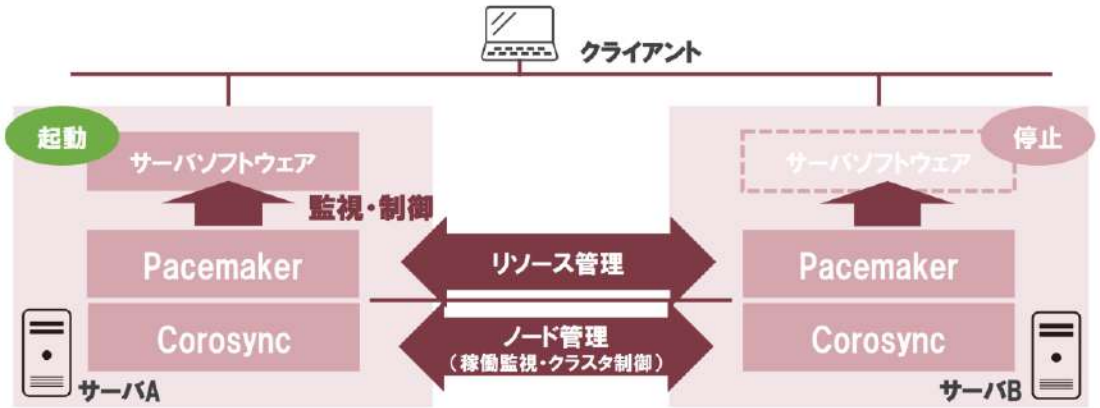
高可用性システムの実現

- 高可用性（High Availability）システム
  - システム障害が発生しにくいシステム
  - システムの一部で故障等が発生しても、システム全体が停止することなくサービス提供し続けられる
- 基本的には、同じ機能や役割に要素をあらかじめ複数用意しておき、異常が発生した場合に肩代わりできる仕組み（冗長化）による実現させる
- システムにおける SPoF を排除するようにシステムを設計
  - SPoF = Single Point of Failure（単一障害点）

・冗長化すると、構成する機器数を増えるため、故障頻
もある

Pacemaker、Corosyncによる構成例

- Pacemaker：サービスの監視や制御
- Corosync：サーバ間のメッセージ交換、ハードウェア制御など



実際に、SPoFをなくそうとすると・・・

- 前スライドほど簡単ではない
- 対処しないといけない課題（例）
  - サーバ内で持つデータの同期、ストレージの切換対応
  - ハードウェア故障への対応
    - いろんなところが壊れる・・・ネットワーク機器や配線の冗長化
  - 監視対象のサービスが正常に動作しているものの、アクセス増などで高負荷状態、レスポンスが悪化しているため異常と判断して、フェイルオーバ
    - フェイルオーバ先でも同じ症状になるので、また切り替わる（を繰り返す 等）
  - 監視用ネットワークのNIC故障やケーブル断になると、監視ができない
    - 双方から見えなくなるので、どちらのサーバも起動しようとして同一サービスが重複してしまう（スプリットブレイン）
  - 仮想化やコンテナで高収容化できたが、ハードウェア故障の影響が広範囲に及ぶ・・・
- 複雑であるが、奥が深く面白い領域でもある！（→ 304試験）

クラウドサービス – IaaS

- 仮想マシン・インスタンスは、様々なサービスが連携して、提供される

本日のゴール

ZEUS enterprise

- LinuCレベル2(202)の全体像を掴む
- LinuCレベル2(202)の学習方法を理解する
- LinuCレベル2(202)の受験計画を立てる

LinuC

資格を取得する、しない！？(取得するメリット)　※再掲

資格(知識)取得 × スキル(構築)保有、両方がある方がいい！

立場の違いはあれど、技術の前では平等！

学習はいつから始めても遅いということはない！
　※ITの分野は幅広く、かつそれぞれが深い

組織にいい流れをもたらす！

コミニティに入るきっかけとなる！

　→好循環な成長のループ♪

ゼウス・エンタープライズからの提供サービス

提供するITサービス

Network Engineering Service

高い専門スキルを有するエンジニア集団だから可能な質の高いソリューション

ゼウス・エンタープライズは、時代変革の要となるネットワーク・セキュリティ分野に特化したエンジニア集団として、顧客のニーズや課題に迅速かつ確実に応え、満足度の高いIT支援サービスを提供しています。情報通信・官公庁・金融・製造などの様々なクライアント先にて TCP/IPスタックの機器や、Linuxにおける豊富な経験と高度な技術を活用し、ネットワークやセキュリティ分野のパフォーマンスを最大限に引き出します。
主な業務としては、小規模LANから大規模WANまでのネットワーク構築や運用支援。各種アプリケーションの実行基盤やデータベースなど業務サーバーの構築や運用支援。また、オンプレ環境やクラウド環境、ハイブリッドクラウドの環境においても、セキュリティを重視した構築や運用支援を提供しています。
そして、当社は活躍する社員一人ひとりの能力を昇華させるべく、「ゼウスITトレーニングセンター」という教育機関を併設しており、ネットワークやLinuxを中心に、時代のトレンドに沿ったインフラ教育を行っています。
日々変革を遂げるIT業界に伴い、研修にて社員のスキルを底上げし、ネットワーク・セキュリティに特化したスペシャリスト集団として、クライアントの課題解決に貢献いたします。

https://www.zeus-enterprise.co.jp/solution/service/

リナックス・ネットワークに強いITスクール

IT Training

未経験者を戦力に育て上げた独自のカリキュラムに定評があるITキャリアスクールです

LPI-Japanのアカデミック認定校であるITキャリアスクール「ゼウスITトレーニングセンター」を運営し、リナックスとネットワークに強いエンジニアを育成します。当社社員の研修カリキュラムを基にしているため、プロの技術者だけでなく未経験者までも現場ですぐに活躍できる人材に育てます。セキュリティ分野から開発分野まで、クライアントの要望に合わせて講座を選択できるよう、幅広いコースを展開しています。

コース紹介
Course introduction

- Linux＆ネットワーク講座
- AWS講座
- Web資格対策講座
- Linux＆Network in English 講座
- 情報セキュリティ講座
- MS Office講座
- パソコン講座
- Kubernetes研修

https://www.it-training.tokyo/

ゼウス・エンタープライズからの提供サービス

技術を楽しみながら、一緒に働ける仲間を募集中！

https://www.zeus-enterprise.co.jp/recruit/

ZEUS enterprise

Thank you for joining my session!

LinuC

ZEUS enterprise

Q and A

LinuC