

# LinuC レベル 1 Version10.0 技術解説無料セミナー

2021/12/12 開催

安全・確実な作業に使用するLinuxコマンド  
～実際の現場で求められる技術を紹介～



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

## ■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 としふみ)

- 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。
- LinuCL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

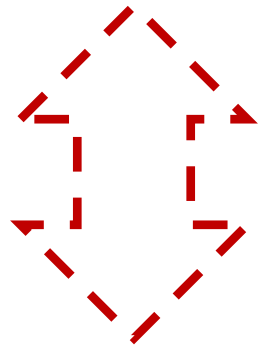
クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定

- ✓ 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
 

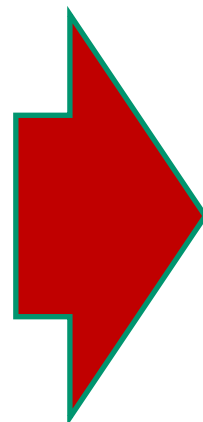
今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み
- ✓ Linuxの範疇だけにとどまらない領域までカバー
 

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

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



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

オンプレミスの  
サーバーサイドLinux技術

【今まで/その他】



- 本セミナーは、前回の2021/5/30に引き続き、「実際の現場で求められる技術の紹介」シリーズの2回目となります。
- 今回のテーマは**安全・確実な作業に使用するLinuxコマンド**です。
- どのように作業を安全・確実に実施するか？の考え方および、作業に使用するLinuxコマンドの実例を学びます。主にLinuCレベル1のコマンドが中心です。
- 安全・確実な作業について知りたい初級インフラエンジニアの方、LinuCで学ぶコマンドが実際の現場でどのように使われるのかを知りたい方、向けです。
- 本セミナーではCentOS7.9を使用しています。

• 主題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

• 主題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

• 主題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

• 主題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

• 主題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

- 主題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
- 主題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
- 主題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
- 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メール配送エージェント(MTA)の基本
- 主題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
- 主題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

- 主題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
- 2.01.3Linux カーネルの構成要素
- 2.01.4Linux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

- 主題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

- 主題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

- 主題2.04：システムの保守と運用管理

- 2.04.1make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

- 主題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

- 主題2.06：コンテナ

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



- ITシステムは構築作業の後には、システムを快適に利用するために更新作業、メンテナンス作業があります。

## システム更新作業

- ソフトウェアのバージョンアップ
- 機能追加
- 運用変更に伴う設定変更など

## システム メンテナンス作業

- 定期的な業務処理
- 定期的なバックアップ
- ソフトウェアのバグ修正アップデートなど

- ITシステムのサービスを提供する事業者は、取り決めたサービス時間帯にはユーザーがいつでもシステムを利用できるようにしています。
- ITシステムの更新作業やメンテナンス作業はサービス時間帯**以外**に行います。
- 万が一、作業が上手くいかなくても作業終了時にはユーザーがサービスを利用できる状態に戻すことが必要です。
- そのため、ITシステムへの作業は安全・確実に行う必要があります。



## 安全確実な作業の例

作業ログをとる

- 想定外発生時の調査のため

設定変更→確認の  
サイクル

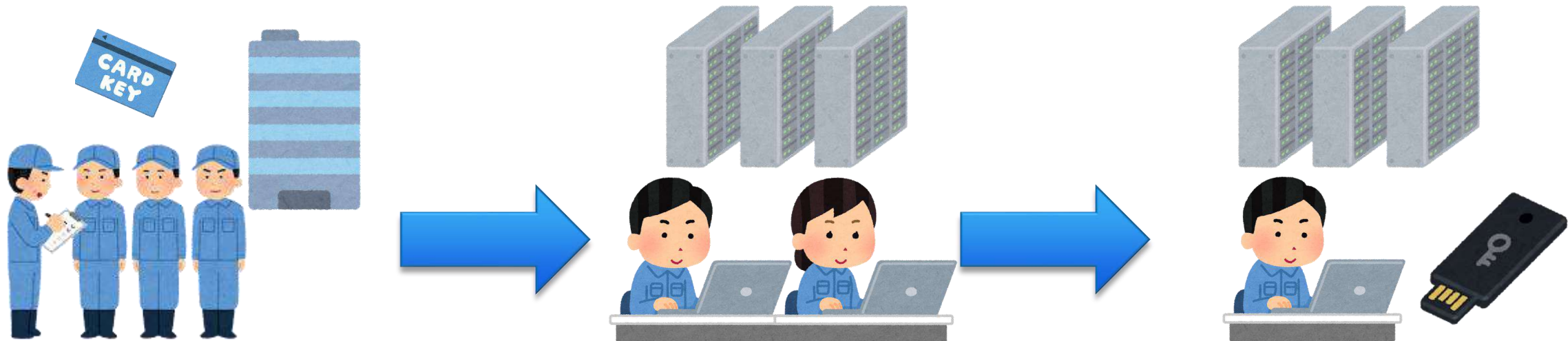
- 確実に実行するため

設定ファイルの  
バックアップ

- 元の状態に戻せるようにするため

手順書のダブル  
チェック

- ケアレスミスを防ぐため



## 事前作業

- 作業ログ取得
- ログイン
- サーバー状態確認
- ログファイル監視

## 作業

- **設定→確認**
  - ディレクトリ移動→確認
  - インストール→確認
  - 設定ファイル編集→確認
  - 設定再読み込み→確認
- など

## 事後作業

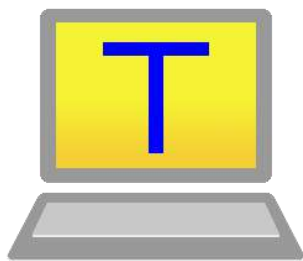
- 作業ファイルのバックアップ
- バックアップファイルを作業PCに転送
- ログアウト前の確認

本セミナーで扱う作業の目的は2点

- ① オリジナルのFile Not Foundページの作成
- ② ApacheでPHPを利用可能にする  
(Apache自体はインストール済)

サーバーに接続するために、ターミナルソフトウェアを使用します。  
 ターミナルソフトウェアとは、ネットワークを通じて他のコンピュータに接続し、コマンドライン方式で遠隔操作するためのソフトウェアのことです。

代表例として下記のようなソフトウェアがあります。  
 本セミナーではTeratermを使用します。



[Teraterm](#)



[PuTTY](#)



[RLogin](#)



[Windows Terminal](#)

①作業ログ取得



②ログイン



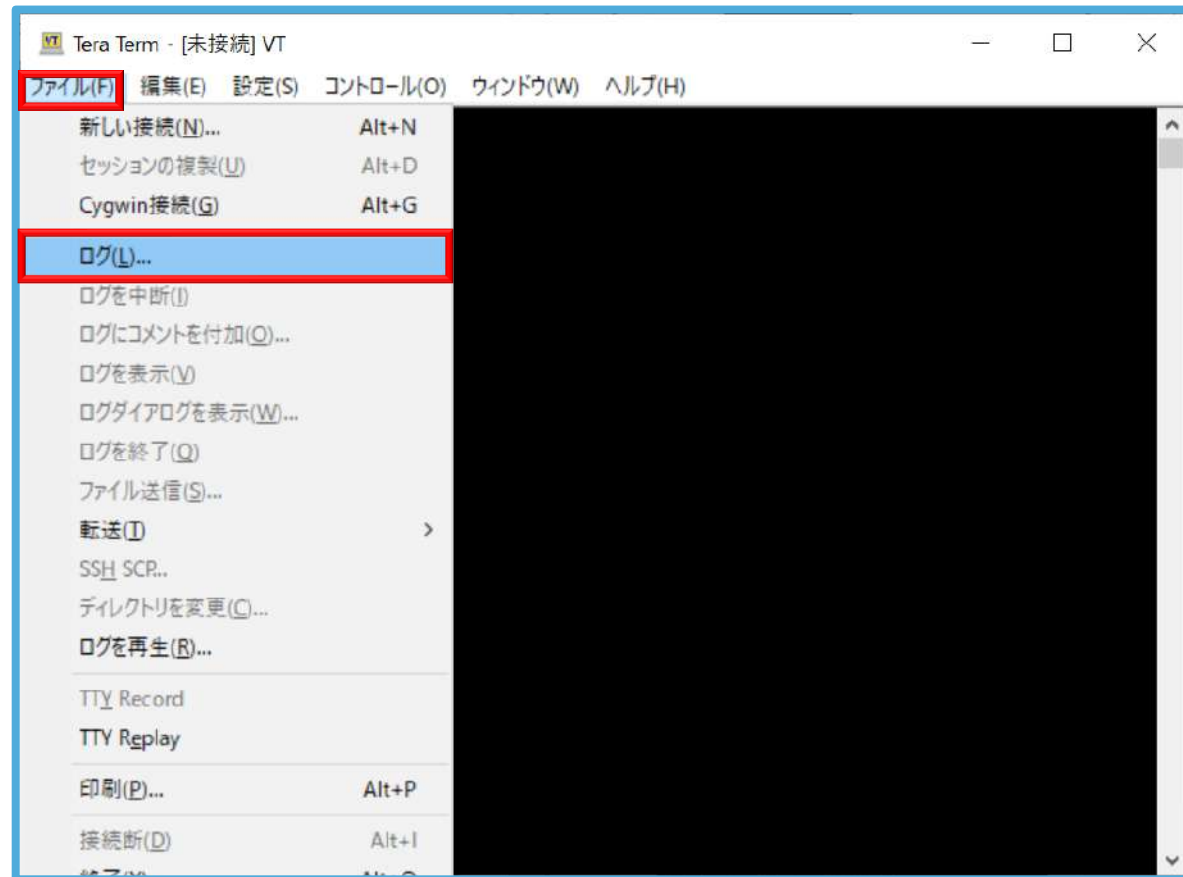
③サーバー状態確認



④ログファイル監視

サーバーに接続する前に作業ログを取得します。  
 作業ログとは作業中にキーボード入力および画面出力された文字を全てテキストファイルに記録するものです。

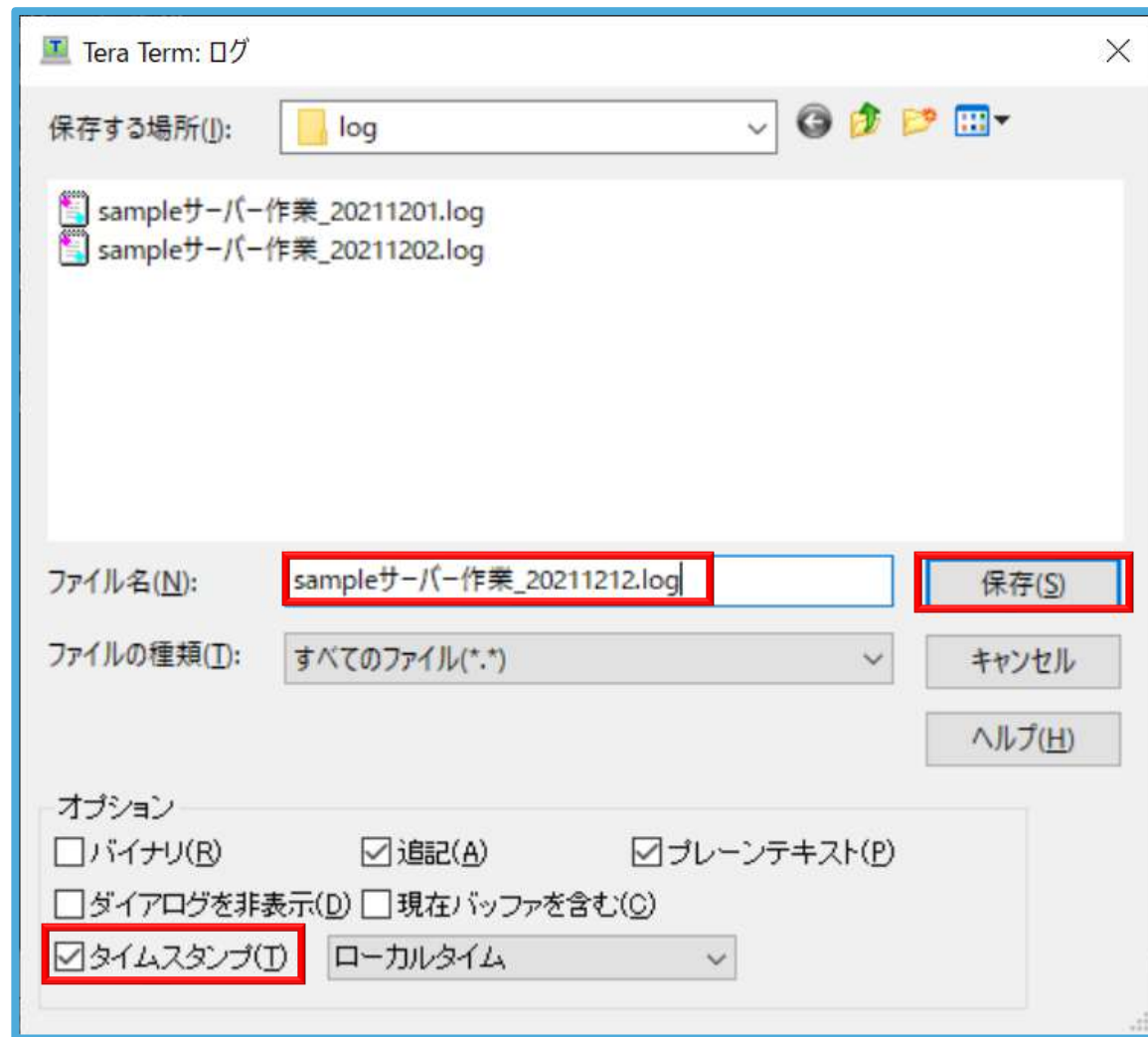
Teratermでは、  
 メニューの「ファイル」→「ログ」から記録します。





ログのファイル名を指定して保存します。  
命名規則は現場によって異なりますが、  
ここでは「作業名\_年月日.log」とします。

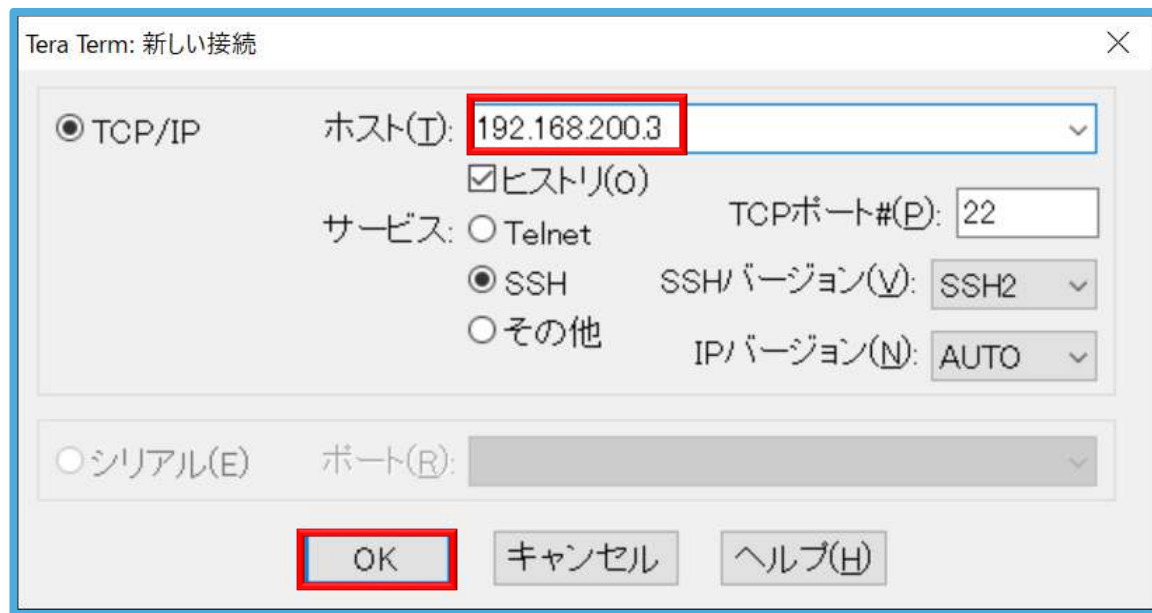
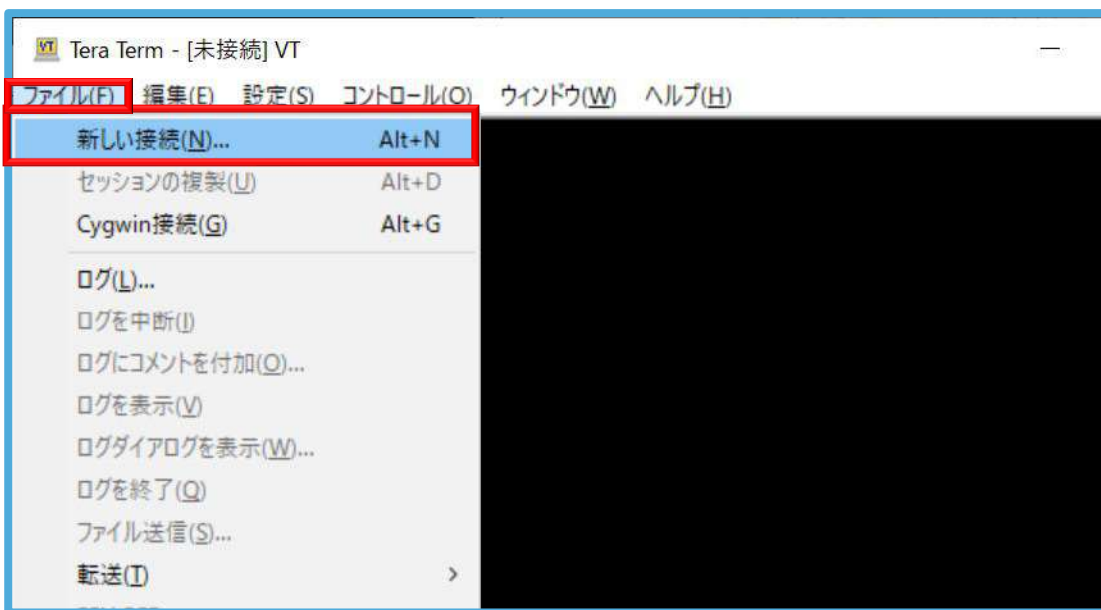
オプションの「タイムスタンプ」をチェックすることで、  
ログの行頭にタイムスタンプが付加されます。  
各コマンド実行にかかった時間が分かるので、作業後の  
確認に便利です。



サーバーにログインします。

「ファイル」→「新しい接続」を選択後、  
ホストに接続するIPアドレスを入力します。

ログイン方式としては、SSHv2の公開鍵認証が一般的です。

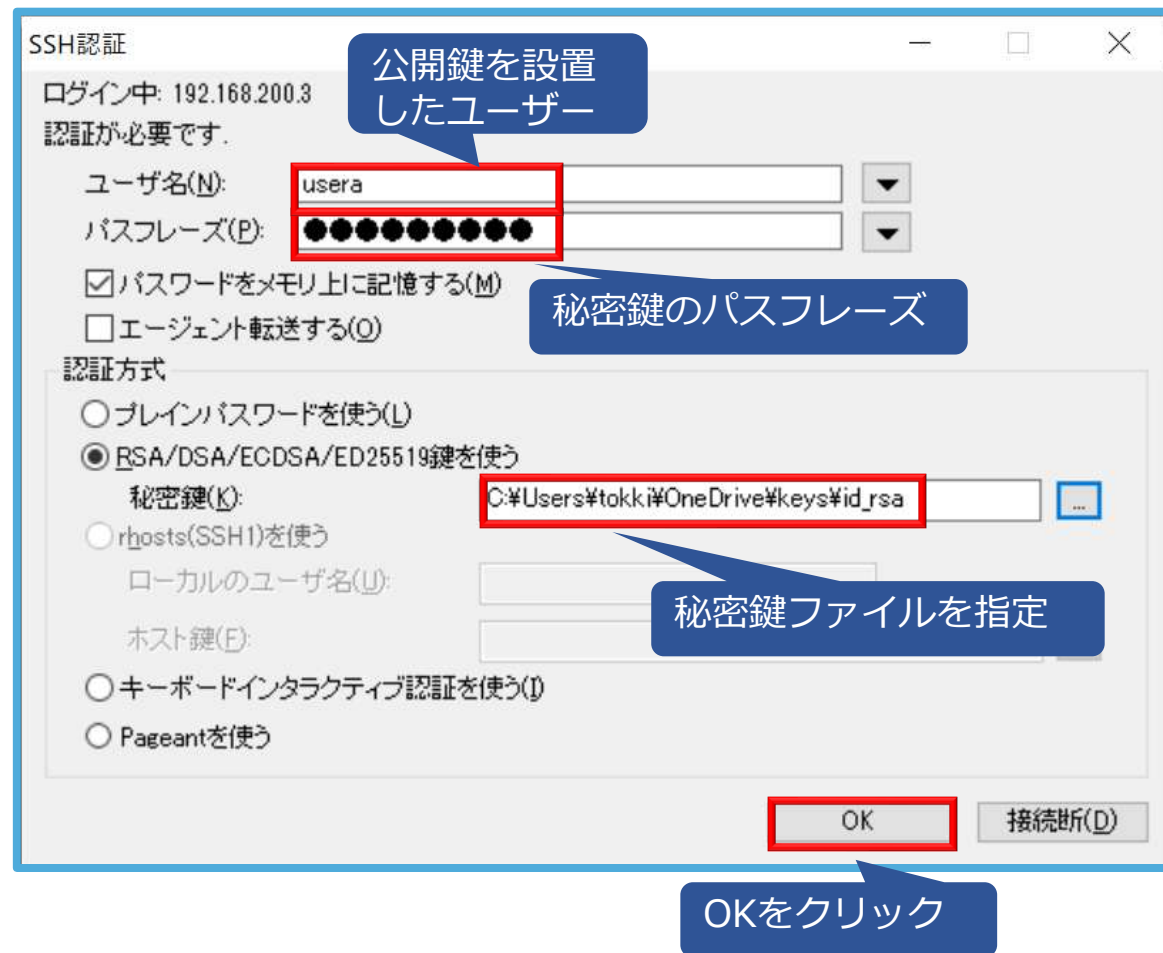


ここではサーバー側に公開鍵が設置済みとして、ペアの秘密鍵でSSHログインをします。

次の項目を入力していきます。

- 公開鍵設置済の一般ユーザー名
- 秘密鍵のパスフレーズ
- 認証方式 RSA/DSA/ECDSA/ED25519鍵を使うから秘密鍵を選択

→OKをクリック



\$ date (作業前の時刻確認)

```
2021年 12月 6日 月曜日 06:12:30 JST
```

\$ hostname ; nmcli | grep inet4 (ログインしたホストに間違いがないことを確認するため、ホスト名とIPアドレスを確認)

```
web01.engineer.jp
  inet4 192.168.255.4/24
  inet4 192.168.200.3/24
```

\$ w (直近のCPU負荷および他にログイン中のユーザーがないことを確認)

```
06:12:40 up 2:21, 1 user, load average: 0.03, 0.02, 0.05
USER      TTY      FROM          LOGIN@      IDLE        JCPU        PCPU        WHAT
usera    pts/1    192.168.200.1 06:00      2.00s      0.15s      0.28s      sshd: usera [priv]
```

\$ last (ログイン履歴の確認)

```
usera    pts/1    192.168.200.1  Mon Dec 6 06:00  still logged in
root     pts/0    192.168.200.1  Mon Dec 6 05:56 - 06:00  (00:03)
root     tty1                    Mon Dec 6 05:51 - 06:04  (00:13)
reboot   system boot 3.10.0-1160.el7. Mon Dec 6 05:50 - 08:12  (02:21)
```

\$ top (CPU、メモリ、プロセスの稼働状況に異常がないことを確認)

```
top - 19:15:00 up 1 day, 13:24, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 101 total, 2 running, 99 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1014756 total, 442900 free, 186648 used, 385208 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used, 666028 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	128156	6788	4196	S	0.3	0.7	0:11.78	systemd
9	root	20	0	0	0	0	S	0.3	0.0	0:06.03	rcu_sched
25630	root	20	0	162108	2248	1544	R	0.3	0.2	0:00.01	top
2	root	20	0	0	0	0	S	0.0	0.0	0:00.04	kthreadd
4	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
5	root	20	0	0	0	0	S	0.0	0.0	0:00.34	kworker/u2:0
6	root	20	0	0	0	0	S	0.0	0.0	0:01.62	ksoftirqd/0
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:01.55	watchdog/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
14	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
15	root	20	0	0	0	0	S	0.0	0.0	0:00.05	khungtaskd

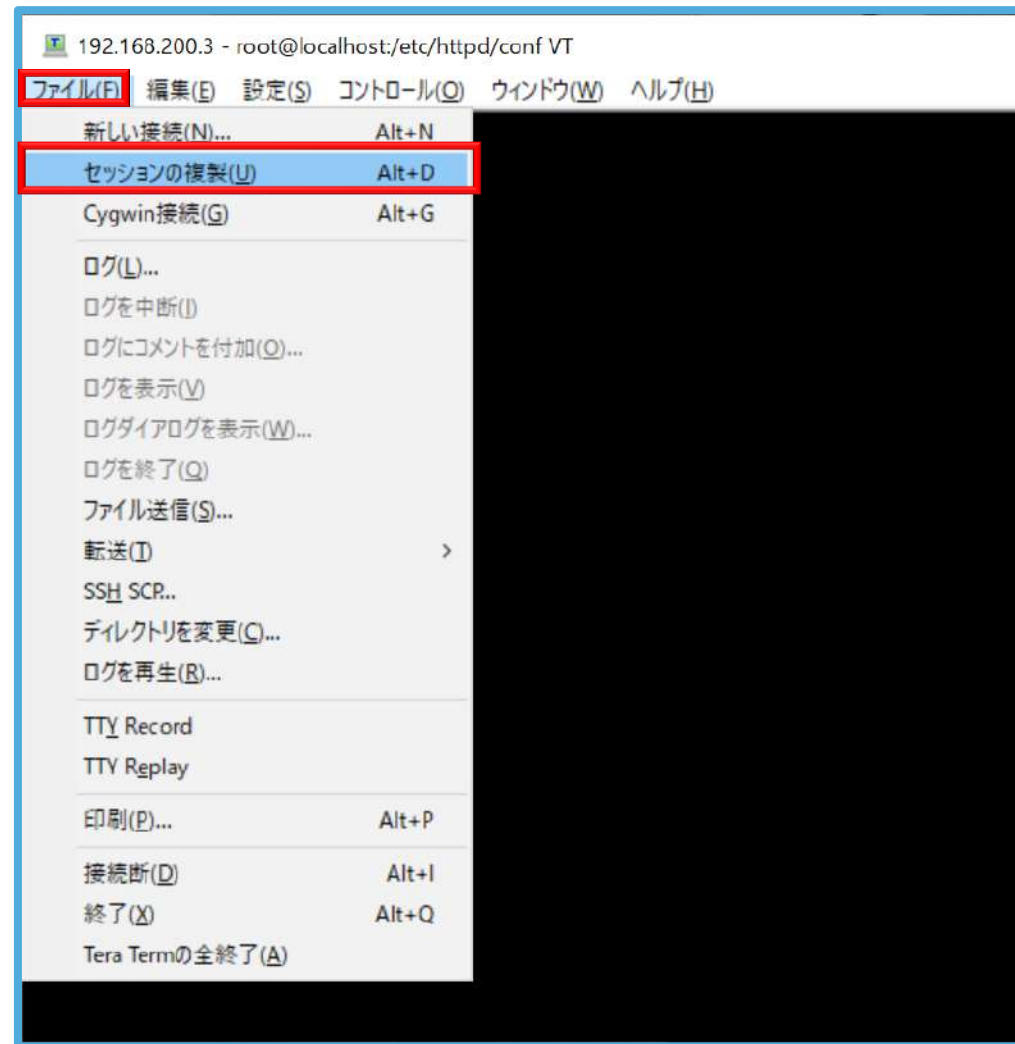
\$ su - (この後は管理者権限が必要なコマンドがあるため、rootユーザーに切り替える)

パスワード:

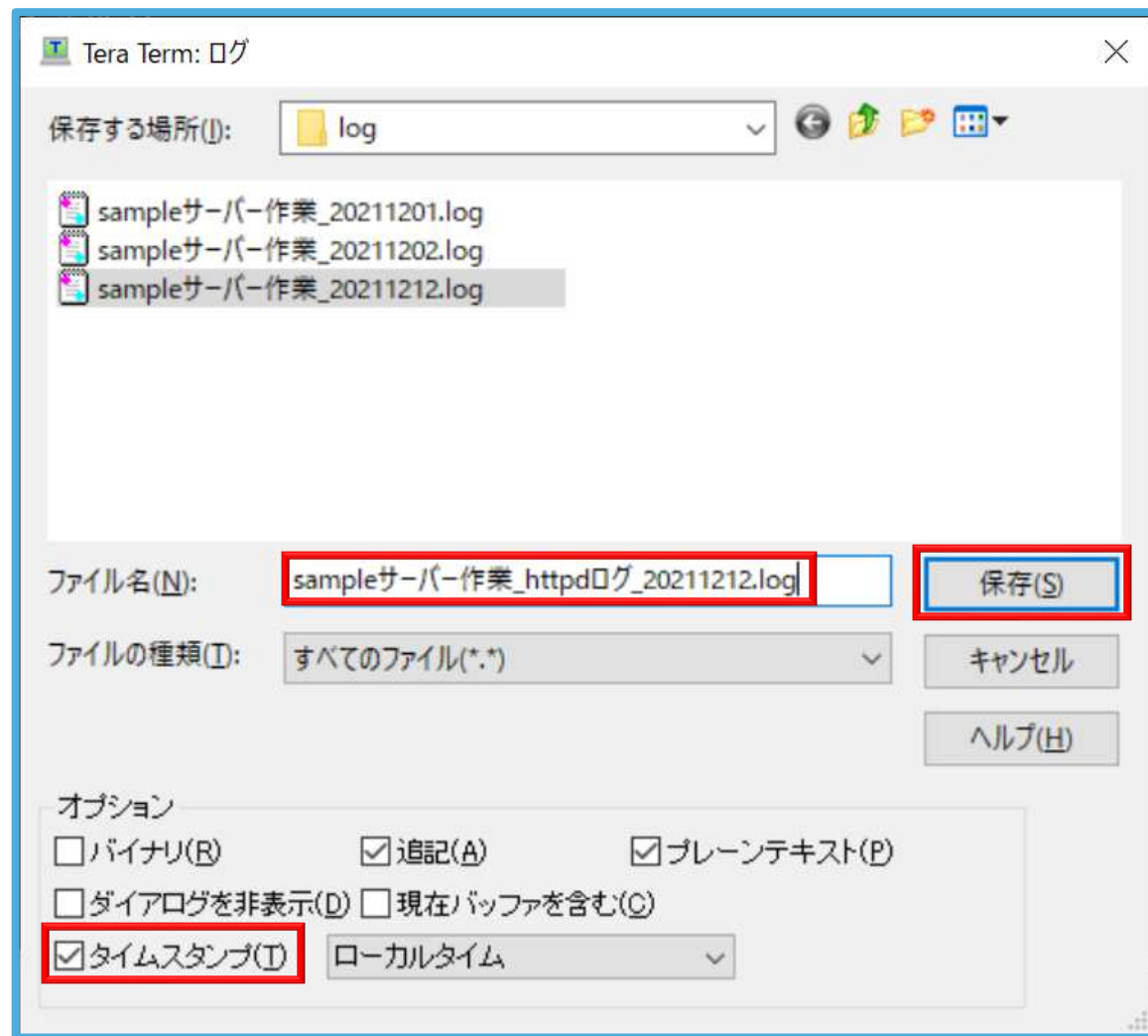
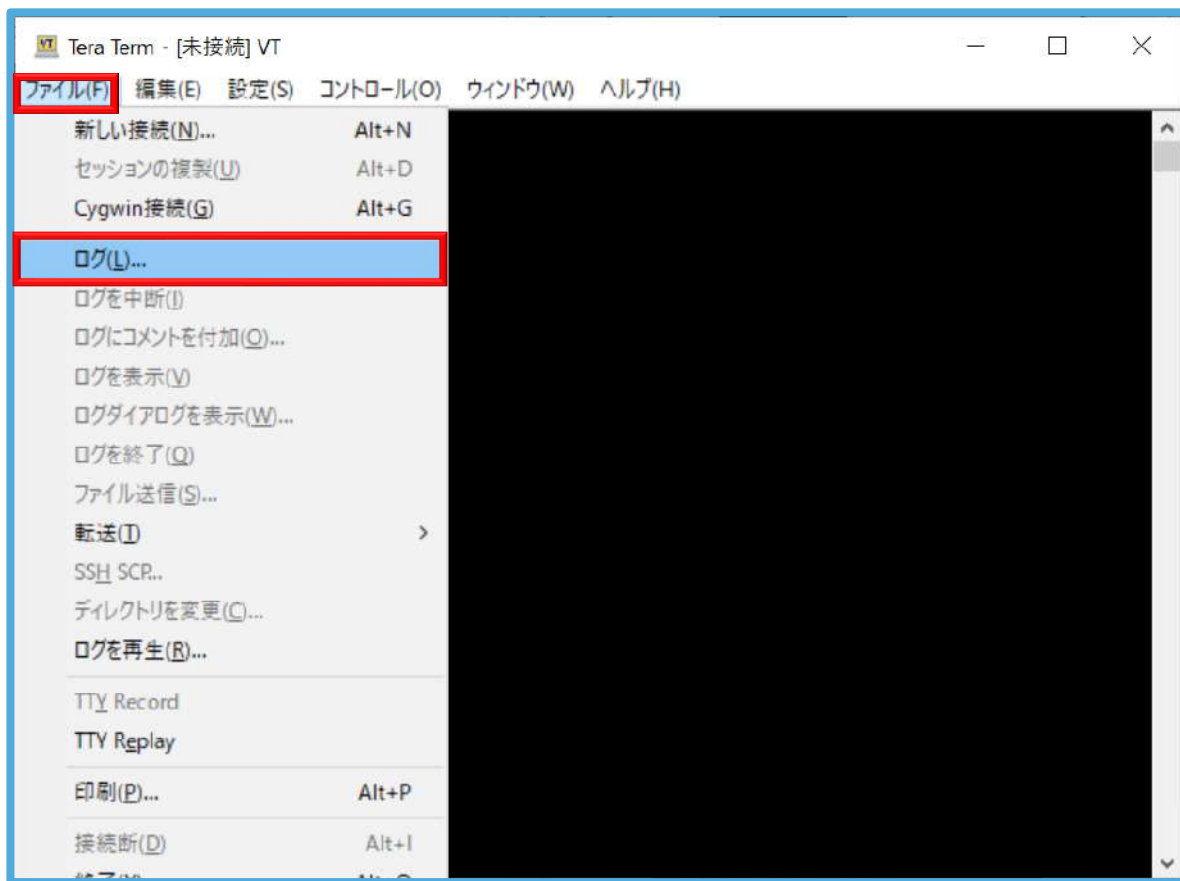
最終ログイン: 2021/12/04 (土) 04:34:08 JST 192.168.200.1から開始日時 pts/1

作業中にログファイルを監視しておくために、別のSSHセッションを開きます。

Teratermでは、メニューの「ファイル」→「セッションの複製」で現在ログイン中のセッションを複製します。



別セッションでも同じく作業ログを取得します。



\$ su - (この後は管理者権限が必要なコマンドがあるため、rootユーザーに切り替える)

パスワード:

最終ログイン: 2021/12/04 (土) 04:36:12 JST 192.168.200.1から開始日時 pts/1

# journalctl -fu httpd (作業中に発生するサーバーログを確認。-fオプションで最新情報を常に更新。)

```
-- Logs begin at 月 2021-12-06 05:50:54 JST. --
12月 06 19:17:40 web01.engineer.jp systemd[1]: Reloading The Apache HTTP Server.
12月 06 19:17:40 web01.engineer.jp systemd[1]: Reloaded The Apache HTTP Server.
12月 06 19:20:25 web01.engineer.jp systemd[1]: Stopping The Apache HTTP Server...
12月 06 19:20:26 web01.engineer.jp systemd[1]: Stopped The Apache HTTP Server.
12月 06 19:20:26 web01.engineer.jp systemd[1]: Starting The Apache HTTP Server...
12月 06 19:20:26 web01.engineer.jp systemd[1]: Started The Apache HTTP Server.
```



① httpdに異常のないことを確認（作業前）



② オリジナルのFile Not Foundページを使用する



③ ApacheでPHPを使用可能にする



④ httpdに異常のないことを確認（作業後）

# 作業 ① httpdに異常のないことを確認 (作業前)

# ps auxf | grep httpd (httpdプロセスが存在することを確認)

```

root      16330  0.0  0.0 112824   968 pts/0    S+   04:34   0:00 |      ¥_ grep --color=auto httpd
root      10456  0.0  0.1 151608  1604 pts/1    S+   12月06   0:00      ¥_ journalctl -fu httpd
root      10564  0.0  0.5 230440  5236 ?        Ss   12月06   0:03 /usr/sbin/httpd -DFOREGROUND
apache    10565  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
apache    10566  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
apache    10567  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
apache    10568  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
apache    10569  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
  
```

httpdのプロセスが存在している

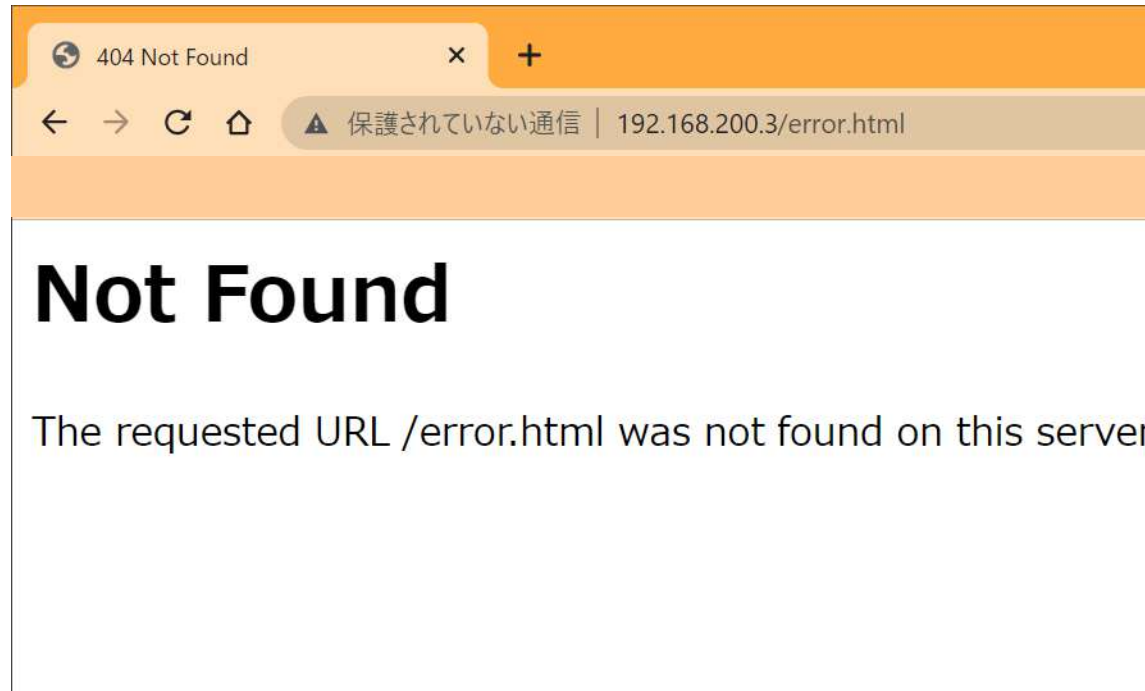
# ss -at | grep http (httpポート (TCP80番)が開いていることを確認)

```

LISTEN    0      128      [::]:http      [::]:*
  
```

File Not Found(404エラー)ページ変更前  
存在しないページ

<http://192.168.200.3/error.html>にアクセス



# 作業 ①オリジナルのFile Not Foundページを使用する

```
# cd /etc/httpd/conf (ディレクトリを移動する)
```

ディレクトリを移動するのは、以降のコマンドのパスの記述が長くなるのを防ぐため

```
# pwd (移動したディレクトリに間違いがないことを確認する)
```

```
/etc/httpd/conf
```

```
# cp -p httpd.conf httpd.conf_20211212 (設定ファイルを編集する前にバックアップ)
```

```
# ls -l httpd.conf* (バックアップできたことを確認)
```

ファイル名の指定に、シェルのメタキャラクタ\*を使用

```
-rw-r--r--. 1 root root 11752 12月 7 05:03 httpd.conf
-rw-r--r--. 1 root root 11752 12月 7 05:03 httpd.conf_20211212
```

```
# sed -i '333s/^#/#/' httpd.conf (333行目の行頭の#を削除)
```

viで編集しても良いが、編集ミスを防ぐためsedを使用

```
# diff -U 0 httpd.conf_20211212 httpd.conf (編集前ファイルとの差分を確認)
```

```
--- httpd.conf_20211212 2021-12-07 06:15:01.049213331 +0900
+++ httpd.conf 2021-12-07 06:23:38.945668917 +0900
@@ -333,333 @@
-#ErrorDocument 404 /missing.html
+ErrorDocument 404 /missing.html
```

diffのオプション  
-U : unified形式で出力する行数を指定 (デフォルトは前後3行)

```
# echo "指定されたページは存在しません。再度こちらからアクセスをお願いいたします。http://web01.engineer.jp" > /var/www/html/missing.html
# echoコマンドの結果をファイルにリダイレクト)
```

```
# cat /var/www/html/missing.html (意図したとおりにファイルに出力されていることを確認する)
```

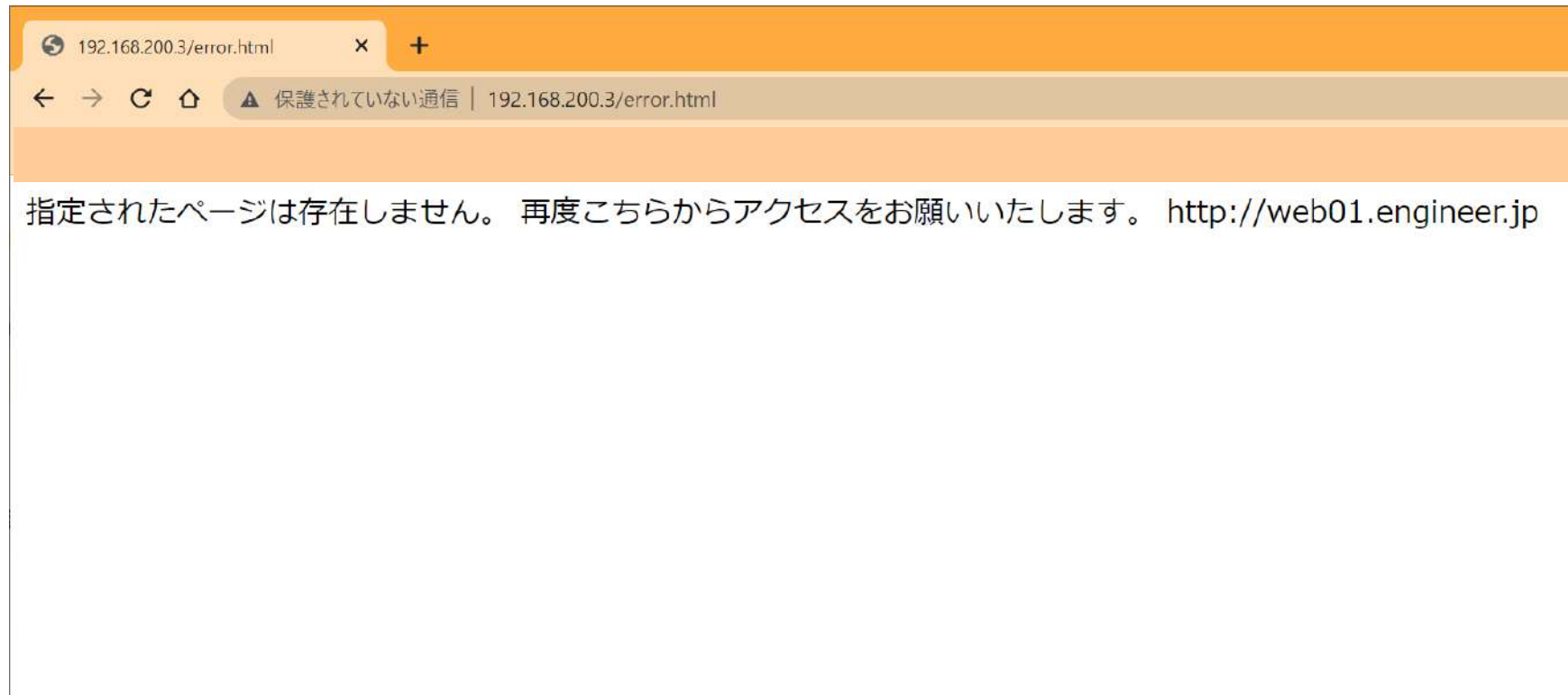
```
指定されたページは存在しません。再度こちらからアクセスをお願いいたします。http://web01.engineer.jp
```

```
# systemctl reload httpd (設定変更を反映させるためhttpdをreloadする)
```

別セッションの  
journalctl -fu httpdも確認して異常がないこと

File Not Found(404エラー)変更後  
存在しないページ

<http://192.168.200.3/error.html>にアクセス





# 作業 ③ ApacheでPHPを使用可能にする

# yum install -y php (yumでphpをインストール。-yですべての質問にyesで回答。)

```
インストール:
php.x86_64 0:5.4.16-48.el7
```

```
完了しました!
```

# yum list php (phpのパッケージがインストールできたことを確認)

```
インストール済みパッケージ
php.x86_64                               5.4.16-48.el7                               @base
```

yum listのみでもOK

# rpm -q php (phpのパッケージがインストールできたことを確認)

```
php-5.4.16-48.el7.x86_64
```

標準出力をリダイレクトで  
ファイルに出力

# echo "<?php phpinfo(); ?>" > /var/www/html/index.php (phpのテストページを作成)

# cat /var/www/html/index.php (意図したとおりにファイルに出力されていることを確認する)

```
<?php phpinfo(); ?>
```

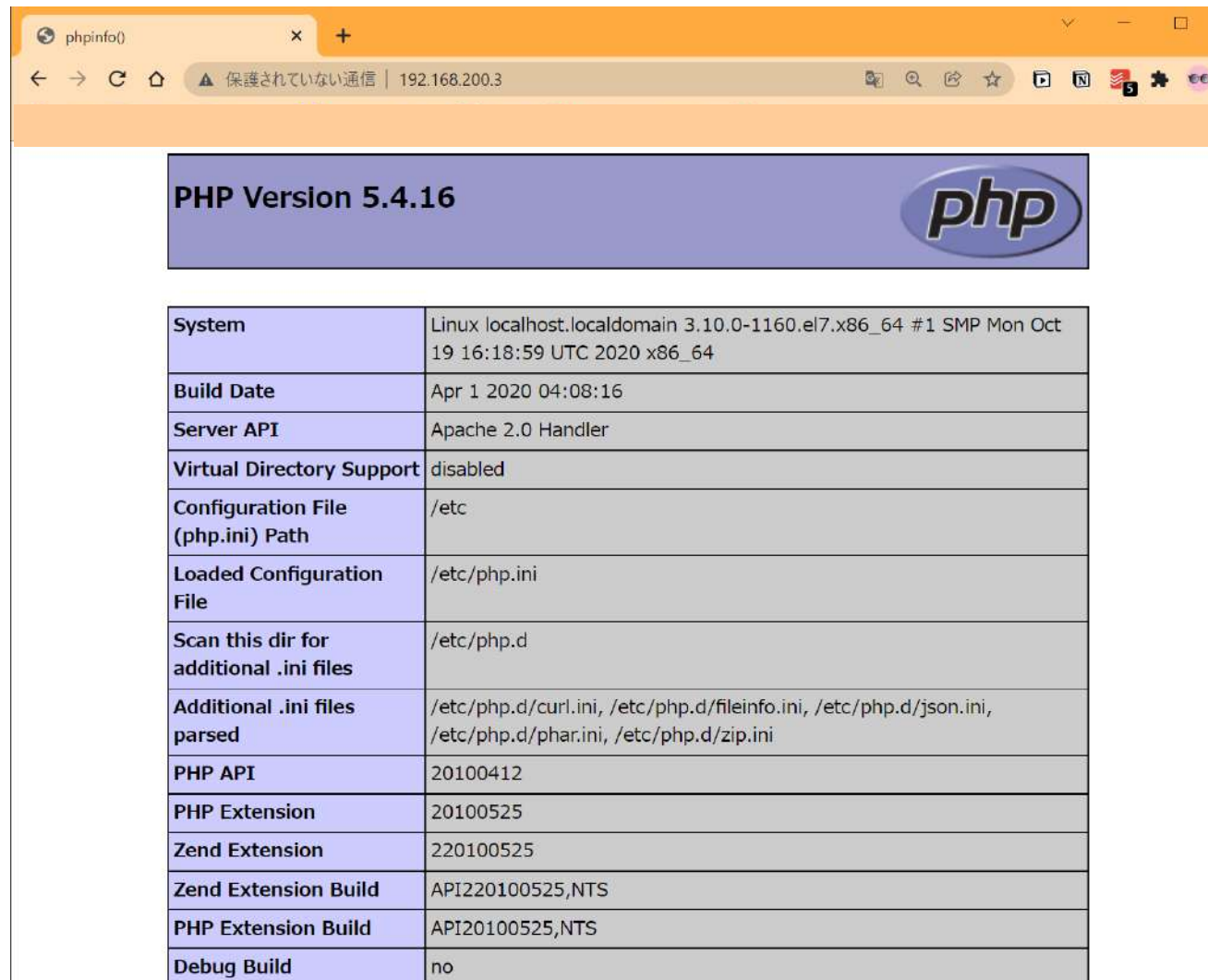
# systemctl restart httpd (phpのモジュールを読み込ませるためhttpdをrestartする)


別セッションの  
journalctl -fu httpdも確  
認して異常がないこと



## トップページ

<http://192.168.200.3>にアクセス



PHP Version 5.4.16 

System	Linux localhost.localdomain 3.10.0-1160.el7.x86_64 #1 SMP Mon Oct 19 16:18:59 UTC 2020 x86_64
Build Date	Apr 1 2020 04:08:16
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/phar.ini, /etc/php.d/zip.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS
PHP Extension Build	API20100525,NTS
Debug Build	no

# 作業 ④ httpdに異常のないことを確認 (作業後)

# ps auxf | grep httpd (httpdプロセスが存在することを確認)

```

root      16330  0.0  0.0 112824   968 pts/0    S+   04:34   0:00 |      ¥_ grep --color=auto httpd
root      10456  0.0  0.1 151608  1604 pts/1    S+   12月06   0:00      ¥_ journalctl -fu httpd
root      10564  0.0  0.5 230440  5236 ?        Ss   12月06   0:03 /usr/sbin/httpd -DFOREGROUND
apache    10565  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
apache    10566  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
apache    10567  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
apache    10568  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
apache    10569  0.0  0.2 230440  3000 ?        S    12月06   0:00 ¥_ /usr/sbin/httpd -DFOREGROUND
  
```

httpdのプロセスが存在している

# ss -at | grep http (httpポート (TCP80番) が開いていることを確認)

```

LISTEN    0      128      [::]:http      [::]:*
  
```

①作業ファイルのバックアップ



②バックアップファイルを作業PCに転送



③ログアウト前の確認

```
# cd (/rootディレクトリに移動)
```

```
# pwd (/rootディレクトリに移動できたことを確認)
```

```
/root
```

```
# mkdir 20211212_backup (バックアップディレクトリ作成)
```

```
# cp -p /etc/httpd/conf/httpd.conf* 20211212_backup  
(httpd.confの変更前後のファイルをバックアップディレクトリにコピー)
```

```
# cp -p /var/www/html/index.php /var/www/html/missing.html 20211212_backup  
(index.phpとmissing.htmlをバックアップディレクトリにコピー)
```

```
# ls -l 20211212_backup (正常にバックアップできたことを確認)
```

```
合計 32  
-rw-r--r--. 1 root root 11752 12月 6 06:20 httpd.conf  
-rw-r--r--. 1 root root 11753 9月 30 2020 httpd.conf_20211212  
-rw-r--r--. 1 root root 20 12月 6 07:36 index.php  
-rw-r--r--. 1 root root 136 12月 6 06:21 missing.html
```

```
# tar cvzf 20211212_backup.tar.gz 20211212_backup (バックアップディレクトリをtarアーカイブ+gzip圧縮)
```

```
20211212_backup/
20211212_backup/httpd.conf
20211212_backup/httpd.conf_20211212
20211212_backup/index.php
20211212_backup/missing.html
```

```
# ls -l 20211212_backup.tar.gz (アーカイブファイルができたことを確認)
```

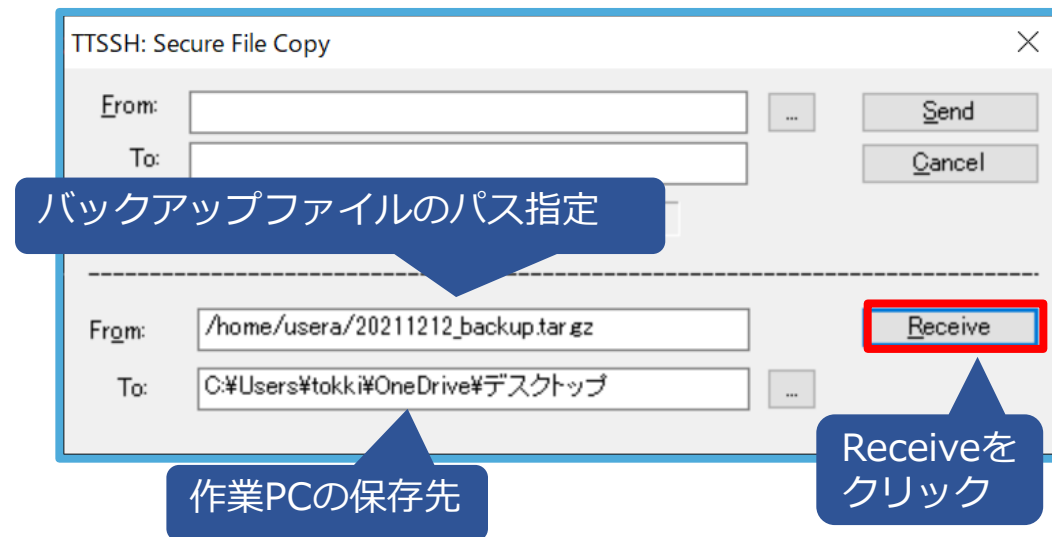
```
-rw-r--r--. 1 root root 5169 12月 6 07:44 20211212_backup.tar.gz
```

```
# mv 20211212_backup.tar.gz /home/usera
(作業PCにアーカイブファイルを転送するため、ログインユーザーのuseraにファイルを移動します)
```

```
# ls -l /home/usera (アーカイブファイルが移動できたことを確認します)
```

```
合計 8
-rw-r--r--. 1 root root 5169 12月 6 07:57 20211212_backup.tar.gz
```

TeratermのSSH SCP機能を使ってバックアップファイルを作業PCに転送します。



```
# ls -ld 20211212_backup (バックアップディレクトリの存在を確認します)
```

```
drwxr-xr-x. 2 root root 88 12月 6 07:44 20211212_backup
```

```
# rm -rf 20211212_backup (バックアップディレクトリを削除します)
```

```
# ls -ld 20211212_backup (バックアップディレクトリが削除できたことを確認します)
```

```
ls: 20211212_backup にアクセスできません: そのようなファイルやディレクトリはありません
```

(httpdのログに異常がないことを確認して、Ctrl+Cで停止)

```

-- Logs begin at 月 2021-12-06 05:50:54 JST. --
12月 06 06:21:35 web01.engineer.jp systemd[1]: Reloading The Apache HTTP Server.
12月 06 06:21:35 web01.engineer.jp systemd[1]: Reloaded The Apache HTTP Server.

12月 06 07:36:54 web01.engineer.jp systemd[1]: Reloading The Apache HTTP Server.
12月 06 07:36:54 web01.engineer.jp systemd[1]: Reloaded The Apache HTTP Server.
12月 06 07:38:02 web01.engineer.jp systemd[1]: Stopping The Apache HTTP Server...
12月 06 07:38:03 web01.engineer.jp systemd[1]: Stopped The Apache HTTP Server.
12月 06 07:38:03 web01.engineer.jp systemd[1]: Starting The Apache HTTP Server...
12月 06 07:38:03 web01.engineer.jp systemd[1]: Started The Apache HTTP Server.

^C

```

# exit (別セッションのrootをログアウト)

ログアウト

\$ exit (別セッションのuseraをログアウト)

ログアウト



# top (CPU、メモリ、プロセスの稼働状況に異常がないことを確認)

```
top - 20:06:59 up 1 day, 14:16, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 101 total, 1 running, 100 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 6.2 sy, 0.0 ni, 93.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1014756 total, 442468 free, 186460 used, 385828 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 666212 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	128156	6788	4196	S	0.0	0.7	0:12.04	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.04	kthreadd
4	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
5	root	20	0	0	0	0	S	0.0	0.0	0:00.34	kworker/u2:0
6	root	20	0	0	0	0	S	0.0	0.0	0:01.65	ksoftirqd/0
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:06.20	rcu_sched
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:01.59	watchdog/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
14	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	netns
15	root	20	0	0	0	0	S	0.0	0.0	0:00.06	khungtaskd

## # last (作業中にログイン履歴があったかを確認)

```

usera pts/0      192.168.200.1  Mon Dec 6 06:07 - 08:10 (02:03)
usera pts/1      192.168.200.1  Mon Dec 6 06:00  still logged in
root  pts/0      192.168.200.1  Mon Dec 6 05:56 - 06:00 (00:03)
root  tty1
reboot system boot 3.10.0-1160.el7. Mon Dec 6 05:50 - 08:12 (02:21)

```

## # w (ログイン中ユーザーの確認)

```

08:12:34 up 2:21, 1 user, load average: 0.03, 0.02, 0.05
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU WHAT
usera    pts/1    192.168.200.1 06:00    2.00s  0.15s  0.28s sshd: usera [priv]

```

## # date (作業終了時刻を記録)

```
2021年 12月 6日 月曜日 08:12:36 JST
```

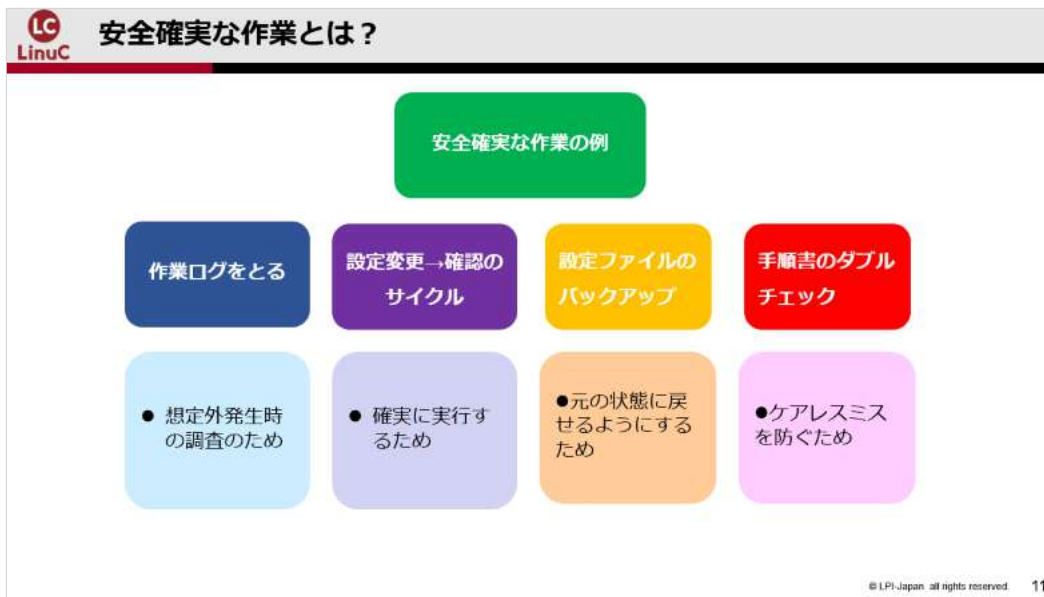
## # exit (rootをログアウト)

```
ログアウト
```

## \$ exit (useraをログアウト)

```
ログアウト
```

- 以上、安全確実に作業をするための工夫を見てきました。



- Linuxコマンドが作業でどのように使われるのかの参考になれば幸いです。