



LinuC レベル2 技術解説 オンラインセミナー

~受験に向けての準備とポイント解説~

2020/5/31 13:00-14:00

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エバンジェリスト)







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エバンジェリスト

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(<u>https://www.opensourcetech.tokyo/</u>)で執筆中。





Linus Torvaldsさん(Linux開発者)



Igor Sysoevさん(nginx開発者)



2







提供するITサービス

リナックス・ネットワークグループ



リナックスに特化したエンジニアとヘルプデスクの常駐型支援サービスを行ってい ます。研修は、エンジニアの育成に特化したLPI-Japanのアカデミック認定校 「Zeus Linux Training Center / Zeus Network Training Center」で行います。 資格取得の支援、新しい知識の習得に努め、つねに高水準のサービスの提供をめざ しています。

コンピュータシステムの企画・開発・保守 請負業務

特定労働者派遣 ・ヘルプデスク



アプリケーショングループ



又接リーこスを行っています。 また、「失敗しないお店選び」をモットーとしたグルメサイト「Gourmet True」 を2020年1月にリリース、これを皮切りに様々なインタースットサービスも開発し

アプリケーション開発 システム開発



クロスボーダーSE&ITソリューショングループ

語学力堪能なエンジニア(クロスボーダーSE)とヘルプデスク(バイリンガルコー ディネータ)による支援サービスを行っています。 技術用語を多く含む会話の通訳、ドキュメントの翻訳という専門性を活かし、オフ ショア開発のプロジェクト、外国人技術者を多数かかえる現場などにありがちな意 思疎通面でのデメリット回避などに力を発揮します。

コンビュータシステムの企画・開発・保守 翻訳業務 通訳業務 積算代行 与信代行サービス 特定労働者派遣 ・バイリンガルコーディネータ ・クロスボーダーSE



Zeus Linux Training Center / Zeus Network Training Center

当社で培われた技術者向け研修カリキュラムを、一般のお客様や企業様向けに提供 することを目的としたITキャリアスクール「Zeus Linux Training Center / Zeus Network Training Center」を運営しています。 当スクールはLPI-Japanのアカデミック認定校です。未経験者からプロの技術者養 成まで、「現場ですぐに活躍できる人材」の育成をめざした研修は、技術者を求め る企業からも高い評価をいただいています。

教育事業(BtoB、BtoC) IT・英会話・マナー 社員教育 試験会場





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









- 1. LinuC Level2試験とは
 - 試験概要と特徴
- 2. 技術解説
 - 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
 - Dockerコンテナ操作デモ
- 3. Q&A





LinuCとは



2020年4月1日 LinuC レベル1・2 新バージョン Version10.0リリース!

LinuCレベル1 /レベル2 Version 10.0は、クラウド時代に活躍する全てのIT技術者が身につけておきたい内容を網羅すべく生まれ変わった、新しい「LinuC」です。

1. 現場で求められている技術要素に対応

・クラウドを支える仮想化技術領域に拡大

・オープンソースの文化への理解を追加

・システムアーキテクチャの要素を導入

2. 全面的に見直した出題範囲

新しい技術要素の追加はもちろん、古い技術の削除や入れ替え、出題構成の見直し、出題範囲の記述の詳細化 と平易な説明文への変更と併せて、学びやすさ教えやすさを実現

3. Linuxにとどまらない認定領域

Linuxシステムの構築・運用に関わるLinux技術者にとどまらず、クラウドシステムや各種アプリケーション開発に携わるIT技術者にとっても有効な技術認定へ発展



旧バージョンとの差分はについては、こちら。 <u>https://linuc.org/docs/linuc_ver10.pdf</u>



LinuC Level2試験とは



仮想環境を含むLinuxのシステム設計、ネットワーク構築において、アーキテクチャに基 づいた設計、導入、保守、問題解決ができる技術者を認定する試験

- ✓ Linuxシステムの設計、構築、監視、トラブルシューティングができる。
 ✓ 仮想マシンやコンテナの仕組みを理解し、その管理と運用ができる。
- ✓ セキュリティとシステムアーキテクチャの基本を理解し、サービスの設計、構築、運用・管理ができる。

201試験/202試験の両方に合格することでレベル2に認定 1試験あたり 15,000円(消費税別) ピアソンVUEにおけるCBT or PBT(団体受験用)

詳細は、<u>https://linuc.org/linuc2/</u>



LinuC レベル 2 Version 10.0 リリース!

LinuC レベル2は、クラウド時代の即戦力として必須となる仮想化マシン・ コンテナを含む Linux システムの設定・構築の技術認定に

クラウド時代に活躍する全ての IT エンジニアヘ

▶ LinuCレベル2 Version 10.0 リリース!





LinuC Level2試験とは

2.13.4 典型的なシステムアーキテクチャ



201試験	202試験
2.01:システムの起動とLinuxカーネル	2.07:ネットワーククライアントの管理
2.01.1 ブートプロセスとGRUB	2.07.1 DHCPサーバーの設定と管理
2.01.2 システム起動のカスタマイズ	2.07.2 PAM認証
2.01.3 Linux カーネルの構成要素	2.07.3 LDAPクライアントの利用方法
2.01.4 Linuxカーネルのコンパイル	2.07.4 OpenLDAPサーバーの設定
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	2.08:ドメインネームサーバー
2.02 : ファイルシステムとストレージ管理	2.08.1 BINDの設定と管理
2.02.1 ファイルシステムの設定とマウント	2.08.2 ゾーン情報の管理
2.02.2 ファイルシステムの管理	2.08.3 セキュアなDNSサーバーの実現
2.02.3 論理ボリュームマネージャの設定と管理	2.09 : HTTPサーバーとプロキシサーバー
2.03 : ネットワーク構成	2.09.1 Apache HTTPサーバーの設定と管理
2.03.1 基本的なネットワーク構成	2.09.2 OpenSSLとHTTPSの設定
2.03.2 高度なネットワーク構成	2.09.3 nginxの設定と管理
2.03.3 ネットワークの問題解決	2.09.4 Squidの設定と管理
2.04:システムの保守と運用管理	2.10 : 電子メールサービス
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	2.10.1 Postfixの設定と管理
2.04.2 バックアップとリストア	2.10.2 Dovecotの設定と管理
2.04.3 ユーザへの通知	2.11 : ファイル共有サービス
2.04.4 リソース使用状況の把握	2.11.1 Sambaの設定と管理
2.04.5 死活監視、リソース監視、運用監視ツール	2.11.2 NFSサーバーの設定と管理
2.04.6 システム構成ツール	2.12 : システムのセキュリティ
2.05:仮想化サーバー	2.12.1 iptables や firewalld によるパケットフィルタリング
2.05.1 仮想マシンの仕組みとKVM	2.12.2 OpenSSH サーバーの設定と管理
2.05.2 仮想マシンの作成と管理	2.12.3 OpenVPNの設定と管理
2.06 : コンテナ	2.12.4 セキュリティ業務
2.06.1 コンテナの仕組み	2.13 : システムアーキテクチャ
2.06.2 Dockerコンテナとコンテナイメージの管理 🚽	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



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











学習に役立つメールマガジン LPI-Japanでは、試験レベルごとの例題解説など、 学習に役立つメールマガジンを無料でお届けしています。

過去のメールマガジンの 例題解説をまとめています。

LPI-Japanでは、試験レベルごとの例題解説など、 学習に役立つメールマガジンを無料でお届けしています。



LPI-Japanでは、『LinuCレベル1~新出題範囲における 受験準備とポイント解説』など、レベル別の 技術解説無料セミナーを開催しています。 学習の仕方で迷ったら是非足を運んでみてください。 他の受験者の方と意見交換もでき、モチベーションもあがります!

過去のセミナー資料のダウンロードはこちら⊙









主題2.06:コンテナ

2.06.1 コンテナの仕組み

重要度 2

概要

基本的なコンテナの仕組みについて理解している。

詳細

物理マシン、仮想マシン、コンテナの特徴と違いを理解している。 コンテナのファイルシステムとイメージの関係を知っている。 コンテナを実現する技術の概念を知っている。 名前空間, cgroups

2.06.2 Dockerコンテナとコンテナイメージの管理

重要度 3

概要

Dockerを導入してコンテナ実行環境を構築できる。

Dockerコンテナを実行できる。

コンテナイメージを管理できる。

詳細

Dockerを導入して、ネットワークを構成する。

ポート変換, フラットL2ネットワーク

Dockerコンテナを実行して、停止する。

docker ps/stats, docker run/create/restart, docker pause/unpause, docker stop/kill, docker rm Dockerコンテナに接続してプロセスを実行する。

docker attach, docker exec

コンテナイメージを管理する。

Dockerレジストリ: docker images, docker pull, docker rmi, docker import Dockerfile: docker build, docker commit









2.06.1 コンテナの仕組み



© LPI-Japan / EDUCO all rights reserved. 10



名前空間 (namespaces)



名前空間は、 グローバルシステムリソースを抽象化層で覆うことで、 名前空間内のプロセスに対して、 自分たちが専用 の分離されたグローバルリソースを持っているかのように見せる仕組みである。 グローバルリソースへの変更は、 名前空 間のメンバーである他のプロセスには見えるが、 それ以外のプロセスには見えない。 名前空間の一つの利用方法はコンテナーの実装である。

名前空間	定数	分離対象	
Cgroup	CLONE_NEWCGROUP	Cgroupルートディレクトリ	こわらた使い
IPC	CLONE_NEWIPC	System V IPC/POSIX メッセージキュー	各コンテナを仮想的
Network	CLONE_NEWNET	ネットワークデバイス/スタック/ポート など	に隔離する
Mount	CLONE_NEWNS	マウントポイント	Linux kernelの機能
PID	CLONE_NEWPID	プロセスID	
Time	CLONE_NEWTIME	起動時刻/モノトニック時刻	
User	CLONE_NEWUSER	ユーザID/グループID	
UTS	CLONE_NEWUTS	ホスト名/NISドメイン名	



http://man7.org/linux/man-pages/man7/namespaces.7.html http://manpages.ubuntu.com/manpages/focal/ja/man7/namespaces.7.html https://linuxjm.osdn.jp/html/LDP_man-pages/man7/namespaces.7.html



LinuC

cgroups



cgroups(Control Groups)は、Linuxカーネルの機能の1つで、プロセスを階層的なグループに編成し、さまざまなタイ プのリソースの使用(メモリ、CPU,ネットワーク帯域など)を制限および監視できるようにします。 カーネルのcgroupインターフェースは、cgroupfsと呼ばれる疑似ファイルシステムを介して提供されます。



http://man7.org/linux/man-pages/man7/cgroups.7.html

https://wiki.archlinux.jp/index.php/Cgroups

https://access.redhat.com/documentation/ja-jp/red hat enterprise linux/8/html/managing monitoring and updating the kernel/setting-limitsfor-applications managing-monitoring-and-updating-the-kernel





















コンテナイメージ



DockerHubなどにあるコンテナイメージを利用できる(例えば、CentOS/Ubuntu/nginxなど)

end ocker hub ရ nginx		Explore	Pricing Sign In	Sign Up
🖶 Docker EE 🛛 🖶 Docker CE 🔲 Conta	s 🗭 Plugins			
Filters	25 of 60,713 results for nginx . <u>Clear search</u>			Most Popular 🔹
Docker Certified	nginx NGINX Updated 2 minutes ago			OFFICIAL IMAGE 👷 10M+ 10K+ Downloads Stars
Images Verified Publisher () Docker Certified And Verified Publisher Content	Official build of Nginx. Container Linux x86-64 ARM	A PowerPC 64 LE ARM 64 386 IBM	Z Application Infrastru	ucture
Official Images Official Images Published By Docker	· · · · · ·			
Categories 🔞	By nginx • Updated 12 hours ago			Downloads Stars
 Analytics Application Frameworks Application Infrastructure 	NGINX Ingress Controller for Kuber Container Linux x86-64	netes		

LinuC

https://hub.docker.com/search?q=&type=image







ベースイメージを取得し、そこにイメージの層を重ねて利用する



2.06.2 Dockerコンテナとコンテナイメージの管理

© LPI-Japan / EDUCO all rights reserved. 17

Containerized Applications

Dockerコンテナのライフサイクル

Windows10 Pro・Mac OS・Linuxの場合

docker

Why Docker? Products Use Cases Developers Company

Q Sign In Get Started

Get Started with Docker

We have a complete container solution for you - no matter who you are and where you are on your containerization journey.

Dockerの始め方

Windows10 Homeなどの場合

Install Docker Toolbox on Windows

•**0** f1269

Legacy desktop solution. Docker Toolbox is for older Mac and Windows systems that do not

meet the requirements of Docker Desktop for Mac and Docker Desktop for Windows. We recommend updating to the newer applications, if possible.

Estimated reading time: 10 minutes

Docker Toolbox provides a way to use Docker on Windows systems that do not meet minimal system requirements for the Docker Desktop for Windows app.

今回は、こちらを使って解説を行います。

v19.03.1

Compare 💌

guillaumerose released this on 1 Aug 2019 · 4 commits to master since this release

Please ensure that your system has all of the latest updates before attempting the installation. In some cases, this will require a reboot. If you run into issues creating VMs, you may need to uninstall VirtualBox before reinstalling the Docker Toolbox.

The following list of components is included with this Toolbox release. If you have a previously installed version of Toolbox, these installers will update the components to these versions.

Included Components

- docker 19.03.1
- docker-machine 0.16.1
- docker-compose 1.24.1
- Kitematic 0.17.7
- Boot2Docker ISO 19.03.1
- VirtualBox 5.2.20
- ▼ Assets 6

DockerToolbox-19.03.1.exe	231 MB
DockerToolbox-19.03.1.pkg	235 MB
T md5sum.txt	102 Bytes
D sha256sum.txt	168 Bytes
Source code (zip)	
Source code (tar.gz)	

>=

Docker Quickstart

Terminal

Dockerの始め方

インストール完了後、 スタートメニューや ショートカットアイコンから起動

LinuC

docker images (取得したコンテナイメージの一覧表示)

使い方: docker images

docker pull (コンテナイメージの取得)

使い方 : docker pull URL/取得するコンテナイメージ[:タグ名] ※デフォルトでは、<u>https://hub.docker.com/</u>からコンテナイメージ取得をする ※タブ名を未指定の場合、:latest が適用される

	kujir@DESKTOP-P5K	C2HP1 MINGW64 /c/Pi	rogram Files/Docker Too	olbox	
	REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
	kujir@DESKTOP-P5K \$ docker pull ngi Using default tag latest: Pulling f 54fec2fa59d0: Pul 4ede6f09aefe: Pul f9dc69acb465: Pul Digest: sha256:86 Status: Downloade docker.io/library	(2HP1 MINGW64 /c/Pr nx g: latest from library/nginx l complete l complete l complete ae264c3f4acb99b2de ed newer image for y/nginx:latest	rogram Files/Docker Too ee4d0098c40cb8c46dcf9e1 nginx:latest	blbox 1148f05d3a51c4df6758	c12
コンテナイメージ名とタグ	kujir@DESKTOP-P5K \$ docker images DEPOCITORY nginx	(2HP1 MINGW64 /c/Pr TAC latest	rogram Files/Docker Too IMAGE ID 602e111c06b6	CREATED 2 weeks ago	SIZE 127MB
				C.	

<u>docker create (コンテナ生成)</u>

使い方: docker create コンテナイメージ

<u>docker ps(稼働コンテナの一覧表示)</u>

使い方: docker ps [-a] ※-aをつけると、停止中などのコンテナを含めて表示する

kujir@DESKTOP-P5K2HP \$ docker images	1 MINGW64 /c/Program	n Files/Docker Toolbo	X			
REPOSITORY nginx	TAG latest	IMAGE ID 602e111c06b6	CREATED 2 weeks ago	SIZE 127MB		
kujir@DESKTOP-P5K2HP \$ docker create ngin 1983ce68f373fd93a046	1 MINGW64 /c/Program x 7dab4072600244769ed	n Files/Docker Toolbo 08603a66aec24cc561dd0)293c			
kujir@DESKTOP-P5K2HP	1 MINGW64 /c/Program	n Files/Docker Toolbo	X			
SontAiner ID 1983ce68f373	IMAGE nginx	COMMAND "nginx -g 'daemon of	CREATED 4 seconds ago	STATUS Created	PORTS	NAMES cocky_galois

コンテナID ※コンテナを制御するために使用

<u>docker run(コンテナ生成 & コンテナ起動)</u>

使い方: docker run オプション コンテナイメージ[:タグ名]

-*d*: バックグラウンドでコンテナを実施 -*i*: コンテナの標準入力を開く -*t*: 端末(*TTY*)を割り当てる -*p*: 公開ポートを指定(ポート変換) -*p* 8080:80 とした場合、ホストOSのTCP8080番ポート経由で、コンテナのTCP80番ポートへアクセスする

kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox
\$ docker run --name first_nginx -d -p 8080:80 nginx
eb8a1ae01cde6bd7046470b4216d9ff2fd46f6f7d14299388febd91b54c14911

ujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox

p docker ps						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
eb8a1ae01cde	nginx	″nginx -g 'daemon of…″	4 seconds ago	Up 3 seconds	0.0.0.0:8080->80/tcp	first_nginx

<u>docker stats(コンテナの稼働状況確認)</u>

使い方: docker stats [コンテナID]
※ <i>ctrl</i> + c で終了させる

kujir@DESKTOP-F	P5K2HP1 MINGW64 /c/	Program Files/Dock	er Toolbox								
CONTAINER ID container in container in	IMAGE nginx	COMMAND ″nginx -g 'o	daemon of…"	CREATED 7 minutes a	go S	TATUS Up 7 minutes	PORTS 0. 0. 0.	0:8080->80/tcp	NAMES firs	S st_ngin	х
kujir@DESKTOP-F \$ docker stats	25K2HP1 MINGW64 /c/l eb8a1ae01cde_	Program Files/Docke	er Toolbox								
😁 MINGW64:/c/Progra	m Files/Docker Toolbox								_		>
CONTAINER ID eb8a1ae01cde	NAME first_nginx	CPU % 0.00%	MEM USA 1.965Mi	GE / LIMIT B / 989.5MiB	MEM % 0. 20%	NET 1.4	I/0 <b 0b<="" td=""><td>BLOCK I/O OB / OB</td><td></td><td>PI 2</td><td>DS</td>	BLOCK I/O OB / OB		PI 2	DS

docker pause (コンテナプロセスの中断)

使い方: docker pause コンテナID

docker unpause (コンテナプロセスの再開)

使い方: docker unpause コンテナID

kujir@DESKTOP-P5K2H	P1 MINGW64 /c/Progra	m Files/Docker Toolbox				
© docker ps CONTAINER ID eb8a1ae01cde	IMAGE nginx	COMMAND "nginx -g 'daemon of…"	CREATED 14 minutes ago	STATUS Up 13 minutes	PORTS 0.0.0.0:8080->80/tcp	NAMES first_nginx
kujir@DESKTOP-P5K2H \$ docker pause eb8a eb8a1ae01cde	P1 MINGW64 /c/Progra 1ae01cde	m Files/Docker Toolbox				
kujir@DESKTOP-P5K2H	P1 MINGW64 /c/Progra	m Files/Docker Toolbox				
©CONTAINER ID eb8a1ae01cde	IMAGE nginx	COMMAND "nginx -g 'daemon of…"	CREATED 14 minutes ago	STATUS Up 14 minute: (Pau	PORTS 0. 0. 0. 0:8080->80	NAMES /tcp first_nginx
kujir@DESKTOP-P5K2H \$ docker unpause eb eb8a1ae01cde	P1 MINGW64 /c/Progra 8a1ae01cde	m Files/Docker Toolbox				
kujir@DESKTOP-P5K2H \$ docker ps	P1 MINGW64 /c/Progra	m Files/Docker Toolbox				
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
eboaraeorede	ng mx	ng mx -g daemon or ···	14 minutes ago	op 14 mmules	0. 0. 0. 0. 0. 0000-200/ LCp	TTSL_NgTNX

https://docs.docker.com/engine/reference/commandline/pause/ https://docs.docker.com/engine/reference/commandline/unpause/

<u>docker stop(コンテナの停止)</u>

使い方: docker stop コンテナID

-t:コンテナ停止の待ち時間を指定(デフォルト10秒)

<u>docker kill(コンテナの強制停止)</u>

使い方: docker kill コンテナID

<u>docker start (停止しているコンテナの起動)</u>

使い方: docker start コンテナID

<u>docker restart(コンテナの再起動)</u>

使い方: docker restart コンテナID

https://docs.docker.com/engine/reference/commandline/stop/ https://docs.docker.com/engine/reference/commandline/kill/ https://docs.docker.com/engine/reference/commandline/start/ https://docs.docker.com/engine/reference/commandline/restart/

<u>docker attach(稼働コンテナへの接続)</u>

使い方: docker attach コンテナID

※接続し作業完了後、ctrl + c でコンテナごと終了、ctrl + $p \cdot ctrl$ + q でコンテナ内のプロセスのみ終了となる

docker exec(稼働コンテナでのコマンド実行)

使い方: docker exec コンテナID 実行コマンド

-i:コンテナの標準入力を開く *-t*:端末(*TTY*)を割り当てる

kujir@DESKTOP-P5K2H	P1 MINGW64 /c/Program	n Files/Docker Toolbox			
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
eb8a1ae01cde 80/tcp first_ngin	nginx x	"nginx -g 'daemon of…"	14 hours ago	Up 4 minutes	0. 0. 0. 0:8080->
kujir@DESKTOP-P5K2H \$ docker exec -it e root@eb8a1ae01cde:/ 10.3 root@eb8a1ae01cde:/ exit	P1 MINGW64 /c/Program b8a1ae01cde /bin/bash # cat /etc/debian_ven # exit	n Files/Docker Toolbox n rsion			

<u>docker rm (コンテナの削除)</u>

使い方: docker rm コンテナID ※停止しているコンテナを削除

-f:稼働コンテナを強制削除

kujir@DESKTOP-P5K	2HP1 MINGW64 /c/	Program Files/Docker Toolbox			
\$ docker ps CONTAINER ID NAMES	IMAGE	COMMAND	CREATED	STATUS	PORTS
eb8a1ae01cde 80/tcp first_ng	nginx ;inx	″nginx -g 'daemon of…″	14 hours ago	Up 12 minutes	0.0.0.0:8080->
kujir@DESKTOP-P5K \$ docker stop eb8 eb8a1ae01cde	2HP1 MINGW64 /c/H a1ae01cde	Program Files/Docker Toolbox			
kujir@DESKTOP-P5K \$ docker rm eb8a1 eb8a1ae01cde	2HP1 MINGW64 /c/l ae01cde	Program Files/Docker Toolbox			

<u>docker rmi(コンテナイメージの削除)</u>

使い方: docker rmi コンテナイメージ[:タグ名]

kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox							
S docker Images REPOSITORY nginx	TAG latest	IMAGE ID 602e111c06b6	CREATED 2 weeks ago	SIZE 127MB			
kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox \$ docker rmi nginx Untagged: nginx:latest Untagged: nginx@sha256:86ae264c3f4acb99b2dee4d0098c40cb8c46dcf9e1148f05d3a51c4df6758c12 Deleted: sha256:602e111c06b6934013578ad80554a074049c59441d9bcd963cb4a7feccede7a5 Deleted: sha256:81eaddad75aaa517b4a597912da28c2f5b905f6e9789dce3aea874b040aad201 Deleted: sha256:73cafa8418003ecfaa02360f181c132b2cf4b61433e1bd5c84012941105865c8 Deleted: sha256:c2adabaecedbda0af72b153c6499a0555f3a769d52370469d8f6bd6328af9b13							
kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox							
\$ docker Images REPOSITORY	TAG	IMAGE ID	CREATED	SIZE			

<u>docker export(コンテナからtarball作成)</u>

使い方: docker export コンテナID > tarballファイル名

※実行時「permission denied」が出力される場合、
 管理者権限でDocker Quickstart Terminalを
 起動して試してください。

kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox \$ cat nginx.tar docker import - nginx:import_test sha256:db0b14f5f094e4475cb9fdf24a050eb1ddaed4c17fc6c468ea32ea3bedf8c690							
kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox \$ docker images							
REPOSITORY nginx nginx	TAG import_test latest	IMAGE ID db0b14f5f094 602e111c06b6	CREATED 6 seconds ago 2 weeks ago	SIZE 125MB 127MB			

<u>docker import(tarballからコンテナイメージ作成)</u>

使い方: cat tarballファイル名 | docker import - コンテナイメージ名:タグ名

kujir@DESKTOP-P \$ cat nginx.tar sha256:db0b14f5 [.]	5 <mark>K2HP1 MINGW64 /c/Pro</mark> docker import - ng f094e4475cb9fdf24a050	o <mark>gram Files/Docker To</mark> ginx:import_test Deb1ddaed4c17fc6c468e	<mark>olbox</mark> a32ea3bedf8c690	
kujir@DESKTOP-P \$ docker images	5K2HP1 MINGW64 /c/Pro	ogram Files/Docker To	olbox	
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	import_test	db0b14151094	6 seconds ago	125MB
nginx	latest	602e111c06b6	2 weeks ago	T2/MB

<u>Dockerfileを使ったコンテナの生成</u>

必要なもの(以下を同一ディレクトリに保存する)

- Dockerfile
- ・プログラムの設定ファイルやWebサーバのコンテンツファイルなど ※必要に応じて

Dockerfileのサンプル

FROM centos:latest・・・利用するベースイメージの指定 MAINTAINER SubMattNesk(kujiraitakahiro) <SMN@test.local>・・・メンテナーの情報記載 COPY ./nginx.repo /etc/yum.repos.d/・・・コンテナビルド時にコピーするファイルとその配置場所 RUN ["/bin/bash","-c","yum -y install nginx"]・・・コンテナビルド後、実行するコマンド EXPOSE 80・・・コンテナで公開するポート LABEL "version" = "0.02" ¥・・・コンテナに付与するラベル(複数行の場合、バックスラッシュを使う) "description" = "presented by SubMattNesk" ADD ./emon.html /usr/share/nginx/html/・・・コンテナにコピーするファイルとその配置場所 ENTRYPOINT ["/usr/sbin/nginx","-g","daemon off;"]・・・コンテナ構築時に実行するコマンド CMD ["-c","/etc/nginx/nginx.conf"]・・・ENTRYPOINTで使うコマンドオプション


```
kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox/Dockerfile
 S
Dockerfile emon.html nginx.repo ファイル配置
kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox/Dockerfile
 cat Dockerfile _
FROM centos:latest
MAINTAINER SubMattNesk(kujiraitakahiro) <*****@gmail.com>
COPY ./nginx.repo /etc/yum.repos.d/
RUN ["/bin/bash","-c","yum -y install nginx"]
EXPOSE 80
LABEL "version" = "0.02" ¥
      "description" = "presented by SubMattNesk"
                                                        Dockerfileの内容
ADD ./emon.html /usr/share/nginx/html/
#ENTRYPOINT : When "docker run", you can't overwrite it.
ENTRYPOINT ["/usr/sbin/nginx","-g","daemon off;"]
#CMD : When "docker run", you can overwrite it.
#For example, using command options.
CMD ["-c","/etc/nginx/nginx.conf"]
```


<u>docker build (Dockerfileからのコンテナイメージ生成)</u>

使い方: docker build Dockerfileがあるディレクトリパス

-t: タグ名を指定する

kujir@DESKTOP-P5K2HP	1 MINGW64 /c/Program	Files/Docker Too	lbox/Dockerfile				
\$ docker build -t sa	mple/nginx ./						
Sending build contex [.]	t to Docker daemon	7. 168kB					
Step 1/9 : FROM cente	os:latest						
> 470671670cac							
Step 2/9 : MAINTAINE	R SubMattNesk(kujira	iitakahiro) < **** ≯	≪**@gmail.com>				
> Running in 84e	c1c7fddc0						
Removing intermediate	e container 84ec1c7f	ddc0					
> 85601520b101							
Step 3/9 : COPY ./ng	inx.repo /etc/yum.re	epos.d/					
> 8a66d4d46391							1
Step 4/9 : RUN L"/bin	Successfully built	4cf1a6d017d5					
> Running in 4c00	Successfully tagged	sample/nginx:lat	est				
CentOS-8 - AppStream	SECURITY WARNING: Y	ou are building a	Docker image from Wi	ndows against a non-Wir	dows Dock	er host. All fil	les and director
CentOS-8 - Base	les added to build	context will have	-rwxr-xr-x permiss	ions. It is recommended	to doubl	e check and rese	et permissions f
Centus-8 - Extras	or sensitive files	and directories.					
nginx repo	IN STREET OF DEKOU			Ibou /Declar of Lin			
Dependencies resolved	KUJIY@DESKIOP-PSKZH	PI MINGW04 /C/Pro	gram Files/Docker Too	IDOX/DOCKERTITE			
Deekeese	a docker images	TAO			0175		
	nomplo/ngipy	lataat	10f106d017d5	About a minuto ago	201MP	生成されたコン	モナイメージ
Inotolling:			401180001705	ADOUL A MITTULE Ago	SZTWD -		
	nginy		602e111e06b6	2 mooks ago	123MD		
nginx	contos	latest	470671670020	2 months ago	227MB		
	Centus	Talest	470071070040	o morrers ago	237110		

docker commit (コンテナからコンテナイメージを生成)

使い方: docker commit コンテナID コンテナイメージ名: タグ名

kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox/Dockerfile							
\$ docker ps CONTAINER ID NAMES	IMAGE	COMMAND	CREATED	STATUS	PORTS		
e6afb3d950f1 80/tcp gallant	nginx t_mendel	″nginx -g 'daemo	n of…″ 3 hours ago	Up 3 hours	0. 0. 0. 0:8080->		
kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox/Dockerfile \$ docker commit e6afb3d950f1 dev_nginx:version1 sha256:58c7c751931d1f1f84b4eaa19e84e960a7a8b436807330f2caac8e2e796353b3 kujir@DESKTOP-P5K2HP1 MINGW64 /c/Program Files/Docker Toolbox/Dockerfile							
	TAC	IMACE ID	ODEATED	017E			
dev_nginx	version1	58c7c751931d	5 seconds ago	127MB 生成されたI	コンテナイメージ		
nginx nginx centos	import_test latest latest	db0b14f5f094 602e111c06b6 470671670cac	2 hours ago 2 weeks ago 3 months ago	125MB 127MB 237MB			

※生成したコンテナイメージは、docker login(Dockerhubへのログイン)・ docker push(Dockerhubへのアップロード)で共有・公開が可能

Dockerコンテナの操作デモ

- ▶ コンテナイメージの取得から起動まで
- Dockerfileを用いたコンテナ起動

docker ~Dockerfile を使用した、コンテナの作成・起動~

https://www.opensourcetech.tokyo/entry/2018/06/12/docker_%E3%80%9CDockerfile_%E3%82%92%E 4%BD%BF%E7%94%A8%E3%81%97%E3%81%9F%E3%80%81%E3%82%B3%E3%83%B3%E3%83%8 6%E3%83%8A%E3%81%AE%E4%BD%9C%E6%88%90%E3%83%BB%E8%B5%B7%E5%8B%95%E3% 80%9C

DockerコンテナのCentOS7で「Failed to get D-Bus connection: Operation not permitted」と出た時の対処方法 https://www.opensourcetech.tokyo/entry/20190222/1550822082

docker ~「Error: No such image: a7a5593ce840」の対応方法~ https://www.opensourcetech.tokyo/entry/2018/11/20/docker_%E3%80%9C%E3%80%8CError%3A_No_ such_image%3A_a7a5593ce840%E3%80%8D%E3%81%AE%E5%AF%BE%E5%BF%9C%E6%96%B9%E 6%B3%95%E3%80%9C

Dockerコンテナのネットワークについて

https://www.opensourcetech.tokyo/entry/2018/11/22/Docker%E3%82%B3%E3%83%B3%E3%83%86% E3%83%8A%E3%81%AE%E3%83%8D%E3%83%83%E3%83%88%E3%83%AF%E3%83%BC%E3%82 %AF%E3%81%AB%E3%81%A4%E3%81%84%E3%81%A6

Docker Compose(複数コンテナをまとめて管理するツール) <u>http://docs.docker.jp/compose/toc.html</u> <u>https://docs.docker.com/compose/</u> <u>https://github.com/docker/compose</u>

Docker Swarm(コンテナのクラスタ環境を提供) <u>http://docs.docker.jp/swarm/overview.html</u> <u>https://docs.docker.com/swarm/overview/</u>

Kubernetes(コンテナのクラスタ環境を提供) <u>https://kubernetes.io/</u> <u>https://kubernetes.io/ja/docs/home/</u> <u>https://github.com/kubernetes/kubernetes</u>

Q & A

 $^{\odot}$ LPI-Japan / EDUCO all rights reserved. 41

Thank you for join today's seminar!

https://www.opensourcetech.tokyo/ https://twitter.com/matt_zeus https://www.facebook.com/takahiro.kujirai.1