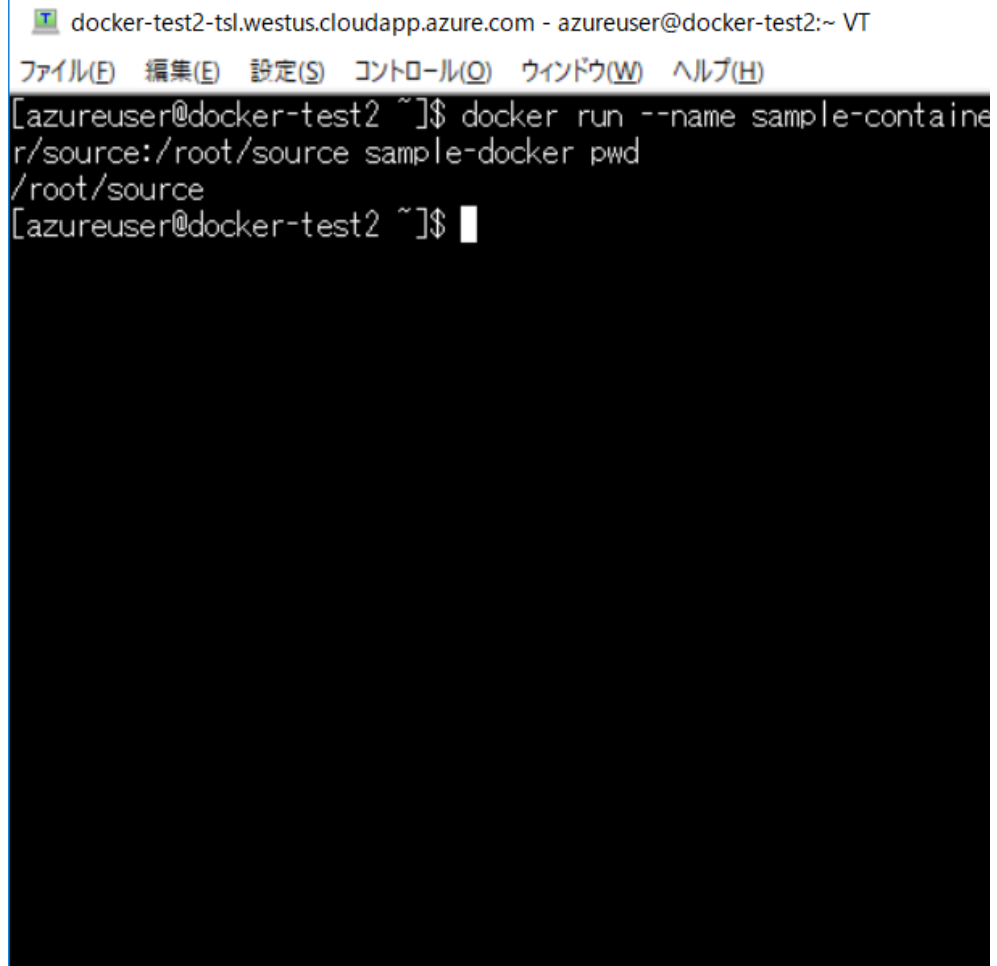


コマンドメモ

コンテナ側の、初期のカレントはどこか？

`docker run --name sample-container -v /home/azureuser/source:/root/source sample-d`



```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ docker run --name sample-container -v /home/azureuser/source:/root/source sample-docker pwd
/root/source
[azureuser@docker-test2 ~]$
```

元にした、docker-Image の、WORKDIR で指定された場所になっている

```
# Alpineイメージが軽いので利用
```

```
FROM alpine:3.5
```

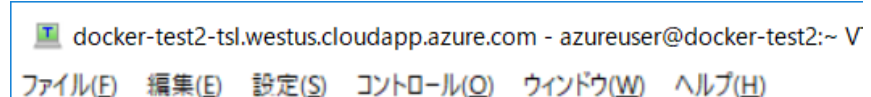
```
# GitからのソースDLと、コンパイルに必要なパッケージを追加する
```

```
RUN apk add --no-cache gcc make git
```

```
# カレントを /root/sourceにして、Gitから最新ソースをDLする
```

```
WORKDIR /root/source
```

この場合、ホスト側に、/root/source がなくても問題ない。（マウントされる）



```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
```

コマンドメモ

```
[azureuser@docker-test2 ~]$ sudo ls /root
anaconda-ks.cfg  original-ks.cfg  sample_jenkins
[azureuser@docker-test2 ~]$
```

ホストには /root/source はない

-v でマウント指定なしで、初期ディレクトリを ls してみると

`docker run --name sample-container`

```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ docker run --name sample-container -v /home/azureuser/source:/root/source sample-docker ls -la
total 0
drwxr-xr-x   2 root    root           6 Mar 31 08:03 .
drwx-----  1 root    root          20 Mar 31 08:03 ..
[azureuser@docker-test2 ~]$
```

なにもなし

-v で、マウントすると？

`docker run --name sample-container -v /home/azureuser/source:/root/source sample-docker`

```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ docker run --name sample-container -v /home/azureuser/source:/root/source sample-docker ls -la
total 8
drwxrwxr-x   2 1000    1000          41 Mar 31 07:27 .
drwx-----  1 root    root          20 Mar 31 08:03 ..
-rw-r--r--   1 1000    1000        215 Mar 31 07:27 Makefile
-rw-r--r--   1 1000    1000        604 Mar 31 07:10 hello.c
[azureuser@docker-test2 ~]$
```

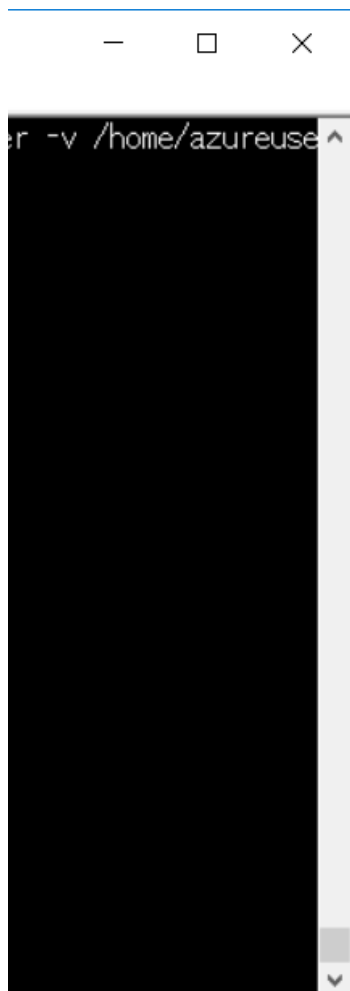
コマンドメモ

ホスト側にあった、`/home/azureuser/source` がマウントされた

■すべてのコンテナを一括削除
`docker rm $(docker ps -aq)`

コマンドメモ

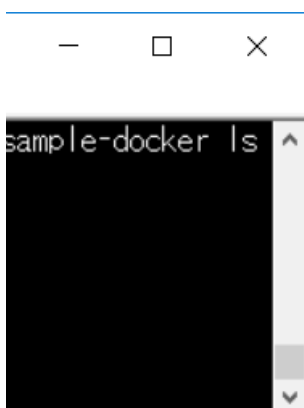
docker pwd



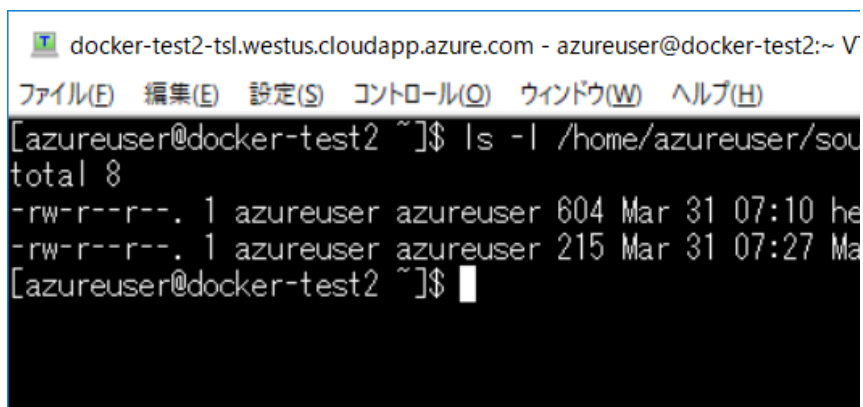
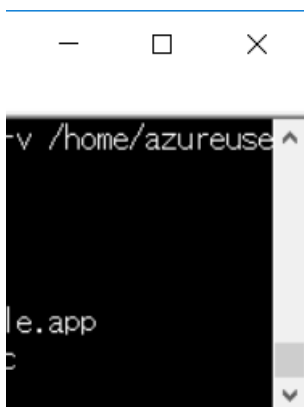
コマンドメモ



ame sample-container sample-docker ls -la

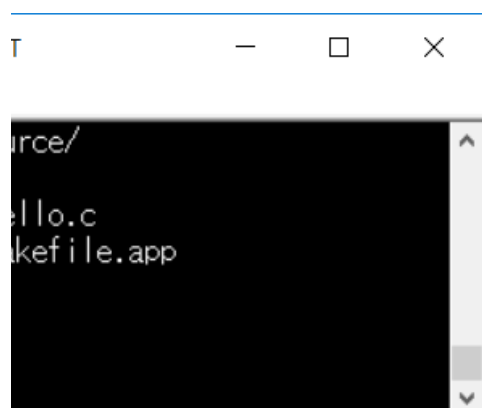


source:/root/source sample-docker ls -la



コマンドメモ
ホスト側でもても同じ

コマンドメモ



実行結果 2

GitからソースをDLして、ビルド

ホスト側 /home/azureuser

```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ ls -l
total 8
drwxrwxr-x. 3 azureuser azureuser 85 Mar 31 08:54 1st
-rw-r--r--. 1 azureuser azureuser 258 Mar 31 13:00 Dockerfile.build
-rw-r--r--. 1 azureuser azureuser 625 Mar 31 09:43 Makefile.docker
[azureuser@docker-test2 ~]$
```

※1st は実行結果1の資材 (今回は使わない)

実行

make -f Makefile.docker build

```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ make -f Makefile.docker build
# テスト環境の元になるイメージを作成
docker build -t sample-docker -f Dockerfile.build .
Sending build context to Docker daemon 39.94kB
Step 1/4 : FROM alpine:3.5
> 6.6094~107.5
```


実行結果 2

```

---> 6c0084ed37e3
Step 2/4 : RUN apk add --no-cache gcc libc-dev make git
---> Running in 656e718cb231
fetch http://dl-cdn.alpinelinux.org/alpine/v3.5/main/x86\_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.5/community/x86\_64/APKINDEX.tar.gz
(1/21) Installing binutils-libs (2.27-r1)
(2/21) Installing binutils (2.27-r1)
(3/21) Installing gmp (6.1.1-r0)
(4/21) Installing isl (0.17.1-r0)
(5/21) Installing libgomp (6.2.1-r1)
(6/21) Installing libatomic (6.2.1-r1)
(7/21) Installing pkgconf (1.0.2-r0)
(8/21) Installing libgcc (6.2.1-r1)
(9/21) Installing mpfr3 (3.1.5-r0)
(10/21) Installing mpc1 (1.0.3-r0)
(11/21) Installing libstdc++ (6.2.1-r1)
(12/21) Installing gcc (6.2.1-r1)
(13/21) Installing ca-certificates (20161130-r1)
(14/21) Installing libssh2 (1.7.0-r2)
(15/21) Installing libcurl (7.59.0-r0)
(16/21) Installing expat (2.2.0-r1)
(17/21) Installing pcre (8.39-r0)
(18/21) Installing git (2.11.3-r0)
(19/21) Installing musl-dev (1.1.15-r8)
(20/21) Installing libc-dev (0.7-r1)
(21/21) Installing make (4.2.1-r0)
Executing busybox-1.25.1-r1.trigger
Executing ca-certificates-20161130-r1.trigger
OK: 117 MiB in 32 packages
Removing intermediate container 656e718cb231
---> a2f970887995
Step 3/4 : RUN git clone --depth 1 https://github.com/hazuki3936/sample-docker
---> Running in e66362f66d67
Cloning into 'test1'...
Removing intermediate container e66362f66d67
---> 4a12cd4d86d0
Step 4/4 : WORKDIR test1
Removing intermediate container 669df054bebb
---> 5dac5380887f
Successfully built 5dac5380887f
Successfully tagged sample-docker:latest
# テスト環境のコンテナを起動
docker run --name sample-container sample-docker make -f Makefile.app
gcc -o hello.exe hello.c
./test1/hello.exe /test1/result.txt
# 実行結果は、 /tmp/ にコピーしておく
cp -f ./result.txt /tmp/result.txt
# 実行結果をホスト側に回収する
docker cp sample-container:/tmp/result.txt /home/azureuser/result.txt
# テスト環境のコンテナは破棄
docker rm sample-container
sample-container
[azureuser@docker-test2 ~]$

```

```

/test/hello.exe /test1/result.txt
# 実行結果は、/tmp/ にコピーしておく
cp -f ./result.txt /tmp/result.txt
# 実行結果をホスト側に回収する
docker cp sample-container:/tmp/result.txt /home/azureuser/result.txt
# テスト環境のコンテナは破棄
docker rm sample-container
sample-container
[azureuser@docker-test2 ~]$

```

結果を確認

```

docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ ls -lrt
total 12
drwxrwxr-x. 3 azureuser azureuser 85 Mar 31 08:54 1st
-rw-r--r--. 1 azureuser azureuser 258 Mar 31 13:00 Dockerfile.build
-rw-r--r--. 1 azureuser azureuser 503 Mar 31 13:14 Makefile.docker
-rw-r--r--. 1 azureuser azureuser 36 Mar 31 13:15 result.txt
[azureuser@docker-test2 ~]$ cat result.txt
[2018/03/31 13:15:59] Hello World!!
[azureuser@docker-test2 ~]$

```

ここで、Gitのソースを変更してみる

Hello World → Good Morning に変更

```

4
5  int main(int argc, char *argv[])
6  {
7      FILE *fo;
8      char nowTime[256] = {'\0'};
9      time_t timer;
10     struct tm *timeptr;
11
12     if (argc < 2) {
13         printf("Error: 引数の指定が誤っています。\\n");
14         return -1;
15     }
16     timer = time(NULL);
17     timeptr = localtime(&timer);
18     strftime(nowTime, 256, "[%Y/%m/%d %H:%M:%S]", timeptr);
19
20     fo = fopen(argv[1], "a");
21     if (fo == NULL) {
22         printf("Error: ファイルオープンに失敗 file[%s]\\n", argv[1]);
23         return -2;

```

```

13         printf("Error: 引数の指定が誤っています。\\n");
14         return -1;
15     }
16     timer = time(NULL);
17     timeptr = localtime(&timer);
18     strftime(nowTime, 256, "[%Y/%m/%d %H:%M:%S]", timeptr);
19
20     fo = fopen(argv[1], "a");
21     if (fo == NULL) {
22         printf("Error: ファイルオープンに失敗 file[%s]\\n", argv[1]);
23         return -2;
24     }
25
26     fprintf(fo, "%s Good Morning!!\\n", nowTime);
27
28     fclose(fo);
29
30     return 0;
31 }
32

```

再度実行

make -f Makefile.docker build

```

docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ make -f Makefile.docker clean
# 元になるイメージを削除
docker rmi sample-docker
Untagged: sample-docker:latest
Deleted: sha256:5dac5380887f69c592c494569d02bc7503dcc93134f01f70
Deleted: sha256:4a12cd4d86d0a4ef5a82dd2a1a6c90f1f965beb728f89c17
Deleted: sha256:8d360c29af80c050b0fb5aea5e7ff373c4694e95d1d8c31d
Deleted: sha256:a2f970887995b64b8a66964686976831f0fa5c87e0926778
Deleted: sha256:f637c0093bded8e49999525189259f864d5adf97df7c89d8
[azureuser@docker-test2 ~]$ make -f Makefile.docker build
# テスト環境の元になるイメージを作成
docker build -t sample-docker -f Dockerfile.build .
Sending build context to Docker daemon 40.96kB
Step 1/4 : FROM alpine:3.5
--> 6c6084ed97e5
Step 2/4 : RUN apk add --no-cache gcc libc-dev make git
--> Running in 957355b09376
fetch http://dl-cdn.alpinelinux.org/alpine/v3.5/main/x86_64/APKINDEX
fetch http://dl-cdn.alpinelinux.org/alpine/v3.5/community/x86_64/APKINDEX
(1/21) Installing binutils-libs (2.27-r1)
(2/21) Installing binutils (2.27-r1)
(3/21) Installing gmp (6.1.1-r0)
(4/21) Installing isl (0.17.1-r0)

```

```
(5/21) Installing libgomp (6.2.1-r1)
(6/21) Installing libatomic (6.2.1-r1)
(7/21) Installing pkgconf (1.0.2-r0)
(8/21) Installing libgcc (6.2.1-r1)
(9/21) Installing mpfr3 (3.1.5-r0)
(10/21) Installing mpc1 (1.0.3-r0)
(11/21) Installing libstdc++ (6.2.1-r1)
(12/21) Installing gcc (6.2.1-r1)
(13/21) Installing ca-certificates (20161130-r1)
(14/21) Installing libssh2 (1.7.0-r2)
(15/21) Installing libcurl (7.59.0-r0)
(16/21) Installing expat (2.2.0-r1)
(17/21) Installing pcre (8.39-r0)
(18/21) Installing git (2.11.3-r0)
(19/21) Installing musl-dev (1.1.15-r8)
(20/21) Installing libc-dev (0.7-r1)
(21/21) Installing make (4.2.1-r0)
Executing busybox-1.25.1-r1.trigger
Executing ca-certificates-20161130-r1.trigger
OK: 117 MiB in 32 packages
Removing intermediate container 957355b09376
---> 4bc2fcc2dc64
Step 3/4 : RUN git clone --depth 1 https://github.com/hazuki3936
---> Running in 784abc0e23a0
Cloning into 'test1'...
Removing intermediate container 784abc0e23a0
---> e72b3854bbc2
Step 4/4 : WORKDIR test1
Removing intermediate container aa43483347c1
---> 1db9cf7c1bde
Successfully built 1db9cf7c1bde
Successfully tagged sample-docker:latest
# テスト環境のコンテナを起動
docker run --name sample-container sample-docker make -f Makefile
gcc -o hello.exe hello.c
/test1/hello.exe /test1/result.txt
# 実行結果は、/tmp/ にコピーしておく
cp -f ./result.txt /tmp/result.txt
# 実行結果をホスト側に回収する
docker cp sample-container:/tmp/result.txt /home/azureuser/result.txt
# テスト環境のコンテナは破棄
docker rm sample-container
sample-container
[azureuser@docker-test2 ~]$ ls -lrt
total 12
drwxrwxr-x. 3 azureuser azureuser 85 Mar 31 08:54 1st
-rw-r--r--. 1 azureuser azureuser 258 Mar 31 13:00 Dockerfile.bu
-rw-r--r--. 1 azureuser azureuser 503 Mar 31 13:14 Makefile.dock
-rw-r--r--. 1 azureuser azureuser 37 Mar 31 13:21 result.txt
[azureuser@docker-test2 ~]$ cat result.txt
```

実行結果 2

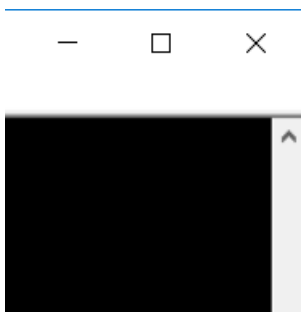
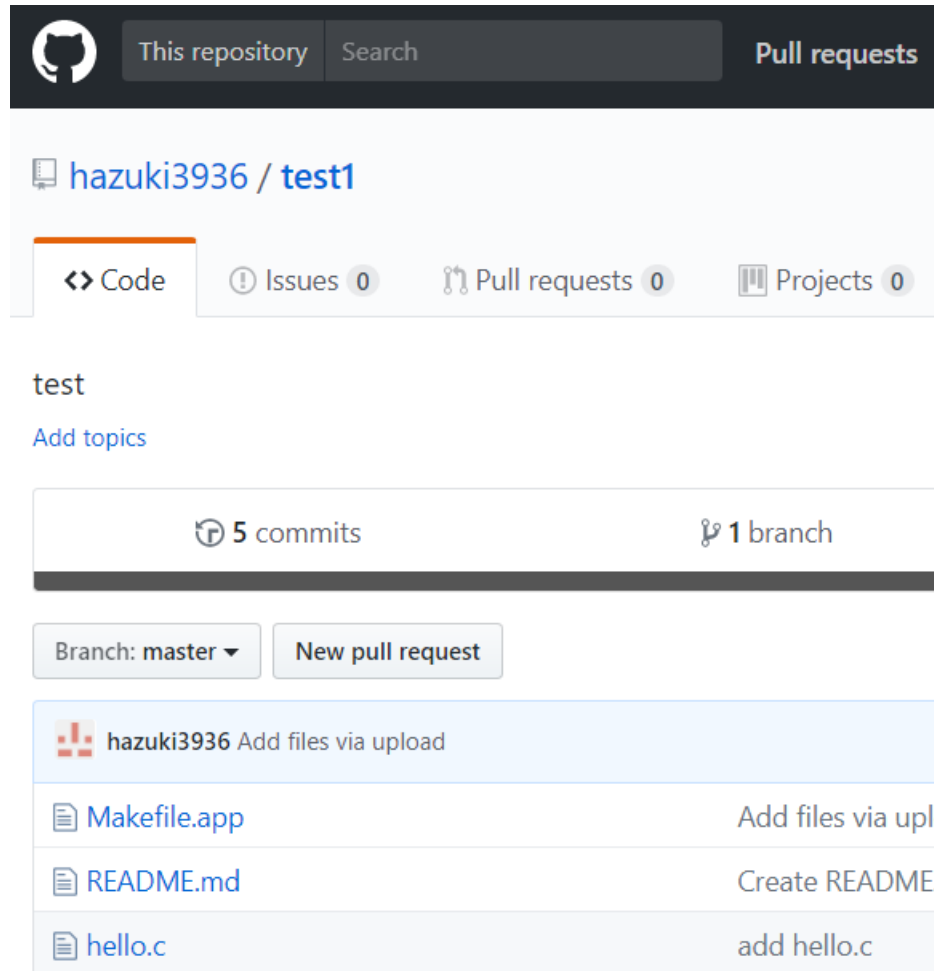
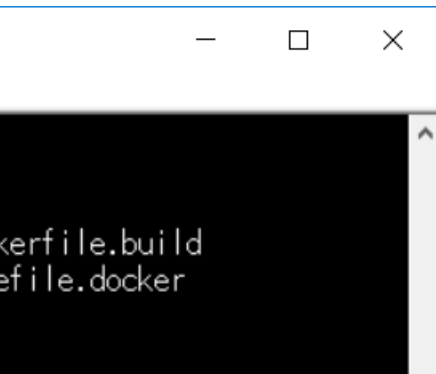
```
azureuser@docker-test2 ~$ cat README.txt  
[2018/03/31 13:21:53] Good Morning!!  
[azureuser@docker-test2 ~]$
```

最新ソースをクローンして、ビルド→実行している

あとはこれを、Jenkinsをつかって、リポジトリがPushされたら実行できれば良い

実行結果 2

Git側のリポジトリ



```
INDEX.tar.gz  
4/APKINDEX.tar.gz
```

```
$/test1.git
```

```
-f Makefile.app
```

```
user/result.txt
```

user/result.txt

実行結果 2

kerfile.build
efile.docker
ult.txt

\n");

tr);

le[%s]\n", argv[1]);


```
\n");
```

```
tr);
```

```
le[%s]\n", argv[1]);
```











実行結果 2




Pull requestsIssuesMarketplaceExplore





 Watch  0


 Star 0


 Fork 0

requests 0


 Projects 0


 Wiki


 Insights


 Settings

Edit

 1 branch




 0 releases

 1 contributor


Create new fileUpload filesFind fileClone or download 

Latest commit e0a8d86 just now

Add files via upload	just now
Create README.md	8 hours ago
add hello.c	7 hours ago



0

 Fork

0

Edit

Contributor

Clone or download ▾

commit e0a8d86 just now

just now

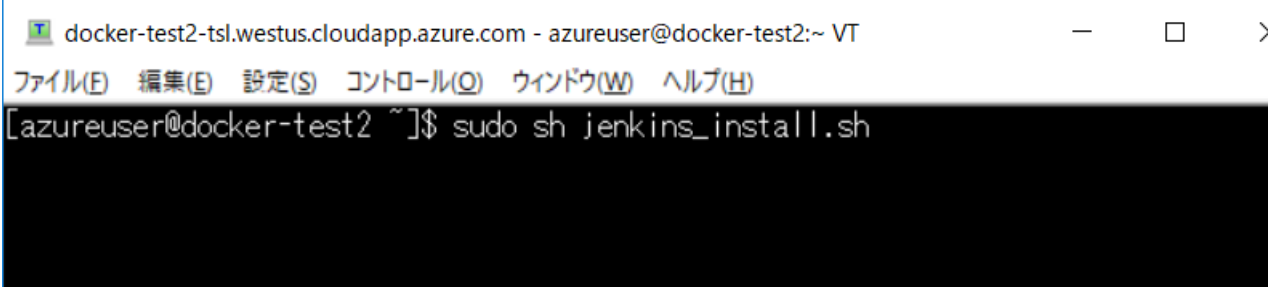
8 hours ago

7 hours ago

実行結果 3

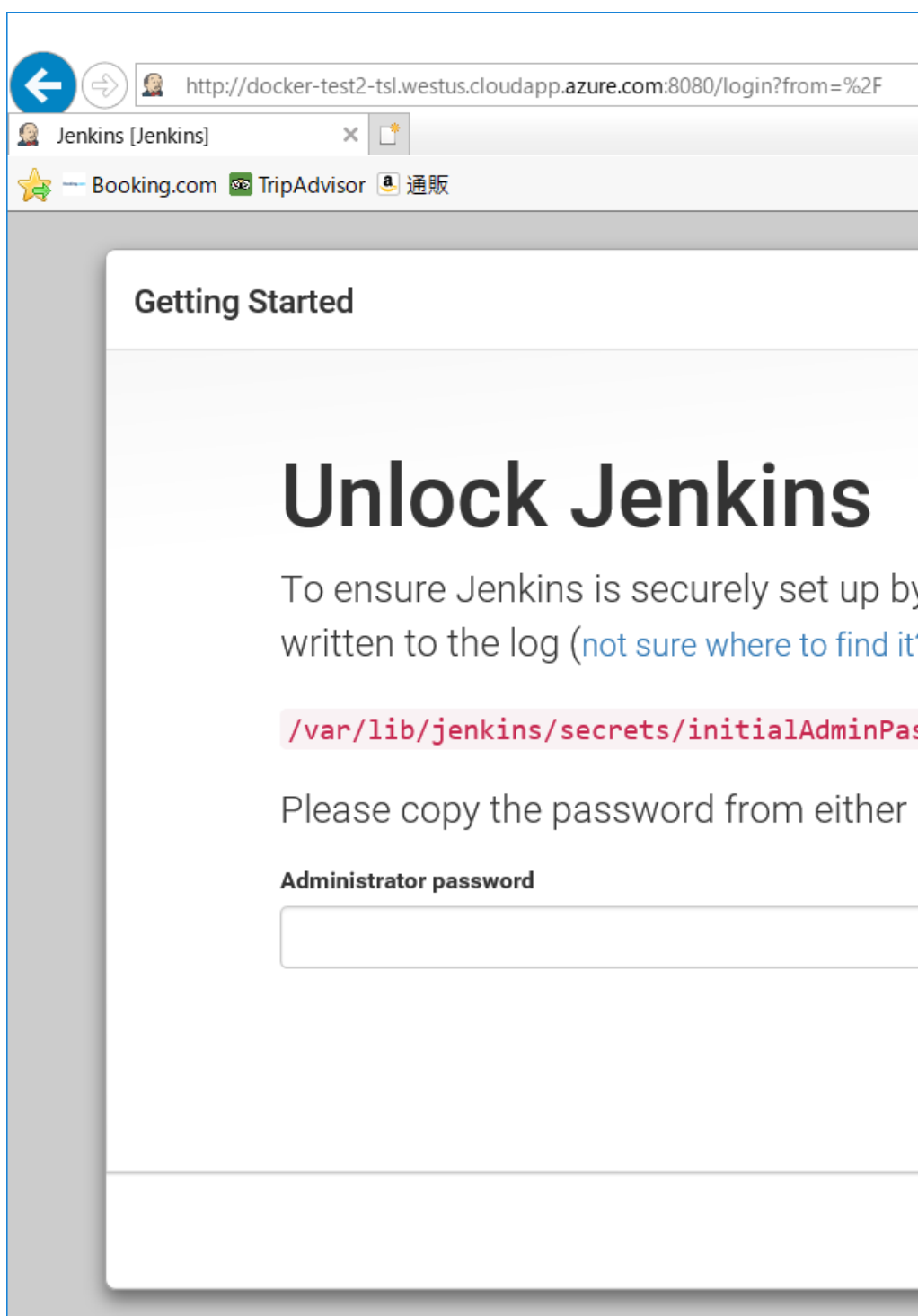
Jenkinsで、リポジトリがPushされたら、テスト環境のコンテナを作って、ビルドして実行。 結果を回収する

Jenkinsをホストへインストール

A terminal window titled "docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT". The window has a menu bar with "ファイル(F)", "編集(E)", "設定(S)", "コントロール(Q)", "ウィンドウ(W)", and "ヘルプ(H)". The command prompt shows "[azureuser@docker-test2 ~]\$ sudo sh jenkins_install.sh". The terminal content is mostly black, indicating the command is running or the output is not visible.

```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ sudo sh jenkins_install.sh
```

ブラウザでアクセス

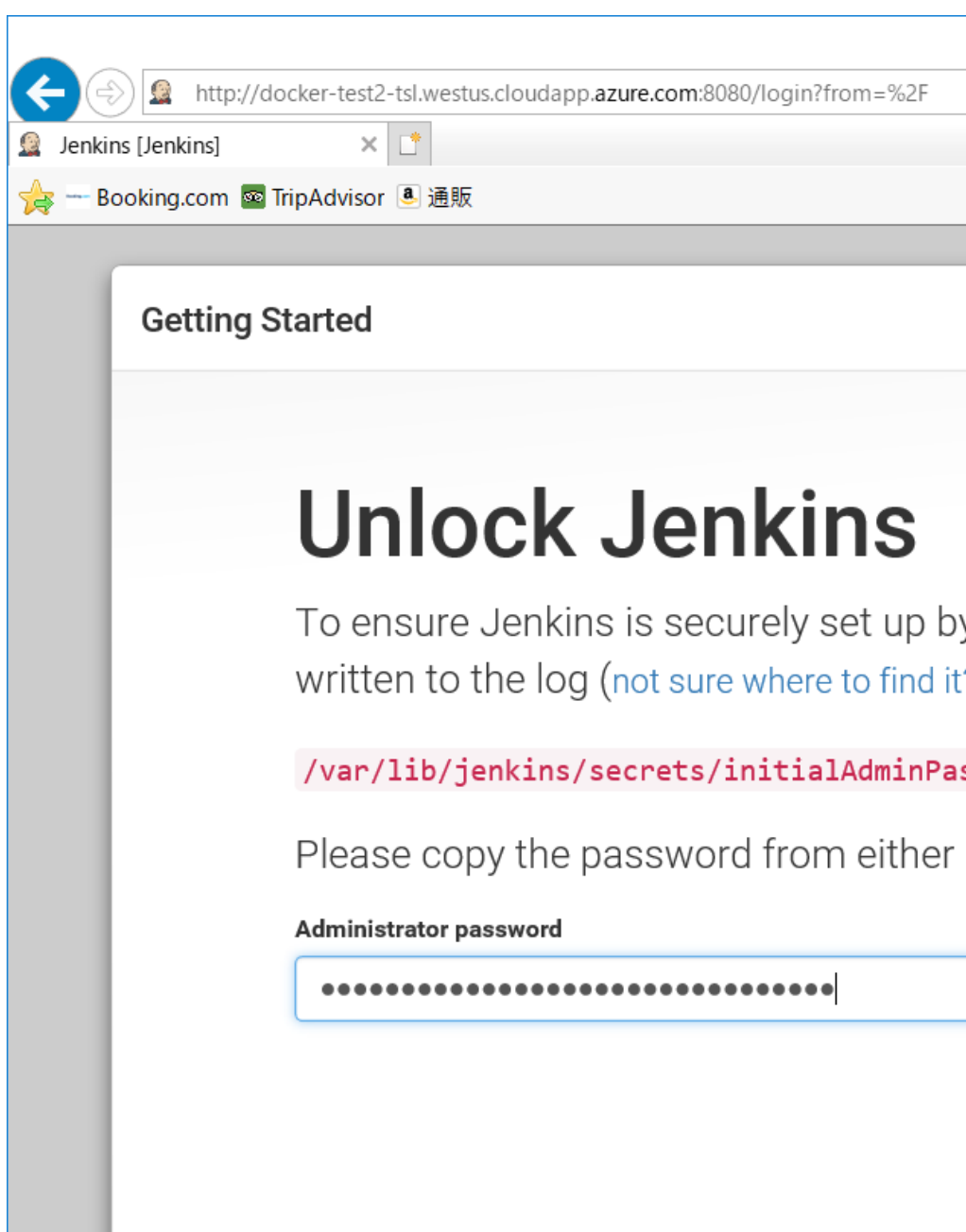


/var/lib/jenkins/secrets/initialAdminPassword
でデフォルトパスワードを確認

```

docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassw
rd
1d454daa33c444a290160adeedb81cc9
[azureuser@docker-test2 ~]$

```



http://docker-test2-tsl.westus.cloudapp.azure.com:8080/j_acegi_security_check

The screenshot shows a web browser window with the address bar displaying `http://docker-test2-tsl.westus.cloudapp.azure.com:8080/`. The browser has a single tab titled 'SetupWizard [Jenkins]'. The page content is titled 'Getting Started' and features a large heading 'Customize Jenki' (likely Jenkins). Below the heading, it says 'Plugins extend Jenkins with additional'. A prominent blue box contains the text 'Install suggested plugins' and 'Install plugins the Jenkins community finds most useful.' At the bottom left of the page, the version 'Jenkins 2.107.1' is displayed.

Getting Started

Customize Jenki

Plugins extend Jenkins with additional

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Jenkins 2.107.1

← → http://docker-test2-tsl.westus.cloudapp.azure.com:8080/

SetupWizard [Jenkins]

★ Booking.com 🗺️ TripAdvisor 📦 通販

Getting Started

Create First Admin User

ユーザー名:

パスワード:

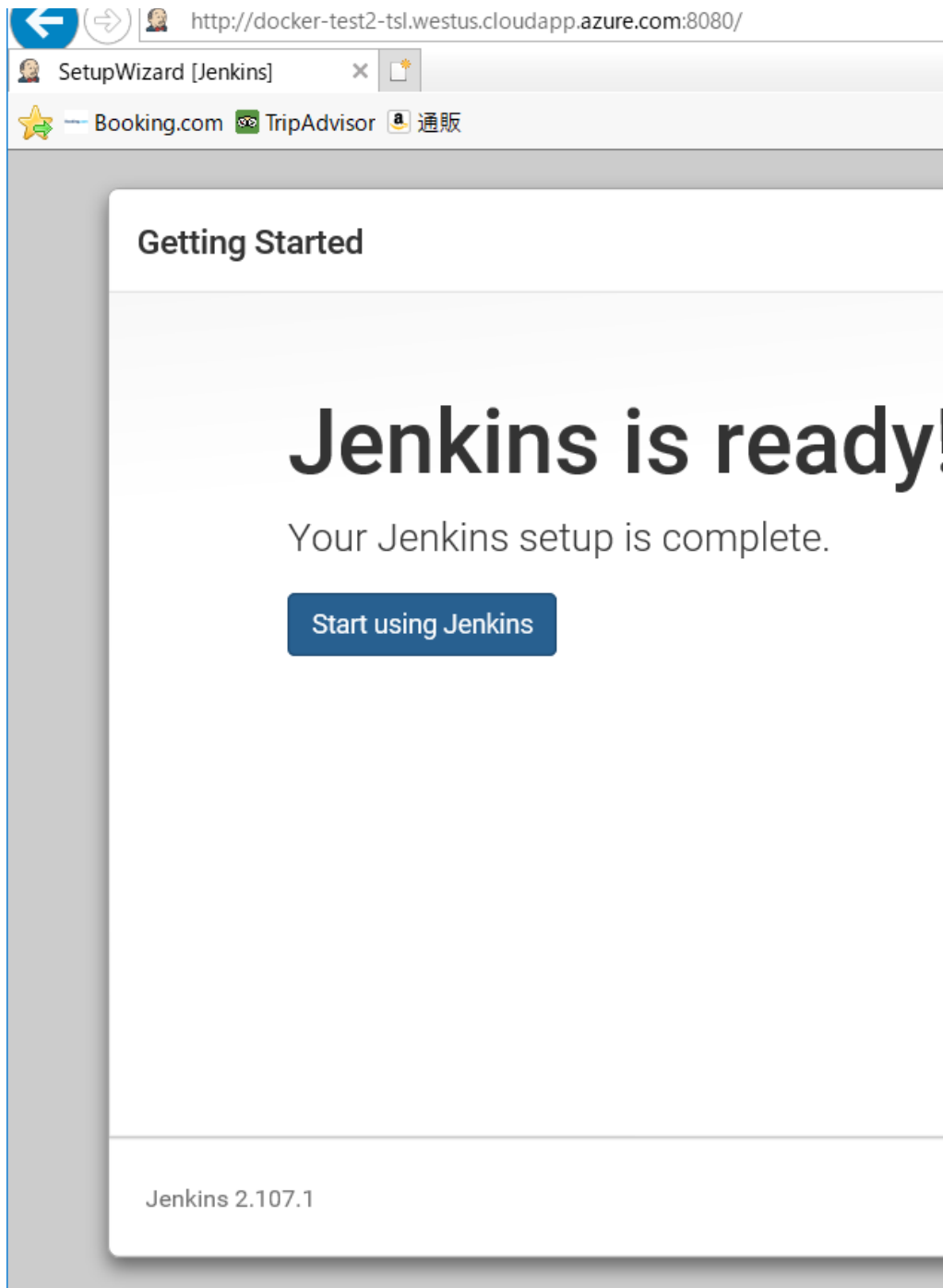
パスワードの確認:

フルネーム:

メールアドレス:

Jenkins 2.107.1

Tsl****&



Enter an item name

docker-test

» Required field

フリースタイル・プロジェクトのビルド
もっとも汎用性の高いJenkinsの中核機能です。システムでプロジェクトがビルドできます。往々にすることができます。

パイプライン
複数のビルドスレーブにまたがる長時間に渡るパイプラインの構築に適しており、フリースタイルです。

マルチ構成プロジェクトのビルド
複数の環境でのテストや、プラットフォームごと適しています。

フォルダ
階層的にアイテムを格納するコンテナを作成し...

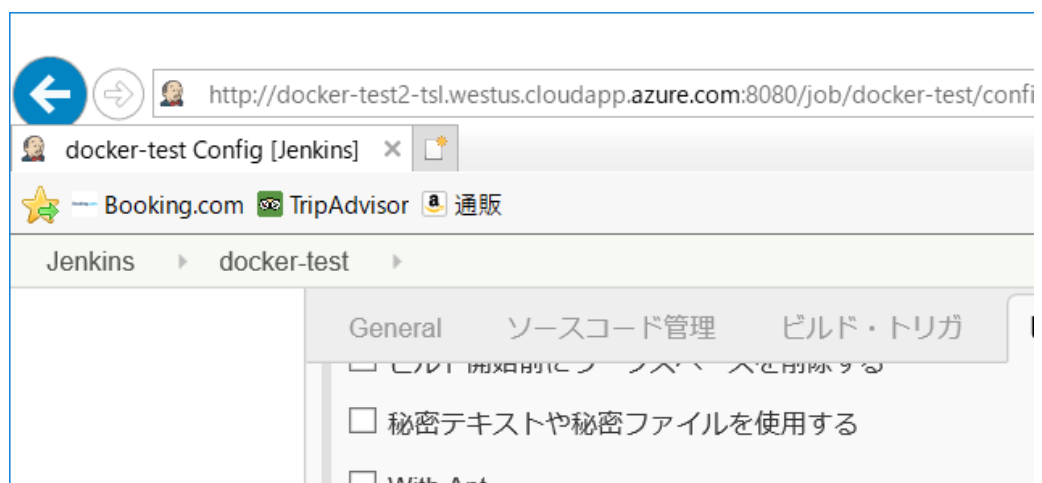
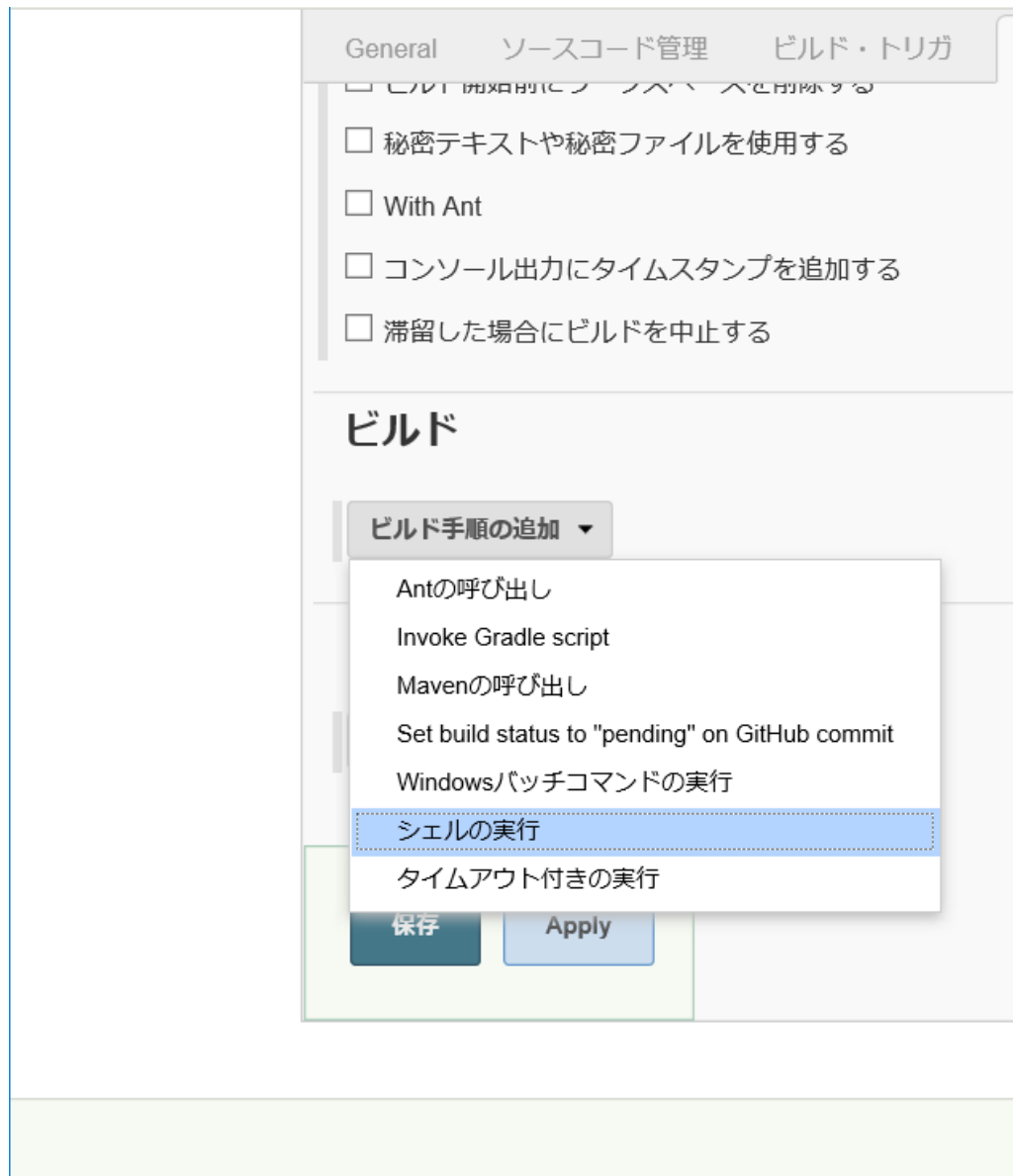
OK

http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/config

docker-test Config [Jenkins]

Jenkins > docker-test >

実行結果 3



実行結果 3

ビルド

コンソール出力にタイムスタンプを追加する

滞留した場合にビルドを中止する

シェルの実行

シェルスクリプト

```
cd /home/azureuser  
make -f Makefile.docker
```

ビルドから利用可能な環境変数の

ビルド手順の追加

保存

Apply

手動でビルド実行

← →  http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/1/console

 docker-test #1 Console [Jen... × 

  Booking.com  TripAdvisor  通販

 **Jenkins**

Jenkins ▶ docker-test ▶ #1

 [プロジェクトへ戻る](#)

 [状態](#)


 [変更履歴](#)


 [コンソール出力](#)


 **コンソール**

Started by user [Admini](#)
Building in workspace

実行結果 3

 プレーンテキスト表示

 説明の編集

 ビルドを削除

```
Building in workspace .
[docker-test] $ /bin/sh
+ cd /home/azureuser
/tmp/jenkins1076826795
Build step 'Execute shell'
Finished: FAILURE
```

jenkinsユーザの権限がないので実行できず

以下の場所に資源を移す

/tmp/jenkins_work/

```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:/tmp VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 jenkins_work]$ pwd
/tmp/jenkins_work
[azureuser@docker-test2 jenkins_work]$ ls -l
total 8
-rw-r--r--. 1 azureuser azureuser 305 Mar 31 14:30 Dockerfile.bu
-rw-r--r--. 1 azureuser azureuser 507 Mar 31 14:36 Makefile.dock
[azureuser@docker-test2 jenkins_work]$ cd ..
[azureuser@docker-test2 tmp]$ ls -l
total 3184
-rw-r--r--. 1 jenkins jenkins 19673 Mar 31 14:17 akuma3258
```

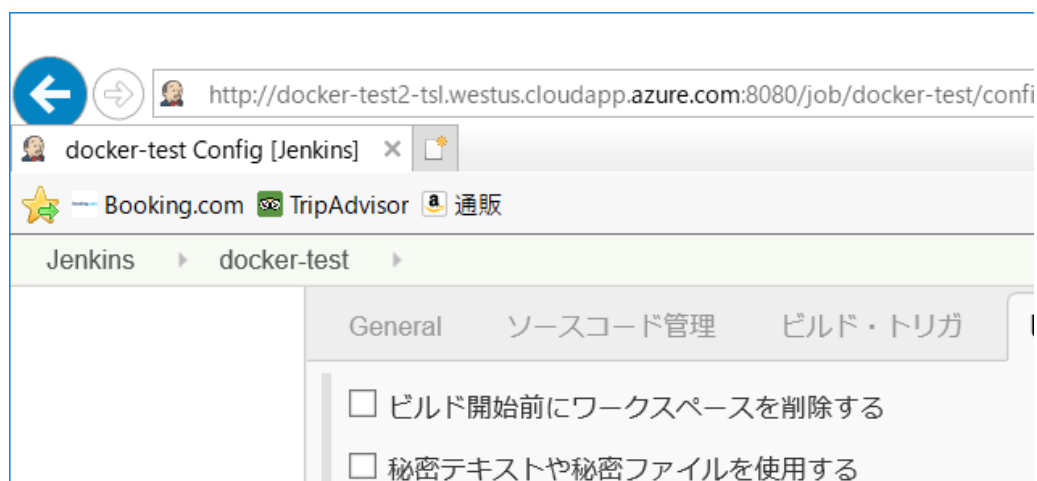
実行結果 3

```

ar
drwxr-xr-x. 2 jenkins jenkins 19 Mar 31 14:17 hsperrfda
drwxr-xr-x. 2 root root 19 Mar 31 14:17 hsperrfda
drwxrwxr-x. 2 azureuser azureuser 53 Mar 31 14:37 jenkins_
drwxr-xr-x. 2 jenkins jenkins 6 Mar 31 14:17 jetty-0.0
any-3655332937199741596.dir
drwxr-xr-x. 2 jenkins jenkins 6 Mar 31 14:17 jna--1712
-rw-r--r--. 1 jenkins jenkins 1137285 Mar 31 14:17 jna879635
-rw-r--r--. 1 jenkins jenkins 2098273 Mar 31 14:17 winstone
24.jar
[azureuser@docker-test2 tmp]$ chown -R jenkins:jenkins jenkins_
chown: changing ownership of 'jenkins_work/Dockerfile.build' :
omitted
chown: changing ownership of 'jenkins_work/Makefile.docker' : O
omitted
chown: changing ownership of 'jenkins_work/' : Operation not pe
[azureuser@docker-test2 tmp]$ sudo chown -R jenkins:jenkins jenk
[sudo] password for azureuser:
[azureuser@docker-test2 tmp]$ ls -l jenkins_work/
total 8
-rw-r--r--. 1 jenkins jenkins 305 Mar 31 14:30 Dockerfile.build
-rw-r--r--. 1 jenkins jenkins 507 Mar 31 14:36 Makefile.docker
[azureuser@docker-test2 tmp]$

```

ジョブを変更



実行結果 3

☐ With Ant

☐ コンソール出力にタイムスタンプを追加する

☐ 滞留した場合にビルドを中止する

ビルド

シェルの実行

シェルスクリプト

```
cd /tmp/jenkins_work  
make -f Makefile.docker
```

[ビルドから利用可能な環境変数の](#)

ビルド手順の追加 ▼

保存

Apply

実行



http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/2/console

docker-test #2 Console [Jen...]

★ Booking.com 🌐 TripAdvisor 📦 通販

 **Jenkins**

Jenkins ▶ docker-test ▶ #2

 プロジェクトへ戻る

 状態

 変更履歴

 **コンソール出力**

 プレーンテキスト表示

 **コンソール**

Started by user [Admini](#)
Building in workspace .
[docker-test] \$ /bin/s
+ cd /tmp/jenkins_work

実行結果 3

 説明の編集

 ビルドを削除

 前のビルド

```
+ make -t Makefile.doc
# テスト環境の元になるイメージ
docker build -t sample
Got permission denied
unix:///var/run/docker
7B%7D&cachefrom=%5B%
5D&cgroupparent=&cpupe
rfile.build&labels=%7B
7D&memory=0&memswap=0&
fd522f1285ed9b33ec79ac
unix /var/run/docker.s
make: *** [base] Error
Build step 'Execute sh
Finished: FAILURE
```

/var/run/docker.sock にjenkinsユーザの実行権がない

sudoパスワードなしに設定

```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:/tmp/jenkins_work...
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
#
# Defaults    env_keep += "HOME"
Defaults     secure_path = /sbin:/bin:/usr/sbin:/usr/bin
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
## Allows members of the 'sys' group to run networking, software
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING
```

```

ATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)    ALL

## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not r
#includedir /etc/sudoers.d

azureuser  ALL=(ALL)    ALL
jenkins    ALL=(ALL)    ALL

:wc

```

実行

← → http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/

docker-test [Jenkins] ×

★ Booking.com 📺 TripAdvisor 📦 通販

Jenkins

Jenkins > docker-test >

📈 [ダッシュボードへ戻る](#)

🔍 [状態](#)

📝 [変更履歴](#)

📁 [ワークスペース](#)

🔄 [ビルド実行](#)

🚫 [プロジェクトの削除](#)

プロジェクト c

📁 [ワークスペース](#)



設定



変更履歴



ビルド履歴

推移



#3

2018/03/31 14:45



#2

2018/03/31 14:40



#1

2018/03/31 14:33



[RSS 全ビルド](#)



[RSS 失敗ビルド](#)

永続リンク

- [最新のビルド \(#2\), 4 分](#)
- [最新の失敗ビルド \(#2\)](#)
- [最新の不成功ビルド \(#](#)
- [最新の完了ビルド \(#2\)](#)

<http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/build?delay=0sec>



<http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/3/console>



docker-test #3 Console [Jen...]



[Booking.com](#)



[TripAdvisor](#)



[通販](#)



Jenkins

Jenkins

docker-test

#3



[プロジェクトへ戻る](#)



[状態](#)



[変更履歴](#)



[コンソール出力](#)



[プレーンテキスト表示](#)



[説明の編集](#)



[ビルドを削除](#)



[前のビルド](#)



コンソール

```
Started by user Admini
Building in workspace .
[docker-test] $ /bin/s
+ cd /tmp/jenkins_work
+ make -f Makefile.doc
# テスト環境の元になるイメージ
sudo docker build -t s
```

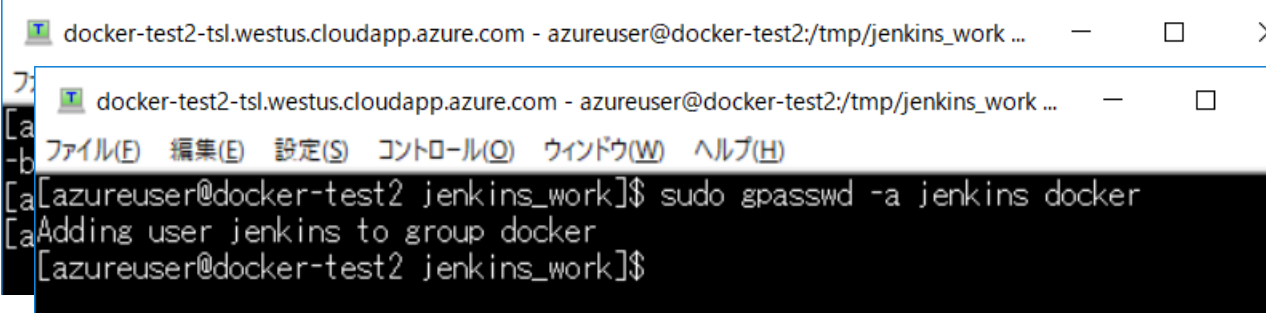
We trust you have received the necessary training for the role of Jenkins Administrator. It usually takes several days to complete this training.

- #1) Respect the privacy of the user.
- #2) Think before you act.
- #3) With great power comes great responsibility.

実行結果 3

```
sudo: no tty present a:  
make: *** [base] Error  
Build step 'Execute sh  
Finished: FAILURE
```

jenkinsをdockerグループへ追加

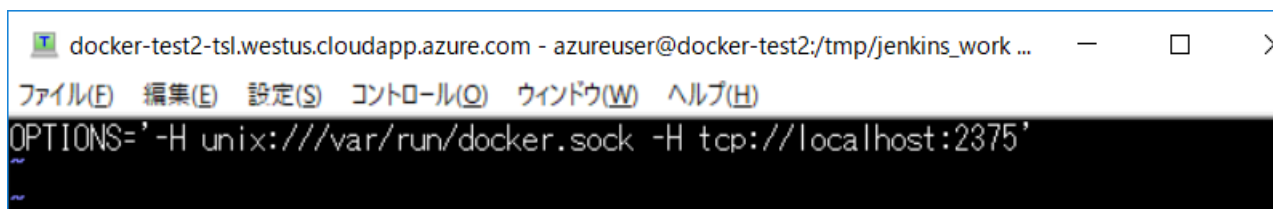


```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:/tmp/jenkins_work ...  
[a  
-b  
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)  
[a[azureuser@docker-test2 jenkins_work]$ sudo gpasswd -a jenkins docker  
[aAdding user jenkins to group docker  
[a[azureuser@docker-test2 jenkins_work]$
```

実行結果 3

参考) <https://techblog.recochoku.jp/1544>


```
sudoedit /etc/sysconfig/docker
```



```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:/tmp/jenkins_work ...
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
OPTIONS='-H unix:///var/run/docker.sock -H tcp://localhost:2375'
```

再起動

```
sudo service docker restart
sudo service jenkins restart
```



```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:/tmp/jenkins_work ...
ファイル(E) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 jenkins_work]$ sudo /etc/init.d/docker start
sudo: /etc/init.d/docker: command not found
[azureuser@docker-test2 jenkins_work]$ sudo service docker restart
Redirecting to /bin/systemctl restart docker.service
[azureuser@docker-test2 jenkins_work]$ sudo service jenkins restart
Restarting jenkins (via systemctl): y [ OK ]
[azureuser@docker-test2 jenkins_work]$
```

再度ビルド実行



docker-test #7 Console [Jen... x]

[★](#)
[Booking.com](#)
[TripAdvisor](#)
[通販](#)

Jenkins

[Jenkins](#)
[docker-test](#)
[#7](#)

[↑ プロジェクトへ戻る](#)

[🔍 状態](#)

[📝 変更履歴](#)

[💻 コンソール出力](#)

[📄 プレーンテキスト表示](#)

[📝 説明の編集](#)

[🚫 ビルドを削除](#)

[← 前のビルド](#)

コンソール

Started by user [Admini](#)
Building in workspace .
[docker-test] \$ /bin/s
+ cd /tmp/jenkins_work
+ make -f Makefile.doc
テスト環境の元になるイメ
docker build -t sample
Sending build context .

Step 1/4 : FROM alpine
3.5: Pulling from libr
550felbea624: Pulling
550felbea624: Verifyin
550felbea624: Download
550felbea624: Pull com
Digest: sha256:9148d06
Status: Downloaded new
---> 6c6084ed97e5

Step 2/4 : RUN apk add
---> Running in 3be45i
fetch <http://dl-cdn.al>
fetch <http://dl-cdn.al>
(1/21) Installing binu
(2/21) Installing binu
(3/21) Installing mp

[←](#)
[→](#)

<http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/7/co>

docker-test #7 Console [Jen... x]

[★](#)
[Booking.com](#)
[TripAdvisor](#)
[通販](#)

[Jenkins](#)
[docker-test](#)
[#7](#)

(18/21) Installing git
(19/21) Installing mus
(20/21) Installing lib
(21/21) Installing mak

Page 39

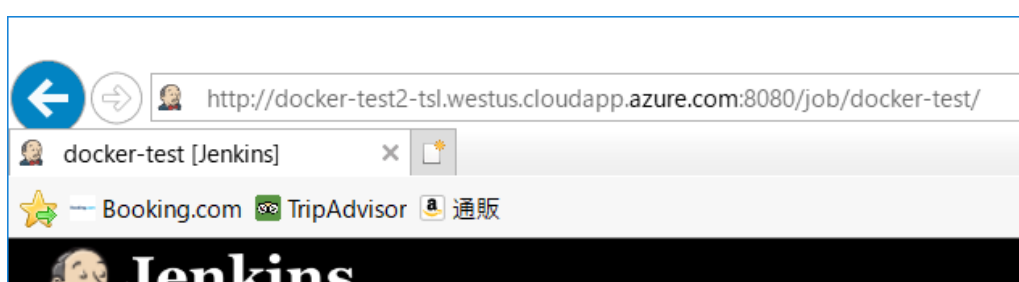
実行結果 3

```
Executing busybox-1.25
Executing ca-certifica
OK: 117 MiB in 32 pack
Removing intermediate ,
---> a7518f23f367
Step 3/4 : RUN git clo:
---> Running in 0cd2c
[91mCloning into 'test
[0mRemoving intermedia
---> 1c39eec82afe
Step 4/4 : WORKDIR tes
Removing intermediate ,
---> 5cde88b3b4b1
Successfully built 5cd
Successfully tagged sa
# テスト環境のコンテナを起
docker run --name samp
Unable to find image '
docker: Error response
not exist or may requi
See 'docker run --help
make: *** [build] Erro
Build step 'Execute sh
Finished: FAILURE
```

今度はイメージ作成まで成功したが
コンテナ起動で、元になるイメージ名がないといわれた

→Makefile.docker が誤っていた

修正後



[ダッシュボードへ戻る](#)

[状態](#)

[変更履歴](#)

[ワークスペース](#)

[ビルド実行](#)

[プロジェクトの削除](#)

[設定](#)

プロジェクト c

[ワークスペース](#)

[変更履歴](#)

ビルド履歴

[推移](#)

find	X
#8	2018/03/31 15:07
#7	2018/03/31 15:03
#6	2018/03/31 14:58
#5	2018/03/31 14:54
#4	2018/03/31 14:52
#3	2018/03/31 14:45
#2	2018/03/31 14:40
#1	2018/03/31 14:33

永続リンク

- [最新のビルド \(#7\), 4 分](#)
- [最新の失敗ビルド \(#7\),](#)
- [最新の不成功ビルド \(#](#)
- [最新の完了ビルド \(#7\),](#)

<http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/8/console>

docker-test #8 Console [Jen...]

[Booking.com](#)
[TripAdvisor](#)
[通販](#)

Jenkins

[Jenkins](#)

[docker-test](#)

[#8](#)

[プロジェクトへ戻る](#)

[状態](#)

コンソール

実行結果 3

 **コンソール出力**

 プレーンテキスト表示



 説明の編集

 ビルドを削除





 前のビルド

Started by user [Admini](#)
 Building in workspace .
 [docker-test] \$ /bin/s
 + cd /tmp/jenkins_work
 + make -f Makefile.doc
 # テスト環境の元になるイメ
 docker build -t sample
 Sending build context .

Step 1/4 : FROM alpine
 ---> 6c6084ed97e5
 Step 2/4 : RUN apk add
 ---> Running in b8227
 fetch <http://dl-cdn.al>
 fetch <http://dl-cdn.al>
 (1/21) Installing binu
 (2/21) Installing binu
 (3/21) Installing gmp
 (4/21) Installing isl
 (5/21) Installing libg
 (6/21) Installing liba
 (7/21) Installing pkgc
 (8/21) Installing libg
 (9/21) Installing mpfr
 (10/21) Installing mpc
 (11/21) Installing lib
 (12/21) Installing gcc
 (13/21) Installing ca-
 (14/21) Installing lib
 (15/21) Installing lib



<http://docker-test2-tsl.westus.cloudapp.azure.com:8080/job/docker-test/8/cons>

docker-test #8 Console [Jen... x]





 通販

Jenkins > docker-test > #8

(16/21) Installing exp
 (17/21) Installing pcr
 (18/21) Installing git
 (19/21) Installing mus
 (20/21) Installing lib
 (21/21) Installing mak
 Executing busybox-1.25

実行結果 3

```
Executing ca-certifica
OK: 117 MiB in 32 pack
Removing intermediate
---> 220c7ed8ec5f
Step 3/4 : RUN git clo
---> Running in 2f6dd
[91mCloning into 'test
[0mRemoving intermedia
---> 61ba3dfa5f37
Step 4/4 : WORKDIR tes
Removing intermediate
---> 0be3ffc505b0
Successfully built 0be
Successfully tagged sa
# テスト環境のコンテナを起
docker run --name samp
gcc -o hello.exe hello
/test1/hello.exe /test
# 実行結果は、/tmp/ にコ
cp -f ./result.txt /tm
# 実行結果をホスト側に回収
docker cp sample-conta
# テスト環境のコンテナは破
docker rm sample-conta
sample-container
Finished: SUCCESS
```

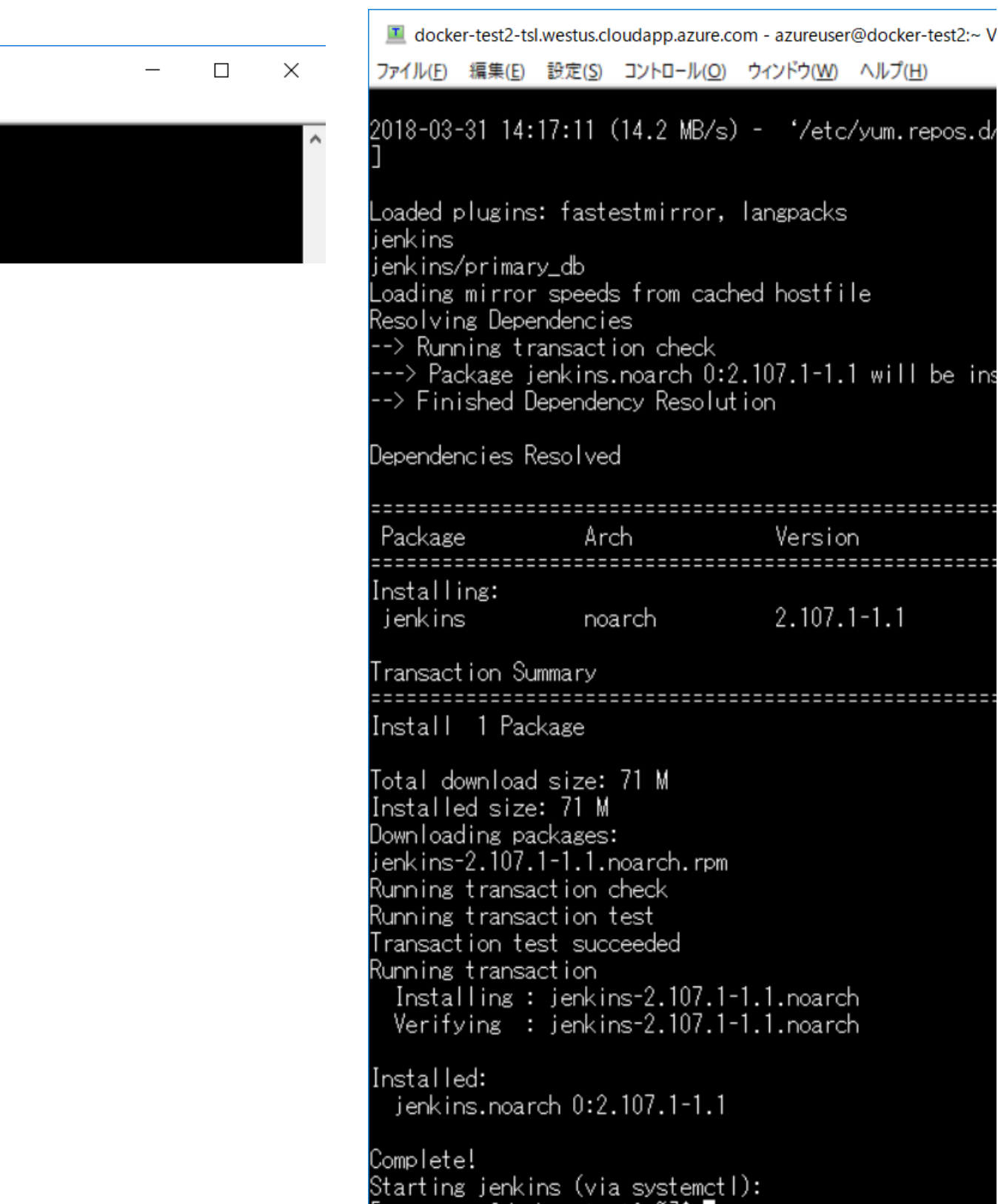
成功！

結果

```
docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:/tmp/jenkins_work ...
ファイル(E) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 jenkins_work]$ ls -lrt
total 12
-rw-r--r--. 1 jenkins docker 305 Mar 31 14:30 Dockerfile.build
-rw-r--r--. 1 jenkins docker 508 Mar 31 15:05 Makefile.docker
-rw-r--r--. 1 jenkins jenkins 37 Mar 31 15:08 result.txt
[azureuser@docker-test2 jenkins_work]$ cat result.txt
[2018/03/31 15:08:11] Good Morning!!
[azureuser@docker-test2 jenkins_work]$
```

```
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[azureuser@docker-test2 jenkins_work]$ ls -lrt
total 12
-rw-r--r--. 1 jenkins docker 305 Mar 31 14:30 Dockerfile.build
-rw-r--r--. 1 jenkins docker 508 Mar 31 15:05 Makefile.docker
-rw-r--r--. 1 jenkins jenkins 37 Mar 31 15:08 result.txt
[azureuser@docker-test2 jenkins_work]$ cat result.txt
[2018/03/31 15:08:11] Good Morning!!
[azureuser@docker-test2 jenkins_work]$
```

コンパイルされて、結果が回収できた！



```

docker-test2-tsl.westus.cloudapp.azure.com - azureuser@docker-test2:~ V
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)

2018-03-31 14:17:11 (14.2 MB/s) - '/etc/yum.repos.d/
]

Loaded plugins: fastestmirror, langpacks
jenkins
jenkins/primary_db
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
---> Package jenkins.noarch 0:2.107.1-1.1 will be ins
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package            Arch           Version
=====
Installing:
jenkins            noarch         2.107.1-1.1

Transaction Summary
=====
Install 1 Package

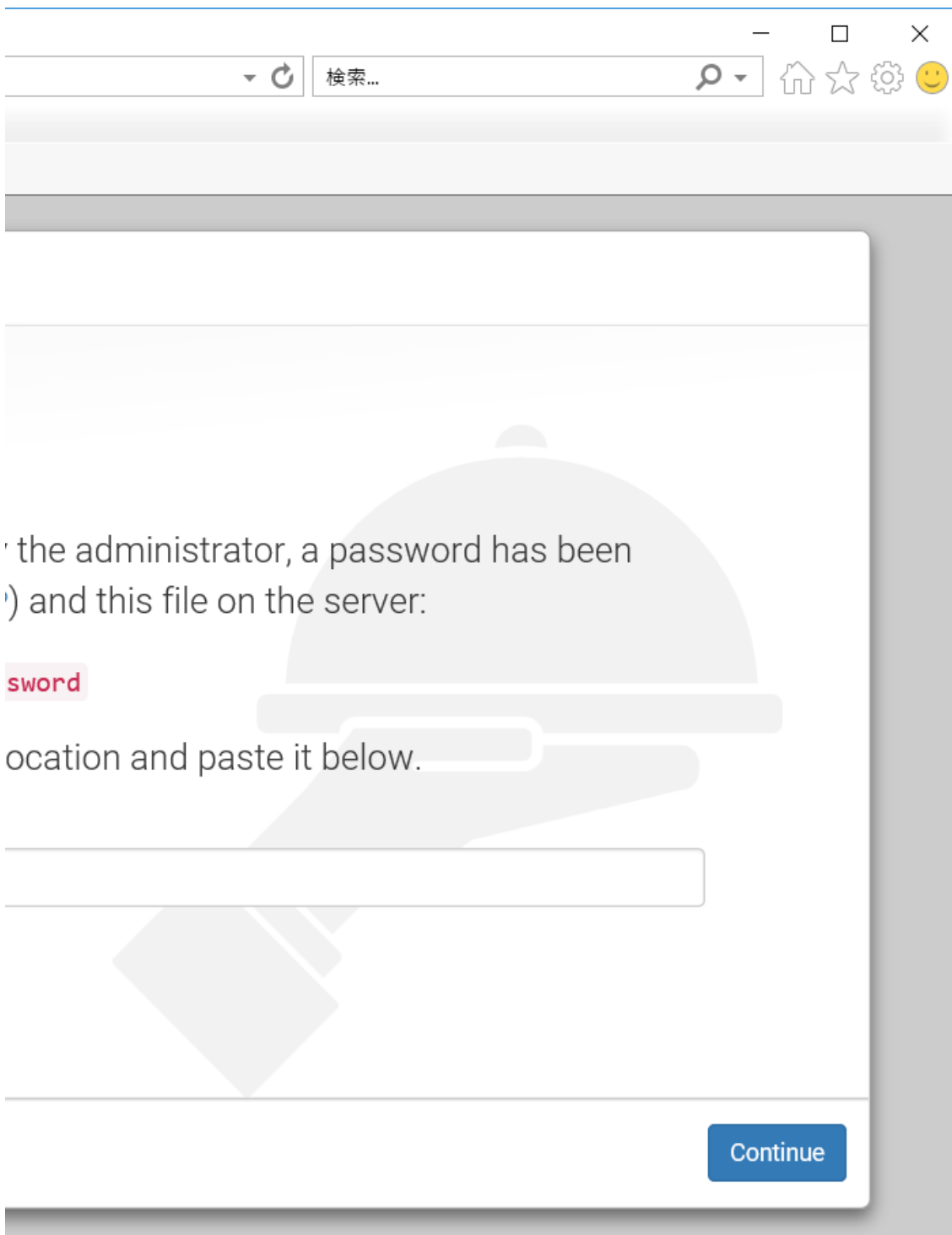
Total download size: 71 M
Installed size: 71 M
Downloading packages:
jenkins-2.107.1-1.1.noarch.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : jenkins-2.107.1-1.1.noarch
  Verifying  : jenkins-2.107.1-1.1.noarch

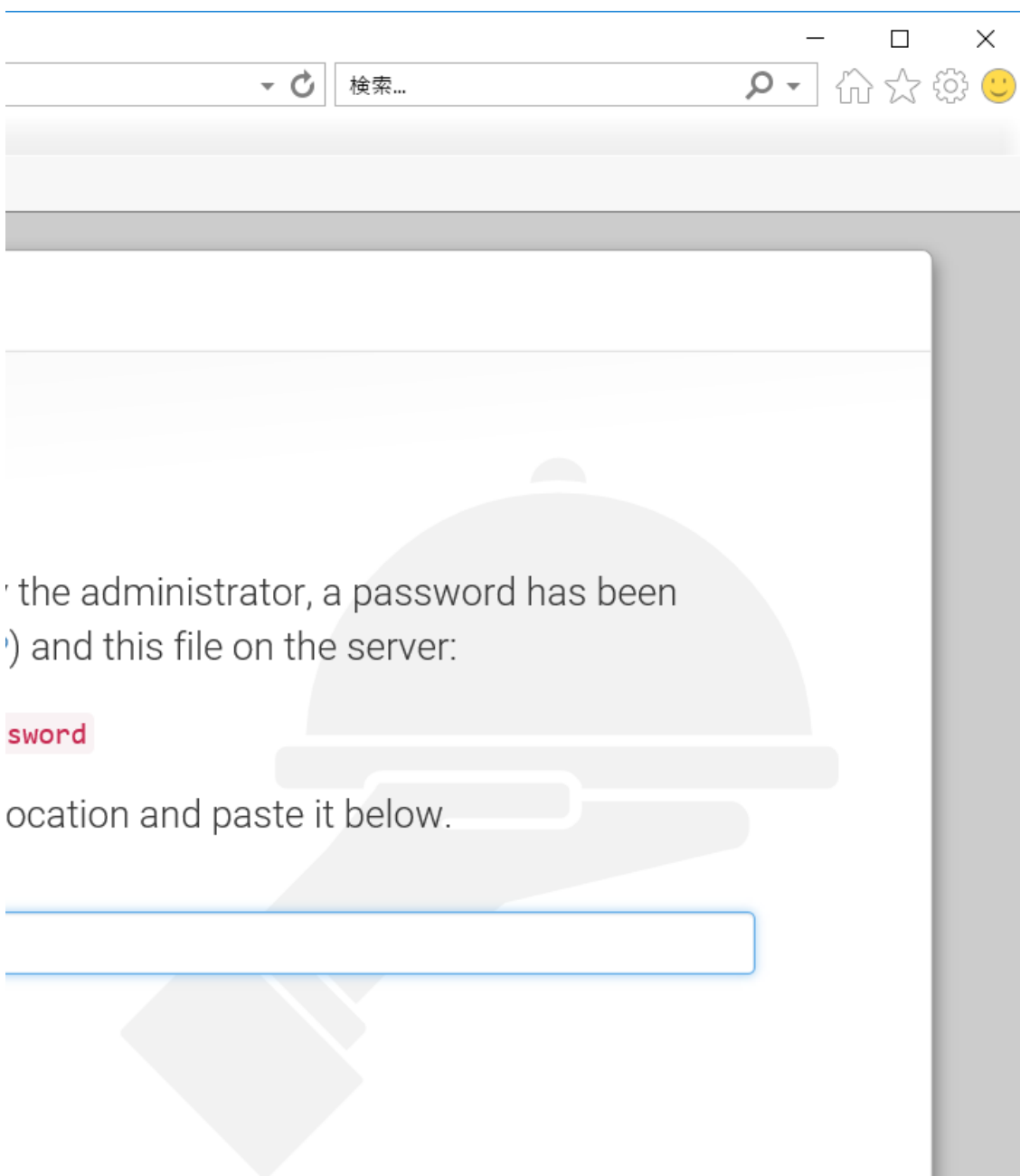
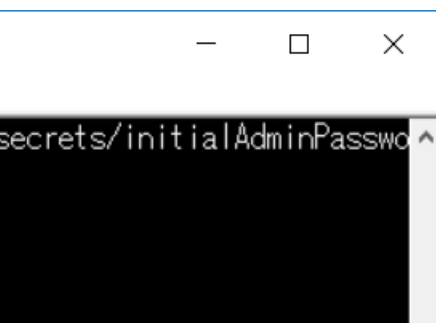
Installed:
jenkins.noarch 0:2.107.1-1.1

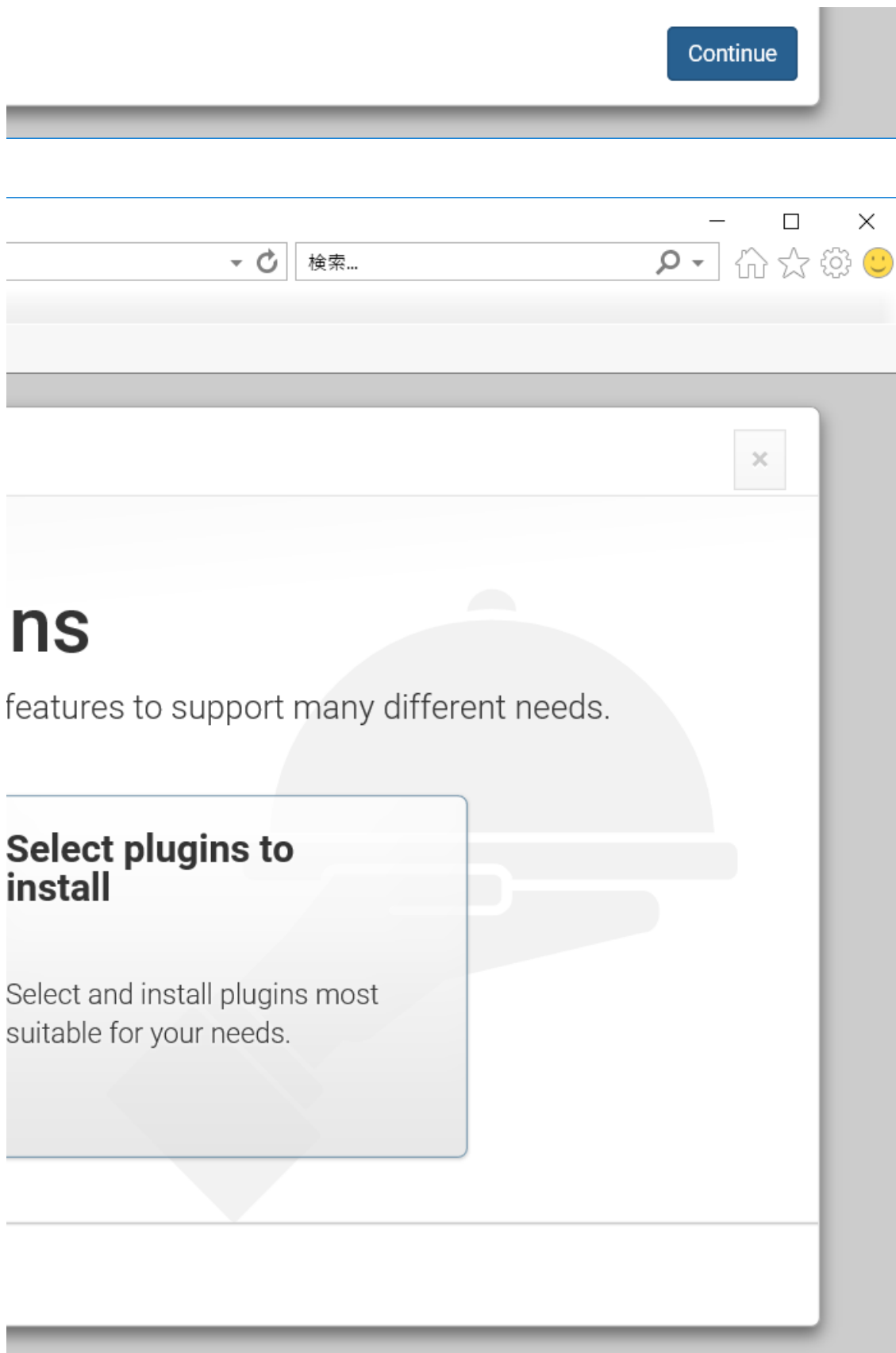
Complete!
Starting jenkins (via systemctl):

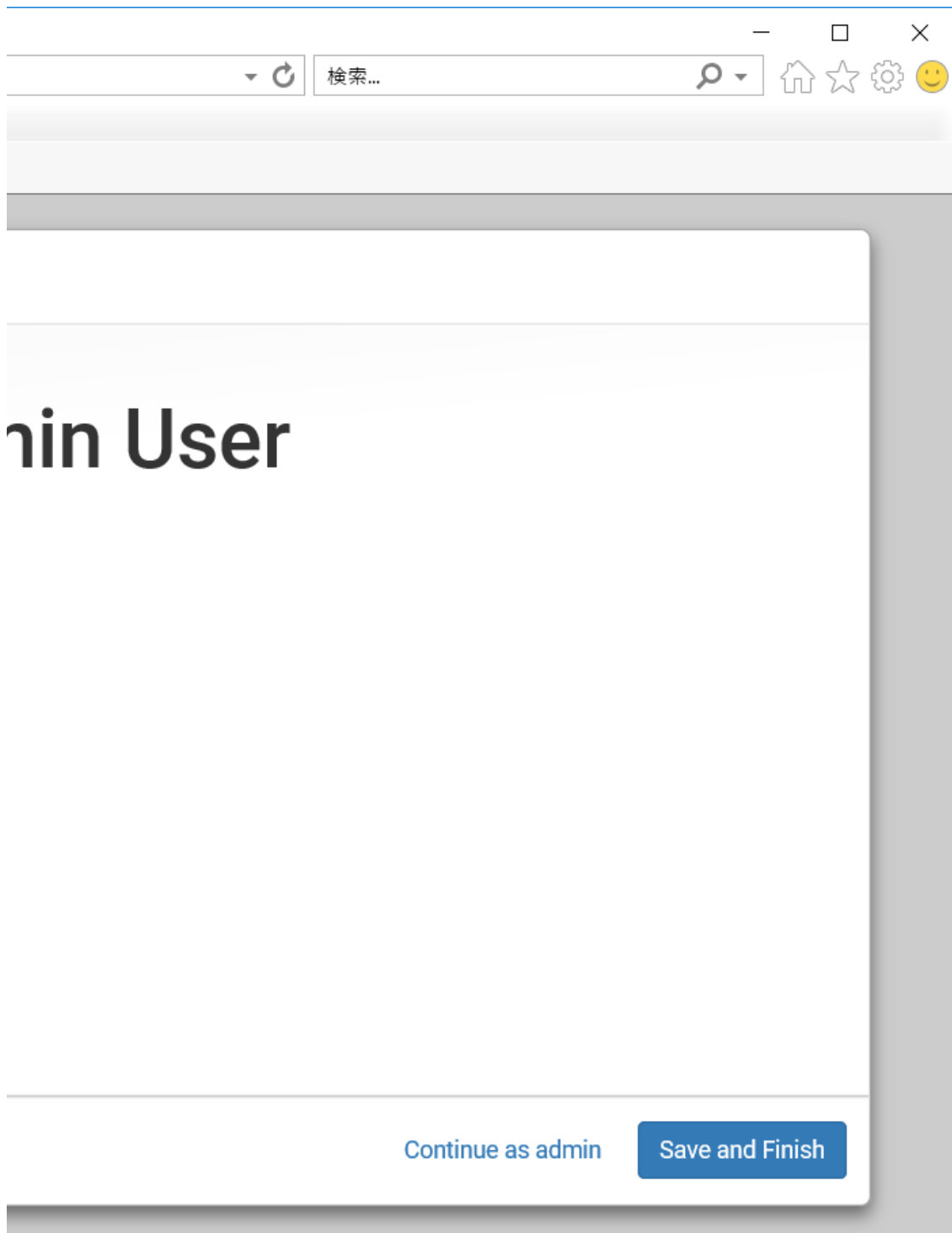
```

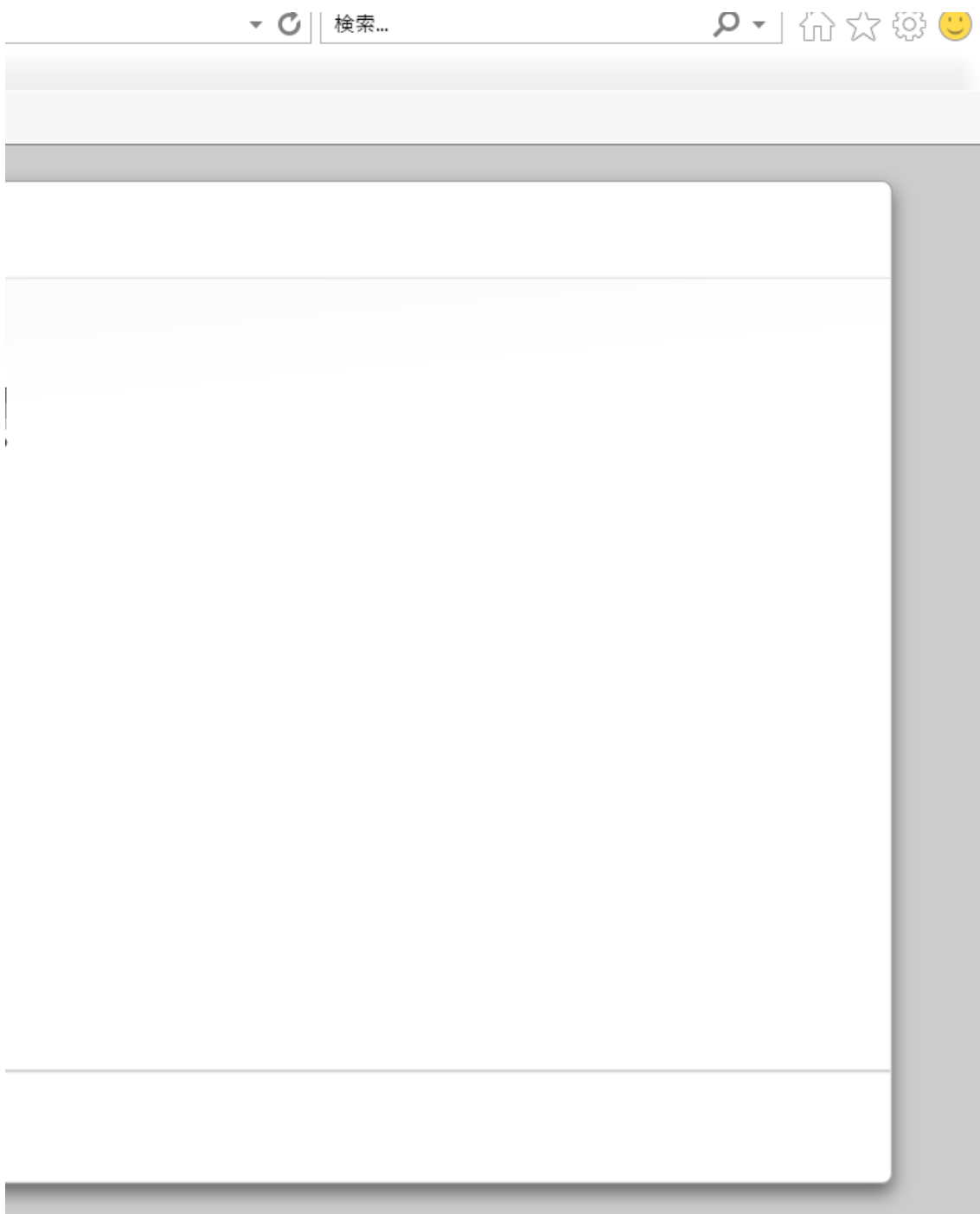
```
lazureuser@docker-test2 ~$
```

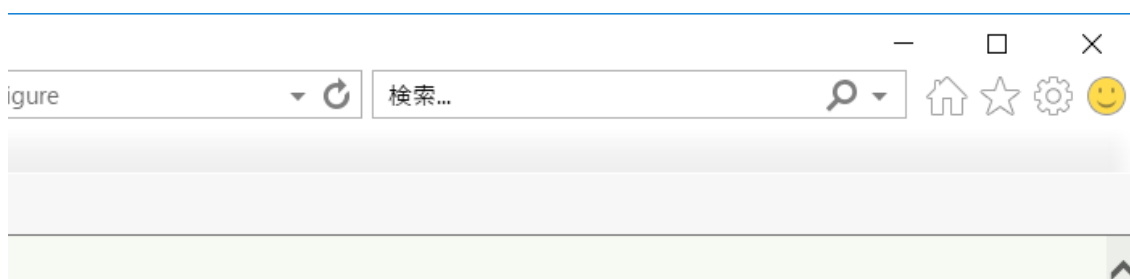












実行結果 3

ビルド環境

ビルド

ビルド後の処理

?

?

ページ更新時: 2018/03/31 14:26:46 UTC [REST API](#) [Jenkins ver. 2.107.1](#)

figure

検索...

?

?

ビルド環境

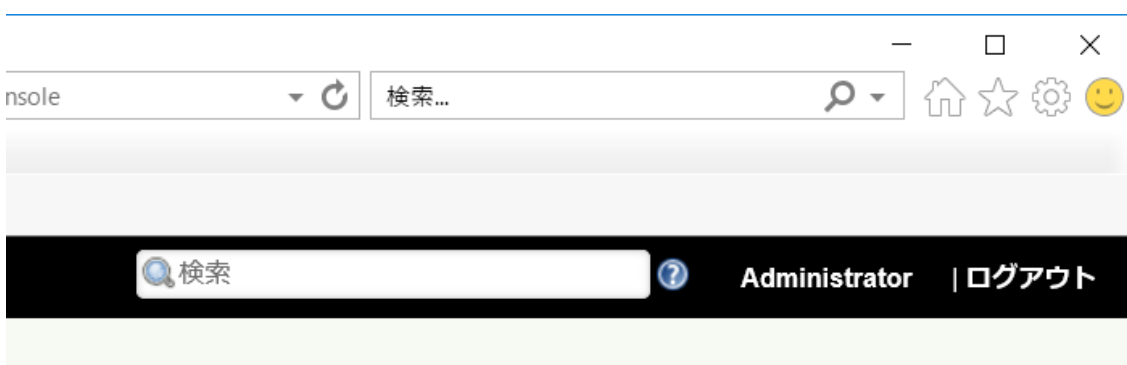
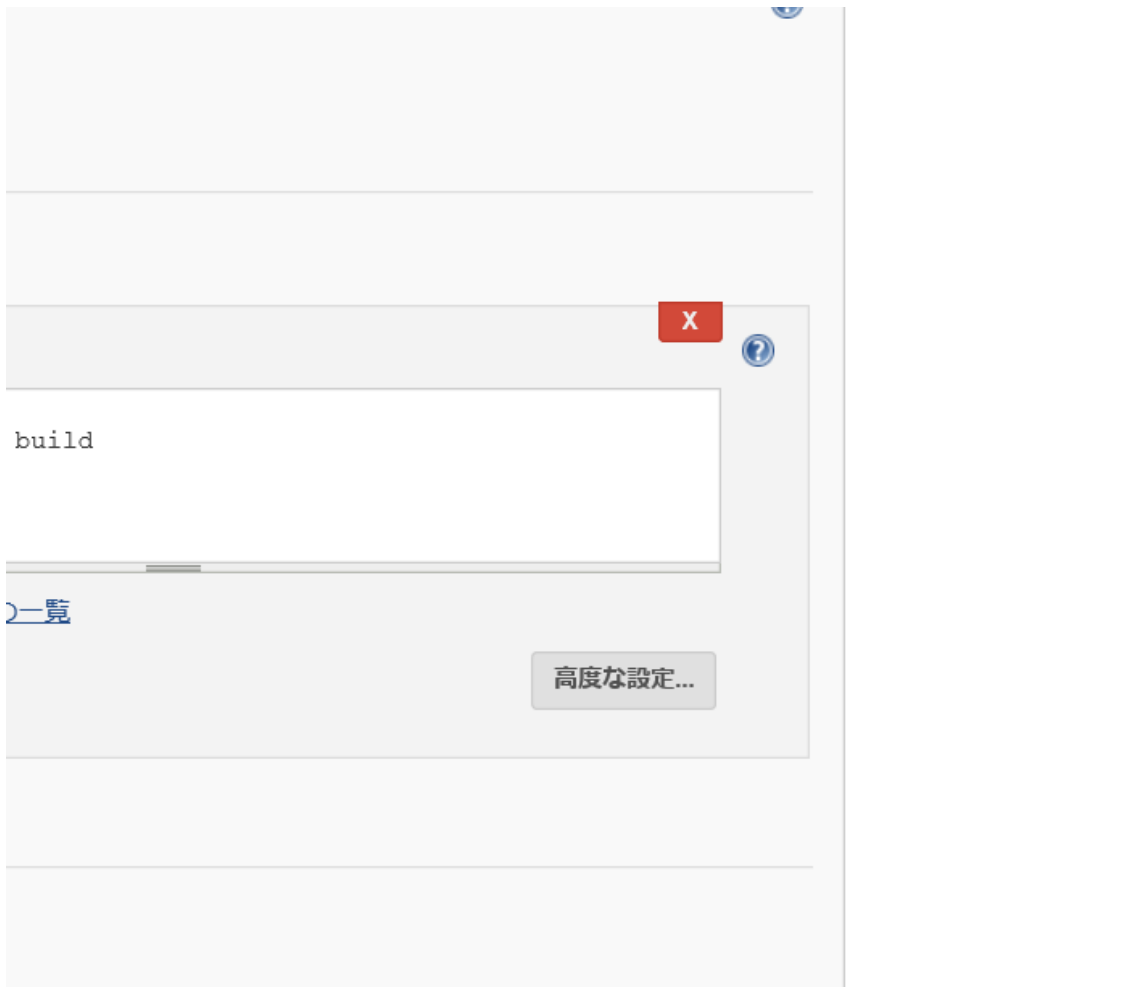
ビルド

ビルド後の処理

?

?

実行結果 3



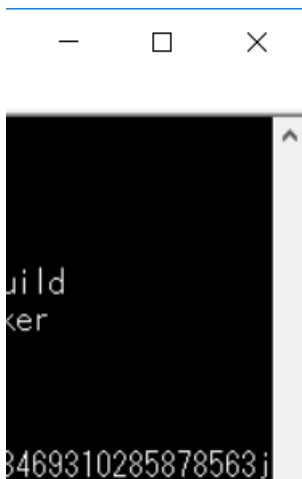
出力

[strator](#)
/var/lib/jenkins/workspace/docker-test

実行結果 3

```
/usr/bin/jenkins, workspace, agent 0000  
h -xe /tmp/jenkins1076826795205876371.sh  
  
205876371.sh: line 2: cd: /home/azureuser: Permission denied  
ell' marked build as failure
```

ページ更新時: 2018/03/31 14:33:23 UTC [REST API](#) [Jenkins ver. 2.107.1](#)

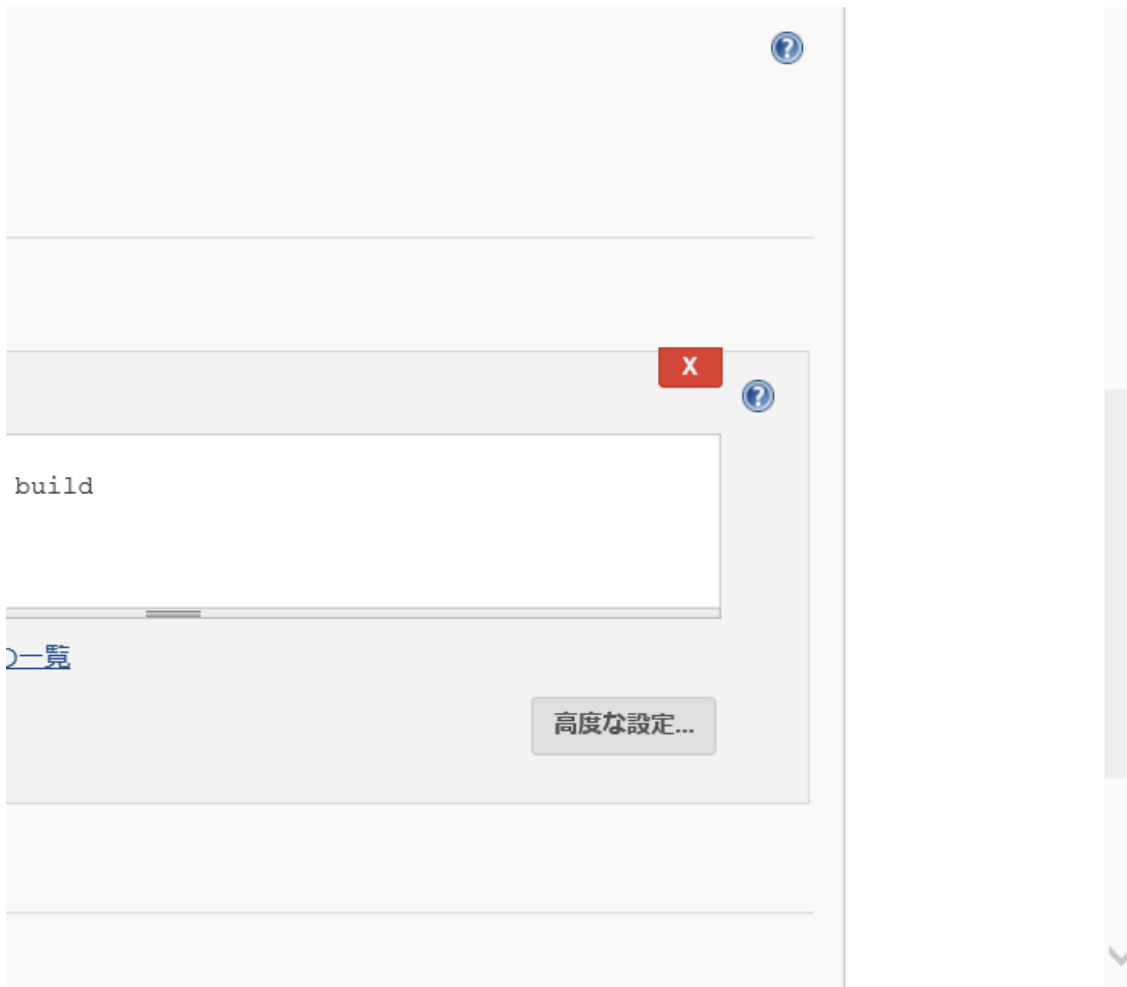


実行結果 3

```
ta_jenkins
ta_root
work
0.0.0-8080-war_-
2433994
55868910573556jar
12449255708192562
work/
Operation not per
Operation not per
mitted
kins_work/
```



実行結果 3



出力

```
strator  
/var/lib/jenkins/workspace/docker-test  
h -xe /tmp/jenkins1273725103116287739.sh
```


実行結果 3

```
ker build
ージを作成
-jenkins -f Dockerfile.build .
while trying to connect to the Docker daemon socket at
.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.37/build?buildargs=%
riod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=Docke
%
networkmode=default&rm=1&session=e0a79aeef353588ea6c77a6e093771fcc5704
b2457&shmsize=0&t=sample-jenkins&target=&ulimits=null: dial
ock: connect: permission denied
1
ell' marked build as failure
```

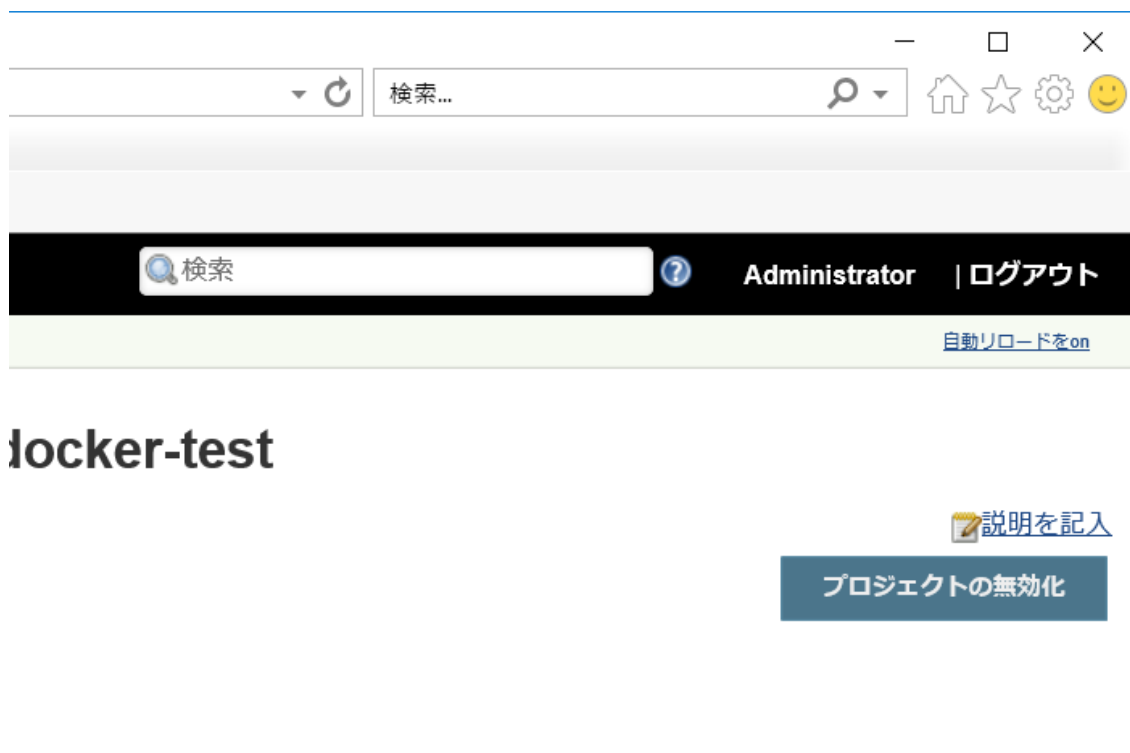
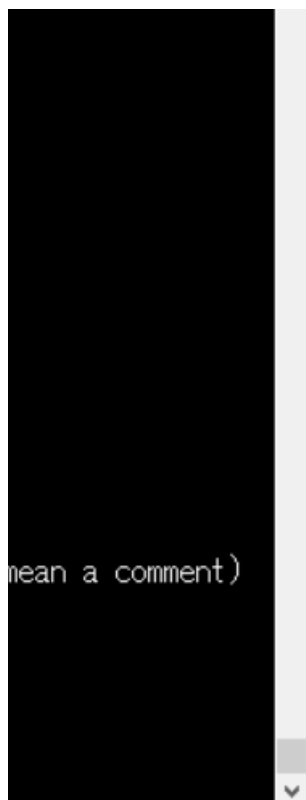
ページ更新時: 2018/03/31 14:40:21 UTC [REST API](#) [Jenkins ver. 2.107.1](#)

dockerを起動するMakefileでは、sudoをつけておく

```

1
2 base:↓
3     # テスト環境の元になるイメージを作成↓
4     sudo docker build -t sample-jenkins -f Dockerfile.build
5
6 build: base↓
7     # テスト環境のコンテナを起動↓
8     sudo docker run --name sample-container sample-docker ma
9
10    # 実行結果をホスト側に回収する↓
11    sudo docker cp sample-container:/tmp/result.txt /tmp/jen
12
13    # テスト環境のコンテナは破棄↓
14    sudo docker rm sample-container↓
15
16 clean:↓
17    # 元になるイメージを削除↓
18    sudo docker rmi sample-jenkins↓
19 [EOF]

```



[45 秒前](#)
[4 分 45 秒前](#)
[2\), 4 分 45 秒前](#)
[4 分 45 秒前](#)

ページ更新時: 2018/03/31 14:45:00 UTC [REST API](#) [Jenkins ver. 2.107.1](#)



出力

```
strator
/var/lib/jenkins/workspace/docker-test
h -xe /tmp/jenkins5366377854486944968.sh

ker build
ージを作成
ample-jenkins -f Dockerfile.build .

ived the usual lecture from the local System
lly boils down to these three things:

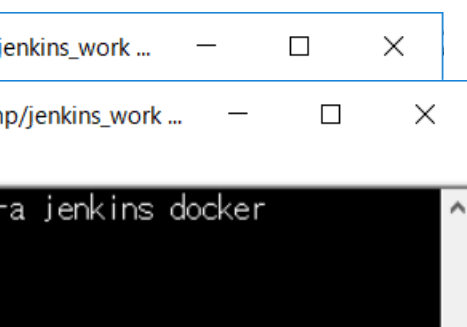
ivacy of others.
ou type.
er comes great responsibility.
```

実行結果 3

```
or some great responsibility.

nd no askpass program specified
1
ell' marked build as failure
```

ページ更新時: 2018/03/31 14:46:16 UTC [REST API](#) [Jenkins ver. 2.107.1](#)

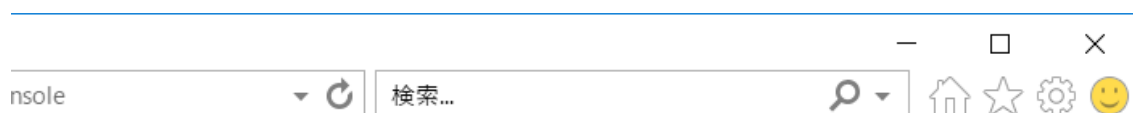
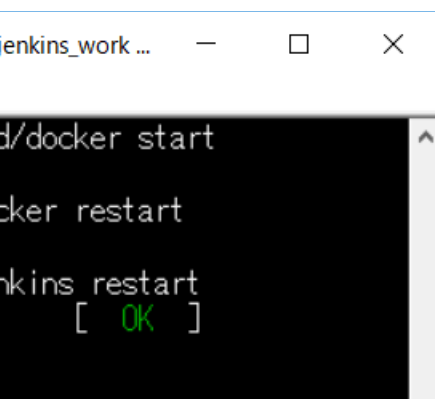
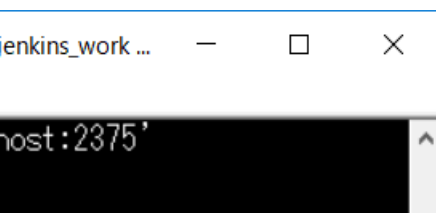


A screenshot of the Jenkins web interface in a browser. The address bar shows 'http://docker-test2-tsl.westus.cloudapp.azure.com:8080/'. The browser tabs include 'docker-test #6 Console [Jen...'. The page header features the Jenkins logo and the text 'Jenkins'. Below the header, a breadcrumb trail shows 'Jenkins > docker-test > #6'. The main content area has a sidebar with several links: 'プロジェクトへ戻る', '状態', '変更履歴', 'コンソール出力', 'プレーンテキスト表示', '説明の編集', 'ビルドを削除', and '前のビルド'. On the right side, there is a red circular icon and a list of build details including 'Star', 'Buil', '[doc', '+ cd', '+ ma', '# テ', 'dock', 'Got', 'unix', '7B%7', '5D&c', 'rfil', '7D&m', 'e077', and 'unix'.

実行結果 3

```
make
Buil
Fini
```

変わらずパーミッションエラー



Administrator | ログアウト

出力

```

strator
/var/lib/jenkins/workspace/docker-test
h -xe /tmp/jenkins3018719518069718218.sh

ker build
ージを作成
-jenkins -f Dockerfile.build .
to Docker daemon 3.072kB

:3.5
ary/alpine
fs layer
g Checksum
complete
plete
9e50eee519ec45e5683e56a1c217b61a52ed90eb77bdce674cc212f1e
er image for alpine:3.5

--no-cache gcc libc-dev make git
b5a8f27
pinelinux.org/alpine/v3.5/main/x86_64/APKINDEX.tar.gz
pinelinux.org/alpine/v3.5/community/x86_64/APKINDEX.tar.gz
tils-libs (2.27-r1)
tils (2.27-r1)
(6.1.1-r0)

```

console

🏠 ☆ ⚙️ 😊

```

(2.11.3-r0)
l-dev (1.1.15-r8)
c-dev (0.7-r1)
e (4.2.1-r0)

```

実行結果 3

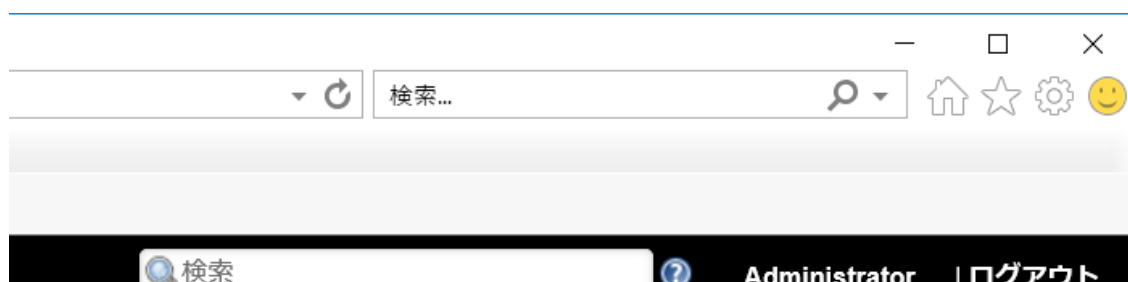
```
.1-r1.trigger
tes-20161130-r1.trigger
ages
container 3be45b5a8f27

ne --depth 1 https://github.com/hazuki3936/test1.git
4567f4a
1'...
te container 0cd2c4567f4a

t1
container d6c0c4049308

e88b3b4b1
mple-jenkins:latest
動
le-container sample-docker make -f Makefile.app
sample-docker:latest' locally
  from daemon: pull access denied for sample-docker, repository does
re 'docker login'.
'.
r 125
ell' marked build as failure
```

ページ更新時: 2018/03/31 15:04:15 UTC [REST API](#) [Jenkins ver. 2.107.1](#) ✓



自動リロードをon

locker-test

説明を記入

プロジェクトの無効化

5 秒 前

4 分 5 秒 前

7), 4 分 5 秒 前

4 分 5 秒 前

console

検索...

検索

?

Administrator

ログアウト

出力

実行結果 3

```

strator
/var/lib/jenkins/workspace/docker-test
h -xe /tmp/jenkins1472754461647161478.sh

ker build
ージを作成
-jenkins -f Dockerfile.build .
to Docker daemon 3.072kB

:3.5

--no-cache gcc libc-dev make git
c977012
pinelinux.org/alpine/v3.5/main/x86_64/APKINDEX.tar.gz
pinelinux.org/alpine/v3.5/community/x86_64/APKINDEX.tar.gz
tils-libs (2.27-r1)
tils (2.27-r1)
(6.1.1-r0)
(0.17.1-r0)
omp (6.2.1-r1)
tomic (6.2.1-r1)
onf (1.0.2-r0)
cc (6.2.1-r1)
3 (3.1.5-r0)
1 (1.0.3-r0)
stdc++ (6.2.1-r1)
(6.2.1-r1)
certificates (20161130-r1)
ssh2 (1.7.0-r2)
curl (7.59.0-r0)

```

nsol

検索...

at (2.2.0-r1)

e (8.39-r0)

(2.11.3-r0)

l-dev (1.1.15-r8)

c-dev (0.7-r1)

e (4.2.1-r0)

.1-r1.trigger

実行結果 3

```
tes-20161130-r1.trigger
ages
container b8227c977012

ne --depth 1 https://github.com/hazuki3936/test1.git
4bc85a5
1'...
te container 2f6dd4bc85a5

t1
container bdecbfd7bf14

3ffc505b0
mple-jenkins:latest
動
le-container sample-jenkins make -f Makefile.app
.c
1/result.txt
ピーしておく
p/result.txt
する
iner:/tmp/result.txt /tmp/jenkins_work/result.txt
棄
iner
```

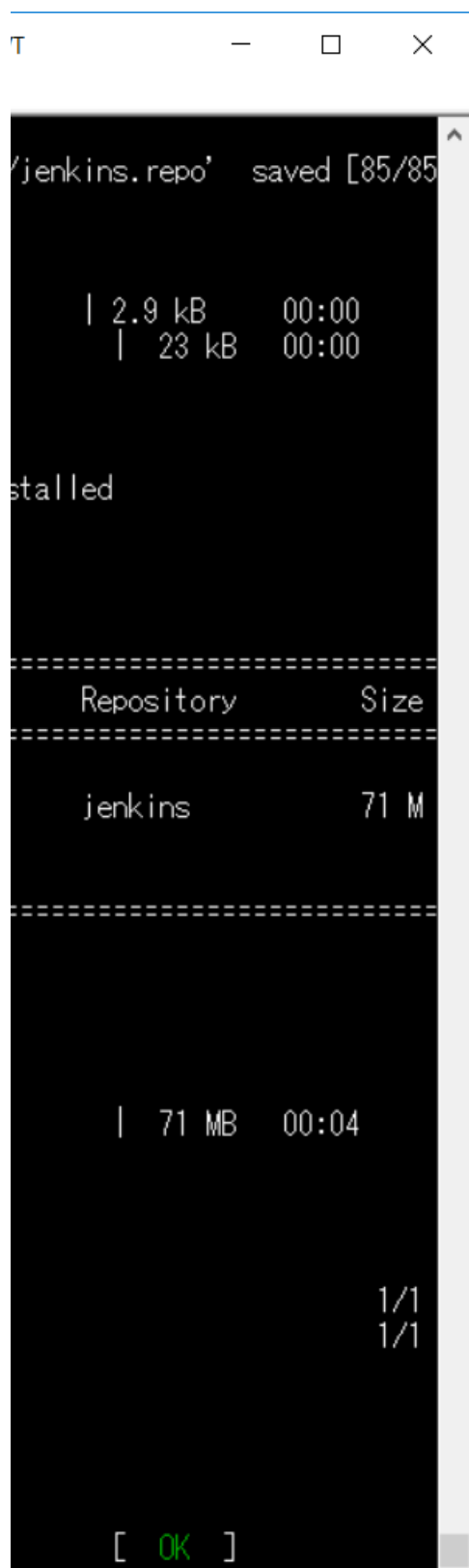
ページ更新時: 2018/03/31 15:08:25 UTC [REST API](#) [Jenkins ver. 2.107.1](#) ▼

jenkins_work ...

ile.build
e.docker
txt

```
ile.build  
e.docker  
txt
```

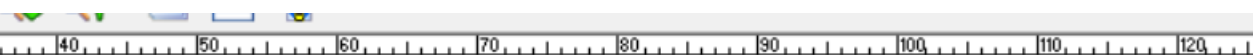
実行結果 3



実行結果 3



実行結果 3



```
シを作成↓  
jenkins -f Dockerfile.build .↓  
  
↓  
e-container sample-docker make -f Makefile.app↓  
  
る↓  
ner:/tmp/result.txt /tmp/jenkins_work/result.txt↓  
  
↓  
ner↓  
  
ns↓
```



コンソール出力

```
ted by user Administrator
ding in workspace /var/lib/jenkins/workspace/docker-test
ker-test] $ /bin/sh -xe /tmp/jenkins2378374155732188768.sh
./tmp/jenkins_work
ke -f Makefile.docker build
スト環境の元になるイメージを作成
er build -t sample-jenkins -f Dockerfile.build .
permission denied while trying to connect to the Docker daemon socket at
:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.37/build?
D&cachefrom=%5B%
groupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dock
e.build&labels=%7B%
emory=0&memswap=0&networkmode=default&rm=1&session=c157f1a8465d46a6c520a352e
f14d6f43eedd09a8e1d4f88&shmsize=0&t=sample-jenkins&target=&ulimits=null: dia
/var/run/docker.sock: connect: permission denied
```

実行結果 3

```
: *** [base] Error 1  
d step 'Execute shell' marked build as failure  
shed: FAILURE
```

ページ更新時: 2018/03/31 14:58:47 UTC [REST API](#) [Je](#)



実行結果 3

enkins ver. 2.107.1