



# Red Hat Developer Tools 1

## Go 1.18.4 Toolset の使用

Go 1.18.4 Toolset のインストールおよび使用



# Red Hat Developer Tools 1 Go 1.18.4 Toolset の使用

---

Go 1.18.4 Toolset のインストールおよび使用

Jacob Valdez  
jvaldez@redhat.com

Eva-Lotte Gebhardt  
egebhard@redhat.com

Peter Macko

Kevin Owen

Vladimir Slavik

Zuzana Zoubkova

## 法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

Go Toolset は、Red Hat Enterprise Linux (RHEL) オペレーティングシステムの開発者向けの Red Hat 製品です。このガイドでは、Go Toolset の概要、さまざまなバージョンの Go ツールを起動して使用する方法、および詳細な情報を含むリソースの検索方法を説明します。

## 目次

多様性を受け入れるオープンソースの強化 .....	3
<b>第1章 GO TOOLSET .....</b>	<b>4</b>
1.1. GO TOOLSET のコンポーネント	4
1.2. GO TOOLSET の互換性	4
1.3. RED HAT ENTERPRISE LINUX 7 での GO TOOLSET へのアクセス	4
1.4. GO TOOLSET のインストール	5
1.5. GO ドキュメントのインストール	6
1.6. 関連情報	7
<b>第2章 GO コンパイラー .....</b>	<b>8</b>
2.1. 前提条件	8
2.2. GO ワークスペースの設定	8
2.3. GO プログラムのコンパイル	8
2.4. GO プログラムの実行	9
2.5. コンパイル済み GO プロジェクトのインストール	9
2.6. GO プロジェクトのダウンロードおよびインストール	10
2.7. 関連情報	11
<b>第3章 GOFMT フォーマットツール .....</b>	<b>12</b>
3.1. 前提条件	12
3.2. コードのフォーマット	12
3.3. コードの変更のプレビュー	12
3.4. コードの簡素化	13
3.5. コードのリファクタリング	14
3.6. 関連情報	15
<b>第4章 GO 競合検出 .....</b>	<b>16</b>
4.1. 前提条件	16
4.2. GO 競合検出の使用	16
4.3. 関連情報	16
<b>第5章 GO TOOLSET が組み込まれたコンテナイメージ .....</b>	<b>17</b>
5.1. RED HAT ENTERPRISE LINUX GO TOOLSET コンテナイメージのコンテンツ	17
5.2. RED HAT ENTERPRISE LINUX コンテナイメージへのアクセス	17
5.3. RHEL 8 での UBI GO TOOLSET コンテナイメージへのアクセス	17
5.4. RHEL 8 のベース UBI コンテナイメージから GO TOOLSET にアクセスする	18
5.5. RED HAT ENTERPRISE LINUX 7 で SOURCE-TO-IMAGE ビルダーイメージとしてコンテナイメージを使用する	18
5.6. 関連情報	19
<b>第6章 GO TOOLSET の変更点 .....</b>	<b>20</b>



## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

# 第1章 GO TOOLSET

Go Toolset は、Red Hat Enterprise Linux (RHEL) の開発者向けの Red Hat 製品です。このツールセットは、Go プログラミング言語のツールとライブラリーを提供します。Go は golang と呼ばれています。

Go Toolset は、Red Hat Developer Tools for Red Hat Enterprise Linux 7 の一部として配布されています。Go Toolset は、Red Hat Enterprise Linux 8 のモジュールとして利用できます。また、Red Hat Enterprise Linux 9 のパッケージとしても利用できます。

## 1.1. GO TOOLSET のコンポーネント

次のコンポーネントが Go Toolset の一部として提供されています。

名前	バージョン	説明
golang	RHEL 7 - 1.18.4、RHEL 8 - 1.18.4、 RHEL 9 - 1.18.4	Go コンパイラー。
delve	RHEL 7 - 1.8.0、RHEL 8 - 1.8.0、 RHEL 9 - 1.8.0	Go デバッガー。

## 1.2. GO TOOLSET の互換性

Go Toolset は、以下のアーキテクチャー上の Red Hat Enterprise Linux 7、Red Hat Enterprise Linux 8、および Red Hat Enterprise Linux 9 で利用できます。

- AMD および Intel 64 ビット
- 64 ビット ARM (RHEL 8 および RHEL 9)
- IBM Power Systems (リトルエンディアン)
- 64 ビット IBM Z

## 1.3. RED HAT ENTERPRISE LINUX 7 での GO TOOLSET へのアクセス

Go Toolset を Red Hat Enterprise Linux 7 にインストールするには、Red Hat Developer Tools リポジトリおよび Red Hat Software Collections リポジトリにアクセスして有効にする必要があります。これらのリポジトリがすでにシステムに割り当てられている場合は、[Go Toolset のインストール](#) を参照してください。

### 手順

1. 以下を実行して **Wget** をインストールします。

```
# yum install wget
```

2. 以下を実行して最新のサブスクリプションデータをダウンロードします。

```
# subscription-manager refresh
```



3. 以下を実行してシステムを登録します。

```
# subscription-manager register
```

グラフィカルユーザーインターフェイス (GUI) を使用してシステムを登録するには、[システム  
の登録と登録解除](#) ガイドを参照してください。

4. 次のコマンドを実行して、利用可能なすべてのサブスクリプションのリストを表示し、プール ID を特定します。

```
# subscription-manager list --available
```

5. **Pool ID** で始まる行でプール ID を検索します。

6. 以下を実行して、**Red Hat Developer Tools** リポジトリへのアクセスを提供するサブスクリプションをシステムに割り当てます。

```
# subscription-manager attach --pool=<pool ID from the subscription>
```

- **<pool ID from the subscription>** は、直前の手順で特定したプール ID に置き換えます。

7. 以下を実行して、システムに割り当てられているサブスクリプションを確認します。

```
# sudo subscription-manager list --consumed
```

8. 以下を実行して **rhel-7-variant-devtools-rpms** リポジトリを有効にします。

```
# subscription-manager repos --enable rhel-7-<variant>-devtools-rpms
```

- **<variant>** は、Red Hat Enterprise Linux システムのバリエーション ( **server** または **workstation** ) に置き換えます。  
最も幅広い開発ツールにアクセスするには、**server** を使用します。

9. 以下を実行して **rhel-variant-rhsc1-7-rpms** リポジトリを有効にします。

```
# subscription-manager repos --enable rhel-<variant>-rhsc1-7-rpms
```

- **<variant>** は、Red Hat Enterprise Linux システムのバリエーション ( **server** または **workstation** ) に置き換えます。

10. 以下を実行して Red Hat Developer Tools GPG キーをシステムに追加します。

```
# cd /etc/pki/rpm-gpg  
# wget -O RPM-GPG-KEY-redhat-devel https://www.redhat.com/security/data/a5787476.txt  
# rpm --import RPM-GPG-KEY-redhat-devel
```

## 関連情報

- システムを登録し、サブスクリプションに関連付ける方法は、[Red Hat Subscription Management](#) ガイドコレクションを参照してください。

## 1.4. GO TOOLSET のインストール

すべての依存パッケージを含む Go Toolset をインストールするには、以下の手順を実行します。

### 前提条件

- Red Hat Enterprise Linux 7 で、Red Hat Developer Tools コンテンツセットへのアクセスを提供するサブスクリプションがシステムに割り当てられている。  
サブスクリプションを割り当てるには、[Red Hat Enterprise Linux 7 での LLVM Toolset へのアクセス](#) を参照してください。
- 利用可能な Red Hat Enterprise Linux のすべての更新がインストールされている。

### 手順

Red Hat Enterprise Linux 7 の場合、**go-toolset-1.18** コレクションをインストールします。

```
# yum install go-toolset-1.18
```

Red Hat Enterprise Linux 8 の場合、以下のコマンドを実行して **go-toolset** モジュールをインストールします。

```
# yum module install go-toolset
```

Red Hat Enterprise Linux 9 の場合、以下のコマンドを実行して **go-toolset** パッケージをインストールします。

```
# dnf install go-toolset
```

## 1.5. GO ドキュメントのインストール

Go プログラミング言語のドキュメントは、ローカルシステムにインストールできます。

### 前提条件

- Go Toolset がインストールされている。  
詳細は、[Go Toolset のインストール](#) を参照してください。

### 手順

**golang-docs** パッケージをインストールするには、以下のコマンドを実行します。

- Red Hat Enterprise Linux 7 の場合:

```
# yum install go-toolset-1.18-golang-docs
```

ドキュメントは、**/opt/rh/go-toolset-1.18/root/usr/lib/go-toolset-1.18-golang/doc/docs.html** にあります。

- Red Hat Enterprise Linux 8 の場合:

```
# yum install golang-docs
```

ドキュメントは、**/usr/lib/golang/doc/effective\_go.html** にあります。

- Red Hat Enterprise Linux 9 の場合:

```
# dnf install golang-docs
```

ドキュメントは、`/usr/lib/golang/doc/go_spec.html` にあります。

## 1.6. 関連情報

- Go プログラミング言語、ツール、およびライブラリーの詳細は、[公式の Go ドキュメント](#) を参照してください。

## 第2章 GO コンパイラー

Go コンパイラーは、Go プログラミング言語のビルドツールおよび依存関係マネージャーです。コードのエラーチェックおよび最適化機能を提供します。

### 2.1. 前提条件

- Go Toolset がインストールされている。  
詳細は、[Go Toolset のインストール](#) を参照してください。

### 2.2. GO ワークスペースの設定

Go プログラムをコンパイルするには、Go ワークスペースを設定する必要があります。

#### 手順

1. ワークスペースディレクトリーを **\$GOPATH/src** のサブディレクトリーとして作成します。  
一般的な選択肢は **\$HOME/go** です。
2. ソースファイルをワークスペースディレクトリーに配置します。
3. 以下を実行して、ワークスペースディレクトリーの場所を環境変数として **\$HOME/.bashrc** ファイルに設定します。

```
$ echo 'export GOPATH=<workspace_dir>' >> $HOME/.bashrc
$ source $HOME/.bashrc
```

<workspace\_dir> は、ワークスペースディレクトリーの名前に置き換えます。

#### 関連情報

- [公式の Go ワークスペースのドキュメント](#)

### 2.3. GO プログラムのコンパイル

Go コンパイラーを使用すると、Go プログラムをコンパイルできます。Go コンパイラーは、コンパイルの結果として、実行可能なバイナリーファイルを作成します。

#### 前提条件

- Go ワークスペースが設定されている。  
ワークスペースの設定方法は、[Go ワークスペースの設定](#) を参照してください。

#### 手順

プロジェクトディレクトリーで以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'go build <output_file>'
```

- <output\_file> は出力ファイルの名前に、<go\_main\_package> はメインパッケージの名前に置き換えます。

- Red Hat Enterprise Linux 8 の場合:

```
$ go build <output_file>
```

- <output\_file> は出力ファイルの名前に、<go\_main\_package> はメインパッケージの名前に置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ go build <output_file>
```

- <output\_file> は出力ファイルの名前に、<go\_main\_package> はメインパッケージの名前に置き換えます。

## 2.4. GO プログラムの実行

Go コンパイラーは、コンパイルの結果として、実行可能なバイナリーファイルを作成します。このファイルを実行し、プログラムを実行するには、以下の手順を実行します。

### 前提条件

- プログラムがコンパイルされている。  
プログラムをコンパイルする方法は、[Go プログラムのコンパイル](#) を参照してください。

### 手順

プログラムを実行するには、実行ファイルが含まれるディレクトリーで実行します。

```
$ ./<file_name>
```

- <file\_name> は、実行ファイルの名前に置き換えます。

## 2.5. コンパイル済み GO プロジェクトのインストール

コンパイル済みの Go プロジェクトをインストールすると、その実行ファイルとライブラリーを他の Go プロジェクトで使用できます。インストール後、プロジェクトの実行ファイルとライブラリーは、Go ワークスペースのディレクトリーに従ってコピーされます。その依存関係もインストールされます。

### 前提条件

- Go ワークスペース。  
詳細は、[Go ワークスペースの設定](#) を参照してください。

### 手順

Go プロジェクトをインストールするには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'go install <go_project>'
```

- <go\_project> は、インストールする Go プロジェクトの名前に置き換えます。

- Red Hat Enterprise Linux 8 の場合:

```
$ go install <go_project>
```

- **<go\_project>** は、インストールする Go プロジェクトの名前に置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ go install <go_project>
```

- **<go\_project>** は、インストールする Go プロジェクトの名前に置き換えます。

## 2.6. GO プロジェクトのダウンロードおよびインストール

オンラインリソースからサードパーティーの Go プロジェクトをダウンロードしてインストールすると、その実行ファイルとライブラリーを他の Go プロジェクトで使用できます。インストール後、プロジェクトの実行ファイルとライブラリーは、Go ワークスペースのディレクトリーに従ってコピーされます。その依存関係もインストールされます。

### 前提条件

- Go ワークスペース。  
詳細は、[Go ワークスペースの設定](#) を参照してください。

### 手順

- Go プロジェクトをダウンロードしてインストールするには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'go get <third_party_go_project>'
```

- **<third\_party\_go\_project>** は、ダウンロードするプロジェクトの名前に置き換えます。

- Red Hat Enterprise Linux 8 の場合:

```
$ go get <third_party_go_project>
```

- **<third\_party\_go\_project>** は、ダウンロードするプロジェクトの名前に置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ go get <third_party_go_project>
```

- **<third\_party\_go\_project>** は、ダウンロードするプロジェクトの名前に置き換えます。

- サードパーティープロジェクトで使用可能な値を確認するには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'go help importpath'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ go help importpath
```

- Red Hat Enterprise Linux 9 の場合:

```
$ go help importpath
```

## 2.7. 関連情報

- Go コンパイラーの詳細は、[公式の Go ドキュメント](#) を参照してください。
- Go Toolset に含まれる **help** インデックスを表示するには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'go help'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ go help
```

- Red Hat Enterprise Linux 9 の場合:

```
$ go help
```

- 特定の Go パッケージのドキュメントを表示するには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'go doc <package_name>'
```

- Red Hat Enterprise Linux 8 の場合:

```
$ go doc <package_name>
```

- Red Hat Enterprise Linux 9 の場合:

```
$ go doc <package_name>
```

Go パッケージの概要は、[Go パッケージ](#) を参照してください。

## 第3章 GOFMT フォーマットツール

Go プログラミング言語は、スタイルガイドではなく、**gofmt** コードフォーマットツールを使用します。**gofmt** は、Go レイアウトルールに従ってコードを自動的にフォーマットします。

### 3.1. 前提条件

- Go Toolset がインストールされている。  
詳細は、[Go Toolset のインストール](#) を参照してください。

### 3.2. コードのフォーマット

**gofmt** フォーマットツールを使用して、特定のパス内のコードをフォーマットできます。パスが1つのファイルである場合、変更はファイルにのみ適用されます。パスがディレクトリーである場合、ディレクトリー内のすべての **.go** ファイルが処理されます。

#### 手順

特定のパス内のコードをフォーマットするには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'gofmt -w <code_path>'
```

- **<code\_path>** は、フォーマットするコードへのパスに置き換えます。

- Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -w <code_path>
```

- **<code\_path>** は、フォーマットするコードへのパスに置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -w <code_path>
```

- **<code\_path>** は、フォーマットするコードへのパスに置き換えます。



#### 注記

元のファイルに書き込みせずにフォーマットされたコードを標準出力に出力するには、**-w** オプションを省略します。

### 3.3. コードの変更のプレビュー

**gofmt** フォーマットツールを使用して、特定のパス内のコードをフォーマットすることによって行われた変更をプレビューできます。統一された diff 形式の出力が、標準出力に出力されます。

#### 手順

特定のパス内のコードの違いを表示するには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:



```
$ scl enable go-toolset-1.18 'gofmt -d <code_path>'
```

- **<code\_path>** は、比較するコードへのパスに置き換えます。
- Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -d <code_path>
```

- **<code\_path>** は、比較するコードへのパスに置き換えます。
- Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -d <code_path>
```

- **<code\_path>** は、比較するコードへのパスに置き換えます。

### 3.4. コードの簡素化

**gofmt** フォーマットツールを使用して、コードを簡素化できます。

#### 手順

- 特定のパス内のコードを簡素化するには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'gofmt -s <code_path>'
```

- **<code\_path>** は、簡素化するコードへのパスに置き換えます。

- Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -s <code_path>
```

- **<code\_path>** は、簡素化するコードへのパスに置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -s <code_path>
```

- **<code\_path>** は、簡素化するコードへのパスに置き換えます。

- 変更を適用するには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'gofmt -w <code_path>'
```

- **<code\_path>** は、フォーマットするコードへのパスに置き換えます。

- Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -w <code_path>
```

- `<code_path>` は、フォーマットするコードへのパスに置き換えます。
- Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -w <code_path>
```

- `<code_path>` は、フォーマットするコードへのパスに置き換えます。

### 3.5. コードのリファクタリング

**gofmt** フォーマットツールを使用して、任意の置換を適用してコードをリファクタリングできます。

#### 手順

- 特定のパス内のコードをリファクタリングするには、以下を実行します。
  - Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'gofmt -r <rewrite_rule> <code_path>'
```

    - `<code_path>` はリファクタリングするコードへのパスに、`<rewrite_rule>` はコードを書き換えるルールに置き換えます。
  - Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -r <rewrite_rule> <code_path>
```

    - `<code_path>` はリファクタリングするコードへのパスに、`<rewrite_rule>` はコードを書き換えるルールに置き換えます。
  - Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -r <rewrite_rule> <code_path>
```

    - `<code_path>` はリファクタリングするコードへのパスに、`<rewrite_rule>` はコードを書き換えるルールに置き換えます。
- 変更を適用するには、以下を実行します。
  - Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'gofmt -w <code_path>'
```

    - `<code_path>` は、フォーマットするコードへのパスに置き換えます。
  - Red Hat Enterprise Linux 8 の場合:

```
$ gofmt -w <code_path>
```

    - `<code_path>` は、フォーマットするコードへのパスに置き換えます。
  - Red Hat Enterprise Linux 9 の場合:

```
$ gofmt -w <code_path>
```

- `<code_path>` は、フォーマットするコードへのパスに置き換えます。

### 3.6. 関連情報

- [公式の gofmt ドキュメント](#)。

## 第4章 GO 競合検出

Go Toolset には、競合状態を検出する Go 標準ライブラリーのツールである Go 競合検出ツールが含まれています。

競合検出を使用すると、ランタイムリソースのオーバーヘッドが著しく増大することに注意してください。

### 4.1. 前提条件

- Go Toolset がインストールされている。  
詳細は、[Go Toolset のインストール](#) を参照してください。

### 4.2. GO 競合検出の使用

Go 競合検出を使用して、競合状態のコードを確認します。

#### 手順

競合検出を使用するには、以下を実行します。

- Red Hat Enterprise Linux 7 の場合:

```
$ scl enable go-toolset-1.18 'go build -race -o <output_file> <go_main_package>'
```

- **<output\_file>** は実行ファイルの名前に、**<go\_main\_package>** はテストするパッケージ名に置き換えます。

- Red Hat Enterprise Linux 8 の場合:

```
$ go build -race -o <output_file> <go_main_package>
```

- **<output\_file>** は実行ファイルの名前に、**<go\_main\_package>** はテストするパッケージ名に置き換えます。

- Red Hat Enterprise Linux 9 の場合:

```
$ go build -race -o <output_file> <go_main_package>
```

- **<output\_file>** は実行ファイルの名前に、**<go\_main\_package>** はテストするパッケージ名に置き換えます。

### 4.3. 関連情報

- [公式の Go 競合検出ドキュメント](#)。

## 第5章 GO TOOLSET が組み込まれたコンテナイメージ

Red Hat Enterprise Linux コンテナイメージまたは Red Hat Universal Base Images (UBI) から独自の Go Toolset コンテナをビルドできます。

### 5.1. RED HAT ENTERPRISE LINUX GO TOOLSET コンテナイメージのコンテンツ

Go Toolset の Red Hat Enterprise Linux 7、Red Hat Enterprise Linux 8、および Red Hat Enterprise Linux 9 コンテナイメージには、次のパッケージが含まれています。

コンポーネント	バージョン	パッケージ
Go	1.18	RHEL 7 - go-toolset-1.18.4 RHEL 8 - go-toolset-1.18.4 RHEL 9 - go-toolset-1.18.4

### 5.2. RED HAT ENTERPRISE LINUX コンテナイメージへのアクセス

コンテナを実行し、アクションを実行する前に、Red Hat レジストリーからコンテナイメージをプルします。

#### 手順

必要なイメージをプルするには、以下を実行します。

- Red Hat Enterprise Linux 7 をベースとするイメージの場合:

```
# podman pull registry.redhat.io/devtools/go-toolset-rhel7
```

- Red Hat Enterprise Linux 8 をベースとするイメージの場合:

```
# podman pull registry.redhat.io/rhel8/go-toolset
```

- Red Hat Enterprise Linux 9 をベースとするイメージの場合:

```
# podman pull registry.redhat.io/rhel9/go-toolset
```

### 5.3. RHEL 8 での UBI GO TOOLSET コンテナイメージへのアクセス

RHEL 8 では、UBI Go Toolset コンテナイメージをインストールして Go Toolset にアクセスします。

または、Go Toolset を RHEL8 ベースの UBI コンテナイメージにインストールすることもできます。詳細は、[RHEL 8 のベース UBI コンテナイメージから Go Toolset にアクセスする](#) を参照してください。

#### 手順

Red Hat レジストリーから UBI Go Toolset コンテナイメージをプルするには、以下を実行します。

```
# podman pull registry.access.redhat.com/ubi8/go-toolset
```

## 5.4. RHEL 8 のベース UBI コンテナイメージから GO TOOLSET にアクセスする

RHEL 8 では、Go Toolset パッケージは Red Hat Universal Base Images (UBI) リポジトリの一部です。つまり、ベース UBI コンテナイメージへの追加として Go Toolset をインストールできます。コンテナイメージのサイズを小さくするには、Go Toolset 全体ではなく、個々のパッケージだけをインストールします。

または、UBI Go Toolset コンテナイメージをインストールして、Go Toolset にアクセスすることもできます。詳細は、[RHEL 8 での UBI Go Toolset コンテナイメージへのアクセス](#) を参照してください。

### 前提条件

- 既存の Containerfile がある。  
Containerfile 作成の詳細は、[Dockerfile reference](#) ページを参照してください。

### 手順

- Go Toolset を含むコンテナイメージを作成するには、以下の行を Containerfile に追加します。

```
FROM registry.access.redhat.com/ubi8/ubi:latest
```

```
RUN yum module install -y go-toolset
```

- 個々のパッケージのみを含むコンテナイメージを作成するには、以下の行を Containerfile に追加します。

```
RUN yum install -y <package-name>
```

- **<package-name>** は、インストールするパッケージの名前に置き換えます。

## 5.5. RED HAT ENTERPRISE LINUX 7 で SOURCE-TO-IMAGE ビルダーイメージとしてコンテナイメージを使用する

Go Toolset コンテナイメージは、Red Hat Enterprise Linux 7 の Source-to-Image (S2I) ビルダーイメージとして使用できます。

### 手順

1. **IMPORT\_URL** 変数を、コードの場所を指定する URL に設定します。
2. **s2i build** コマンドを実行して、S2I ビルダーイメージをビルドします。



### 注記

メインパッケージの場所が **IMPORT\_URL** 変数で指定された場所と同じでない場合は、**INSTALL\_URL** 変数を、ビルド時にアプリケーションのメイン実行ファイルを提供するパッケージの場所を指定する URL に設定します。

### 関連情報

- [OpenShift Container Platform イメージ作成ガイド](#)

- Source-to-Image (S2I) の使用方法の詳細は、[Red Hat Software Collections コンテナイメージの使用](#) を参照してください。

## 5.6. 関連情報

- [Red Hat コンテナレジストリーの Go Toolset コンテナイメージ](#)
- [Red Hat Software Collections コンテナイメージを使用したアプリケーションイメージの構築](#)
- Red Hat UBI イメージの詳細は、[コンテナイメージの使用](#) を参照してください。
- Red Hat UBI リポジトリの詳細は、[Universal Base Images \(UBI\): イメージ、リポジトリ、パッケージ、およびソースコード](#) を参照してください。

## 第6章 GO TOOLSET の変更点

Go Toolset は、RHEL 7、RHEL 8、および RHEL 9 でバージョン 1.17 から 1.18.4 に更新されました。

以下は、主な変更点です。

- 以前のバージョンの Go との下位互換性を維持しながらジェネリクスを導入しました。
- fuzzing ライブラリーが新しくなりました。
- **debug/buildinfo** および **net/netip** パッケージが新しくなりました。
- **go get** ツールがパッケージをビルドまたはインストールしなくなりました。現在は、**go.mod** の依存関係のみを処理します。
- メインモジュールの **go.mod** ファイルで **go 1.17** 以降が指定されている場合、追加の引数なしで **go mod download** コマンドを使用すると、メインモジュールの **go.mod** ファイルで明示的に必要なモジュールのソースコードのみがダウンロードされます。推移的な依存関係のソースコードもダウンロードするには、**go mod download all** コマンドを使用します。
- **go mod vendor** サブコマンドが、出力ディレクトリーを設定する **-o** オプションをサポートするようになりました。
- **go mod tidy** コマンドが、インポートされた各パッケージを提供しているビルドリスト内のモジュールが1つだけであることを確認するためにソースコードが必要なモジュールの追加のチェックサムを **go.sum** ファイルに保持するようになりました。この変更は、メインモジュールの **go.mod** ファイルの Go バージョンに左右されません。

更新の詳細は、アップストリームの [Go 1.18 Release Notes](#) を参照してください。