



Red Hat Developer Tools 1

使用 Go 1.20.10 Toolset

安装和使用 Go 1.20.10 Toolset

Red Hat Developer Tools 1 使用 Go 1.20.10 Toolset

安装和使用 Go 1.20.10 Toolset

Jacob Valdez

jvaldez@redhat.com

法律通告

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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.

摘要

Go Toolset 是 Red Hat Enterprise Linux (RHEL) 操作系统的开发人员提供的红帽产品。使用本指南了解 Go Toolset 的概述，了解如何调用和使用不同版本的 Go 工具，并通过更深入的信息查找资源。

目录

使开源包含更多	3
对红帽文档提供反馈	4
第 1 章 GO TOOLSET	5
1.1. GO TOOLSET 组件	5
1.2. GO TOOLSET 兼容性	5
1.3. 安装 GO TOOLSET	5
1.4. 安装 GO 文档	5
1.5. 其他资源	6
第 2 章 GO 编译器	7
2.1. 先决条件	7
2.2. 设置 GO 工作区	7
2.3. 编译 GO 程序	7
2.4. 运行 GO 程序	8
2.5. 安装编译的 GO 项目	8
2.6. 下载并安装 GO 项目	8
2.7. 其他资源	9
第 3 章 GOFMT 格式工具	11
3.1. 先决条件	11
3.2. 格式化代码	11
3.3. 预览代码更改	11
3.4. 简化代码	12
3.5. 重构代码	12
3.6. 其他资源	12
第 4 章 GO 竞争条件	13
4.1. 先决条件	13
4.2. 使用 GO 竞争条件	13
4.3. 其他资源	13
第 5 章 使用 GO TOOLSET 的容器镜像	14
5.1. RED HAT ENTERPRISE LINUX GO TOOLSET 容器镜像内容	14
5.2. 访问 RED HAT ENTERPRISE LINUX 容器镜像	14
5.3. 访问 RHEL 8 上的 UBI GO TOOLSET 容器镜像	14
5.4. 从 RHEL 8 上的基本 UBI 容器镜像访问 GO TOOLSET	14
5.5. 其他资源	15
第 6 章 GO TOOLSET 中的更改	16

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 [CTO Chris Wright 的信息](#)。

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

提交对具体内容的评论

1. 查看 **Multi-page HTML** 格式的文档，并确保在页面完全加载后看到右上角的 **Feedback** 按钮。
2. 使用光标突出显示您要评论的文本部分。
3. 点击在高亮文本旁的 **Add Feedback** 按钮。
4. 添加您的反馈并点 **Submit**。

通过 Jira 提交反馈（需要帐户）

1. 登录到 [Jira](#) 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括到文档相关部分的链接。
5. 点对话框底部的 **Create**。

第1章 GO TOOLSET

Go Toolset 是 Red Hat Enterprise Linux (RHEL) 的开发人员提供的红帽产品。它提供 Go 编程语言工具和库。请注意，Go 也称为 golang。

Go Toolset 作为 Red Hat Enterprise Linux 8 的模块提供。Go Toolset 作为 Red Hat Enterprise Linux 9 的软件包提供。

1.1. GO TOOLSET 组件

以下组件作为 Go Toolset 的一部分提供：

Name	版本	描述
golang	RHEL 8 – 1.20.10, RHEL 9 – 1.20.10	Go 编译器。
delve	RHEL 8 - 1.20.2, RHEL 9 - 1.20.2	Go 调试器。

1.2. GO TOOLSET 兼容性

Go Toolset 可用于 Red Hat Enterprise Linux 8 和 Red Hat Enterprise Linux 9 在以下构架中：

- AMD 和 Intel 64 位
- 64-bit ARM
- IBM Power Systems, Little Endian
- 64-bit IBM Z

1.3. 安装 GO TOOLSET

完成以下步骤以安装 Go Toolset，包括所有依赖软件包。

先决条件

- 已安装所有可用的 Red Hat Enterprise Linux 更新。

流程

在 Red Hat Enterprise Linux 8 中运行以下命令安装 **go-toolset** 模块：

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

在 Red Hat Enterprise Linux 9 中，运行以下命令安装 **go-toolset** 软件包：

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

1.4. 安装 GO 文档

您可以在本地系统中安装 Go 编程语言的文档。

先决条件

- Go Toolset 已安装。
如需更多信息，[请参阅安装 Go Toolset](#)。

流程

要安装 **golang-docs** 软件包，请运行以下命令：

- 在 Red Hat Enterprise Linux 8 中：

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

您可以在以下路径中找到文档：[/usr/lib/golang/doc/go_spec.html](#)。

- 在 Red Hat Enterprise Linux 9 中：

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

您可以在以下路径中找到文档：[/usr/lib/golang/doc/go_spec.html](#)。

1.5. 其他资源

- 有关 Go 编程语言、工具和库的更多信息，[请参阅官方 Go 文档](#)。

第 2 章 GO 编译器

Go 编译器是 Go 编程语言的构建工具和依赖项管理器。它提供了代码的错误检查和优化。

2.1. 先决条件

- Go Toolset 已安装。
如需更多信息，[请参阅安装 Go Toolset](#)。

2.2. 设置 GO 工作区

要编译 Go 程序，您需要设置 Go 工作区。

流程

1. 创建一个工作区目录作为 **\$GOPATH/src** 的子目录。
一个常见选择是 **\$HOME/go**。
2. 将您的源文件放在您的工作区目录中。
3. 运行以下命令，将工作区目录的位置设置为 **\$HOME/.bashrc** 文件：

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

将 `<workspace_dir>` 替换为工作区目录的名称。

其他资源

- [官方 Go 工作区文档](#)。

2.3. 编译 GO 程序

您可以使用 Go 编译器编译 Go 程序。Go 编译器会创建一个可执行二进制文件，作为编译的结果。

先决条件

- 使用配置的模块设置 Go 工作区。
有关如何设置工作区的详情，请参考 [设置 Go 工作区](#)。

流程

在项目目录中，运行：

- 在 Red Hat Enterprise Linux 8 中：

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

- 将 `<output_file>` 替换为输出文件所需的名称，将 `<go_main_package>` 替换为主软件包的名称。
- 在 Red Hat Enterprise Linux 9 中：

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

- 将 **<output_file>** 替换为输出文件所需的名称，将 **<go_main_package>** 替换为主软件包的名称。

2.4. 运行 GO 程序

Go 编译器会创建一个可执行二进制文件，作为编译的结果。完成以下步骤以执行此文件并运行您的程序。

先决条件

- 您的程序已编译。
有关如何编译程序的更多信息，请参阅 [编译 Go 程序](#)。

流程

要运行您的程序，请在包含可执行文件的目录中运行：

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

- 将 **<file_name>** 替换为可执行文件的名称。

2.5. 安装编译的 GO 项目

您可以安装已编译的 Go 项目，以便在其他 Go 项目中使用其可执行文件和库。安装后，项目的可执行文件和库会根据 Go 工作区中的目录复制到。其依赖项也被安装。

先决条件

- 带有配置的模块的 Go 工作区。
如需更多信息，请参阅 [设置 Go 工作区](#)。

流程

要安装 Go 项目，请运行：

- 在 Red Hat Enterprise Linux 8 中：

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

- 将 **<go_project>** 替换为您要安装的 Go 项目的名称。

- 在 Red Hat Enterprise Linux 9 中：

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

- 将 **<go_project>** 替换为您要安装的 Go 项目的名称。

2.6. 下载并安装 GO 项目

您可以从在线资源下载并安装第三方 Go 项目，以进一步在 Go 项目中使用其可执行文件和库。安装后，项目的可执行文件和库会根据 Go 工作区中的目录复制到。其依赖项也被安装。

先决条件

- Go 工作区。
如需更多信息，[请参阅设置 Go 工作区](#)。

流程

- 要下载并安装 Go 项目，请运行：
 - 在 Red Hat Enterprise Linux 8 中：

```
$ go install <third_party_go_project>
```

 - 将 **<third_party_go_project>** 替换为您要下载的项目名称。
 - 在 Red Hat Enterprise Linux 9 中：

```
$ go install <third_party_go_project>
```

 - 将 **<third_party_go_project>** 替换为您要下载的项目名称。
- 如需有关第三方项目的可能值的信息，请运行：
 - 在 Red Hat Enterprise Linux 8 中：

```
$ go help importpath
```
 - 在 Red Hat Enterprise Linux 9 中：

```
$ go help importpath
```

2.7. 其他资源

- 有关 Go 编译器的更多信息，[请参阅官方 Go 文档](#)。
- 要显示 Go Toolset 中包含的 **帮助** 索引，请运行：
 - 在 Red Hat Enterprise Linux 8 中：

```
$ go help
```
 - 在 Red Hat Enterprise Linux 9 中：

```
$ go help
```
- 要显示特定 Go 软件包的文档，请运行：
 - 在 Red Hat Enterprise Linux 8 中：

```
$ go doc <package_name>
```
 - 在 Red Hat Enterprise Linux 9 中：

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

- 有关 [Go 软件包](#) 的概述信息，请参阅 [Go 软件包](#)。

第 3 章 GOFMT 格式工具

Go 编程语言使用 **gofmt** 代码格式化工具，而不是风格的指南。**gofmt** 根据 Go 布局规则自动格式化您的代码。

3.1. 先决条件

- Go Toolset 已安装。
如需更多信息，[请参阅安装 Go Toolset](#)。

3.2. 格式化代码

您可以使用 **gofmt** 格式工具来格式化给定路径中的代码。当路径导致单个文件时，更改仅应用于该文件。当路径指向某个目录时，将处理目录中的所有 **.go** 文件。

流程

要在指定路径中格式化您的代码，请运行：

- 在 Red Hat Enterprise Linux 8 中：

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

- 将 **<code_path>** 替换为您要格式的代码的路径。

- 在 Red Hat Enterprise Linux 9 中：

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

- 将 **<code_path>** 替换为您要格式的代码的路径。



注意

要将格式化代码打印到标准输出而不是将其写入原始文件中，请省略 **-w** 选项。

3.3. 预览代码更改

您可以使用 **gofmt** 格式工具来预览通过给定路径中格式化代码来完成的更改。统一 diff 格式的输出输出到标准输出。

流程

要显示给定路径中的代码的不同，请运行：

- 在 Red Hat Enterprise Linux 8 中：

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

- 将 **<code_path>** 替换为您要比较的代码的路径。

- 在 Red Hat Enterprise Linux 9 中：

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

- 将 `<code_path>` 替换为您要比较的代码的路径。

3.4. 简化代码

您可以使用 `gofmt` 格式工具来简化代码。

流程

- 要简化给定路径中的 `anf` 格式代码，请运行：

- 在 Red Hat Enterprise Linux 8 中：

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

- 将 `<code_path>` 替换为您要简化的代码的路径。

- 在 Red Hat Enterprise Linux 9 中：

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

- 将 `<code_path>` 替换为您要简化的代码的路径。

3.5. 重构代码

您可以通过应用任意替换来使用 `gofmt` 格式工具来重构代码。

流程

- 要在给定路径中重构并格式化您的代码，请运行：

- 在 Red Hat Enterprise Linux 8 中：

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

- 将 `<code_path>` 替换为您要重构的代码的路径，`<rewrite_rule>` 替换为您要重写的规则。

- 在 Red Hat Enterprise Linux 9 中：

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

- 将 `<code_path>` 替换为您要重构的代码的路径，`<rewrite_rule>` 替换为您要重写的规则。

3.6. 其他资源

- [官方 gofmt 文档](#)。

第 4 章 GO 竞争条件

Go Toolset 包含 Go 竞争检测器，这是用于查找竞争条件的 Go 标准库的工具。请注意，竞争条件检测器具有显著的运行时资源开销。

4.1. 先决条件

- Go Toolset 已安装。
如需更多信息，[请参阅安装 Go Toolset](#)。

4.2. 使用 GO 竞争条件

使用 Go 竞争检测器检查您的代码是否有竞争条件。

流程

要使用竞争条件，请运行：

- 在 Red Hat Enterprise Linux 8 中：

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

- 将 **<output_file>** 替换为可执行文件的名称，**<go_main_package>** 替换为您要测试的软件包的名称。

- 在 Red Hat Enterprise Linux 9 中：

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

- 将 **<output_file>** 替换为可执行文件的名称，**<go_main_package>** 替换为您要测试的软件包的名称。

4.3. 其他资源

- [官方 Go 竞争条件文档](#)。

第 5 章 使用 GO TOOLSET 的容器镜像

您可以从 Red Hat Enterprise Linux 容器镜像或 Red Hat Universal Base Images (UBI) 构建自己的 Go Toolset 容器。

5.1. RED HAT ENTERPRISE LINUX GO TOOLSET 容器镜像内容

Go Toolset 的 Red Hat Enterprise Linux 8 和 Red Hat Enterprise Linux 9 容器镜像包含以下软件包：

组件	Version	软件包
Go	1.20	RHEL 8 - go-toolset-1.20.10 RHEL 9 - go-toolset-1.20.10

5.2. 访问 RED HAT ENTERPRISE LINUX 容器镜像

在运行容器并执行操作前，从红帽 registry 中拉取容器镜像。

流程

要拉取所需的镜像，请运行：

- 对于基于 Red Hat Enterprise Linux 8 的镜像：

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

- 对于基于 Red Hat Enterprise Linux 9 的镜像：

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

5.3. 访问 RHEL 8 上的 UBI GO TOOLSET 容器镜像

在 RHEL 8 中，安装 UBI Go Toolset 容器镜像以访问 Go Toolset。

另外，您可以将 Go Toolset 安装到 RHEL 8 基本 UBI 容器镜像。如需更多信息，请参阅 [RHEL 8 上从基本 UBI 容器镜像访问 Go Toolset](#)。

流程

要从 Red Hat registry 中拉取 UBI Go Toolset 容器镜像，请运行：

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

5.4. 从 RHEL 8 上的基本 UBI 容器镜像访问 GO TOOLSET

在 RHEL 8 中，Go Toolset 软件包是 Red Hat Universal Base Images (UBI) 存储库的一部分，这意味着您可以安装 Go Toolset 作为基本 UBI 容器镜像添加。要将容器镜像大小保持小，请只安装单个软件包，而不是整个 Go Toolset。

另外，您可以安装 UBI Go Toolset 容器镜像来访问 Go Toolset。如需更多信息，请参阅 [访问 RHEL 8 上的 UBI Go Toolset 容器镜像](#)。

先决条件

- 现有的 Containerfile。
有关创建 Containerfiles 的详情，请查看 [Dockerfile 参考](#) 页面。

流程

- 要创建包含 Go Toolset 的容器镜像，请在 Containerfile 中添加以下行：

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

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

- 要创建仅包含单个软件包的容器镜像，请在 Containerfile 中添加以下行：

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

- 将 **<package-name>** 替换为您要安装的软件包的名称。

5.5. 其他资源

- [Go Red Hat Container Registry 中的 Toolset](#) 容器镜像。
- 如需有关 Red Hat UBI 镜像的更多信息，请[参阅使用容器镜像](#)。
- 如需有关 Red Hat UBI 存储库的更多信息，请[参阅通用基础镜像\(UBI\)：镜像、存储库、软件包和源代码](#)。

第 6 章 GO TOOLSET 中的更改

Go Toolset 已从 RHEL 8 和 RHEL 9 上的 1.19 版本更新至 1.20.10。

主要变更包括：

- 在 **unsafe** 软件包中添加了新功能，以处理片段和字符串，而不依赖于内部表示。
- 可比较类型现在可以满足可比较约束。
- 新的 **crypto/ecdh** 软件包。
- **go build** 和 **go test** 命令不再接受 **-i** 标志。
- **go generate** 和 **go test** 命令现在接受 **-skip pattern** 选项。
- **go build**、**go install** 和其它与构建相关的命令现在支持 **-pgo** 和 **-cover** 标志。
- **go** 命令现在在没有 C 工具链的系统上默认禁用 **cgo**。
- **go version -m** 命令现在支持读取更多的 Go 二进制文件类型。
- **go** 命令现在在没有 C 工具链的系统上默认禁用 **cgo**。
- 添加了对从应用程序和集成测试收集代码覆盖配置文件的支持，而不是仅从单元测试收集它们。

有关更新的详情，请查看 [上游 Go 1.20 发行注记](#)。