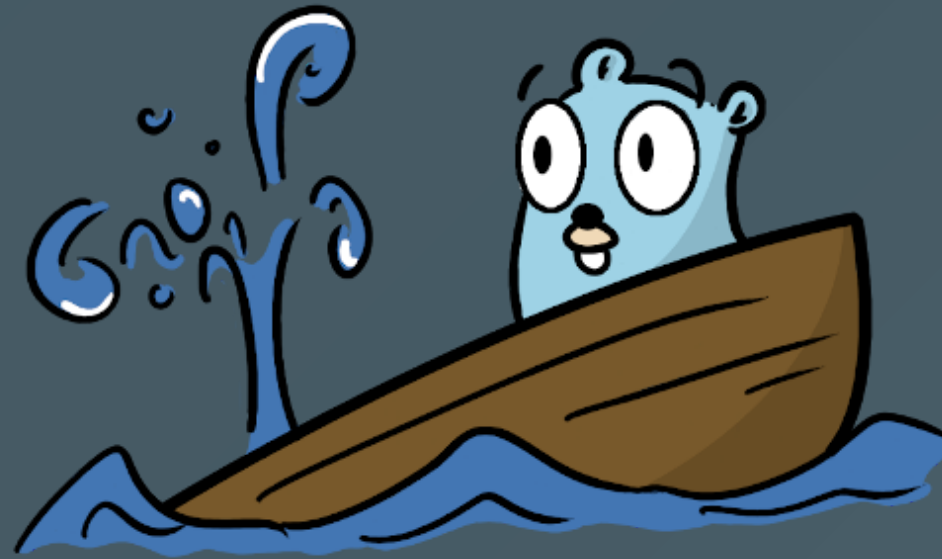


# Shipping Go Without Sinking



Go West Conference 2022

# Speaker Introduction

- **Name:** Sebastian Spaink
- Writing Go for 4 years
- Software Engineer at InfluxData



# What is Telegraf?

Collect data, organizes it, and push it where you want

- 40+ releases of Telegraf
- Open Source, MIT License
- 1000+ Unique Contributors



# A successful release of Telegraf

- Single Binary
- Cross Platform
- Dockerfile + Helm Chart
- Announce release



# Before Release Day

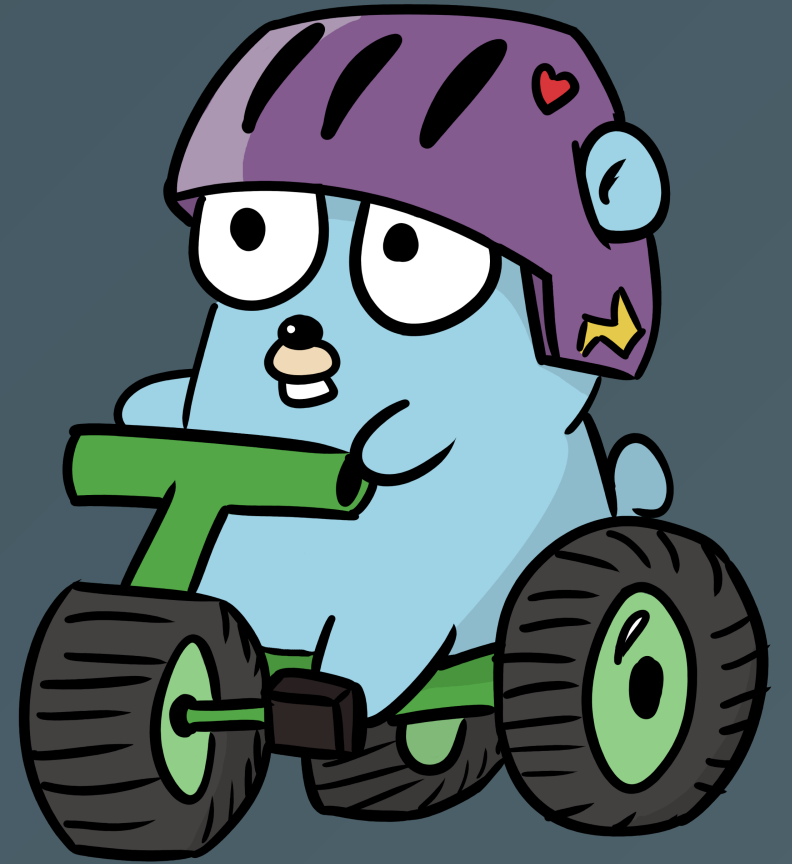
# When? The Release Cycle

## Feature releases

- End of every quarter  
(March, June, September, December)

## Maintenance releases

- Every 3 weeks



# Semantic Versioning

example current version 1.19.0

Type	Version	Description
Patch:	1.19.1	maintenance release
Minor:	1.20.0	feature release
Major:	2.0.0	feature release with breaking changes

# Semantic Versioning In Go

```
// "github.com/Masterminds/semver"  
// Version represents a single semantic version.  
type Version struct {  
    major, minor, patch int64  
    pre                 string  
    metadata            string  
    original             string  
}  
  
currentVersion, _ := semver.NewVersion(version)  
nextVersion := currentVersion.IncMinor()
```



# Conventional Commit Messages

Adding human and machine readable meaning to commit messages

**Example:** feat(inputs.directory\_monitor): Traverse sub-directories

Type	Optional Scope	Description
feat	inputs.directory_monitor	Traverse sub-directories

# Automatic Alpha Builds



telegraf-tiger bot commented 1 hour ago

Contributor



Download PR build artifacts for [linux\\_amd64.tar.gz](#), [darwin\\_amd64.tar.gz](#), and [windows\\_amd64.zip](#).

Downloads for additional architectures and packages are available below.

😊 This pull request doesn't significantly change the Telegraf binary size (less than 1%)

▶ 📄 [Click here to get additional PR build artifacts](#)

# Keep things tidy

```
.PHONY: tidy
tidy:
    go mod verify
    go mod tidy
    @if ! git diff --quiet go.mod go.sum; then \
        echo "please run go mod tidy and check in changes,  
you might have to use the same version of Go as the CI"; \
        exit 1; \
    fi
```

# Lint free code: **golangci-lint**

## lint-pr-changes

succeeded 2 hours ago in 1m 56s

- > ✓ Set up job
- > ✓ Run actions/setup-go@v3
- > ✓ Run actions/checkout@v3
- > ✓ golangci-lint
- > ✓ Post golangci-lint
- > ✓ Post Run actions/checkout@v3
- > ✓ Post Run actions/setup-go@v3
- > ✓ Complete job



# Staying updated

chore: support new golang version v1.17.2 #9925

✓ Closed bot 1 comment 👁

---

 **telegraf-tiger** bot commented on Oct 13, 2021 Contributor 😊 ✎ ⋮

There is a new golang version available. Please update Telegraf to use version 'v1.17.2'.

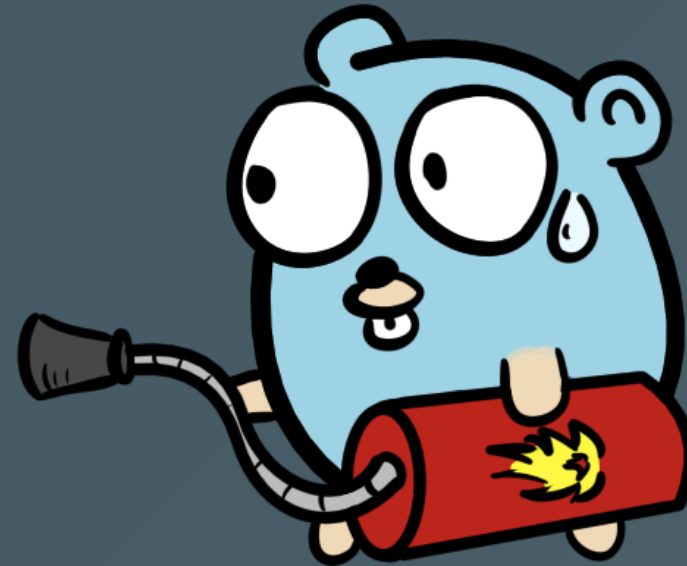
 **sspaink** on Oct 21, 2021 Member 😊 ✎ ⋮

Good bot, Go version has been updated.

✓  **sspaink** closed this as completed on Oct 21, 2021

<https://go.dev/VERSION?m=text>







# Release Day



# Starting the Release

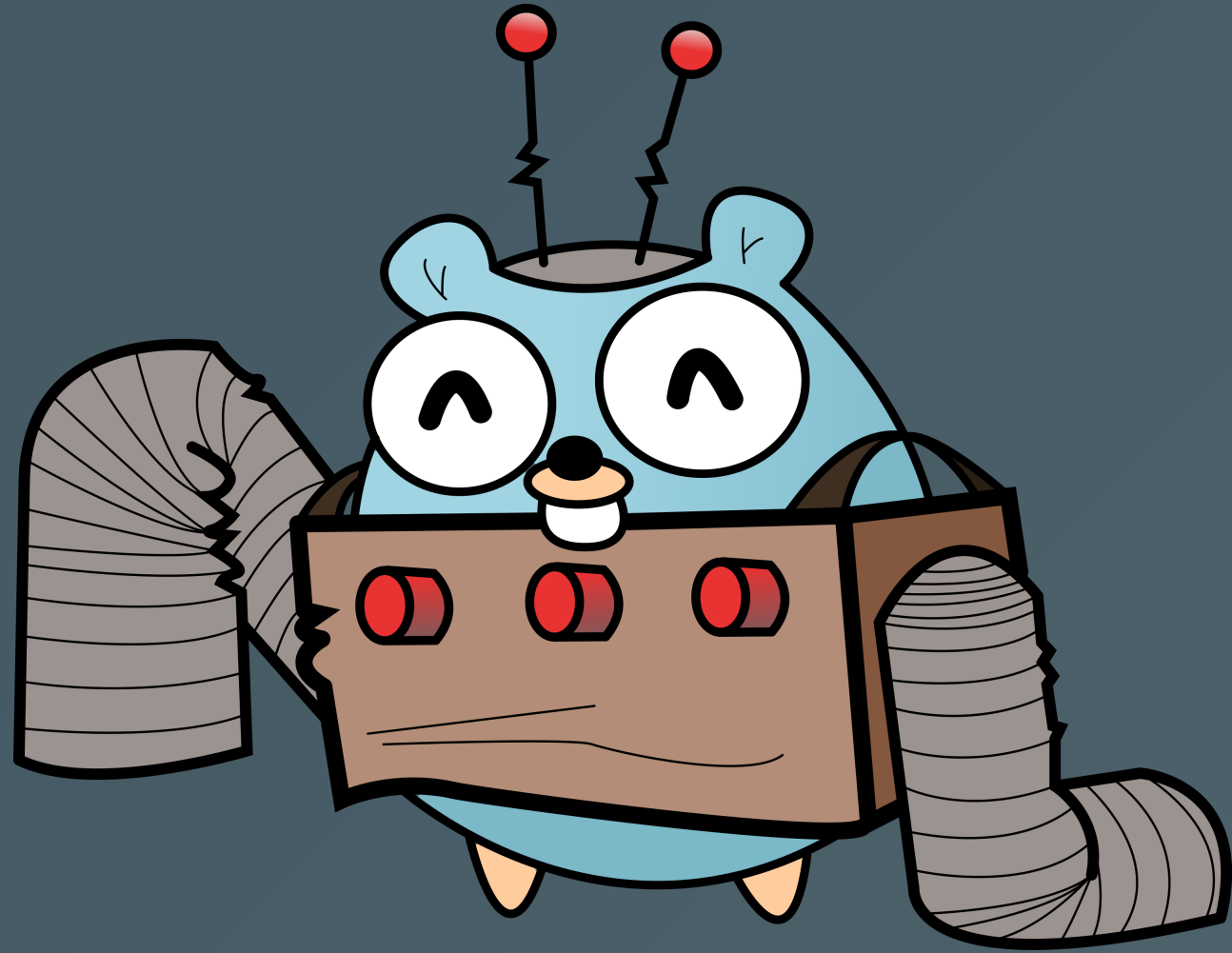
- Eeny, meeny, miny moe, Catch a tiger by the toe
- Track progress in slack

## Sebastian Spaink v1.23.0 release thread

-  Prepare release branch
-  Sign packages
-  Update the [repos.influxdata.com](https://repos.influxdata.com) repository
-  Update Docker images
-  Update downloads page
-  Update Helm charts

# Steps Overview

1. Prepare release branch
2. Building and packaging
3. Announcing the release





# Prepare release branch



# Git branching strategy

- copy master branch
- branch for each release
- maintain release branch



# Executing external commands

```
func RunCommand(path, command string, arguments ...string) (string, error) {  
    var w io.Writer  
    var stdBuffer bytes.Buffer  
    w = io.MultiWriter(os.Stdout, &stdBuffer)  
  
    cmd := exec.Command(command, arguments...)  
    cmd.Dir = path  
    cmd.Stdout = w  
    cmd.Stderr = w  
    err := cmd.Run()  
    if err != nil {  
        return "", err  
    }  
    return stdBuffer.String(), nil  
}
```

# Adding some color

```
// Wrap color implements the io.Writer interface  
// Using fatih/color package it prints to stdout but with color  
type WrapColor struct{}
```

```
func (w *WrapColor) Write(b []byte) (int, error) {  
    color.Cyan(string(b))  
    return len(b), nil  
}
```

.....

```
var w io.Writer  
var stdBuffer bytes.Buffer  
var c WrapColor  
w = io.MultiWriter(&c, &stdBuffer)
```

# Update git branches

```
RunCommand(path, "make", "test")
// maintenance release
for response != "Y" && response != "N" {
    fmt.Println(color.YellowString("Cherry-pick this commit?: %s", c.Title))
    response, _ = prompt.Run()
    switch response {
    case "Y", "y":
        arguments = []string{"cherry-pick", "-x", c.Hash}
        _, err := RunCommand(path, "git", arguments...)
    }
}

// feature release
arguments := []string{"checkout", "-b", branchName}
output, err := RunCommand(path, "git", arguments)
```

# Working with git commits

```
type Commit struct {  
    Hash          string  
    AuthorName    string  
    Type          string // (e.g. `feat`)  
    Scope         string // (e.g. `core`)  
    Subject       string // (e.g. `Add new feature`)  
    Title         string // (e.g. `feat(core): Add new feature`)  
}
```

# Collecting git commits

```
separator := "@@__CHGLOG__@"
delimiter := "@@__CHGLOG_DELIMITER__@"
logFormat := separator + strings.Join([]string{
    "HASH",
    "AUTHOR",
    "SUBJECT",
    "BODY",
}, delimiter)

arguments := []string{"log", fmt.Sprintf("--pretty=%s", logFormat), fromBranch + ".." + toBranch}
RunCommand(path, "git", arguments...)
```

# Changelog template

```
## {{ .Version }} [{{ .Date }}]  
{{ range .CommitGroups }}  
### {{ println .Title }}  
{{ range .Commits -}}  
- {{ .PullRequestLink }} {{ if .Scope }}`{{ .Scope }}` {{ end }}{{ println .Subject }}  
{{- end -}}  
{{ end -}}
```



# Example

**v1.24.1 [2022-09-19]**

## Bugfixes

- [#11787](#) Clear error message when provided config is not a text file

## Dependency Updates

- [#11788](#) `deps` Bump [cloud.google.com/go/pubsub](https://cloud.google.com/go/pubsub) from 1.24.0 to 1.25.1

# Preparing the code: lessons learned

- Conventional commit messages unlocks automation
- Automate using Go

# Preparing the code: future improvements

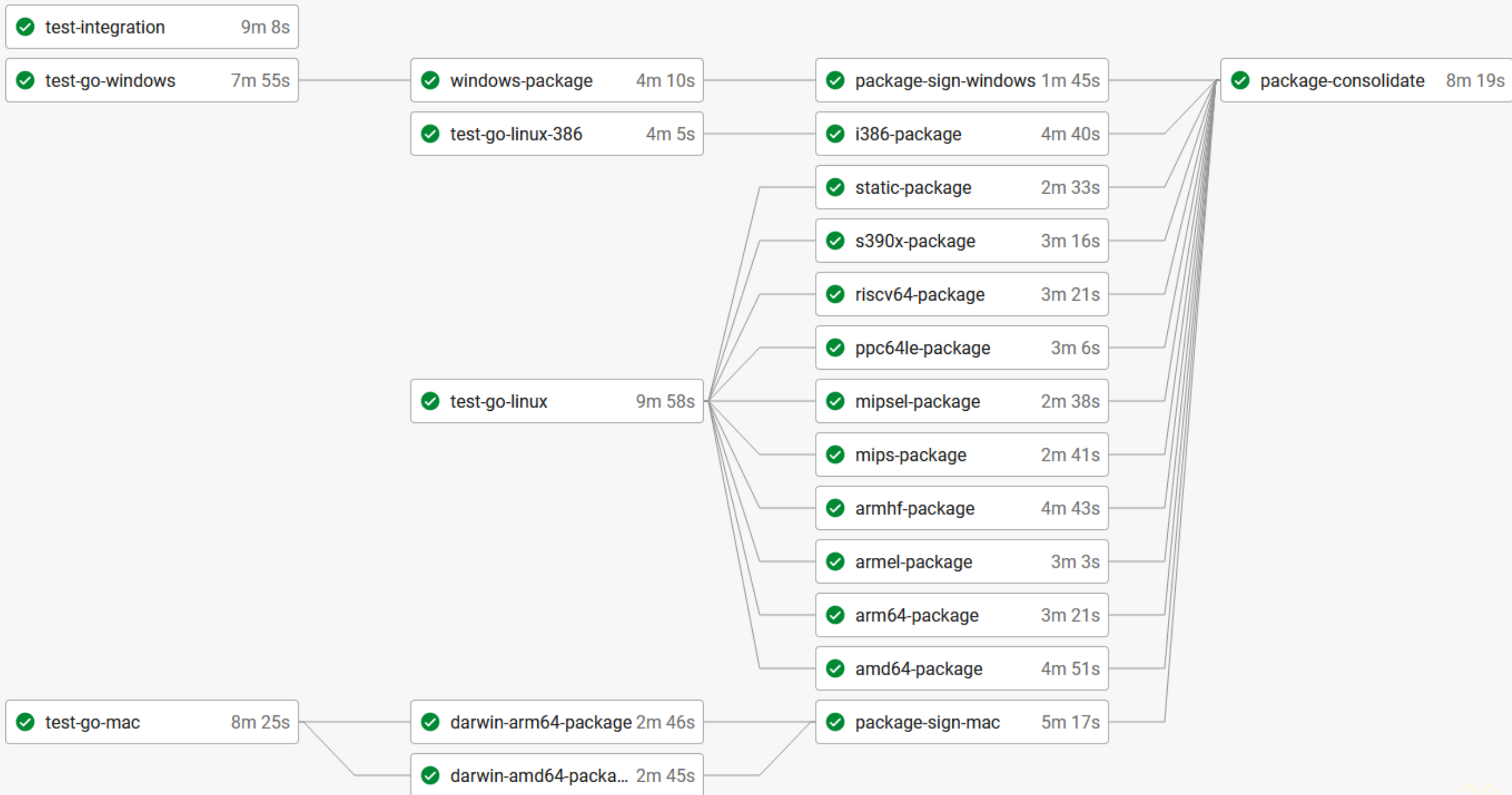
- Don't wait for release day
  - Cherry-pick continuously
  - Update changelog continuously

## Challenges:

- What if change is fixing a feat?

# Building and Packaging





# Build tags

Identifier added to determine when code should be included in build

Syntax after Go v1.17	Syntax before Go v1.17
//go:build windows	// +build windows
//go:build linux && amd64	// +build linux,amd64

# Architectures Defined

```
# e.g. make package include_packages="$(make amd64) "
arm64 += linux_arm64.tar.gz arm64.deb aarch64.rpm
.PHONY: arm64
arm64:
    @ echo $(arm64)
amd64 += freebsd_amd64.tar.gz linux_amd64.tar.gz amd64.deb x86_64.rpm
.PHONY: amd64
amd64:
    @ echo $(amd64)
windows += windows_i386.zip windows_amd64.zip
.PHONY: windows
windows:
    @ echo $(windows)
....
```

# make build

use -ldflags to change the value of variables at build time

```
tag := $(shell git describe --exact-match --tags 2>/dev/null)
version := $(tag:v%=%)
```

```
INTERNAL_PKG=github.com/influxdata/telegraf/internal
LDFLAGS := -X $(INTERNAL_PKG).Version=$(version)
```

```
.PHONY: build
```

```
build:
```

```
    go build -ldflags "$$(LDLAGS)" ./cmd/telegraf
```



# make package

```
include_packages := $(mips) $(mipsel) $(arm64) $(amd64) $(static) $(armel)
$(armhf) $(riscv64) $(s390x) $(ppc64le) $(i386) $(windows) $(darwinamd64) $(darwinarm64)

.PHONY: $(include_packages)
$(include_packages):
    @$(MAKE) install
    @mkdir -p $(pkgdir)

@if [ "$(suffix $@)" = ".rpm" ]; then \
    fpm ...
elif [ "$(suffix $@)" = ".deb" ]; then \
    fpm ....
elif [ "$(suffix $@)" = ".zip" ]; then \
    (cd $(dir $(DESTDIR)) && zip -r - ./*) > $(pkgdir)/telegraf-$(tar_version)_$@ ;\
elif [ "$(suffix $@)" = ".gz" ]; then \
    tar -czvf $(pkgdir)/telegraf-$(tar_version)_$@ -C $(dir $(DESTDIR)) . ;\
fi
```

# Mage

Mage is a make/rake-like build tool using Go.

```
func build(p platform) (string, error) {  
    env := map[string]string{"GOOS": p.OS, "GOARCH": p.ARCH}  
    folderName := fmt.Sprintf("%s_%s", p.OS, p.ARCH)  
    binName := productName + p.Extension  
    filePath := filepath.Join("bin", folderName, binName)  
    _ := runCmd(env, "go", "build", "-o", filePath, "cmd/main.go")  
  
    return filePath, nil  
}
```

Check it out: <https://github.com/magefile/mage>

# Adding version and icon to Windows

Using package `josephspurrier/goversioninfo`

```
//go:generate goversioninfo -icon=../../assets/windows/tiger.ico
```

```
{  
  "StringFileInfo": {  
    "ProductName": "Telegraf",  
    "ProductVersion": "1.25.0-71b4a0af"  
  }  
}
```

# Generating versioninfo.json

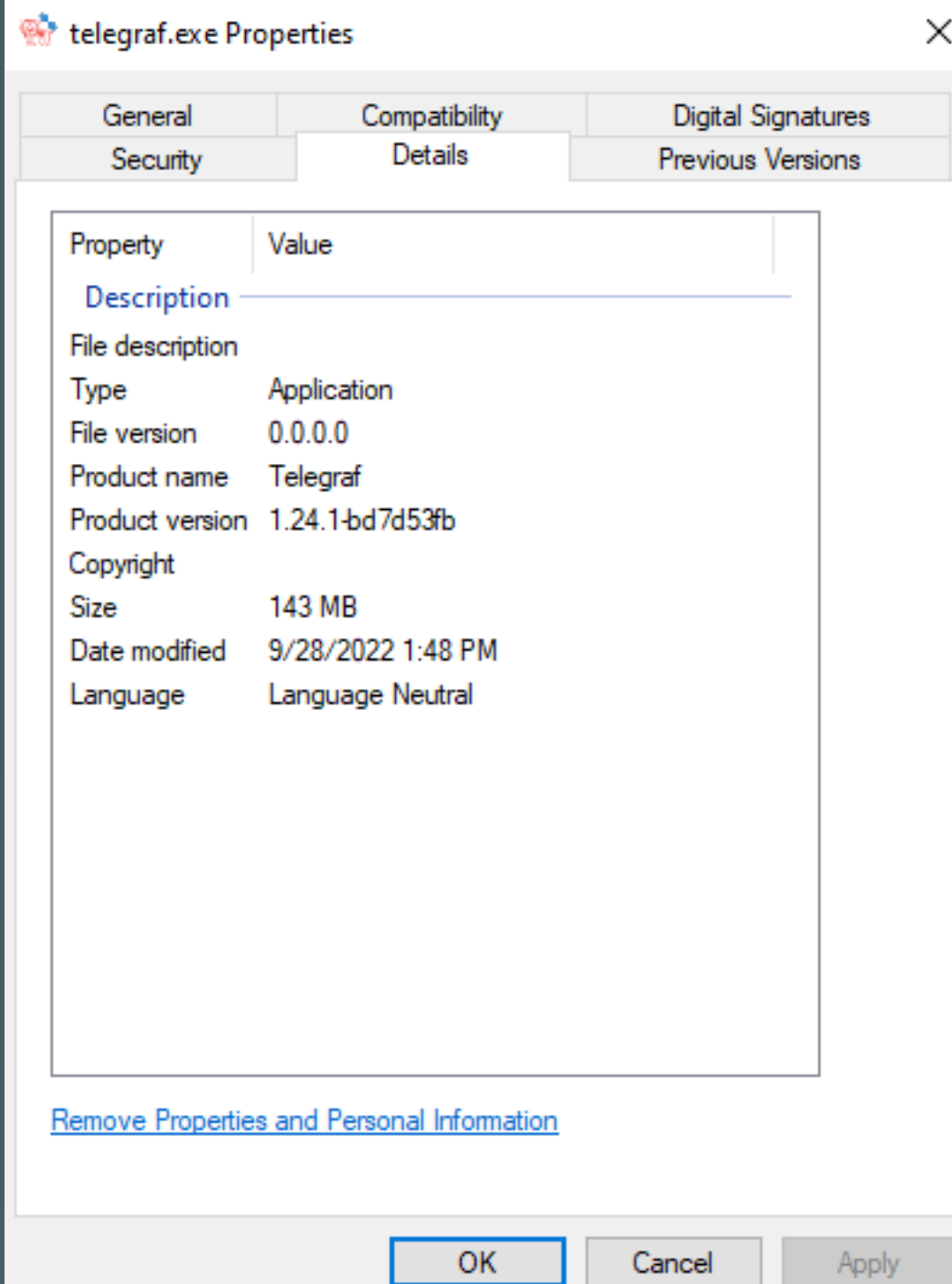
```
version, _ := exec.Command("make", "version").Output()

v := VersionInfo{
    StringFileInfo: StringFileInfo{
        ProductName:    "Telegraf",
        ProductVersion: version,
    },
}

if err := ioutil.WriteFile("cmd/telegraf/versioninfo.json", file, 0644); err != nil {
    log.Fatalf("Failed to write versioninfo.json: %v", err)
}
```



telegraf.exe



# Update Package Repository

- Sign artifacts with GPG key
- Upload to S3 bucket
- Update <http://repos.influxdata.com/>

# Building and Packaging: lessons learned

- Don't run the CI on wednesdays
- Make artifacts readily available
- Sign mac nightly, replicate release pipeline

# Building and Packaging: future improvements

- Migrate more steps to continuous integration pipeline



# Announcing the release





# Update multiple locations

- [influxdata/telegraf](#) (public)
- [influxdata/influxdata-docker](#) (public)
- [influxdata/official-images](#) (public, fork of docker-library)
- [influxdata/helm-charts](#) (public)
- [influxdata/website](#) (private)

Releases

Tags

 Choose a tag ▾










 Target: master ▾

Choose an existing tag, or create a new tag when you publish this release.

Release title

Write

Preview

Generate release notes

Describe this release



Attach binaries by dropping them here or selecting them.

☐ This is a pre-release

We'll point out that this release is identified as non-production ready.

Publish release

Save draft

## Packages

Arch	Platform	Package	SHA256
arm64	CentOS	<a href="#">telegraf-1.24.2-1.aarch64.rpm</a>	3ff70e356f59f4d96f026a2d5f920b41cce8694f3968aa27056db7255738
armel	CentOS	<a href="#">telegraf-1.24.2-1.armel.rpm</a>	570da9d13d4e4f2d8a433d392984f4d68c543844d3e3659ae5ab742cbea9
armv6hl	CentOS	<a href="#">telegraf-1.24.2-1.armv6hl.rpm</a>	ae75a4f741301378389deed6f3c5c4423e01af700348dec34eda8b99562d
i386	CentOS	<a href="#">telegraf-1.24.2-1.i386.rpm</a>	3dea0a3f682da558517f1e42b4401708f9383a0c6c4518ff0fc8d905b08e
ppc64le	CentOS	<a href="#">telegraf-1.24.2-1.ppc64le.rpm</a>	6b803409c7879e4abd9830fc48ffeca4ed5423c8b83b7464a2e1fd835b1d
riscv64	CentOS	<a href="#">telegraf-1.24.2-1.riscv64.rpm</a>	c9cb0821a12bff9d4167287ad0aff47fbad433954bdd992381f5c08559b1
s390x	CentOS	<a href="#">telegraf-1.24.2-1.s390x.rpm</a>	03ebabef6381514152c54708b49f70d508a7655a5cedac451b24e7144eb7
amd64	CentOS	<a href="#">telegraf-1.24.2-1.x86_64.rpm</a>	f7e7c75b99aa1950905c0c542e5c08b15dc99d91546bfee35887a65eb5f0
amd64	macOS	<a href="#">telegraf-1.24.2_darwin_amd64.dmg</a>	34abd69bddea6ac58f6f94d5c9101f1966d5e87ab162d6e59379debb8459
amd64	macOS	<a href="#">telegraf-1.24.2_darwin_amd64.tar.gz</a>	e94000bc4a8e8eb1e60349d3ec52b5c83b6b66d3ef3cc817842be0d16100
arm64	macOS	<a href="#">telegraf-1.24.2_darwin_arm64.dmg</a>	6822fb62711ee8ecd9859614ada022cf04cfbf8e2b0dee2a528430d1259e
arm64	macOS	<a href="#">telegraf-1.24.2_darwin_arm64.tar.gz</a>	7a7871671a57cb64ccae4e2744b34b9d197b07eec0fd18bb5b588f6d52f1
amd64	FreeBSD	<a href="#">telegraf-1.24.2_freebsd_amd64.tar.gz</a>	a8652ca1b90b9806e26613c7088d907668c466a633d527185201808731bb

<https://portal.influxdata.com/downloads/>

**Version**

Telegraf v1.24.1



**Platform**

Linux Binaries (64-bit)



SHA256: `f3a3d80909e54599fec d98dcfc7ae1e0202153d89f1f4ecd9c28a51658ac357d`

```
wget https://dl.influxdata.com/telegraf/releases/telegraf-1.24.1_linux_amd64.tar.gz
tar xf telegraf-1.24.1_linux_amd64.tar.gz
```

[Documentation](#) | [Release Notes](#)

# Updating the Website

```
b, err = sjson.SetBytes(b, "telegraf_stable.version", version)
if err != nil {
    return nil, fmt.Errorf("failed to set telegraf_stable.version: %w", err)
}
```

# Update Docker manifest.json

```
// Read the manifest file to determine current versions and add latest version
type Manifest struct {
    Name          string    `json:"name"`
    Maintainers   []string  `json:"maintainers"`
    Versions      []string  `json:"versions"`
    Architectures []string  `json:"architectures"`
    Variants      []Variant `json:"variants"`
}

ver, _ := semver.NewVersion(version)

// Drop oldest version and add newest version
var oldestVersion string
oldestVersion, m.Versions = m.Versions[0], m.Versions[1:]
m.Versions = append(m.Versions, fmt.Sprintf("%d.%d", ver.Major(), ver.Minor()))
```



# Telegraf v1.24.2 #13278

[Code](#)

Merged by tianon `docker-library:master` ← `influxdata:telegraf-v1.24.2` 12 days ago

Conversation 1 Commits 1 Checks 11 Files changed 1 +3 -3

all commits File filter Conversations Jump to

0 / 1 files viewed

Review changes

6 library/telegraf

Viewed

@@ -5,7 +5,7 @@ Maintainers: David Reimchussel <dreimchussel@influxdata.com> (@reimda),	
5 .....Sven Rebhan <srebhan@influxdata.com> (@srebhan),	5 .....Sven Rebhan <srebhan@influxdata.com> (@srebhan),
6 .....Alan Pope <apope@influxdata.com> (@popey)	6 .....Alan Pope <apope@influxdata.com> (@popey)
7 GitRepo: <a href="https://github.com/influxdata/influxdata-docker.git">https://github.com/influxdata/influxdata-docker.git</a>	7 GitRepo: <a href="https://github.com/influxdata/influxdata-docker.git">https://github.com/influxdata/influxdata-docker.git</a>
8 GitCommit: <code>b827ab25d730659b98439d1c0cbfb17baf60b468</code>	8 GitCommit: <code>aa51ef23824f6a7f9c3ff3329012dc0fe966c126</code>
9	9
10	10
11 Tags: 1.22, 1.22.4	11 Tags: 1.22, 1.22.4
@@ -22,9 +22,9 @@ Directory: telegraf/1.23	
22 Tags: 1.23-alpine, 1.23.4-alpine	22 Tags: 1.23-alpine, 1.23.4-alpine
23 Directory: telegraf/1.23/alpine	23 Directory: telegraf/1.23/alpine
24	24
25 Tags: 1.24, 1.24.1, latest	25 Tags: 1.24, 1.24.2, latest
26 Architectures: amd64, arm32v7, arm64v8	26 Architectures: amd64, arm32v7, arm64v8
27 Directory: telegraf/1.24	27 Directory: telegraf/1.24
28	28
29 Tags: 1.24-alpine, 1.24.1-alpine, alpine	29 Tags: 1.24-alpine, 1.24.2-alpine, alpine
30 Directory: telegraf/1.24/alpine	30 Directory: telegraf/1.24/alpine



# Testing Dockerfile

```
container, _ := testcontainers.GenericContainer(ctx,
testcontainers.GenericContainerRequest{
    ContainerRequest: req,
    Started:          true,
})

name, _ := container.Name(ctx)

cmd := exec.Command("docker", "exec", name, "telegraf", "--version")
out, _ := cmd.CombinedOutput()

// Verify the correct version of Telegraf exists
if !strings.Contains(string(out), ver.String()) {
    return fmt.Errorf("Expected Telegraf version %s but got %s", version, string(out))
}
```

# Update Helm Chart

```
import yaml "gopkg.in/yaml.v3"

var n yaml.Node
err := yaml.Unmarshal(data, &n)
if err != nil {
    return nil, fmt.Errorf("failed to unmarshal YAML: %w", err)
}

_, err = walkContent(&n, "tag", fmt.Sprintf("%d.%d-alpine", version.Major(), version.Minor()))
if err != nil {
    return nil, fmt.Errorf("failed to update tag in YAML: %w", err)
}

updated, err := yaml.Marshal(&n)
if err != nil {
    return nil, fmt.Errorf("failed to marshal YAML: %w", err)
}
```

# chore: update Telegraf versions for v1.24.2 #512

Merged by powersj master ← telegraf-v1.24.2 11 days ago

Conversation 0 Commits 1 Checks 2 Files changed 2

all commits File filter Conversations +

0 / 2 files viewed

4 charts/telegraf-ds/Chart.yaml

@@ -1,7 +1,7 @@

```
1 apiVersion: v1
2 name: telegraf-ds
3 version: 1.1.3
4 appVersion: 1.24.1
5 deprecated: false
6 description: Telegraf is an agent written in Go for
  collecting, processing, aggregating, and writing metrics.
7 keywords:
```

```
1 apiVersion: v1
2 name: telegraf-ds
3 version: 1.1.4
4 appVersion: 1.24.2
5 deprecated: false
6 description: Telegraf is an agent written
  collecting, processing, aggregating, and w
7 keywords:
```

4 charts/telegraf/Chart.yaml

@@ -1,7 +1,7 @@

```
1 apiVersion: v2
2 name: telegraf
3 version: 1.8.21
4 appVersion: 1.24.1
5 deprecated: false
6 description: Telegraf is an agent written in Go for
  collecting, processing, aggregating, and writing metrics.
7 keywords:
```

```
1 apiVersion: v2
2 name: telegraf
3 version: 1.8.22
4 appVersion: 1.24.2
5 deprecated: false
6 description: Telegraf is an agent written
  collecting, processing, aggregating, and w
7 keywords:
```

# Distributing Binaries: lessons learned

- Document all permissions required
- If there is a problem increment patch, or the hash will change

# Distributing Binaries: future improvements

- Migrate more steps to continuous integration pipeline

# Conquering a Major Release

- Breaking changes
- Backwards compatibility important
- Deprecating configuration options slowly
- Getting into the habit

