

Juju

Extending go
6 October 2015

Domas Monkus
Canonical

About me

- live in Vilnius
- work for Canonical
- 3 yrs go experience
- previous to that: c, c++, python

What is juju?



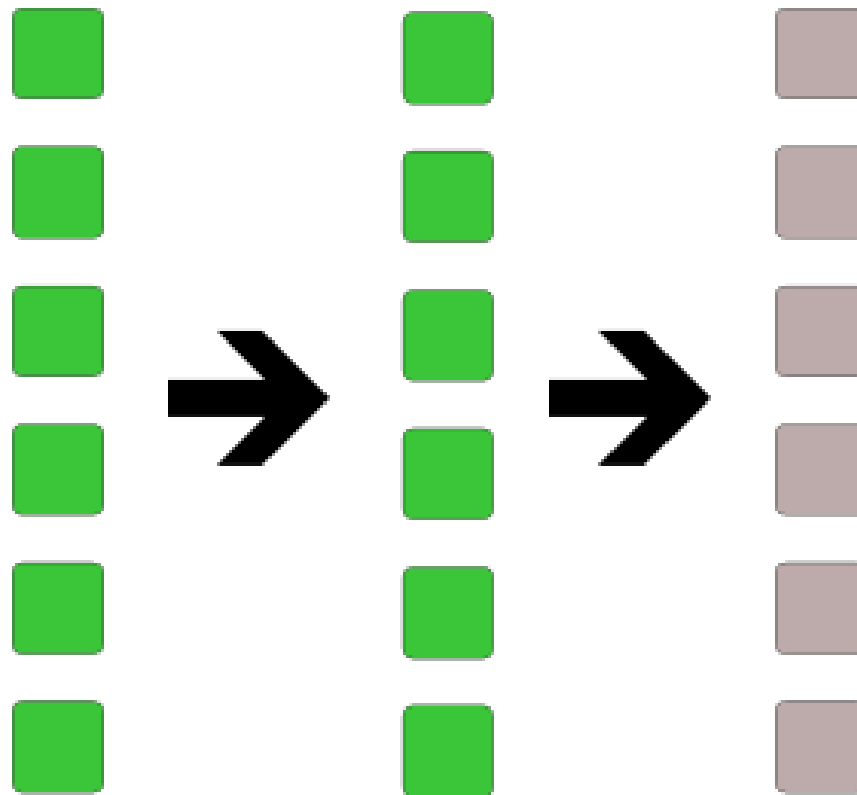
- A cloud orchestration tool from **Canonical**.
- Allows you to configure, manage and scale services (known as charms) in a cloud environment.
- One of the largest projects written in go (**120k+ lines of go madness!**).
- Rewritten from python.

A short demo

Managing goroutines - gopkg.in/tomb.v2

- goroutines are cheap
- goroutines are not easy to manage (esp. 100s of them)
- Forced shutdowns vs error conditions

Control worker lifetimes



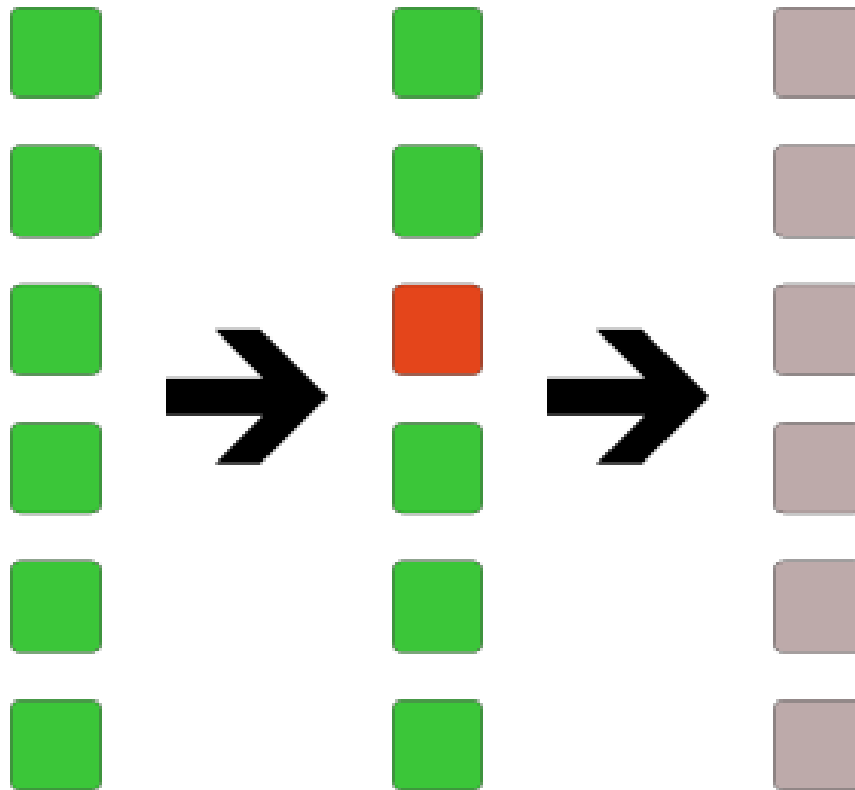
kill

Example

```
func main() {  
    t := &tomb.Tomb{}  
    for i := 0; i < 5; i++ {  
        t.Go(func() error {  
            for {  
                select {  
                case <-t.Dying():  
                    fmt.Printf("dying because: %v\n", t.Err())  
                    return nil  
                }  
            }  
        })  
    }  
  
    // shutdown  
    t.Kill(errors.New("shutdown"))  
    err := t.Wait()  
    fmt.Println(err)  
}
```

[Run](#)

Catch worker errors



Example

```
// start bad worker
t.Go(func() error {
    return errors.New("bad worker died")
})

err := t.Wait()
fmt.Println(err)
```

[Run](#)

Error handling - github.com/juju/errors

- error is very simple
- error is not very helpful in root cause analysis
- we need error tracing
- we need error annotations

```
os.ErrNotExist
```

sometimes means

```
errors.New("config file blah.yaml not found")
```

Example

```
err := errors.Annotate(os.ErrNotExist, "wrapped error")

// the error
fmt.Println(err)
// the cause
fmt.Println(errors.Cause(err))
// the stack
fmt.Println(errors.ErrorStack(err))
```

[Run](#)

Testing - gopkg.in/check.v1

- we need fixtures!

Example

```
// Boilerplate.
func Test(t *testing.T) {
    gc.TestingT(t)
}

// The test suite.
type TSuite struct{}

var _ = gc.Suite(&TSuite{})

func (t *TSuite) SetUpSuite(c *gc.C) {}

func (t *TSuite) TearDownSuite(c *gc.C) {}

func (t *TSuite) SetUpTest(c *gc.C) {}

func (t *TSuite) TearDownTest(c *gc.C) {}

func (t *TSuite) TestSomething(c *gc.C) {}

func (t *TSuite) BenchmarkSomething(c *gc.C) {}
```

Other useful juju packages

- github.com/juju/schema - coercing dynamic structures into known forms
- github.com/juju/deputy - smart os/exec
- github.com/juju/ratelimit - ratelimiting
- github.com/juju/loggo - module-level logging
- launchpad.net/gnuflag - gnu format flags
- gopkg.in - package api version locking
- launchpad.net/godeps - repeatable builds

Talk source files

Find the talk source files at github.com/tasdomas/go-talk (<https://github.com/tasdomas/go-talk>)

Thank you

Domas Monkus

Canonical

domas.monkus@canonical.com (mailto:domas.monkus@canonical.com)

<http://domas.monkus.it> (http://domas.monkus.it)

[@tasdomas](http://twitter.com/tasdomas) (http://twitter.com/tasdomas)

