

Travis CI

Miro Hrončok

InstallFest 2015

8. března 2015

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle Wikipedie:
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ vzniklo to jako metoda extrémního programování
- ▶ 20 let starý pojem
- ▶ dle [Wikipedie](#):
 - ▶ kód ve společném repozitáři
 - ▶ automatické zbuildění
 - ▶ automatické testy
 - ▶ do repozitáře přispívají všichni každý den
 - ▶ každý příspěvek se zbuildí a otestuje
 - ▶ build je rychlý
 - ▶ testuje se v kopii produkčního prostředí
 - ▶ každá nová změna jde jednoduše získat
 - ▶ všichni by měli vidět výsledek posledního testu
 - ▶ automatický deployment
- ▶ odeberme VSC (git apod.)

Continuous integration

- ▶ sypeme do toho commity
- ▶ zbuildí se to
- ▶ otestuje se to
- ▶ případně se to nasadí

Continuous integration

- ▶ sypeme do toho commity
- ▶ zbuildí se to
- ▶ otestuje se to
- ▶ případně se to nasadí

Continuous integration

- ▶ sypeme do toho commity
- ▶ zbuildí se to
- ▶ otestuje se to
- ▶ případně se to nasadí

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

travis.yml

Demo

Continuous integration

- ▶ sypeme do toho commity
- ▶ zbuildí se to
- ▶ otestuje se to
- ▶ případně se to nasadí

Výhody Continuous integration

- ▶ zjistíme, že se rozbily testy
- ▶ zjistíme, kdy se rozbily testy
- ▶ získáme development build

Výhody Continuous integration

- ▶ zjistíme, že se rozbily testy
- ▶ zjistíme, kdy se rozbily testy
- ▶ získáme development build

Výhody Continuous integration

- ▶ zjistíme, že se rozbily testy
- ▶ zjistíme, kdy se rozbily testy
- ▶ získáme development build

Co potřebujeme k Continuous integration

Travis CI

Miro Hrončok

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

travis.yml

Demo

- ▶ nějaké testy
- ▶ nějaký CI systém
- ▶ automatizaci CI

Co potřebujeme k Continuous integration

Travis CI

Miro Hrončok

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

travis.yml

Demo

- ▶ nějaké testy
- ▶ nějaký CI systém
- ▶ automatizaci CI

Co potřebujeme k Continuous integration

Travis CI

Miro Hrončok

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

travis.yml

Demo

- ▶ nějaké testy
- ▶ nějaký CI systém
- ▶ automatizaci CI

Co je Travis CI

- ▶ travis-ci.org
- ▶ služba pro automatické CI
- ▶ pouze pro GitHub repozitáře
 - ▶ zdarma pro veřejné repozitáře
 - ▶ placené travis-ci.com pro privátní
- ▶ zdarma v rámci GitHub Student Developer Pack
- ▶ výborná dokumentace na docs.travis-ci.com

Co je Travis CI

- ▶ travis-ci.org
- ▶ služba pro automatické CI
- ▶ pouze pro GitHub repozitáře
 - ▶ zdarma pro veřejné repozitáře
 - ▶ placené travis-ci.com pro privátní
 - ▶ zdarma v rámci Travis CI od 1. listopadu 2020
- ▶ výborná dokumentace na docs.travis-ci.com

Co je Travis CI

- ▶ travis-ci.org
- ▶ služba pro automatické CI
- ▶ pouze pro GitHub repozitáře
 - ▶ zdarma pro veřejné repozitáře
 - ▶ placené travis-ci.com pro privátní
 - ▶ zdarma v rámci GitHub Student Developer Pack
- ▶ výborná dokumentace na docs.travis-ci.com

Co je Travis CI

- ▶ travis-ci.org
- ▶ služba pro automatické CI
- ▶ pouze pro GitHub repozitáře
 - ▶ zdarma pro veřejné repozitáře
 - ▶ placené travis-ci.com pro privátní
 - ▶ zdarma v rámci GitHub Student Developer Pack
- ▶ výborná dokumentace na docs.travis-ci.com

Co je Travis CI

- ▶ travis-ci.org
- ▶ služba pro automatické CI
- ▶ pouze pro GitHub repozitáře
 - ▶ zdarma pro veřejné repozitáře
 - ▶ placené travis-ci.com pro privátní
 - ▶ zdarma v rámci GitHub Student Developer Pack
- ▶ výborná dokumentace na docs.travis-ci.com

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

travis.yml

Demo

Co je Travis CI

- ▶ travis-ci.org
- ▶ služba pro automatické CI
- ▶ pouze pro GitHub repozitáře
 - ▶ zdarma pro veřejné repozitáře
 - ▶ placené travis-ci.com pro privátní
 - ▶ zdarma v rámci [GitHub Student Developer Pack](https://github.com/github-education)
- ▶ výborná dokumentace na docs.travis-ci.com

Co je Travis CI

- ▶ travis-ci.org
- ▶ služba pro automatické CI
- ▶ pouze pro GitHub repozitáře
 - ▶ zdarma pro veřejné repozitáře
 - ▶ placené travis-ci.com pro privátní
 - ▶ zdarma v rámci [GitHub Student Developer Pack](https://github.com/github-student-developer-pack)
- ▶ výborná dokumentace na docs.travis-ci.com

Jak začít používat Travis CI

Travis CI

Miro Hrončok

- ▶ **vležete na travis-ci.org**
- ▶ přihlásíte se přes GitHub
- ▶ zapnete webhook pro repozitář
- ▶ přidáte `travis.yml`
- ▶ pushnete

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak začít používat Travis CI

Travis CI

Miro Hrončok

- ▶ vlezete na travis-ci.org
- ▶ přihlásíte se přes GitHub
- ▶ zapnete webhook pro repozitář
- ▶ přidáte `travis.yml`
- ▶ pushnete

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak začít používat Travis CI

- ▶ vlezete na travis-ci.org
- ▶ přihlásíte se přes GitHub
- ▶ zapnete webhook pro repozitář
- ▶ přidáte `travis.yml`
- ▶ pushnete

Jak začít používat Travis CI

- ▶ vlezete na travis-ci.org
- ▶ přihlásíte se přes GitHub
- ▶ zapnete webhook pro repozitář
- ▶ přidáte `travis.yml`
- ▶ pushnete

Jak začít používat Travis CI

Travis CI

Miro Hrončok

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

travis.yml

Demo

- ▶ vlezete na travis-ci.org
- ▶ přihlásíte se přes GitHub
- ▶ zapnete webhook pro repozitář
- ▶ přidáte `travis.yml`
- ▶ pushnete

Jak funguje Travis CI

Travis CI

Miro Hrončok

- ▶ **vývojář pushne nové commity**
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nainstaluje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak funguje Travis CI

Travis CI

Miro Hrončok

- ▶ vývojář pushne nové commity
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nastartuje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak funguje Travis CI

Travis CI

Miro Hrončok

- ▶ vývojář pushne nové commity
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nainstaluje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak funguje Travis CI

Travis CI

Miro Hrončok

- ▶ vývojář pushne nové commity
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nashartuje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak funguje Travis CI

Travis CI

Miro Hrončok

- ▶ vývojář pushne nové commity
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nainstaluje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak funguje Travis CI

Travis CI

Miro Hrončok

- ▶ vývojář pushne nové commity
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nainstaluje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak funguje Travis CI

- ▶ vývojář pushne nové commity
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nainstaluje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak funguje Travis CI

Travis CI

Miro Hrončok

- ▶ vývojář pushne nové commity
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nastartuje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

Jak funguje Travis CI

Travis CI

Miro Hrončok

- ▶ vývojář pushne nové commity
- ▶ GitHub řekne Travisu, že se to stalo
- ▶ Travis CI:
 - ▶ naklonuje vaše repo a přepne se na ten daný commit
 - ▶ přečte `travis.yml`
 - ▶ nainstaluje kontajnery
 - ▶ spustí build a testy
 - ▶ reportuje úspěch/selhání
 - ▶ (udělá deploy)

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

`travis.yml`

Demo

A co ten travis.yml

- ▶ **YAML soubor**
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ například Python
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

A co ten travis.yml

- ▶ YAML soubor
- ▶ definuje:
 - ▶ jazyk projektu
 - ▶ runtimy
 - ▶ např. různé verze Pythonu
 - ▶ co nainstalovat za závislosti
 - ▶ služby a jejich konfigurace
 - ▶ jak to zbuildit
 - ▶ jak pustit testy
 - ▶ jak deploynout
 - ▶ další volby specifické pro daný jazyk

[Co je CI](#)[K čemu CI](#)[Co je potřeba k CI](#)[Co je Travis CI](#)[Jak to použít](#)[Jak to funguje](#)[travis.yml](#)[Demo](#)

travis.yml – příklad

```
language: python
python:
  - "2.6"
  - "2.7"
  - "3.3"
  - "3.4"
  - "pypy"
  - "pypy3"
install:
  - pip install -r requirements.txt
  - if [[ $TRAVIS_PYTHON_VERSION == '2.6' ]]; then...
script: nosetests
```

► příklady pro jiné jazyky

travis.yml – pořadí vykonání sekcí

- ▶ before_install
- ▶ install
- ▶ before_script
- ▶ script
- ▶ after_success nebo after_failure
- ▶ after_script

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

[Co je CI](#)[K čemu CI](#)[Co je potřeba k CI](#)[Co je Travis CI](#)[Jak to použít](#)[Jak to funguje](#)[travis.yml](#)[Demo](#)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – instalace závislostí

- ▶ docs.travis-ci.com/user/installing-dependencies
- ▶ interprety/kompilátory se instalují samy (podle jazyka)
- ▶ služby jsou nainstalované, ale nemusí běžet
- ▶ knihovny pomocí language-specific metod
 - ▶ PyPI
 - ▶ Rubygemns
 - ▶ ...
- ▶ můžete doinstalovávat balíčky pomocí apt-get
- ▶ případně stahovat tary a kompilovat
 - ▶ (jde dělat prakticky cokoliv)

travis.yml – služby

- ▶ docs.travis-ci.com/user/database-setup
- ▶ MySQL, PostgreSQL, MongoDB, Redis, RabbitMQ, Memcached...

```
services:  
  - redis  
  - memcached  
  # mysql defaultně samo  
before_script:  
  # username travis/root, blank password  
  - mysql -e 'create database myapp_test;'
```

travis.yml – deployment

- ▶ docs.travis-ci.com/user/deployment
- ▶ OpenShift, Heroku...
- ▶ PyPI, Rubygems...

deploy:

```
provider: openshift
user: eggs
password: foo
domain: spam
```

- ▶ kontrolní otázka

travis.yml – deployment

- ▶ docs.travis-ci.com/user/deployment
- ▶ OpenShift, Heroku...
- ▶ PyPI, Rubygems...

deploy:

```
provider: openshift
user: eggs
password: foo
domain: spam
```

- ▶ kontrolní otázka

travis.yml – notifikace

- ▶ docs.travis-ci.com/user/notifications
- ▶ e-mail, IRC, webhook...
- ▶ rozumné defaultní chování

```
notifications:
```

```
  irc:
```

```
    channels:
```

```
      - "chat.freenode.net#my-channel"
```

```
      - "chat.freenode.net#some-other-channel"
```

```
    on_success: [always|never|change] # default: change
```

```
    on_failure: [always|never|change] # default: always
```

[Co je CI](#)[K čemu CI](#)[Co je potřeba k CI](#)[Co je Travis CI](#)[Jak to použít](#)[Jak to funguje](#)[travis.yml](#)[Demo](#)

travis.yml – build matrix

- ▶ docs.travis-ci.com/user/build-configuration
- ▶ použít se několik kontajnerů (paralelně)
- ▶ defaultně matice runtime × env (× ještě něco)

rvm:

- 1.8.7
- 1.9.3
- jruby
- ree

gemfile:

- gemfiles/Gemfile.rails-2.3.x
- gemfiles/Gemfile.rails-3.1.x
- gemfiles/Gemfile.rails-edge

env:

- ISOLATED=true
- ISOLATED=false

travis.yml – příklady

- ▶ github.com/rails/rails – komplexní příklad
- ▶ github.com/vrutkovs/dogtail – GTK+ GUI testing
- ▶ github.com/travis-ci-examples – příklady přímo od Travisáků

Demo

Demo

Travis CI

Miro Hrončok

Co je CI

K čemu CI

Co je potřeba k CI

Co je Travis CI

Jak to použít

Jak to funguje

travis.yml

Demo