

Sadržaj

1	Uvod	1
2	Teorijske osnove	2
2.1	Open source razvoj	2
2.2	Python, PyPI	2
2.3	Github	2
2.4	Github akcije - mozda spojiti sa prethodnim	2
2.5	Python paketi	2
2.6	Poetry	2
2.6.1	Podešavanje Poetry-ja	3
2.6.2	Kreiranje Python projekta koji koristi Poetry	3
2.6.3	Kreiranje, pakovanje i objavljivanje Python paketa	4
2.7	Bash	5
3	Specifikacija i implementacija projekta	6
3.1	pjisp-assignment-template	6
3.2	pjisp-template-name	6
3.2.1	način imenovanja repo-a	6
3.3	pjisp-diff	6
3.4	poetry-publish action	6
3.5	Error code u smoke-test-u - zasto je potreban?	6
4	Primeri korišćenja	7
5	Diskusija i zakljuci	8

5.1	Da li su github akcije spremne za produkciju?	8
6	Literatura	9

Spisak slika

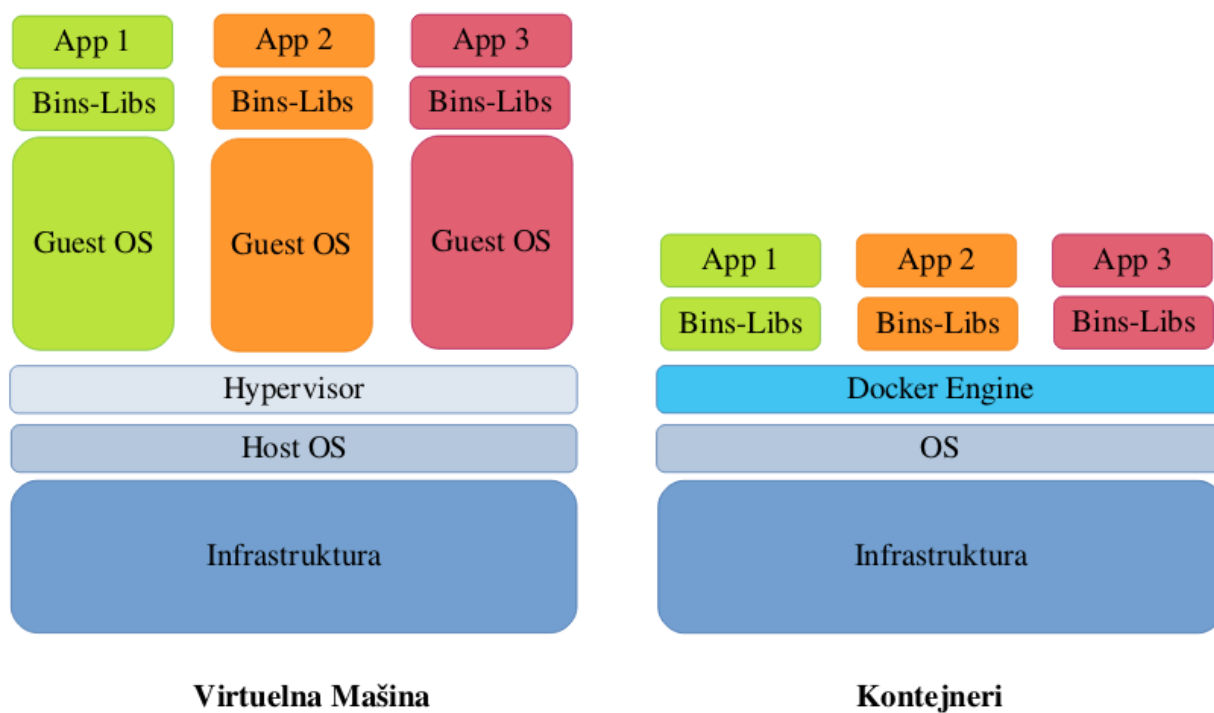
1.1	Razlika između arhitekture Virtuelne Mašine a arhitekture kontejnera	1
-----	--	---

Skraćenice

API – *Application programming interface*, Interfejs za programiranje aplikacija

1. Uvod

U današnje vreme[1]



Slika 1.1: Razlika između arhitekture Virtuelne Mašine a arhitekture kontejnera

2. Teorijske osnove

U ovom poglavlju će biti ukratko objašnjeni svi koncepti i alati koji su bili potrebni za izradu projekta o kojem ovaj rad govori. NABROJ

2.1 Open source razvoj

2.2 Python, PyPI

2.3 Github

2.4 Github akcije - mozda spojiti sa prethodnim

2.5 Python paketi

2.6 Poetry

Poetry [2] je alat otvorenog koda (eng. *open-source*) za kreiranje Python paketa i upravljanje paketima koje Python projekat koristi (eng. *dependency management*). Pomaže korisnicima tako što umesto njih instalira i menja verzije bibliotekama od kojih projekat zavisi, a koje korisnik mora pretkodno da specificira. Poetry je takođe znatno olakšao kreiranje, pakovanje i objavljivanje paketa na PyPI - repozitorijum softvera za Python programski jezik i time omogućio lakše deljenje Python projekata sa drugim korisnicima tog jezika.

Prva verzija alata objavljena je 28.02.2018. godine. Verzije programskog jezika Python koje su

podržane su 2.7 i 3.4+. Ideja je da Poetry radi podjednako dobro na različitim platformama, između ostalog na Linux-u, Windows-u i OSX-u.

2.6.1 Podešavanje Poetry-ja

Za razliku od drugih Python alata za upravljanje paketima od kojih projekat zavisi, umesto pip-a - instalera za Python pakete, Poetry koristi svoj specifičan način za instalaciju. Instalator doda Poetry u korisnički direktorijum, tako da može da se koristi sa bilo kojom verzijom Python-a. Omogućena je i instalacija Poetry-ja korišćenjem pip-a, ali se ne preporučuje iz dva razloga:

1. može dovesti do konflikta sa drugim sistemskim fajlovima
2. otežava održavanje konzistentnosti pre korišćenju različitih verzija Python-a i različitih virtuelnih okruženja (eng. *virtual environments*)

2.6.2 Kreiranje Python projekta koji koristi Poetry

Nakon instalacije alata, moguće je kreirati projekat koji koristi Poetry za upravljanje paketima ili dodati Poetry u postojeći projekat i time kreirati virtuelno okruženje u kojem će se izvršavati naredne akcije. U oba slučaja će se kreirati *pyproject.toml* datoteka koja čuva sve bitne informacije o projektu. U početku ova datoteka sadrži samo osnovne informacije o projektu:

```
[tool.poetry]
name = "poetry-demo"
version = "0.1.0"
description = ""
authors = ["Name Surname <email>"]

[tool.poetry.dependencies]
python = "*"

[tool.poetry.dev-dependencies]
pytest = "^3.4"
```

Informacije o samom projektu, njegovom autoru, verziji, licenci, opisu, dokumentaciji, itd. nalaze se u prvom segmentu - `[tool.poetry]`. Informacije o paketima koje projekat koristi nalaze se u naredne dve sekcije - `[tool.poetry.dependencies]` i

[`tool.poetry.dev-dependencies`], gde se u prvoj od ove dve sekcije nalaze informacije o paketima koji se koriste u produkcionom okruženju, a u drugoj informacije o paketima koji se koriste u toku razvoja projekta. Pri instalaciji novog paketa, njegov naziv i verzija će se dopisati u jednu od ove dve sekcije, ili u obe ako je to potrebno.

Ovu datoteku je moguće ručno menjati, ali se sa njom najčešće interaguje pozivanjem `poetry` komandi u konzoli. Neke od tih komandi su:

- `poetry add` - dodaje novi paket u *pyproject.toml* datoteku
- `poetry remove` - briše paket iz projekta

Komande koje takođe interaguju sa *pyproject.toml* datotekom, ali je ne menjaju su:

- `poetry install` - instalira pakete definisane u *pyproject.toml* datoteci
- `poetry update` - ažurira pakete u skladu sa verzijama koje pišu u *pyproject.toml* datoteci
- `poetry lock` - zaključava pakete u prijektu
- `poetry check` - proverava ispravnost *pyproject.toml* datoteke

Za pokretanje komandi u virtuelnom okruženju opisanom u *pyproject.toml* datoteci koristi se komanda:

- `poetry run`

Za sve ostale detalje najbolje je pogledati dokumentaciju pomoću komande:

- `poetry help`

2.6.3 Kreiranje, pakovanje i objavljivanje Python paketa

Da bi se Python projekat mogao objaviti, potrebno je kreirati arhivu sa svim potrebnim konfiguracijama za njega. To se postiže upotrebom komande:

- `poetry build`

Nakon toga, paket je spreman za objavljivanje. U projektu se pojavi novi direktorijum sa nazivom `dist` i u njemu se nalaze dve datoteke:

1. `poetry-demo-0.1.0-py2.py3-none-any.whl`
2. `poetry-demo-0.1.0.tar.gz`

Repozitorijum na kojem se paket objavljuje se podešava pomoću komande:

- `poetry config`

Paket se objavljuje pozivanjem komande:

- `poetry publish`

U tom momentu, paket postaje dostupan na izabranom repozitorijumu i svako ko ima pristup repozitorijumu može da instalira paket. Najčešće se paketi objavljuju na PyPI repozitorijumu.

2.7 Bash

3. Specifikacija i implementacija projekta

3.1 pjisp-assignment-template

3.2 pjisp-template-name

3.2.1 način imenovanja repo-a

3.3 pjisp-diff

3.4 poetry-publish action

Koristi se za publishovanje pjisp-diff i pjisp-template-name

3.5 Error code u smoke-test-u - zasto je potreban?

4. Primeri korišćenja

5. Diskusija i zakljuci

5.1 Da li su github akcije spremne za produkciju?

6. Literatura

- [1] Docker Inc. *Docker documentation*. URL: [http : / / web . archive . org / web / 20190610115307/https://docs.docker.com/](http://web.archive.org/web/20190610115307/https://docs.docker.com/).
- [2] Poetry. *Poetry documentation*. URL: <https://python-poetry.org>.