

Primus

software deployment package

Oct 2012, Andrei Vishneuski

Outline

- Mission
- Sample of project deployed with Primus helps
- Supported deployment artifacts list
- Questions
- Appendix

Mission

- Has a way to describe a project deployment procedure less talkative way
- Keep deployment script in human readable format
- Make hands free from absolute paths. Make everything in a context of project home
- Centralize configuration properties (for many configs) in one place

Sample project: description

- User has a nice project "BigBang"
- This project is written in python (code in "lib" folder) and java (code in "src" folder)
- Project can work only with Python v2.5+
- Project requires "time.jar" third party library (available in local http repo)
- Project has to be passed through test cases

Sample project: "a user have a nice project"

```
PROJECT=Project("BigBang",  
    [  
        ...  
    ]  
)
```


Sample project: project requires python 2.5+

```
PROJECT=Project("BigBang",  
[  
    CheckPythonEnvironment(version=(2,5))  
])
```


Sample project: deploy third party library

```
PROJECT=Project("BigBang",  
[  
    CheckPythonEnvironment(version=(2,5)),  
    DownloadFile("lib", "time/time1.0.jar")  
])
```


Sample project: compile and validate code code

```
PROJECT=Project("BigBang",  
[  
    CheckPythonEnvironment(version=(2,5)),  
    DownloadFile("lib", "time/time1.0.jar"),  
    CompileJavaCode("src", "lib"),  
    ValidatePythonCode("lib")  
])
```


Sample project: run java test cases

```
PROJECT=Project("BigBang",  
[  
    CheckPythonEnvironment(version=(2,5)),  
    CompileJavaCode("src", "lib"),  
    ValidatePythonCode("lib"),  
    DownloadFile("lib", "time/time1.0.jar"),  
    RunJavaCode("lib", "universe.bigbang.Test")  
]  
)
```


Sample project: play deployment script

project layout

[<BigBang home>]

|_lib] # compiled code and jars

|_src] # java source code

|_.primus] # that the place for deployment stuff

| |_project.py # stored configuration

| |_deploy.py # script to play the configuration

| |_deploy.conf # all project config properties

...

\$python <BigBang home>/deploy/deploy.py

Let's make life more complicated: dependencies

```
PROJECT=Project("BigBang",  
[  
    RunJavaCode("lib", "universe.bigbang.Test",  
    [  
        CompileJavaCode("src", "lib", [  
            DownloadFile("lib", "time/time1.0.jar")  
        ])  
    ]),  
    ValidatePythonCode("lib", [  
        CheckPythonEnvironment(version=(2,5))  
    ])  
])
```


Available Primus artifacts

- MySQL artifacts (create DB, load schema)
- Configuration artifact (adjust config params)
- Conditionals artifacts (if, if not, group)
- Repository artifacts (http and local repo)
- Python/java artifacts (run, compile, check)
- File operations artifact (copy, unzip, remove)

Questions



Appendix

- Condition artifacts usage
- Extending Primus with a new artifacts
- Sample of Primus output

Condition artifact: optional deployment actions

```
PROJECT=Project("BigBang",  
[  
    DoIfNot(TestDBExistance(...), [  
        CreateDB(...),  
        LoadDBSchema(..., "db/shema.sql")  
    ])  
])
```


Extending Primus

```
class AddLicenseHeader(PermanentFileArtifact):  
    # implement the two methods below  
    def build(self):  
        ...  
    def what_it_does(self):  
        return "Add license header to %s" % self.name  
  
    # optional part, override if necessary  
    def is_expired(self): ...  
    def cleanup(self): ...
```


Sample of Primus output

```
:: Primus - python deployment package  
:: v1.4.0, Oct. 2012  
:: target project home: '/home/avishneu/projects/tae'
```

Deploy project 'TAE' v1.0.0

Test environment:

Check Python version ≥ 2.5

Validate Python code in 'lib'

Download third party software:

Download and unzip 'lib/quartz/ner' package

Download and unzip 'lib/quartz/tnt' package

Download and unzip 'lib/quartz/mbasp' package

Setup python package: 'lib/ilps/tae/match'

Compile Java code from 'lib' into 'lib' dir.

DEPLOYMENT SUCCESS:

Project 'TAE' has been successfully deployed.