Defect Analysis of Infrastructure as Code (IaC)

Akond Rahman

Adviser: Dr. Laurie Williams
Project Overview

- Hard problem area: Resilient architectures
- Maturity: middle stage
- Motivation: IaC scripts can be complex in nature containing hundreds of lines of code, exposing them to defects that are potentially difficult to debug.
Research Goal

The goal of this research study is to help software practitioners in discovering potential bad patterns in IaC scripts through detection of IaC defects using IaC metrics.
Research Questions

• RQ1: What categories of IaC defects exist? How frequently do these defects occur?
• RQ2: What patterns in IaC scripts contribute to defects? How can these patterns be used to predict IaC defects?
Hypotheses

• H1: Multiple categories of defects exist within IaC scripts
• H2: Certain patterns exist within IaC scripts that contribute to defects
Success Criteria

• Empirical evidence of defect categorization for IaC scripts
• High accuracy of the prediction model built using patterns of IaC scripts
Metrics

• Code metrics
  – Code patterns of IaC scripts
  – General metrics: lines of code, size, static analysis alerts

• Prediction performance
  – Precision, recall, F-measure
Data

• Open source repositories
  – Mozilla release engineering
  – Openstack
  – Wikimedia
Research Design

• Collect repositories
• Filter repertories
• Identify commits related to IaC scripts
• Detect commits related to defects
• Categorize defects
• Extract code metrics
• Create prediction model using extracted code metrics
Research Design (Contd.)

• Detect commits related to defects and categorize defects
  – Human subject involvement
  – Coder#1 will analyze all the commits of all selected repositories to identify defects and categorize them
  – A 50% simple random sample of all the commits will be distributed among voluntary coders
  – Inter rater reliability: overall and per category
  – Coder#1 and all voluntary coders will use ODC for categorization
Analysis

• Prediction Model
  – Feature selection
  – Statistical learners
  – Sampling techniques
  – Hyper-parameter tuning of statistical learners
IaC Defects: Example-1

- Dependency

```python
$file = "${python35::virtualenv::settings::misc_python_dir}/pip-check.py"

file {
  "$file":

-    source => "puppet:///modules/python/pip-check.py",
+    source => "puppet:///modules/python35/pip-check.py",
```
IaC Defects: Example-2

• Control & data flow

Bug 1292523 - Puppet fails to set root password on buildduty-tools server

Status: RESOLVED FIXED

Reported: 2016-08-05 05:23 PDT
Modified: 2016-08-05 06:01 PDT
CC List: 1 user (show)
IaC Defects: Example-3

- Syntax

**Bug 1281992 - puppet-lint 2.0.0 breaks our travis-ci checks for puppet**

- **Status:** RESOLVED FIXED
- **Reported:** 2016-06-23 21:53 PDT
- **Modified:** 2016-06-23 22:16 PDT
- **CC List:** 1 user (show)
- **Keywords:**
- **Product:** Release Engineering (show info)
- **Component:** Buildduty (show other bugs) (show info)
- **See Also:**
IaC Defects: Example-4

• Non functional

create a separate wikidata main page

works around the default main page being in the main namespace, which is supposed to be for wikibase entities.

bug: t128466
Prediction

\[ F_1, F_2, F_3, \ldots, F_n \]

\[ SF_1, SF_2, SF_3, \ldots, SF_n \]

Feature Selection \rightarrow Prediction \rightarrow IaC Defect