

# Bolted System: Auto-deployment Cloud Project

Vidya Anandamurali

Pei Jia

Yuxi Jiang

Jiangnan Zou

# Project Description (Recap)

Automate the deployment of Bolted which consists of:

- Installation and configuration of each component of the Bolted system (HIL, BMI, Keylime and orchestration)
- Integrating all the components together
- Bolted system should be able to be installed on any cloud platform

# Use Case

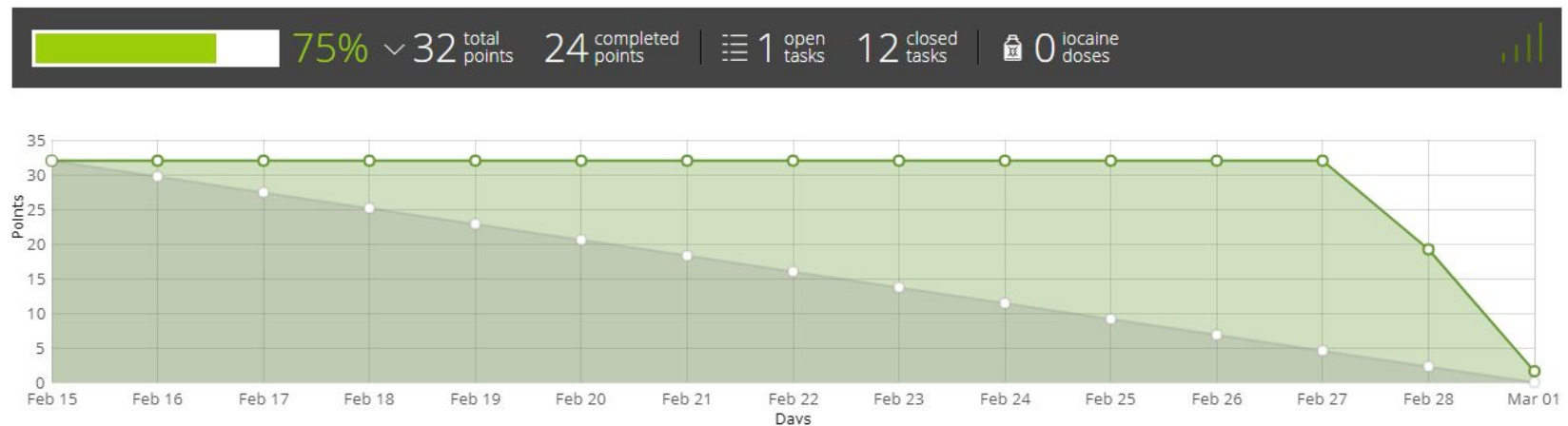
As a cloud provider, I want to deploy Bolted System on my cloud platform easily without having to call upon a personnel to configure the components.

## Minimum Viable Product

- To deliver a system that doesn't require separate installation and configuration procedure for each component of the bolted system
- Develop an ansible playbook for each component first and then an overall playbook for all four components to work together.

# Burndown Chart

2018 BUCS528 SECURE CLOUD AUTOMATED DEP... BU CS 528 CLOUD COMPUTING - DEMO 2 15 FEB 2018-01



# Last Sprint Report

- Install HIL on Virtual Machine (CentOS/VMware)
  - ✓ Install through terminal command line
  - Automate installation using Ansible
    - Unittest
- Install Keylime on Virtual Machine (CentOS/VMware)
  - ✓ Install through terminal command line
  - ✓ Bash script automated installation
  - Test script in another virtual machine
- Install BMI on Virtual Machine (CentOS/VMware)
  - Install BMI on a CentOS environment contains HIL
    - Automate installation using Ansible
    - Unittest
- ✓ Learning Ansible

# Project Progress of Keylime

- TPM (Trusted Platform Module) This is the core module of Keylime, because keylime uses this module to collect node information, (operating system, application and etc), using [quote].
- For now, installation is under virtual machine, which doesn't contain a TPM module. Testing is needed for later test.
- A bash script is developed to automated the installation procedure, targeted for [CentOS VM using vmware]
- Testing passed, bash to ansible script is in progress.

# Project Progress of HIL

- Installing CentOS on VMware workstation.
- Setting up the server upon which HIL runs. (epel and python - using pip)
- Configure HIL (hil.cfg)
- Setting up the HIL database
- Starting the server
- Testing the setup

```

(.venv)[Vidya@rhc1918 hil]$ cd hil/
(.venv)[Vidya@rhc1918 hil]$ ls
api.py          cli.py          dev_support.pyc  __init__.py     network_allocator.py
api.pyc         cli.pyc         errors.py        __init__.pyc    network_allocator.pyc
auth.py         commands        errors.pyc       migrations       rest.py
auth.pyc        config.py       ext              migrations.py   rest.pyc
class_resolver.py  config.pyc     flaskapp.py      migrations.pyc  server.py
class_resolver.pyc deferred.py     flaskapp.pyc     model.py        server.pyc
client          dev_support.py hil.db           model.pyc       test_common.py
(.venv)[Vidya@rhc1918 hil]$ which hil
~/hil/.venv/bin/hil
(.venv)[Vidya@rhc1918 hil]$ _

```



# Project Progress of BMI

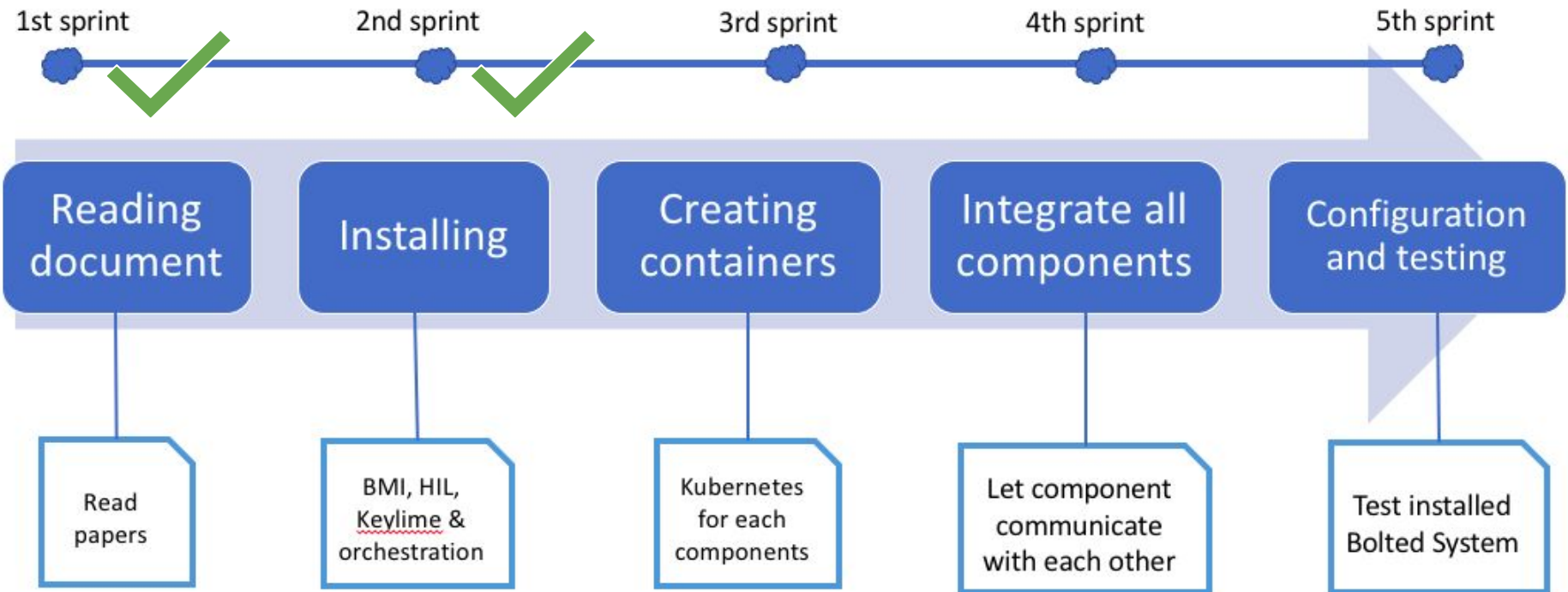
- Set up CentOS 7.0 on virtual machine
- HIL environment checking
- Installing the Ceph Client
  - Learn Ceph server and client simultaneously
- Configuring iSCSI Server
  - Deal with system version inconsistency
- Configuring DHCP Server
- Installing BMI
  - Did not settle File missing problem yet

# iSCSI Server Demo

```
[Vidya@rfc1918 ~]$ systemctl status tgtd
■ tgtd.service - tgtd iSCSI target daemon
   Loaded: loaded (/usr/lib/systemd/system/tgtd.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2018-03-01 13:35:25 EST; 28min ago
   Process: 1482 ExecStartPost=/usr/sbin/tgtadm --op update --mode sys --name State -v ready (code=exited, status=0/SUCCESS)
   Process: 1476 ExecStartPost=/usr/sbin/tgt-admin -e -c $TGTD_CONFIG (code=exited, status=0/SUCCESS)
   Process: 1475 ExecStartPost=/usr/sbin/tgtadm --op update --mode sys --name State -v offline (code=exited, status=0/SUCCESS)
   Process: 1075 ExecStartPost=/bin/sleep 5 (code=exited, status=0/SUCCESS)
  Main PID: 1074 (tgtd)
    CGroup: /system.slice/tgtd.service
            └─1074 /usr/sbin/tgtd -f

Mar 01 13:35:20 rfc1918.address.not.used.bu.edu systemd[1]: Starting tgtd iSCSI target daemon...
Mar 01 13:35:20 rfc1918.address.not.used.bu.edu tgtd[1074]: tgtd: iser_ib_init(3436) Failed to i...?
Mar 01 13:35:20 rfc1918.address.not.used.bu.edu tgtd[1074]: tgtd: work_timer_start(146) use time...r
Mar 01 13:35:20 rfc1918.address.not.used.bu.edu tgtd[1074]: tgtd: bs_init_signalfd(267) could no...e
Mar 01 13:35:20 rfc1918.address.not.used.bu.edu tgtd[1074]: tgtd: bs_init(386) use signalfd noti...n
Mar 01 13:35:25 rfc1918.address.not.used.bu.edu systemd[1]: Started tgtd iSCSI target daemon.
Hint: Some lines were ellipsized, use -l to show in full.
```

# Project Plan



# Responsibilities for next sprint

1. Finish installation process of each component and start configuring communication between each component on local machine
2. Write ansible playbook script for automated installation of each component.
3. Test installation result on local virtual machine
4. Test ansible script on server
5. Create containers and install each component in container using Kubernetes

**Thank you.**

Question ?