



**Whamcloud**

# IML Roadmap and Community

Joe Grund

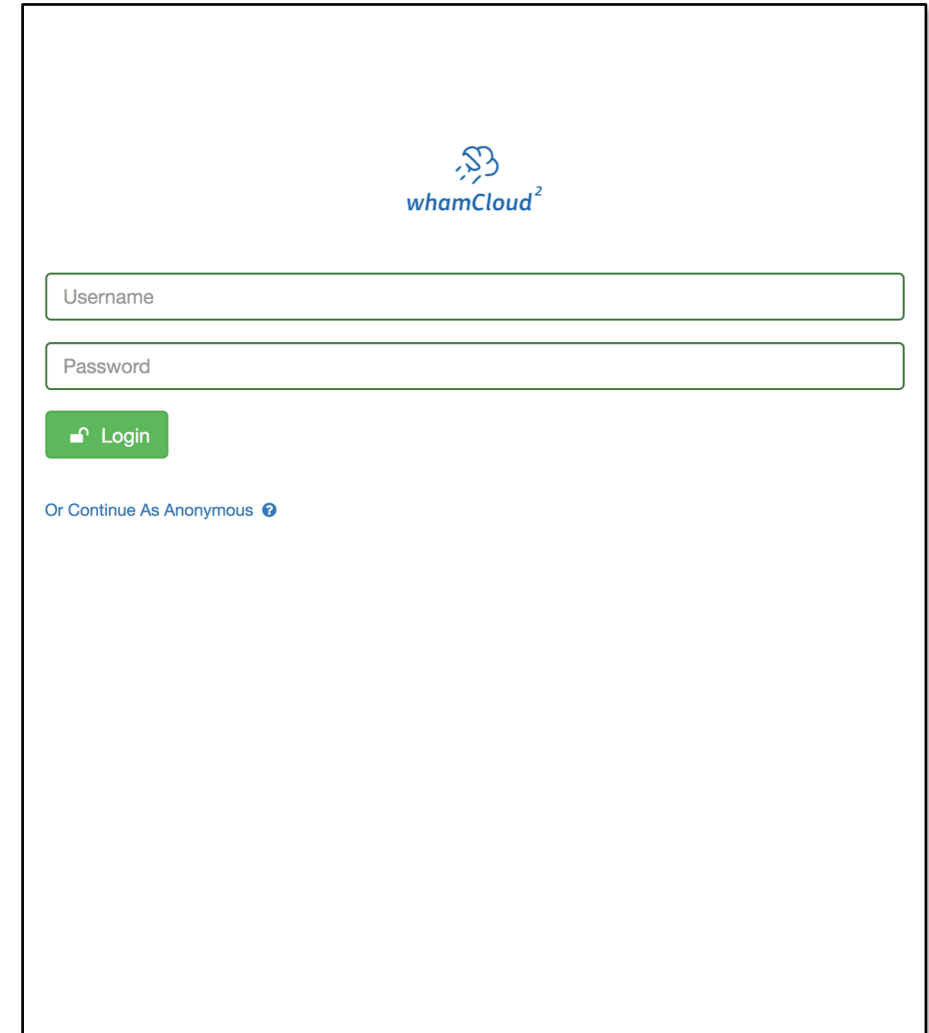
IML Team Lead

[jgrund@whamcloud.com](mailto:jgrund@whamcloud.com)



# Agenda

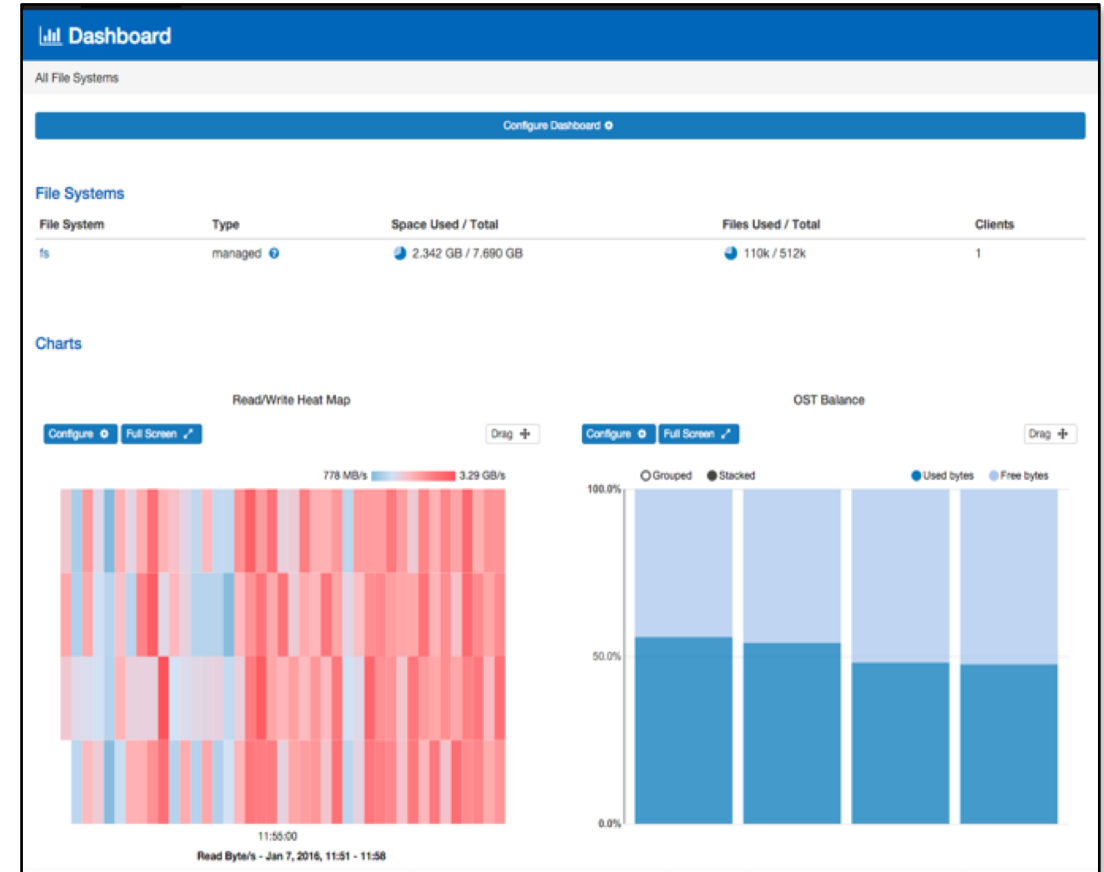
- ▶ IML Background / Overview
- ▶ Current Work
- ▶ Potential Future Work
- ▶ Where to find project / communicate with team



The screenshot shows the WhamCloud login interface. At the top center is the WhamCloud logo, which consists of a stylized blue icon of three interlocking loops above the text "whamCloud<sup>2</sup>". Below the logo are two input fields: "Username" and "Password". The "Password" field has a small eye icon on the right side, indicating a toggle for password visibility. Below the input fields is a green "Login" button with a white key icon. At the bottom of the login area, there is a link that says "Or Continue As Anonymous" followed by a small question mark icon.

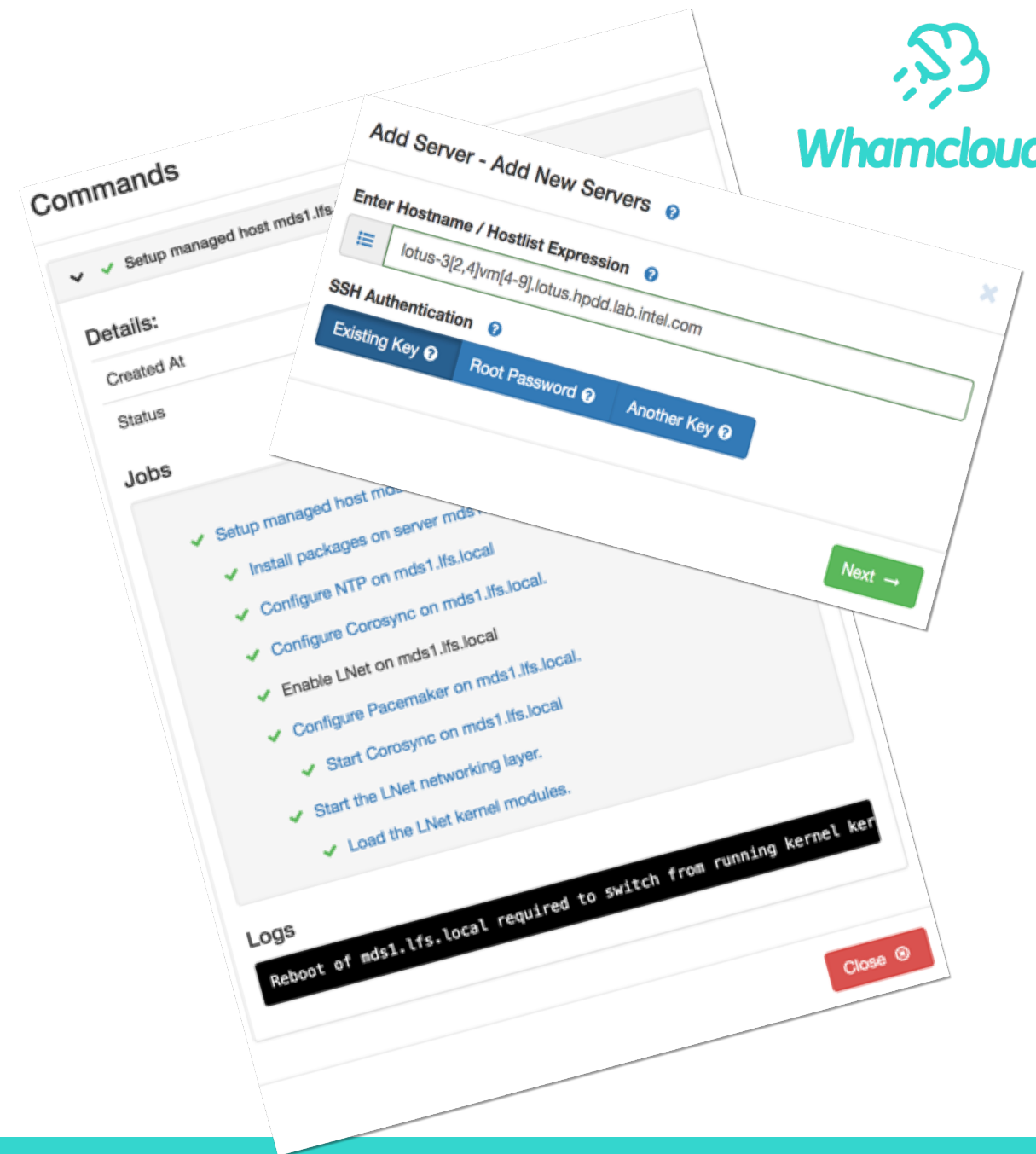
# Background

- ▶ Integrated Manager for Lustre (IML) is an open source suite of tools for deploying, managing, and monitoring Lustre filesystems
- ▶ IML simplifies Lustre administration with intuitive interfaces and near real-time feedback
- ▶ Works with new and existing Lustre installations
- ▶ Monitors performance and system health
- ▶ Proven in production at hundreds of sites
- ▶ Used successfully in environments with over 100 OSTs.



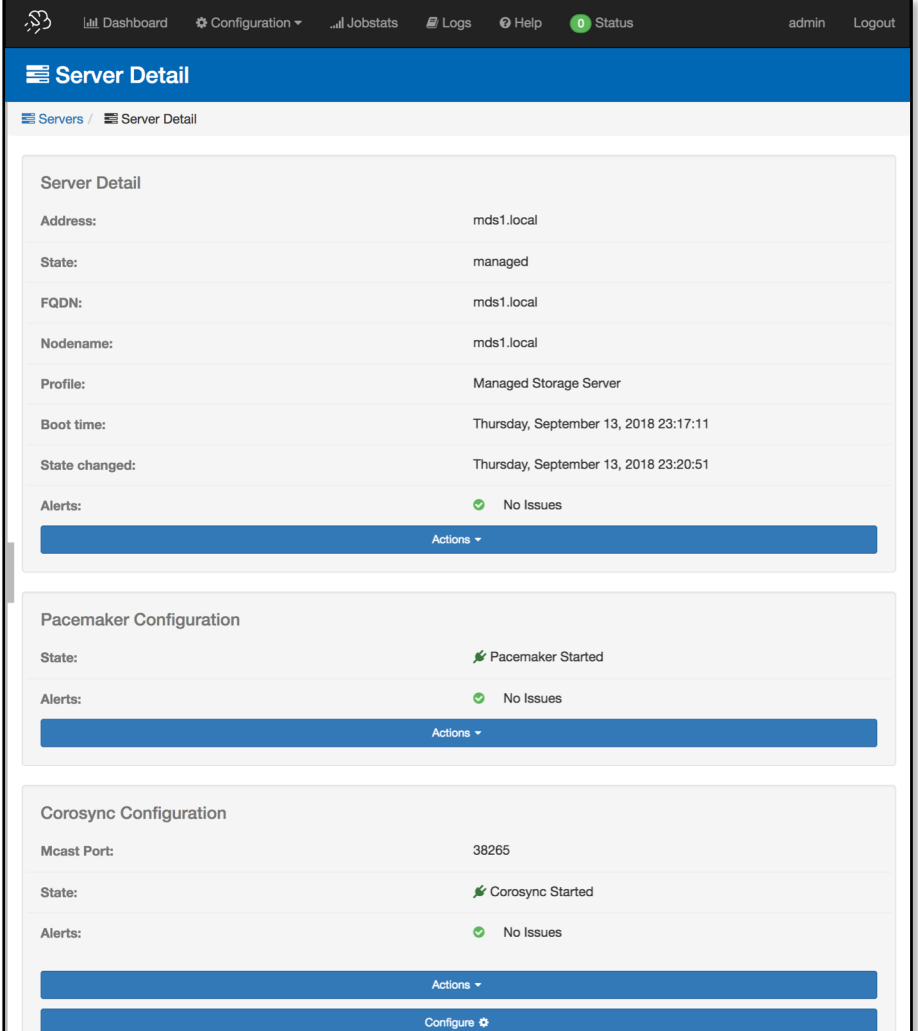
# Background - Deployment

- ▶ Deploy Lustre filesystems from one centralized location
- ▶ Near-realtime feedback
- ▶ Bring filesystem online from first principles or deploy monitoring for an existing filesystem
- ▶ Deploy specialized assets, HSM
- ▶ Add more assets over time



# Background - Management

- ▶ Configure / change state of Lustre and related components
  - Uses state-machine to reach end state from different starting points
    - Starting LNet, state machine ensures packages are installed + kernel modules loaded before bringing LNet up
- ▶ Handle recovery situations fencing, failover
  - Automatic configuration of High Availability through Corosync, Pacemaker, and PDU / IPMI integration



The screenshot displays the Whamcloud management interface. At the top, there is a navigation bar with links for Dashboard, Configuration, Jobstats, Logs, Help, and Status. The main content area is titled "Server Detail" and shows the following information:

Address:	mds1.local
State:	managed
FQDN:	mds1.local
Nodename:	mds1.local
Profile:	Managed Storage Server
Boot time:	Thursday, September 13, 2018 23:17:11
State changed:	Thursday, September 13, 2018 23:20:51
Alerts:	✓ No Issues

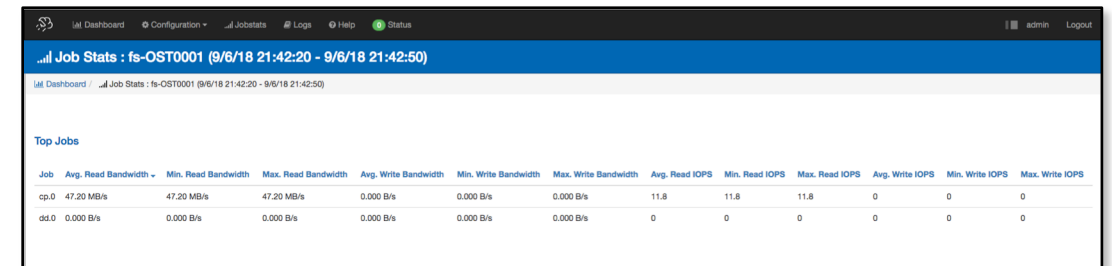
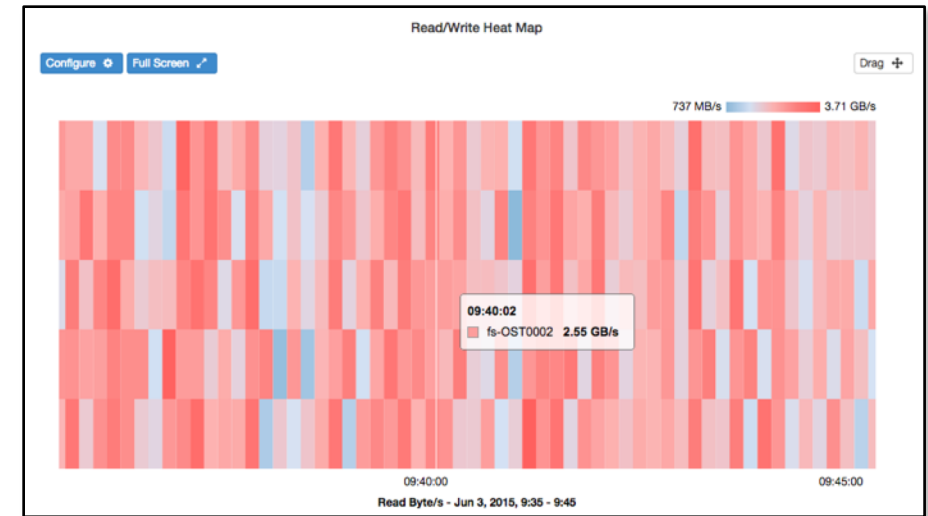
Below the server details, there are three configuration sections, each with its own "Alerts" and "Actions" bar:

- Pacemaker Configuration:** State: ✓ Pacemaker Started; Alerts: ✓ No Issues
- Corosync Configuration:** Mcast Port: 38265; State: ✓ Corosync Started; Alerts: ✓ No Issues

At the bottom of the interface, there are "Actions" and "Configure" buttons.

# Background - Monitoring

- ▶ Holistic system metrics
  - Rich visualizations
  - Drill into filesystem, target, server
  - Find and monitor top jobs
- ▶ Aggregate logs across cluster
- ▶ HSM Copytool activity monitoring
- ▶ Alerts to cluster issues
  - GUI / Email / API
- ▶ Searchable command / event / alert log / history



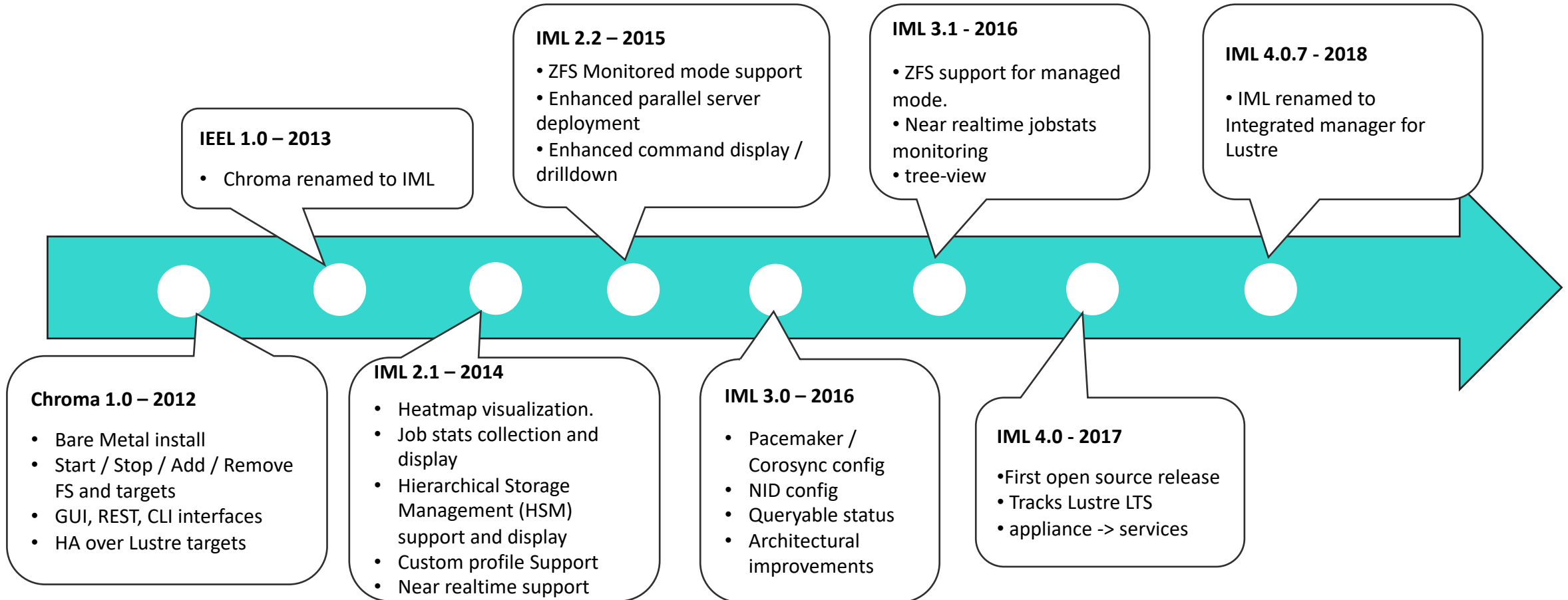
Job Stats : fs-OST0001 (9/6/18 21:42:20 - 9/6/18 21:42:50)

Top Jobs

Job	Avg. Read Bandwidth	Min. Read Bandwidth	Max. Read Bandwidth	Avg. Write Bandwidth	Min. Write Bandwidth	Max. Write Bandwidth	Avg. Read IOPS	Min. Read IOPS	Max. Read IOPS	Avg. Write IOPS	Min. Write IOPS	Max. Write IOPS
cp.0	47.20 MB/s	47.20 MB/s	47.20 MB/s	0.000 B/s	0.000 B/s	0.000 B/s	11.8	11.8	11.8	0	0	0
dsd.0	0.000 B/s	0.000 B/s	0.000 B/s	0.000 B/s	0.000 B/s	0.000 B/s	0	0	0	0	0	0

The screenshot shows a web interface for monitoring job statistics. The main heading is "Job Stats : fs-OST0001 (9/6/18 21:42:20 - 9/6/18 21:42:50)". Below this, there is a section titled "Top Jobs" which contains a table with 13 columns: Job, Avg. Read Bandwidth, Min. Read Bandwidth, Max. Read Bandwidth, Avg. Write Bandwidth, Min. Write Bandwidth, Max. Write Bandwidth, Avg. Read IOPS, Min. Read IOPS, Max. Read IOPS, Avg. Write IOPS, Min. Write IOPS, and Max. Write IOPS. Two jobs are listed: "cp.0" and "dsd.0". The "cp.0" job shows high read bandwidth (47.20 MB/s) and IOPS (11.8), while "dsd.0" shows zero activity across all metrics.

# Background - Development History



# Background - Upgradability



- ▶ Support upgrades from closed-source IEEL versions to open-source Whamcloud versions
  - Documents describe how to upgrade from
    - 2.4.x [https://whamcloud.github.io/Online-Help/docs/Upgrade Guide/Upgrade EE-2.4-el6 to LU-LTS-el7.html](https://whamcloud.github.io/Online-Help/docs/Upgrade%20Guide/Upgrade%20EE-2.4-el6%20to%20LU-LTS-el7.html)
    - 3.1.x [https://whamcloud.github.io/Online-Help/docs/Upgrade Guide/Upgrade EE-3.1-el7 to LU-LTS-el7.html](https://whamcloud.github.io/Online-Help/docs/Upgrade%20Guide/Upgrade%20EE-3.1-el7%20to%20LU-LTS-el7.html)



# Current Work

- ▶ Increase scalability of device detection
- ▶ Increase modularity of components
- ▶ Start moving towards separate management / monitoring code paths
- ▶ Add standard deployment with Docker
- ▶ Updated metrics

# Current Work

## ▶ libzfs integration

- <https://github.com/whamcloud/integrated-manager-for-lustre/issues/535>

## ▶ ZED integration

- <https://github.com/whamcloud/integrated-manager-for-lustre/issues/536>

## ▶ Reactive Architecture

- <https://github.com/whamcloud/integrated-manager-for-lustre/issues/533>

## ▶ Full Modularity

- <https://github.com/whamcloud/integrated-manager-for-lustre/issues/534>

## ▶ Docker support

- <https://github.com/whamcloud/integrated-manager-for-lustre/issues/705>

## ▶ Re-worked metrics

- <https://github.com/whamcloud/lustre-monitor/issues/1>

# Current Work - libzfs integration

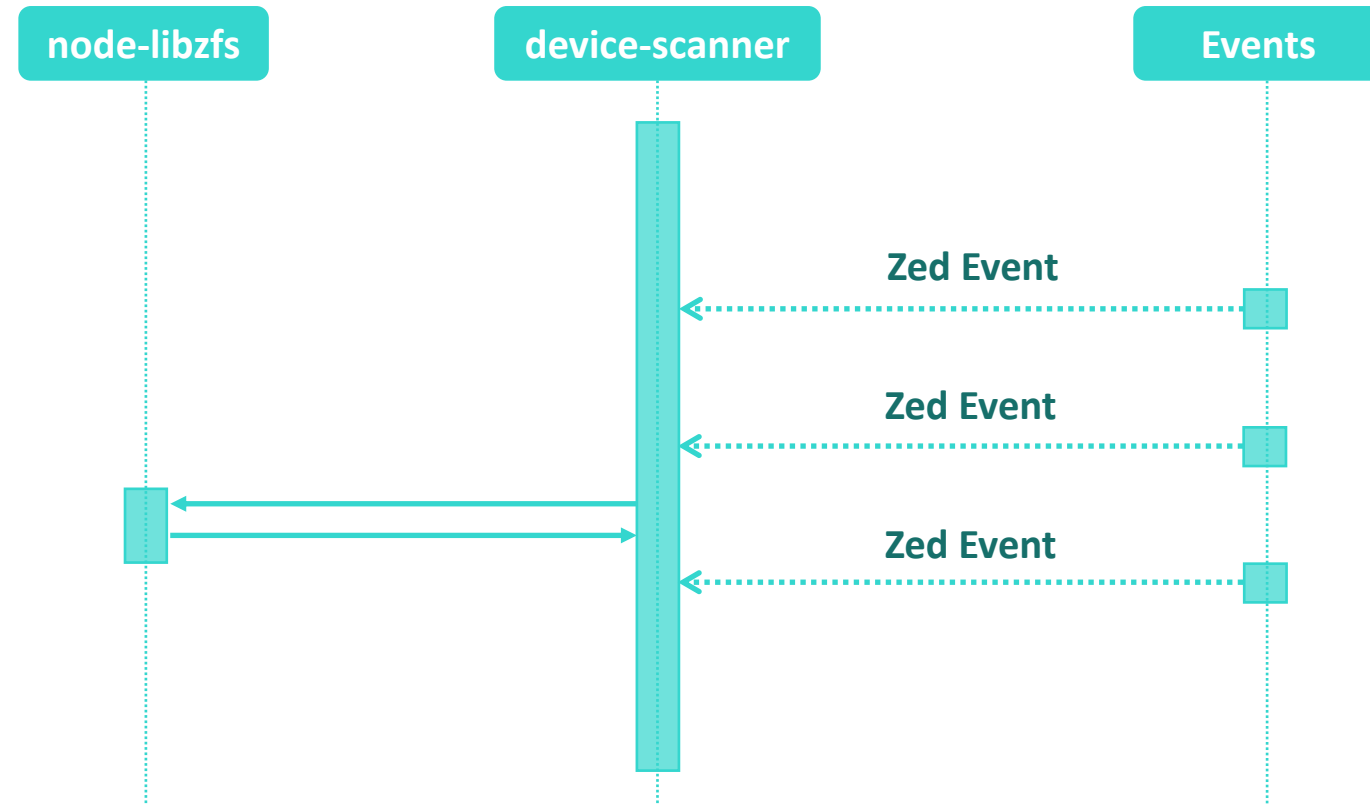


- ▶ IML looking to utilize libzfs for multiple purposes
  - Lower level interface over invoking commands / parsing CLI output
  - Fine grained collection of pools / datasets / props / VDEV tree
    - easy to collect more data later
  - Useful for monitoring + management enhancements

```
[
{
  "name": "test",
  "guid": "14919184393193585238",
  "health": "ONLINE",
  "hostname": "localhost.localdomain",
  "hostid": 3914625515,
  "state": "ACTIVE",
  "readonly": false,
  "size": 83886080,
  "vdev": {
    "Root": {
      "children": [
        {
          "Mirror": {
            "children": [
              {
                "Disk": {
                  "guid": "0xBE4606AF1C39DC3F",
                  "state": "ONLINE",
                  "path": "/dev/sdb1",
                  "dev_id": "ata-VBOX_HARDDISK_081118FC1221NCJ6G8G1-part1",
                  "phys_path": "pci-0000:00:0d.0-ata-2.0",
                  "whole_disk": true,
                  "is_log": null
                }
              },
              {
                "Disk": {
                  "guid": "0xCC43D91716DA2522",
                  "state": "ONLINE",
                  "path": "/dev/sdc1",
                  "dev_id": "ata-VBOX_HARDDISK_081118FC1221NCJ6G8G2-part1",
                  "phys_path": "pci-0000:00:0d.0-ata-3.0",
                  "whole_disk": true,
                  "is_log": null
                }
              }
            ]
          },
          "is_log": false
        }
      ]
    }
  }
},
],
]
```

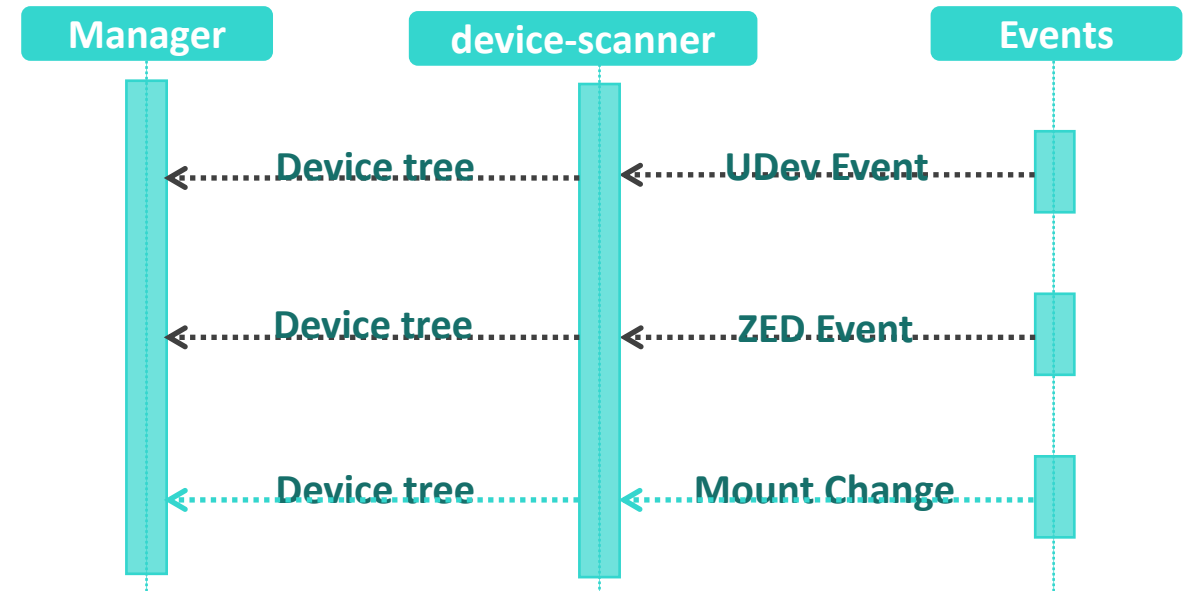
# Current Work - ZED Integration

- ▶ IML looking to use ZED in for multiple purposes
  - Discovery of pool / dataset / property / VDEV changes
    - This is currently a manual scanning process
    - Will allow for closer to realtime changes to propagate
    - Better scaling (very fast for very large sites)
- ▶ Surfacing alerts in the IML GUI / API
  - Alerting on critical events across a cluster
  - Searchable history of all events across a cluster



# Current Work - Reactive Architecture

- ▶ IML looking to flip device discovery from push to pull
  - Adds further scalability
  - Has lower resource usage
  - More responsive
  - Current iteration uses polling + serial introspection of devices

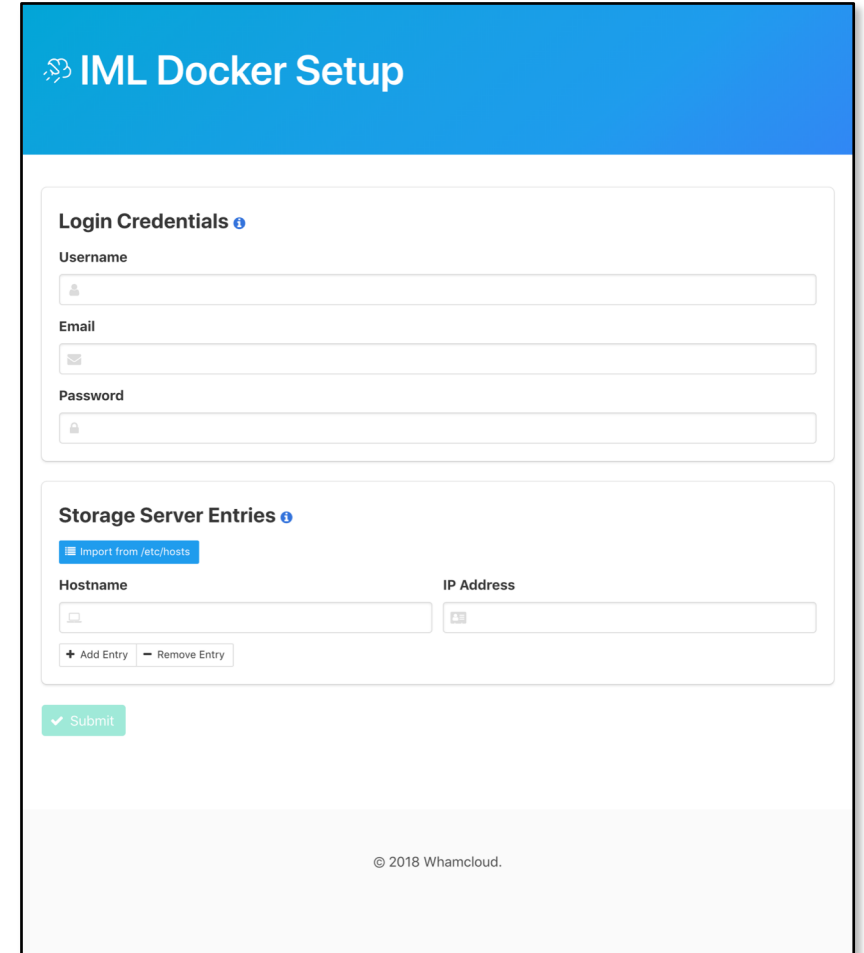


# Current Work - Modularity

- ▶ IML looking to deliver itself completely via RPM
  - No more tarball
  - Ship everything via Fedora Copr
- ▶ **Benefits**
  - Semver minor updates via yum update
  - Components evolve independently
- ▶ **Approach**
  - Two repos, devel + non-devel
  - devel gets continuous updates
  - non-devel gets production ready packages.
- ▶ Those not wanting stream of updates can disable upgrades / perform offline install / upgrade

# Current Work - Docker Support

- ▶ Adding support for docker on the manager
  - Can run IML manager on any OS that supports latest Docker
- ▶ Images built / available on docker cloud
  - <https://cloud.docker.com/swarm/imlteam/repository/list>
- ▶ Provide an (optional) installer that fully configures IML upon first install.
- ▶ Future possibility of HA for IML manager + collocation on storage servers



The screenshot shows the 'IML Docker Setup' web interface. It features a blue header with the Whamcloud logo and the title 'IML Docker Setup'. Below the header, there are two main sections: 'Login Credentials' and 'Storage Server Entries'. The 'Login Credentials' section includes input fields for Username, Email, and Password. The 'Storage Server Entries' section includes a button to 'Import from /etc/hosts', input fields for Hostname and IP Address, and buttons for '+ Add Entry' and '- Remove Entry'. At the bottom of the form is a green 'Submit' button. The footer of the page contains the copyright notice '© 2018 Whamcloud.'

# Current Work - Re-worked Metrics

- ▶ Increasing range of Lustre versions supported
- ▶ Solution: utilize `lctl get_param + lnetctl export`
  - Create a standalone binary that only aggregates stats and outputs in JSON / YAML (other formats to follow)
  - Reusable tool, zero runtime dependencies
- ▶ Solution: utilize timeseries database + Grafana dashboard
  - Allow for stat storage and display unassociated with IML
  - Allow for operators to create custom charts as needed
  - Embed charts back into IML dashboard

```
[root@oss1 tmp]# ./lustre_collector | jq
[
  {
    "param": "memused",
    "value": 80418885
  },
  {
    "param": "memused_max",
    "value": 80499061
  },
  {
    "param": "lnet_memused",
    "value": 15568477
  },
  {
    "param": "health_check",
    "value": "healthy"
  },
  {
    "kind": "OST",
    "param": "job_stats",
    "target": "fs-OST0001",
    "value": null
  },
  {
    "kind": "OST",
    "param": "job_stats",
    "target": "fs-OST0003",
    "value": [
      {
        "job_id": "cp.0",
        "snapshot_time": 1537073485,
        "read_bytes": {
          "samples": 179,
          "unit": "bytes",
          "min": 4194304,
          "max": 4194304,
          "sum": 750780416
        },
        "write_bytes": {
          "samples": 0,
          "unit": "bytes",
          "min": 0,
          "max": 0,

```



# Potential Future Work

## ▶ Enhanced Deployment

- IML should make it even easier to setup Lustre
  - Deploy large scale clusters with minimal operator intervention
    - Describe ideal cluster state
    - Expose variants as scalable UI widgets
    - Deploy installation in parallel with a single click

## ▶ Lustre Snapshot Management

- IML should be able to manage Lustre snapshots via GUI
  - Schedule snapshots for filesystems at some regular interval
  - Ad-hoc snapshot on filesystem(s)
  - View / delete previous snapshots
  - Rollback to a given snapshot
  - Rename a snapshot
  - Take snapshot at key points (i.e. Lustre upgrade)

# Potential Future Work

## ▶ Full ZFS Management

- IML should provide full ZFS management
  - Show all pools and datasets across a cluster
  - Provide drill-down navigation to elicit more detail on a selected target
  - Show the status of pools and datasets
    - Where imported, mounted, error conditions, configuration
  - Management
    - Create zpools / datasets
    - Support creation of various pool configurations: RAID-Z, Mirrored...
    - JBOD enclosure GUI

## ▶ I18n Support

- IML text currently English, but IML is used all over the world
- Modify/contribute \* .po files consumed by services

# Where to find IML

- ▶ 4.0.x Releases: <https://github.com/whamcloud/integrated-manager-for-lustre/releases>
- ▶ RPMs: <https://copr.fedorainfracloud.org/coprs/managerforlustre/>
- ▶ Help docs: <https://whamcloud.github.io/Online-Help/>
- ▶ Issues: <https://github.com/whamcloud/integrated-manager-for-lustre/issues>
- ▶ Direct line of communication via: <https://gitter.im/whamcloud/integrated-manager-for-lustre>
- ▶ Demo sandbox through vagrant: <https://github.com/whamcloud/Vagrantfiles/>
- ▶ Email: [iml@whamcloud.com](mailto:iml@whamcloud.com)

# Closing



- ▶ IML is a project with a long history and continues advancing
  - In production at hundreds of sites worldwide
  - IML 4.0 was first open source release in Oct 2017
  - Eight maintenance updates to 4.0 release
  - Next release IML 4.1 brings even more enhancements
  - Continue iterating / evolving
- ▶ Possible to upgrade from IEEL to IML 4
  - Upgrade docs for both 2.4.x and 3.1.x lines
  - [https://whamcloud.github.io/Online-Help/docs/Upgrade\\_Guide/Upgrade\\_EE-2.4-el6\\_to\\_LU-LTS-el7.html](https://whamcloud.github.io/Online-Help/docs/Upgrade_Guide/Upgrade_EE-2.4-el6_to_LU-LTS-el7.html)
  - [https://whamcloud.github.io/Online-Help/docs/Upgrade\\_Guide/Upgrade\\_EE-3.1-el7\\_to\\_LU-LTS-el7.html](https://whamcloud.github.io/Online-Help/docs/Upgrade_Guide/Upgrade_EE-3.1-el7_to_LU-LTS-el7.html)



**Whamcloud**