Uniface in the cloud

Michael Taylor
Product Manager/Product Owner

November 17, 2017

UNIFACE
Advanced Development Technology

Agenda

- Roadmap
- The story so far
- Next steps
- Final thoughts

What is the cloud?

Cloud Roadmap – a phased approach

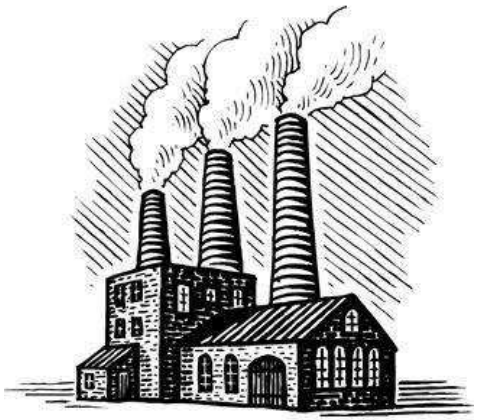
- Phase 1 - Support Linux deployment
- Phase 2 - Add Windows deployment
- Phase 3 – Containerize Uniface
- Phase 4 - PaaS

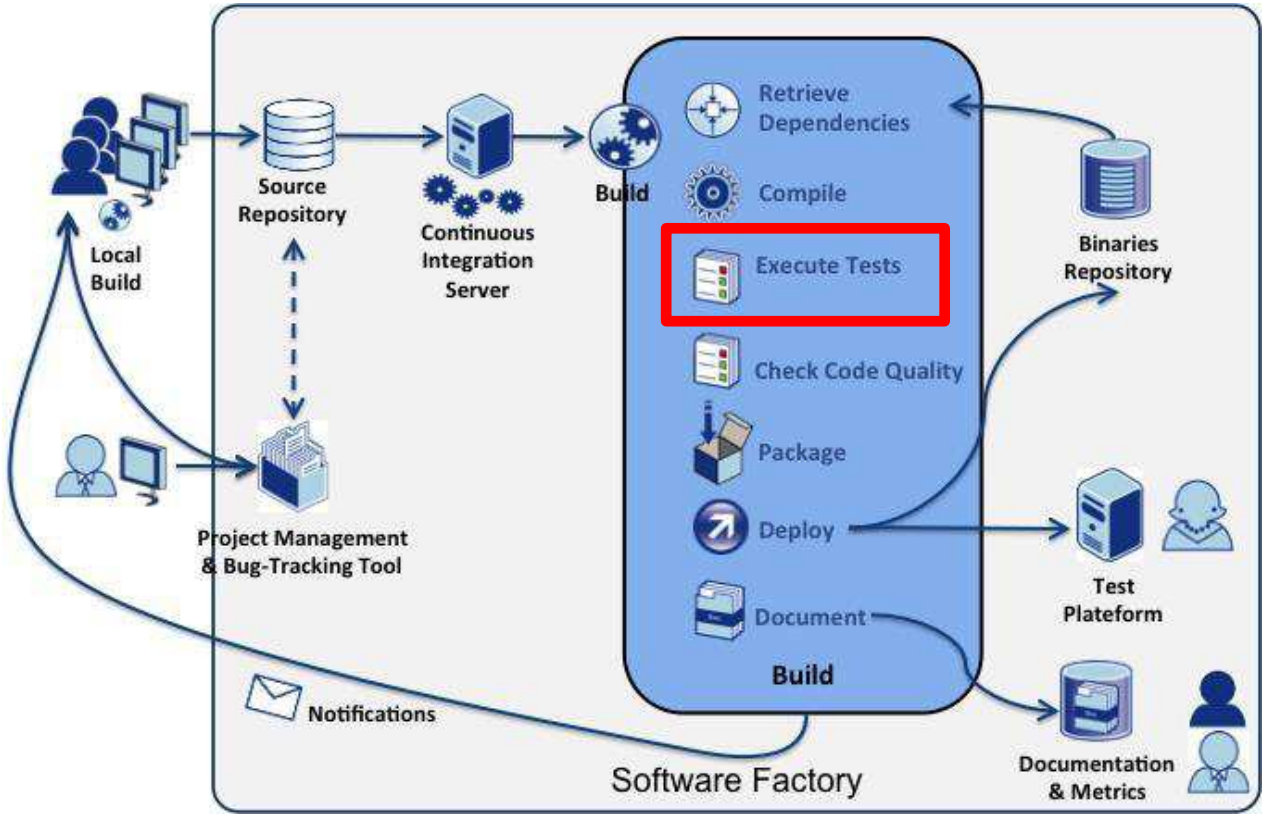
What does supported mean?

- Listed in the PAM
- Helpdesk will accept support requests

- Tested and verified

Factory





UNIFACE
Advanced Development Technology

RT and other test frameworks

- Uniface, Ranorex and other testing applications
- Automated Tests
  - > 400 4GL Test components,
  - > 100,000 individual test cases
- Grouped into functionality areas
  - XML
  - Pop Mail
  - SOAP
  - Oracle Unicode
  - GUI
  - ...

In-house test environment

- >100 Database servers
- 4 Test Build machines (Drive tests)
  - 105 slave machines (Execute tests)
- 20 Windows machines per code line for GUI testing
  - 4 (10 patch, 9.7 patch, 9.7 Service pack 10.3)
- LDAP, POP mail, SMTP, Web Servers, Proxy Servers, Call-In/Out Applications, …

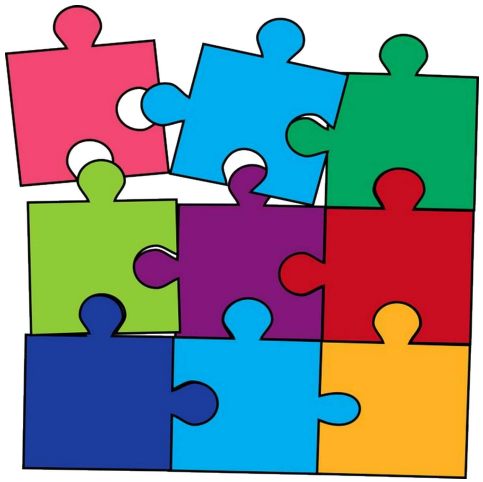
Limitations

- Services are shared
  - Locked when in use
  - Outage to a service affects all builds
- Static environment
  - Extending can be time consuming
  - Fragile to change
- Uniface assembled rather than installed

Cloud test environment

- Clean machines
  - Automatically created when needed
  - Deleted when the test run complete
- Services not shared
- Easy to extend
- Installed from e-dist + patch

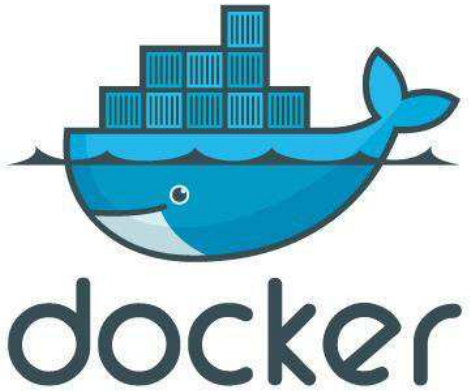
How?



- Base Infrastructure
  - Machine image created (Packer)
  - Database servers (Terraform/Ansible)
- Per platform
  - Infrastructure created/destroyed (Terraform)
  - Deploy and install Uniface (Ansible)
  - Deploy RT application (rsync)
- Per test
  - Create/destroy databases and clients (Ansible)
  - Configure and run test (Ansible)

Tests need services

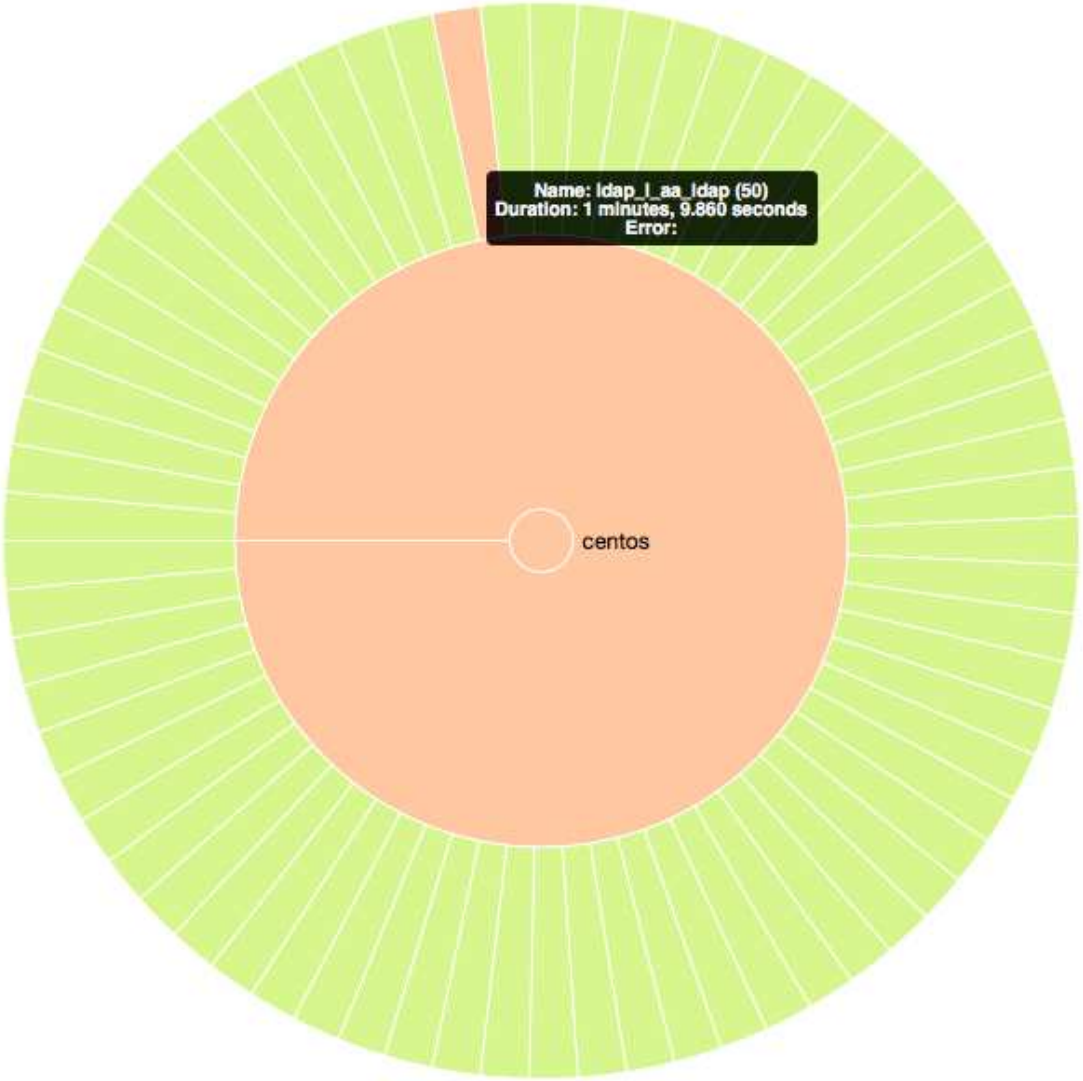
- Docker images made available for each service
  - Ldap
  - Pop/SMTP
  - Web server
  - Proxy server
  - …
- Started only for test that need them (Ansible)



Automating the run

- Tests to run defined using JSON
- Jenkins controls the automatic runs
  - Platforms run in parallel
  - Split tests over multiple instances of the same platform
- Results returned using rsync
- Report generated as web page

R



runtime:
script: **aa_ldap**
error_message:
total_runtime: **1 minutes, 9.860 seconds**
cloud_setenv: **envvars**
name: **ldap_l**
number: **50**
group: **ldap**
rt_pri_level:
status: **failure**
end_date: **2017-11-10T15:07:34+0000**
db_version:
artifacts_url:
Artifacts
aadib.log
Pipeline Steps
db:
start_date: **2017-11-10T15:06:24+0000**

Easy access for debugging

- Each step given consistent interface (make)
- Virtual machine makes deployment easy (Vagrant)
  - Factory
  - Support
  - Development
- Remote debugging available

Next

- SQL server
  - Port U5.1 driver to Linux
- Windows
  - Different
    - Language
    - Installer
    - Tests (GUI)

Final thoughts

- Scripted (DevOps)
  - Repeatable
  - Predictable
- Transient
  - Create when something is needed
  - Destroy it when it is not
- Scalable

Thank You!

Q&A

UNIFACE

Advanced Development Technology