DATA IS POTENTIAL

Demo: R2 SW Upgrade (single node, disruptive, mocked)

Provisioner

20/18/21

SEAGATE

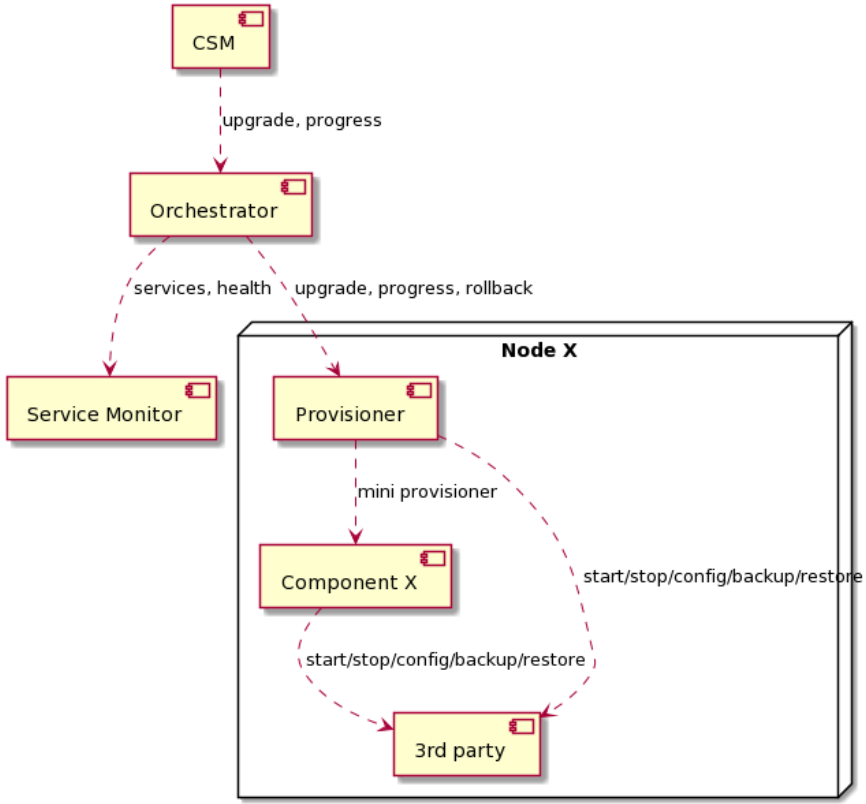
Goals

Goals

- **show we can validate and setup upgrade bundle**
  - Can validate it (some initial level)
  - Validation can be done as a separate command
  - Can mount it and make yum repositories inside available
- **show we can upgrade CORTX SW**
  - Can upgrade packages
  - Call provisioner mini APIs for each component
- **show we can rollback CORTX SW if upgrade fails**
  - Can downgrade packages
  - Call provisioner mini APIs for each component

High level view

Provisioner API on a node level



Initial state

Prepared environment state

1. Provisioner is installed
2. Deploy of CORTX component packages is mocked
    1. Mocked packages are installed
    2. Provisioner mini API for all components is available (part of packages installation)
3. Cluster is stopped (under maintenance) as it is assumed for that (node) level of Upgrade logic
    1. So provisioner shouldn't deal with cluster management and health checks

Steps:

1. Provisioner setup (e.g. using "provisioner setup_provisioner …" command)
2. salt "srvnode-1" state.apply components.misc_pkgs.mocks.cortx
3. salt "srvnode-1" state.apply components.misc_pkgs.mocks.cortx.build_upgrade pillar='{"inline": {"upgrade_repo_dir": "/tmp/cortx-upgrade"}}'
4. cp -r /tmp/cortx-upgrade /tmp/cortx-upgrade.invalid
5. rm -f /tmp/cortx-upgrade.invalid/RELEASE.INFO
6. mkisofs -graft-points -r -l -iso-level 2 -J -o /tmp/cortx-upgrade.invalid.iso /tmp/cortx-upgrade.invalid

Scenario

Show cases

Upgrade bundle validation

1. Validation of invalid CORTX SW upgrade bundle
2. Validation of valid CORTX SW upgrade bundle

Upgrade bundle setup

1. Mount valid CORTX SW upgrade bundle
2. Verify: upgrade packages are available

SW Upgrade API

1. Trigger an upgrade API
2. show upgrade related provisioner API are triggered for all components in an expected order
    1. Backup API
    2. (Package upgrade) [provsioner does that]
    3. Config API
3. Show packages have been upgraded

SW Upgrade failure with Rollback

1. Patch sw upgrade to fail
2. Tigger an upgrade API
3. show rollback related provisioner API were triggered for all components in an expected order
    - (Package downgrade) [provisioner does that as well]
    - Restore API
    - Config API
4. Show packages versions are the same

Steps

1. **Bundle validation**
    1. provisioner set_swupgrade_repo --source /tmp/cortx-upgrade.invalid.iso 2.1.0 --dry-run
    2. provisioner set_swupgrade_repo --source /tmp/cortx-upgrade.iso 2.1.0 --dry-run
2. **Bundle setup**
    1. provisioner set_swupgrade_repo --source /tmp/cortx-upgrade.iso 2.1.0
    2. Verification:
        - yum list installed | grep cortx
        - yum info cortx-csm_agent
3. **SW Upgrade**
    1. provisioner sw_upgrade --logfile --logfile-filename sw_upgade.1.log --logfile-level DEBUG
    2. tail -f /tmp/mock.log | grep Stage
    3. Verification
4. **SW Upgrade failure and Rollback**
    1. yum history rollback -y <ID>
    2. patch "/usr/local/lib/python3.6/site-packages/provisioner/commands/sw_upgrade.py" to emulate a failure
    3. provisioner sw_upgrade --logfile --logfile-filename sw_upgade.2.log --logfile-level DEBUG
    4. Verification

Thank you