Setup Exascale on KVM for clone and delete guest

Problem

We have an environment with 8 computes and 8 clusters : 4 Exascale + 4 ASM. We only deploy the clusters on 1 and 2 compute nodes and not others. After deployment, we tried to clone one of the guests in exascale clusters from compute node 1 to compute node 3 and we encountered an error: https://bug.oraclecorp.com/pls/bug/webbug_edit.edit_info_top?rptno=37932048

Reason

Since we are only deploying clusters on nodes 1 and 2, Exascale is not being setup for other nodes, which is why we are unable to clone the guest into those nodes

Proposal

Analyze the deployment step “Configure Exascale Storage on Cell Servers” to understand which functions are required to call to setup EDV and then call the same functions during clone guest

Exascale EGS document

https://docs.oracle.com/en/engineered-systems/exadata-database-machine/exscl/exascale-services.html

Exascale cluster services, also known as Exascale global services (EGS), provide the core foundation for the Exascale system. EGS primarily manages the storage allocated to Exascale storage pools. It also manages storage cluster membership, provides security and identity services for storage servers and Exascale clients, and monitors the other Exascale services. Exascale cluster services use the Raft consensus algorithm.

For high availability, every Exascale cluster contains five EGS instances.

For Exascale clusters with five or more Exadata storage servers, one EGS instance runs on each of the first five storage servers.

For Exascale configurations with fewer than five storage servers, one EGS instance runs on each Exadata storage server, and the remaining EGS instances run on the Exadata compute nodes to make up the required total of five. In a bare-metal configuration, EGS compute node instances run on the server operating system. In a configuration with compute nodes running on virtual machines (VMs), EGS compute node instances run in the hypervisor.

Flow for Configure Exascale Storage on Cell Servers

Existing flow during full deployment	Proposed flow for stand-alone Exascale setup in Non EGS node
• configureExaScaleOnCells • configureExaScaleOnCluster(engineeredSystem, cluster) • List\<Machine> exascaleMachines = ExascaleXmlUtils.getExascaleCellsFromCluster(engineeredSystem, cluster) • For each of these cells: ○ validate celldisks exist in cells- throw errors if unable to get celldisk status ○ startupPrimaryCellServices(engineeredSystem,cluster) ■ Starts up rs, ms, cellsrv on all cells in cluster ○ isCellServersReadyForExascaleDeployment(engineeredSystem, cluster) - checks if all services started up correctly ○ performValidations - validate cluster name, storage pool names, image version, ers ip address, exascale cluster configurrations, private ips ○ stop useless services - iptables, cell walls ○ validateSpaceOnExascaleCells - validate enough free hc, ef space exists cells ○ createPoolDisks(engineeredSystem, cluster, exaScaleCells) - create hc and ef pool disks on storage pools ○ Clean up config files on cells ○ createEgsCluster(engineeredSystem,cluster) - create egs cluster on the candidate cells for this cluster ○ setRootCertsOnCluster(exascaleMachines) - setup trust store on egs cells ○ setEgsIpAddressesOnCluster(engineeredSystem, cluster, exascaleMachines) - set private ips to egs cells ○ generateAndSetKeyPairOnCluster(engineeredSystem,cluster) - generate private key on egs leader node and ad this file to the wallet of every cel ○ setupWalletsOnNonEgscells(engineeredSystem, cluster) - on cells with no egs running, if we need services like cellsrv or EBS, we need to setup wallets	

- registerStorageNodes(engineeredSystem, cluster) - get leader node, collect public key from each of the nodes and register that public key on the leader node
- restartRsAndCellSrvOnCluster(engineeredSystem,cluster)
- setupRestSerivcesOnCluster(engineeredSystem,cluster) - setup rest certificates on each server
- setEgsDeploymentMode(engineeredSystem, cluster)
- Startup Eds, UsrEds, Ifd services on cluster
- createStoragePool(engineeredSystem, cluster) - only creates if it does not exist - create storage pool on egs leader
- createExternalExascaleUsers
  - get exascale users from exascluster
  - get the exascale leader
  - create/copy all public key file in leader cell
  - create exascale user and set public keys to the users
- createCellDiagUserOnCluster - create cell user on all exascale cells
- setupWalletsOnOtherCells - sets up wallets in non leader cells, in case we want to connect to them from other non leader cells
- configureErsIPonCells - get cells which are neither front end server nor backend server, give ms privilege to them and restart ms
- enableAep
- setRestNetworkId - set rest network id on leader cell
- fillTruststoreOnStorageNodes
- validateFlachCache - log which cells have flashcache and which ones don't
- startupBlockStoreServicesOnCells - startup bsw, bsm on all cells irrespective of edv configuration
- setExascaleSystemProfileCellServers - set system profile on cell servers
- setErsIpOnWallets - set ers ip on both egs and non egs cells. On every cellnode, chwallet to have rest end point as ers end point
- configureEgsOnKVMExascaleCluster
  - Perform all egs related stuff on computes
  - In all exascale clusters, do the following
  - Configure egs on kvm hosts. Get egs kvm hosts, in each host
    - validate network for exascale
    - config cellinitora on hosts using dbmcli
    - restart RS on these hosts
    - Startup MS on these hosts
    - create EGS cluster on these KVM hosts
    - Create node users on KVM hosts
    - setRootCertsOnEgsCellsAndKvmHosts
    - setEgsIpOnAllComputesAndCells - set private IPs on all egs servers
    - configureErsIPonComputes - for all nodes with bsw, set ers ip, mkuser, give MS privilege to user, generate public key and copy it to the cell which is running escli admin (**EDV cannot start without running this step**)
    - **Startup ESNP and EDV services on KVM hosts**
    - Start blockstore services on KVM hosts
    - Create virtual interfaces
    - Start Ifd service on KVM hosts
    - updateExaRootUrlOnEgsCells on computes with EGS
  - configureExascaleOnNonEgsKvmHosts
    - Entry point to configure Exascale on non EGS KVM hosts i. e. KVM hosts that do not qualify to run EGS process
    - Configure cellinitora
    - Restart RS, Start MS
    - Create node user, setup wallets
      - createWalletsonNonEgsKvmHosts
        - setupWalletOnNonEgsKvmHost
          - mkwallet
          - assign wallet to user
          - if not cloud - point to ERS ip
          - Point to exaRootUrl
          - Clear old trusted certs
          - get trust cert file name
            - If cloud, only use existing file in local machine
            - If not cloud and file not present in local machine, get the file from one of the existing Exascale nodes
        - runWalletSetupScriptOnCompute
    - Startup BSW on hosts if app vms do not exist
    - create bswvoluehavips if they exist in es xml
    - **Startup EDV on KVM hosts**
- deletePublicAndPrivateKeysFromCells
- validateExascaleConfiguration - get admin cell and run lsservice on escli to check if all services are online

- Call this method during CREATE_GUEST step of clone guest flow
- configureExascaleOnNonEgsKvmHost
  - Check if exascale is already configured on node
    - Yes: Return
    - No: Acquire lock on the node and do the following
  - Configure cellinitora
  - Restart RS, Start MS
  - Create node user, setup wallets
    - createWalletsonNonEgsKvmHosts
      - setupWalletOnNonEgsKvmHost
        - mkwallet
        - assign wallet to user
        - if not cloud - point to ERS ip
        - Point to exaRootUrl
        - Clear old trusted certs
        - get trust cert file name
          - If cloud, only use existing file in local machine
          - ~~If not cloud and file not present in local machine, get the file from one of the existing Exascale nodes~~
          - NEW : If not cloud, ALWAYS get from existing Exascale nodes. Do not use local trust cert file
      - runWalletSetupScriptOnCompute
  - Startup BSW on hosts if app vms do not exist
  - create bswvoluehavips if they exist in es xml
  - **Startup EDV on KVM hosts**

Flow for clone guest

- Create guest
- Create Users
- Deploy cell connectivity
- deploy config software
- run root script
- Add db home
- add db instance

Questions:

Based on the above flow, it seems like **configureEgsOnKVMExascaleCluster** is where EDV is being started up. So,

1. Should we call that function in our clone guest flow?
    a. Ans: No - the function to call is configureExascaleOnNonEgsKvmHost
2. Is it the only function we need to call?
    a. If a target KVM during clone guest does not have Exascale in it, yes
3. Which step of the clone guest flow should we call it?
    a. Ans: Create_Guest

Versions:

Version 1:

- For clone guest flow: Added the call to ExascaleComputeDeployUtils.configureExascaleOnNonEgsKvmHostWithLock here:
    - EdvConfigUtils.doCreateAttachEdvVolumesWithGuestVolObjects
    - Comments: Min said it is not a good idea to add it here because this method may be called from other places, not only clone guest so it could affect other flows
- For deployment with install.sh: Added the call ExascaleComputeDeployUtils.configureExascaleOnNonEgsKvmHostWithLock here:
    - EdvConfigUtils.createAttachEdvVolumesForClusterMachines
    - Comments: Min said it is not a good idea to add it here because this method may be called from other places, not only deployment so it could affect other flows

Version 2:

- For clone guest flow: Added the call to ExascaleComputeDeployUtils.configureExascaleOnNonEgsKvmHostWithLock here:
    - GuestCli.doDeployCloneGuest
    - Comments
        - Min thinks this is the correct place to add this as it is specifically for the clone guest flow
        - Krish thinks we should add it in VmUtils.createVms instead
- For deployment with install.sh:
    - Min thinks we should add another flag in if condition in ExascaleUtils.configureExaScaleOnCluster here so that we don't skip if just the cells are setup for exascale and setup exascale on non egs machines(remaining computes)
    -
      ```
      if (
          isCellServersReadyForExascaleDeployment(engineeredSystem, cluster) &&
          ExascaleComputeDeployUtils.getListOfNonEgsMachinesForConfig(engineeredSystem, cluster).size()
      < 1
      )
      ```
    - Comments
        - Saeed thinks this is not the right place to add as if it is an && condition, the following code will try to setup exascale on the cells as well and fail because they were already setup before