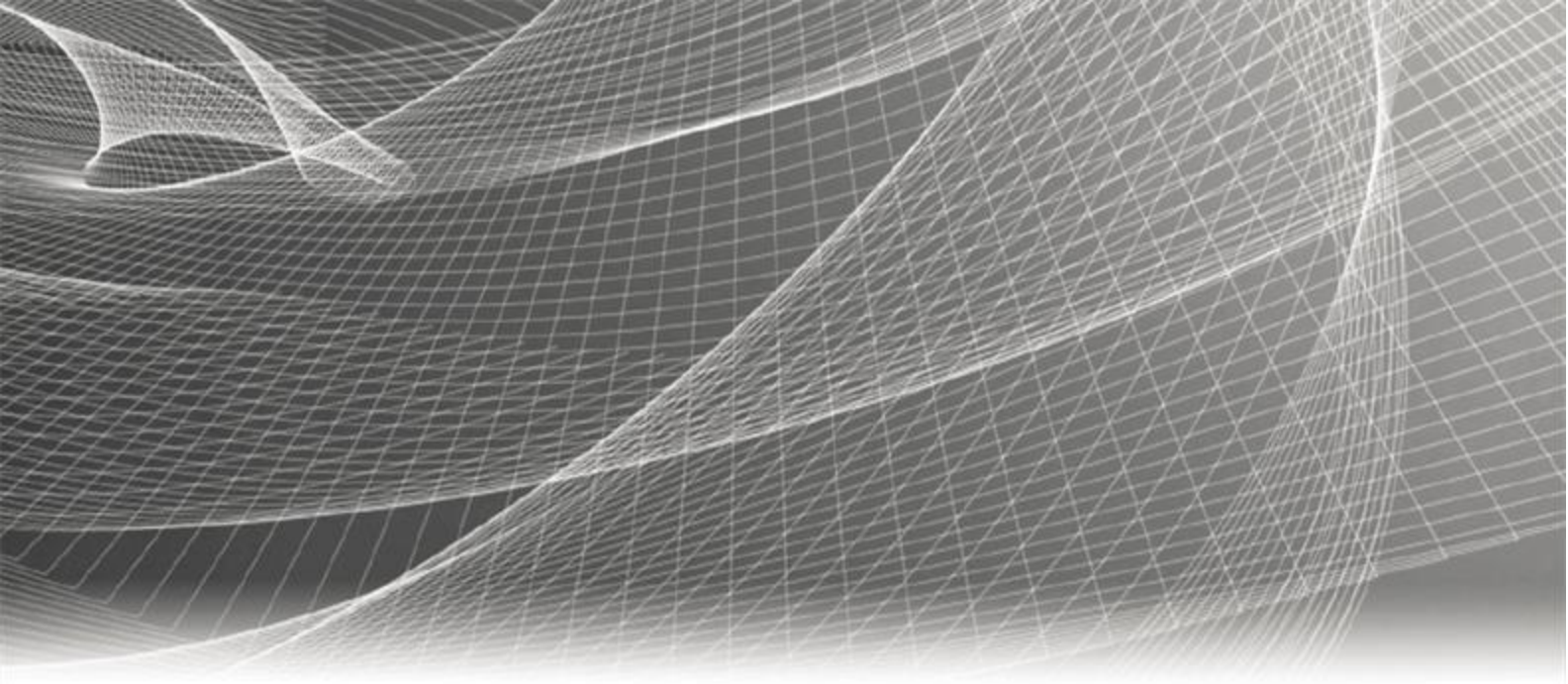


RSA IDENTITY GOVERNANCE AND LIFECYCLE

AWS Installation and Clustering Guide

7.5

Contact Information
RSA Link at https://community.rsa.com contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Contents

Preface 4

About This Guide 4

Documentation Set 4

Support and Service 4

Chapter 1: Overview 5

Chapter 2: Deploy a Standalone Instance 7

Provision an Amazon RDS Instance 7

Provision an EC2 Instance 8

Configure AWS Environment for a Standalone RSA Identity Governance and Lifecycle Installation 9

Configure AWS Environment for a Standalone RSA Identity Governance and Lifecycle Installation with RSA-provided database 11

Upgrade Standalone RSA Identity Governance and Lifecycle 13

Chapter 3: Clusteringon AWS 14

Performance Results on AWS (v7.5) 17

An AWS deployment of RSA Identity Governance and Lifecycle comprises the following components: 17

WildFly Collections in the AWS Environment 18

Average Collection Performance comparison 20

Single User Volume Tests 21

Business Description Imports 21

Reports 21

Reviews 22

Rules and Roles 24

500 concurrent users test on AWS 26

Access Requests on AWS with 500 concurrent users 26

User Reviews on AWS with 500 concurrent users 28

ACME Specification 30

Identity Information 30

Account Information 30

Application Information 30

Preface

About This Guide

This guide provides the requirements and procedures required to deploy RSA Identity Governance and Lifecycle using Amazon Web Services (AWS).

Documentation Set

The latest product documentation is always available at https://community.rsa.com/community/products/governance-and-lifecycle.

Document	Description
Release Notes	What's new in the release, fixed issues, known issues and workarounds.
Installation Guide	Product installation instructions.
Upgrade and Migration Guide	Instructions for upgrading your product version and data.
Database Setup and Management Guide	Instructions for setting up and managing a customer-supplied Oracle database for RSA Identity Governance and Lifecycle.
Configuring WildFly Clusters	Instructions to set up and configure a WildFly application server cluster in an RSA Identity Governance and Lifecycle deployment.
Online Help	All concepts and instructions you need to configure and use the product.
Administrator's Guide	How to configure and manage RSA Identity Governance and Lifecycle. Contains a subset of the information provided in the Online Help.
Public Database Schema Reference	The public view of the database schema.

Support and Service

You can access community and support information on RSA Link at https://community.rsa.com/community/products/governance-and-lifecycle. RSA Link contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

You can also access the RSA Identity Governance and Lifecycle community at https://community.rsa.com/community/products/governance-and-lifecycle/client-partner-community. This private community is only available to RSA Identity Governance and Lifecycle customers, partners, and internal RSA staff.

Chapter 1: Overview

This guide provides instructions for deploying a standalone RSA Identity Governance and Lifecycle deployment in an Amazon Web Services (AWS) environment and configuring clustering between multiple nodes in AWS.

This guide is intended for RSA Identity Governance and Lifecycle installers and administrators. To follow these procedures, users must have the appropriate network and administrative permissions to install and configure the RSA Identity Governance and Lifecycle application and familiarity with AWS.

An AWS deployment of RSA Identity Governance and Lifecycle requires the following components:

- A remote Oracle database deployed using an Amazon Relational Database Service (RDS) instance or an RSA-provided Oracle database deployed in an Amazon EC2 instance or an RSA-provided Oracle database deployed in an Amazon EC2 instance. The remote database must adhere to the same requirements specified in the *RSA Identity Governance and Lifecycle Database Setup and Maintenance Guide*.
- One or more RSA Identity Governance and Lifecycle nodes deployed on Amazon EC2 instances.
- A Virtual Private Cloud (VPC) to allow communication between the nodes and the database.

Deployment Overview

The high-level steps to deploy RSA Identity Governance and Lifecycle in an AWS environment are:

1. Configure an Amazon Virtual Private Cloud (VPC). For more information, see documentation provided by Amazon.
2. Deploy a database using one of the two following options:
   - Provision an Amazon RDS Instance for the RSA Identity Governance and Lifecycle database. For more information, see Deploy a Standalone Instance .
   - Deploy an RSA-provided Oracle database in an EC2 instance using the same procedures as for a software bundle deployment.
2. Provision an EC2 Instance. For more information, see Deploy a Standalone Instance.
3. Configure the AWS environment for a standalone/single-node RSA Identity Governance and Lifecycle installation. For more information, see Deploy a Standalone Instance.
4. In a deployment with multiple RSA Identity Governance and Lifecycle nodes, the install script on each node prompts for and gathers cluster related info and automatically configures wildfly for clustering. For more information, see Clustering on AWS.

Requirements

The following tables list the requirements for deploying RSA Identity Governance and Lifecycle in an AWS environment.

Database Requirements

Database engine version	Oracle 19c (19.0.0.0.ru-2021-01.rur-2021-01.r1 or Latest) – Certified Oracle 12c (Oracle 12.1.0.2v22) - Supported Oracle 12c (12.2.0.1.ru-2021-01.rur-2021-01.r1 or Latest) - Supported
Database instance class	db.m5.2xlarge or better

Allocated storage	Minimum: 300 GB Recommended: 1 TB or greater depending on data load requirements

EC2 Instance Requirements

Amazon Machine Image (AMI)	SUSE Linux Enterprise Server 12 SP4+, Red Hat Enterprise Linux 7.6+ and Red Hat Enterprise Linux 8
Instance Type	t2.xlarge or better
Storage	20 GB or more. **Note:** 20 GB is the minimum space required by RSA Identity Governance and Lifecycle and your deployment may require additional storage depending on your environment.

Related Documentation

For additional information, refer to the following documentation:

- *RSA Identity Governance and Lifecycle Installation Guide*
- *RSA Identity Governance and Lifecycle Database Setup and Maintenance Guide*

Chapter 2: Deploy a Standalone Instance

Provision an Amazon RDS Instance

Note: The following procedure is applicable only if you plan to deploy the database in an Amazon Relational Database Service (RDS) instance. You may alternately deploy the RSA-provided Oracle database in an Amazon EC2 environment using the same procedures as for a software bundle deployment. For more information, see the *RSA Identity Governance and Lifecycle Installation Guide*. If you deploy the RSA-provided database in an EC2 instance, your organization takes on the additional tasks of maintaining and managing the database.

This section describes how to provision an Amazon Relational Database Service (RDS) instance for the RSA Identity Governance and Lifecycle database. The following procedure indicates fields that have specific requirements to configure an RSA Identity Governance and Lifecycle database. Fields in the Amazon RDS console that are not listed here should be configured as appropriate for your environment.

For more information about configuring a database for RSA Identity Governance and Lifecycle, see the *RSA Identity Governance and Lifecycle Database Setup and Management Guide*.

Before you begin

- You must have a license for an Oracle database that meets the requirements stated in Overview on page 6.
- You must have a Virtual Private Cloud (VPC) configured, where you will deploy the RSA Identity Governance and Lifecycle nodes.
- The parameter group for the database must include the following Oracle init-parameter: _fix_control='17376322:OFF'.

Note: RSA Identity Governance and Lifecycle as deployed in an AWS environment is not compatible with Oracle Statspack.

Follow the steps below to create an RDS instance

1. In the AWS console for RDS, click **Create Database**.
2. Select **Oracle** and the edition for your Oracle license.
3. Under Choose use case, select the appropriate use case for your deployment.
4. In the Instance specifications section, enter the following details:

Field	Details
License model	bring-your-own-license
DB engine version	Oracle 19c (19.0.0.0.ru-2021-01.rur-2021-01.r1 or Latest) – Certified Oracle 12c (Oracle 12.1.0.2v22) - Supported Oracle 12c (12.2.0.1.ru-2021-01.rur-2021-01.r1 or Latest) - Supported
DB instance class	db.m5.2xlarge or better
Allocated storage	Minimum: 300 GB. Recommended: 1 TB or greater depending on data load requirements.

5. In the Settings section, enter the following details:

Field	Details
DB instance identifier	Enter a unique identifier.

Master username	<Any username>.
Master password	<Any 8-digit password supported by RDS>.
Confirm password	Same as above.

6. Click **Next** to proceed to the **Configure Advanced Settings** page.
7. In the **Network & Security** section, provide the details for your Virtual Private Cloud (VPC). The RDS must be configured to connect to the VPC where the RSA Identity Governance and Lifecycle nodes are deployed.
8. In the **Database options** section, enter the following details:

Field	Details
Database name	AVDB
Port	1555
DB parameter group	Indicate a parameter group that meets the database requirements specified in the *RSA Identity Governance and Lifecycle Database Setup and Management Guide.*
	Note: Ensure that the parameter group includes the following Oracle init-parameter: _ fix_control='17376322:OFF'.
Option group	Specify an option group that meets the database requirements specified in the *RSA Identity Governance and Lifecycle Database Setup and Management Guide.* Ensure that the option group specifies the correct timezone for the instance.
Character set name	AL32UTF8

9. Configure the Encryption, Backup, Monitoring, Log exports, and Maintenance sections as appropriate for your environment.
10. Click **Create Database**.
    A notification appears when database creation is complete, and the new instance is displayed in the list of DB instances.

After you finish

- Record the location of the instance, which is needed to deploy the EC2 instance where you configure RSA Identity Governance and Lifecycle. Click the name of the instance on the RDS console to open the instance details, and the location is provided in the **Endpoint** field.
- In SQLDeveloper, as AVUSER, use the following command to set the Oracle instance timezone:

  ```
  exec rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz =>
  'timezone');
  ```

  Where *timezone* is the same timezone that was selected in the **Option group**.

Provision an EC2 Instance

Perform this procedure to provision an Amazon EC2 instance, on which you deploy the RSA Identity Governance and Lifecycle software.

Before you begin

- You must have a Virtual Private Cloud (VPC) configured, where you will deploy the RSA Identity Governance and Lifecycle nodes.
- Provision an RDS instance that is configured to communicate with your VPC.

Follow below steps to install EC2 instance

1. In the AWS Console, open the EC2 Dashboard and click **Launch Instance**.
2. In the Choose an Amazon Machine Image (AMI) section, select the appropriate image for your environment, such as **SUSE Linux Enterprise Server 12 SP4+** or **Red Hat Enterprise Linux 7.6+** or **Red Hat Enterprise Linux 8**
3. In the **Choose an Instance Type** section, select an instance type that is **t2.xlarge** or better.
4. Click Next: **Configure Instance Details**.
5. In the **Configure Instance Details** section, provide the appropriate details for your environment.
6. Click **Next: Add Storage**.
7. Click **Add New Volume** and add a storage volume of type **root** that provides at least 20 GB of storage. **Note:** 20 GB is the minimum space required by RSA Identity Governance and Lifecycle, and your deployment may require additional storage depending on your environment.
8. Click Next: **Add Tags.**
9. Click Add Tag and specify the appropriate key pair for your environment.
10. Click Next: **Configure Security Group.**
11. Click Add Rule and add the appropriate security group details for your environment.
12. Click **Review and Launch**.
13. On the Review Instance Launch screen, review the EC2 instance details. If correct, click **Launch**.
14. In the pop-up, specify the SSH key pair to use for aveksa.pem. You can specify an existing key pair or create a new one.
15. A notification appears when the instance creation is complete, and the new instance is listed in the EC2 Dashboard of the AWS Console.

Configure AWS Environment for a Standalone RSA Identity Governance and Lifecycle Installation

Perform the following steps to configure the AWS environment for either a standalone RSA Identity Governance and Lifecycle installation or an individual instance of a cluster.

Before you begin

- Ensure that the Amazon RDS has been provisioned and the remote Oracle database has been set up. For more information, see Provision an Amazon RDS Instance and the *RSA Identity Governance and Lifecycle Database Setup and Management Guide*.
- Provision an Amazon EC2 instance. For instructions, see Provision an EC2 Instance.
- Ensure that you meet the prerequisites for installing RSA Identity Governance and Lifecycle. For more information, see the *RSA Identity Governance and Lifecycle Installation Guide*.

Follow below steps for RSA Identity Governance and Lifecycle Installation

1. Use the following command to SSH into the EC2 instance as ec2-user with the **aveksa.pem** key and run the below command to switch to root

   ```
   sudo -i
   ```

2. Set SSH as root, and use the following command to set the appropriate timezone for your deployment:

   ```
   unset DISPLAY; yast2 timezone
   ```

3. Install the following prerequisite packages:

   For SUSE Linux Enterprise Server 12 SP4+ :

   a. Transfer below required files for RSA Identity Governance and Lifecycle installation to the /tmp/aveksa/packages folder

   Required 3rd-party libraries

   - bea-stax-api-1.2.0-5.25.2.noarch.rpm
   - lcms2-2.7-7.2.x86_64.rpm
   - rhino-1.7-6.25.noarch.rpm
   - xmlbeans-2.1.0-2.27.noarch.rpm

   The following RSA Identity Governance and Lifecycle installer components:

   - adoptjdk_8u252b09.tar.gz
   - wildfly-21.0.1.Final.tar.gz

   b. Run below command to the install the RPM files:

   ```
   zypper in /tmp/aveksa/packages/*.rpm
   ```

   c. Run below command for SLES 12 SP4

   ```
   zypper in syslinux
   ```

   d. **IMPORTANT**: Install the latest OS patches and packages by running the following command:

   ```
   zypper up
   ```

   Amazon provides important patches for OpenJDK, Network Security Services, Linux Kernel, OpenSSL and other components that must be installed on the system.

   For Red Hat Enterprise Linux environment:

   a. Transfer below required files for RSA Identity Governance and Lifecycle installation to the /tmp/aveksa/packages folder

   - adoptjdk_8u252b09.tar.gz
   - wildfly-21.0.1.Final.tar.gz

   b. Run below commands

   - `yum install perl`
   - `yum install nfs-utils`
   - `yum install syslinux`
   - `yum install bind-utils`

- `yum install GConf2`
- `yum install zip`
- `yum install unzip`
- `yum update`

4. Reboot the machine to enable changes and load the new kernel.
5. Transfer the desired RSA Identity Governance and Lifecycle build into the aveksa-<product-version>.tar.bz2/tmp/aveksa/staging folder.
   cd/tmp/aveksa/staging
   tar xvf
   aveksa-<product-version>.tar.bz2
   If AFX is required, copy AFX Standard and Premium connector templates.
6. Load the Aveksa_System.cfg configuration file located in /tmp/aveksa/staging/deploy, and ensure that the AMAZON_RDS setting is present and has a value of Y. If the parameter is not present, add it to the beginning of the file with the value of Y.
7. Edit Aveksa_System.cfg to ensure that the following parameters refer to the Amazon RDS instance:
   - REMOTE_ORACLE_IP=
   - REMOTE_ORACLE_PORT=
   - ORACLE_SID=
   - AVEKSA_PASS=<Password that added while creating RDS>
8. If the machine is part of a VPC that is not configured to use the DNS hostnames option, edit the /etc/hosts file to create a new entry for the IP address of the machine. For example, if the IP address is 172.24.216.5, create one entry: "172.24.216.5 ip-172-24-216-5.ec2.internal ip-172-24-216-5".
9. Use the following command to start the installer, and follow the prompts:

   - In a standalone instance or the first node in a cluster:

     ```
     cd /tmp/aveksa/staging/deploy;
     ./install.sh -createschema
     ```

   - In a clustered environment, for all nodes other than the first:

     ```
     cd /tmp/aveksa/staging/deploy;
     ./install.sh
     ```

Note: If the installer indicates that there is a missing record in the /etc/hosts file, add the missing record. If the installer indicates that the noelision libraries are missing, add them as the first line in the file /etc/ld.so.conf.

Configure AWS Environment for a Standalone RSA Identity Governance and Lifecycle Installation with RSA-provided database

RSA-provided database options:
Database installed locally on the IGL app server EC2 instance (Software Bundle with local RSA-provided database)
Database installed on a remote EC2 instance (RSA provided database tar package would be used).

Procedure to setup Database in a EC2 instance

1. Log into the database installation machine as root.
   Note: You must log in as the root account to ensure that the proper permissions are granted. This procedure may not work using any other account.
2. In a SuSE Enterprise Linux environment, install all required OS packages using the following command:

   ```
   zypper install binutils gcc gcc48 glibc glibc mksh
   libaio1 libaio-devel libcap1 libstdc++48-devel
   ```

```
libstdc++6 libstdc++-devel libgcc_s1 make sysstat xorg-
x11-driver-video xorg-x11-server xorg-x11-essentials
xorg-x11-Xvnc xorg-x11-fonts-core xorg-x11 xorg-x11-
server-extra xorg-x11-libs xorg-x11-fonts libcap2
oracleasm-kmp-default javapackages-tools lcms2 bea-stax-
api rhino xmlbeans syslinux zip unzip bzip2
```

In a Red Hat Enterprise Linux environment, run the following prerequisite packages:

```
yum install binutils gcc gcc48 glibc glibc mksh libaio-devel
libstdc++-devel make sysstat javapackages-tools lcms2 bea-stax-api
rhino syslinux zip unzip bzip2 compat-libcap1-1.10-7.el7.x86_64
compat-libstdc++-33-3.2.3-72.el7.x86_64 gcc-c++-4.8.5-4.el7.x86_64
ksh iptables-services
```

3. Install the latest SuSE OS patches and packages by running following command:

   ```
   zypper up
   ```

   In a Red Hat Enterprise Linux environment, run the following command:

   ```
   yum update
   ```

4. Download the following two OS packages from the Oracle's website:
   https://www.oracle.com/linux/downloads/linux-asmlib-sles12-downloads.html

   a. oracleasm-support-2.1.8-1.SLE12.x86_64.rpm
   b. oracleasmlib-2.0.12-1.sle12.x86_64.rpmb.

   Copy the two package files to any path on the virtual machine, then manually install them using the following commands as root:

   ```
   rpm -i oracleasmlib-2.0.12-1.sle12.x86_64.rpm
   rpm -i oracleasm-support-2.1.8-1.SLE12.x86_64.rpm
   ```

5. If the machine is part of a VPC that is not configured to use the DNS hostnames option, edit the /etc/hosts file to create a new entry for the IP address of the machine. For example, if the IP address is 172.24.216.5, create one entry: "172.24.216.5 ip-172-24-216-5.ec2.internal ip-172-24-216-5".
6. Extract the file RSA_IGL_DatabaseOnly.tar.bz2 to the /tmp directory using the following command:
   `tar xjf RSA_IGL_DatabaseOnly.tar.bz2`

7. Execute the script installDatabaseOnly.sh using the following command:
   `./installDatabaseOnly.sh`

8. When the script completes, a standalone Oracle database server is configured with the appropriate tablespace, users, and schema for RSA Identity Governance and Lifecycle. Logs are accessible in the /tmp/aveksa-install.log directory.

Install Identity Governance and Lifecycle in a EC2 instance

1. Follow the steps upto step 5 in page 11 for RSA Identity Governance and Lifecycle installation.
2. If the machine is part of a VPC that is not configured to use the DNS hostnames option, edit the /etc/hosts file to create a new entry for the IP address of the machine. For example, if the IP address is 172.24.216.5, create one entry: "172.24.216.5 ip-172-24-216-5.ec2.internal ip-172-24-216-5".
3. Use the following command to start the installer, and follow the prompts:
   `cd /tmp/aveksa/staging; ./install.sh`

Upgrade Standalone RSA Identity Governance and Lifecycle

To upgrade to RSA Identity Governance and Lifecycle 7.5.0, follow the steps below:

1. Delete the files located in /tmp/aveksa/staging and /tmp/aveksa/packages.
2. Transfer below required files for RSA Identity Governance and Lifecycle installation to the /tmp/aveksa/packages folder.
    - adoptjdk_8u252b09.tar.gz
    - wildfly-21.0.1.Final.tar.gz
3. Transfer the RSA Identity Governance and Lifecycle 7.5.0 build into the aveksa-<product-version>.tar.bz2/tmp/aveksa/staging folder.
   cd/tmp/aveksa/staging
   tar xvf aveksa-<product-version>.tar.bz2
   Copy the AFX standard and premium connectors templates, if required.
4. Edit the Aveksa_System.cfg configuration file located in /tmp/aveksa/staging/deploy, and ensure that the AMAZON_RDS setting is present and has a value of Y. If the parameter is not present, add it to the beginning of the file with the value of Y.
5. Edit Aveksa_System.cfg to ensure that the following parameters refer to the Amazon RDS instance:
    - REMOTE_ORACLE_IP=
    - REMOTE_ORACLE_PORT=
    - ORACLE_SID=
    - AVEKSA_PASS=<Password that added while creating RDS>
6. Use the following command to start the upgrade and follow the prompts:

```
cd /tmp/aveksa/staging/deploy;
./install.sh
```

RDS Upgrade

For upgrading RDS to latest (Oracle 19c) please refer https://aws.amazon.com/blogs/database/best-practices-for-upgrading-amazon-rds-for-oracle-db-instances-from-11-2-0-4-to-19c/

If you are using an RSA-provided database on an EC2 instance, we urge you to upgrade to 19C.

Chapter 3: Clustering on AWS

You can configure a WildFly application server cluster in an RSA Identity Governance and Lifecycle on Amazon Web Services (AWS). This deployment requires an Oracle Relational Database Service (RDS) instance as the remote database.

For setting up a Widlfy cluster and deploying IG&L, the installation process needs to be done on the domain node and on each of the host nodes. The installation process asks a series of questions during set up. Note the following:

- The values specified for ***server name*** and ***host/domain controller name*** must be unique across all nodes in the cluster.
- The ***management user name, management user password*** and ***server group name*** must be same across all the nodes in the cluster.

Prerequisites for Clustering in an AWS Environment

Before setting up WildFly clustering in an AWS environment, perform the following tasks.

- Set up both EC2 and RDS instances as mentioned in Deploy a Standalone Instance.
- Configure the EC2 instances as mentioned in Configure AWS Environment until step 7.
- Add all the cluster nodes under /etc/hosts file. Example: for a two-node cluster create the below entries:
  "172.24.216.5 ip-172-24-216-5.ec2.internal ip-172-24-216-5"
  "172.24.216.6 ip-172-24-216-6.ec2.internal ip-172-24-216-6"
- Ensure that the required VPC is correctly configured to allow connectivity between nodes. To confirm this, try to SSH between RSA Identity Governance and Lifecycle nodes, or telnet to the RDS node on port 1555.

Configure the Firewall

Firewalls are disabled on the SLES image provided by Amazon. As an alternate to a firewall, access to the node can be managed using a combination of an AWS Security Group and a Virtual Private Cloud (VPC) configuration. AWS provides the ability to create a VPC, and access to all nodes within the VPC is controlled by rules specified in the Security Group configuration.

In a clustered environment, the required connectivity can be achieved using either a firewall or a security group configuration.

The list of TCP ports that must be opened specifically for cluster communication are 9999, 8080, 7600, 57600. Other ports required for connecting to the node are 21, 22, 1158, 1555, 8443, 8444.

Using Security Groups

Security groups can be configured to have specific ports opened between the source and target IP address range. Inbound and outbound ports can be configured in the rule to allow the above list of ports.

Using firewall

In the case of an individual firewall:

```
vi /etc/sysconfig/SuSEfirewall2
```

You do not need to open UDP in an AWS network. Include the below ports for TCP under the variable FW_SERVICES_EXT_TCP:

```
FW_SERVICES_EXT_TCP="21 22 1158 1555 8080 8443 8444 9999 7600 57600"
/sbin/SuSEfirewall2 on
```

Setting up domain node

Follow the below steps for setting up the domain node.

1. Run `./install.sh –createschema` under /tmp/aveksa/staging/deploy directory.
2. Respond to the following installer prompts:

a. Have you reviewed the release notes and agree to the license terms (yes or no)? Enter "yes".
b. What is the location for installation [/home/oracle]?
c. Where are the package files located [/tmp/aveksa/packages]?
d. Do you want to install the application as a service (yes or no)? Enter "yes".
e. Do you want to setup a Wildfly cluster (yes or no)? Enter "yes".
f. Are you configuring the Domain Controller (yes -> Domain Controller / no -> Host Controller)? (yes or no)? Enter "yes".
g. What is the Wildfly Management User Name [AveksaClusterAdmin]?
h. Enter the Wildfly Management User Password:
i. Confirm the Wildfly Management User Password:
j. What is the Wildfly Domain Controller Name (Should be unique in the cluster) [master]?
k. What is the Wildfly Server Name (Should be unique in the cluster) [img-server-1]?
l. What is the Wildfly Server Group Name (Should be same across the cluster) [img-servergroup]?
m. Enter the host controller IP addresses separated by comma Eg: 10.101.xxx.xxx,10.101.xxx.xxx: Enter 10.101.249.144,10.101.249.150
n. What is the Oracle listener hostname []? Enter <RDS IP/Host Name>
o. What is the Oracle listener port number []? Enter 1555
p. What is the Oracle SID [AVDB]?
q. Is the Oracle Service name the same as the Oracle SID [AVDB]? (yes/no)
r. Enter the AVUSER username [avuser]?
s. Enter the AVUSER password:
t. Confirm the AVUSER password:
u. Enter the AVDWUSER username [avdwuser]?
v. Enter the AVDWUSER password:
w. Confirm the AVDWUSER password:
x. Enter the ACMDB username [ACMDB]?
y. Enter the ACMDB password:
z. Confirm the ACMDB password:
aa. Enter the PERFSTAT username [perfstat]?
bb. Enter the PERFSTAT password ?
cc. Confirm the PERFSTAT password ?

After the successful installation, ensure that the Domain node is running.

Install a Host Node

Make sure Domain node is running. Follow the steps below to install a Host node.

1. Run `./install.sh` under /tmp/aveksa/staging/deploy directory.
2. Respond to the following installer prompts:

a. Have you reviewed the release notes and agree to the license terms (yes or no)? Enter "yes".
b. What is the location for installation [/home/oracle]?
c. Where are the package files located [/tmp/aveksa/packages]?
d. Do you want to install the application as a service (yes or no)? Enter "yes".
e. Do you want to setup a wildfly cluster (yes or no)? Enter "yes".
f. Are you configuring the Domain Controller (yes -> Domain Controller / no -> Host Controller)? (yes or no)? Enter "no".
g. Enter the Domain Controller Hostname or IP Address:Enter <Domain Controller IP>
h. What is the Wildfly Management User Name (Same as the one provided when configuring the Domain node) [AveksaClusterAdmin]?
i. Enter the Wildfly Management User Password (Same as the one provided when configuring the Domain node):
j. Confirm the Wildfly Management User Password:
k. What is the Wildfly Host Controller Name (Should be unique in the cluster) [master]? Enter secondary-1
l. What is the Wildfly Server Name (Should be unique in the cluster) [img-server-1]? Enter img-server-2
m. What is the Wildfly Server Group Name (Should be same across the cluster) [img-servergroup]?
n. What is the Oracle listener hostname []? Enter <RDS IP/Host Name>

o. What is the Oracle listener port number []? Enter 1555
p. What is the Oracle SID [AVDB]?
q. Is the Oracle Service name the same as the Oracle SID [AVDB]? (yes/no)
r. Enter the AVUSER username [avuser]?
s. Enter the AVUSER password:
t. Confirm the AVUSER password:
u. Enter the AVDWUSER username [avdwuser]?
v. Enter the AVDWUSER password:
w. Confirm the AVDWUSER password:
x. Enter the ACMDB username [ACMDB]?
y. Enter the ACMDB password:
z. Confirm the ACMDB password:
aa. Enter the PERFSTAT username [perfstat]?
bb. Enter the PERFSTAT password ?
cc. Confirm the PERFSTAT password ?

Maintaining a Wildfly Cluster

Add a New Host Node to a Cluster

Perform the following steps to add a new host node to a cluster.

Before you begin

Set up the host node as described in Clustering on AWS

Perform the following steps to ADD a host node to a cluster

1. Stop the aveksa service on all nodes.
   ```
   service aveksa_server stop
   ```
2. Edit domain.xml and add the new host IP or FQDN under the under system-properties tag.
3. Save domain.xml.
4. Start the domain and host nodes.
5. To test the new host node, access the ACM URL for the newly added node.

Remove a Host Node from a Cluster

Perform the following steps to remove a host node from a cluster.

1. Log in to the host node to be removed.
2. As Root user, run uninstall.sh under AVEKSA_HOME/deploy folder.
3. When the uninstallation is complete, log in to the domain node.
4. Stop the aveksa service.
   ```
   service aveksa_server stop
   ```
5. Edit the domain.xml and remove the deleted host node under the under system-properties tag.
6. Save the domain.xml.
7. Restart the domain node.

Performance Results on AWS (v7.5)

Overview

This chapter summarizes performance tests run in two configurations of an Amazon Web Services (AWS) environment, db.r4.2xlarge and db.r4.4xlarge, and the results for RSA Identity Governance and Lifecycle. ACME Corporation is a large, fictional enterprise that provides the basis for the test data and use cases.

Performance was evaluated with the remote database deployed on RDS instance in AWS and EC2 instance as Appserver.

An AWS deployment of RSA Identity Governance and Lifecycle comprises the following components:

- A remote Oracle database deployed using an Amazon Relational Database Service (RDS) instance or an RSA-provided Oracle database deployed in an Amazon EC2 instance. The remote database must adhere to the same requirements specified in the RSA Identity Governance and Lifecycle Database Setup and Maintenance Guide.
- One or more RSA Identity Governance and Lifecycle nodes deployed on Amazon EC2 instances.
- A Virtual Private Cloud (VPC) to allow communication between the nodes and the database.

AWS Configuration

	EC2 Instance	RDS Instance
2xlarge Configuration	Instance type: t2.xlarge 4 vCPUs 2.3 GHz Intel Broadwell E5-2686v4 16 GiB memory EBS only - 30 GB storage	Instance type: db.r4.2xlarge 8 CPUs 61 GiB memory 201 GB storage Engine Version: 12.1.0.2.v15
4xlarge Configuration	Instance type: t2.xlarge 4 vCPUs, 2.3 GHz, Intel Broadwell E5-2686v4 16 GiB memory EBS only - 30 GB storage	Instance type: db.r4.4xlarge 16 CPUs 122 GiB memory 201 GB storage Engine Version: 12.1.0.2.v16

To deploy RSA Identity Governance and Lifecycle in an AWS environment, see the *RSA Identity Governance and Lifecycle 7.5 AWS Installation and Clustering Guide.*

Executive Summary

This section highlights the performance test results in two configurations of an Amazon Web Services (AWS) environment, db.r4.2xlarge and db.r4.4xlarge using the deployment metrics based on the ACME Specification described previously.

The performance comparison is to provide adequate insight into the performance of different AWS configurations and how they specifically perform. This allows customers to make an informed decision about which environment to invest in.

Collections

- Average collection time in 4.x large instance is 23 mins and in 2.x large instance is 29 mins
- Total collection time in 4.x large is 2hrs 25 minutes and in 2.x large instance is 2hrs 54 mins.

Volume Tests with Single User

- Importing Business Descriptions increased linearly in processing time with the number of descriptions. 222,500 Business Descriptions imported in about 147 seconds in 2.x large instance and 117 seconds in 4.x large instance.
- Annual Review generation for accounts with as many as 900,000 entitlements took approximately 11 minutes in 2.x instance and 10 minutes in 4.x instance.
- Processed 100 SoD rules in 15 seconds in 2.x large instance and 13 seconds in 4.x large instance.
- Unauthorized Change Detection feature (Rule) detected and processed 30,000 unauthorized changes in 49 minutes in 2.x large instance and 46 minutes in 4.x large instance.
- Suggested roles in 4 minutes and committed changes in 80 minutes in 2.x large instance. And suggested roles in 3 minutes and committed changes in 67 minutes.

Tests with Concurrent Users

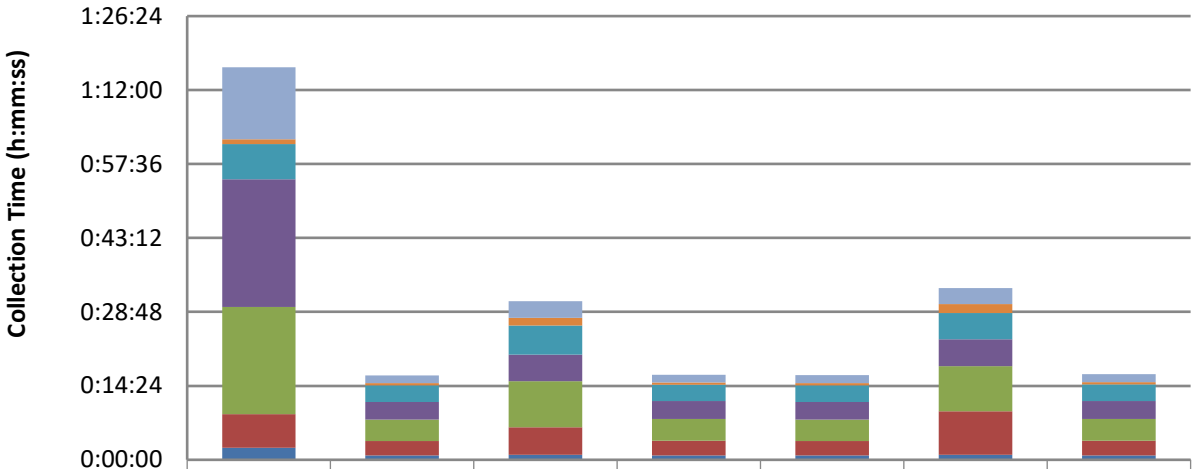
- Access Request and User Review scenarios met the server response time target with 500 concurrent users for both 2.x large and 4.x large instances
- All the requests in the User Review scenario had less than one second of server response time. The overall iteration (end to end flow, excluding user input) completed in about 1.28 seconds in 2.x large instance and 1.26 seconds in 4.x large instance.

WildFly Collections in the AWS Environment

ACME Corporation needs to collect all of the identity, account, entitlement and role information in their RSA Identity Governance and Lifecycle 7.5.0 environment in order to remain up-to-date with identity and access changes. The collection time for all their collectors helps determine their maintenance window. If they know the time their collectors may take, they will know when to avoid impacting their users' work in the system.

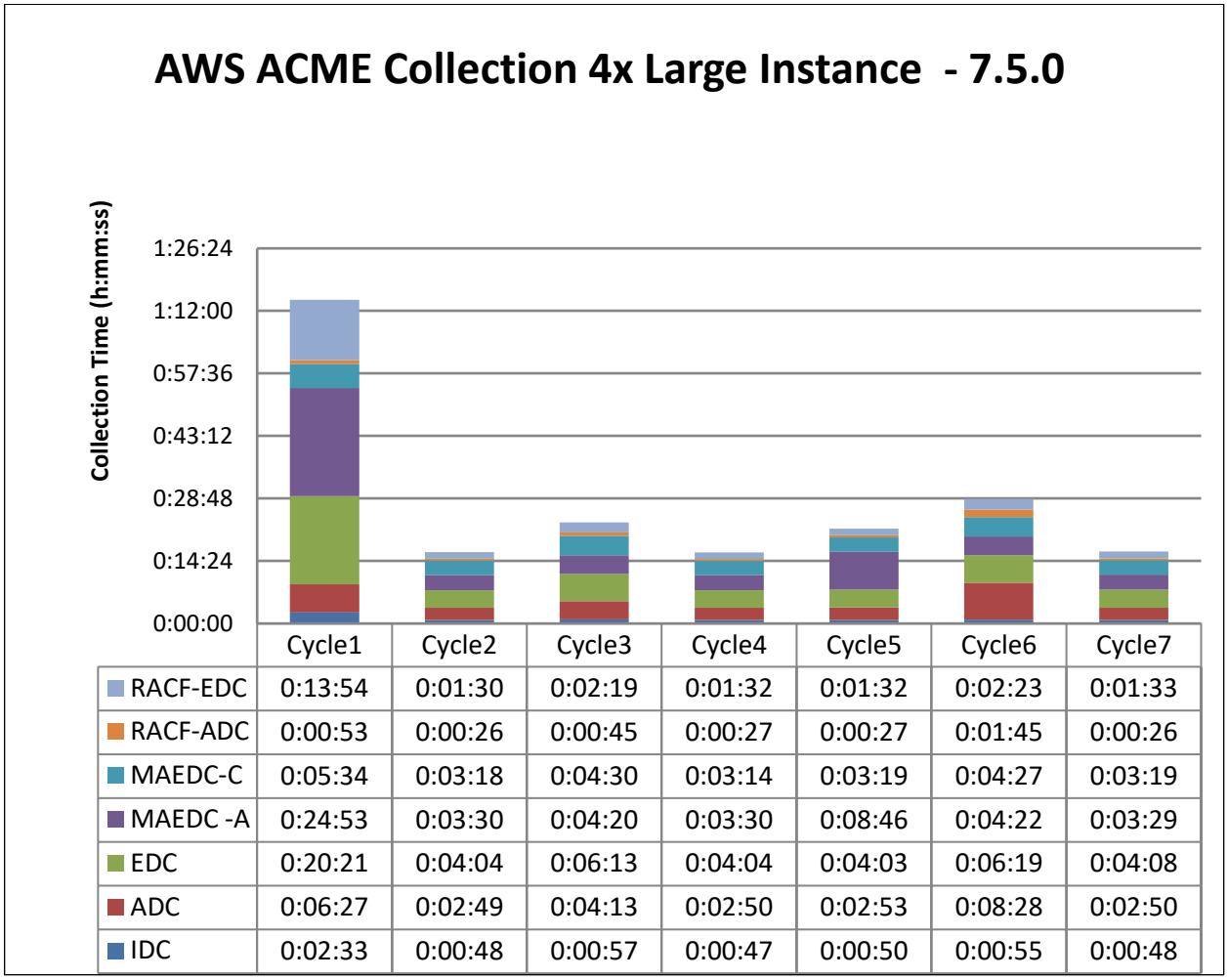
The following two charts show the collection time for each of the AWS configurations in a WildFly environment. This testing included seven collection cycles with a one percent change in in Identity, Account and Entitlements data for the third and sixth cycles.

RDS instance type: r4.2xlarge



	Cycle1	Cycle2	Cycle3	Cycle4	Cycle5	Cycle6	Cycle7
RACF-EDC	0:14:02	0:01:29	0:03:14	0:01:29	0:01:31	0:03:07	0:01:32
RACF-ADC	0:00:56	0:00:26	0:01:30	0:00:26	0:00:27	0:01:46	0:00:27
MAEDC-C	0:06:51	0:03:14	0:05:43	0:03:14	0:03:14	0:05:07	0:03:15
MAEDC -A	0:24:51	0:03:27	0:05:10	0:03:29	0:03:27	0:05:13	0:03:30
EDC	0:20:52	0:04:10	0:08:55	0:04:12	0:04:08	0:08:47	0:04:12
ADC	0:06:35	0:02:48	0:05:22	0:02:51	0:02:49	0:08:28	0:02:51
IDC	0:02:20	0:00:51	0:01:00	0:00:53	0:00:52	0:01:00	0:00:53

RDS instance type: r4.4xlarge



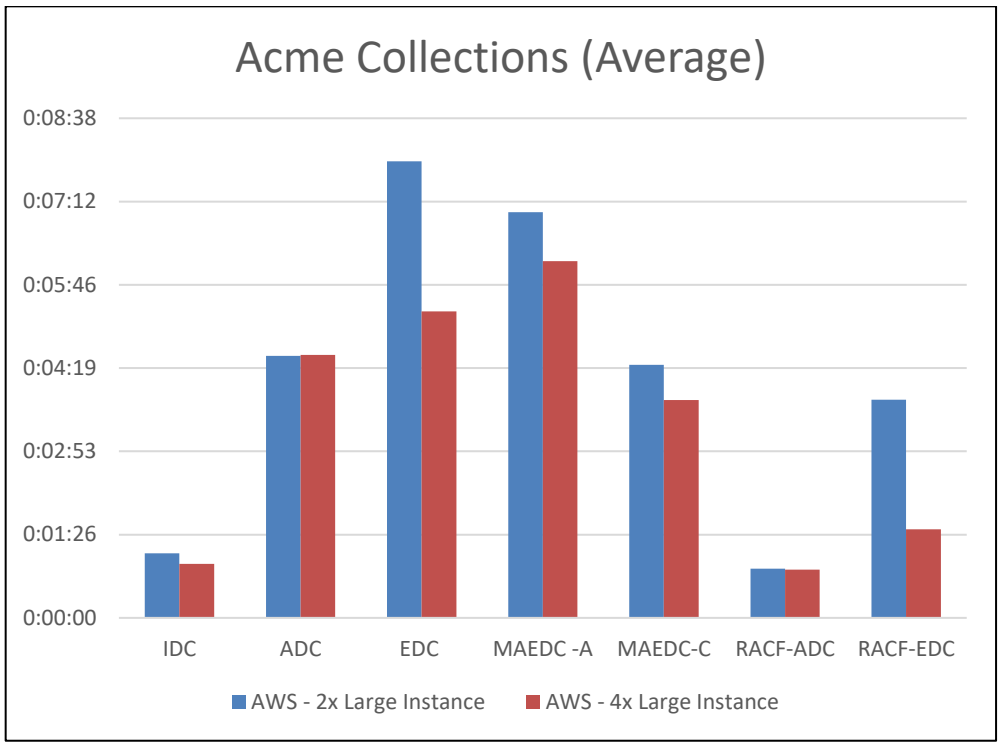
	Cycle1	Cycle2	Cycle3	Cycle4	Cycle5	Cycle6	Cycle7
RACF-EDC	0:13:54	0:01:30	0:02:19	0:01:32	0:01:32	0:02:23	0:01:33
RACF-ADC	0:00:53	0:00:26	0:00:45	0:00:27	0:00:27	0:01:45	0:00:26
MAEDC-C	0:05:34	0:03:18	0:04:30	0:03:14	0:03:19	0:04:27	0:03:19
MAEDC -A	0:24:53	0:03:30	0:04:20	0:03:30	0:08:46	0:04:22	0:03:29
EDC	0:20:21	0:04:04	0:06:13	0:04:04	0:04:03	0:06:19	0:04:08
ADC	0:06:27	0:02:49	0:04:13	0:02:50	0:02:53	0:08:28	0:02:50
IDC	0:02:33	0:00:48	0:00:57	0:00:47	0:00:50	0:00:55	0:00:48

The results for collections in a WildFly environment indicate the following:

- The average collection time in the 4x instance was better than the collection time in the 2x instance.
- In the 2x instance, RSA Identity Governance and Lifecycle collected 250,000 users in about one minute, 650,000 accounts in about 4 minutes, and approximately 15 million entitlements (pre-exploded) in about 19 minutes.
- In the 4x instance, RSA Identity Governance and Lifecycle collected 250,000 users in about one minute, 650,000 accounts in about 4 minutes, and approximately 15 million entitlements (pre-exploded) in about 7 minutes.

Average Collection Performance comparison

The following graph compares the average collection performance between 2x Large and 4X Large instances. The tests ran seven collection cycles with a one percent delta change in the data for the third and sixth.

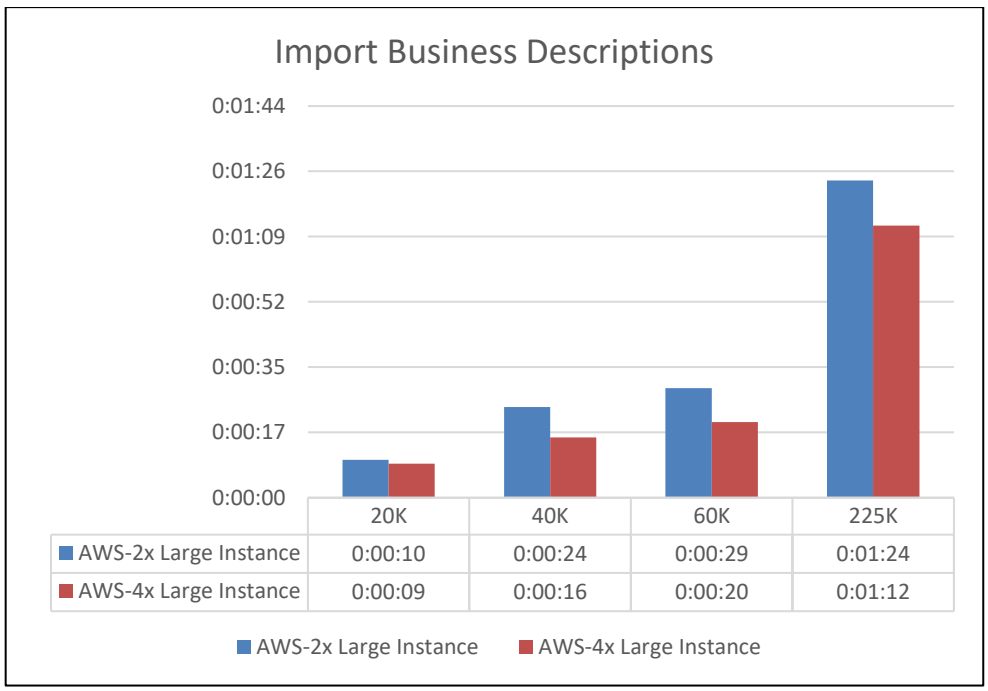


Single User Volume Tests

The assorted performance scenarios for ACME involve a high volume of data processing from a single user.

Business Description Imports

The following graph plots ACME's import time in the AWS 2x and 4x instances against the number of business descriptions in a text file.

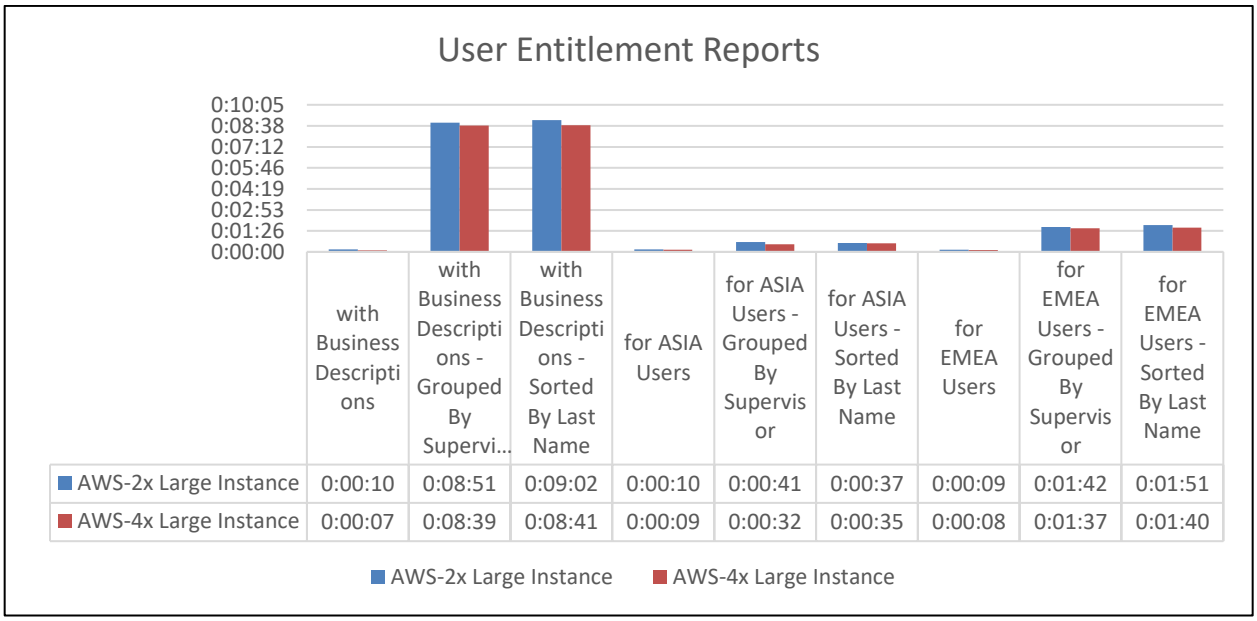


	20K	40K	60K	225K
AWS-2x Large Instance	0:00:10	0:00:24	0:00:29	0:01:24
AWS-4x Large Instance	0:00:09	0:00:16	0:00:20	0:01:12

Reports

After unification, orphan account cleanup, and import of business descriptions, ACME runs a few basic reports on user entitlements and archives them in a CSV file. The following report compares shows the user entitlement reports that

ran against 2x and 4x instances.



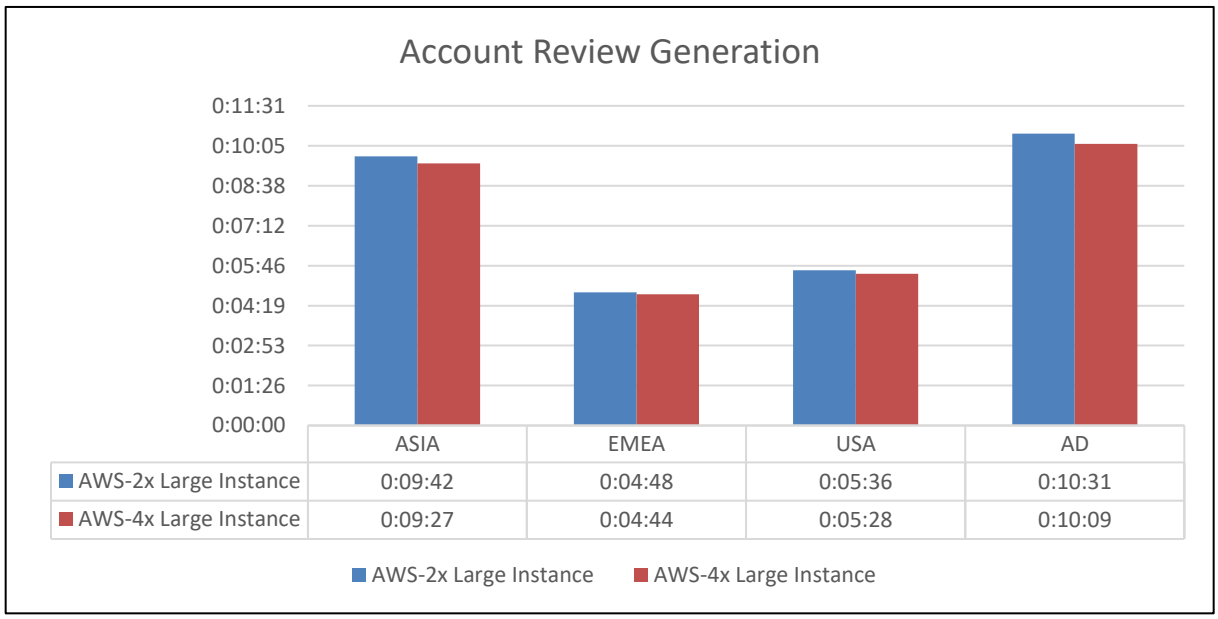
	with Business Descriptions	with Business Descriptions - Grouped By Supervi...	with Business Descriptions - Sorted By Last Name	for ASIA Users	for ASIA Users - Grouped By Supervisor	for ASIA Users - Sorted By Last Name	for EMEA Users	for EMEA Users - Grouped By Supervisor	for EMEA Users - Sorted By Last Name
AWS-2x Large Instance	0:00:10	0:08:51	0:09:02	0:00:10	0:00:41	0:00:37	0:00:09	0:01:42	0:01:51
AWS-4x Large Instance	0:00:07	0:08:39	0:08:41	0:00:09	0:00:32	0:00:35	0:00:08	0:01:37	0:01:40

Reviews

After collecting the necessary access information, ACME runs multiple recertification processes to verify access and remove it as needed. ACME needs to plan its maintenance window accordingly.

Annual Review Generation Performance

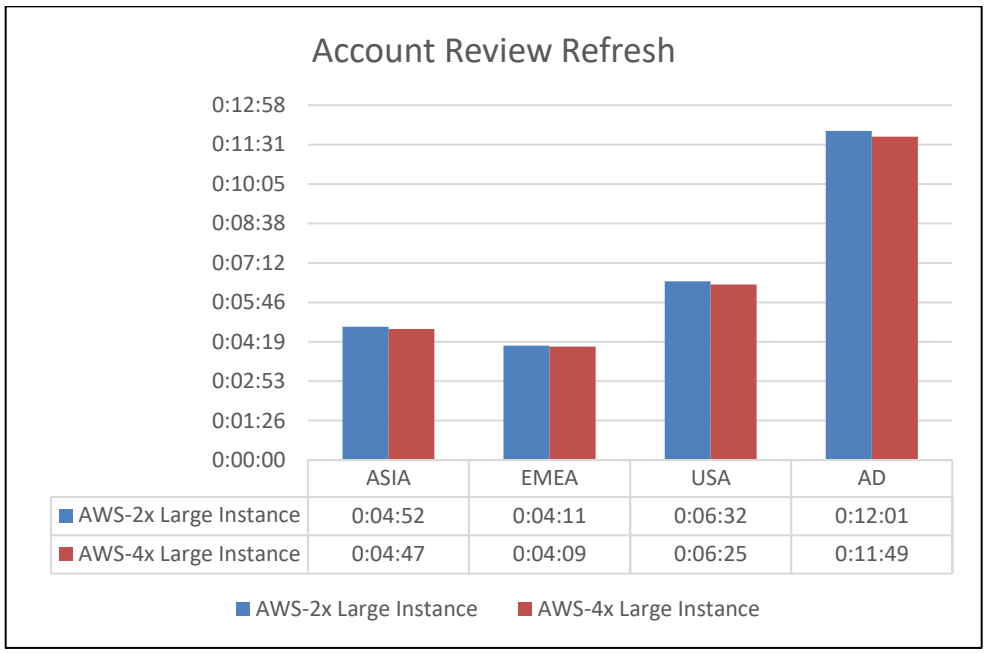
The ACME system generates three annual reviews for four different business units.



	ASIA	EMEA	USA	AD
AWS-2x Large Instance	0:09:42	0:04:48	0:05:36	0:10:31
AWS-4x Large Instance	0:09:27	0:04:44	0:05:28	0:10:09

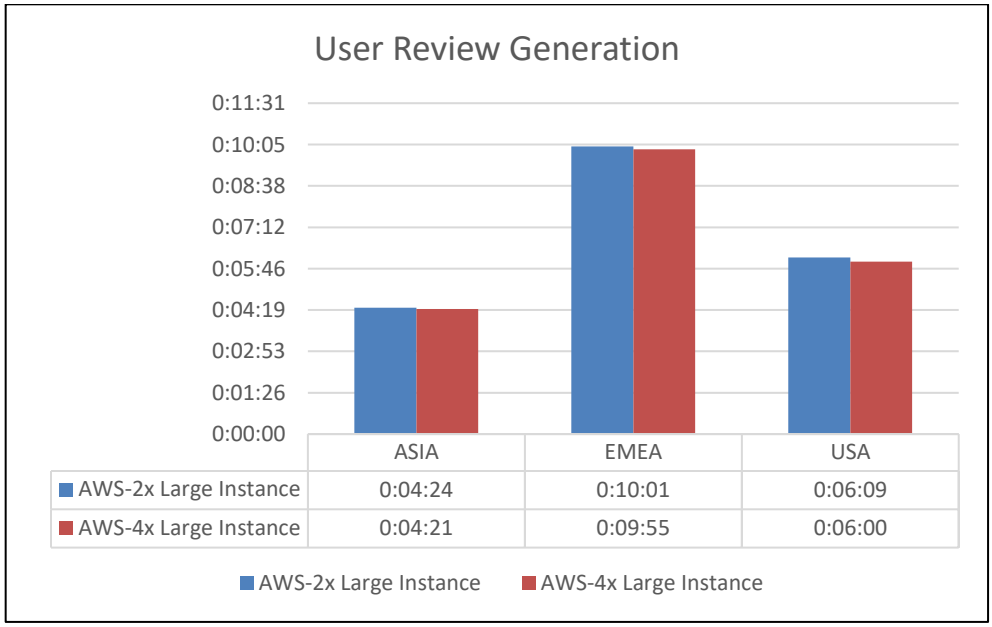
Annual Review Refresh

ACME refreshes the comprehensive, annual reviews once a week during the eight-week cycle. ACME wants to know how long that would take, assuming that 5 percent of the entitlements change on every refresh.



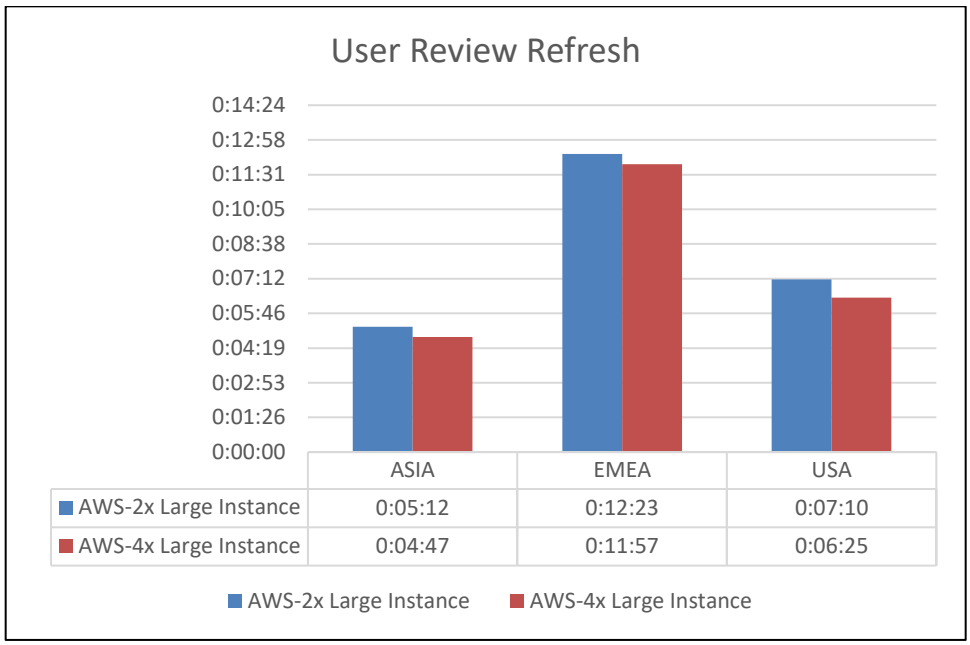
User Review Generation Performance

The ACME system generates three User reviews for three different business units.



User Review Refresh

ACME refreshes the comprehensive user reviews once a week during the eight-week cycle. ACME wants to know the time taken, assuming that 5 percent of the entitlements change on every refresh.

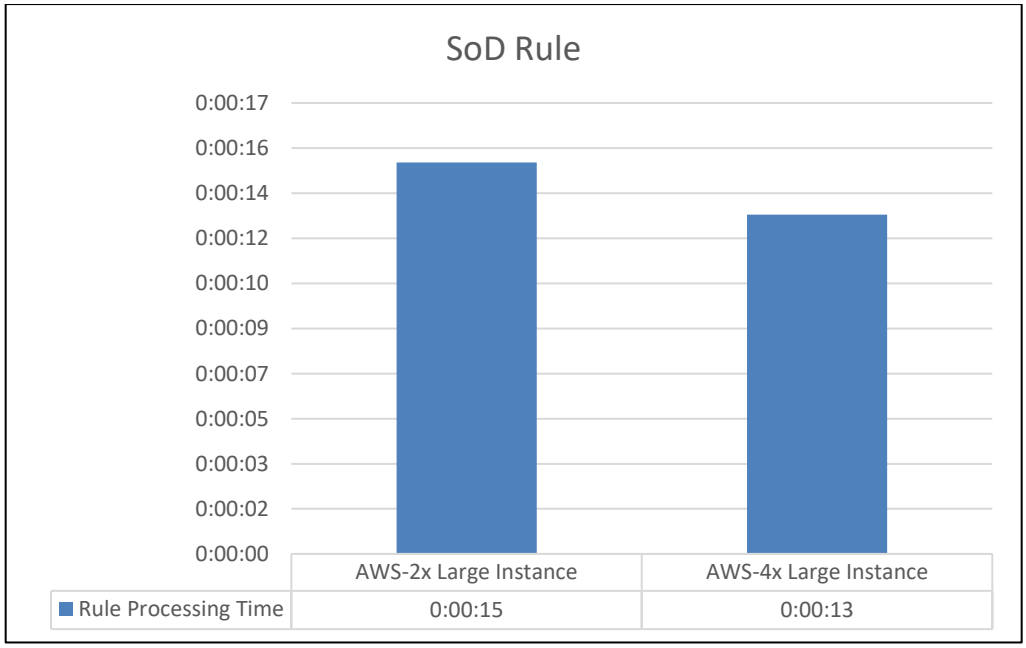


Rules and Roles

Separation of Duty (SoD) Rule Processing

The following chart show the rule processing time for AWS 2x and 4x instances for 100 SoD rules.

Processing time of 100 SOD rules



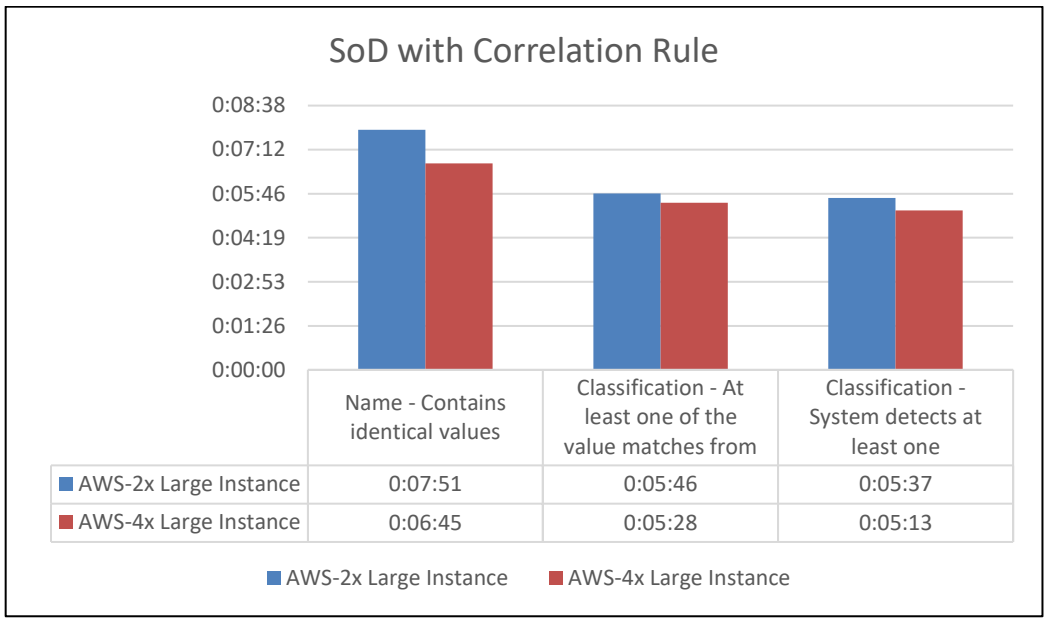
SoD Rule Processing with correlation condition

When processing SOD an optional constraint will be defined on business source attributes. For SOD Rule definition, the user has the option of not selecting any Correlation Specification Option or any of the three. The logic associated with the options are as follows:

- Match for exact values between set1 and set2.
- One of the values matches (Match for any of the common value between set1, set2 and rule definition value).

- System detects at least one common value (Match for any of the common value between set1 and set2). Performance testing of the feature defining the SoD rule with the correlation condition and processing that rule.

The following chart shows SoD with different correlation condition options:

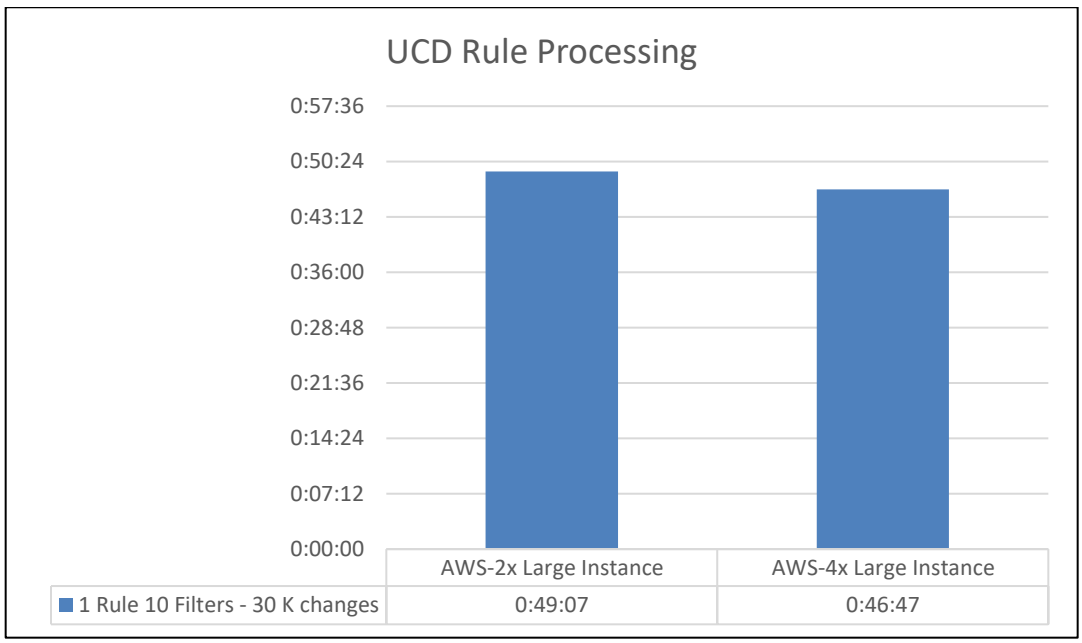


Unauthorized Change Detection (UCD) Rule Processing

UCD checks the changes in the account access found through collections against the change requests in RSA Identity Governance and Lifecycle. Changes without a change request on record are flagged as unauthorized.

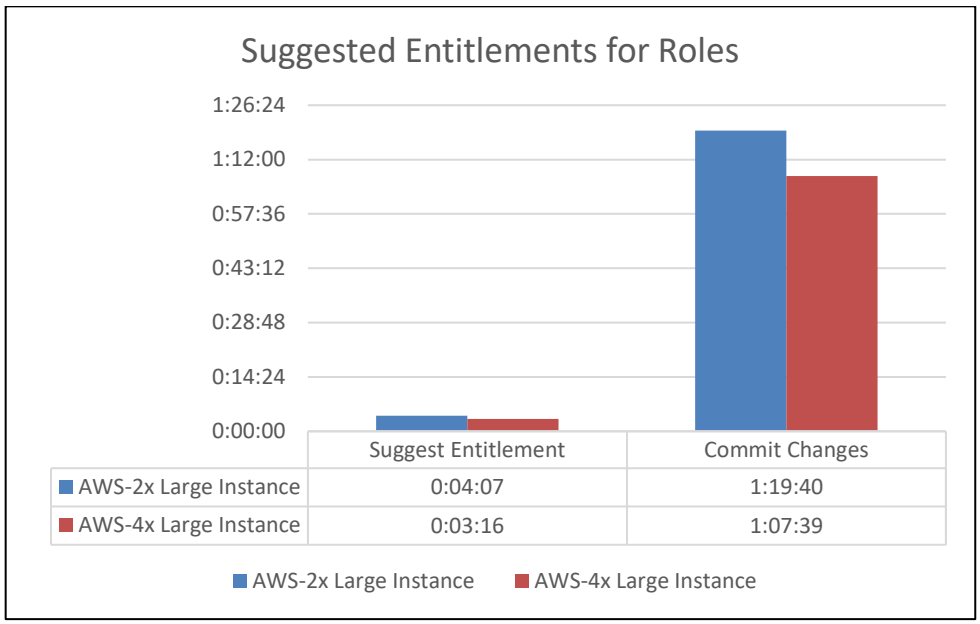
1 Rule – 10 Filters

The ACME system applies 1 UCD rule with 10 filters to detect unauthorized access in any of the 10 applications. Collections run to detect additional unauthorized access. The ACME system generates a change request for each unauthorized change to revoke access. Tests for 30,000 unauthorized changes run with no other change requests in queue.



Suggest Entitlements for Roles

An ACME Role Analyst performs the 'Suggest and add entitlements' operation for each of the roles that were created for each business unit. The following table records the time taken to commit the changes to 500 roles.



500 concurrent users test on AWS

The following sections describe the results of performance tests using 500 concurrent users on AWS deployment.

The load pattern of this test is as follows:

- 300 users gradually ramped up over the course of 30 minutes, followed by 20 minutes of steady state.
- Added another 100 users with a ramp up of 10 minutes, followed by 20 minutes of steady state.
- Added the final group of 100 users with a ramp up of 10 minutes, followed by 30 minutes of steady state with the total of 500 users.

Access Requests on AWS with 500 concurrent users

RDS instance type: r4.2xlarge

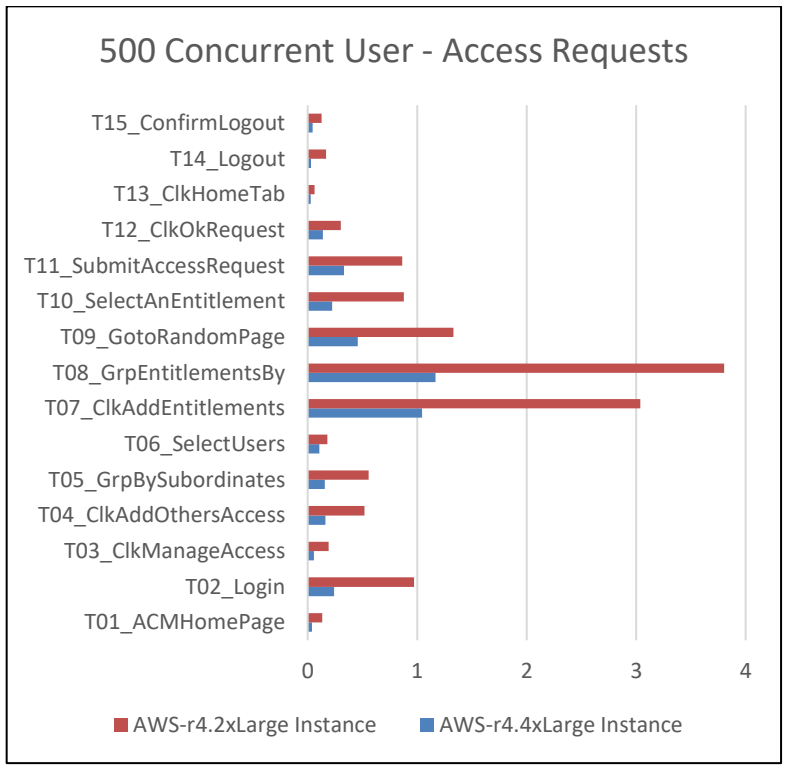
Response Time (Seconds)				
Transactions	**Minimum**	**Average**	**Maximum**	**95th Percentile**
ACM Home Page	0.006	0.133	17.576	0.683
Login	0.09	0.972	30.474	3.623
Click Manage Access	0.022	0.192	18.713	0.698
Click Add Others Access	0.078	0.518	21.557	1.574
Group By Subordinates	0.072	0.557	21.61	1.781
Select Users	0.004	0.18	32.338	0.224
Click Add Entitlements	0.451	3.038	40.415	9.267
Group Entitlements By	0.652	3.804	30.734	11.725
Goto Random Page	0.233	1.332	20.116	4.167
Select An Entitlement	0.119	0.877	21.676	3.203

Submit Access Request	0.035	0.864	21.73	2.762
Click Ok Request	0.032	0.302	20.827	0.392
Click Home Tab	0.004	0.064	9.94	0.111
Click Logout	0.002	0.167	21.654	0.458
Confirm Logout	0.011	0.128	21.362	0.221

RDS instance type: r4.4xlarge

Response Time (Seconds)				
Transactions	**Minimum**	**Average**	**Maximum**	**95th Percentile**
ACM Home Page	0.008	0.04	16.282	0.038
Login	0.092	0.242	17.038	0.484
Click Manage Access	0.022	0.057	11.627	0.09
Click Add Others Access	0.08	0.161	14.646	0.294
Group by Subordinates	0.076	0.155	14.937	0.276
Select Users	0.005	0.107	29.816	0.024
Click Add Entitlements	0.464	1.044	29.244	1.902
Group Entitlements By	0.661	1.168	18.81	2.137
Go to Random Page	0.234	0.457	16.64	0.857
Select an Entitlement	0.116	0.223	16.456	0.412
Submit Access Request	0.106	0.332	15.473	0.755
Click Ok Request	0.01	0.138	15.061	0.191
Click Home Tab	0.004	0.028	14.159	0.015
Click Logout	0.002	0.03	14.591	0.007
Confirm Logout	0.011	0.045	15.22	0.056

Performance Comparison



Observations

- The average response time of concurrent user test with 2x Large instance is very high compared to 4x Large instance.
- The user CPU usage of the ACM host with a 500-user load was approximately 6%, and the usage of the database host at the same load was approximately 4%.

User Reviews on AWS with 500 concurrent users

RDS instance type: r4.2xlarge

Response Time (Seconds)				
Transaction Name	Minimum	Average	Maximum	95th Percentile
ACM Home Page	0.006	0.03	13.99	0.013
Login	0.055	0.19	16.08	0.246
Click Review Link	0.012	0.047	15.14	0.026
Click Perform Review	0.199	0.331	15.26	0.521
Navigate to Random Entitlement Page	0.036	0.288	12.96	0.491
Single Entitlement Review Maintain	0.04	0.08	13.95	0.087
Single Entitlement Review Revoke	0.153	0.254	14.95	0.313
Logout	0.002	0.026	13.34	0.006
Confirm Logout	0.01	0.037	13.86	0.025

RDS instance type: r4.4xlarge

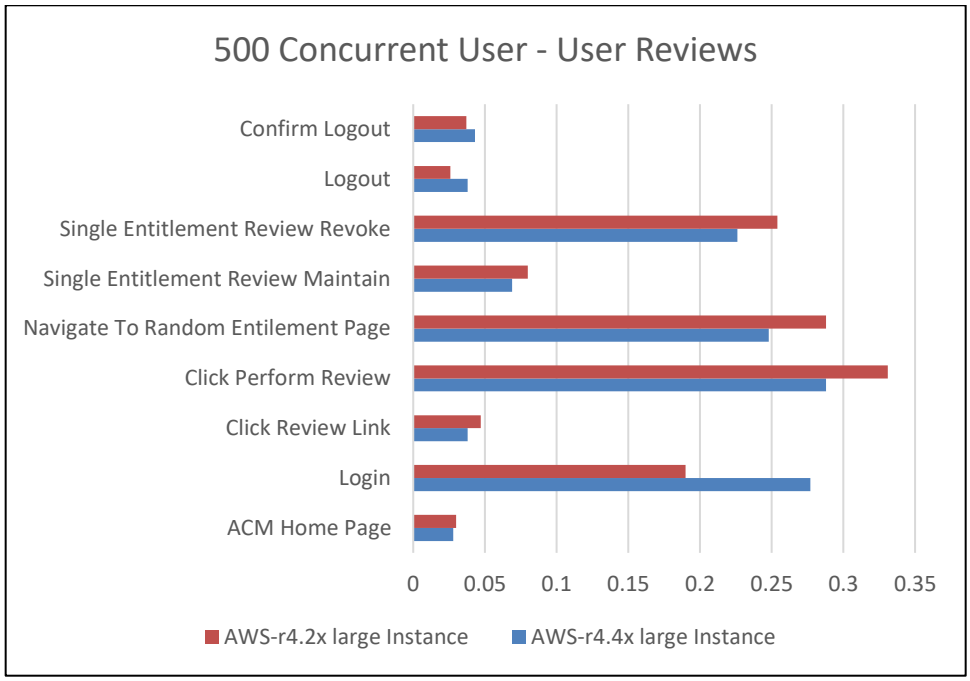
Response Time (Seconds)				
Transaction Name	Minimum	Average	Maximum	95th

				Percentile
ACM Home Page	0.003	0.028	14.438	0.007
Login	0.095	0.277	17.178	0.431
Click Review Link	0.011	0.038	15.753	0.019
Click Perform Review	0.198	0.288	16.335	0.362
Navigate to Random Entitlement Page	0.034	0.248	16.112	0.343
Single Entitlement Review Maintain	0.038	0.069	14.537	0.064
Single Entitlement Review Revoke	0.14	0.226	16.427	0.248
Logout	0.001	0.038	16.368	0.004
Confirm Logout	0.01	0.043	15.859	0.019

Observations

- The average response time of concurrent user test with 2x Large instance is high compared to 4x Large instance.
- The average and 95th percentile response time for all user review transactions was less than one second. CPU utilization and Memory utilization was normal.

Performance Comparison



ACME Specification

Identity Information

Total Business Units: 4 (USA, EMEA, ASIA, Latin America)
Total Users: 250,000 (225,000 users are employees, 25,000 users are contractors)

- USA -149,000 Users (1 Supervisor for every 10 users) - 200 Job Titles.
- ASIA – 10,000 Users (1 Supervisor for every 2 users) - 500 Job Titles.
- EMEA – 90,000 Users (1 Supervisor for every 11 users) - 2000 Job Titles.
- Latin America – 1,000 Users (1 Supervisor for every 10 users) - 50 Job Titles.

Account Information

ACME has 675,000 Accounts distributed across the following Account Data Collectors (ADCs):

- ADC_AD - Covers 250,000 Users and 300,000 accounts (1 account per user and 50,000 Orphan/System Accounts), 60,000 groups, 1st Group having 50,000 users, groups 2-11 having 2,000 users. Rest of the groups have 3 users per group.
- ASIA_ADC - Covers 10,000 users and 11,000 accounts (1 account per user in ASIA Business Unit and 1,000 Orphan accounts).
- EMEA_ADC - Covers 90,000 users and 99,000 accounts (1 account per user in EMEA BU and 9,000 orphan accounts).
- USA1_ADC - Covers 70,000 users and 77,000 ccounts (1 account per user and 7,000 Orphan Accounts).
- USA2_ADC - Covers 80,000 users and 88,000 accounts (1 account per user and 8,000 Orphan Accounts).
- RACF_ADC - Covers 10,000 users across all BUs and 100,000 accounts (10,000 accounts are Orphans).

Application Information

ACME has 1,000 Applications under management:

- Across all 1,000 applications there are 15 Million user entitlements, including 20,000 application roles. ACME's application entitlements are managed as follows:
    - ***App Entitlement Database A***: Has entitlements for 500 applications and 5 million user entitlements (including application roles) across applications.
        - 100 of the Applications have 100 Application Roles each.
        - Ads Applications are marked as SOX (classification attribute).
        - 400 of the Applications have 20 Application Roles each.
        - Exploded entitlements: 10,996,701
        - Granular Entitlements: 4,498,684
    - ***App Entitlement Database C***: Has entitlements for 210 applications and 1 million user entitlements across applications
        - Exploded entitlements: 808,000
        - Granular Entitlements: 808,000
    - ***10 Individual Application Databases***: Each has 100,000 user entitlements
        - **HugeApplication1 – 4:** 183,750 Exploded entitlements, 87,500 Granular entitlements (each)
        - **HugeApplication5 – 7:** 189,000 Exploded entitlements, 90,000 Granular entitlements (each)
        - **HugeApplication8 – 9:** 190,910 Exploded entitlements, 90,910 Granular entitlements (each)
        - **HugeApplication10:** 190,908 Exploded entitlements, 90,908 Granular entitlements

RACF Mainframe - ACME's mainframe environment has 7.03 million user entitlements. The mainframe environment has 10,000 user profiles that have access to 100,000 mainframe accounts.

- ACME maintains a separate database repository to manage associated business descriptions for all granular entitlements in the organization. This business description repository is updated quarterly by the application data stewards.
- App Roles: 10,000
- App Roles Members: 222,500
- Exploded Entitlements: 12,673,000
- Granular Entitlements: 5,670,000