SAP on Amazon Web Services
High Availability Guide

Author: Amazon Web Services
sap-on-aws@amazon.com

Version: 3.3 – August 2016

Contents

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Document History

Important
Before installing the SAP solution on AWS, make sure to have the latest version of this document. The latest version can be found at http://aws.amazon.com/sap/ in the technical content section.

Version	Date	Description
1.0	9/18/2012	Document created
2.0	9/22/2012	Extension into multiple AZs for increased availability & durability. SAP Web Dispatcher & external access information
2.1	9/25/2012	Minor edits
3.0	10/5/2012	Removed AWS component level documentation
3.1	7/10/2013	Consolidated SAP notes
3.2	16/12/2014	Minor edits, updates
3.3	26/08/2016	Updated Oracle support and HANA DB support

About This Guide

The intent of this guide is to provide an overview of how to configure SAP systems on Amazon Elastic Compute Cloud (Amazon EC2) in such a way as to be able to protect the application from various single points of failure. This guide will explore how features native to the Amazon Web Services (AWS) platform in combination with SAP installation techniques can greatly improve the availability of an SAP deployment. This guide is not meant to be an exhaustive list of all possible configuration options but is meant to serve as a guide for solutions common to most deployment scenarios.

What is not included in this guide…

Please note that this guide is not intended to replace the standard SAP installation guides, as well as operating system and/or RDBMS documentation. In addition, much of this guide builds on concepts and Amazon EC2 technology components discussed in both the SAP on AWS Operation and Implementation guides.

Apart from some examples, this guide does not include detailed instructions on how to configure databases for high availability, for instance using SQL Server Mirroring or Oracle Data Guard. Please refer to the specific documentation for RDBMS high availability features for setup and configuration.

This guide is also not intended to cover all the business reasons and decision criteria for deciding whether or not to implement a high-availability solution on AWS.

Prerequisite Documentation

To ensure SAP systems are installed on the AWS platform in manner consistent with SAP's support requirements, it is recommended to first begin installation planning by referring to the standard SAP documentation and notes for each respective SAP solution being installed.

- http://service.sap.com/instguides
- http://service.sap.com/notes

SAP on Amazon Web Services

This guide assumes that the reader is already familiar with implementing and operating SAP solutions on the Amazon Web Services infrastructure. Please be sure to read the SAP on AWS Implementation Guide and the SAP on AWS Operations Guide before continuing. All AWS guides for SAP can be found at http://aws.amazon.com/sap.

Table 1 lists the available SAP notes for deploying SAP on AWS infrastructure.

Table 1: SAP notes for deploying SAP on AWS

Note #	Description
1588667	SAP on AWS: Overview of related SAP Notes and Web-Links
1656099	SAP on AWS: Supported SAP, DB/OS and AWS EC2 products
1656250	SAP on AWS: Support prerequisites

Feedback on this guide can be sent to sap-on-aws@amazon.com.

Other general documentation that may be helpful:

SAP on Linux

- SAP SCN Page on Linux
- SAP on Linux in general (FAQ)
- Red Hat Enterprise Linux Knowledge Repository
- SUSE Linux Enterprise Server 11 documentation

SAP Note #	Description
171356	SAP Software on Linux: Essential information

SAP on Windows

- SAP SCN Page on Microsoft Windows
- SAP on Windows Server 2008 R2 (FAQ)
- SAP on Windows Server 2012 (FAQ)

SAP Note #	Description
1486772	SAP Systems on Windows Server 2008 R2
1732161	SAP Systems on Windows Server 2012 2012R2
1564275	How to install an SAP System or SAP components on Windows using Virtual Host names

Overview of the High Availability Concept

High availability or HA solutions are designed to protect the single points of failure (SPoF) of a software system. As previously stated, there are some key differences in the high-availability concept on the AWS platform as compared to conventional software cluster-based solutions, the two primary differences being shared-storage devices and the restriction of broadcast or multicast traffic within the AWS network.

Traditional clustering solutions typically use shared storage devices for quorum or "lock devices" as well as shared application and/or database volumes. These volumes are usually presented to multiple hosts with read/write access being controlled by the cluster software. Conversely, an Amazon Elastic Block Store (Amazon EBS) volume can only be attached to one instance at a given time. This is for both write consistency and redundancy for the Amazon EBS volume.

In addition, many clustering solutions (Microsoft Clustering, for example) use layer 2 network functions including multicast or broadcast packets to check for node failures within the cluster. This type of traffic is not allowed in Amazon Virtual Private Cloud (Amazon VPC), which results in many cluster solutions not being able to function correctly. Furthermore, these solutions also rely on the ability to swap IP addresses from within the guest operating system, which is also not currently supported. Before discussing the specifics of the high-availability solution, it's important to understand the classic single points of failure in an SAP system. These components of the system are critical for operation and include functions such load balancing, license management, and lock management. The SAP application does have built-in redundancy for many of the other key components through its ability to distribute and scale out dialog instances.

SAP Environment SPoF
Message Server - (A)SCS instance
Enqueue Server - (A)SCS instance
SAP Web Dispatcher
Database

Native SAP High Availability
ABAP Dialog and Batch work processes
Update work processes
Gateway work processes
Spool work processes
J2EE cluster nodes

Further information on SAP high availability scenarios can be found on the SCN High Availability web page.

High Availability for SAP on AWS

The concepts discussed in this guide will attempt both to provide a high-availability solution that closely resembles a typical on-premise installation, and to show how features delivered by the AWS platform in combination with SAP installation options allow for a high-availability solution that extends beyond a single datacenter.

The high-availability solution as discussed in this guide will include the following key components:

- Installing or copying the SAP system into an Amazon VPC, while leveraging multiple subnets and Availability Zones.
- Distributing the various SAP application and database components onto multiple instances.
- Using the SAP installer option SAPINST_USE_HOSTNAME for virtual hostnames, which are then mapped to static IP addresses via DNS.
- Using a secondary AWS Elastic Network Interface (ENI) to relocate the aforementioned IP address associated with the virtual hostname from one AWS instance to another.
- Using AWS security groups in combination with the re-locatable ENI to properly place a virtual network "fence" around the SAP central services and database instances so as to avoid miscommunication and to isolate failed resources.
- Using database stand-by and/or mirroring techniques to protect the database layer.
- Leveraging the ability to re-map Amazon EBS volumes from one instance to another to relocate global SAP file systems for failover of the SAP central services instance within a single Availability Zone.
- Protection of the SAP central services instance in the event of a complete Availability Zone failure.
- Use of Amazon Machine Images (AMIs) to quickly provision additional SAP instances for capacity or to cover failures.

Planning the Deployment

Planning the deployment beforehand is a key step to ensuring success in an SAP high-availability deployment on AWS. There are a number of items that should be considered up front. The Amazon VPC IP Address range/CIDR block and subnet ranges are completely user configurable. Throughout the examples in this guide the solution will leverage two separate Availability Zones and spread the SAP Application, database, and administrative services across both to maximize availability and durability. In most cases, two Elastic Network Interfaces (ENIs) will be attached to each instance, the first being associated with a "management" subnet, and the second with an "application" subnet. The following table should help in planning an SAP deployment into an Amazon VPC for this particular solution.

Item	Key Considerations	Example
Region	Latency requirements, distance from end users	*us-west-2*
Availability Zone 1		*us-west-2a*
Availability Zone 2		*us-west-2b*
VPC IP Range/CIDR Block	Ensure range does not overlap with existing internal IP range. Size IP range appropriate for number of hosts and planned growth	*192.168.0.0/16*
Management Network IP Subnet Range/CIDR block - Availability Zone #1	Size subnet should accommodate for growth	*192.168.1.0/24*
Application Network IP Subnet Range/CIDR block - Availability Zone #1	Size subnet should accommodate for growth	*192.168.2.0/24*

Application Network IP Subnet Range/CIDR block - Availability Zone #2	Size subnet should accommodate for growth	*192.168.3.0/24*
Management Network IP Subnet Range/CIDR block - Availability Zone #2	Size subnet should accommodate for growth	*192.168.4.0/24*
Setup VPN Tunnel	On premise router configuration, choice of VPN tunnel over Internet to Amazon VPC, or Amazon Direct Connect to VPN Gateway	
Active Directory (AD)	Consider using on-premise AD as primary with secondary in each Amazon VPC Availability Zone.	
DNS	Use on-premise DNS Server as primary with secondary in each AWS VPC Availability Zone.	
Bastion and/or Remote Desktop Gateway	For remote administration over an Internet connection	

To help isolate single points of failure, the distribution of the various SAP application and database components will be onto multiple AWS instances. The following high-level architecture diagram shows all application and network components. Each layer will be explored in detail throughout the remainder of this guide.

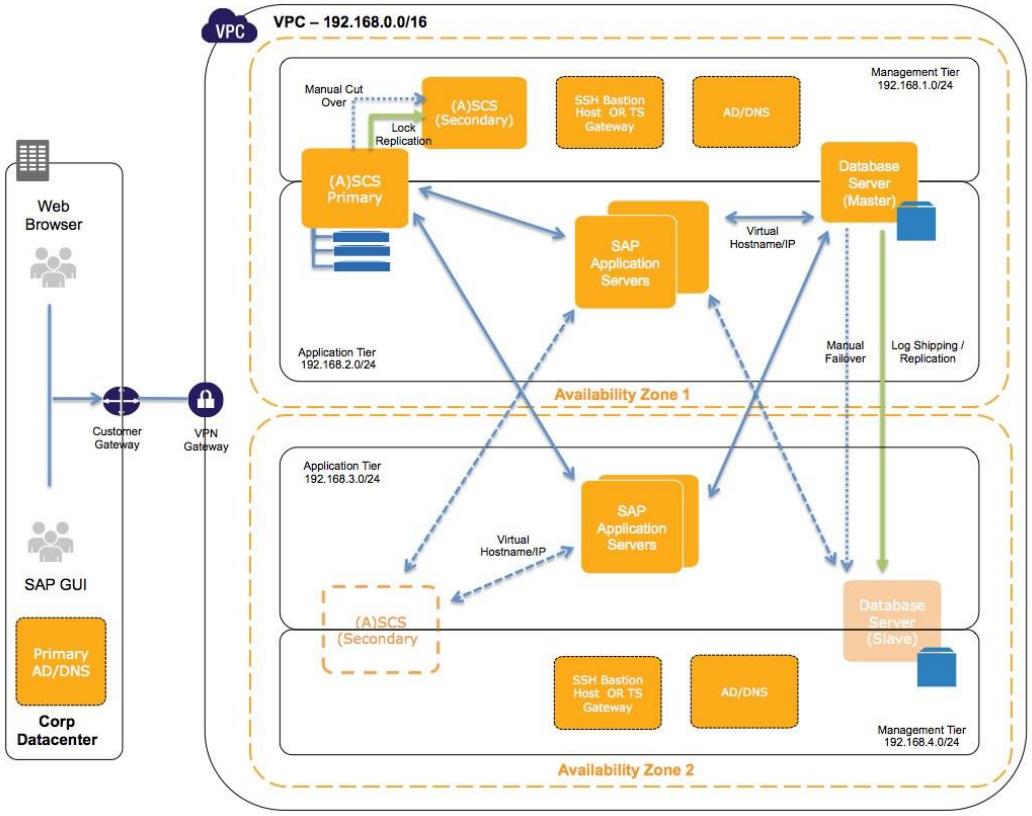


Figure 1 – SAP distributed landscape deployment

Security Groups

It's helpful at this point to define the security groups that are used for controlling access to instances for administrative functions, application and DB level communication, and isolating failed resources.

Note
Security groups are firewall rules that the user defines at the instance or network interface level to open or close specific ports for network communication. As the user, you will need to come up with your own set of rules and configure these based on your application connectivity, setup, and integration requirements. A fairly comprehensive list of TCP/IP ports used by SAP Applications has been made available by SAP.

It is strongly recommended that the SAP deployment team work closely with the networking team to understand what network traffic to allow in each tier and make configurations accordingly. The following ideas should help provide some structure and guidance:

- Set up a virtual private gateway and one customer gateway. These provide VPN connectivity between the corporate data center and the Amazon VPC.
- Set up route table configurations for all the traffic to and from the corporate data center over the VPN tunnel.
- Define all communication on required protocols and ports using network ACLs.

- Set up security groups on management servers with restricted access from certain on-premise networks or IP addresses.
- Set up security groups with limited inbound and outbound protocols and ports for each instance to be used at launch time.
- Configure additional security groups based on the specific SAP application and database communication requirements and assign these to secondary interfaces.

Note

Servers within a particular Amazon VPC subnet may need to access resources on the Internet for things such as software updates. Such actions can be accomplished by adding an Internet gateway to the VPC and using a network address translation (NAT) instance placed within a public subnet to protect internal resources. The other method is to create network routes to direct the traffic to traverse the VPN tunnel, into the corporate data center, and out through corporate proxy servers.

Installing the SAP Environment

Creating an Amazon VPC

Create an Amazon VPC in one of the available regions and specify a contiguous IP address range in CIDR block format (e.g., 192.168.0.0/16). It's important to choose a range that doesn't overlap with an already existing range being used internally on the corporate network. Next, create four new subnets, associating each with the new Amazon VPC, and split them across two different Availability Zones. To continue with the installation process, access to the instances deployed into the Amazon VPC will be necessary. This can be done either by using the VPN tunnel between the on-premise data center and the Amazon VPC, or by leveraging an Internet gateway combined with Elastic IP addresses (EIPs). In both cases, be sure to create appropriate route tables, associate them with the new subnets, and adjust the network ACLs with rules to meet internal security requirements.

Note

Additional subnets combined with network ACLs can further isolate or restrict access to different tiers of the SAP environment and also create multiple public and private subnets for isolation. Also note that any subnet that is associated with an Internet gateway will become a public subnet.

Once the Amazon VPC has been created, the next step is to deploy supporting infrastructure instances that will provide key services leveraged by the SAP environment from within the Amazon VPC. Some of these services might include:

- Active Directory services
- DNS
- Network address translation (NAT) services
- Remote Desktop gateways
- Bastion hosts
- Other

It's important to deploy these services in a highly available manner across Availability Zones. When supporting infrastructure services are in place, continue with the SAP deployments into the Amazon VPC.

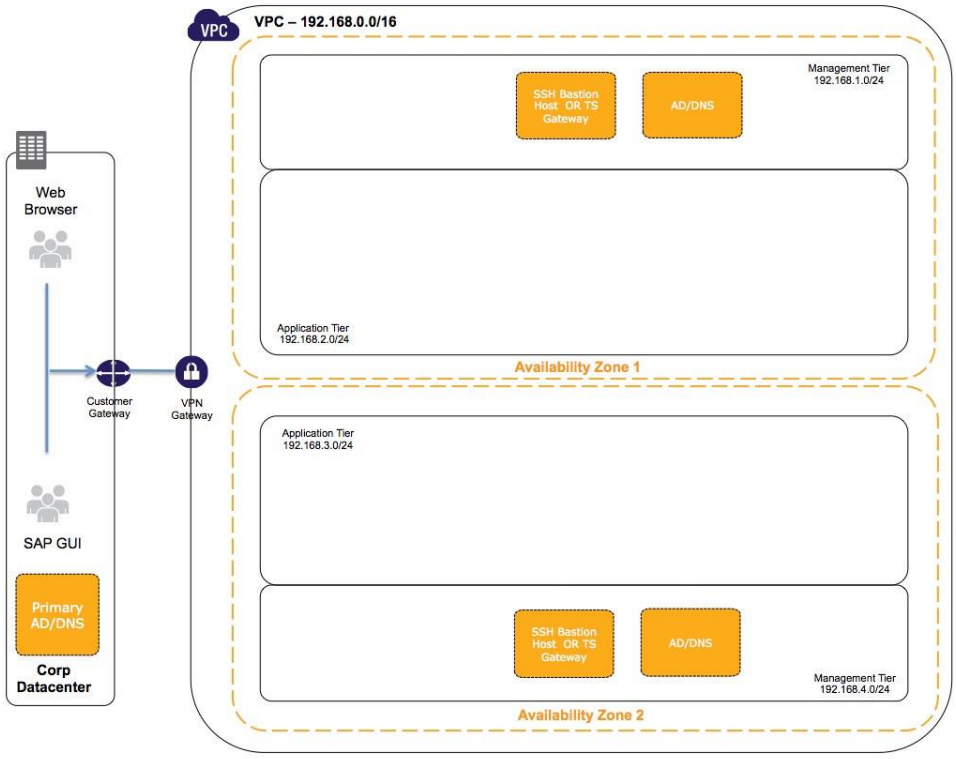
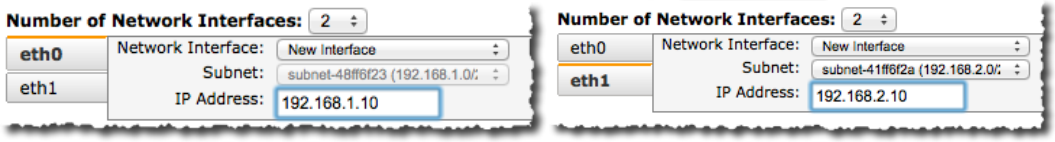


Figure 2 – Amazon VPC configured and ready for deployments

Preparing and Installing the First (A)SCS Instance

1. Launch a new Amazon EC2 Linux or Windows instance into Availability Zone 1 and specify IP addresses for both the primary (management) and secondary (application) level interfaces.



2. Next, choose to allocate Amazon EBS volumes, which will be used for both SAP local as well as global file systems. Volume sizes and the number will vary depending on installation needs.

   ✓ **Linux specifics**

   An example drive configuration for a Linux-based SAP central services instance might look like the following. A separate Amazon EBS volume is used for the local file systems as well as individual volumes for each global file system that will be moved to the secondary SAP central services host.

Device	File system used	Mount point(s)
/dev/sda1	local	/
/dev/sdf	local – (hostagent & ERS instance)	LVM (/usr/sap & /usr/sap/<SID>/ERS<##>)
/dev/sdg	relocatable	/export/usr/sap/trans
/dev/sdh	relocatable	/export/sapmnt/<SID>
/dev/sdi	relocatable	/usr/sap/<SID>/ASCS<##>

Tip

It is recommended to enter a 0 value for the sixth field entry (fs_passno) in /etc/fstab for any of the devices that could be relocated to another instance. If the value is non-zero, the operating system will attempt to fsck the file system upon start and will hang waiting for input from the console if the disk device is missing. Maintaining a value of 0 tells the fsck process to skip the file system. **IMPORTANT**: If an Amazon EBS volume has been force detached from an instance during a failure scenario, the file system should be manually checked once the drive has been attached to the failover (A)SCS host.

```
sapscsnode0:~ # cat /etc/fstab
/dev/sda1            /                    ext3       acl,user_xattr        1 1
proc                 /proc                proc       defaults              0 0
sysfs                /sys                 sysfs      noauto                0 0
debugfs              /sys/kernel/debug    debugfs    noauto                0 0
devpts               /dev/pts             devpts     mode=0620,gid=5       0 0
/dev/sap/usrsap      /usr/sap             ext3       acl,user_xattr        1 2
/dev/sap/sapers      /usr/sap/P00/ERS01   ext3       acl,user_xattr        1 2
/dev/sdg             /export/usr/sap/trans ext3      acl,user_xattr        1 0
/dev/sdh             /export/sapmnt/P00    ext3      acl,user_xattr        1 0
/dev/sdi             /usr/sap/P00/ASCS00    ext3     acl,user_xattr        1 0
```

Figure 3 - Example /etc/fstab entries

✓ **Windows specifics**

An example drive configuration for a Windows based SAP central services instance might look like the following. In this example, only two Amazon EBS volumes have been used. One is for the local directories leveraged by the host agent, ERS instance, etc. The other Amazon EBS volume is used for the global sapmnt share.

Device	File system used	Windows drive
/dev/sda1	local	C:
/dev/sdf	local – (hostagent & ERS instance)	D:
/dev/sdg	relocatable	E:

3. Next, specify an already existing security group or create a new security group for this instance. Consider associating a more restrictive administrative security group to the instance during launch time and assign a different security group customized for the SAP application to the ENI associated with eth1 shortly after launch.

4. Once the instance has been launched, continue host preparation steps as outlined in both the SAP NetWeaver Installation guides as well as the SAP on AWS Implementation guide to prepare the instance for installing an SAP system.

   ✓ **Additional Linux system configuration**

   a) Be sure to check and update the static IP addresses and DNS info for both interfaces.
   b) Update the information for DNS, and ensure hostname to IP address resolution is working properly.
   c) Because the secondary network interface (eth1) is on a different subnet, additional configuration may be needed to route the packets coming in the eth1 interface back out the same interface.

   As root: echo "100 SAPHA" >> /etc/iproute2/rt_tables
   ip rule add from 192.168.2.0/24 table SAPHA priority 100
   ip route add default via 192.168.2.1 dev eth1 table SAPHA

   Tip
   See Appendix B for an example script on how to automate this process each time the interface is brought up or down, or when the ENI is either attached or detached.

   d) Configure NFS exports for global/export/sapmnt/<SID> & /export/usr/sap/trans
   e) Consider configuration of autofs to automatically mount /sapmnt/<SID> & /usr/sap/trans from the virtual (A)SCS hostname (for example, scsvirthost.mysapdomain.com)

   ✓ **Additional Windows system configuration**

   a) Set the preferred and alternate DNS servers in the TCP/IP properties for each local area connection.
   b) Join the domain.
   c) Follow the instructions in OSS note 1564275 - How to install an SAP System or SAP components on Windows using virtual host names.

5. Finally, install the first SAP Central services, or (A)SCS, node using sapinst and the SAPINST_USE_HOSTNAME = <hostname> flag.

   a) Do not choose the highly available installation; instead choose Distributed System -> ASCS Instance. For example:

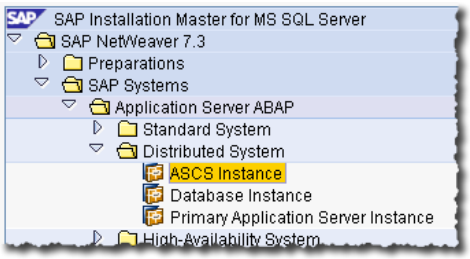


Figure 4 - Example SAP installer distributed system installation

- ✓ **Windows only** - Be sure to select the correct drives for the local host agent and SAPOSCOL log directory and choose the re-locatable EBS volume/drive Letter for the "SAP System Destination drive."

Tip
When performing the installation on Windows, it is highly recommended to use a Windows domain administrator account for the installation. If a local account is used, permission issues related to updating and/or installing files to the virtual NetBIOS sapmnt share are likely.

6. Once the installation is complete, verify that the virtual hostname has been maintained in both the SAP DEFAULT.PFL as well as the (A)SCS instance profile.

- ✓ **Linux**

```
sapscsnode0:p00adm 57> pwd
/sapmnt/P00/profile
sapscsnode0:p00adm 58> grep scsvirthost DEFAULT.PFL *ASCS*
DEFAULT.PFL:SAPGLOBALHOST = scsvirthost
DEFAULT.PFL:rdisp/mshost = scsvirthost
DEFAULT.PFL:enque/serverhost = scsvirthost
P00_ASCS00_scsvirthost:SAPLOCALHOST = scsvirthost
P00_ASCS00_scsvirthost:_PF = $(DIR_PROFILE)/P00_ASCS00_scsvirthost
```

Figure 5 - Sample SAP profile virtual host configuration – Linux

- ✓ **Windows**

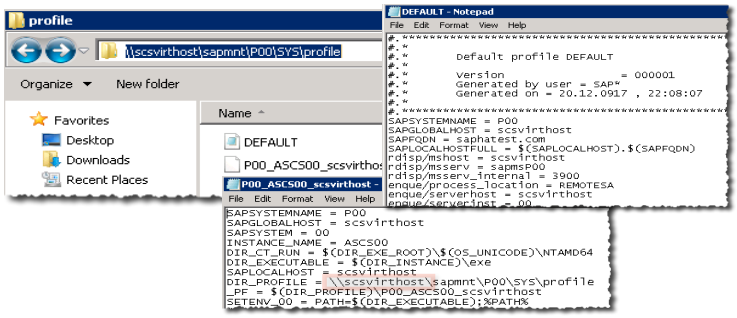


Figure 6 – Sample SAP profile virtual host configuration - Windows

7. Take a point in time image of the installation now by using the Create Image feature in the Amazon EC2 console. This will create a new Amazon Machine Image (AMI) based on the

selected instance's configuration. Copies of all attached Amazon EBS volumes will be stored in Amazon Simple Storage Service (Amazon S3) as snapshots.

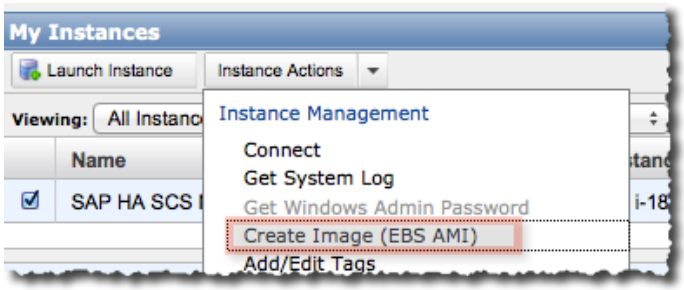


Figure 7 - Create Image in the AWS console

8. Once the AMI creation is complete, proceed to install the enqueue replication server instance on the primary (A)SCS host.

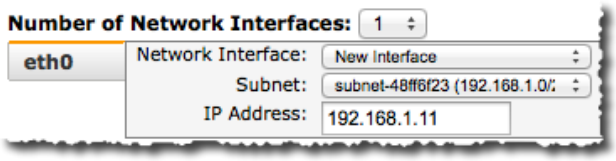
Configuring the Second Server for the (A)SCS Instance – Local Failover Scenario

The local failover scenario can protect the SAP central services instance from single component failures or degraded instances and can provide failover times very near traditional cluster methods.

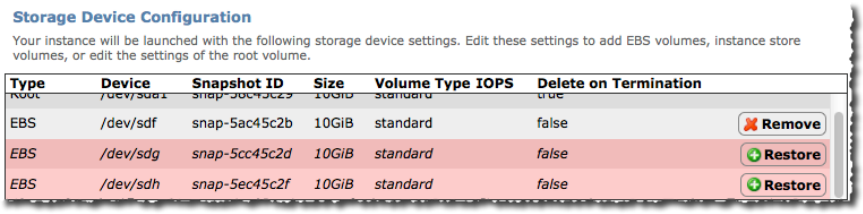
The installation and configuration steps required for the second instance is much less involved since a snapshot of all the work done previously has been captured in an Amazon Machine Image (AMI).

1. Launch a new instance based off of the AMI created of the primary (A)SCS instance into the same Availability Zone as the primary instance.

2. Configure the primary interface for the instance. The secondary or failover ENI was already created with the primary (A)SCS instance.



3. Remove all the Amazon EBS volumes that are associated with failover volumes from the launch console. To continue with the previous example, leave the /dev/sdf device since it's the local volume, but remove devices /dev/sd[g-i].



4. Once the system is up and running, log in and change the hostname.

5. Lastly, install and start the Enqueue Replication Server (ERS) also on this new failover node. Check for the following message in the dev_enrepsrv file, which is located in the work directory for the ERS instance to ensure that the lock table is being replicated from the main central services enqueue process to the ERS instance. This is important for maintaining transaction state should a failure occur.

```
[Thr 140168357275424] Thu Sep 20 22:35:10 2012
[Thr 140168357275424] profile    /usr/sap/P00/ERS01/profile/P00_ERS01_sapscsnode1
[Thr 140168357275424] hostname   sapscsnode1

[Thr 140168357275424] Thu Sep 20 22:35:11 2012
[Thr 140168357275424] Replication server start with instance number 00
[Thr 140168357275424] Enqueue server on host scsvirthost, IP-addr 192.168.2.10, port 50016
[Thr 140168357275424] ShadowTable:create: ShmCreate(,SHM_CREATE,len=43573360) -> 7f7b6de68000
[Thr 140168357275424] Connected to Enqueue Server and created repl. table with 57033 lines
```

Figure 8 - dev_enqsrv trace file

Testing the SAP (A)SCS Instance Failover – Local Failover Scenario

Now that the distributed SAP (A)SCS setup is complete, we can now test the local failover scenario.

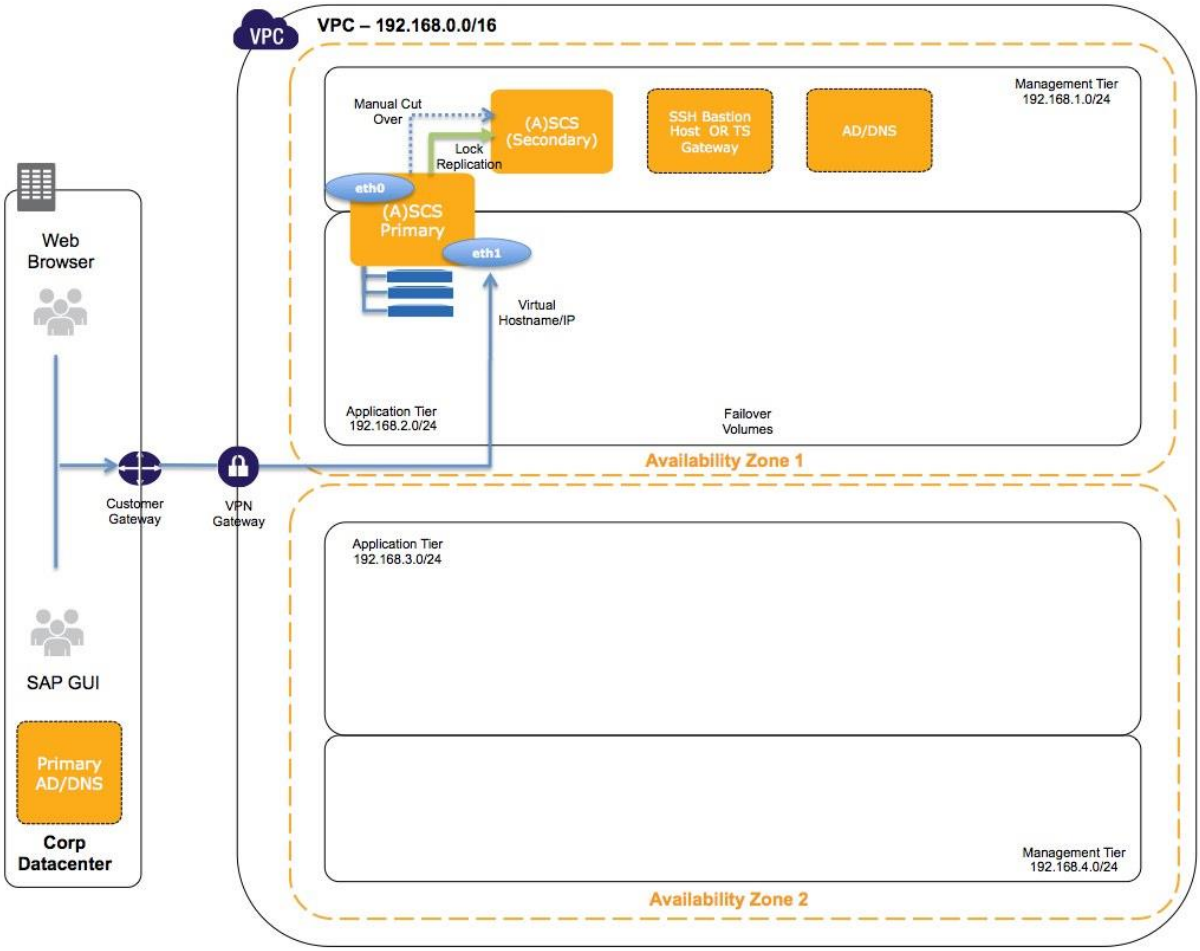


Figure 9 - Phase 1 (A)SCS and failover node architecture

To complete the manual failover, transfer the following resources from the primary (A)SCS node to the secondary node:

- Amazon EBS volumes containing the sapmnt, transport, and central services instance directories.
- Secondary ENI (eth1), which is assigned the virtual IP, associated with the virtual central services hostname.

The steps required to carry out these tasks are as follows:

1. When possible, gracefully shut down the primary (A)SCS node.
    a. Shut down the SAP (A)SCS instance using the MMC console or the stopsap command.

b. Kill all remaining processes that might be using resources associated with the file systems/volumes that will be moved.
c. If the (A)SCS host is based on Linux, unmount the /export/usr/sap/trans, /usr/sap/<SID>/ASCS##, and /export/sapmnt/<SID> file systems.
d. Stop the autofs process (if applicable).

2. Shut down the primary (A)SCS host or instance by using the AWS console or with the Amazon EC2 API command: **ec2-stop-instances *instance_id [--force]***

3. When the server is down, detach the previously identified Amazon EBS failover volumes. Either use the AWS console or use the following Amazon EC2 API command: **ec2-detach-volume *volume_id* [--instance *instance_id* [--device *device*]] [--force]**

Important
If you are unable to shut down the primary instance, you may have to force detach the Amazon EBS volumes. You can verify if the volumes have been successfully detached from the primary instance either in the console or with the Amazon EC2 API command: **ec2-describe-volumes**

4. Attach the volumes to the secondary instance with the console or the Amazon EC2 API command: **ec2-attach-volume *volume_id* --instance *instance_id* --device *device***

5. If the operating system is Linux, manually mount the file systems. The NFS server should already be configured to export the appropriate file systems. If, for some reason, the /sapmnt/<SID> mount has become stale and is not responding:
a. Manually unmount /usr/sap/trans and /sapmnt/<SID> file systems.
b. Restart NFS/Autofs
c. Export NFS shares with 'exportfs –a'
d. Mount up the NFS exports locally to /usr/sap/trans and /sapmnt/<SID> file systems, making sure to use the virtual hostname for the central services host.

6. Start the SAP central services or (A)SCS instance on the secondary node now.

7. Once this is up and running, detach the failover ENI from the primary instance using the AWS console or with the Amazon EC2 API command:
ec2-detach-network-interface *NETWORKATTACHMENT* -f, *--force*

8. Attach the failover ENI to this secondary instance using console or with the Amazon EC2 API command: **ec2-attach-network-interface *NETWORKINTERFACE* -i, --instance *INSTANCE* -d, --device-index *DEVICEINDEX***

Tip
This process can easily be automated by use of scripts on a witness server or a Bastion host.

The following diagram illustrates the local (A)SCS failover architecture and process.

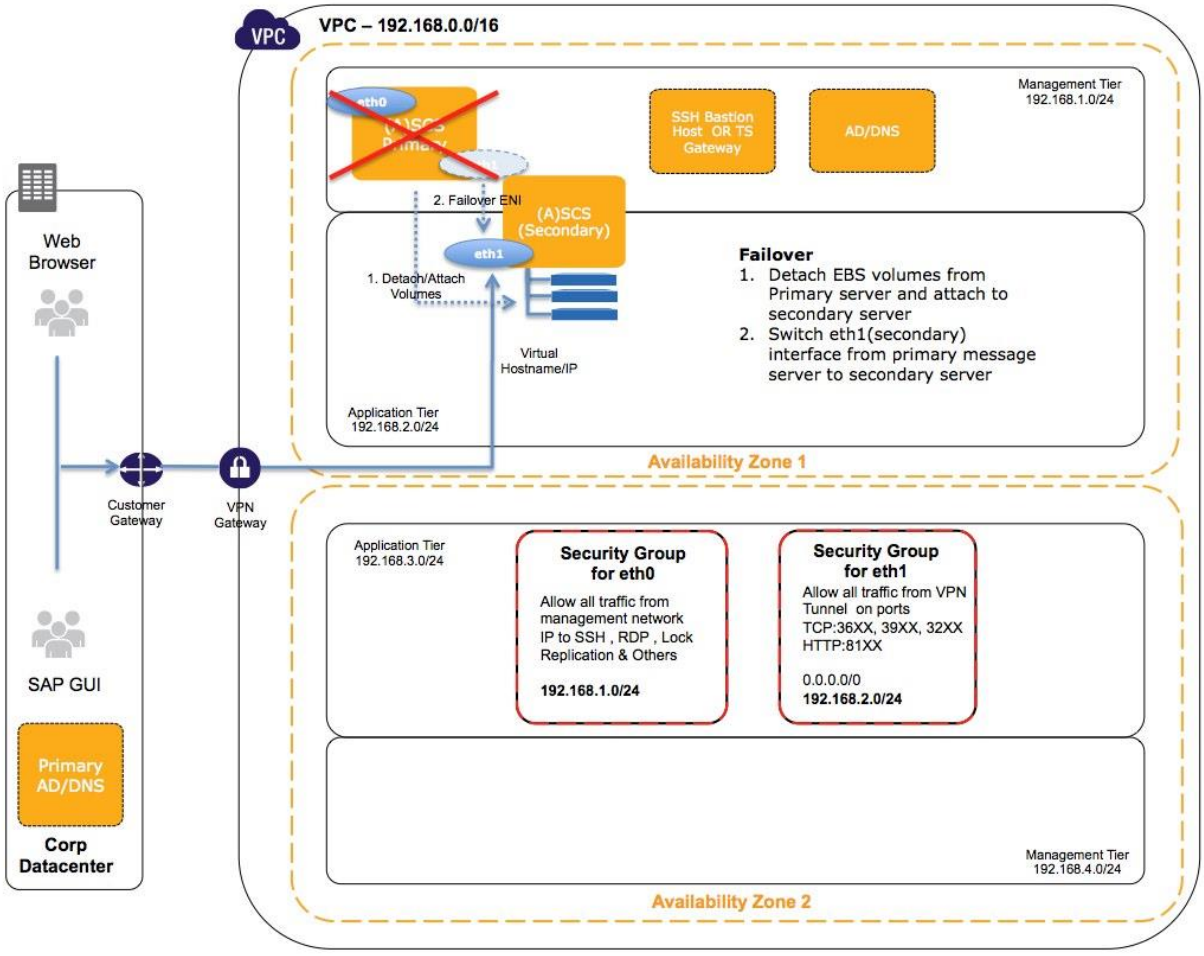


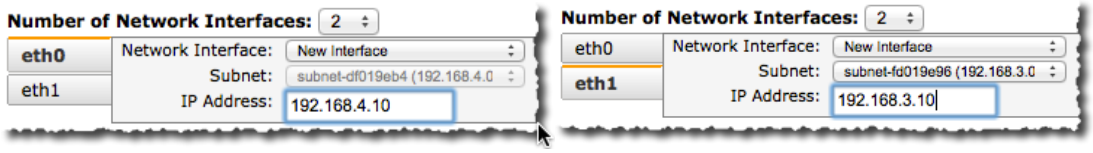
Figure 10 - Failover (A)SCS instance to local secondary instance

Configuring the Third Server for the (A)SCS instance – Multi-AZ Failover Scenario

With the capabilities of the AWS platform, we can also protect the central services instance from a much broader outage by installing a third central services instance in a second Availability Zone. The installation steps mirror those described previously with a few key differences.

1. Launch a new instance based off the AMI created of the primary (A)SCS instance, but this time into the second Availability Zone.

2. Configure both interfaces for this (A)SCS instance similar to the primary (A)SCS installation. Be sure to configure eth0 into the management subnet and eth1 into the application subnet.



3. Keep all of the volumes associated with the AMI.

4. Be sure to set the correct security group at launch time to restrict access at the instance level to only administrative functions. As before, assign a security group customized for the SAP application to the ENI associated with eth1 sometime after launch.

5. Once the system is up and running, log in and change the hostname.

6. (Optional) Install the Enqueue Replication Server (ERS) on this new failover node.

> **Important**
> Only one ERS instance can be connected to the primary enqueue service at a given time. Decide which ERS instance to operate based on SLA and availability targets.

The installation for the secondary central services instance is now complete. This instance can be started up in the event of a broader outage such as an Availability Zone failure. Since Amazon EBS volumes can't be moved from one instance to another in a different Availability Zone, further steps are necessary to ensure that any updates to the SAP profiles and files under the global transport directory are synchronized. This can be accomplished by a number of different methods:

- Take frequent snapshots of the primary global sapmnt and transport volumes. These can be used to create new volumes and subsequently attached to the failover instance in the secondary Availability Zone.
- Create an AMI based off the primary (A)SCS instance on a frequent basis.
- Use simple file system synchronization tools or create scripts to copy new or changed files.
- Leverage block level replication tools such as DRDB.
- Use backup/restore methods to ensure all updated files are placed onto the remote (A)SCS instance

Testing the SAP (A)SCS Instance Failover – Remote Failover Scenario

The failover method for this scenario is fairly straightforward:

1. If the secondary instance isn't already running, start it up or launch a new one from the most recent AMI created from the primary.
2. Update the DNS entry for the virtual (A)SCS hostname (scsvirthost in our example) to the IP address associated with the secondary interface on this remote SCS host.

3. (optional) As an added measure, detach the secondary ENI from the primary (A)SCS host to place a virtual network fence around this host if security groups are configured accordingly.

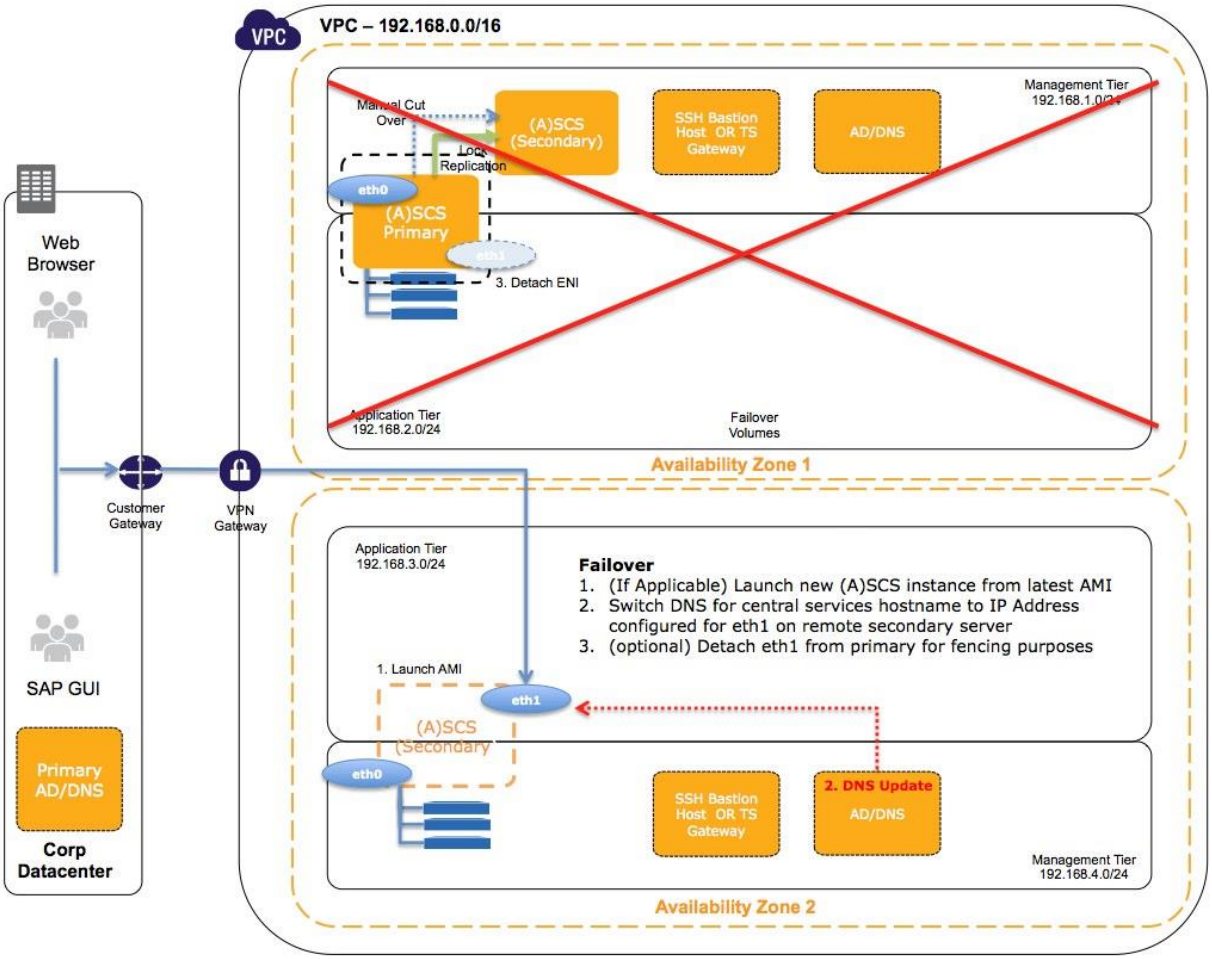


Figure 11 - Failover (A)SCS instance

Installing the Primary Database

Most if not all enterprise class databases used in an SAP environment have various options for configuring highly available solutions. Database high availability on AWS is accomplished by using native database replication techniques or even custom log shipping methods. In this case, each database instance has its own locally attached Amazon EBS volumes.

It is beyond the scope of this guide to cover all the various configuration options for each SAP supported database. However, please see Appendix A for a list of resources to help in planning and configuring the right database failover solution to meet availability requirements.

Once a database platform has been decided, proceed to install the primary database in the very same manner as the first SAP Central Services Instance:

1. Ensure that a virtual hostname is configured in DNS for the database.
2. Launch a new AWS instance with multiple ENIs, assigning the appropriate IP addresses.
3. As before, the security group assigned to the instance at launch should only allow limited access.
4. Assign a separate security group to allow database access from the SAP application to the secondary interface or eth1.
5. Configure the host for the database installation in a manner similar to the procedures outlined earlier in this guide as well as in other SAP on AWS and SAP installation guides.
6. Install the database leveraging the virtual hostname option with the SAP installer.
7. Lastly, ensure that the virtual hostname for the database has been configured correctly in the SAP default profile, DEFAULT.PFL, as well as any database client configuration for accessing the database.

Configuring the Secondary Database

There are a myriad of options for creating the second database instance. Installing the secondary database instance by leveraging an AMI created from the primary database instance may be one of the quickest methods and also helps ensure that operating system, database software, and file system configuration are the same as the primary database instance. This is usually a key requirement when setting up a warm standby or mirror database node.

There are also some key considerations that will affect a recovery point objective (RPO) and a recovery time objective (RTO), both of which should be decided based on business requirements. Some of the common items that need to be considered are:

- What is the update method for the stand-by database? Will updates be synchronous or asynchronous? Synchronous updates typically provide the maximum protection, as transactions typically are not considered “committed” until both database nodes have

confirmed that committed transactions have been applied, or, in some cases, just received by the stand-by node.

- Does the completion of all transactions matter? If asynchronous updates are configured, the commit operation of the primary database may not wait for the standby database to acknowledge receipt before completing the write process on the primary database. This has the potential to improve performance but also increases the risk of loss of in-flight transactions if a failure occurs.
- How long should you delay changes from being applied to the standby database? Choosing to delay updates to the standby database for a set period of time can help protect from human errors made within the SAP application. A delay of 3-4 hours can protect the standby database from applying one or more destructive transactions and give a database administrator a good chance to react.

Answers to these and other questions can only be decided after discussions with the business to set specific SLAs as well as common items such as a recovery point objective (RPO) and a recovery time objective (RTO).

Note
It is highly recommended to install the database failover or stand-by in the secondary Availability Zone. This increases the availability and durability of the database tier.

Testing Failover for the Database

The specific procedures for testing the failover mechanism for the database will be different depending on the database chosen for use in conjunction with the SAP deployment. However, the switchover functions from the primary node to the standby usually involves key functions such as:

- Failure detection
- Role transition where the stand-by becomes primary
- Applying pending transactions on the new primary node (where applicable)
- Allowing resumption of transactions by the SAP application

Once the database switchover, whether automatic or manual, has occurred, the ENI associated with the virtual database hostname/IP address should be detached from the primary database instance to ensure network isolation from the SAP application. This helps to avoid a split-brain scenario should the primary become available again unexpectedly.

This can be accomplished from within the AWS console or by using the Amazon EC2 API command:
`ec2-detach-network-interface` ***`NETWORKATTACHMENT`*** **`-f,`** ***`--force`***

For example: *#ec2-detach-network-interface eni-attach-869828ee –force*
ATTACHMENT eni-attach-869828ee detaching

Important
Assuming the stand-by database is in the secondary Availability Zone, update DNS to ensure that the virtual hostname associated with the database, as configured in the SAP DEFAULT.PFL profile, is associated with the correct IP address.

If security groups are set up appropriately, the end result is that the original primary database becomes virtually isolated from a network perspective, and the SAP application can resume communication with the database layer in the second Availability Zone.

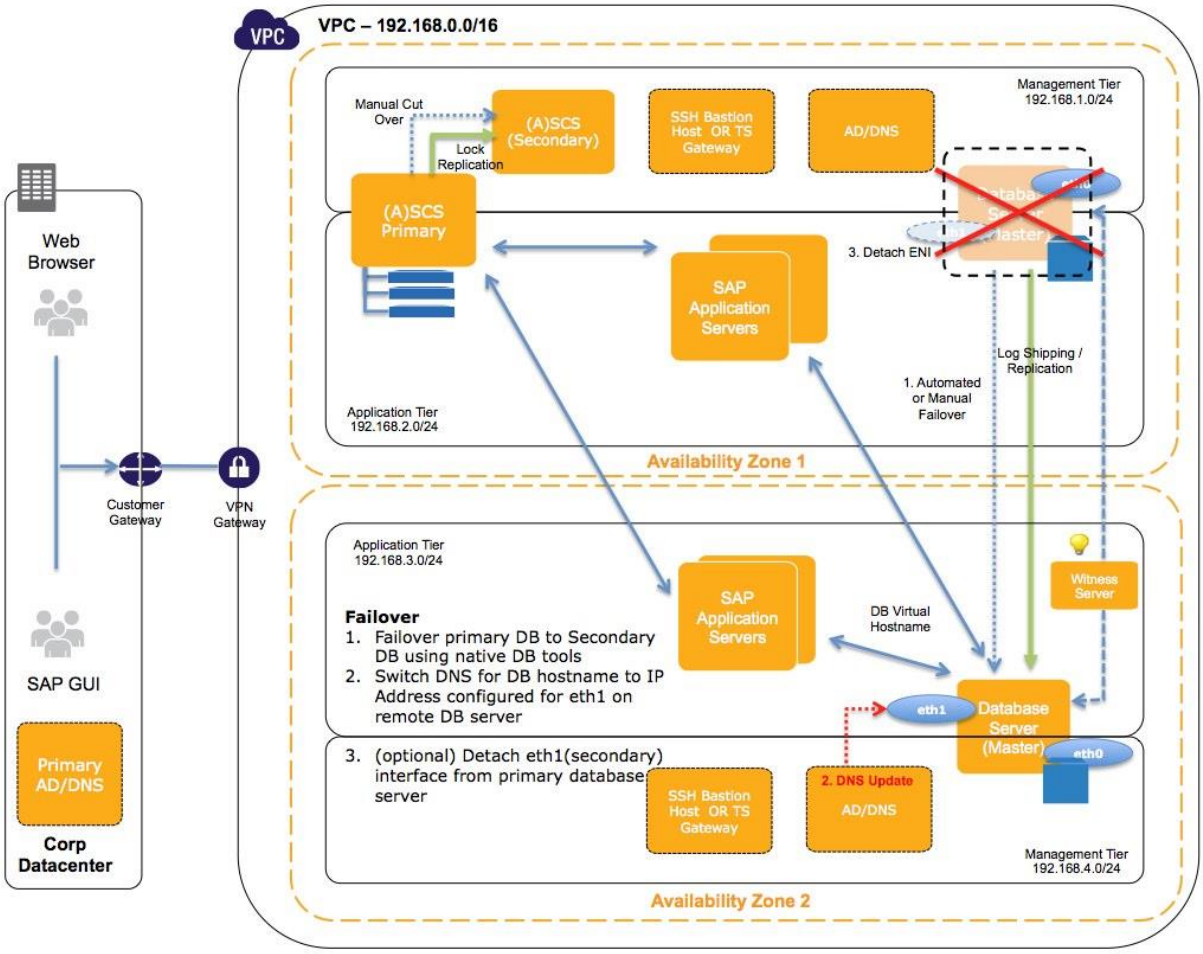


Figure 12 - Database Failover

Tip
When SQL Server database mirroring is used, a witness server can be installed to automate the failover process. Furthermore, SAP profiles can be adjusted to allow for automatic failover to the database mirror without requiring a DNS update. See SAP Note 965908.

Installing the Primary Application Server Instance and Subsequent Dialog Instances

The steps for installing the final piece of a distributed SAP landscape are very similar to steps completed previously. This process will be the same for the installation of the first "Primary application server" and subsequent dialog instance installations.

1. Launch a new AWS instance and prepare it for the installation of SAP software.
2. Configure the primary interface for the application subnet and assign the appropriate security groups.
3. Once the system is up and running, be sure to change the hostname, and check DNS and network settings.
4. When the installation/configuration items are complete, create an Amazon EBS-backed AMI of the fully configured SAP application server. This image will be used to create additional SAP application server instances in the event of a failure or even just to add additional capacity to the SAP system.
5. (optional) Prebuild a library of AMIs to be used for each application server type.

Install multiple SAP dialog instances in both Availability Zones to ensure application availability. The following diagram depicts how AMIs can be used to re-create SAP dialog instances in the event of single instance failures as well as a complete Availability Zone failure.

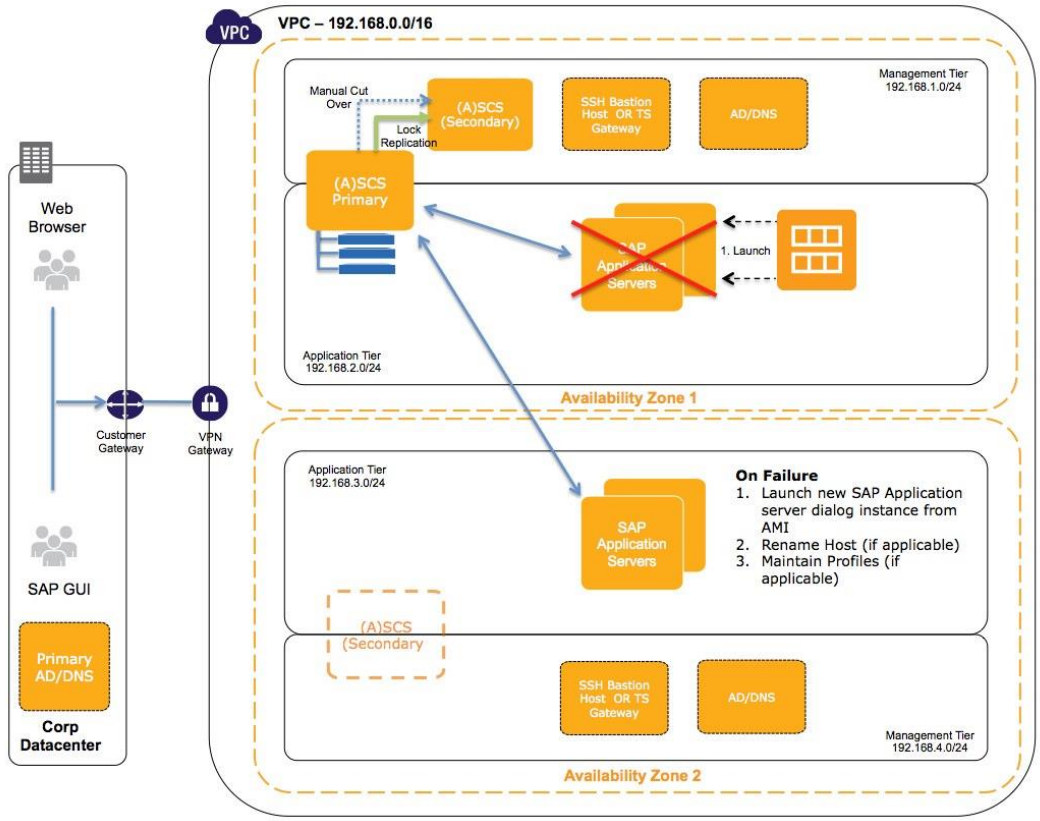


Figure 13 - SAP application server failure

Making the Web Tier Highly Available and External Access

For company-internal web-based SAP solutions, the SAP Web Dispatcher can be installed within the application subnet or on multiple SAP application servers for high availability. Please note that the appropriate route tables, network ACLs, and security groups will need to be configured. Company internal DNS or calling programs should be configured to route to both SAP Web Dispatchers for high-availability purposes. Both SAP Web Dispatchers should be configured to communicate to the message server by virtual hostname.

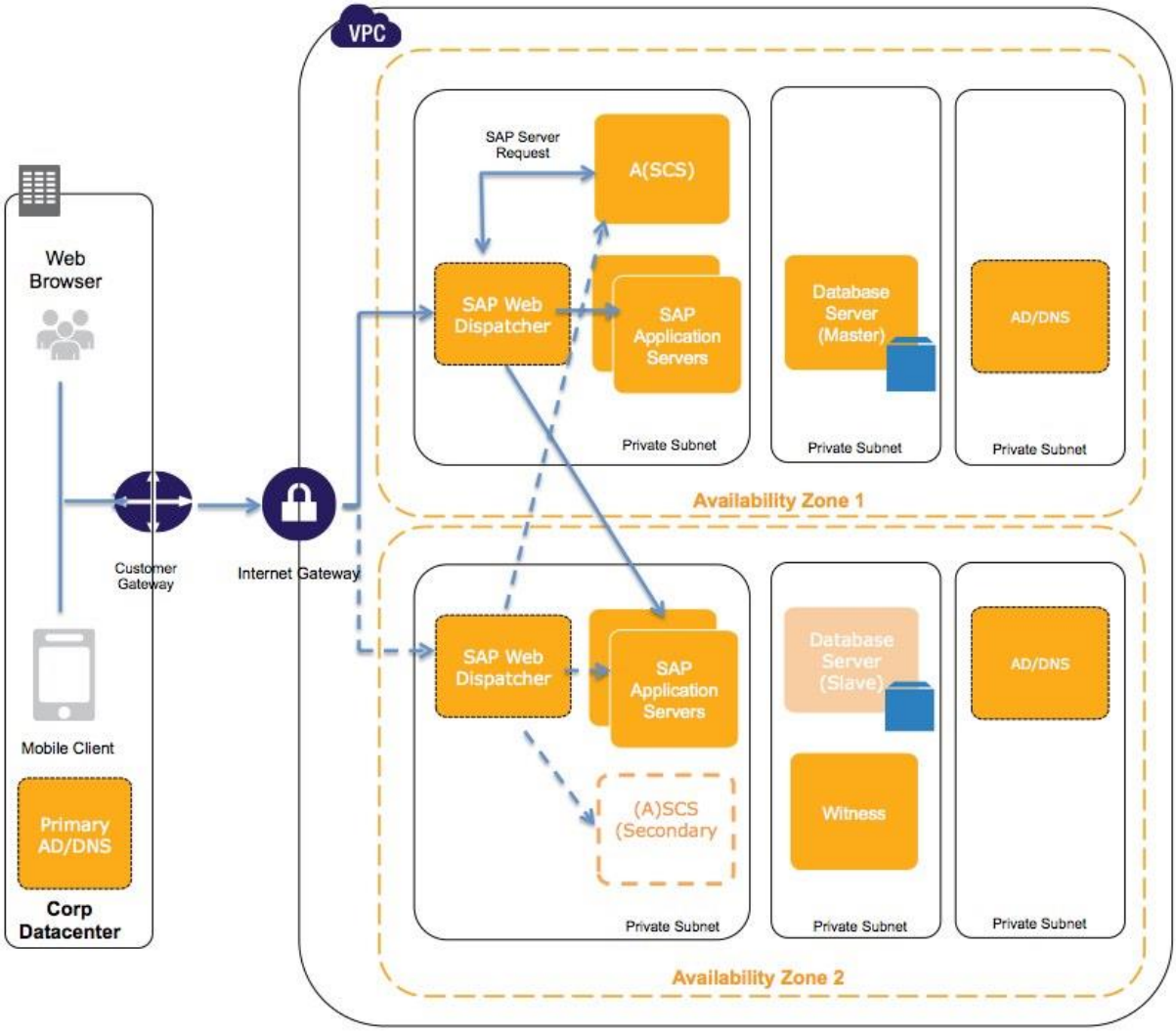


Figure 14 – Company-internal SAP web application

For public Internet-facing scenarios, SAP Web Dispatchers can be deployed in the public subnet to allow Internet-based traffic to reach the Web Dispatcher's public IP address using Elastic IP addresses (EIPs). Based on requirements, a NAT instance can also be added to allow the system's network to be able to reach the Internet for things such as updates or patches. A Bastion host can also be set up in the private management subnet and exposed with an Elastic IP address for systems management purposes. For added security, this Elastic IP address can be detached when not in use.

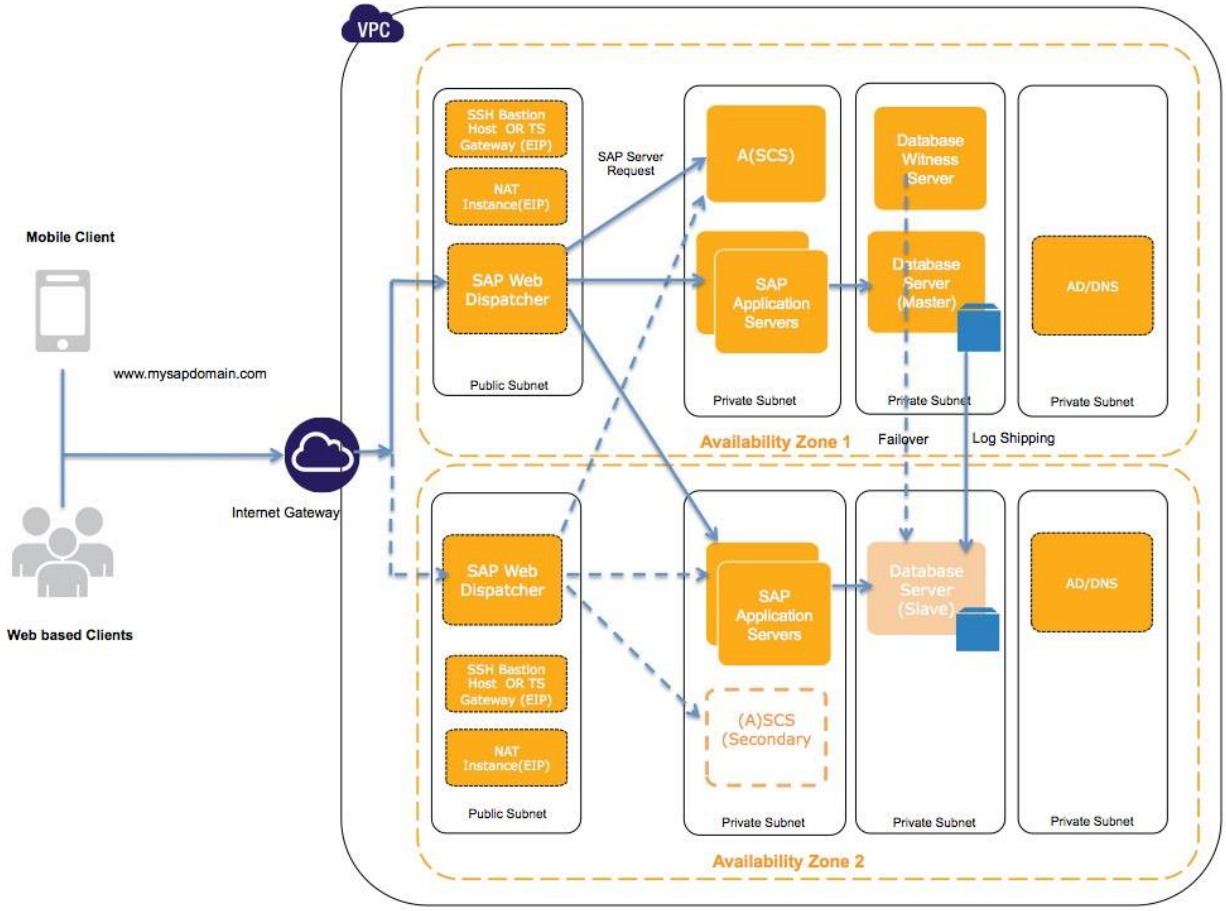


Figure 15 – Internet-based SAP web application

Summary

SAP customers now have the flexibility to deploy SAP solutions and landscapes on the scalable, on-demand Amazon EC2 platform in a highly available manner without having to invest in costly capital expenditures for the underlying infrastructure. By combining the flexibility of the AWS platform with SAP installation techniques, SAP customers can greatly improve the availability of their SAP deployments. For further information, including case studies of customers who have deployed SAP systems on AWS, please visit http://aws.amazon.com/sap.

Appendix A – Resources for Database HA solutions

SAP HANA

Three third-party vendors of automatic failover/takeover add on products for AWS:

SAP Note #	Description
2309342	SUSE Linux Enterprise High Availability Extension auf AWS
2302728	Supported scenarios with NEC Expresscluster on Amazon Web Services
1662610	Support details for SIOS Protection Suite for Linux

SQL Server

- SAP with SQL Server Best Practices Guide
- Microsoft TechNet - High Availability with SQL Server 2008 R2
- RDBMS in the Cloud: Microsoft SQL Server 2008 R2

SAP Note #	Description
965908	SQL Server Database Mirroring and SAP Applications
1550337	Database Mirroring causes blocking application instance

DB2

- SCN DB2 for LUW
- IBM HADR Redbook for DB2 on LUW
- DB2 for Linux, UNIX, and Windows – High Availability Disaster Recovery (HADR)

SAP Note #	Description
1612105	FAQ for DB2 High Availability Disaster Recovery (HADR)
1568539	HADR – Virtual IP or Automatic Client Reroute

MaxDB

- SCN SAP on MaxDB
- MaxDB Standby Database
- MaxDB Hot Standby (not recommend, but doable) – need to decide how to handle this.

SAP Note #	Description
952783	FAQ: MaxDB high availability
846890	FAQ: SAP MaxDB Administration

ASE

- SCN SAP on Sybase ASE
- Getting Started with Sybase ASE and the SAP System

SAP Note #	Description
1650511	High Availability Offerings with Sybase ASE (See Warm Standby Option & Log Shipping attachment)

Oracle

- SCN SAP on Oracle
- Oracle 11g Data Guard for SAP customers
- Data Guard 12c

SAP Note #	Description
105047	Support for Oracle functions in the SAP environment

Appendix B – Automating Default Route and Policy Configurations

Replace the IP addresses, default routers, and CIDR blocks used in this example with your individual settings. Items to be replaced with your individual settings have been underlined.

1. Add a table entry to /etc/iproute2/rt_tables on both (A)SCS HA nodes.
   echo "100 SAPHA" >> /etc/iproute2/rt_tables

2. Create a script to automatically create default route and policy upon attaching ENI on both nodes. The script will get executed when the interface is brought online.
   vi /etc/sysconfig/network/scripts/ifup.local.eth1

   ```
   #!/bin/bash
   if [ "$1" = 'eth1' ]
     then
     ip route flush table SAPHA
     ip rule add from 192.168.2.0/24 table SAPHA priority 100
     ip route add default via 192.168.2.1 dev eth1 table SAPHA
   fi
   ```

3. Make the script executable.
   chmod +x /etc/sysconfig/network/scripts/ifup.local.eth1

4. Create a symbolic link to this new script in the directory /etc/sysconfig/network/if-up.d
   cd /etc/sysconfig/network/if-up.d
   ln –s ../scripts/ifup.local.eth1 ifup.local.eth1

5. Create a script to clean up the route and policy if the interface gets removed.
   vi /etc/sysconfig/network/scripts/ifdown.local.eth1

```
#!/bin/bash
if [ “$1” = 'eth1' ]
  then
   ip route flush table SAPHA
   ip rule del from 192.168.2.0/24 table SAPHA priority 100
fi
```

6. Make this script executable.
   chmod +x /etc/sysconfig/network/scripts/ifdown.local.eth1

7. Create a symbolic link to this new script in the directory /etc/sysconfig/network/if-down.d
   cd /etc/sysconfig/network/if-down.d
   ln –s ../scripts/ifdown.local.eth1 ifdown.local.eth1

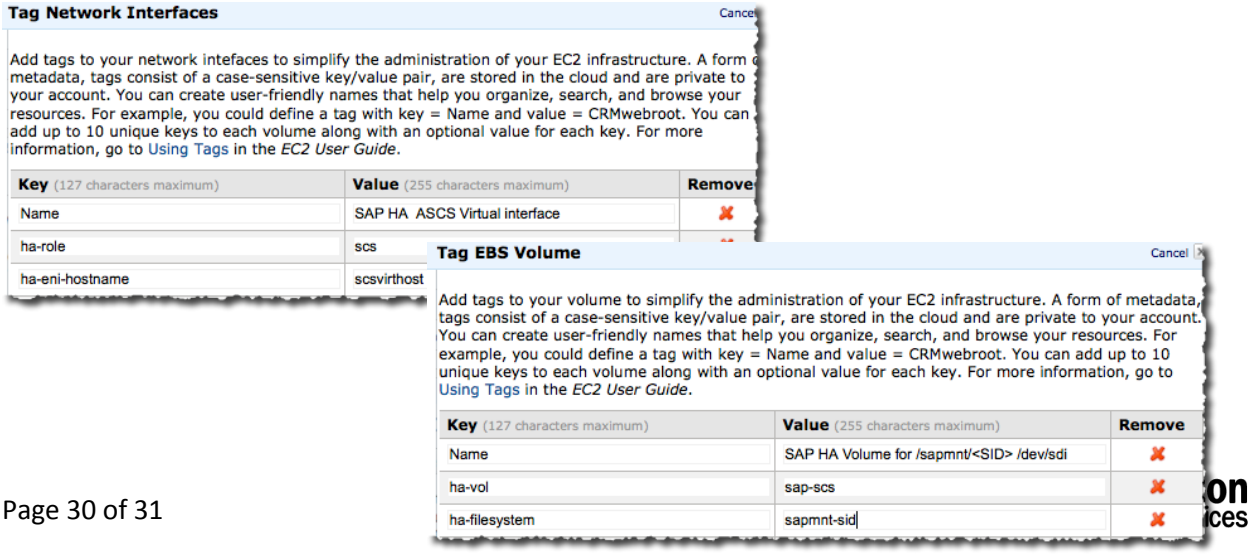
8. Test the setup by detaching and re-attaching the ENI interface in question.

Appendix C – Additional Tips

Licensing Considerations
Failover scenarios like the ones described in this guide, which require starting the message server on a new instance, may need to address the acquisition and installation of a new license. SAP has introduced an Amazon-aware licensing scheme which is described in detail in SAP Note 1178686. In short, setting the environment variable SLIC_HW_VERSION in the <sid>adm login shell will affect how the license key algorithm works. Multiple licenses may need to be maintained in order to support running the message server on multiple Amazon instances.

Tagging AWS Resources
Adding tags to the various AWS objects will not only make managing the SAP HA environment much easier but can also be used to quickly search for resources when executing a manual failover. Many Amazon EC2 API calls can be used in conjunction with a special tag filter.

For example:



Third-Party Software Components

There may be additional third-party components that are integral to running business processes in conjunction with an SAP environment. After determining actual requirements in this area, consider leveraging some of the concepts discussed in this guide such as:

- Installing third party software components on multiple instances
- Creating Amazon EBS backed AMI images of key third-party systems, so you can launch them on-demand in case of failures
- Using multiple interfaces to control access to certain software components
- Leveraging multiple Availability Zones for critical third-party software components